

An Unholy Trinity: TFNP, Polynomial Systems, and the Quantum Satisfiability Problem

Marco Aldi  

Department of Mathematics and Applied Mathematics, Virginia Commonwealth University, Richmond, VA, USA

Sevag Gharibian  

Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany

Dorian Rudolph  

Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany

Abstract

The theory of Total Function NP (TFNP) and its subclasses says that, even if one is promised an efficiently verifiable proof *exists* for a problem, *finding* this proof can be intractable. Despite the success of the theory at showing intractability of problems such as computing Brouwer fixed points and Nash equilibria, subclasses of TFNP remain arguably few and far between. In this work, we define two new subclasses of TFNP borne of the study of complex polynomial systems: Multi-homogeneous Systems (MHS) and Sparse Fundamental Theorem of Algebra (SFTA). The first of these is based on Bézout’s theorem from algebraic geometry, marking the first TFNP subclass based on an algebraic geometric principle. At the heart of our study is the computational problem known as Quantum SAT (QSAT) with a System of Distinct Representatives (SDR), first studied by [Laumann, Läuchli, Moessner, Scardicchio, and Sondhi 2010]. Among other results, we show that QSAT with SDR is MHS-complete, thus giving not only the first link between quantum complexity theory and TFNP, but also the first TFNP problem whose classical variant (SAT with SDR) is easy but whose quantum variant is hard. We also show how to embed the roots of a sparse, high-degree, univariate polynomial into QSAT with SDR, obtaining that SFTA is contained in a zero-error version of MHS. We conjecture this construction also works in the low-error setting, which would imply $SFTA \subseteq MHS$.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Quantum complexity theory

Keywords and phrases quantum complexity theory, Quantum Merlin Arthur (QMA), Quantum Satisfiability Problem (QSAT), total function NP (TFNP)

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.7

Related Version *Full Version:* <https://arxiv.org/abs/2412.19623> [3]

Funding *Marco Aldi:* supported in part by VCU Quest Award “Quantum Fields and Knots: An integrative Approach”.

Sevag Gharibian: DFG under grant numbers 432788384 and 450041824, the BMBF within the funding program “Quantum Technologies – from Basic Research to Market” via project PhoQuant (grant number 13N16103), and the project “PhoQC” from the programme “Profilbildung 2020”, an initiative of the Ministry of Culture and Science of the State of North Rhine-Westphalia.

Dorian Rudolph: DFG projects 432788384 and 563388236.

Acknowledgements We thank Niel de Beaudrap, Neal Bushaw, Bruno Grenet, David Gosset, Christian Ikenmeyer, Pascal Koiran, Grégoire Lecerc and Thomas Vidick for helpful discussions. We thank Simon-Luca Kremer for pointing out a mistake in an earlier version of the proof of Theorem 10. Some of the results in this paper were obtained while MA was visiting Paderborn University. MA is grateful for the hospitality and the excellent working conditions.



© Marco Aldi, Sevag Gharibian, and Dorian Rudolph;
licensed under Creative Commons License CC-BY 4.0

17th Innovations in Theoretical Computer Science Conference (ITCS 2026).

Editor: Shubhangi Saraf; Article No. 7; pp. 7:1–7:24



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

The genesis of this work consists of three elements: TFNP, Bézout’s theorem, and the quantum satisfiability problem. As such, we begin by giving background on these three. The Fundamental Theorem of Algebra’s role will then be introduced when stating our results in Section 1.1.

The first element: TFNP. The late 1980’s and early 1990’s witnessed the emergence [31, 41, 44] of a complexity theoretic framework which answered the question: *How can one characterize the complexity of problems for which an efficiently verifiable solution is guaranteed to exist, but finding this solution appears difficult?* Specifically, Total Function NP (TFNP) [41] was defined as the class of NP search problems with a guaranteed witness – in other words, the *decision* versions of these problems are trivial, so the challenge is “just” to find the witness. This definition encompasses numerous old-school mathematical principles – Brouwer’s fixed point theorem, for example, says that any continuous function f from a non-empty compact convex to itself has a fixed point (i.e. an x such that $f(x) = x$), but *finding* said fixed point appears difficult. Likewise, Nash’s theorem states that any non-cooperative game with a finite number of players and a finite number of actions has a Nash equilibrium, but efficiently finding a Nash equilibrium remains elusive.

Formally, to show that a given search problem $\Pi \in \text{TFNP}$ is intractable, one proves hardness of Π for one of the known subclasses of TFNP, each of which is itself based on an old-school mathematical principle. The five most prominent subclasses are [31, 44]:

- Pigeonhole Principle (PPP) corresponds to NP search problems guaranteed to have a solution via application of the *pigeonhole principle*.
 - Polynomial Parity Argument (PPA) leverages the *handshaking lemma*: In any finite undirected graph, the number of odd-degree vertices is even.
 - Polynomial Parity Argument on Directed Graphs (PPAD) uses the fact that any directed graph with an unbalanced node (meaning with in-degree \neq out-degree) must have another unbalanced node.
 - Polynomial Parity Argument on Directed Graphs with a Sink (PPADS) is identical to PPAD, except one requires finding an oppositely balanced node.
 - Polynomial Local Search (PLS) uses the fact that every directed acyclic graph has a sink.
- Although *a priori*, these subclasses appear to have nothing to do with (say) finding fixed points, appearances can be deceiving: Finding a Brouwer fixed point [44] and a Nash equilibrium [16, 13] are both PPAD-complete. Even the ubiquitous gradient descent algorithm has not escaped the reach of this framework – its complexity was shown PPAD \cap PLS-complete in a recent breakthrough work [19].

Unfortunately, beyond the “Big Five” subclasses above, defining genuinely new subclasses of TFNP has proven challenging. In fact, some of the handful of other known subclasses of TFNP have surprisingly recently turned out to equal *intersections* of the “Big Five”: $\text{CLS} = \text{PPAD} \cap \text{PLS}$ [19], $\text{EOPL} = \text{PLS} \cap \text{PPAD}$ and $\text{SOPL} = \text{PLS} \cap \text{PPADS}$ [25] (see also [38]).

The second element: Bézout’s theorem. In this work, we first define a new subclass of TFNP based on computing solutions to systems of multivariate polynomial equations, given a mathematical principle guaranteeing the existence of a solution. There is only one line of TFNP work we are aware of in a related direction, which we mention first to set context. Specifically, for *finite* fields, Papadimitriou [44] defined the problem CHEVALLEY by invoking

the Chevalley-Warning theorem, which states: Given is a system of polynomials $\{f_i\}_{i=1}^r$ over $\mathbb{F}_p[X_1, \dots, X_n]$ for finite field \mathbb{F}_p , where polynomial f_i has degree d_i . If $n > \sum_{i=1}^r d_j$, then the number of common solutions to the system is divisible by the characteristic p of \mathbb{F}_p . CHEVALLEY then asks: Given such a polynomial system and one solution, find a second solution. Although CHEVALLEY is known to be in PPA [44], it is not expected to be PPA-complete; however, two variants of CHEVALLEY have been shown PPA-complete [7, 26].

In this work, we instead consider polynomial systems over *complex* numbers. This necessitates a move from the domain of number theory to, for the first time in the study of TFNP, *algebraic geometry*. The old-school algebraic geometric principle we invoke is Bézout’s theorem from 1779, nowadays stated as follows: Over an algebraically closed field, any system of n homogeneous polynomials in $n + 1$ variables always has either an infinite number of solutions, or exactly $d_1 \cdots d_n$ solutions, for d_i the degree of the i th polynomial. For our purposes, we actually require a more recent *multi*-homogeneous extension due to Shafarevich [49], which gives a similar statement for the more general setting of systems of *multi*-homogeneous polynomials (Definition 32), which we now informally define.

Recall that a homogeneous polynomial is one whose non-zero monomials all have the same degree. A *multi*-homogeneous polynomial $p \in \mathbb{C}[x_1, \dots, x_n]$ generalizes this definition: One first partitions the variables $\{x_i\}$ into sets S_i as desired, and then requires that for each S_i , if we treat only the elements of S_i as variables, the resulting polynomial is homogeneous. For example, for variable sets $S_1 = \{x_1, x_2\}$ and $S_2 = \{y_1, y_2, y_3\}$, $x_1 y_1 y_2 + x_2 y_2 y_3$ is multi-homogeneous, whereas the homogeneous polynomial $x_1 + y_1$ is not. (Nevertheless, any homogeneous polynomial is trivially multi-homogeneous relative to the partition with one set S containing all variables.)

The multi-homogeneous Bézout theorem (Theorem 36) now first defines, corresponding to the product of degrees $d_1 \cdots d_n$ from the original Bézout theorem, a more general quantity known as the *Bézout number* $d_{Béz}$ (Definition 33). Then, it states that for any multi-homogeneous system of n equations $\{p_j\}_{j=1}^n \subseteq \mathbb{C}[x_1, \dots, x_{n+t}]$, where the variables are partitioned into t sets S_i , if $d_{Béz} > 0$, then the system has a solution. Note this generalizes Bézout’s theorem when all variables are placed into one set, S , so that $t = 1$. Roughly, our first new subclass of TFNP, denoted MHS (defined shortly in Definition 2), is the set of TFNP problems reducible to a multi-homogeneous system satisfying the multi-homogeneous Bézout theorem. Importantly, it can be efficiently checked if $d_{Béz} > 0$, which suffices for our purposes (Remark 34).

The third element: The quantum satisfiability problem. With two members of our trinity in hand, TFNP and Bézout’s theorem, we introduce the “unholy” member of the fellowship: The quantum satisfiability (QSAT) problem. We say “unholy” because of the unexpected nature of this trio – not only is this the first time quantum complexity and TFNP have been formally linked, but the classical Boolean satisfiability analogue of the problem we consider is a textbook example of an *easy* search problem. To elaborate on the latter, consider 3-SAT when the constraint system has a System of Distinct Representatives¹ (SDR). Then, for each clause $c_i = (x_i \vee y_i \vee z_i)$ of formula ϕ , one can “match” one of the variables in $\{x_i, y_i, z_i\}$ *uniquely* to c_i . Since no variable is matched twice in this process, setting each matched literal to true yields a satisfying assignment for ϕ . As an SDR can be found efficiently (e.g. via reduction to network flow [20]), the search version of 3-SAT with SDR is poly-time solvable.

¹ Given subsets $S_1, \dots, S_m \subseteq [n]$, an SDR is a set of distinct elements r_1, \dots, r_m such that $r_i \in S_i$ for all $i \in [m]$. In the context of 3-SAT, each S_i is the set of variables in clause c_i , and elements 1 through n correspond to the set of all variables.

The *quantum* analogue of this story has played out differently. Here, the Quantum Satisfiability problem (k -QSAT) on n qubits generalizes k -SAT, and is defined as follows: Given a set of projectors $\{\Pi_S\}_S$, each acting non-trivially² on some subset $S \subseteq [n]$ of qubits, does there exist an n -qubit quantum state $|\psi\rangle \in \mathbb{C}^{2^n}$ simultaneously satisfying all quantum clauses, i.e. $\Pi_S|\psi\rangle = 0$ for all Π_S ? First, the commonalities: Just as 3-SAT is NP-complete, 3-QSAT is QMA₁-complete [27], where QMA₁ is Quantum Merlin Arthur (QMA) with perfect completeness. See also [39, 46, 47] for recent progress on QMA₁-completeness of different QSAT variants.

Likewise, both 2-SAT [5] and 2-QSAT [4, 17] can be solved in linear time. Finally, for k -QSAT with SDR, Laumann, Läuchli, Moessner, Scardicchio, and Sondhi [35] (see also [36, 37]) showed that, like SAT with SDR, QSAT with SDR on qubits always has a solution. In fact, the solution is an NP witness, being a *tensor product* state (i.e. of form $|\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle \in (\mathbb{C}^2)^{\otimes n}$). And this is precisely where the stories diverge: Efficiently *finding* this tensor product state/NP witness for QSAT with SDR appears difficult.

There are two works in this direction to be mentioned now. In the positive direction, Aldi, de Beaudrap, Gharibian and Saeedi [2] gave a *parameterized*³ algorithm solving a special class of QSAT with SDR instances efficiently. In the opposite direction, Goerdt showed [24] QSAT with SDR and the additional restriction that only *real-valued* solutions are allowed is NP-hard. Thus, it remained unclear in which direction the complexity of QSAT with SDR should fall.

1.1 Our results

Briefly, our main contributions (denoted (b) and (c) below) are the definitions and complexity theoretic study of MHS and a second new TFNP subclass based on the Fundamental Theorem of Algebra (Theorem 41), denoted *Sparse Fundamental Theorem of Algebra (SFTA)*. However, the broader story of this paper involves the following sequence of results, which hold for any local qudit dimension $d \geq 2$: (a) QSAT on qudits has a product state solution if and only if the instance has a *weighted* SDR (WSDR). This yields containment in TFNP. (b) QSAT with WSDR on qudits is complete for MHS. (c) To better understand the complexity of MHS, as well as to build on the theme of TFNP subclasses related to complex polynomials, we show containment of SFTA into a zero-error version of MHS, and as a bonus, use this construction to obtain NP-hardness results for slight variants of QSAT with SDR. (d) Finally, special cases of QSAT with WSDR on qudits can be efficiently solved.

We now discuss our results in detail. Throughout, we refer to instances of QSAT by their interaction hypergraph $G = (V, E)$, where vertices correspond to qudits, and hyperedges to clauses. We do not restrict the type, number, or geometry of clauses allowed per qudit. A “clause” for us is⁴ a rank-1 projector.

a. Existence results via Weighted SDRs. We begin by introducing the new framework of *Weighted SDRs (WSDR)*, which underlies much of this work. Roughly, a WSDR (Definition 20) generalizes an SDR by introducing a *weight* function $w : V \rightarrow \mathbb{Z}_{>0}$, such that for any vertex $v \in V$ corresponding to a qudit, v can be matched to $w(v)$ clauses. Which weight function

² Formally, one sets $\Pi_S \otimes I_{[n]\setminus S}$ to ensure each projector acts on the correct space, \mathbb{C}^{2^n} .

³ “Parameterized” as in parameterized complexity, i.e. the runtime of the algorithm scales polynomially in the input size, but exponentially in structural parameters of the constraint hypergraph.

⁴ “Stacking” multiple rank-1 projectors to obtain a d -dimensional clause is allowed, but for clarity, we count this as d constraints. This is important for the definition of Weighted SDRs.

should one choose? In this work, when we say a given QSAT instance $G = (V, E)$ on n qudits of local dimensions d_1, \dots, d_n has a WSDR, we mean with respect to weight function $w(v_i) = d_i - 1$ for each $i \in \{1, \dots, n\}$. Thus, on n -qubit systems, a WSDR is just an SDR. Checking whether G has an WSDR can be done efficiently (Remark 28 in full version [3]).

Our first main result is that WSDRs are tightly connected to when a QSAT instance on qudits has a product state solution.

► **Theorem 1.** *Let $\Pi = \{\Pi_i\}$ be an instance of QSAT on n qudits of local dimensions d_1, \dots, d_n , respectively. If (G, w) admits a WSDR, then Π admits a satisfying product assignment. If (G, w) does not admit a WSDR and Π is generic, then Π has no satisfying product assignment.*

Theorem 1 is the qudit generalization of [35], which showed the analogous result for qubit systems with SDR. We thus have that for any $d \geq 2$, QSAT with WSDR on qudits is in TFNP. Above, “generic” (Definition 18) means “for almost all” instances. For example, 2-local constraints are generically entangled, whereas constraints in tensor product form are not. We remark that high-dimensional quantum systems are natural to study: From a computer science perspective, they can lead to surprising transitions in hardness (e.g. 1D Local Hamiltonian problem on qudits for $d \geq 8$ is QMA-complete [1, 29], whereas 1D Boolean Satisfiability on dits is in P via dynamic programming), and from a physics perspective, many natural systems (e.g. bosonic/fermionic systems) are high-dimensional systems.

With this said, while interesting in its own right, the primary appeal of Theorem 1 for us here is the techniques behind its proof, which will be crucial for our study of MHS. Specifically, we give two independent proofs of Theorem 1. The first (Section 4.1) is completely different than [35], and introduces the use of the Chow ring (Section 4.1) to obtain a simple proof of just a few lines. The second (Section 4.2) gives a poly-time mapping reduction from QSAT on qudits with WSDR to QSAT on qubits with SDR, and then plugs in [35]. This reduction, in particular, will play a key role in our MHS-hardness result of Theorem 3.

WSDRs beyond QSAT. As an aside, we demonstrate the power of WSDRs beyond the study of QSAT by using Theorem 1 to give a simple proof of a result of Parthasarathy [45], which says that any completely entangled subspace⁵ has dimension at most $\prod_{i=1}^k d_i - \sum_{i=1}^k d_i + k - 1$ (Corollary 49 in full version [3]).

b. A new subclass of TFNP based on Bézout’s theorem. We now discuss our first main result, for which we define our first subclass of TFNP, which involves *systems* of *low-degree, multi-variate* polynomial equations:

► **Definition 2** (Multi-homogeneous Systems (MHS) (Informal; see Definition 37)). *MHS is the set of total NP search problems poly-time reducible to finding an ϵ -approximate solution to a system $F = \{f_1, \dots, f_n\} \subseteq \mathbb{C}[x_1, \dots, x_{n+t}]$ of multi-homogeneous equations over \mathbb{C} with $d_{Béz} > 0$, where t is the number of subsets S_i partitioning the variable set. We require the size s of each S_i and degree d per monomial to be constant, and the precision ϵ must be at least inverse exponential.*

Comments regarding the constant bounds on the variable set size s and degree d : (1) This ensures $\text{MHS} \subseteq \text{TFNP}$ even for inverse exponential ϵ , since poly-time Turing machines can efficiently perform basic arithmetic with polynomial bits of precision. (2) For Theorem 3

⁵ A subspace is *completely entangled* if it does not contain any product states [3, Definition 48].

■ **Table 1** The complexity of variants of Classical SAT with SDR (denoted SAT with SDR) versus Quantum SAT with SDR (QSAT with SDR). Formally, “poly-time solvable” means in the complexity class Function Polynomial Time (FP), i.e. a poly-time classical Turing machine can compute a satisfying assignment.

Problem	Complexity	Reference
SAT with SDR	Poly-time solvable	Folklore (?)
QSAT with SDR	MHS-complete	This paper (Theorem 3)
SAT with SDR + $O(1)$ additional clauses	Poly-time solvable	This paper (Theorem 8)
QSAT with SDR + one additional clause	NP-complete	[24], this paper (Theorem 7)

below, $d \in O(1)$ is what yields constant locality k , whereas $s \in O(1)$ yields constant local dimensional qudits. (3) Formally, MHS is a union of complexity classes $MHS_{s,d}$ over all positive natural numbers s and d . (4) MHS does not obviously include general homogeneous systems as a special case due to $s \in O(1)$, i.e. one cannot trivially place all variables into one variable group. Finally, for precision ϵ , we shall utilize MHS_ϵ when we wish to specify a particular precision ϵ . We now show that QSAT with SDR is “MHS-complete”⁶:

► **Theorem 3** (Informal; formal statement in Theorem 39). *For any $\epsilon \in \Omega(1/\text{exp})$ and constant $d \geq 2$, computing an ϵ -approximate product-state solution to k -QSAT on qudits with WSDR is $MHS_{\Theta(\epsilon)}$ -complete.*

As even finding common roots of homogeneous polynomial systems in $n + 1$ variables and n equations remains an open problem [28], we interpret Theorem 3 as implying QSAT with SDR is intractable. Thus, we have the surprising juxtaposition that while classical SAT with SDR is easy, its quantum analogue is not.

c. A new subclass of TFNP based on the Fundamental Theorem of Algebra. To help understand the complexity of MHS, we give our second main result, which defines a second TFNP subclass, involving a *single, high-degree, univariate* polynomial equation. Below, a *sparse* polynomial (Definition 40), is one whose number of non-zero coefficients is logarithmic in its degree.

► **Definition 4** (Sparse Fundamental Theorem of Algebra (SFTA) (Informal; see Theorem 41)). *SFTA is the set of total NP search problems poly-time reducible to finding an ϵ -approximate root $r \in \mathbb{C}$ of a sparse monic univariate polynomial $p \in \mathbb{C}[x]$ of degree d , where $|r| \in [0, 1 + 2 \log(d)/d]$. We view d as exponentially large in the input size, and require $\epsilon \in \Omega(1/\text{poly}(d))$.*

As implied by its name, SFTA is inspired by the Fundamental Theorem of Algebra (Theorem 41), which recall states that any non-constant complex polynomial has a complex root r . Two comments regarding restrictions in the definition: First, the sparsity ensures⁷

⁶ We use the term “MHS-complete” in the introduction for simplicity, but the formal statement is more subtle (Theorem 39). In the case of MHS-hardness, for example, it says any problem in $MHS_{s,d}(\epsilon)$ can be reduced to solving k -QSAT on qubits with locality $k \geq (s + 1)^d$ within precision $\Theta(\epsilon)$, for $s, d, k \in O(1)$. In particular, to contain this k -QSAT instance in $MHS_{s',d'}$, we now need a larger value $d' \geq (s + 1)^d$. In other words, our reduction does not produce a fixed k which simultaneously yields hardness for all s and d . This is similar to how for each level Σ_k^p of the Polynomial-Time Hierarchy (PH), Quantified Boolean Satisfiability with $k - 1$ alternations (QBF $_k$) is Σ_k^p -complete, while problems simultaneously complete for all levels of PH are not known.

⁷ Another possible definition generalizing ours is to encode a non-sparse polynomial succinctly via a poly-size circuit which, given index i , outputs the i th coefficient of p . For us, however, the sparsity is necessary for our proof technique behind Theorem 5.

by definition that the degree d is exponential in the encoding size of polynomial p . This is important, as root approximations can be computed in $\text{poly}(d)$ time (see e.g. [48], as used in Section 4.4 of [2]), and thus the roots of a non-sparse polynomial can in general be efficiently approximated. Second, requiring $|r| \in [0, 1 + 2 \log(d)/d]$ is without loss of generality (Lemma 43), and is in fact necessary in order to prove $\text{SFTA} \subseteq \text{TFNP}$ (Theorem 44)⁸.

We now ask: *What is the relationship between MHS and SFTA?* We first conjecture $\text{SFTA} \subseteq \text{MHS}$, and are able to prove the following:

► **Theorem 5** (SFTA is in zero-error MHS (Informal; see Theorem 45)). *Let p be an s -sparse polynomial of degree d . Then, p can be efficiently reduced to an instance Π of QSAT with SDR of size $O(s \log(d))$, meaning $p(x/y) = 0$ if and only if $|v\rangle := |v_1\rangle \otimes \cdots \otimes |v_N\rangle$ is an exact solution to Π , for $|v_1\rangle = (x, y)^T \in \mathbb{C}^2$.*

In words, SFTA can be reduced to QSAT with SDR if we require $|v\rangle$ to *perfectly* satisfy all clauses, i.e. SFTA is contained in the version of MHS with error $\epsilon = 0$. (Recall, however, that we do not allow $\epsilon = 0$ in Definition 2, as the resulting class does not obviously allow poly-time verification of solutions.) We believe a more careful analysis of our construction behind Theorem 5 should yield the desired containment in MHS.

In the reverse direction, we believe $\text{MHS} \not\subseteq \text{SFTA}$. This belief notwithstanding, by leveraging an old result of Canny [11], we show that generic (Definition 18) instances of QSAT with WSDR *can* be embedded into the roots of a single, high-degree polynomial p (see Theorem 83 of the full version [3]). (In fact, one obtains something stronger, known as a *geometric resolution*, i.e. a set of rational functions $\{r_i\}$, so that when r_i is fed the j th root of p , it produces the i th amplitude of the j th solution to QSAT.) The polynomials p and r_i , however, are only poly-space computable, which is why this cannot yield $\text{MHS} \subseteq \text{SFTA}$.

NP-hardness results. Via the construction of Theorem 5, we can also show that even *slight* variants of QSAT with SDR are no longer in TFNP (assuming $\text{P} \neq \text{NP}$), but rather NP-hard.

► **Theorem 6.** *It is NP-hard to decide whether a 3-QSAT system with an SDR has a product state solution, such that $|x| = |y|$, where x, y are the entries of a prespecified qubit.*

► **Theorem 7** (c.f. [24]). *It is NP-hard to decide whether a 3-QSAT system with an SDR and one additional clause has a product state solution.*

The second result above was first shown by Goerdt [24] using different techniques.

Finally, to complete the picture, we show that in contrast to Theorem 7, classical SAT with SDR with $O(1)$ additional clauses again becomes easy! This mirrors precisely the behavior Theorem 3 exhibits for MHS-hardness of QSAT with SDR versus the fact that classical SAT with SDR is efficiently solvable; see Table 1.

► **Theorem 8** (Informal; see full version [3, Theorem 78]). *Given a SAT instance on n variables with an SDR, and k additional clauses, we can determine satisfiability in time $n^{k+O(1)}$.*

d. Efficiently solvable special cases of QSAT with WSDR. Since the MHS-completeness of Theorem 3 suggests QSAT with WSDR cannot be efficiently solved, the last part of this work rounds out our study by showing how to extend the parameterized algorithm of [2] in three different directions to solve new special cases efficiently.

⁸ For example, if d is exponential, then $p(2)$ can be *doubly* exponentially large, and thus not representable with polynomially many bits.

Our first two results here concern the qubit case, and are complementary. In this setting, [2] efficiently solves QSAT with SDR for generic (Definition 18) instances of *transfer type* $b = n - m + 1$ [3, Definition 85], where m denotes the number of constraints and n the number of qubits. Transfer type b means that we can find a vertex set of size b , and an ordering of the hyperedges (called *transfer filtration*), such that each edge adds at most one vertex. Recall *non-generic* instances allow constraints that are not entangled across some bipartite cuts.

We first show that the generic assumption can be dropped if one assumes an “almost extending edge order” [3, Definition 87], which in turn implies the existence of an SDR [2]. We say an ordering of the hyperedges is *k-almost extending* if all but k edges add a new vertex (for $k = 1$, we just say *almost extending*). Kremer [33] gives an algorithm to efficiently compute these.

► **Theorem 9** (Informal; see full version [3, Theorem 90]). *Let Π be a k -QSAT instance on qubits whose interaction hypergraph G has an almost extending edge order of radius r . Then an ϵ -approximate solution can be computed in time $\text{poly}(L, \log 1/\epsilon, k^r)$, where L is the input size.*

We then show that, instead of dropping the generic assumption, one can instead relax the transfer type assumption and still obtain a parameterized algorithm:

► **Theorem 10** (Informal; see full version [3, Theorem 93]). *Let Π be a k -QSAT instance on qubits whose interaction hypergraph G is k -uniform and has a $(k - 1)$ -almost extending edge order with radius r . Then an ϵ -approximate solution can be computed in time $\text{poly}(L, |\log \epsilon|, k^r, m^k)$, where L is the input size.*

Finally, we sketch how to extend the algorithm of [2] to QSAT on qudits with WSDR. This allows us to obtain an exponential speedup over brute force for solving a new high-dimensional, non-trivial (but artificial) infinite family of instances on *Pinwheel Hypergraphs* (Section 7.5.1 and Figure 5 in full version [3]).

1.2 Techniques

For brevity, we focus on our main results, (b) and (c). Brief technique overviews for (a) and (d) are given at the beginning of their respective sections, Section 4 and Section 7 of the full version [3].

b. A new subclass of TFNP based on Bézout’s theorem. For the MHS-completeness in Theorem 3, containment in MHS holds since PRODSAT can be written as a special case of solving multi-homogeneous systems as follows. In the case of 2-QSAT, for example, a tensor product state $|\alpha_1, \beta_2\rangle := |\alpha\rangle \otimes |\beta\rangle$ on two qubits satisfies a 2-local constraint $|\phi\rangle$ if and only if $0 = \langle \phi | \alpha_1, \beta_2 \rangle = \sum_{i,j \in [2]} \phi_{i,j}^* \alpha_i \beta_j$. The right hand side above is a multilinear polynomial in the amplitudes $\{\alpha_1, \alpha_2\}$ (respectively, $\{\beta_1, \beta_2\}$) of $|\alpha\rangle$ (respectively, $|\beta\rangle$). So, we will treat these amplitudes as variables in a system of multi-linear polynomials. The catch is that there is an independent normalization condition implicit on each qudit’s amplitudes; in our example here, both $|\alpha_1|^2 + |\alpha_2|^2 = 1$ and $|\beta_1|^2 + |\beta_2|^2 = 1$ must be independently satisfied. Since we will later work in projective space, however, this normalization is not explicitly enforced (other than the implicit constraint $|\alpha\rangle, |\beta\rangle \neq 0$). Instead, we must allow the amplitudes of $|\alpha\rangle$ and $|\beta\rangle$ to adhere to different “length scales”, since the assignments our system gives to them may lead to different norms for each vector. And now we come to why we require *multi-homogeneous* systems instead of homogeneous systems in this paper – recall that by

definition, a multi-homogeneous system allows us to partition variables into sets S_i , so that each polynomial is homogeneous with respect to each S_i . Thus, by setting S_i to represent the amplitudes of qudit i , we obtain that each quantum constraint is independently homogeneous with respect to each qudit i . (Each monomial will have degree 0 or 1, depending on whether the constraint acts on qudit i .) In other words, each qudit’s amplitudes implicitly has its own independent normalization.

As for hardness, to reduce multi-homogeneous systems to PRODSAT, the ideal aim is to represent each variable group by a single qudit. In other words, if variable group S_i contains n_i variables, we embed each variable as an amplitude of an n_i -dimensional qudit q_i . The first problem this presents is that monomials in a multi-homogeneous system need not be *linear* in each variable set S_i . To thus simulate non-linearity, we create multiple copies of each q_i ; by placing constraints on these simultaneously, we can create products of amplitudes from q_i . However, this raises a second challenge – this logic only holds when each copy of q_i has an *identical* assignment! The natural way to resolve this is to enforce equality between all copies of q_i by adding projectors onto the antisymmetric subspace. This, however, does not work for us, as the rank of the antisymmetric subspace for qudits with $d > 2$ is too large, requiring the addition of too many rank-1 constraints for an SDR to exist. To overcome this, we instead utilize the qudit-to-qubit reduction from our second proof of Theorem 1, which is a mapping iteratively replacing each d -dimensional qudit with a pair of 2- and $(d - 1)$ -dimensional qudits. Thus, each qudit is replaced with $d - 1$ qubits, and we show that the mapping preserves PRODSAT solutions. We are finally now in business, because on pairs of *qubits*, the projector onto the antisymmetric subspace is of rank 1, and thus we can show that there exists an SDR for the instance output by our reduction.

c. A new subclass of TFNP based on the Fundamental Theorem of Algebra. We discuss the proof of Theorem 5, which recall shows how to embed the roots of an arbitrary sparse polynomial p of exponential degree d into the solution set of a QSAT with SDR instance. The tool we start with is a *transfer function* (used also, e.g., in [9, 35]; see Lemma 14), which roughly is the quantum generalization of the following standard classical approach for propagating assignments: Given (e.g.) clause $(x \vee y \vee z)$, if $x = y = 0$, then $z = 1$ necessarily. Via this tool, we show how to design 2-local (respectively, 3-local) rank-1 QSAT constraints which force a target qubit to encode any desired *linear* (respectively, *quadratic*) operations on an input state $(x, y)^T$. For example, via a 2-local constraint $|\phi_{12}\rangle$ on qubits 1 and 2, we can enforce that if qubit 1 has assignment $(x, y)^T$, then in order to satisfy ϕ_{12} , qubit 2 must be set (proportional to) $(a_1x + a_2y, b_1x + b_2y)^T$, for any desired $|a_1|^2 + |a_2|^2 = |b_1|^2 + |b_2|^2 = 1$.

With these gadgets in hand, we then move to encoding input polynomial p into QSAT by designing three sets of clauses. To begin, we homogenize $p(x)$ to a bivariate polynomial $q(x, y)$, and let $|v_0\rangle = (x, y)^T$ denote an assignment to the first qubit. Ultimately, this x and y will end up encoding our roots to p . Our first set of constraints uses transfer functions and square-and-multiply to create new qubits of various powers of x and y , i.e. “power qubits” whose assignments must be proportional to $(x^i, y^i)^T$. Our second set of constraints then combines these power qubits with our transfer function gadgets to recursively construct $q(x, y)$ in a final target qubit, whose assignment must be proportional to $(q(x, y), y^d)^T$. The third set is a single constraint, which forces the target qubit’s state $(q(x, y), y^d)^T$ to be proportional to $(0, 1)$, which enforcing $q(x, y) = 0$. By “undoing” the homogenization, we can then show that $p(x/y)$ must be a root of p .

1.3 Discussion and open questions

Question and answer. As this work bridges rather disjoint areas of study (TFNP, polynomial systems, and quantum satisfiability), we address possible comments/questions to set further context.

1. *Are product state solutions to quantum satisfiability problems interesting?* Generally speaking, yes. Although solutions to quantum satisfiability problems are typically entangled, product state solutions have a long history of being used as an ansatz to study properties of local Hamiltonians (i.e. “quantum constraint satisfiability problems”) in the *mean-field theory* physics literature [22]. For example, mean-field ansatzes suffice to efficiently approximate ground state energies of planar [6, 8] and dense [23, 8] local Hamiltonians to within any desired relative error $(1 \pm \epsilon)$ for $\epsilon > 0$. In the case of 2-local frustration free Hamiltonians (as in 2-QSAT), *exact* product-state solutions always exist and can be found [10, 12], which has implications such as the fact that such Hamiltonians cannot be used to prepare resource states for one-way quantum computing [12].
2. *Why is adding SDRs to the picture interesting?* PRODSAT with SDR is interesting as it falls under the “dimer model” of physics [32], which is useful as it is (1) exactly solvable and (2) aids in understanding phase transitions, which are typically difficult to study. For example, the original motivation of [35] was to understand the SAT-UNSAT phase transition in random QSAT instances. Therein, dimer coverings/SDRs were used to show that for clause densities below a certain k -dependent threshold, random k -QSAT instances are satisfiable with probability 1 by a product state solution. While this did not perfectly resolve the exact SAT-UNSAT threshold, it significantly improved previously known lower bounds.
3. *Typically TFNP classes (e.g. PPAD) are defined via a complete problem whose input is a circuit succinctly encoding an exponentially large object (e.g. a circuit succinctly encoding an exponentially large graph for END-OF-LINE). On the other hand, MHS and SFTA, have their input explicitly written out?* This is a good discussion point. Traditional “syntactic” circuit-based definitions have the advantage that the existence principle for the class is captured by a simple combinatorial complete problem, which can make reasoning about the class easier. This, however, has a downside – proving hardness results for new problems *not* specified by input circuits, which are arguably more natural, can be more challenging (see, e.g. Göös, Kamath, Sotiraki and Zampetakis’ [26] non-circuit based PPA $_p$ -complete problem ($p \geq 3$ a prime) for the Chevalley-Waring theorem). In contrast, MHS and SFTA may be thought of as “white-box” TFNP subclasses, in that the object to be studied (i.e. polynomial equations) is specified explicitly, rather than succinctly via circuit. On the negative side, this has the downside of potentially obscuring the relationship between the class and the existence principle. On the positive side, it can bring establishing hardness results for further natural problems within reach, since the artificial circuit input encoding is bypassed. In our case, this motivation is further strengthened by the fact that MHS and SFTA are based on polynomials, which themselves are ubiquitous in the sciences, yielding a potentially promising route for characterizing the complexity of new TFNP problems.
4. *Is there also combinatorial principle underlying MHS?* Yes and no. No, in that the existence principle for MHS is Bézout’s theorem, which is algebraic geometric. Yes, in that checking if the Bézout number $d_{\text{Béz}} > 0$ boils down to checking if a certain bipartite graph has a perfect matching (see Observation 55 in the full version [3]). More generally, computing $d_{\text{Béz}}$ itself counts the number of perfect matchings in said graph (which is intractable, but also not necessary for our purposes).

5. *Can MHS or SFTA be related to existing TFNP subclasses?* This would be indeed ideal, but our attempts thus far have not succeeded. The most obvious candidate is PPAD, due to its connection [44] to Brouwer’s fixed point theorem. This is because there is a natural algorithm⁹ using transfer functions to attempt to solve QSAT with SDR; roughly, this algorithm aims to converge to a product state assignment which is a fixed point under all local transfer functions. Unfortunately, Brouwer’s theorem requires convex sets, and the set of product state solutions is *not* convex. Moreover, the standard approach of moving to the convex hull of product states (i.e. mixed separable states) seems to break the transfer function formalism. We thus leave this as what we feel is an important and interesting open question.

Conclusion and open questions. We have defined and studied two TFNP subclasses connected to complex polynomial systems. The first, Multi-Homogeneous Systems (MHS), leads to the first formal proof of a quantum problem which, on the one hand, is guaranteed to have a “simple” (i.e. tensor product) solution, and on the other hand, is potentially intractable. As even the “simpler” setting of finding common roots of homogeneous polynomial systems in $n + 1$ variables and n equations is believed difficult [28], we thus view MHS-hardness as a viable indicator for computational hardness. Our second class, Sparse Fundamental Theorem of Algebra (SFTA), was used to show that the problem of computing roots of sparse high-degree univariate polynomials can be embedded into computing exact solutions to QSAT with SDR, thus showing SFTA is contained in the zero-error version of MHS. We conjecture in fact that $SFTA \subseteq MHS$ – can this be shown?

As each member of the trinity studied here (TFNP, polynomial systems, and quantum satisfiability problems) is unto itself a research field, many questions in their intersection remain open. For example, which natural *classical* problems might be complete for MHS or SFTA? Are there other TFNP subclasses related to polynomial systems over complex numbers? As discussed in “question and answer” above, can MHS or SFTA be related to standard TFNP subclasses such as PPAD? Similarly, how is the setting of “syntactic” (i.e. circuit-based) TFNP subclasses to be understood versus our “white-box” setting for MHS and SFTA? Finally, a more subtle question is whether our definition of MHS as a union of complexity classes $MHS_{s,d}$ is necessary, or whether there is a fixed choice of (s, d) which suffices to capture the family of the union? Recall the roadblock here was that our reduction from $MHS_{s,d}$ to k -QSAT with SDR introduced a dependency in the locality k on (s, d) .

Organization. Due to space constraints, we can only give the technical statements including preliminaries for our main results in the remainder of this paper. We give proof sketches for Theorems 1 and 3. All other proofs are deferred to the full version [3], which also contains additional exposition, examples, and figures to aid understanding.

Section 2 states basic definitions, including formally defining QSAT, PRODSAT, and the connection between PRODSAT and polynomial systems. Section 3 introduces Weighted SDRs (WSDR), which are then used in Section 4 to give our two proofs of Theorem 1, i.e. that QSAT with WSDR always has a solution. Section 5 defines our class MHS and proves MHS-completeness of QSAT with SDR (Theorem 3). Section 6 defines class SFTA, studies its relationship to MHS, and gives the NP-hardness results of Theorem 6 and Theorem 7. Section 7 of the full version [3] gives efficient algorithms for special cases of QSAT with WSDR.

⁹ Due to David Gosset via private communication.

2 Preliminaries

We assume a basic background in quantum computation, see e.g. [43]. Basic background in algebraic geometry (e.g. definitions of projective space and varieties) would be helpful for Section 4.1 in particular, which introduces the Chow ring, though we have attempted to make this accessible with intuition throughout; see e.g. [49, 14] for references.

Notation and basic definitions. We use $:=$ to indicate a definition. For $|\psi\rangle \in \mathbb{C}^d$, we define $\|\psi\|_p := (\sum_{i=1}^d |\psi_i|^p)^{1/p}$. For a linear operator $M : \mathbb{C}^d \rightarrow \mathbb{C}^d$, we analogously define $\|M\|_p$ on the singular values of M . $\mathbb{C}[x_1, \dots, x_n]$ denotes the set of complex polynomials acting on variables x_1 through x_n . Throughout this work, we work with polynomials over \mathbb{C} , unless stated otherwise.

► **Definition 11** (Lipschitz continuity). *We say function $f : \mathbb{C} \rightarrow \mathbb{C}$ is K -Lipschitz continuous if for all $x, y \in X$, $|f(x) - f(y)| \leq K|x - y|$.*

► **Fact 12.** *Let $X \subseteq \mathbb{C}$ be such that $\forall x \in X, |x| \leq r$. Consider any complex polynomial $p = \sum_{k=0}^d c_k x^k$ of degree d , with s non-zero coefficients each of magnitude at most c . Then, over set X , p is K -Lipschitz continuous with $K = scr^{d-1}d$.*

Thus, when $c, d \in O(1)$, $K \in O(1)$. Note that Definition 11 and Fact 12 can be straightforwardly generalized to the setting of multivariate polynomials.

Quantum SAT. We begin by stating our basic formalism for QSAT on qudits. Formally, our QSAT Hamiltonians act on $\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \dots \otimes \mathbb{C}^{d_n}$ for some integers $d_1, \dots, d_n \geq 2$. As is standard, we fix a computational basis $\{|0\rangle, \dots, |d_i - 1\rangle\}$ for each qudit, so that an arbitrary vector in \mathcal{H} can be written $|\psi\rangle = \sum_{j_1=0}^{d_1-1} \dots \sum_{j_n=0}^{d_n-1} a_{j_1 \dots j_n} |j_1 \dots j_n\rangle$ for some choice of complex coefficients $a_{j_1 \dots j_n}$ satisfying $\sum_{j_1=0}^{d_1-1} \dots \sum_{j_n=0}^{d_n-1} |a_{j_1 \dots j_n}|^2 = 1$. (Since solutions to QSAT are null space vectors, the normalization of $|\psi\rangle$ will often not be important.)

► **Definition 13** (Quantum k -SAT on qudits (k -QSAT)). *For k -QSAT on n qudits:*

- *Input: A pair $\Pi = (\{\Pi_i\}_i, \alpha)$, for rational $\alpha > 1/p(n)$ for some fixed polynomial p , and for projectors or clauses $\Pi_1, \dots, \Pi_m \in \mathcal{L}(\mathcal{H})$ of the form $\pi^{-1}(|\psi_i\rangle\langle\psi_i| \otimes I_{n-k})\pi$, where π is a permutation of the qudits, $|\psi_i\rangle\langle\psi_i|$ is a rank-1 projector acting on the first k qudits, and I_{n-k} is the identity on the remaining $n - k$ qudits.*
- *Output: YES if there exists a unit vector $|\psi\rangle \in \mathcal{H}$ such that $\Pi_i|\psi\rangle = 0$ for all i , or NO if for all unit vectors $|\psi\rangle$, $\langle\psi| \sum_i \Pi_i |\psi\rangle \geq \alpha$.*

When working with QSAT, we use the concept¹⁰ of *transfer functions* on qubits from [2], for which we give a slightly simplified construction. Intuitively, a transfer function gives a necessary and sufficient condition for a rank-1 k -local clause $|\phi\rangle$ to be satisfied, given a partial assignment $|\varphi_1\rangle \dots |\varphi_{k-1}\rangle$ to its first $k - 1$ qubits.

► **Lemma 14** (Transfer function, g). *Let $|\phi\rangle$ be a k -local constraint on qubits. There exists a polynomial $g : (\mathbb{C}^2)^{k-1} \rightarrow \mathbb{C}^2$ such that, for any partial assignment v_1, \dots, v_{k-1} , the clause $|\phi\rangle$ is satisfied (i.e. $\langle\phi|v_1, \dots, v_k\rangle = 0$) iff¹¹ $|v_k\rangle \propto g(v_1, \dots, v_{k-1})$ or $g(v_1, \dots, v_{k-1}) = 0$. Moreover, g is linear in the coefficients of each v_i .*

¹⁰Transfer functions are a formal generalization of the transfer matrix formalism, which has appeared in previous works, e.g. [9, 36].

¹¹ \propto means up to scaling up to non-zero constant.

PRODSAT and homogeneous polynomial systems. In this paper, we interested in (approximate) *product* solutions to QSAT, for which one defines the following problem, ϵ -approximate PRODSAT.

► **Definition 15** (ϵ -approximate k -PRODSAT on qudits, decision version). *First, k -PRODSAT is defined as k -QSAT on qudits (Definition 13), except in the output the assignment $|\psi\rangle$ must be a pure tensor product state, i.e. $|\psi\rangle = |\varphi_1\rangle \otimes \cdots \otimes |\varphi_n\rangle$ with $|\varphi_i\rangle \in \mathbb{C}^{d_i}$ for each $i \in \{1, \dots, n\}$. Then, ϵ -approximate k -PRODSAT relaxes the YES case condition to $\langle \psi | \sum_i \Pi_i | \psi \rangle \leq \epsilon$.*

Our main results, i.e. involving MHS and SFTA, focus on the search version of this problem, for which we assume (as is standard for QSAT) that $k, d \in O(1)$:

► **Definition 16** (ϵ -approximate k -PRODSAT on qudits, search version). *Defined as ϵ -approximate k -PRODSAT, except in the YES case, a satisfying assignment $|\psi\rangle = |\varphi_1\rangle \otimes \cdots \otimes |\varphi_n\rangle$ with $\langle \psi | \sum_i \Pi_i | \psi \rangle \leq \epsilon$ is to be output. In terms of precision, recalling that m is the number of clauses, it suffices to output each entry of each $|\varphi_i\rangle$ within additive error $\epsilon/\text{poly}(m)$ to verify a YES case in NP (Remark 17).*

► **Remark 17** (Verifying ϵ -approximate k -PRODSAT in NP). Given $|\psi\rangle = |\varphi_1\rangle \otimes \cdots \otimes |\varphi_n\rangle$, we wish to verify $\langle \psi | \sum_i \Pi_i | \psi \rangle = \sum_i \langle \psi | \Pi_i | \psi \rangle \leq \epsilon$. For any i , suppose Π_i without loss of generality acts on qudits 1 through $k \in O(1)$. Then,

$$\langle \psi | \Pi_i | \psi \rangle = \langle \varphi_1 | \otimes \cdots \otimes \langle \varphi_k | \Pi_i | \varphi_1 \rangle \otimes \cdots \otimes \langle \varphi_k \rangle, \quad (1)$$

which only involves matrix multiplication on systems of dimension $d^k \in O(1)$, and thus can be computed using a poly-time Turing machine. Thus, if each entry of each $|\varphi_i\rangle$ is specified within additive error $\epsilon/\text{poly}(m)$, then for any i , Equation (1) can also be computed with additive error $\epsilon/\text{poly}(m)$. Note this holds even for inverse exponential ϵ , since the verification is on a classical Turing machine (as opposed to a quantum circuit verifier). Finally, since there are m clauses Π_i , and each clause is a projector (i.e. has spectral norm 1), the total additive error over all clauses can be upper bounded by ϵ .

To next connect PRODSAT with homogenous polynomial systems, expand both the qudits $|\varphi_i\rangle$ and the (possibly entangled) projectors Π_i with respect to the computational basis $|j_1 \cdots j_n\rangle$. Then, the problem of finding a satisfying assignment in product form is equivalent to solving a system of m homogeneous equations of the form

$$\sum_{j_1=0}^{d_1-1} \cdots \sum_{j_k=0}^{d_k-1} a_{j_1 \cdots j_k} x_{i_1, j_1} \cdots x_{i_k, j_k} = 0, \quad (2)$$

where i_1, \dots, i_k are the qudits on which the projector acts non-trivially, the constants $a_{j_1 \cdots j_k}$ the (complex conjugate of the) amplitudes of the rank-1 constraint Π_i , and each variable $x_{i,j}$ the j th amplitude of the i th qudit.

Projective space and algebraic geometric view of PRODSAT. In parts of this paper (particularly Section 4.1), it will be useful to view PRODSAT via the lens of projective space. Specifically, recall that vectors in \mathbb{C}^{d_i} differing by non-zero scaling represent the same physical state in the corresponding qudit, and that the property of being a non-zero null vector of a Hamiltonian is invariant under such scaling. Thus, PRODSAT solutions correspond to points in $(d_i - 1)$ -dimensional complex projective space $\mathbb{P}^{d_i-1}(\mathbb{C})$. (Formally, projective space treats two non-zero rays in the same direction as equivalent, regardless of their respective norms.)

The drop in dimension from d_i to $d_i - 1$ happens since one can rescale each qudit's local assignment $|\varphi_i\rangle \in \mathbb{C}^{d_i}$ so that its first amplitude is 1, and thus can be ignored. Of course, this assumes the assignment $|\varphi_i\rangle$ did not set its first amplitude to zero, which is generically the case (Definition 18), i.e. holds for almost all positive PRODSAT instances.

We thus have that n -qudit product states are in correspondence with points of the complex projective variety¹²

$$\mathcal{X}_{d_1, \dots, d_n} := \mathbb{P}^{d_1-1}(\mathbb{C}) \times \dots \times \mathbb{P}^{d_n-1}(\mathbb{C}). \quad (3)$$

In this geometric interpretation, each clause Π_i defines a hypersurface $V_i \subseteq \mathcal{X}_{d_1, \dots, d_n}$ which is of degree 1 in each of the variables corresponding to qudits on which Π_i acts nontrivially. As a consequence, the problem of finding a product solution to the given instance of QSAT is equivalent to the geometric problem of finding a point in the intersection $V_1 \cap V_2 \cap \dots \cap V_m$.

Finally, when we speak of *generic* instances of PRODSAT, we mean with respect to the following definition.

► **Definition 18** (Genericity [15, Def. 5.6]). *A property is said to hold generically for a set of polynomials f_1, \dots, f_n with indeterminate coefficients $c_{i,j}$ if there is a nonzero polynomial g in the $c_{i,j}$ such that the property holds for all f_1, \dots, f_n for which $g(\dots) \neq 0$.*

As mentioned above, “generic” means “for almost all” instances. A simple example of a property which holds generically is that of a 2×2 real matrix M being invertible. In this case, the polynomial g is the determinant $\det(M) = M_{11}M_{22} - M_{12}M_{21}$, since M is invertible if and only if $\det(M) \neq 0$.

3 Weighted Systems of Distinct Representatives (WSDR)

We now define a Weighted System of Distinct Representatives (WSDR), and prove several properties.

► **Definition 19** (Weighted hypergraph). *A weighted hypergraph is a pair (G, w) consisting of a hypergraph G and a weight function $w : V(G) \rightarrow \mathbb{Z}_{\geq 0}$.*

► **Definition 20** (Weighted System of Distinct Representatives (WSDR)). *A WSDR for weighted hypergraph (G, w) is a mapping $f : E(G) \rightarrow V(G)$, such that*

1. *(edges contain their representatives) for any $e \in E(G)$, $f(e) \subseteq e$,*
2. *(each edge has at least one representative) $|f(e)| \geq 1$ for all $e \in E$, and*
3. *(each vertex $v \in V(G)$ is the representative for at most $w(v)$ edges) $|f^{-1}(v)| \leq w(v)$ for all $v \in V(G)$.*

► **Definition 21** (Vertex set size with respect to a weight function). *Let (G, w) be a weighted hypergraph and let S a set of vertices of G . The size of S with respect to w is the integer $|S|_w := \sum_{v \in S} w(v)$.*

When does a weighted hypergraph have a WSDR? Hall's classic Marriage theorem gives a necessary and sufficient condition for when a (non-weighted) hypergraph has a (non-weighted) SDR. Here, we state its weighted case. As we were not able to find a proof thereof of such a statement in the literature, we provide one in Appendix A of the full version [3] for completeness.

¹²Roughly, a variety is simply the set of solutions to a given set of polynomial equations.

► **Theorem 22** (Hall’s Marriage Theorem for weighted hypergraphs). *Let (G, w) be a weighted hypergraph. For each collection X of edges of G , let V_X be the set of vertices that are contained in at least one edge of X . Then (G, w) has a WSDR if and only if $|V_X|_w \geq |X|$ for every $X \subseteq E(G)$.*

In the special case $w = 1$, Theorem 22 reduces to the usual Hall’s Marriage Theorem.

► **Corollary 23.** *Let (G, w) be a weighted hypergraph such that $\deg(v) \leq |e|_w$ for every $v \in V(G)$ and every $e \in E(G)$, where $\deg(v)$ denotes the degree of the vertex v . Then (G, w) has a WSDR.*

In uniform hypergraphs, precise necessary and sufficient criteria can be formulated as follows.

► **Definition 24** (k -Uniform Hypergraph). *A weighted hypergraph (G, w) is k -uniform for some positive integer k if $|e|_w = k$ for every $e \in E(G)$.*

► **Corollary 25.** *Let (G, w) be a k -uniform weighted hypergraph such that $\deg(v) = d$ for every $v \in V(G)$. Then (G, w) has a WSDR if and only if $d \leq k$.*

4 Existence results via Weighted SDRs

4.1 Approach 1: Via the Chow Ring

Brief overview of techniques. At a high level, the Chow ring approach uses intersection theory [21, 18, 49]. One reason for the effectiveness of this approach in the study of PRODSAT (i.e. product state solutions to QSAT) is that intersection theory is designed to work with generic constraints. This is in essence why important intersection-theoretic quantities, such as the Bézout number, are encoded into the interaction hypergraph. More concretely, the key property of the Chow ring we leverage is as follows (Fact 28): Given a set of rank-1 QSAT constraints with solution sets $\{V_1, \dots, V_r\}$ (formally, hypersurfaces), the Chow ring has a canonical mapping from each V_i to a “representative” of the Chow ring itself, denoted $[V_i]$. Then, if the product of these representatives is non-zero, i.e. $[V_1] \cdots [V_r] \neq 0$, one immediately has that $V_1 \cap \cdots \cap V_r \neq \emptyset$, i.e. the solution sets to each constraint share a common solution. Conversely, if $[V_1] \cdots [V_r] = 0$, generically, no joint solution exists.

We refer to [18, 21] for an in-depth discussion of the Chow ring of a variety. Here we limit ourselves to the multi-projective case which is relevant to PRODSAT. Recall we define $\mathcal{X}_{d_1, \dots, d_n} := \mathbb{P}^{d_1-1}(\mathbb{C}) \times \cdots \times \mathbb{P}^{d_n-1}(\mathbb{C})$.

► **Definition 26.** *The Chow ring of $\mathcal{X}_{d_1, \dots, d_n}$ is the commutative ring $CH(\mathcal{X}_{d_1, \dots, d_n}) = \mathbb{Z}[H_1, \dots, H_n]/(H_1^{d_1}, \dots, H_n^{d_n})$.*

This first proof of Theorem 1 will crucially use the notion of “representatives” $[V]$ of subvarieties V relative to the Chow ring. For this, let $Z(\mathcal{X}_{d_1, \dots, d_n})$ be the free abelian group of *cycles*, generated by subvarieties of $\mathcal{X}_{d_1, \dots, d_n}$. Linear combinations $n_1 V_1 + \cdots + n_k V_k$ with positive coefficients can be thought of as the union of n_1 copies of the subvariety V_1 , n_2 copies of the subvariety V_2 , etc.

► **Definition 27** (Subvariety representative, $[V]$). *There is a \mathbb{Z} -linear map $Z(\mathcal{X}_{d_1, \dots, d_n}) \rightarrow CH(\mathcal{X}_{d_1, \dots, d_n})$ that, to each subvariety V of $\mathcal{X}_{d_1, \dots, d_n}$, assigns an element of the Chow ring denoted by $[V]$. If V is a hypersurface of multidegree $(\delta_1, \dots, \delta_n)$ (i.e. cut out by a polynomial of degree δ_i in the homogeneous coordinates on $\mathbb{P}^{d_i-1}(\mathbb{C})$), then $[V] = \delta_1 H_1 + \cdots + \delta_n H_n$.*

Here is the key fact we will need about subvariety representatives.

► **Fact 28** (Sufficient criterion for non-empty intersection, and Bézout number). *If V_1, \dots, V_r are hypersurfaces in $\mathcal{X}_{d_1, \dots, d_n}$ such that $[V_1] \cdots [V_r]$ is non-zero, then $V_1 \cap \dots \cap V_r$ is non-empty. If $[V_1] \cdots [V_r] = 0$ then $W_1 \cap \dots \cap W_r = \emptyset$ for almost all hypersurfaces W_1, \dots, W_r such that $[W_1] = [V_1], \dots, [W_r] = [V_r]$ (i.e. each W_i has the same multidegree as the corresponding V_i). If $[V_1] \cdots [V_r] = NH_1^{d_1-1} H_2^{d_2-1} \cdots H_n^{d_n-1}$ for some positive integer N , then the generic intersection $W_1 \cap \dots \cap W_r$ consists of N points and N is referred to as the Bézout number.*

Theorem 1 now follows almost directly. Let $\Pi = \{\Pi_i\}$ be an instance of QSAT on qudits $|\varphi_1\rangle, \dots, |\varphi_n\rangle$ of dimensions d_1, \dots, d_n , respectively. Recall the weighted hypergraph (G, w) with $V(G) = \{v_1, \dots, v_n\}$, $E(G) = \{e_1, \dots, e_m\}$ such that $v_i \in e_j$ if and only if the clause Π_j acts non-trivially on the qu- d_i -it $|\varphi_i\rangle$. The weight function w encodes the information regarding the dimension of the qudits, namely $w(v_i) = d_i - 1$ for each $i \in \{1, \dots, n\}$.

► **Theorem 1.** *Let $\Pi = \{\Pi_i\}$ be an instance of QSAT on n qudits of local dimensions d_1, \dots, d_n , respectively. If (G, w) admits a WSDR, then Π admits a satisfying product assignment. If (G, w) does not admit a WSDR and Π is generic, then Π has no satisfying product assignment.*

Proof. Let V_i be the hypersurfaces corresponding to the clauses Π_i , $i = 1, \dots, m$. Since V_i is of degree 1 in the variables corresponding to the qubits on which Π_i acts non-trivially and of degree 0 in the remaining ones (see Equation (2)), its image in the Chow ring is

$$[V_i] = \sum_{v_j \in E_i} H_j. \quad (4)$$

Hence,

$$\prod_i [V_i] = \sum_{v_{j_1} \in E_1, \dots, v_{j_m} \in E_m} H_{j_1} \cdots H_{j_m}, \quad (5)$$

which is non-zero if and only if there is a summand in which each H_j appears at most $d_j - 1$ times i.e. if and only if (G, w) has a WSDR. The claim now follows from Fact 28. ◀

Actually, the proof shows an additional fact, which we will utilize in Section 4.2:

► **Corollary 29** (Counting number of SDRs and product solutions). *Let N denote the Bézout number. By the proof above of Theorem 1, if Fact 28 applies (i.e. $\prod_i [V_i] = NH_1^{d_1-1} \cdots H_n^{d_n-1}$), then N equals both the number of WSDRs on (G, w) , as well as the generic (and minimum, when counted with multiplicity) number of product solutions to any instance of QSAT with underlying weighted hypergraphs (G, w) .*

► **Observation 30.** *If in Theorem 1, the number of clauses matches the total degrees of freedom, meaning if $m = \sum_{i=1}^n d_i - 1$, then $\prod_i [V_i] = NH_1^{d_1-1} \cdots H_n^{d_n-1}$ for $N \in \mathbb{N}$.*

4.2 Approach 2: Reduction to qubits

We next give a completely different proof of Theorem 1, this time via direct reduction from a Hamiltonian with a weighted SDR on qudits to a Hamiltonian with an SDR on qubits (and subsequently using [36]). The result follows from the main theorem of this section, Theorem 31, through which a qubit Hamiltonian can be constructed by iteratively replacing a $(d + 1)$ -qudit by a qubit and a d -qudit, while preserving the existence of a WSDR. This second proof approach will also prove important later for our second main result on TFNP in Section 5.2.

► **Theorem 31.** *Let Π be a QSAT instance on a Hilbert space $\mathcal{H} = \mathbb{C}^{d+1} \otimes \bigotimes_{i=2}^n \mathbb{C}^{d_i}$ whose underlying weighted hypergraph (G, w) has a WSDR. There exists a linear-time constructible QSAT instance Π' on Hilbert space $\mathcal{H}' = \mathbb{C}^2 \otimes \mathbb{C}^d \otimes \bigotimes_{i=2}^n \mathbb{C}^{d_i}$ whose underlying weighted hypergraph (G', w') also has a WSDR. Given a product state solution to Π' (Π), we can compute a product solution to Π (Π') in polynomial time.*

5 Low-degree, multi-homogeneous systems and TFNP

5.1 Definitions and Bézout's Theorem

We begin with a formal definition of a multi-homogeneous polynomial. (For clarity, recall we consider polynomials over \mathbb{C} in this work.)

► **Definition 32** (Multi-homogeneous polynomial [42]). *A polynomial f is multi-homogeneous if there are m sets of variables $Z_j = \{z_{0,j}, \dots, z_{n_j,j}\}$ and $d_1, \dots, d_m \in \mathbb{Z}_{\geq 0}$ with at least one $d_j > 0$ such that*

$$f = \sum_{\substack{I_1, \dots, I_m: \\ \forall j \ |I_j|=d_j}} a_{I_1, \dots, I_m} Z_1^{I_1} \cdots Z_m^{I_m}, \quad (6)$$

where $I_j = (i_{0,j}, \dots, i_{n_j,j}) \in \mathbb{Z}_{\geq 0}^{n_j+1}$, $|I_j| := \sum_{k=0}^{n_j} i_{k,j} = d_j$, $Z_j^{I_j} = z_{0,j}^{i_{0,j}} \cdots z_{n_j,j}^{i_{n_j,j}}$, and coefficients $a_{I_1, \dots, I_m} \in \mathbb{C}$.

► **Definition 33** (Bézout number [42]). *Let $F = \{f_1, \dots, f_n\}$ be a system of $n = n_1 + \dots + n_m$ multi-homogeneous polynomials with degrees $\{d_{i,j} \mid i \in [n], j \in [m]\}$. The Bézout number $d_{\text{Béz}}$ of F is defined as the coefficient of $\prod_{j=1}^m \alpha_j^{n_j}$ in $\prod_{i=1}^n \sum_{j=1}^m d_{i,j} \alpha_j$, where $\alpha_1, \dots, \alpha_m$ are symbolic variables representing the m variable sets.*

► **Remark 34.** Computing $d_{\text{Béz}}$ in general is difficult [40]. Checking if $d_{\text{Béz}}$ is non-zero, however, is tractable, which suffices for our purposes.

For clarity, as in Definition 32 the system F is defined over variable subsets Z_j , each of size $n_j + 1$. For each polynomial f_i , $d_{i,j}$ is now the degree of f_i relative to variable set Z_j .

► **Example 35.** Let $F = (f_1, f_2, f_3)$ with

$$f_1 = x_1 y_1 y_2 + x_2 y_2 y_3 \quad d_{1,1} = 1 \quad d_{2,1} = 2 \quad (7a)$$

$$f_2 = x_1 y_1 + x_2 y_2 \quad d_{1,2} = 1 \quad d_{2,2} = 1 \quad (7b)$$

$$f_3 = y_1 y_2 + y_2 y_3 \quad d_{1,3} = 0 \quad d_{2,3} = 2, \quad (7c)$$

where $Z_1 = \{x_1, x_2\}$, $Z_2 = \{y_1, y_2, y_3\}$, $n_1 = 1$, $n_2 = 2$, $m = 2$. Then,

$$\prod_{i=1}^3 \sum_{j=1}^2 d_{i,j} \alpha_j = (\alpha_1 + 2\alpha_2)(\alpha_1 + \alpha_2)(2\alpha_2) = 2\alpha_1^2 \alpha_2 + 6\alpha_1 \alpha_2^2 + 4\alpha_2^3. \quad (8)$$

The coefficient of $\alpha_1 \alpha_2^2$, and thus the Bézout number, is $d_{\text{Béz}} = 6$.

► **Theorem 36** (Bézout's Theorem [42, 49]). *A multi-homogeneous system $F(Z) = 0$ has no more than $d_{\text{Béz}}$ geometrically isolated solutions in $\mathbb{P}^{n_1}(\mathbb{C}) \times \dots \times \mathbb{P}^{n_m}(\mathbb{C})$. If $F(Z) = 0$ does not have an infinite number of solutions in $\mathbb{P}^{n_1}(\mathbb{C}) \times \dots \times \mathbb{P}^{n_m}(\mathbb{C})$, then it has exactly $d_{\text{Béz}}$ solutions, counting multiplicities.*

Applied to Example 35, this tells us that either the number of solutions to $F = (f_1, f_2, f_3)$ is infinite, or there are exactly $d_{\text{Béz}} = 6$ solutions. Thus, if the Bézout number is positive, there is a solution.

5.2 The class MHS and completeness results

Since a positive Bézout number implies the existence of a solution, and finding an approximate solution is clearly in TFNP, we now define a new subclass of TFNP to capture this, MHS.

► **Definition 37** ((Low-Degree) Multi-homogeneous Systems (MHS)). *Define $\text{MHS}_{s,d}$ as the set of TFNP relations $R(x, y)$ poly-time reducible (as defined in [44]) to finding an ϵ -approximate solution to a system $F = \{f_1, \dots, f_n\}$ of n multi-homogeneous equations, where*

1. (a solution exists) $d_{\text{Béz}} > 0$,
2. (at most s variables per variable group Z_j) for all $j \in [m]$, $n_j \leq s$,
3. (each equation f_i is of total degree at most d) for all $i \in [n]$, $\sum_{j=1}^m d_{i,j} \leq d$, and
4. $\epsilon \in \Omega(2^{-\text{poly}(n)})$.

For clarity, ϵ and n are inputs and thus may depend on $|x|$, whereas s and d are parameters and considered constants independent of $|x|$. More formally, there exist $\text{poly}(|x|)$ -time computable functions g and h , such that $g(x)$ outputs ϵ and a description of a multi-homogeneous system F , and $R(x, h(x, Y))$ holds, where Y is an approximate solution to $F(Y) = 0$ with $\sum_{k=1}^n |f_k(Y)| \leq \epsilon$, assuming each equation f_i and variable group Z_j is normalized in the Euclidean norm¹³. Finally, define $\text{MHS} := \bigcup_{s,d \in \Theta(1)} \text{MHS}_{s,d}$.

In words, MHS requires constant bounds on the variable set sizes s and total degree d per equation (i.e. the number of variables in each monomial), and allows up to inverse exponential precision additive error ϵ .

► **Observation 38.** $\text{MHS} \subseteq \text{TFNP}$.

We now show that PRODSAT captures the complexity of MHS.

► **Theorem 39.** *Let M denote input size, and consider any $\epsilon \in \Omega(2^{-\text{poly}(M)})$.*

1. (Containment in MHS) *For any local dimension $d \in O(1)$ and locality $k \in O(1)$, ϵ -approximate PRODSAT with WSDR for k -local constraints on qudits of dimension d is in $\text{MHS}_{d-1,k}(\epsilon)$.*
2. (MHS-hardness) *$\text{MHS}_{s,d}(\epsilon)$ is poly-time reducible to $\Theta(\epsilon)$ -approximate PRODSAT on qubits (i.e. local dimension 2) with an SDR and locality $k \geq (s+1)^d$.*

Proof sketch. For MHS-hardness, consider a multi-homogeneous system $F = \{f_1, \dots, f_n\}$ with variable sets Z_1, \dots, Z_m , $\sum_{j=1}^m d_{i,j} \leq d$ for all equations $i \in [n]$, $n_j \leq s$ for all variable sets $j \in [m]$, and $d_{\text{Béz}} > 0$. First, we embed F into a qudit system. Each variable group Z_j has, by definition, $n_j + 1$ variables, and so each assignment to these variables can be represented by an $(n_j + 1)$ -dimensional state $|\psi_j\rangle$. However, F need not be multi-linear, meaning monomials in equation f_i each contain exactly $d_{i,j}$ variables (counting multiplicity) from Z_j . To simulate this non-linearity, we instead create $c_j := \max_{i \in [n]} d_{i,j}$ states in our system, $|\psi_{1,j}\rangle, \dots, |\psi_{c_j,j}\rangle$, each again of dimension $n_j + 1$. Let Q_j denote the set of qudits created by this mapping for Z_j , and consider any f_i acting on some set of variable sets $A_i \subseteq \{Z_1, \dots, Z_m\}$. Since f_i has degree $d_{i,j}$ in variable set Z_j , we will construct our corresponding clause $|\phi_i\rangle$ to act without loss of generality on the first $d_{i,j}$ qudits in Q_j . (Assume the qudits in Q_j have an arbitrary, fixed order.) Under this mapping, let $B_i \subseteq Q_1 \cup \dots \cup Q_m$ denote the corresponding

¹³This is to prevent trivial solutions such as setting all variables to approximately 0. Formally, we mean the coefficient vector of each f_i is normalized with respect to the Euclidean norm, and likewise for each variable group Z_j , the corresponding assignment vector. For example, to normalize $f = x_1 y_1 y_2 + x_2 y_2 y_3$, the right hand side is multiplied by $(\|f\|_2 \|x\|_2 \|y\|_2)^{-1}$, for $f = (1, 1)$, $x = (x_1, x_2)$, and $y = (y_1, y_2)$.

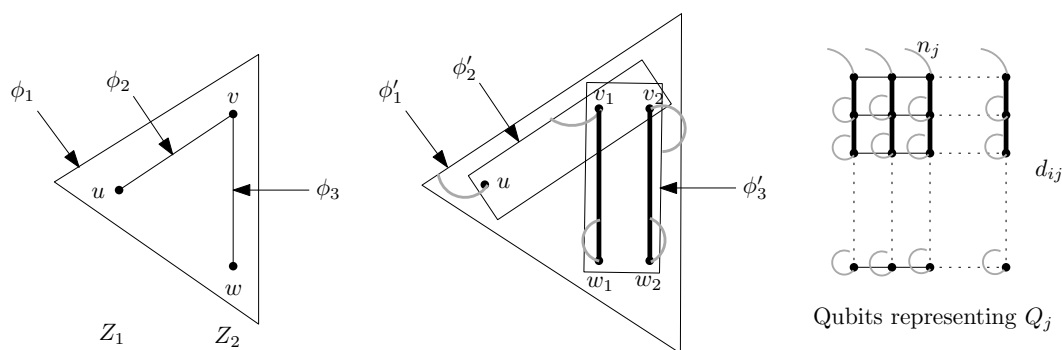


Figure 1 (Left) The reduction of Theorem 39 before the reduction to qubits and without equality constraints, as illustrated on Example 35. The latter has equations $f_1 = x_1y_1y_2 + x_2y_2y_3$, $f_2 = x_1y_1 + x_2y_2$, and $f_3 = y_1y_2 + y_2y_3$ with variable sets $Z_1 = \{x_1, x_2\}$ and $Z_2 = \{y_1, y_2, y_3\}$, $n_1 = 1$, $n_2 = 2$, $m = 2$, and $d = 3$. Variable sets Z_1 and Z_2 are represented by vertex sets $\{u\}$ and $\{v, w\}$, respectively. Vertices u, v, w correspond to $|\psi_{1,1}\rangle \in \mathbb{C}^2$ and $|\psi_{1,2}\rangle, |\psi_{2,2}\rangle \in \mathbb{C}^3$, respectively. The joint product state assignment thus has form $|\psi_{1,1}\rangle|\psi_{1,2}\rangle|\psi_{2,2}\rangle = \sum_{i=0}^1 \sum_{j,k=0}^2 \alpha_i \alpha_j \alpha_k |i\rangle|j\rangle|k\rangle \in \mathbb{C}^2 \otimes (\mathbb{C}^3)^{\otimes 2}$. Each constraint f_i is encoded into a rank-1 projector onto $|\phi_i\rangle$. Specifically, $|\phi_1\rangle = |001\rangle + |112\rangle$ (acting on all three systems), $|\phi_2\rangle = |00\rangle + |11\rangle$ (acting on the first two systems), and $|\phi_3\rangle = |01\rangle + |12\rangle$ (acting on the last two systems). (Middle) The figure on the left after the reduction to qubits is applied, followed by addition of equality constraints via 2-local projectors onto the antisymmetric subspace. Here, $v, w \in \mathbb{C}^3$ have been mapped to $v_1, v_2 \in \mathbb{C}^2$ and $w_1, w_2 \in \mathbb{C}^2$, respectively. Edge $\{u, v\}$ is now a hyperedge $\{u, v_1, v_2\}$. Thick black edges represent equality constraints. Thinner gray edges represent the SDR, i.e. which qubit is matched to which hyperedge. (Right) A “close-up” of all qubits representing Q_j when the full reduction is applied to a general multi-homogeneous system. Thick black edges represent equality constraints. Thinner gray edges represent the SDR. The first row, labelled $q_{i,1}$ through q_{i,n_j} in the proof, are matched with the n_j hyperedges incident on Q_j corresponding to the original equations f_i (hyperedges not depicted). Vertices in rows i with $i > 1$ are matched with their incident edge to row $i - 1$.

set of qudits to be acted on by $|\phi_i\rangle$. To now design $|\phi_i\rangle$, ideally for any $j \in [m]$, we would like all qudits in Q_j to have identical local assignments, i.e. $|\psi_{1,j}\rangle = \dots = |\psi_{d_i,j,j}\rangle$. In such a case, we can represent the multi-homogeneous polynomial f_i by a projective rank-1 constraint $|\phi_i\rangle$ acting on B_i , since the amplitudes (with respect to the computational basis) of $\bigotimes_{j=1}^m \bigotimes_{i=1}^{d_{i,j}} |\psi_{i,j}\rangle$ are in one-to-one correspondence with all possible monomials of f_i , as given by Equation (6). Figure 1 illustrates the construction thus far.

Enforcing equality. To indeed enforce equality among all qudits in Q_j , since we are considering product state assignments, it suffices to place 2-local projectors onto the antisymmetric subspace for each consecutive pair of qudits in Q_j . Unfortunately, this would add too many constraints when our qudits have local dimension $d > 2$, so that a WSDR cannot exist.

In fact, *no* projector of rank n_j can enforce equality between qudits of dimension $n_j + 1$ (Observation 62 in the full version [3]). To overcome this obstacle, we instead apply the reduction to qubits from Theorem 31, and then use the projectors onto the antisymmetric subspace to force the equality among the resulting qubits (Figure 1, middle). ◀

6 High-degree, sparse univariate polynomials and TFNP

Section 5 focused on low-degree multi-homogeneous systems and their relationship to TFNP. In this section, we study roots of a single high-degree univariate sparse polynomial. Section 6.1 first defines a new subclass of TFNP based on the Fundamental Theorem of Algebra, denoted

SFTA. Section 6.2 shows that $\text{SFTA} \subseteq \text{TFNP}$. Section 6.3 shows how to reduce computing a root of a sparse univariate polynomial to QSAT with SDR. We can currently prove this reduction works in the exact case. We conjecture it also works in the approximate case, which would imply $\text{SFTA} \subseteq \text{MHS}$.

6.1 Definitions, the Fundamental Theorem of Algebra, and SFTA

Sparse polynomials are well studied in the polynomial systems literature (e.g. [30]). For our purposes, we use the following definition.

► **Definition 40** (Sparse polynomial). *An s -sparse polynomial $p(x) \in \mathbb{C}[x]$ of degree d has only $s \in O(\text{polylog}(d))$ non-zero coefficients $a_i \in \mathbb{C}$. The specification of p is a list of $\lceil \log d \rceil$ -bit approximations¹⁴ \tilde{a}_i of each non-zero a_i , along with the corresponding indices $i \in \{0, \dots, d\}$.*

Thus, the degree is, by definition, exponentially larger than the input size. In this paper, we only consider *univariate* sparse polynomials. We can now define our second complexity class, SFTA. For this, recall that a *monic* polynomial has the coefficient of its highest degree non-zero term set to 1.

► **Theorem 41** (Sparse Fundamental Theorem of Algebra (SFTA)). *Define SFTA as the set of TFNP relations $R(a, b)$ poly-time reducible (as defined in [44]) to finding an ϵ -approximate root $r \in \mathbb{C}$ of a sparse monic univariate polynomial $p \in \mathbb{C}[x]$ of degree d , where $|r| \in [0, 1 + 2 \log(d)/d]$, and ϵ and d may be functions in the input size. That is, there exist poly-time computable functions g and h , such that $g(a)$ outputs a sparse polynomial p , and $R(a, h(a, r))$ holds, where r satisfying $|r| \in [0, 1 + 2 \log(d)/d]$ is an approximate root of p with $|p(r)| \leq \epsilon$.*

Note the two restrictions to (1) roots in $[0, 1 + 2 \log(d)/d]$ and (2) p being monic. We use both to obtain containment in TFNP in Section 6.2. For clarity, $2 \log(d)$ can be replaced with any asymptotically larger term scaling as $\text{polylog}(d)$, and containment in TFNP would still hold (Theorem 44).

6.2 SFTA is in TFNP

Ideally, we would like $\text{SFTA} \subseteq \text{TFNP}$. And here we run into our first obstacle. Given a sparse polynomial p , it is not difficult to see that via square-and-multiply, the number of *field operations* over \mathbb{C} to compute $p(x)$ is $\text{poly}(n)$, for n the size of input a in Theorem 41. However, TFNP is a class concerning *bit complexity*, not field operation complexity. Unfortunately, it is immediate that if, say, $x = 2$, then $p(x) = x^{2^n}$ for $x = 2$ requires 2^n bits to represent, which is exponential in the input size. Moreover, even if the $p(x)$ itself has an encoding of size $\text{poly}(n)$, the intermediate terms in its calculation (e.g. each monomial's value on x) may require exponentially large encodings. This phenomenon is sometimes referred to as *intermediate expression swell*, and occurs for example in Euclid's GCD algorithm [50].

To circumvent this in our setting, we require two tricks. First, in Theorem 41 we restrict attention to complex numbers x satisfying $|x| \in [0, 1 + \text{polylog}(d)/d]$. Since $(1 + \text{polylog}(d)/d)^d \in O(\text{polylog}(d))$, this avoids the exponential blowup seen in the example above. More formally, one can show that $p(x)$ can be evaluated on this interval to within

¹⁴One could also consider, e.g., exact representations via field extensions. For simplicity, we use approximate representations, which suffices as our goal is to find approximate roots.

additive error 2^{-L} in time polynomial in L and n . The following is essentially identical to Lemma 1 of [30], except for a constant factor overhead since we are dealing with complex numbers, whereas [30] considers real numbers. This overhead disappears into the Big-Oh notation.

► **Lemma 42** (Adaptation of Lemma 1 of [30]). *Let $p \in \mathbb{C}[x]$ be an s -sparse polynomial, $x \in \mathbb{C}$, and $L \geq 0$ an integer. Then, $p(x)$ can be computed to within additive error 2^{-L} with bit complexity $\tilde{O}((s + \log d)(L + d \log[\max(1, |x|)] + \log d + s))$, where \tilde{O} omits logarithmic factors.*

The second trick we need for containment in TFNP is to argue that we have not “broken” the Fundamental Theorem of Algebra in restricting to range $|x| \in [0, 1 + \text{polylog}(d)/d]$ – namely, we must show that there always *exists* a root in this range. This is where the monic property of our polynomial will play a role, coupled with an application of Landau’s inequality [34].

► **Lemma 43.** *Let $p = \sum_{i=0}^d a_i x^i$ be an s -sparse polynomial as per Definition 40, which is additionally monic. Then, there exists an $x \in \mathbb{C}$ with $|x| \leq 1 + \left(\frac{\ln(\sqrt{sd})}{d}\right)$, such that $p(x) = 0$.*

Combining Lemmas 42 and 43 immediately yields the desired claim.

► **Theorem 44.** $\text{SFTA} \subseteq \text{TFNP}$.

6.3 Embedding univariate polynomials into QSAT with SDR: NP-hardness and towards SFTA \subseteq MHS

Theorem 44 showed $\text{SFTA} \subseteq \text{TFNP}$. Does the stronger containment $\text{SFTA} \subseteq \text{MHS}$ also hold? The main contribution of this section is to give a poly-time many-one reduction from SFTA to exact MHS i.e. to MHS with $\epsilon = 0$:

► **Theorem 45.** *Let P be an s -sparse polynomial of degree d . There exists an efficiently computable set $\Pi = \{\Pi_i\}_{i \in [m]}$ of $m = O(s \log(d))$ 3-local and one 2-local rank-1 constraints on $N = O(s \log d)$ qubits with an SDR, such that $P(x/y) = 0$ iff $\Pi(|v_1\rangle \otimes \cdots \otimes |v_N\rangle) = 0$ with unit vector $|v_1\rangle = (x, y)^T \in \mathbb{C}^2$.*

From this, we immediately obtain the following.

► **Corollary 46.** *Given monic s -sparse polynomial $p(x) \in \mathbb{C}[x]$ of degree d , the problem of computing a root x such that $p(x) = 0$ is in $\text{MHS}_{s',d'}(\epsilon)$, with number of equations $n = O(s \log d)$, at most $s' = 2$ variables per group, total degree at most $d' = 3$ per equation, and precision $\epsilon = 0$.*

Recall, however, that in Definition 37 we defined MHS with an allowed error tolerance ϵ at least inverse exponential in the input size, whereas Theorem 45 requires $\epsilon = 0$. We believe the construction of Theorem 45 also yields an analogous result for the approximate case of inverse exponential ϵ , but have not yet been able to prove it. The main challenge appears to be controlling the error in the reduction, i.e. one would like to say that if one can find an ϵ -approximate solution to PRODSAT with SDR, then one can resolve the roots of P within some controlled precision $f(\epsilon)$. This is tricky, as the degree of P is exponential, which may amplify errors. We thus conjecture the following.

► **Conjecture 47.** $\text{SFTA} \subseteq \text{MHS}$.

References

- 1 Dorit Aharonov, Daniel Gottesman, Sandy Irani, and Julia Kempe. The Power of Quantum Systems on a Line. *Communications in Mathematical Physics*, 287(1):41–65, 2009. doi:10.1007/s00220-008-0710-3.
- 2 Marco Aldi, Niel de Beaudrap, Sevag Gharibian, and Seyran Saeedi. On efficiently solvable cases of quantum k-SAT. *Communications in Mathematical Physics*, 381(1):209–256, January 2021. doi:10.1007/s00220-020-03843-9.
- 3 Marco Aldi, Sevag Gharibian, and Dorian Rudolph. An unholy trinity: TFNP, polynomial systems, and the quantum satisfiability problem, 2025. arXiv:2412.19623.
- 4 Itai Arad, Miklos Santha, Aarthi Sundaram, and Shengyu Zhang. Linear time algorithm for quantum 2SAT. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:14, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2016.15.
- 5 Bengt Aspvall, Michael F. Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979. doi:10.1016/0020-0190(79)90002-4.
- 6 Nikhil Bansal, Sergey Bravyi, and Barbara M. Terhal. Classical approximation schemes for the ground-state energy of quantum and classical ising spin hamiltonians on planar graphs. *Quantum Information & Computation*, 9(7):701–720, 2009. doi:10.26421/QIC9.7-8-12.
- 7 Aleksandrs Belovs, Gábor Ivanyos, Youming Qiao, Miklos Santha, and Siyi Yang. On the Polynomial Parity Argument Complexity of the Combinatorial Nullstellensatz. In *32nd Computational Complexity Conference (CCC 2017)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.CCC.2017.30.
- 8 Fernando G. S. L. Brandão and Aram W. Harrow. Product-State Approximations to Quantum States. *Communications in Mathematical Physics*, 342(1):47–80, 2016. doi:10.1007/s00220-016-2575-1.
- 9 Sergey Bravyi. Efficient algorithm for a quantum analogue of 2-SAT, 2006. arXiv:quant-ph/0602108.
- 10 Sergey Bravyi, Cristopher Moore, and Alexander Russell. Bounds on the quantum satisfiability threshold. In Andrew Chi-Chih Yao, editor, *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 5-7, 2010. Proceedings*, pages 482–489. Tsinghua University Press, 2010. arXiv:0907.1297v2.
- 11 John Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing, STOC '88*, pages 460–467, New York, NY, USA, 1988. Association for Computing Machinery. doi:10.1145/62212.62257.
- 12 Jianxin Chen, Xie Chen, Runyao Duan, Zhengfeng Ji, and Bei Zeng. No-go theorem for one-way quantum computing on naturally occurring two-level systems. *Physical Review A*, 83(5):050301, 2011. doi:10.1103/PhysRevA.83.050301.
- 13 Xi Chen, Xiaotie Deng, and Shang-Hua Teng. Settling the Complexity of Computing Two-player Nash Equilibria. *J. ACM*, 56(3):14:1–14:57, 2009. doi:10.1145/1516512.1516516.
- 14 David A. Cox, John Little, and Donal O’Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Undergraduate Texts in Mathematics. Springer International Publishing, Cham, 2015. doi:10.1007/978-3-319-16721-3.
- 15 David A. Cox, John Little, and Donal O’Shea. *Using Algebraic Geometry*. Graduate Texts in Mathematics. Springer-Verlag, 2005. doi:10.1007/b138611.
- 16 Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a Nash equilibrium. In *Thirty-Eighth Annual ACM Symposium on Theory of Computing, STOC '06*, pages 71–78, New York, NY, USA, 2006. ACM. doi:10.1145/1132516.1132527.

- 17 Niel de Beaudrap and Sevag Gharibian. A linear time algorithm for quantum 2-SAT. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 27:1–27:21, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2016.27.
- 18 David Eisenbud and Joe Harris. *3264 and all that: A second course in algebraic geometry*. Cambridge University Press, Cambridge, 2016. doi:10.1017/CB09781139062046.
- 19 John Fearnley, Paul Goldberg, Alexandros Hollender, and Rahul Savani. The Complexity of Gradient Descent: $CLS = PPAD \cap PLS$. *Journal of the ACM*, 70(1):7:1–7:74, 2022. doi:10.1145/3568163.
- 20 L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956. doi:10.4153/CJM-1956-045-5.
- 21 William Fulton. *Intersection theory, Second Edition*, volume 2 of *Ergebnisse der Mathematik und ihrer Grenzgebiete*. Springer, second edition, 1998. doi:10.1007/978-1-4612-1700-8.
- 22 Sevag Gharibian, Yichen Huang, Zeph Landau, and Seung Woo Shin. Quantum Hamiltonian complexity. *Foundations and Trends® in Theoretical Computer Science*, 10(3):159–282, 2015. doi:10.1561/04000000066.
- 23 Sevag Gharibian and Julia Kempe. Approximation algorithms for QMA-complete problems. *SIAM Journal on Computing*, 41(4):1028–1050, 2012. doi:10.1137/110842272.
- 24 Andreas Goerdt. Matched instances of quantum satisfiability (QSat) – product state solutions of restrictions. In René van Bevern and Gregory Kucherov, editors, *Computer Science – Theory and Applications*, Lecture Notes in Computer Science, pages 156–167, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-19955-5_14.
- 25 Mika Göös, Alexandros Hollender, Siddhartha Jain, Gilbert Maystre, William Pires, Robert Robere, and Ran Tao. Further Collapses in TFNP. In *37th Computational Complexity Conference (CCC 2022)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.33.
- 26 Mika Göös, Pritish Kamath, Katerina Sotiraki, and Manolis Zampetakis. On the Complexity of Modulo-q Arguments and the Chevalley - Warning Theorem. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.CCC.2020.19.
- 27 David Gosset and Daniel Nagaj. Quantum 3-SAT Is QMA1-Complete. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, FOCS '13*, pages 756–765, USA, 2013. IEEE Computer Society. doi:10.1109/FOCS.2013.86.
- 28 Bruno Grenet. On the complexity of polynomial system solving, 2014. Talk at *XXVth Rencontres Arithmétiques de Caen*. https://membres-ljk.imag.fr/Bruno.Grenet/publis/talk_tatihou14.pdf.
- 29 Sean Hallgren, Daniel Nagaj, and Sandeep Narayanaswami. The local Hamiltonian problem on a line with eight states is QMA-complete. *Quantum Information & Computation*, 13(9-10):721–750, 2013. doi:10.26421/QIC13.9-10-1.
- 30 Gorav Jindal and Michael Sagraloff. Efficiently Computing Real Roots of Sparse Polynomials. In *Proceedings of the 2017 ACM International Symposium on Symbolic and Algebraic Computation, ISSAC '17*, pages 229–236, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3087604.3087652.
- 31 David S. Johnson, Christos H. Papadimitriou, and Mihalis Yannakakis. How easy is local search? *Journal of Computer and System Sciences*, 37(1):79–100, 1988. doi:10.1016/0022-0000(88)90046-3.
- 32 Richard Kenyon and Andrei Okounjov. What is... a dimer? *Notices of the AMS*, 52(3), 2005.
- 33 Simon-Luca Kremer, Dorian Rudolph, and Sevag Gharibian. Quantum k-SAT related hypergraph problems, 2025. doi:10.48550/arXiv.2506.17066.
- 34 E. Landau. Sur quelques théorèmes de M. Petrovitch relatifs aux zéros des fonctions analytiques. *Bulletin de la Société Mathématique de France*, 33:251–261, 1905. URL: <https://eudml.org/doc/86123>.

- 35 C. R. Laumann, A. M. Läuchli, R. Moessner, A. Scardicchio, and S. L. Sondhi. Product, generic, and random generic quantum satisfiability. *Physical Review A*, 81(6):062345, 2010. doi:10.1103/PhysRevA.81.062345.
- 36 C. R. Laumann, R. Moessner, A. Scardicchio, and S. L. Sondhi. Phase transitions and random quantum satisfiability. *Quantum Information & Computation*, 10:1–15, 2010.
- 37 Joon Lee, Nicolas Macris, Jean Bernoulli Ravelomanana, and Perrine Vantalon. The PRODSAT phase of random quantum satisfiability, 2024. doi:10.48550/arXiv.2404.18447.
- 38 Yuhao Li, William Pires, and Robert Robere. Intersection classes in TFNP and proof complexity. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ITCS.2024.74.
- 39 Henry Ma and Anand Natarajan. Two bases suffice for qma1-completeness, 2025. doi:10.48550/arXiv.2509.24390.
- 40 Gregorio Malajovich and Klaus Meer. Computing Minimal Multi-homogeneous Bézout Numbers Is Hard. In Volker Diekert and Bruno Durand, editors, *STACS 2005*, pages 244–255, Berlin, Heidelberg, 2005. Springer. doi:10.1007/978-3-540-31856-9_20.
- 41 Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991. doi:10.1016/0304-3975(91)90200-L.
- 42 Alexander Morgan and Andrew Sommese. A homotopy for solving general polynomial systems that respects m-homogeneous structures. *Applied Mathematics and Computation*, 24(2):101–113, 1987. doi:10.1016/0096-3003(87)90063-4.
- 43 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- 44 Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *J. Comput. Syst. Sci.*, 48(3):498–532, 1994. doi:10.1016/S0022-0000(05)80063-7.
- 45 K. R. Parthasarathy. On the maximal dimension of a completely entangled subspace for finite level quantum systems. *Proceedings Mathematical Sciences*, 114(4):365–374, November 2004. doi:10.1007/BF02829441.
- 46 Dorian Rudolph. Towards a universal gateset for QMA₁, 2025. arXiv:2411.02681.
- 47 Dorian Rudolph, Sevag Gharibian, and Daniel Nagaj. Quantum 2-SAT on low dimensional systems is QMA₁-complete: Direct embeddings and black-box simulation, 2025. doi:10.4230/LIPIcs.ITCS.2025.85.
- 48 Arnold Schönhage. Equation solving in terms of computational complexity. In *Proc. Int. Congress of Mathem.*, pages 131–153, Berkeley, USA, 1986. URL: <https://www.mathunion.org/fileadmin/ICM/Proceedings/ICM1986.1/ICM1986.1.ocr.pdf>.
- 49 Igor R. Shafarevich. *Basic Algebraic Geometry*. Springer Berlin Heidelberg, 1974. doi:10.1007/978-3-642-96200-4.
- 50 Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra (2. ed.)*. Cambridge University Press, 2003.