Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 9 (2025)

ECCC

An unholy trinity: TFNP, polynomial systems, and the quantum satisfiability problem

Marco Aldi* Sevag Gharibian† Dorian Rudolph†

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.

Contents

*Department of Mathematics and Applied Mathematics, Virginia Commonwealth University, USA. Email: maldi2@vcu.edu.

†Department of Computer Science and Institute for Photonic Quantum Systems (PhoQS), Paderborn University, Germany. Email: {sevag.gharibian, dorian.rudolph}@upb.de.

ISSN 1433-8092

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 [JPY88, MP91, Pap94] 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) [MP91] 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 $\Pi \in$ 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 [JPY88, Pap94]:

- 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 [Pap94] and a Nash equilibrium [DGP06, CDT09] 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 [FGHS22].

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": CLS = PPAD $\cap$ PLS [FGHS22], EOPL = PLS $\cap$ PPAD and SOPL = PLS $\cap$ PPADS [GHJ+22] (see also [LPR24]).

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 [Pap94] defined the problem CHEVALLEY by invoking the Chevalley-Warning theorem, which states: Given is a system of polynomials $\{f_i\}_{i=1}^r$ over $\mathbb{F}_p[X_1, \ldots, X_n]$ 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 [Pap94], it is not expected to be PPA-complete; however, two variants of CHEVALLEY have been shown PPA-complete [BIQ+17, GKSZ20].

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 ith polynomial. For our purposes, we actually require a more recent *multi*-homogenous extension due to Shafarevich [Sha74], which gives a similar statement for the more general setting of systems of *multi*-homogeneous polynomials (Definition 50), 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 $p \in \mathbb{C}[x_1, \ldots, x_n]$ 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 $S_2 = \{y_1, y_2, y_3\}$, $x_1y_1y_2 + x_2y_2y_3$ is multihomogeneous, whereas the homogeneous polynomial $x_1 + y_1$ is not. (Nevertheless, any homogeneous polynomial is trivially multihomogeneous relative to the partition with one set S containing all variables.)

The multi-homogeneous Bézout theorem (Theorem 56) 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 52). Then, it states that for any multi-homogeneous system of n equations $\{p_j\}_{j=1}^n \subseteq \mathbb{C}[x_1, \ldots, x_{n+t}]$, 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 53).

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 Representatives[1] (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 [FF56]), the search version of 3-SAT with SDR is poly-time solvable.

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-trivially[2] 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 QMA_1-complete [GN13], where QMA_1 is Quantum Merlin Arthur (QMA) with perfect completeness. Likewise, both 2-SAT [APT79] and 2-QSAT [ASSZ16, BG16] can be solved in linear time. Finally, for k-QSAT with SDR, Laumann, Läuchli, Moessner, Scardicchio, and Sondhi [LLM+10] (see also [LMSS10, LMRV24]) showed that,

[1] Given subsets $S_1, \ldots, S_m \subseteq [n]$, an SDR is a set of distinct elements $r_1, \ldots, r_m$ 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.

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

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 [AdBGS21] gave a *parameterized*[3] algorithm solving a special class of QSAT with SDR instances efficiently. In the opposite direction, Goerdt showed [Goe19] 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 64), 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 qu*d*it 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 is[4] 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 19) generalizes an SDR by introducing a *weight* function $w : V \to \mathbb{Z}_{\geq 0}$, such that for any vertex $v \in V$ corresponding to a qudit, v can be matched to $w(v)$ clauses. Which weight function should one choose? In this work, when we say a given QSAT instance $G = (V, E)$ on n qudits of local dimensions $d_1, \ldots, d_n$ has a WSDR, we mean with respect to weight function $w(v_i) = d_i - 1$ for each $i \in \{1, \ldots, n\}$. Thus, on n-qu*b*it systems, a WSDR is just an SDR. Note that checking whether G has an WSDR can be done efficiently (Remark 28).

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* $d_1, \ldots, d_n$*, 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.*

[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.

Theorem 1 is the qu$d$it generalization of [LLM+10], which showed the analogous result for qu$b$it systems with SDR. We thus have that for any $d \geq 2$, QSAT with WSDR on qudits is in TFNP. Above, "generic" (Definition 17) 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 [AGIK09, HNN13], 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 [LLM+10], 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 [LLM+10]. 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 [Par04], which says that any completely entangled subspace[5] has dimension at most $\prod_{i=1}^{k} d_i - \sum_{i=1}^{k} d_i + k - 1$ (Corollary 49).

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 57))**.** MHS is the set of total NP search problems poly-time reducible to finding an ϵ-approximate solution to a system $F = \{f_1, \ldots, f_n\} \subseteq \mathbb{C}[x_1, \ldots, x_{n+t}]$ 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 MHS $\subseteq$ 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 below, $d \in O(1)$ is required for our proof and yields constant locality k, whereas $s \in O(1)$ yields constant local dimensional qudits. (3) Formally, MHS is a union of complexity classes $\text{MHS}_{s,d}$ 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]:

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

[6] We use the term "MHS-complete" in the introduction for simplicity, but the formal statement is more subtle (Theorem 59). In the case of MHS-hardness, for example, it says any problem in $\text{MHS}_{s,d}(\epsilon)$ can be reduced to solving k-QSAT on qubits with locality $k \geq (s+1)^d$ within precision $\Theta(\epsilon)$, for $s, d, k \in O(1)$. Namely, 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.

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 78)
QSAT with SDR + one additional clause	NP-complete	[Goe19], this paper (Theorem 7)

Figure 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.

Theorem 3 (Informal; formal statement in Theorem 59)**.** *For any* $\epsilon \in \Omega(1/\exp)$ *and constant* $d \geq 2$*, computing an* ϵ*-approximate product-state solution to k-QSAT on qudits with WSDR is* $\mathrm{MHS}(\Theta(\epsilon))$*-complete.*

As even finding common roots of homogeneous polynomial systems in $n+1$ variables and n equations remains an open problem [Gre14], 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, *uni*variate polynomial equation. Below, a *sparse* polynomial (Definition 63), is one whose number of non-zero coefficients is logarithmic in its degree.

Definition 4 (Sparse Fundamental Theorem of Algebra (SFTA) (Informal; see Definition 65))**.** 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 $\epsilon \in \Omega(1/\operatorname{poly}(d))$.

As implied by its name, SFTA is inspired by the Fundamental Theorem of Algebra (Theorem 64), which recall states that any non-constant complex polynomial has a complex root r. Two comments regarding restrictions in the definition: First, the sparsity ensures[7] 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 $\operatorname{poly}(d)$ time (see e.g. [Sch85], as used in Section 4.4 of [AGS21]), 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 68), and is in fact necessary in order to prove SFTA $\subseteq$ TFNP (Theorem 69)[8]. [Sch85]

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

Theorem 5 (SFTA is in zero-error MHS (Informal; see Theorem 70))**.** *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*

[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 ith coefficient of p. For us, however, the sparsity is necessary for our proof technique behind Theorem 5.

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

$O(s\log(d))$, 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 MHS $\not\subseteq$ SFTA. This belief notwithstanding, by leveraging an old result of Canny [Can88], we show that generic (Definition 17) instances of QSAT with WSDR *can* be embedded into the roots of a single, high-degree polynomial p (Theorem 83). (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 jth root of p, it produces the ith amplitude of the jth solution to QSAT.) The polynomials p and r_i, however, are only poly-*space* computable, which is why this cannot yield MHS $\subseteq$ 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 $\mathbb{P} \neq$ 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. [Goe19]) 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 [Goe19] 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 (Theorem 78)! 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 Figure 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 [AGS21] in three different directions to solve new special cases efficiently.

Our first two results here concern the qubit case, and are complementary. In this setting, [AdBGS21] efficiently solves QSAT with SDR for generic (Definition 17) instances of *transfer type* $b = n - m + 1$ (Definition 85), where m denotes the number of constraints and n the number of qubits. Recall *non-generic* instances allow constraints which are not entangled across some bipartite cuts, and a transfer filtration (Definition 85) of transfer type b is a type of hyperedge ordering built on an initial subset of b qubits.

We first show that the generic assumption can be dropped if one assumes an "almost extending edge order" (Definition 87), which in turn implies the existence of an SDR [AdBGS21]:

Theorem 8 (Informal; see 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 $\mathrm{poly}(L, \log 1/\epsilon, k^r)$, 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 9 (Informal; see 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 [AdBGS21] to QSAT on qu$d$its 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* (Figure 5).

1.2 Techniques

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

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

$$0 = \langle\phi|\alpha_1, \beta_2\rangle = \sum_{i,j\in[2]} \phi^*_{i,j}\alpha_i\beta_j. \tag{1}$$

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 qu*d*its 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 qu*d*it is replaced with $d-1$ qu*b*its, 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 [Bra06, LLM+10]; see Lemma 73), 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 contraints 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 [GHLS15]. For example, mean-field ansatzes suffice to efficiently approximate ground state energies of planar [BBT09, BH16] and dense [GK12, BH16] 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 [BMR09, CCD+11], which has implications such as the fact that such Hamiltonians cannot be used to prepare resource states for one-way quantum computing [CCD+11].

2. *Why is adding SDRs to the picture interesting?* PRODSAT with *SDR* is interesting as it falls under the "dimer model" of physics [KO05], 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 [LLM+10] 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' [GKSZ20] non-circuit based PPA_p-complete problem ($p \geq 3$ a prime) for the Chevalley-Warning 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 strengthed 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 $d_{Béz} > 0$ boils down to checking if a certain bipartite graph has a perfect matching (Observation 55). More generally, computing $d_{Béz}$ 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 [Pap94] to Brouwer's fixed point theorem. This is because there is a natural algorithm 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 [Gre14], 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?

Organization. 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 give efficient algorithms for special cases of QSAT with WSDR.

2 Preliminaries

We assume a basic background in quantum computation, see e.g. [NC00]. 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. [Sha74, CLO15] for references.

Notation and basic definitions. We use $:=$ to indicate a definition. For $|\psi\rangle \in \mathbb{C}^d$, we define $\| \, |\psi\rangle \, \|_p := (\sum_{i=1}^d |\psi_i|^p)^{1/p}$. For a linear operator $M : \mathbb{C}^d \to \mathbb{C}^d$, we analogously define $\| M \|_p$ on the singular values of M. $\mathbb{C}[x_1, \ldots, x_n]$ 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 10 (Lipschitz continuity)**.** We say function $f : \mathbb{C} \to \mathbb{C}$ is K-Lipschitz continuous if for all $x, y \in X$, $|f(x) - f(y)| \leq K\,|x - y|$.

Fact 11. 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 $K = scr^{d-1}d$.

Proof. Let S be the set of non-zero coefficients of p. Then, for any $x, y \in X$,

$$|p(x) - p(y)| \leq \sum_{i \in S} |c_i| \left| x^i - y^i \right| = |x - y| \sum_{i \in S} |c_i| \left| \sum_{j=1}^{i} x^{i-j} y^{j-1} \right| \leq |x - y|\, scr^{d-1}d. \tag{2}$$

□

Thus, when $c, d \in O(1)$, $K \in O(1)$. Note that Definition 10 and Fact 11 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 $\mathcal{H} = \mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2} \otimes \cdots \otimes \mathbb{C}^{d_n}$ for some integers $d_1, \ldots, d_n \geq 2$. As is standard, we fix a computational basis $\{|0\rangle, \ldots, |d_i - 1\rangle\}$ for each qudit, so that an arbitrary vector in $\mathcal{H}$ can be written

$$|\psi\rangle = \sum_{j_1=0}^{d_1-1} \cdots \sum_{j_n=0}^{d_n-1} a_{j_1\cdots j_n} |j_1 \cdots j_n\rangle \tag{3}$$

for some choice of complex coefficients $a_{j_1\cdots j_n}$ satisfying $\sum_{j_1=0}^{d_1-1} \cdots \sum_{j_n=0}^{d_n-1} |a_{j_1\cdots j_n}|^2 = 1$. (Since solutions to QSAT are null space vectors, the normalization of $|\psi\rangle$ will often not be important.)

Definition 12 (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* $\Pi_1, \ldots, \Pi_m \in \mathcal{L}(\mathcal{H})$ of the form

$$\pi^{-1}(|\psi_i\rangle\langle\psi_i| \otimes I_{n-k})\pi, \tag{4}$$

  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: 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 $|\psi\rangle$, $\langle\psi| \sum_i \Pi_i |\psi\rangle \geq \alpha$.

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 13 (ϵ-approximate k-PRODSAT on qudits, decision version)**.** First, k-PRODSAT is defined as k-QSAT on qudits (Definition 12), 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 $i \in \{1, \ldots, n\}$. 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 14 (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 $\epsilon/\operatorname{poly}(m)$ to verify a YES case in NP (Remark 15).

Remark 15. (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,

$$\langle\psi|\Pi_i|\psi\rangle = \langle\varphi_1| \otimes \cdots \otimes \langle\varphi_k|\Pi_i|\varphi_1\rangle \otimes \cdots \otimes |\varphi_k\rangle, \tag{5}$$

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 $\epsilon/\operatorname{poly}(m)$, then for any i, Equation (5) can also be computed with additive error $\epsilon/\operatorname{poly}(m)$. 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

$$\sum_{j_1=0}^{d_1-1} \cdots \sum_{j_k=0}^{d_n-1} a_{j_1\cdots j_k} x_{i_1,j_1} \cdots x_{i_k,j_k} = 0, \tag{6}$$

where $i_1, \ldots, i_k$ 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 jth amplitude of the ith qudit.

Example 16. For instance, suppose $d_1 = 2$ and $d_2 = 3$ so that the first and second qudits are, respectively, a qubit $|\varphi_1\rangle = x_{1,0}|0\rangle + x_{1,1}|1\rangle$ and a qutrit $|\varphi_2\rangle = x_{2,0}|0\rangle + x_{2,1}|1\rangle + x_{2,2}|2\rangle$. A general two-local constraint $\Pi_1 = |\psi\rangle\langle\psi|$ for $|\psi\rangle = (a_{0,0}, a_{0,1}, a_{0,2}, a_{1,0}, a_{1,1}, a_{1,2})^T$ being satisfied by assignment $|\varphi_1\rangle \otimes |\varphi_2\rangle$ is equivalent to the multilinear equation

$$a_{0,0}x_{1,0}x_{2,0} + a_{0,1}x_{1,0}x_{2,1} + a_{0,2}x_{1,0}x_{2,2} + a_{1,0}x_{1,1}x_{2,0} + a_{1,1}x_{1,1}x_{2,1} + a_{1,2}x_{1,1}x_{2,2} = 0\,. \tag{7}$$

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 17), 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 variety[9]

$$\mathcal{X}_{d_1,\ldots,d_n} := \mathbb{P}^{d_1-1}(\mathbb{C}) \times \cdots \times \mathbb{P}^{d_n-1}(\mathbb{C})\,. \tag{8}$$

[9] Roughly, a variety is simply the set of solutions to a given set of polynomial equations.

In this geometric interpretation, each clause Π_i defines a hypersurface $V_i \subseteq \mathcal{X}_{d_1,\ldots,d_n}$ 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 $V_1 \cap V_2 \cap \cdots \cap V_m$.

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

Definition 17 (Genericity [CLO05, Def. 5.6])**.** A property is said to *hold generically* for a set of polynomials $f_1, \ldots, f_n$ 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 $f_1, \ldots, f_n$ for which $g(\cdots) \neq 0$.

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.

3.1 Definitions

Definition 18 (Weighted hypergraph)**.** A *weighted hypergraph* is a pair (G, w) consisting of a hypergraph G and a *weight function* $w : V(G) \to \mathbb{Z}_{\geq 0}$.

Thus, a hypergraph G without weights on its vertices may be viewed as a weighted hypergraph $(G, 1)$ with the weight function defined by $w(v) = 1$ for all $v \in V(G)$.

Definition 19 (Weighted System of Distinct Representatives (WSDR))**.** A *Weighted System of Distinct Representatives* for weighted hypergraph (G, w) is a mapping $f : E(G) \to V(G)$, 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 $v \in V(G)$.

Remark 20. A hypergraph G has a (non-weighted) system of distinct representatives (SDR) if and only if $(G, 1)$ has a WSDR. Hence, WSDRs generalize SDRs.

As an aside, a function f that to each edge $e \in E(G)$ assigns a vertex $f(e) \in e$ is more generally known as a hypergraph orientation [FKK03]. There exist works which study connections between hypergraph orientations and multi-homogeneous polynomial systems (e.g. [BEKT22]), but for clarity, as far as we are aware our definition of WSDR appears distinct from the hypergraph orientations used previously in the literature.

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

$$|S|_w := \sum_{v \in S} w(v). \tag{9}$$

Example 22. If w is the constant function 1, then $|S|_1 = |S|$ is the cardinality of S.

3.2 Existence and computation of WSDRs

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 for completeness.

Theorem 23. *(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 it at least one edge of X. Then (G, w) has a WSDR if and only $|V_X|_w \geq |X|$ for every $X \subseteq E(G)$.*

In the special case $w = 1$, Theorem 23 reduces to the usual Hall's Marriage Theorem.

Remark 24. An immediate consequence of Hall's Marriage Theorem is that $|V(G)|_w \geq |E(G)|$ is a necessary condition for (G, w) to have a WSDR.

Via Theorem 23, we thus obtain the following sufficient condition for when G has a WSDR.

Corollary 25. *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.*

Proof. For every $X \subseteq E(G)$, by double counting,

$$|X| \min_{e \in X} |e|_w \leq \sum_{e \in X} |e|_w \leq \sum_{v \in V_X} w(v) \deg(v) \leq |V_X|_w \max_{v \in V(G)} \deg(v) \leq |V_X|_w \min_{e \in X} |e|_w \,. \tag{10}$$

Hence $|X| \leq |V_X|_w$ and the results follows from Theorem 23. □

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

Definition 26 (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 27. *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 $d \leq k$.*

Proof. In one direction this follows immediately from Corollary 25. In the opposite direction, if (G, w) has a WSDR, then $|E(G)| \leq |V(G)|_w$ by Theorem 23. Hence $d|V(G)|_w = |E(G)|k \leq |V(G)|_w k$ from which the result easily follows. □

Remark 28 (Computation of WSDRs)**.** WSDRs can be efficiently computed. Namely, given a weighted hypergraph (G, w) satisfying the conditions of Theorem 23, computing a WSDR reduces to computing a maximum matching in the bipartite graph G' with $V(G') = V_1 \cup V_2$, where $V_1 = E(G)$, $V_2 = \{v_i \mid v \in V(G), i \in [w(v)]\}$, and $E(G') = \{\{e, v_i\} \mid e \in V_1, v_i \in V_2, v \in e\}$ (see [Gal86] for a survey). Alternatively, the WSDR may also be computed using a maximum flow algorithm (see [CML23] for a survey).

3.3 WSDRs under graph operations

Finally, we study WSDRs under the cartesian product of hypergraphs, defined next. This will be useful in Section 7.5.

Definition 29 (Hypergraph Cartesian Product)**.** The *cartesian product* of two weighted hypergraphs (G_1, w_1) and (G_2, w_2) is the weighted hypergraph $(G_1, w_1)\Box(G_2, w_2) = (G_1\Box G_2, w_1\Box w_2)$ where $G_1\Box G_2$ is the usual cartesian product of hypergraphs such that $V(G_1\Box G_2) = V(G_1)\times V(G_2)$ and

$$E(G_1\Box G_2) = \left(\bigcup_{v_1\in V(G_1)} \{v_1\} \times E(G_2)\right) \cup \left(\bigcup_{v_2\in V(G_2)} E(G_1) \times \{v_2\}\right) \tag{11}$$

while $(w_1\Box w_2)((v_1, v_2)) = w_1(v_1) + w_2(v_2)$ for all $v_1 \in V(G_1)$ and $v_2 \in V(G_2)$.

Remark 30 (WSDRs under cartesian products)**.** Cartesian products preserve WSDRs in the following sense. Let (G_1, w_1) and (G_2, w_2) be weighted hypergraphs admitting, respectively, WSDRs f_1 and f_2. Let $f_1\Box f_2 : E(G_1\Box G_2) \to V(G_1)\times V(G_2)$ be such that $(f_1\Box f_2)(\{v_1\}\times e_2) = (v_1, f_2(e_2))$ for all $e_2 \in E(G_2)$, $v_1 \in V(G_1)$ and $(f_1\Box f_2)(e_1\times\{v_2\}) = (f_1(e_1), v_2)$ for all $e_1 \in E(G_1)$, $v_2 \in V(G_2)$. Since

$$(f_1\Box f_2)^{-1}(v_1, v_2) = (\{v_1\} \times f_2^{-1}(v_2)) \cup (f_1^{-1}(v_1) \times \{v_2\}) \tag{12}$$

then $f_1\Box f_2$ is a WSDR for $(G_1, w_1)\Box(G_2, w_2)$.

Example 31. Let C_n be a cycle on $n \geq 3$ vertices. Then C_n has an SDR and $(C_n\Box C_m, 2)$ has a WSDR for every $n, m \geq 3$. However $C_n\Box C_m$ has no SDR since $|V(C_n\Box C_m)| = nm < 2nm = |E(C_n\Box C_m)|$.

4 Existence results via Weighted SDRs

We now show our first main result, Theorem 1, which recall shows that QSAT with WSDR always has a product state solution. We give two proofs of this fact: Via the Chow ring (Section 4.1) and via reduction to the qubit case (Section 4.2).

4.1 Approach 1: Via the Chow Ring

Our first proof goes via the Chow Ring from algebraic geometry, which is defined in Section 4.1.1. With the necessary definitions in hand, the proof itself is simple and given in Section 4.1.2.

Brief overview of techniques. To show that QSAT with WSDR always has a solution (Theorem 1), recall we give two proofs, one based on the Chow ring, and the other based on a reduction from qu*d*its to qu*b*its. We now give an overview for the former. (The latter was already sketched in Section 1.) At a high level, the Chow ring approach uses intersection theory [Ful98, EH16, Sha74]. 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 36): Given a set of rank-1 QSAT constraints with solution sets $\{V_1, \ldots, V_r\}$

(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.

4.1.1 Background on the Chow Ring

We refer to [EH16, Ful98] 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 $\mathcal{X}_{d_1,\ldots,d_n} := \mathbb{P}^{d_1-1}(\mathbb{C}) \times \cdots \times \mathbb{P}^{d_n-1}(\mathbb{C})$.

Definition 32. The *Chow ring* of $\mathcal{X}_{d_1,\ldots,d_n}$ is the commutative ring

$$CH(\mathcal{X}_{d_1,\ldots,d_n}) = \mathbb{Z}[H_1,\ldots,H_n]/(H_1^{d_1},\ldots,H_n^{d_n}). \tag{13}$$

Example 33. The Chow ring of $\mathbb{P}^2(\mathbb{C}) = \mathcal{X}_3$ is $CH(\mathcal{X}_3) = \mathbb{Z}[H]/(H^3)$. As a set, it consists of linear combinations $a1 + bH + cH^2$, with $a, b, c \in \mathbb{Z}$, and multiplication

$$(a1 + bH + cH^2)(a'1 + b'H + c'H^2) = aa'1 + (ba' + ab')H + (ca' + bb' + ac')H^2. \tag{14}$$

Example 34. The Chow ring of $\mathbb{P}^1(\mathbb{C}) \times \mathbb{P}^1(\mathbb{C}) = \mathcal{X}_{2,2}$ is $CH(\mathcal{X}_{2,2}) = \mathbb{Z}[H_1, H_2]/(H_1^2, H_2^2)$. As a set it consists of linear combinations $a + bH_1 + cH_2 + dH_1H_2$, for all $a, b, c, d \in \mathbb{Z}$ with multiplication

$$(a1 + bH_1 + cH_2 + dH_1H_2)(a'1 + b'H_1 + c'H_2 + d'H_1H_2) = a''1 + b''H_1 + c''H_2 + d''H_1H_2 \tag{15}$$

where $a'' = aa'$, $b'' = ab' + ba'$, $c'' = ac' + ca'$, and $d'' = ad' + bc' + cb' + da'$.

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 $Z(\mathcal{X}_{d_1,\ldots,d_n})$ be the free abelian group of *cycles*, generated by subvarieties of $\mathcal{X}_{d_1,\ldots,d_n}$. Linear combinations $n_1V_1 + \cdots + n_kV_k$ 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 35 (Subvariety representative, $[V]$)**.** There is a $\mathbb{Z}$-linear map $Z(\mathcal{X}_{d_1,\ldots,d_n}) \to CH(\mathcal{X}_{d_1,\ldots,d_n})$ that, to each subvariety V of $\mathcal{X}_{d_1,\ldots,d_n}$, assigns an element of the Chow ring denoted by $[V]$. If V is a hypersurface of multidegree $(\delta_1,\ldots,\delta_n)$ (i.e. cut out by a polynomial of degree δ_i in the homogeneous coordinates on $\mathbb{P}^{d_i-1}(\mathbb{C})$), then $[V] = \delta_1H_1 + \cdots + \delta_nH_n$.

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

Fact 36 (Sufficient criterion for non-empty intersection, and Bézout number)**.** If $V_1,\ldots,V_r$ are hypersurfaces in $\mathcal{X}_{d_1,\ldots,d_n}$ such that $[V_1]\cdots[V_r]$ is non-zero, then $V_1 \cap \ldots \cap V_r$ is non-empty. If $[V_1]\cdots[V_r] = 0$ then $W_1 \cap \ldots \cap W_r = \emptyset$ for almost all hypersurfaces $W_1,\ldots,W_r$ such that $[W_1] = [V_1]$,..., $[W_r] = [V_r]$ (i.e. each W_i has the same multidegree as the corresponding V_i). If

$$[V_1]\cdots[V_r] = NH_1^{d_1-1}H_2^{d_2-1}\cdots H_n^{d_n-1} \tag{16}$$

for some positive integer N, then the generic intersection $W_1 \cap \ldots \cap W_r$ consists of N points and N is referred to as the *Bézout number*.

We remark that later in Definition 52, we will give a more precise definition of the Bézout number (needed for stating Bézout's Theorem). The definition above suffices for our discussion in this section.

Example 37. Let C, C' be curves in the complex projective plane $\mathcal{X}_3$ of respective degree δ, δ'. Then $[C] = \delta H$ and $[C'] = \delta' H$, which implies $[C][C'] = \delta\delta' H^2$. Hence the two curves will intersect in at least $\delta\delta'$ points. For generic choices of C, C' as above, the two curves will intersect in exactly $\delta\delta'$ points (Bézout's Theorem).

Example 38. Let C, C' be curves in $\mathcal{X}_{2,2}$ of respective bidegree (δ_1, δ_2) and (δ_1', δ_2'). Then $[C][C'] = (\delta_1\delta_2' + \delta_2\delta_1')H_1H_2$. This could be zero e.g. if $\delta_1 = \delta_1' = 0$, corresponding to the case in which C and C' are both of the form $\bigcup_i (\mathbb{P}^1(\mathbb{C}) \times \{p_i\})$ (which do not intersect for generic choices of p_i). On the other hand, consider the case $\delta_1 = 2$ and $\delta_2' = 1$. Then

$$C = (\{p_1\} \times \mathbb{P}^1(\mathbb{C})) \cup (\{p_2\} \times \mathbb{P}^1(\mathbb{C})) \tag{17}$$

and $C' = \bigcup \mathbb{P}^1(\mathbb{C}) \times \{p'\}$ for some $p_1, p_2, p' \in \mathbb{P}^1(\mathbb{C})$. Generically, $p_2 \neq p_1$ and $|C \cap C'| = |\{(p_1, p'), (p_2, p')\}| = 2 = \delta_1\delta_2'$. However, in the nongeneric case $p_1 = p_2$, we have $|C \cap C'| = \infty$.

4.1.2 Proof of Theorem 1 via the Chow Ring

With Fact 36 in hand, we are ready to give our first proof of Theorem 1. For this, let $\Pi = \{\Pi_i\}$ be an instance of QSAT on qudits $|\varphi_1\rangle, \ldots, |\varphi_n\rangle$ of dimensions $d_1, \ldots, d_n$, respectively. Recall that to such an instance Π, we assign a weighted hypergraph (G, w) as follows. We let $V(G) = \{v_1, \ldots, v_n\}$ and define $E(G) = \{e_1, \ldots, e_m\}$ 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 $i \in \{1, \ldots, n\}$.

Theorem 1. *Let $\Pi = \{\Pi_i\}$ be an instance of QSAT on n qudits of local dimensions $d_1, \ldots, d_n$, 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, $i = 1, \ldots, m$. 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 (6)), its image in the Chow ring is

$$[V_i] = \sum_{v_j \in E_i} H_j. \tag{18}$$

Hence,

$$\prod_i [V_i] = \sum_{v_{j_1} \in E_1, \ldots, v_{j_m} \in E_m} H_{j_1} \cdots H_{j_m}, \tag{19}$$

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 36. □

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

Corollary 39 (Counting number of SDRs and product solutions)**.** *Let N denote the Bézout number. By the proof above of Theorem 1, if Equation (16) holds (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 40. *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 natural number N. This is easiest to see with an explicit example, given next.*

Example 41. Consider $QSAT$ on 4 qutrits with underlying weighted graph (G,w) with vertices $V(G) = \{1,2,3,4\}$, and edges $E(G) = \{e_1,\ldots,e_8\}$ where $e_1 = \{1,2,3\}$, $e_2 = \{2,3,4\}$, $e_3 = \{3,4,1\}$, $e_4 = \{4,1,2\}$, $e_5 = e_6 = e_7 = e_8 = \{1,2,3,4\}$. In this case, $m = \sum_{i=1}^n d_i - 1$, and Equation (16) holds, since

$$(H_1+H_2+H_3)(H_2+H_3+H_4)(H_3+H_4+H_1)(H_4+H_1+H_2)(H_1+H_2+H_3+H_4)^4 \quad (20)$$
$$= 864H_1^2H_2^2H_3^2H_4^2\,. \quad (21)$$

To see this without any calculation, pick from each bracketed term a single term H_i. Any non-zero summand in Equation (19) must have picked any H_i at most $d_i - 1 = 2$ times. But since $m = \sum_{i=1}^n d_i - 1$, each H_i must be picked at least $d_i - 1$ times to ensure all edges are covered. Thus, Equation (16) holds. We conclude that *every* instance of QSAT with interaction graph (G,w) has at least 864 product solutions (counted with multiplicity) and almost all such instances have exactly 864 product solutions. Moreover, (G,w) has exactly 864 WSDRs.

Example 42. If every qudit of dimension d_i occurs in at most $d_i - 1$ constraints, then there exists a product solution. The WSDR exists trivially because it is impossible to assign a qudit to more than $d_i - 1$ constraints. To compute a product solution, iterate through the qudits in arbitrary order, keeping track of reduced constraints. We can assign each qudit i to a value in the common nullspace of the $\leq d_i - 1$ (reduced) 1-local constraints on qudit i.

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 [LMSS10]). The result follows from the main theorem of this section, Theorem 43, 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 43. *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.*

Proof. Let z denote the first qudit in Π of dimension $d+1$. To replace z by a qubit x and a qudit y, we will define and use a mapping $f : \mathbb{P}^1 \times \mathbb{P}^{d-1} \to \mathbb{P}^d$,

$$f(x,y) \coloneqq \begin{pmatrix} x_1 y_1 \\ x_2 y_d \\ x_1 y_2 - x_2 y_1 \\ x_1 y_3 - x_2 y_2 \\ \vdots \\ x_1 y_d - x_2 y_{d-1} \end{pmatrix}. \tag{22}$$

Via Lemma 45, we will then be able to argue that f allows us to create Π' which is satisfiable by a product state if and only if Π is.

To begin, let Π_i be a constraint of Π with associated hyperedge $e_i = \{z, v_2, \ldots, v_k\}$. We can view Π_i as a multilinear polynomial p whose monomials are the entries of $|z\rangle \otimes |v_2\rangle \cdots |v_k\rangle$ (taking $z, v_2, \ldots, v_k$ as symbolic vectors). The corresponding constraint in Π'_i with hyperedge $e'_i = \{x, y, v_2, \ldots, v_k\}$ is obtained by replacing every occurrence of z_j in p with $f(x,y)_j$. Π'_i is a valid constraint since its monomials are the entries of $|x\rangle \otimes |y\rangle \otimes |v_2\rangle \cdots |v_k\rangle$ (see Example 44). For constraints Π_i not acting on z, let $\Pi'_i = \Pi_i$.

What remains to show is the correspondence between product solutions to Π and Π' as well as the existence of a WSDR. The latter is straightforward, taking the d edges assigned to z in Π and assigning one of them to x and the remaining $d-1$ edges to y. To construct a product solution for Π from Π', just set $z = f(x,y)$, which is non-zero by Lemma 45. For the other direction, assign a preimage of z to (x,y), which again is efficiently computable by Lemma 45. $\square$

Example 44. To illustrate Theorem 43, let $\langle\phi| \in \mathbb{C}^6$ be a constraint on a qutrit z and a qubit v. A product state $|z\rangle \otimes |v\rangle$ satisfies this constraint if $p(z,v) = \sum_{i=1}^3 \sum_{j=1}^2 \phi_{ij} z_i v_j = 0$. The construction of Theorem 43 replaces the qutrit z with two qubits x, y. The new constraint $\langle\phi'|$ is defined via the polynomial

$$p'(x,y,v) = \sum_{i=1}^{3} \sum_{j=1}^{2} \phi_{ij} f(x,y)_i v_j = \sum_{j=1}^{2} (\phi_{1j} x_1 y_1 + \phi_{2j} x_2 y_2 + \phi_{3j}(x_1 y_2 - x_2 y_1)) v_j, \tag{23}$$

giving $\langle\phi'| = (\phi_{11}, \phi_{12}, \phi_{31}, \phi_{32}, -\phi_{31}, -\phi_{32}, \phi_{21}, \phi_{22})$ (where monomomials $x_i y_j z_k$ are listed in increasing binary order with respect to $ijk \in \{0,1\}^3$).

Lemma 45. *The map f given in (22) is well-defined (i.e. $f(x,y) \neq 0$ if $x \neq 0, y \neq 0$), and surjective with polynomial-time computable preimage.*

Proof. To show f is well-defined, let $x \in \mathbb{P}^1, y \in \mathbb{P}^{d-1}$, i.e., $x \neq 0, y \neq 0$. Consider cases:

(i) ($x_1 = 0$) Then $x_2 \neq 0$. There exists i with $y_i \neq 0$. If $i = d$, then $x_2 y_d \neq 0$. Otherwise $x_1 y_{i+1} - x_2 y_i \neq 0$.

(ii) ($x_1 \neq 0$) Let i be minimal such that $y_i \neq 0$. If $i = 1$, then $x_1 y_1 \neq 0$. Otherwise, $x_1 y_i - x_2 y_{i-1} = x_1 y_i \neq 0$.

Hence, $f(x,y) \neq 0$ and therefore well-defined.

To next show f is surjective, consider any $z \in \mathbb{P}^d$. We compute $x \in \mathbb{P}^1, y \in \mathbb{P}^{d-1}$ such that $f(x,y) = z$ via cases:

(i) ($z_1 = 0$) Set $x_1 = 0$ and $x_2 = 1$, satisfying the equation $x_1 y_1 = z_1$. The remaining equations are $y_d = z_2, y_1 = -z_3, y_2 = -z_4, \ldots, y_{d-1} = -z_{d+1}$. Since $z_1 = 0$, there exists an i with $y_i \neq 0$.

(ii) ($z_1 \neq 0$) Without loss of generality, assume $z_1 = 1$. Set $x_1 = 1, y_1 = 1$ to satisfy the first equation and ensure $x \neq 0, y \neq 0$. Substituting $y_1 = 1, x_1 = 1$, the remaining equations are:

$$x_2 y_d = z_2 \tag{24a}$$
$$y_2 = z_3 + x_2 \tag{24b}$$
$$y_3 = z_4 + x_2 y_2 \tag{24c}$$
$$\vdots$$
$$y_d = z_{d+1} + x_2 y_{d-1} \tag{24d}$$

Combining Equations (24b) to (24d), we have $y_d = x_2^{d-1} + \sum_{i=3}^{d+1} z_i x_2^{d+1-i}$. Substituting y_d in Equation (24a), we have $x_2^d + \sum_{i=3}^{d+1} z_i x_2^{d+2-i} = z_2$, which is a polynomial with solution in x_2. Finally, set $y_2, \ldots, y_d$ according to Equations (24b) to (24d), step by step. □

Remark 46. By Corollary 39, each application of Theorem 43 increases the number of product solutions by a factor of d (counted with multiplicity). This matches the intuition from Lemma 45, where computing the preimage of f requires solving a polynomial of degree d.

Remark 47 (Relation to the Segre embedding)**.** The map $f : \mathbb{P}^1 \times \mathbb{P}^{d-1} \to \mathbb{P}^d$ is a linear map from the Segre embedding of $\mathbb{P}^1 \times \mathbb{P}^{d-1}$ to $\mathbb{P}^d$, i.e. $f(x, y) = L\sigma(x, y)$ for some linear map L.

4.3 Application: Maximal dimension of a completely entangled subspace

Finally, we demonstrate the applicability of the WSDR framework beyond the setting of QSAT. Specifically, Parthasarathy [Par04] studies the notion of a completely entangled subspace and gives its maximal dimension. We can recover this result as a corollary of Theorem 1.

Definition 48 ([Par04])**.** Let $\mathcal{H}_1, \ldots, \mathcal{H}_k$ be complex Hilbert spaces of dimension d_i and $\mathcal{H} = \bigotimes_{i=1}^k \mathcal{H}_i$. A subspace $S \subseteq \mathcal{H}$ is said to be *completely entangled* if $|\psi_1\rangle \otimes \cdots \otimes |\psi_k\rangle \notin S$ for any non-zero product vector with $|\psi_i\rangle \in \mathcal{H}_i$.

Corollary 49 (*c.f.* [Par04])**.** *The maximal dimension of a completely entangled subspace is* $\prod_{i=1}^k d_i - \sum_{i=1}^k d_i + k - 1$.

Proof. Let $D = \dim(\mathcal{H}) = \prod_{i=1}^k d_i$. Let $S \subset \mathcal{H}$ be a subspace of dimension d_S and let $\Pi_{S^\perp} = \sum_{i=1}^{D-d_S} |\psi_i\rangle\langle\psi_i|$ be a spectral decomposition of the projector onto the orthogonal complement of S. If $D - d_S \leq \sum_{i=1}^k d_i - k$, then $\Pi_{S^\perp}$ has a WSDR, treating space $\mathcal{H}_i$ as a qudit of dimension d_i. Hence, if $d_S \geq \prod_{i=1}^k d_i - \sum_{i=1}^k d_i + k$, S must contain a product state by Theorem 1. Equivalently, if S is completely entangled, $d_S \leq \prod_{i=1}^k d_i - \sum_{i=1}^k d_i + k - 1$. This bound is tight because generic instances without WSDR have no product solution. □

5 Low-degree, multi-homogeneous systems and TFNP

We next study low-degree, multi-homogeneous polynomial systems. Section 5.1 first defines multi-homogeneous polynomial systems, and states the multihomogeneous Bézout Theorem. Section 5.2 then defines our first new TFNP subclass, MHS, and shows MHS-completeness of QSAT with SDR. The latter uses the WSDR techniques of Section 4.2.

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 50 (Multi-homogeneous polynomial [MS87])**.** A polynomial f is multi-homogeneous if there are m sets of variables $Z_j = \{z_{0,j}, \ldots, z_{n_j,j}\}$ and $d_1, \ldots, d_m \in \mathbb{Z}_{\geq 0}$ with at least one $d_j > 0$ such that

$$f = \sum_{\substack{I_1,\ldots,I_m: \\ \forall j \;\; |I_j|=d_j}} a_{I_1,\ldots,I_m} Z_1^{I_1} \cdots Z_m^{I_m}, \tag{25}$$

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

Let us repeat this in words, and subsequently give it context relative to QSAT. Above, each variable set Z_j has $n_j + 1$ variables. Each $Z_j^{I_j}$ term is a product of some subset of d_j variables from Z_j, with the precise choice of variables given by index subset I_j. Thus, d_j can be thought of as the *degree* of the polynomial relative to variables Z_j.

Example 51. A simple example of a multi-homogeneous polynomial is $x_1y_1y_2 + x_2y_2y_3$, where $Z_1 = \{x_1, x_2\}$, $Z_2 = \{y_1, y_2, y_3\}$, $d_1 = 1$, and $d_2 = 2$.

Let us return to product-state solutions for QSAT (i.e. PRODSAT). Why is *multi*-homogeneous the right formulation? When each monomial of f in Definition 50 contains at most one variable from each Z_j (i.e. $d_j \in \{0, 1\}$ for all $j \in [m]$), the equivalence between Equation (25) and a QSAT constraint is straightforward. Each set of variables Z_j corresponds to the $n_j + 1$ amplitudes of the jth qu*d*it of our system with $d = n_j + 1$. Thus, the number m of subsets Z_j is the number of qudits in our system. Any projective constraint $|\psi\rangle$ acting on subset of qudits $S \subseteq [m]$ is now equivalent to a polynomial f with $d_j = 1$ for $j \in S$ and $d_j = 0$ for $j \in [m] \setminus S$. As for the more general case where f has $d_j > 1$ for some Z_j, when higher degree terms in the variable sets are permitted, the reduction from a multi-homogeneous system back to QSAT is non-trivial, and given shortly in Theorem 59.

Bézout's theorem. We now state the mathematical principle on which our TFNP subclass rests, Bézout's theorem. For this, we first define the Bézout number. Below, the terms n_i are from Definition 50.

Definition 52 (Bézout number [MS87])**.** Let $F = \{f_1, \ldots, f_n\}$ be a system of $n = n_1 + \cdots + n_m$ multi-homogeneous polynomials with degrees $\{d_{i,j} \mid i \in [n], j \in [m]\}$. The *Bézout number* $d_{Béz}$ 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 $\alpha_1, \ldots, \alpha_m$ are symbolic variables representing the m variable sets.

Remark 53. Computing $d_{Béz}$ in general is difficult [MM05]. Checking if $d_{Béz}$ is non-zero, however, is tractable, which suffices for our purposes.

For clarity, as in Definition 50 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 54. Let $F = (f_1, f_2, f_3)$ with

$$f_1 = x_1y_1y_2 + x_2y_2y_3 \qquad d_{1,1} = 1 \qquad d_{2,1} = 2 \tag{26a}$$

$$f_2 = x_1y_1 + x_2y_2 \qquad d_{1,2} = 1 \qquad d_{2,2} = 1 \tag{26b}$$

$$f_3 = y_1y_2 + y_2y_3 \qquad d_{1,3} = 0 \qquad d_{2,3} = 2, \tag{26c}$$

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

$$\prod_{i=1}^{3}\sum_{j=1}^{2} d_{i,j}\alpha_j = (\alpha_1 + 2\alpha_2)(\alpha_1 + \alpha_2)(2\alpha_2) = 2\alpha_1^2\alpha_2 + 6\alpha_1\alpha_2^2 + 4\alpha_2^3. \tag{27}$$

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

Observation 55 (Number of weighted SDRs equals Bézout number)**.** *The number of weighted SDRs in a PRODSAT instance is equal to the Bézout number of the corresponding multi-homogeneous system. (For clarity, by definition of the Bézout number (Definition 52), we mean for the case of $n = n_1 + \cdots + n_m$.) To see this, observe that in $\prod_{i=1}^{n}\sum_{j=1}^{m} d_{i,j}\alpha_j$, the product is over all n equations, and for each equation f_i, the Bézout number corresponds to choosing from the inner sum (which represents variable groups) a single variable from a single variable group Z_j, such that this variable appears in f_i (i.e. $d_{i,j} > 0$). The coefficient of $\prod_{j=1}^{m}\alpha_j^{n_j}$ then counts the number of ways we can "cover" all f_i in this manner using using variables from each group Z_j precisely n_j times. The claim follows by observing that in the corresponding QSAT instance, any single such covering is equivalent to a single weighted SDR.*

With the Bézout number $d_{Béz}$ in hand, we state Bézout's theorem, which gives a sufficient condition for a multi-homegenous system having a solution.

Theorem 56 (Bézout's Theorem [MS87, Sha74])**.** *A multi-homogeneous system $F(Z) = 0$ has no more than $d_{Béz}$ geometrically isolated solutions in $\mathbb{P}^{n_1}(\mathbb{C}) \times \cdots \times \mathbb{P}^{n_m}(\mathbb{C})$. If $F(Z) = 0$ does not have an infinite number of solutions in $\mathbb{P}^{n_1}(\mathbb{C}) \times \cdots \times \mathbb{P}^{n_m}(\mathbb{C})$, then it has exactly $d_{Béz}$ solutions, counting multiplicities.*

Applied to Example 54, this tells us that either the number of solutions to $F = (f_1, f_2, f_3)$ is infinite, or there are exactly $d_{Béz} = 6$ 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 57 ((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 [Pap94]) to finding an ϵ-approximate solution to a system $F = \{f_1, \ldots, f_n\}$ of n multi-homogeneous equations, where

1. (a solution exists) $d_{Béz} > 0$,
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. $\epsilon \in \Omega(2^{-\operatorname{poly}(n)})$.

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 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 norm[10]. Finally, define

$$\text{MHS} := \bigcup_{s,d \in \Theta(1)} \text{MHS}_{s,d}. \tag{28}$$

In words, Equation (28) says 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.

As remarked in Section 1, the following observation follows straightforwardly since poly-time Turing machines can efficiently perform basic arithmetic with polynomial bits of precision, and since the degrees and set sizes in MHS are constant.

Observation 58. MHS $\subseteq$ TFNP.

We now show that PRODSAT captures the complexity of MHS.

Theorem 59. *Let M denote input size, and consider any $\epsilon \in \Omega(2^{-\operatorname{poly}(M)})$.*

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 $k \geq (s+1)^d$.*

Remark 60. Recall from Section 1 that our result does not specify a single k for which MHS-hardness is obtained *for all* $s, d \in O(1)$. Regarding this, a blowup in k is perhaps expected, since in the $k = 2$ case (i.e. 2-QSAT on qubits), producing a satisfying assignment is well-known to be efficiently solvable, even without an SDR (assuming a satisfying assignment exists) [Bra06]. It is, however, plausible that Theorem 59 can be extended in the $k = 2$ case on qudits for some local dimension $d > 2$, since 2-QSAT on qudits remains QMA_1-complete [ER08, Nag08, RGN24]. As for the bound $k \geq (s+1)^d$ in Theorem 59, in the simplest non-trivial case of quadratic equations on variable sets Z_i of size 2 each (i.e. $s = 1$), this bound yields $k = 4$.

Proof of Theorem 59. For containment in MHS, as argued above, any PRODSAT system with SDR can be represented as a system of multi-homogeneous equations. For simplicity, we consider the case of qubits; the qudit case is analogous. Without loss of generality, we may assume there are m qubits and $n = m$ clauses, since if $n < m$ an SDR cannot exist, and if $n > m$ we can add trivially satisfied constraints to the system. An equation f_i corresponding to a k-local constraint is multilinear in k variable groups, so we get $\sum_{j=1}^{m} d_{i,j} = k$ for all equations $f_i \in [n]$. Since the PRODSAT system

[10] This 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 $f = x_1y_1y_2 + x_2y_2y_3$, the right hand side is multiplied by $(\| f \|_2 \| x \|_2 \| y \|_2)^{-1}$, for $f = (1, 1)$, $x = (x_1, x_2)$, and $y = (y_1, y_2)$.

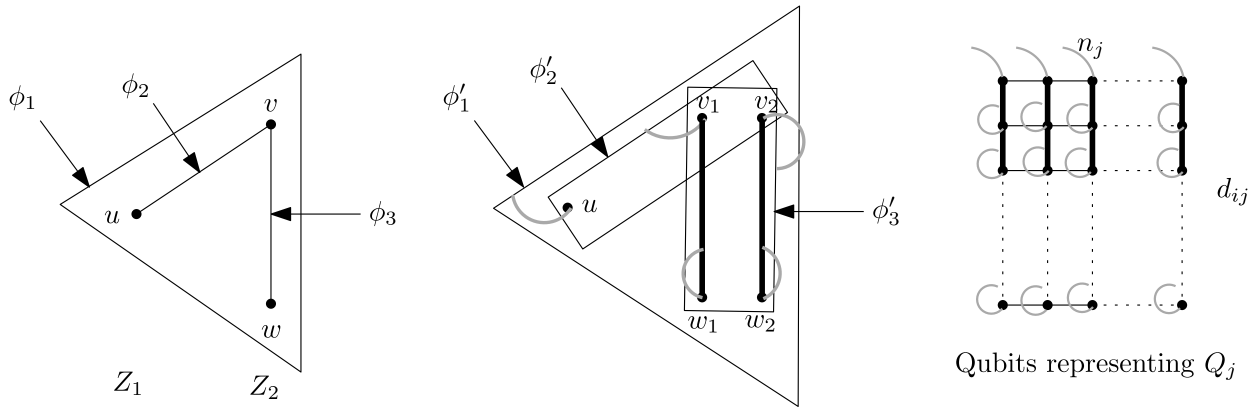

Figure 2: (Left) The reduction of Theorem 59 before the reduction to qubits and without equality constraints, as illustrated on Example 54. The latter has equations $f_1 = x_1y_1y_2 + x_2y_2y_3$, $f_2 = x_1y_1 + x_2y_2$, and $f_3 = y_1y_2 + y_2y_3$ with variable sets $Z_1 = \{x_1, x_2\}$ and $Z_2 = \{y_1, y_2, y_3\}$, $n_1 = 1$, $n_2 = 2$, $m = 2$, and $d = 3$. Variable sets Z_1 and Z_2 are represented by vertex sets $\{u\}$ and $\{v, w\}$, respectively. (For simplicity, the reduction actually creates $d = 3$ vertices for each Z_i, in order to be able to accommodate monomials of degree 3 in each Z_i. However, the system f_1, f_2, f_3 is at most linear in Z_1 and quadratic in Z_2, so 3 vertices per Z_i is overkill; we thus depict only the vertices needed to encode f_1, f_2, f_3.) 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 block 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 multihomogeneous 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 $i - 1$.

only contains qubits, we have $n_j = 1$ for all $j \in [m]$ and thus $s = 1$. By Observation 55, the Bézout number equals the number of SDRs, which is at least one. Finally, by construction and the definition of MHS(ϵ), cumulatively satisfying all f_i within total additive ϵ precision immediately yields a PRODSAT solution with ϵ precision.

For MHS-hardness, consider a multi-homogeneous system $F = \{f_1, \ldots, f_n\}$ with variable sets $Z_1, \ldots, Z_m$, $\sum_{j=1}^{m} d_{i,j} \leq d$ for all equations $i \in [n]$, $n_j \leq s$ for all variable sets $j \in [m]$, and $d_{Béz} > 0$. 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, $|\psi_{1,j}\rangle, \ldots, |\psi_{c_j,j}\rangle$, 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 $A_i \subseteq \{Z_1, \ldots, Z_m\}$. 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 $B_i \subseteq Q_1 \cup \cdots \cup Q_m$ denote the corresponding 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. $|\psi_{1,j}\rangle = \cdots = |\psi_{d_{i,j},j}\rangle$. 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 (25). Figure 2 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 qu*d*its have local dimension $d > 2$, so that a WSDR cannot exist. To see this, assume the worst case scenario in which $c_j = d_{i,j}$ for all $i \in [n]$, i.e. each variable group Z_j has the same degree in all equations. Now, by Observation 55, each variable set Z_j must "cover" n_j equations f_i, and so in principle each Q_j must also cover these same n_j equations. Recalling we have $n = \sum_{j=1}^{m} n_j$ equations, observe that a WSDR on our qudits can cover at most

$$\sum_{j=1}^{m} c_j n_j \tag{29}$$

clauses in our construction. (Each Q_j has c_j qudits, each of dimension $n_j + 1$, meaning each qudit in Q_j affords a WSDR n_j degrees of freedom.) Since Q_j must cover n_j of the equations f_i, in order for a WSDR to exist, it is necessary for our construction to implement *all* equality constraints for Z_j using at most $n_j(c_j - 1)$ rank-1 projectors. At least $c_j - 1$ 2-local constraints are necessary to ensure equality among c_j qudits, implying each equality constraint must have rank at most n_j. Unfortunately for $d > 2$, the antisymmetric subspace on two qudits of dimension $n_j + 1$ has dimension $(n_j + 1)^2 - \binom{n_j+2}{2} > n_j$ for $n_j > 1$ [Wat18]. In fact, *no* projector of rank n_j can enforce equality between qudits of dimension $n_j + 1$ (Observation 62).

To overcome this obstacle, we instead apply the reduction to qubits from Theorem 43, and then use the projectors onto the antisymmetric subspace to force the equality among the resulting qubits (Figure 2, middle). Specifically, consider any Q_j consisting of $d_{i,j}$ qudits of dimension n_j+1. Label these qudits $q_1, \ldots, q_{d_{i,j}}$. Theorem 43 replaces each q_i with n_j qubits which we label here as $q_{i,1}, \ldots, q_{i,n_j}$, such that any hyperedge acting on q_i now acts instead on $q_{i,1}, \ldots, q_{i,n_j}$. To simulate equality between the qudits q_i, by the construction of Theorem 43, it now suffices to place projectors onto the singlet state $|01\rangle - |10\rangle$ between $q_{i,k}$ and $q_{i+1,k}$ for all $i \in \{1, \ldots, c_j - 1\}$ and $k \in \{n_j\}$ (thick vertical edges in Figure 2, middle). This yields $\sum_{j=1}^{m} (d_{i,j} - 1) n_j$ equality constraints for Q_j.

The SDR. It remains to show that the resulting QSAT instance on qubits has an SDR. The argument is similar to the discussion surrounding Equation (29), i.e. we have $\sum_{j=1}^{m} d_{i,j} n_j$ degrees of freedom which which to cover all clauses, where each degree of freedom corresponds to a unique qubit in our system. Note Theorem 43 does not alter the number of hyperedges; thus, our system has precisely $n = \sum_{i=1}^{m} n_i$ clauses corresponding to $\{f_i\}_{i=1}^{n}$ to cover. Now, since $d_{Béz} > 0$ for $\{f_i\}_{i=1}^{n}$, and since we assumed each clause $|\phi_i\rangle$ acts without loss of generality on the first $d_{i,j}$ qudits in Q_j, by Observation 55 we may use the set of n_j qubits in Z_j which replaced the first qudit, q_1, in our use of Theorem 43 to cover all clauses acting on Q_j. The remaining qubits $q_{i,k}$ for $i > 1$ can now be straightforwardly used to cover all $\sum_{j=1}^{m} (d_{i,j} - 1) n_j$ equality constraints (Figure 2, right).

Precision. That an ϵ-approximate solution for the PRODSAT instance suffices to produce an $\Theta(\epsilon)$-approximate solution for MHS follows by the Lipschitz continuity of polynomials on a compact set and the fact that degrees and group sizes are bounded by $O(1)$. $\square$

Remark 61. MHS-hardness in Theorem 59 is stated in terms of qubits; however, the statement holds for any constant local dimension d. (The case of $d = 2$ simply yields the strongest hardness result.) Specifically, hardness can be shown by embedding each qubit output by our reduction into a qu$d$it and adding projector onto $\mathrm{Span}(|2\rangle, \ldots, |d-1\rangle)$ onto each qudit.

Finally, in the proof of Theorem 59, we claimed no low rank projector could test for equality — this follows by the definition of the antisymmetric subspace, but we include an explicit proof below for completeness.

Observation 62. *For $d > 2$, there exists no projector $\Pi \in \mathbb{C}^{d^2 \times d^2}$ of rank $\leq d-1$ such that for all $|\psi\rangle, |\phi\rangle \in \mathbb{C}^d$, $\Pi|\psi\rangle|\phi\rangle = 0$ iff $|\psi\rangle \propto |\phi\rangle$.*

Proof. Assume there exists such a projector Π. If $\mathrm{rank}(\Pi) \leq d-2$, we can easily find orthogonal $|\psi\rangle, |\phi\rangle$ such that $\Pi|\psi\rangle|\phi\rangle = 0$. Thus we must have $\mathrm{rank}(\Pi) = d-1$ and Π has the spectral decomposition $\Pi = \sum_{i=1}^{d-1} |v_i\rangle\langle v_i|$.

The constraint $\langle v_i|\psi, \phi\rangle = 0$ is then equivalent to $(L_i|\psi\rangle)|\phi\rangle = 0$ for $L_i|\psi\rangle := \langle v_i|(|\psi\rangle \otimes I)$. Let $V = \mathrm{Span}\{(L_i|\psi\rangle)^\dagger \mid i \in [d-1]\}$. By construction, $V^\perp$ is the set of all vectors $|\phi\rangle$ such that $\langle v_i|\psi, \phi\rangle = 0$ for all $i = 1, \ldots, d-1$, i.e., $\Pi|\psi\rangle|\phi\rangle = 0$. By assumption, this only holds for $|\phi\rangle \propto |\psi\rangle$. Thus, $V^\perp = \mathrm{Span}\{|\psi\rangle\}$ and $\dim(V) = d - \dim(V^\perp) = d-1$. Therefore, the $\{L_i|\psi\rangle \mid i \in [d-1]\}$ are linearly independent for any $|\psi\rangle$.

The multi-homogeneous system $\sum_{i=1}^{d-1} x_i L_i|\psi\rangle = 0$ (d equations) with variable sets $x \in \mathbb{P}^{d-2}$ and $|\psi\rangle \in \mathbb{P}^{d-1}$ then has a solution by Theorem 56. Therefore, $\sum_{i=1}^{d-1} x_i(L_i|\psi\rangle) = 0$ with $|\psi\rangle \neq 0$ and $x \neq 0$, which contradicts the linear independence of the $L_i|\psi\rangle$. $\square$

5.3 A brief aside: Solving a special case of PRODSAT on qudits with multi-homogeneous systems

We have seen that any $O(1)$-approximate PRODSAT instance reduces to an MHS instance (Theorem 59), which raises the question: Can one leverage techniques from solving multi-homogeneous systems to solve PRODSAT instances? Here, we briefly mention one such application, though it is not intended to be a focus of this work. Namely, Safey El Din and Schost [SS18] give an exact algorithm for computing all non-singular solutions (i.e. where the Jacobian matrix of the polynomial system has full rank) of *dehomogenized* rational multi-homogeneous systems with a finite number of solutions. We will not state their result, but note that in applying [SS18] to PRODSAT the computational complexity is polynomial in the number of WSDRs after removing one edge, which can generally be exponentially greater than just the number of WSDRs. On some hypergraphs, however, this number is bounded, and thus [SS18] provides a poly-time algorithm for PRODSAT. For example, a star of $n+1$ qu$d$its, such that there are d edges to $d-1$ qudits and $d-1$ edges to the others, only has a polynomial number of WSDRs for a fixed d, even after removing one edge (Figure 3).

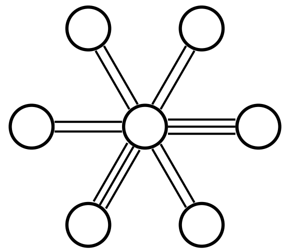

Figure 3: A PRODSAT instance with a star-like topology. The circles represent qutrits. All edges have size 2 and there are 9 WSDRs (assign one from each set of triple edges to the center).

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 SFTA $\subseteq$ 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 SFTA $\subseteq$ MHS. Finally, Section 6.4 studies the converse question — could MHS $\subseteq$ SFTA?

6.1 Definitions, the Fundamental Theorem of Algebra, and SFTA

Sparse polynomials are well studied in the polynomial systems literature (e.g. [JS17]). For our purposes, we use the following definition.

Definition 63 (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 approximations[11] $\widetilde{a}_i$ 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.

Next, we recall the Fundamental Theorem of Algebra:

Theorem 64 (Fundamental Theorem of Algebra)**.** *Every non-constant univariate polynomial $p \in \mathbb{C}[x]$ has at least one complex root.*

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.

Definition 65 (Sparse Fundamental Theorem of Algebra (SFTA))**.** Define SFTA as the set of TFNP relations $R(a, b)$ poly-time reducible (as defined in [Pap94]) 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 $|p(r)| \leq \epsilon$.

[11] One 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.

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 69).

6.2 SFTA is in TFNP

Ideally, we would like SFTA $\subseteq$ 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 Definition 65. 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 [zGJ03].

To circumvent this in our setting, we require two tricks. First, in Definition 65 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 additive error 2^{-L} in time polynomial in L and n. The following is essentially identical to Lemma 1 of [JS17], except for a constant factor overhead since we are dealing with complex numbers, whereas [JS17] considers real numbers. This overhead disappears into the Big-Oh notation.

Lemma 66 (Adaptation of Lemma 1 of [JS17])**.** *Let $p \in \mathbb{C}[x]$ be an s-sparse polynomial, $x \in \mathbb{C}$, and $L \geq 0$ an integer. Then, $f(x)$ can be computed to within additive error 2^{-L} with bit complexity*

$$\tilde{O}((s + \log d)(L + d\log[\max(1, |x|)] + \log d + s)), \tag{30}$$

where $\tilde{O}$ omits logarithmic factors.

The following corollary is immediate.

Corollary 67. *For s-sparse polynomial p with encoding size n, $p(x)$ can be computed within additive error 2^{-L} for any $x \in [0, 1 + \text{polylog}(d)/d]$ with bit complexity*

$$\tilde{O}((s + \log d)(L + s + \log d)) \in \text{poly}(n). \tag{31}$$

The proof of Lemma 66 follows identically to [JS17]: By choosing

$$K \in \Omega(L + \log s + d\log d \cdot \log[\max(1, |x|)]), \tag{32}$$

one can approximately evaluate $p(x)$ (using square-and-multiply to compute powers) by truncating intermediate expressions to their K most significant bits, while keeping the accrued additive error under control. The only difference here is that we need to independently track the error accumulated on both real and imaginary components of each complex number, which adds a constant factor overhead in the bit complexity. The details are omitted.

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 [Lan05].

Lemma 68. *Let $p = \sum_{i=0}^{d} a_i x^i$ be an s-sparse polynomial as per Definition 63, which is additionally monic. Then, there exists an $x \in \mathbb{C}$ with*[12]

$$|x| \leq 1 + \left(\frac{\ln(\sqrt{s}d)}{d} \right). \tag{33}$$

such that $p(x) = 0$.

Proof. Assume without loss of generality d is a power of 2, by which $\{|a_i|\} \leq d$ for all $i \in [0, \ldots, d]$. The *Mahler measure* of p is defined $M(p) = |a_d| \Pi_{j=1}^{d} \max(1, |z_j|)$, where $\{z_j\}_{j=1}^{d}$ is the set of roots of p, and in our setting the leading coefficient $a_d = 1$ by assumption. An upper bound on $M(p)$ can be derived as follows. Landau's inequality [Lan05] says

$$M(p) \leq \sqrt{\sum_{j=0}^{d} |a_j|^2}. \tag{34}$$

Combining this with the fact that, by Definition 63, each coefficient a_i of p satisfies $\{|a_i|\} \leq d$, we have

$$M(p) \leq \sqrt{s}d. \tag{35}$$

We now obtain a contradiction by lower bounding $M(p)$. Assume, for sake of contradiction, that all roots $|z_j|$ of p satisfy $|z_j| > (1 + c/d)^c$ for natural number $c \gg 1$ to be chosen shortly. Then,

$$M(p) = \Pi_{j=1}^{d} \max(1, |z_j|) > \left(1 + \frac{c}{d}\right)^{cd} \geq \left(1 + \frac{c}{d}\right)^{d + \frac{c}{2}} \geq e^c, \tag{36}$$

where the third statement holds for $c \in \Theta(\text{polylog}(d))$, and the last inequality follows since for all positive reals n and t, $(1 + t/n)^{n + t/2} \geq e^t$. Setting $c = \ln(\sqrt{s}d)$ completes the proof of the upper bound.

□

Combining Lemma 66 and Lemma 68 immediately yields the desired claim.

Theorem 69. SFTA $\subseteq$ TFNP.

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

Theorem 69 showed SFTA $\subseteq$ TFNP. Does the stronger containment SFTA $\subseteq$ 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 70. *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 $|v_1\rangle = (x, y)^T \in \mathbb{C}^2$.*

[12]We thank an anonymous reviewer for catching a minor bug in a previous statement of this lemma, which claimed $|x| > 0$ (which is incorrect) in addition to the upper bound currently stated (which is correct). Indeed, the polynomial x^d is sparse, but has only 0 as roots, thus violating $|x| > 0$. As this lower bound is not necessary for our purposes in this work, we have omitted it.

From this, we immediately obtain the following.

Corollary 71. *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 $\mathrm{MHS}_{s',d'}(\epsilon)$, 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 57 we defined MHS with an allowed error tolerance ϵ at least inverse exponential in the input size, whereas Theorem 70 requires $\epsilon = 0$. We believe the construction of Theorem 70 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 72. SFTA $\subseteq$ MHS*.*

In the meantime, Theorem 70 will allow us to obtain *NP-hardness* results for variants of QSAT with SDR, as given in Section 6.3.3.

Organization. Section 6.3.1 first develops tools for embedding univariate polynomials into QSAT instances. Section 6.3.2 shows the analogue of Theorem 70 for *non-sparse* polynomials, i.e. for polynomial degree d (Theorem 76). This will be useful for our NP-hardness results in Section 6.3.3. Section 6.3.2 then gives the proof of Theorem 70, which proceeds similarly to Theorem 76.

6.3.1 Building blocks

We now give the basic building blocks, using 3-local and 2-local constraints, to design PRODSAT instances whose solutions correspond to the roots of a univariate polynomial. For this, we use the concept[13] of *transfer functions* on qubits from [AdBGS21], 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 $|\varphi_1\rangle \cdots |\varphi_{k-1}\rangle$ to its first $k-1$ qubits.

Lemma 73. *(Transfer function, g) Let $|\phi\rangle$ be a k-local constraint on qubits. There exists a polynomial $g : (\mathbb{C}^2)^{k-1} \to \mathbb{C}^2$ such that, for any partial assignment $v_1, \ldots, v_{k-1}$, the clause $|\phi\rangle$ is satisfied (i.e. $\langle \phi | v_1, \ldots, v_k \rangle = 0$) iff[14] $|v_k\rangle \propto g(v_1, \ldots, v_{k-1})$ or $g(v_1, \ldots, v_{k-1}) = 0$. Moreover, g is linear in the coefficients of each v_i.*

Proof. If $g(v_1, \ldots, v_{k-1}) = 0$, we are trivially done, since the partial assignment already satisfies $|\phi\rangle$. For the remaining case, let $v' := v_1 \otimes \cdots \otimes v_{k-1}$ and $x := (v' \otimes I)^\dagger \phi$.[15] Note that g has the desired property if $g(v_1, \ldots, v_{k-1}) = y$ is orthogonal to x, i.e. if $x^\dagger y = 0$. To compute y, first compute $\overline{x} = (v' \otimes I)^T \overline{\phi}$. Then $y := ZX\overline{x}$. For $x = (x_1, x_2)^T$, we have $y^\dagger = (x_2, -x_1)$ and thus $y^\dagger x = 0$. Since we are on qubits, y is the *unique* choice of satisfying assignment for v_k, given $v_1, \ldots, v_{k-1}$. $\overline{x}$ is clearly linear in the coordinates of each $v_1, \ldots v_{k-1}$. We also define $f(v_1, \ldots, v_{k-1}) := \overline{x}$. ☐

[13] Transfer functions are a formal generalization of the transfer matrix formalism, which has appeared in previous works, e.g. [Bra06, LMSS10]

[14] $\propto$ means up to scaling up to non-zero constant.

[15] We do not use Dirac notation here as we make use of complex conjugates ($\overline{a}$) and transpositions (a^T) on their own.

Simulating linear operations via 2-local constraints. Consider first the transfer function for a 2-local constraint $H = \phi\phi^\dagger$. By Lemma 73, $g(v_1) = ZX\overline{x}$ with

$$\overline{x} = (v_1 \otimes I)^T \phi = \left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \otimes I \right)^T \left(\begin{bmatrix} a_1 \\ a_2 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) = \begin{bmatrix} a_1x_1 + a_2y_1 \\ b_1x_1 + b_2y_1 \end{bmatrix}. \tag{37}$$

In words, the assignment on the second qubit must be *orthogonal* to the the right hand side, $[a_1x_1 + a_2y_1, b_1x_1 + b_2y_1]^T$, in order to satisfy constraint $|\phi\rangle$. Note for the second equality that $[a_1, a_2]^T$ and $[b_1, b_2]^T$ are not necessarily orthogonal. In words, we can choose H such that g encodes an arbitrarily chosen linear combination of x_1 and y_1 in both coordinates.

Example 74. Suppose one wishes to enforce equality (up to rescaling) on product states on qubits 1 and 2, and suppose qubit 1's state is $(x_1, y_1)^T$. Setting $a_1 = 0$, $a_2 = -1$, $b_1 = 1$, and $b_2 = 0$, the right hand side of Equation (37) equals $(-y_1, x_1)^T$. The unique assignment to qubit 2 orthogonal to this is (x, y), thus enforcing qubit 2 to equal qubit 1.

Simulating quadratic operations via 3-local constraints. Similarly, we can choose 3-local H such that

$$\begin{aligned} \overline{x} = (v_1 \otimes v_2 \otimes I)^T \phi &= \left(\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \otimes \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} \otimes I \right)^T \phi \\ &= \left(\begin{bmatrix} x_1x_2 \\ x_1y_2 \\ y_1x_1 \\ y_1y_2 \end{bmatrix} \right)^T \left(\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right) \\ &= \begin{bmatrix} a_1x_1x_2 + a_2x_1y_2 + a_3y_1x_1 + a_4y_1y_2 \\ b_1x_1x_2 + b_2x_1y_2 + b_3y_1x_1 + b_4y_1y_2 \end{bmatrix} = \sum_{i,j \in [2]} \begin{bmatrix} a_{ij}v_{1,i}v_{2,j} \\ b_{ij}v_{1,i}v_{2,j} \end{bmatrix} \end{aligned} \tag{38}$$

and can therefore encode arbitrary linear combinations of the products $x_1x_2, x_1y_2, x_2y_1, x_2y_2$.

Example 75. Suppose given assignment $(x, y)^T$ to qubits 1 and 2, we wish to enforce qubit 3's assignment to encode the state (proportional to) $(x^2, y^2)^T$. Setting $a_1 = a_2 = a_3 = 0$, $a_4 = -1$, $b_1 = 1$, and $b_2 = b_3 = b_4 = 0$, the right hand side of Equation (37) equals $(-y^2, x^2)^T$. The unique assignment to qubit 3 orthogonal to this is (x^2, y^2), as desired.

6.3.2 Embedding sparse polynomials into PRODSAT

With our building blocks in hand, we first show how to embed non-sparse polynomials into QSAT instances, i.e. where the degree d is polynomial in the input size. Once we have this, a similar proof will yield Theorem 70.

Theorem 76. *Let p be a polynomial of degree d with $p(0) \neq 0$. There exists an efficiently computable set $\Pi = \{\Pi_i\}_{i \in [m]}$ of $m = O(d)$ 3-local and one 2-local rank-1 constraints on $N = O(d)$ 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 $|v_1\rangle = (x, y)^T \in \mathbb{C}^2$.*

Proof. Write $p(x) = \sum_{i=0}^{d} c_i x^i$ with $c_d \neq 0$ and $c_0 \neq 0$. First, we homogenize p by adding a variable y such that $q(x, y) := \sum_{i=0}^{d} c_i x^i y^{d-i}$. We now construct three sets of qubits and corresponding constraints.

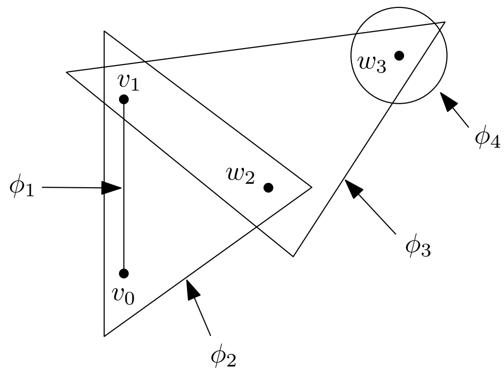

Figure 4: Construction of Theorem 76 illustrated on input $p(x) = x^3 - 4x + 5$, i.e. $d = 3$. Then, $q(x,y) = x^3 - 4xy^2 + 5y^3$. Constraint ϕ_1 is the equality constraint enforcing $|v_1\rangle \propto |v_0\rangle$. So, $|v_0\rangle = |v_1\rangle = [x,y]^T$. Next, we wish to enforce $|w_2\rangle = [x^2 - 4y^2, y^2]$. This is achieved via constraint ϕ_2. Next, ϕ_3 enforces $|w_3\rangle = [q(x,y), y^3]$. Finally, ϕ_4 enforces Equation (40). Observe this QSAT instance has an SDR: $(v_0, \phi_1), (v_1, \phi_2), (w_2, \phi_3), (w_3, \phi_4)$.

First set. The first set sets up the basic powers of x and y we need to simulate q. Specifically, the first qubit $v_0 = (x,y)^T$ represents variables x, y in q. With a 2-local constraint, we create $|v_1\rangle \propto |v_0\rangle$ (see Example 74). Then, we can use 3-local constraints and square-and-multiply to construct terms $|v_i\rangle := (x^i, y^i)^T$ for any required $2 \leq i \leq d-2$ (see Example 75). Observe that each time we add such a rank-1 constraint, we also add a new qubit to store the "answer" to the arithmetic operation the constraint is simulating.

Second set. We next embed q by recursively constructing a qubit with state $|w_d\rangle = (q(x,y), y^d)^T$. The base case is $\deg(q) \leq 2$, i.e., $q(x,y) = c_2x^2 + c_1xy + c_0y^2$. Then, $|w_2\rangle = (c_2x^2 + c_1xy + c_0y^2, y^2)$ is constructed with a 3-local constraint on v_0 and v_1 with $a_1 = a_2 = a_3 = 0$, $a_4 = -1$, $b_1 = c_2$, $b_2 = c_1$,$b_3 = 0$, $b_4 = c_0$. For $\deg(q) > 2$, we embed q recursively, assuming that we can embed polynomials of degree $< d$.

For each step $t \geq 1$ of the recursion, let $j_t > 0$ be minimal such that $c_{j_t} \neq 0$. We construct polynomial $q_t(x,y)$ with degree d_t defined as

$$q_t(x,y) := \sum_{i=0}^{d_t} c_{t,i} x^i y^{d_t - i} = x^{j_t} \cdot \underbrace{\sum_{i=j}^{d_t} c_{t,i} x^{i-j} y^{d_t - i}}_{r_t(x,y)} + c_{t,0} y^j \cdot y^{d_t - j}. \tag{39}$$

Note that $t = 1$ encodes our starting polynomial, i.e. $q_1(x,y) := q(x,y)$ of degree $d_1 = d$. In timestep t, we recursively construct $|w_{d_t - j_t}\rangle := (r_t(x,y), y^{d_t - j_t})^T$. (Note that $|w_{d_t - j_t}\rangle = 0$ iff $x = y = 0$.) Given $|w_{d_t - j_t}\rangle$, we then construct $|w_{d_t}\rangle$ by adding a 3-local constraint on v_{j_t}, $w_{d_t - j_t}$, and new qubit w_{d_t} with $a_1 = a_2 = a_3 = 0$, $a_4 = -1$, $b_1 = 1$, $b_2 = b_3 = 0$, $b_4 = c_{t,0}$ (as per Equation (38)).

Third set. Thus far, our constraints force the ground space of our QSAT instance to encode $q(x,y)$. We need a final check to enforce this to correspond to a root for the original polynomial p. For this, we add a 1-local constraint $|0\rangle$ onto w_d, enforcing the equality

$$\begin{bmatrix} q(x,y) \\ y^d \end{bmatrix} = \alpha \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \tag{40}$$

where α is some non-zero constant of proportionality, which is *a priori* unknown. The full construction is illustrated in Figure 4.

Correctness. First, if there exists x such that $p(x) = 0$, $|v_1\rangle = (x, 1)^T$ satisfies Equation (40). (All other constraints are immediately satisfied since they enforce the logic of the building blocks in Section 6.3.1.) Conversely, consider some satisfying assignment to the set of QSAT constraints constructed. It must necessarily also satisfy (40) on qubit w_d for some $\alpha \neq 0$. Observe that $y \neq 0$, as otherwise $x = 0$ as well (since $c_d \neq 0$), which is not permitted for homogeneous coordinates. Finally, since (40) implies $q(x, y) = 0$, we must have by homogeneity

$$p\left(\frac{x}{y}\right) = q\left(\frac{x}{y}, \frac{y}{y}\right) = \frac{q(x,y)}{y^d} = 0. \tag{41}$$

SDR. To see that the constructed QSAT instance has an SDR, note first that we can trivially make it 3-uniform by adding two ancilla qubits. Then, since all but the last recursive step of our construction simultaneously adds a new hyperedge and a new qubit, the system has an *almost extending edge order* (defined later in Definition 87). The claim now follows from Corollary 89. □

Remark 77. Theorem 76 is not yet for sparse polynomials, but it will nevertheless be instructive to recall that in the definition of SFTA, we focused on roots of polynomial $p \in \mathbb{C}[x]$ in range $[0, 1 + 2\log(d)/d]$, for d the degree. Given any root $x^* \in [0, 1 + 2\log(d)/d]$ of p, the constructed QSAT instance of Theorem 76 has a solution with

$$|v_1\rangle \propto (x^*, 1)^T. \tag{42}$$

The bounds $x^* \in [0, 1 + 2\log(d)/d]$ now ensure $\left\| (x^*, 1)^T \right\|_2$ is constant, so that the proportionality factor in Equation (42) is constant.

We now proceed to showing the sparse version of Theorem 76, but first remark that Theorem 76 suffices already to show NP-hardness results of slight variants of QSAT with SDR in Section 6.3.3.

The sparse case. The proof of the sparse case now proceeds analogously to the non-sparse case.

Theorem 70. *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 $|v_1\rangle = (x, y)^T \in \mathbb{C}^2$.*

Proof. The key observation is that in recursive step t of Equation (39), we factor x_{j_t} for $j_t > 0$ the minimal value satisfying $c_{j_t} \neq 0$. This implies the number of recursive calls scales with sparsity s, not degree d. Thus, a construction analogous to Theorem 76 can be used, except in the first set of constraints, we will need to construct $O(s \log d)$ terms $|v_i\rangle = (x^i, y^i)$, where i can now be exponential in the input size. This is easily handled by using square-and-multiply on qubits encoding the various $|v_i\rangle$ to obtain high powers i using $\log(i)$ steps (similar to Example 75). □

6.3.3 Detour: NP-hardness results for slight variants of QSAT with SDR

With Theorem 76 (non-sparse case) in hand, we first immediately obtain an NP-hardness result for a variant of QSAT with SDR. This complements Goerdt's result that deciding whether there exists a *real* product state solution is NP-hard [Goe19].

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.*

Proof. This theorem follows from the NP-hardness of deciding whether a sparse polynomial has a root of modulus 1 [Pla84]. □

Goerdt also shows that deciding whether a QSAT instance with an SDR and *just one* additional constraint is NP-hard. We can also recover this result here via our construction.

Theorem 7. *(c.f. [Goe19]) It is* NP*-hard to decide whether a* 3*-QSAT system with an SDR and one additional clause has a product state solution.*

Proof. Plaisted proves that it is NP-hard to decide whether two sparse polynomials have a common root [Pla84]. We can embed this problem into PRODSAT by adding a second adding a second polynomial in the above construction, which requires only a single unmatched edge. □

This stands in stark contrast to the classical setting, where deciding whether a CNF-SAT formula with an SDR and $O(1)$ additional clauses is still in P. The following theorem generalizes a result due to Berman, Karpinski, and Scott [BKS07], who prove that satisfiability of $(3, 4^{(k)})$-SAT (i.e. a 3-SAT instance in which k variables occur 4 times and the remaining variables 3 times) is efficiently solvable.

Theorem 78. *Let* $\mathscr{C}$ *be the set of clauses of a SAT instance in CNF on* n *variables* V *such that there exists a subset* $\mathscr{C}' \subseteq \mathscr{C}$ *with an SDR, i.e. a perfect matching between* $\mathscr{C}'$ *and* V*. Satisfiability of* $\mathscr{C}$ *can be determined in time* $(2n)^k \operatorname{poly}(n)$ *for* $k := |\mathscr{C}| - n$*.*

Proof. This proof follows the same outline as [BKS07, Theorem 1], but we need to give a different argument for the existence of a surjective witness function. Consider a satisfying assignment ϕ to $\mathscr{C}$ and define a *witness function* $w : \mathscr{C} \to V$ such that for each $C \in \mathscr{C}$, the variable $x = w(C)$ occurs in C and its literal evaluates to true under ϕ, i.e., if $\phi(x) = 1$, then C contains the literal x, and otherwise $\neg x$. We argue that if $\mathscr{C}$ is satisfiable, then there exists a satisfying assignment with a surjective witness function. Let ϕ be a satisfying assignment with witness function w. If w is surjective, we are done. Otherwise, there exists a variable $x \notin \operatorname{Im}(w)$. Let $C \in \mathscr{C}'$ be the clause assigned to x in the SDR. Create ϕ', w' by only changing $\phi(x)$ and $w(C)$ such that the literal of x in C evaluates to true and $w(C) = x$. ϕ' is still a satisfying assignment as $x \notin \operatorname{Im}(w)$ and $\phi'(y) = \phi(y)$ for all $y \neq x$. Repeat until $\operatorname{Im}(w) = V$, which takes at most n iterations since each iteration increases the number of clauses $C \in \mathscr{C}'$ such that $w(C)$ is matched with C in the SDR. The remainder of the proof is the same as [BKS07]. □

6.4 Is MHS in SFTA?

We now ask the question — could MHS $\subseteq$ SFTA? In words, can the solutions of a low-degree multi-homogeneous system be mapped to the roots of a high-degree univariate polynomial? We conjecture MHS $\not\subseteq$ SFTA, according to which no such efficient reduction should be possible. However, one can still show a non-trivial result in this direction — we show that in the generic setting (Definition 17), a low-degree multi-homogeneous system can be reduced to a single high-degree univariate polynomial p, where p requires polynomial *space* to compute. Under the hood, this utilizes a clever lemma of Canny, which we first state.

Lemma 79 (Canny's Lemma (Lemma 2.2 of [Can88]))**.** *Let p_1 through p_n be homogeneous polynomials in variables $x_0, \ldots x_n$, with $D \leq d_1 \cdots d_n$ isolated solution rays $(\alpha_{0,j}, \ldots, \alpha_{n,j})$, $j = 1, \ldots, D$. Let $N \leq D$ be the number of solution rays not at infinity, for example, with $\alpha_{0,j} \neq 0$. Then there is a univariate polynomial $q(x)$ of degree N, and rational functions $r_1(x), \ldots, r_n(x)$, such that every solution ray not at infinity is of form $(1, r_1(\theta), \ldots, r_n(\theta))$ for some root θ of $q(x)$. The polynomials $q(x)$ and $r_k(x)$ can be computed in polynomial space.*

We will also require two further tools from algebraic geometry (see, e.g., [CLO05]).

Definition 80 (Newton polytope (page 310 of [CLO05]))**.** Let $f \in \mathbb{C}[x_1, \ldots, x_n]$ be such that $f = \sum_{\alpha \in \mathbb{Z}^n_{\geq 0}} c_\alpha x^\alpha$. The Newton polytope of f is $\mathrm{Conv}(\{\alpha \in \mathbb{Z}^n_{\geq 0} \mid c_\alpha \neq 0\})$.

Theorem 81 (Berstein-Khovanksii-Kushnirenko (BKK; theorem 5.4 of [CLO05]))**.** *Given Laurent polynomials $f_1, \ldots f_n$ over $\mathbb{C}$ with finitely many common zeroes in $(\mathbb{C}^*)^n$, let P_i be the Newton polytope of f_i in $\mathbb{R}^n$. Then the number of common zeroes of the f_i in $(\mathbb{C}^*)^n$ is bounded above by the n-dimensional mixed volume of $(P_1, \ldots, P_n)$ (Definition 4.11 of [CLO05]). Moreover, for generic choices of the coefficients in the f_i, the number of common solutions is exactly the n-dimensional mixed volume of $(P_1, \ldots, P_n)$.*

We are now ready to prove the results of this section.

Proposition 82. *Let (G, w) be a weighted hypergraph with a WSDR and such that $|V(G)|_w = |E(G)|$. Let H be a generic instance of QSAT with underlying weighted hypergraph (G, w). Then every product ground state of H is of the form*

$$|\psi_t\rangle = (|0\rangle + t_{1,1}|1\rangle + \cdots t_{1,w(1)}|w(1)\rangle) \otimes \cdots \otimes (|0\rangle + t_{|V(G)|,1}|1\rangle + \cdots t_{|V(G)|,w(|V(G)|)}|w(|V(G)|)\rangle) \quad (43)$$

with $t_{i,j} \neq 0$ for all $i = 1, \ldots, |V(G)|$, and $j = 1, \ldots, w(i)$.

Proof. Let H_e be the clause corresponding to $e \in E(G)$ and consider the multivariate polynomial $p_e(t)$ in the variables $t_{i,j}$ such that $p_e(t) = 0$ if and only if $H_e|\psi_t\rangle = 0$. The Newton polytope Q_e of p_e is the product of simplices of dimension $w(i)$, one for each vertex $i \in e$. Hence for $\lambda_e > 0$, $e \in E(G)$,

$$V\left(\sum_{e \in E(G)} \lambda_e Q_e\right) = \prod_{i \in V(G)} \frac{\left(\sum_{v \in e} \lambda_e\right)^{w(i)}}{w(i)!} = N(G, w) \prod_{e \in E(G)} \lambda_e + \text{lower order terms} \quad (44)$$

where $N(G, w)$ is the number of WSDRs of (G, w). On the other hand, by definition, $N(G, w)$ is the mixed volume of the polytopes Q_e, $e \in E(G)$. Therefore, by the BKK theorem (Theorem 81), there are $N(G, w)$ product solutions of the form (43) with all $t_{i,j} \neq 0$. But since $N(G, w)$ is also equal to the Bézout number of the multi-homogeneous system associated with H, we conclude that, generically, this accounts for *all* product solutions of H. □

We can now show that, generically, QSAT with SDR can be reduced in polynomial space to solving for the roots of a single high degree univariate polynomial.

Theorem 83. *Let (G,w) be a weighted hypergraph with a WSDR and such that $|V(G)|_w = |E(G)|$. Let H be a generic instance of QSAT with underlying weighted hypergraph (G,w). Then there is a univariate polynomial $q(x)$ of degree at most*

$$D = \prod_{e\in E(G)} |e| \tag{45}$$

and rational functions $r_{i,j}(x)$ for every $i = 1,\ldots,|V(G)|$ and $j = 1,\ldots,w(i)$ such that if x is a root of q and

$$r(x) = \prod_{i=1}^{|V(G)|}\prod_{j=1}^{w(i)} r_{i,j}(x) \neq 0 \tag{46}$$

then

$$(|0\rangle + r_{1,1}(x)|1\rangle + \cdots + r_{1,w(1)}|w(1)\rangle) \otimes \cdots \otimes (|0\rangle + r_{|V(G)|,1}(x)|1\rangle + \cdots + r_{|V(G)|,w(|V(G)|)}|w(|V(G)|)\rangle) \tag{47}$$

is a product solution of H. Conversely, every product solution is of this form for some root x of q such that $r(x) \neq 0$. Moreover, $q(x)$ and all the rational functions $r_{i,j}(x)$ can be calculated in polynomial space.

Proof. Consider product solutions of H of the form

$$|\psi_t\rangle = (|0\rangle + t_{1,1}|1\rangle + \cdots t_{1,w(1)}|w(1)\rangle) \otimes \cdots \otimes (|0\rangle + t_{|V(G)|,1}|1\rangle + \cdots t_{|V(G),w(|V(G)|)}|w(|V(G)|)\rangle). \tag{48}$$

Let P_e be the homogenization of p_e obtained by adding the single variable t_0 so that $P_e = 0$ defines a hypersurface X_e of degree $|e|$ in $\mathbb{P}^{|V(G)|_w}$. By Canny's Lemma (Lemma 79), there is a polynomial $q(x)$ of degree $N \le D$ and rational functions $r_{i,j}(x)$ for every $i = 1,\ldots,|V(G)|$ and $j = 1,\ldots,w(i)$ such that every point in $(\bigcap_{e\in E(G)} X_e) \setminus \{t_0 = 0\}$ has coordinates $t_0 = 1$ and $t_{i,j} = r_{i,j}(x)$ whenever x is a root of $q(x)$. Then $r_{i,j}(x) = 0$ for some i and j if and only if the corresponding element of $\bigcap_{e\in E(G)} X_e$ belongs to one of the coordinate planes and thus represent a "spurious" solution in the sense that the corresponding product state (43) is not a solution of H (since by the BKK Theorem (Theorem 81) all product solutions satisfy the additional condition $t_{i,j} \neq 0$ for all i and j). The last statement of the claim follows directly from Canny's Lemma. □

Remark 84. When $w = 1$ (so that all qu-d-its are qubits), we can be more precise about the degree N of $q(x)$. By Canny's Lemma, $D - N$ is the number of points in the intersection of $\bigcap_{e\in E(G)} X_e$ with the hyperplane at infinity. On the other hand, setting $t_0 = 0$ drastically reduces the polynomial P_e to $\prod_{i\in e} t_i$. Let f be the Boolean function in CNF form with all positive literals and underlying hypergraph G. If n denotes the number of satisfying assignments of f, then $N = D - n + 1$.

7 Efficiently solvable special cases of QSAT with WSDR

We next give parameterized classical algorithms for QSAT with SDR, which allow for efficient solutions in special cases.

Brief overview of techniques. We first briefly sketch the ideas for two of our three algorithms for special cases of QSAT with (W)SDR. The first algorithm we discuss, which solves *non-generic* PRODSAT instances (Theorem 8), begins with the same approach as [AdBGS21]. At a high-level, this approach takes the qubits comprising the hard "core" of the instance, sets these qubits in a specific manner so as project onto a smaller space, and subsequently forces assignments onto all other qubits via transfer functions. This approach breaks down in the non-generic case, which can have unentangled constraints that can prevent this propagation of assignments. The classical analogue to this problem can be seen with constraint $x \vee y$: When $x = 1$, the constraint is already satisfied, and thus no assignment is propagated onto y. (Note all such classical SAT constraints are unentangled when embedded into QSAT.) To overcome this, the key idea we introduce is that, when this algorithm gets stuck, we prove that we can actually *recurse* the entire process, as its existing "almost extending order" remains valid.

The second algorithmic contribution we discuss is our extension of using *transfer filtrations* (the framework enabling transfer functions) to QSAT on qu*d*its. This requires a careful arrangement of clauses into a convenient order (exploiting the geometry of the instance) so as to reduce the problem to a system with fewer equations in fewer variables. The trade-off is that the degree of the resulting equations can be rather large. Nevertheless, we show that for certain non-trivial infinite families of interaction hypergraphs, such as the Pinwheel graph (Figure 5), we can efficiently solve the corresponding instance of PRODSAT, exponentially outperforming the brute force approach.

Organization. Section 7.1 introduces necessary definitions and lemmas. Section 7.2 solves *non-generic* special cases of QSAT on qubits with an SDR; this improves on [AdBGS21], which worked only for generic instances. Section 7.3 returns to the generic setting with SDR, but instead widens the class of qubit QSAT instances one can efficiently solve generically beyond [AdBGS21]. Section 7.4 shows how to extend the transfer filtration technique of [AdBGS21] from qu*b*its to qu*d*its and WSDRs, solving the Pinwheel graph in Section 7.5.1 exponentially faster than via brute force.

7.1 Transfer functions, filtrations, and extending edge orders

We begin by restating the notion of transfer functions for convenience:

Lemma 73. *(Transfer function, g) Let $|\phi\rangle$ be a k-local constraint on qubits. There exists a polynomial $g : (\mathbb{C}^2)^{k-1} \to \mathbb{C}^2$ such that, for any partial assignment $v_1, \ldots, v_{k-1}$, the clause $|\phi\rangle$ is satisfied (i.e. $\langle\phi|v_1, \ldots, v_k\rangle = 0$) iff*[16] *$|v_k\rangle \propto g(v_1, \ldots, v_{k-1})$ or $g(v_1, \ldots, v_{k-1}) = 0$. Moreover, g is linear in the coefficients of each v_i.*

Transfer filtrations. In the qubit setting, [AdBGS21] efficiently solves QSAT with SDR for *generic* instances of *transfer type* $b = n - m + 1$ (Definition 85 below), where m denotes the number of constraints and n the number of qubits. This transfer type restriction is important, as it allows [AdBGS21] to reduce the entire QSAT with SDR instance to approximating a root of a single univariate polynomial. Note also the algorithm is parameterized, i.e. its runtime is polynomial in the input size but exponential in the *foundation size* (Definition 85) and *radius* (Definition 86).

We begin by stating the required definitions, and give intuition as to why transfer type $b = n - m + 1$ allows reductions to the univariate polynomial case in [AdBGS21]. We first recall the

[16] $\propto$ means up to scaling up to non-zero constant.

definition of a *transfer filtration*, which is a particular type of hyperedge ordering useful for solving PRODSAT.

Definition 85 (Transfer filtration [AdBGS21])**.** A hypergraph $G = (V, E)$ is of *transfer type* b if there exists a chain of subhypergraphs (denoted a *transfer filtration of type* b) $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_m = G$ and an ordering of the edges $E(G) = \{e_1, \ldots, e_m\}$ such that

(1) $E(G_i) = \{e_1, \ldots, e_i\}$ for each $i \in \{0, \ldots, m\}$,

(2) $|V(G_i)| \leq |V(G_{i-1})| + 1$ for each $i \in \{1, \ldots, m\}$,

(3) if $|V(G_i)| = |V(G_{i-1})| + 1$, then $V(G_i) \setminus V(G_{i-1}) \subseteq e_i$,

(4) $|V(G_0)| = b$, where we call $V(G_0)$ the *foundation*,

(5) and each edge of G has at least one vertex not in $V(G_0)$.

Definition 86 (Radius of transfer filtration [AdBGS21])**.** Let G be a hypergraph admitting a transfer filtration $G_0 \subseteq \cdots \subseteq G_m = G$ of type b. Consider the function $r : \{0, \ldots, m\} \to \{0, \ldots, m-1\}$ such that $r(0) = 0$ and $r(i)$ is the smallest integer such that $|e_i \setminus V(G_{r(i)})| = 1 \ \forall i \in \{1, \ldots, m\}$. The *radius of the transfer filtration* $G_0 \subseteq \cdots \subseteq G_m = G$ *of type* b is the smallest integer β such that $r^\beta(i) = 0$ for all $i \in \{1, \ldots, m\}$ (r^β denotes composition of r with itself β times). The *type* b *radius of* G is the minimum value $\rho(G, b)$ of β over the set of all possible transfer filtrations of type b on G.

Intuition. We can view the transfer filtration as a sequence of edges wherein each edge adds at most one extra node, as enforced by condition (2) above. The foundation is made up by all but one of the vertices in edge e_1. Then transfer type $b = n - m + 1$ implies that $n = b + m - 1$ and thus one edge does not add an additional vertex (i.e. $V(G_i) = V(G_{i+1})$ in (2)). Note, given a product assignment to the qubits in $V(G_{i-1})$, we can satisfy the constraint of edge e_i using the corresponding transfer function (see Lemma 73). This leaves a single *non-extending* constraint that does not add a new qubit, and thus cannot immediately be satisfied. To solve the system, assign the foundation qubits $v_1 = \cdots = v_{b-1} = |0\rangle$, and $v_b = |0\rangle + x|1\rangle$. The transfer functions then set each qubit to a polynomial expression in x. Satisfying the non-extending constraint then reduces to finding a root of a univariate polynomial of degree exponential in the radius. Note, the above algorithm outline does not quite match [AdBGS21], where qubits are duplicated so that every edge adds a new qubit and then equality of copies is enforced via *qualifier constraints*.

Extending edge order. As outlined above, the transfer filtration gives us an order of the constraints that we can use to solve the system. We formalize this notion by defining the *extending edge order*, which turns out to be equivalent to the transfer filtration, but is useful in handling vanishing transfer functions algorithmically.

Definition 87 (Extending Edge Order)**.** Let $G = (V, E)$ be a hypergraph. An edge order $e_1, \ldots, e_m$ is *extending* if $e_i \setminus V_{i-1} \neq \emptyset$ for $i \in [m]$, where $V_i \coloneqq \bigcup_{j=1}^{i} V(e_i)$ and $V_0 = \emptyset$. We say the order is a*-almost extending* if $|\{i : V_i = V_{i-1}\}| \leq a$. We say it is *almost extending* if $a = 1$.

Lemma 88. *Let $G = (V, E)$ be a hypergraph, b^* its minimum transfer type and a^* minimal such that G has an a^*-almost extending edge order. Then $b^* = n - m + a^*$.*

Proof. First, show $a^* \leq b^* - n + m$ by constructing an a-almost extending order given a transfer filtration of type $b = n - m + a$. Let $G_0 \subseteq \cdots \subseteq G_m = G$ be a transfer filtration of type b. By Definition 85, $E(G_i) = \{e_1, \dots, e_i\}$. Let $V_i = \bigcup_{j=1}^{i} e_j$. Then $V(G_i) = V_i \cup V(G_0)$. We have $n = b + m - a$. So if $a = 0$, every edge must cover one additional vertex and $e_1, \dots, e_m$ is an extending edge ordering. If $a > 0$, then there are exactly a edges that do not cover a new vertex, since one edge can add at most one new vertex. Let $i_1 < \cdots < i_{m-a}$ the indices of edges that add a new vertex (i.e. $|V(G_i)| = |V(G_{i-1})| = 1$), and $j_1 < \cdots < j_a$ the indices of the remaining edges (i.e. $V(G_i) = V(G_{i-1})$). Note, by definition e_1 always adds at least one vertex. Then $e_{i_1}, \dots, e_{i_{m-a}}, e_{j_1}, \dots, e_{j_a}$ is a-almost extending.

Second, we show $b^* \leq n - m + a^*$. Let $e_1, \dots, e_m$ be an a^*-almost extending order. Without loss of generality, $e_1, \dots, e_{m-a^*}$ are extending. Define vertices $u_1, \dots, u_{m-a^*}$ such that $u_i \in e_i \setminus V_{i-1}$. Then we argue a valid foundation is given by the "redundant vertices" $R \coloneqq \bigcup_{i=1}^{m-a^*} (e_i \setminus V_{i-1} \setminus \{u_i\})$. Hence, the transfer filtration is defined with $V(G_0) = R$ and $E(G_i) = \{e_1, \dots, e_i\}$. The transfer type is then $b = |R| = n - m + a^*$. Conditions (1) to (4) are satisfied by construction. For condition (5) we have to show that $e \nsubseteq R$ for all $e \in E$. For an extending edge e_i, we have $u_i \notin R$, because $u_i \notin V_{j<i}$ and $u_i \in V_{j \geq i}$, and thus $u_i \notin R$. For a non-extending edge e_i, we argue that that $e_i \subseteq R$ would violate minimality of a^*: Suppose there exists a minimal j such that $e_i \subseteq R_j \coloneqq \bigcup_{i=1}^{j} (e_i \setminus V_{i-1} \setminus \{u_i\})$. Then we could construct a new $(a^* - 1)$-almost extending edge order by moving e_i in between e_{j-1} and e_j. Then e_i would be extending because it contains at least one of the "redundant vertices" of e_j and e_j is still extending as it adds u_j. The edges $e_{j+1}, \dots, e_{m-a^*}$ remain extending because $e_i \subseteq R_j \subseteq V_j$. □

Finally, we state a corollary which we used in Section 6.3.

Corollary 89 ([AGS21])**.** *Let G be a k-uniform hypergraph for any $k > 0$. If G has an almost extending edge order, then G has an SDR.*

Proof. This follows immediately from Lemma 88 and the fact that if G is a k-uniform hypergraph of transfer type b and such that $|E(G)| = |V(G)| - b + 1$, then G has an SDR [AGS21]. □

7.2 Solving non-generic instances on qubits of transfer type $b = n - m + 1$

We now introduce an efficient algorithm for QSAT with SDR on qubits without genericity requirements, i.e. that can handle constraints which are "edge cases" (e.g. Schmidt rank-1 or unentangled constraints). For this, we define the radius of an almost extending edge order as the radius of the transfer filtration constructed in the proof of Lemma 88.

The challenge. In the non-generic case, one issue we need to deal with is that transfer functions can become 0, i.e., after assigning the first $k-1$ qubits of a k-local constraint, the corresponding constraint is already satisfied for every choice of the k-th qubit (this is the case of $g = 0$ in Lemma 73). For example, $|\phi\rangle = |000\rangle_{123}$ with $|v_1\rangle = |1\rangle$ is satisfied for all choices of $|v_2\rangle$ and $|v_3\rangle$. As a result, assignments to a subset of qubits are not propagated throughout the system. This issue is circumvented in [AdBGS21] through the genericity assumption, which we shall remove.

The algorithm. The next theorem generalizes the algorithm of [AdBGS21, Section 4.4], which solved *generic* instances of transfer type $b = n - m + 1$. We say a product state $|\psi\rangle = |\psi_1, \dots, \psi_n\rangle$ is an ϵ-approximate solution to a PRODSAT instance if $|\Pi_i|\psi\rangle| \leq \epsilon$ for all constraints Π_i. We require an approximately normalized solution, i.e., $\langle\psi_i|\psi_i\rangle \in [1-\epsilon, 1+\epsilon]$ for all $i \in [n]$. The error incurred by normalization was not considered in [AdBGS21]. Here we handle this issue by mostly computing with exact representations of algebraic numbers.

Theorem 90. *Let Π be a QSAT instance on qubits with coefficients in $\mathbb{Q}[i]$ such that the constraint hypergraph G has an almost extending edge order of radius r, and edges have size at most k. Then an ϵ-approximate solution can be computed in time $\mathrm{poly}(L, \log\epsilon^{-1}, k^r)$, where L is the input size. For sufficiently generic instances, an exact representation of a solution can be obtained.*

Before giving the proof, a comment on the dependence of the runtime above on radius r: The function r in the definition of radius divides the edges into layers such that layer β consists of the edges such e_i such that $r^\beta(i) = 0 \neq r^{\beta-1}(i)$. Note, the radius generally depends on the choice of vertices $u_1, \ldots, u_{m-1}$. Kremer [Kre24] gives a poly-time algorithm to compute an almost extending edge order and choice of vertices $u_1, \ldots, u_{m-1}$ that minimize the radius.

Proof of Theorem 90. Let $e_1, \ldots, e_m$ be an almost extending edge order such that e_m is the single non-extending constraint. Let $u_1, \ldots, u_{m-1}$ be defined as in the proof of Lemma 88. We also assume that $u_{m-1} \notin e_m$, i.e., $e_m \nsubseteq e_1 \cup \cdots \cup e_{m-2}$. This is valid because once we have found a product solution that satisfies the non-extending constraint, it becomes trivial to add more extending constraints and find product assignments for the added qubits that satisfy the added constraint.

Next we describe the algorithm. Let R be the set of "redundant" vertices as in the proof of Lemma 88. We say a vertex v depends on a vertex u if we reach v from u when following the edge order. There must be at least one vertex $u_0 \in R$, such that u_{m-1} *depends* on u_0, even after removing $R \setminus \{u_0\}$.

Next, add a 1-local constraint $|1\rangle$ to all qubits in $R \setminus \{u_0\}$. Assign all qubits corresponding to vertices $v \in R \setminus \{u_0\}$ to $|0\rangle$. Next, remove all 1-local constraints (hyperedges of size 1) on vertices besides u_{m-1} by assigning the orthogonal state to the corresponding qubit and reducing the remaining constraints. The resulting edge order remains almost extending, although there may now be a single 1-local constraint on u_{m-1}. However, either e_m or e_{m-1} remains of size ≥ 2 because u_{m-1} depends on u_0 and 1-local residual constraints on a vertex u_i are only created after all vertices in $e_i \setminus \{u_i\}$ have been assigned, which is not possible on the path from u_0 to u_{m-1}. Repeat these two steps until either the edge order is extending or we obtain an almost extending edge order with a single redundant vertex u_0.

We may now assume that u_0 is the single redundant vertex, and therefore $u_0 \in e_1$. Then via the transfer functions, we can write any vertex as a homogeneous polynomial in the amplitudes of the qubit u_0, i.e., $g_i(u_0) = u_i$ (see Lemma 73). For a satisfying assignment, we have $g_{m-1}(u_0) = \lambda g_m(u_0)$ (for some $\lambda \in \mathbb{C}^*$), or equivalently $q(u_0) = f_{m-1}(u_0)^T \cdot g_m(u_0) = 0$, where f_{m-1} is defined as in Lemma 73. q is then a homogeneous polynomial of degree at most $(k-1)^r$ (see [AdBGS21] for more details). q is not constant since u_{m-1} depends on u_0 and so one of e_{m-1}, e_m is not 1-local and f_{m-1} or g_m is not constant. First, we check whether $|u_0\rangle = |0\rangle$ gives an ϵ-approximate solution. If not, let $|u_0\rangle = x|0\rangle + |1\rangle$ and compute a root x of $q(x) \coloneqq q(x|0\rangle + |1\rangle)$. x has an exact representation in the field of algebraic numbers, which can be obtained in polynomial time in the degree and description size [AS20, Theorem 8]. After computing x, we can compute the $g_i(x)$ with [AS20, Theorem 4]. As argued in [AdBGS21], we have $g_i(x) \neq 0$ for all i if the constraint system is chosen generically, and we have an exact representation of a PRODSAT solution.

However, for non-generic instances, we can have $g_i(x) = 0$. In that case, compute the non-zero $g_1(u_0), \ldots, g_m(u_0)$ up to significant $\tau \geq \mathrm{poly}(m \log\epsilon^{-1})$ bits in polynomial time (in τ and the bit size of the constraints) using [AS20, Theorem 2] to compute the and [AS20, Proposition 1] to lower bound the non-zero values.[17]

For all $i = 0, \ldots, m-2$, assign $|u_i\rangle = g_i(x)$ if $g_i(x) \neq 0$. Then reduce the remaining constraints and again compute the amplitudes up to τ significant bits and then normalize. The additive error in the assigned qubits and the reduced constraints is then $\mathrm{poly}(\epsilon/m)$ for a sufficiently large τ. We have to reduce the system so that it either becomes extending or remains almost extending. First note that the reduction produces no 1-local constraints on a vertex u_i with $i < m-1$, because then we would have $g_i(x) \neq 0$. Thus, the remaining reduced edges from $e_1, \ldots, e_{m-2}$ are still extending. If both $g_{m-1}(x) \neq 0$ and $g_m(x) \neq 0$, then we can assign $|u_{m-1}\rangle = g_{m-1}(x) = g_m(x)$ and the remaining edge order is extending. If $g_{m-1}(x) = g_m(x) = 0$, then the order remains almost extending. If $g_{m-1}(x) = 0$ and $g_m(x) \neq 0$ (or vice versa), then we obtain a new 1-local constraint on u_{m-1}. But only one of e_{m-1}, e_m becomes 1-local, and thus we can solve the residual system recursively. In total, we need at most r recursions. The error increases additively with each recursion, so the total error is at most $\mathrm{poly}(\epsilon)$: Assuming we can compute a solution with error ϵ' on the residual system, we get total error $\epsilon' + \mathrm{poly}(\epsilon/m)$. □

7.3 Solving generic instances on qubits of transfer type $b = n - m + k - 1$

Section 7.2 showed how to improve on the paramaterized algorithm of [AGS21] by keeping the transfer type fixed to $b = n - m + 1$, but extending to non-generic instances. Here, we do the opposite — we give a parameterized algorithm for the generic case, but now extend the set of transfer types we are able to handle to $b = n - m + k - 1$, so that for any constant k, we obtain an efficient algorithm (under the assumption, as before, that radius $r \in O(\log n)$).

Lemma 91. *Let H be a generic PRODSAT instance on qubits with underlying hypergraph $G = (V, E)$, such that G has an SDR and $|V| = |E|$. Then G has $d_{Béz}$ product solutions, and none of these solutions* breaks *any transfer function (i.e. no transfer function in G maps a solution of G to 0).*

Proof. Consider some constraint $|\phi\rangle$ corresponding to the edge $e = \{v_1, \ldots, v_k\} \in E$ on qubits $1, \ldots, k$ and let $t\colon (\mathbb{C}^2)^{k-1} \to \mathbb{C}^2$ be the associated transfer function from qubits $v_1, \ldots, v_{k-1}$ to v_k. We can write $|\phi\rangle = |\phi_0\rangle_{v_1,\ldots,v_{k-1}}|0\rangle_{v_k} + |\phi_1\rangle_{v_1,\ldots,v_{k-1}}|1\rangle_{v_k}$. Then $t(v_1, \ldots, v_{k-1}) = 0$ iff $\langle\phi_0|v_1, \ldots, v_{k-1}\rangle = \langle\phi_1|v_1, \ldots, v_{k-1}\rangle = 0$, where $v_i \in \mathbb{C}^2$ also denotes the assignment to qubit v_i. Denote by H' the PRODSAT instance obtained by replacing constraint $|\phi\rangle_e$ by $|\phi_0\rangle_{e'}$ and $|\phi_1\rangle_{e'}$, where $e' = \{v_1, \ldots, v_{k-1}\}$, and let $G' = (V', E')$ be its underlying hypergraph. The product solutions of H' are precisely the product solutions of H that also break the transfer function t. Since $|\phi\rangle$ is the direct sum of $|\phi_0\rangle$ and $|\phi_1\rangle$ (up to permutation), the coefficients of $|\phi\rangle$ split into two disjoint subsets: the coefficients of $|\phi_0\rangle$ and those of $|\phi_1\rangle$. Hence, H' is still generic. Since $|V'| < |E'|$, H' does not have an SDR and generically no solutions by [LLM+10]. Thus, H' there exists a polynomial g' in the coefficients of H' such that H' is unsolvable if $g'(\cdot) \neq 0$. There also exists a polynomial g in the coefficients of H, such that H has exactly $d_{Béz}$ solutions if $g(\cdot) \neq 0$. Since H and H' have the same coefficient set, we have that H has no solution that breaks the transfer function t if $gg'(\cdot) \neq 0$. By the same argument, generically none of the solutions of H break *any* transfer function. □

[17] The reason for rounding to the rationals is that if we continue in the exact regime, the degree of algebraic numbers grows doubly exponentially in the number of recursions because every application of [AS20, Theorem 8] introduces a new algebraic number whose degree is only bounded by the product of the previous solutions.

Lemma 92 ([AdBGS21])**.** *Let $G = (V, E)$ be a k-uniform hypergraph of transfer type $b = n-m+k-1$ (equivalently, an $(k-1)$-almost extending edge order). Then G has an SDR.*

Theorem 93. *Let H be a generic PRODSAT instance with constraints in $\mathbb{Q}[i]$ on qubits with underlying k-uniform hypergraph $G = (V, E)$ of transfer type $b = n - m + k - 1$ (equivalently, a $(k-1)$-almost extending edge order) with radius r. We can compute an ϵ-approximate product state solution in time $\text{poly}(L, |\log \epsilon|, k^r, m^k)$, where L is a bound on the bit size of the instance's rational coefficients, and ϵ the Euclidean distance to the closest product state solution.*

Proof. Kremer [Kre24] gives a polynomial time algorithm to compute an edge order with minimum radius as well as the corresponding transfer filtration. The key insight is that the last vertex in an extending edge order must have degree 1, which allows us to greedily partition the edges into layers, starting with all edges containing a vertex of degree 1 as last layer. By trying all combinations for the $k-1$ non-extending constraints, we can compute the $(k-1)$-almost extending edge order of minimum radius in time $m^{O(k)}$.

Observe that every transfer function depends on at least $k-1$ foundation variables. Via the transfer functions, we can write all qubits as a polynomial in the foundation qubits of degree at most $(k-1)^r$. Hence, every non-extending constraint is a polynomial in at least $k-1$ variables, of degree at most $k(k-1)^r$. The next step is to remove foundation qubits so that there exists a finite number of solutions generically, while maintaining the existence of an SDR. We argue that G has an SDR matching only $k-1$ of the foundation vertices $V(G_0)$.

Let $\widetilde{e}_1, \ldots, \widetilde{e}_{k-1}$ be the non-extending edges, and choose distinct vertices $\widetilde{v}_1, \ldots, \widetilde{v}_{k-1}$, such that $\widetilde{v}_i \in \widetilde{e}_i$ for $i = 1, \ldots, k-1$. These exist by Hall's marriage theorem. For each extending edge $e_i \in E$, let $v(e_i) \in V(G_i) \setminus V(G_{i-1})$ be the added vertex. Construct a directed graph $\widehat{G} = (\widehat{V}, \widehat{E})$ with $\widehat{V} = V \cup \{\widehat{u}, \widehat{v}\}$ and

$$\begin{aligned}\widehat{E} = \{(\widehat{u}, v) \mid v \in V(G_0)\} \cup \{(\widetilde{v}_i, \widehat{v}) \mid i \in [k-1]\} \\ \cup \{(u, v(e)) \mid e \in E \setminus \{\widetilde{e}_1, \ldots, \widetilde{e}_{k-1}\}, u \in e \setminus \{v(e)\}\},\end{aligned} \tag{49}$$

i.e., edges from $\widehat{u}$ to the foundation, edges from all nodes in a hyperedge e to the added vertex $v(e)$, and edges from the $\widetilde{v}_1, \ldots, \widetilde{v}_{k-1}$ to $\widehat{v}$. Note that each vertex in $V \subseteq \widehat{V}$ has at least $k-1$ incoming edges from a "lower layer". Hence, one has to remove at least $k-1$ vertices from $\widehat{G}$ to disconnect $\widehat{u}, \widehat{v}$. By Menger's theorem, there exist $k-1$ internally disjoint paths from $\widehat{u}$ to $\widehat{v}$. By construction, each of these paths goes via a foundation vertex $u_i \in V(G_0)$ to $\widetilde{v}_i$. We can construct an SDR by matching $\widetilde{e}_i$ to $\widetilde{v}_i$, then matching e s.t. $v(e) = \widetilde{v}_i$ to the predecessor of $\widetilde{v}_i$ in the path. Iterate until reaching the foundation. We can assign all remaining edges e to $v(e)$, since their $v(e)$ are outside the $k-1$ paths. Set the unmatched foundation qubits to $|0\rangle$ and let the resulting system be H' on graph $G' = (V', E')$. The SDR constructed above is also valid for G'. H' still has generic constraints, since setting variables to $|0\rangle$ just means we discard coefficients, but not change them.

Let F be the multi-homogeneous system obtained by writing every qubit of G' as polynomials in the entries of the foundation qubits via the transfer functions. The solutions of F also contains the foundation qubits of all solutions of H', which can be extended to the qubits outside the core via the transfer functions. However, the solution set of F can also contain assignments to the foundation that break transfer functions. By Lemma 91, none of the transfer functions are broken if the foundation is set to an actual solution to H'. An additional polynomial inequality g of degree at most $n(k-1)^r$ ensures that we only find solutions that break no transfer functions. We can use the existential theory of the reals to find a solution that satisfies both F and g. For rational entries, Renegar's algorithm [Ren92, Theorem 1.2] can compute an ϵ-approximate solution in time poly$(L, k^r, |\log \epsilon|)$, where L is a bound on the bit size of the constraints. We introduce separate variables for the real and imaginary parts, which allows us to also use complex conjugates in our constraints. □

7.4 Solving higher dimensional systems via weighted transfer filtrations

Finally, we show how to extend the technique of transfer filtrations (Definition 85) from qu*b*its to qu*d*its, and give an explicit family of high-dimensional QSAT with WSDR instances which we can solve exponentially faster than brute force (Section 7.5.1).

The basic idea is to still consider a hypergraph with a filtration $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_m = G$ but now allowing for the addition of more edges, and potentially more vertices, at each step in the filtration. The most straightforward generalization is to maintain the requirement $|V(G_i)| \leq |V(G_{i-1})| + 1$ for each $i \in \{1, \ldots, m\}$ but, in the case in which $|V(G_i)| = |V(G_{i-1})| + 1$ to allow for as many edges to be added as the weight of the new vertex (while maintaining the provision that each new edge must contain the new vertex). These type of *weighted transfer filtrations* can be used, to explicitly (and in some cases, depending on the growth of the radius, efficiently) construct solutions to the corresponding instances of PRODSAT along the lines of Section 7.2.

More generally we can relax the condition $|V(G_i)| \leq |V(G_{i-1})| + 1$ to the requirement that the induced subhypergraph of G_i induced by $V(G_i) \setminus V(G_{i-1})$ has itself a transfer filtration of type $b = n - m + 1$. As formalizing the high-dimensional case in full generality becomes technically cluttered, for pedagogical purposes we instead demonstrate the idea with concrete examples.

Qubits on a 1D periodic lattice. In order to set up the notation for more general examples, we begin by considering a system of n qubits located at the vertices of a 1D periodic lattice, i.e. a cycle of length n. This system is efficiently solvable via transfer functions [Bra06, BG16] using transfer functions, along the following lines. We parametrize the i-th qubit state as $x_i^0|0\rangle + x_i^1|1\rangle$, for $i \in \mathbb{Z}/n\mathbb{Z}$. Each edge corresponds to a 2-local QSAT constraint φ_i of the form

$$\sum_{p,q=0}^{1} \varphi_i^{pq} x_i^p x_{i+1}^q = 0\,. \tag{50}$$

Passing to affine coordinates $z_i = \frac{x_i^0}{x_i^1}$, this translates to

$$z_{i+1} = -\frac{\varphi_i^{01} z_i + \varphi_i^{11}}{\varphi_i^{00} z_i + \varphi_i 10} \tag{51}$$

which, after n iterations, leads to an expression of the z_i as solution of a quadratic equation $a_i z_i^2 + b_i z_i + c_i = 0$ whose coefficients coefficients a_i, b_i, c_i are multilinear polynomials of total degree n in the variables φ_i^{pq}.

Qutrits on a 1D periodic lattice. Our next stepping stone is to keep the same interaction hypergraph (the 1D periodic lattice), but to allow qubits to be replaced by n qu*t*rits. We parametrize the i-th qutrit as $x_i^0|0\rangle + x_i^1|1\rangle + x_i^2|2\rangle$, $i \in \mathbb{Z}/n\mathbb{Z}$. Each edge corresponds to a 2-local QSAT constraint

φ_i of the form

$$\sum_{p,q=0}^{2} \varphi_i^{pq} x_i^p x_{i+1}^q = 0\,. \tag{52}$$

We can further impose 1-local constraints on each qutrit, which, in terms of affine coordinates $z_i^0 = \frac{x_i^0}{x_i^2}$, $z_i^1 = \frac{z_i^1}{z_i^2}$ can be written as $z_i^0 = \alpha_i^1 z_i^1 + \alpha_i^2$. Substituting into the constraint we obtain

$$z_{i+1}^1 = -\frac{A_i z_i^1 + B_i}{C_i z_i^1 + D_1} \tag{53}$$

where

$$\begin{aligned}
A_i^1 &= \varphi_i^{00}\alpha_i^1\alpha_{i+1}^2 + \varphi_i^{10}\alpha_{i+1}^2 + \varphi_i^{02}\alpha_i^1 + \varphi_i^{12} \\
B_i^1 &= \varphi_i^{00}\alpha_i^2\alpha_{i+1}^2 + \varphi_i^{02}\alpha_i^2 + \varphi_i^{20}\alpha_{i+1}^2 + \varphi_i^{22} \\
C_i^1 &= \varphi_i^{00}\alpha_i^1\alpha_{i+1}^1 + \varphi_i^{10}\alpha_{i+1}^1 + \varphi_i^{01}\alpha_i^1 + \varphi_i^{11} \\
D_i^1 &= \varphi_i^{00}\alpha_i^2\alpha_{i+1}^1 + \varphi_i^{01}\alpha_i^2 + \varphi_i^{20}\alpha_{i+1}^1 + \varphi_i^{21}
\end{aligned}$$

After n iterations, we obtain the z_i^1 as solutions of quadratic equations whose coefficients are polynomials of total degree 3n, linear in each of the φ_i^{pq} and quadratic in each of the α_i^{pq}s.

Qutrits on a 2D periodic lattice. Now we are ready to describe our first example of genuinely more general transfer filtrations in presence of qutrits. Specifically, consider now a system of mn qutrits located at the vertices of a square lattice with periodic boundary conditions. We parametrize the qutrit on the (i,j) node of the lattice as

$$x_{i,j}^0|0\rangle + x_{i,j}^1|1\rangle + x_{i,j}^2|2\rangle \tag{54}$$

for all $i \in \mathbb{Z}/m\mathbb{Z}$ and $j \in \mathbb{Z}/n\mathbb{Z}$. We have "horizontal" 2-local constraint

$$\sum_{p,q=0}^{2} \varphi_{i,j}^{p,q} x_{i,j}^p x_{i+1,j}^q = 0 \tag{55}$$

as well as "vertical" ones

$$\sum_{p,q=0}^{2} \psi_{i,j}^{p,q} x_{i,j}^p x_{i,j+1}^q = 0 \tag{56}$$

for each $i \in \mathbb{Z}/m\mathbb{Z}$ and $j \in \mathbb{Z}/n\mathbb{Z}$. We work in affine coordinates $z_{i,j}^0 = \frac{x_{i,j}^0}{x_{i,j}^2}$ and $z_{i,j}^1 = \frac{x_{i,j}^1}{x_{i,j}^2}$ and impose arbitrary 1-local constraints on the qutrits of one of the "rows" of the lattice, say, $z_{i,0}^0 = \alpha_{i,0}^1 z_{i,0}^1 + \alpha_{i,0}^2$ for all $i \in \mathbb{Z}/m\mathbb{Z}$. Then we solve the 0-th row using the method outlined above expressing each $z_{i,0}^1$ as a solution of a quadratic equation with coefficients of total degree $2m$ in the alphas. Then imposing the $\psi_{i,0}$ constraints, we obtain constraints of the form $z_{i,1}^0 = \alpha_{i,1}^1 z_{i,1}^1 + \alpha_{i,1}^2$ where the $\alpha_{i,1}^p$ are fractions with both numerator and denominator are linear in the $z_{i,0}^1$. Iterating this process n-times we can solve the rows one by one in terms of the $\alpha_{i,0}^p$ until, thanks to the periodic boundary conditions, return to $z_{i,0}^p$. This results to a system of equations in the $\alpha_{i,0}^p$ whose degree is (simply) exponential in n.

7.5 Weighted graphs with constant weights

The example of qutrits on a 2D periodic lattice can be generalized to qudits of local dimension d on a periodic $(d-1)$-dimensional lattice, i.e. on the weighted graph $(C_{m_1} \Box C_{m_2} \Box \cdots \Box C_{m_N}, d-1)$, for $\Box$ the graph Cartesian product (Definition 29). This can be done iteratively. For instance, when $d = 4$, and the corresponding graph is $C_{m_1} \Box C_{m_2} \Box C_{m_3}$, we can isolate a 2-dimensional slice, say, $C_{m_1} \Box C_{m_2} \Box \{1\}$, impose 1-local constraints on each of its vertices, solve using the method above, and then use the constraints corresponding to edges "orthogonal" to the 2D slice to reduce by one unit the local dimension of the qudits of the slice $C_{m_1} \Box C_{m_2} \Box \{2\}$ and repeat.

More generally, one can replace the cyclic graphs C_m with pseudoforests (i.e. a disjoint union of graphs having at most one cycle). This is because [Bra06, ASSZ16, BG16], instances of 2-QSAT on qubits whose interaction graph is a pseudoforest are solvable in linear time. Moreover we know that, since pseudoforests have SDRs and the property of admitting a WSDR is preserved under cartesian products, the cartesian product of N pseudoforests admits a WSDR with constant weight $w = N$.

Consider a graph G together with a finite filtration by subgraphs $G_0 \subseteq G_1 \subseteq \cdots \subseteq G_n = G$ constructed as follows. First, we let G_0 (the foundation) be a graph with no edges. Then let P_0 be an arbitrary pseudoforest. Then G_1 is constructed by adding edges to $G_0 + P_0$ connecting vertices of G_0 to vertices of P_0 with the provision that the degree of the vertices of P_0 increases at most by one. Similarly, G_2 is constructed by taking the disjoint union of G_1 with a pseudoforest P_1 and adding edges to $G_1 + P_1$ connecting vertices of G_1 to vertices of P_0 in a way that the degree of the vertices of P_1 increases by at most one unit. And so forth.

7.5.1 An explicit example with exponential speedup: The Pinwheel graph

The goal of our next example is to illustrate how a modification of the 2D lattice construction can give rise to an infinite family of instances of 2-QSAT on qutrits that are efficiently solvable.

For each positive integer n, consider the graph Γ_n, which we refer to as a *Pinwheel graph* (Figure 5). The vertices are v_0, located at the origin and $v_{j,k}$ located at the point in the plane with polar coordinates $(j, 2^{1-j}\pi k)$ for all $j = 1, \ldots, n$ and $k \in \mathbb{Z}/2^j\mathbb{Z}$.

There are three kinds of edges:

1. $e_{j,k}$ connecting $v_{j,k}$ to $v_{j,k+1}$ for each $k \in \mathbb{Z}/2^j\mathbb{Z}$ (colored in black in the picture);
2. $\epsilon_{j,k}$ connecting $v_{j,k}$ to $v_{j-1,k/2}$ if k is even and to $v_{j-1,k-1/2}$ if k is odd (colored in blue in the picture);
3. ε_i connecting $v_{n,2^{n-i}-1}$ to v_0 for $i \in \mathbb{Z}/2\mathbb{Z}$ (colored in green in the picture).

Γ_n has a total of $1+2+4+\cdots 2^n = 2^{n+1}-1$ vertices and $(2^{n+1}-2)+(2^{n+1}-2)+2 = 2(2^{n+1}-1)$ edges. Hence placing a qutrit at each vertex and a 2-local constraint at each edge we obtain a system with as many degrees of freedom as constraints and thus finitely many solutions.

Moreover, Γ_n has a natural WSDR with constant weight $w = 2$ defined by $f(\varepsilon_0) = v_0 = f(\varepsilon_1)$ and $f(e_{j,k}) = v_{j,k} = f(\epsilon_{j,k})$ for all $j = 1, \ldots, n$ and $k = 1, \ldots, 2^j$.

Starting with an arbitrary assignment of the qutrit located at v_0 and imposing the constraints corresponding to the edges $\epsilon_{1,\bullet}$ we reduce the qutrits located at $v_{1,\bullet}$ to qubits subject to the 2-local constraints corresponding to the edges $e_{1,\bullet}$ This is a 1D periodic lattice of qubits that can be solved in linear time. Imposing the constraints corresponding to the edges $\epsilon_{2,\bullet}$ we reduce the qutrits located

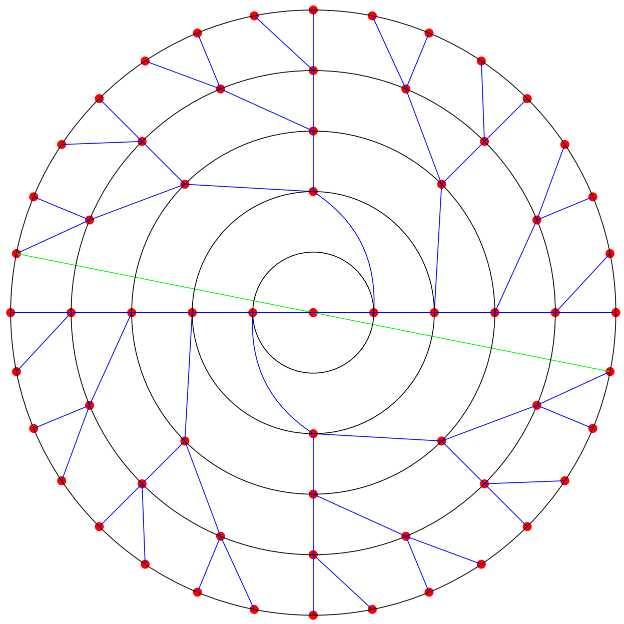

Figure 5: Pinwheel graph Γ_n for the case of $n = 5$.

at $v_{2,\bullet}$ to qubits and iterate the previous until we have a product assignments for all qutrits in terms of the initial assignment at v_0 that satisfies all e and ϵ constraints. At this point we impose the $\varepsilon_\bullet$ constraints and realize admissible assignments at v_0 as the solution of a system of two polynomial equations in two variables. This can be solved using, say, the resultant (see, e.g. [CLO15]). Note that both the degree of these polynomials and the number of degrees of freedom grows (simply) exponentially with n.

Acknowledgements

We thank Niel de Beaudrap, Neal Bushaw, Bruno Grenet, David Gosset, Christian Ikenmeyer, Pascal Koiran, Grégoire Lecerf 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 93. SG was supported by the 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 Northrhine Westphalia. DR was supported by the DFG under grant number 432788384. MA was supported in part by VCU Quest Award "Quantum Fields and Knots: An integrative Approach". 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.

References

[AGS21] 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, 2021.

[AdBGS21] 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, Jan 2021.

[AGIK09] 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.

[APT79] 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.

[AS20] P. E. Alaev and V. L. Selivanov. Fields of algebraic numbers computable in polynomial time. i. *Algebra and Logic*, 58(6):447–469, Jan 2020.

[ASSZ16] 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 fuer Informatik.

[BBT09] 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.

[BG16] 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 fuer Informatik.

[BEKT22] Evangelos Bartzos, Ioannis Z. Emiris, Ilias S. Kotsireas, and Charalambos Tzamos. Bounding the number of roots of multi-homogeneous systems. In *ISSAC '22—Proceedings of the 2022 International Symposium on Symbolic and Algebraic Computation*, pages 255–262. ACM, New York, [2022] ©2022.

[BH16] Fernando G. S. L. Brandão and Aram W. Harrow. Product-State Approximations to Quantum States. *Communications in Mathematical Physics*, 342(1):47–80, 2016.

[BIQ+17] 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.

[BKS07] Piotr Berman, Marek Karpinski, and Alexander D. Scott. Computational complexity of some restricted instances of 3-sat. *Discrete Applied Mathematics*, 155(5):649–653, 2007.

[BMR09] S. Bravyi, C. Moore, and A. Russell. Bounds on the quantum satisfibility threshold. *Available at arXiv.org quant-ph/0907.1297v2*, 2009.

[Bra06] Sergey Bravyi. Efficient algorithm for a quantum analogue of 2-SAT, 2006.

[Can88] 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.

[CCD+11] 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.

[CDT09] 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.

[CLO05] David A. Cox, John Little, and Donal O'Shea. *Using Algebraic Geometry*. Graduate Texts in Mathematics. Springer-Verlag, 2005.

[CLO15] 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.

[CML23] Oliverio Cruz-Mejía and Adam N Letchford. A survey on exact algorithms for the maximum flow and minimum-cost flow problems. *Networks*, 2023.

[DGP06] 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.

[EH16] David Eisenbud and Joe Harris. *3264 and all that—a second course in algebraic geometry*. Cambridge University Press, Cambridge, 2016.

[ER08] Lior Eldar and Oded Regev. Quantum SAT for a Qutrit-Cinquit Pair Is QMA1-Complete. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 881–892, Berlin, Heidelberg, 2008. Springer.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[FGHS22] 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.

[FKK03] András Frank, Tamás Király, and Zoltán Király. On the orientation of graphs and hypergraphs: Submodularity. *Discrete Appl. Math.*, 131(2):385–400, 2003.

[Ful98] William Fulton. *Intersection theory*, volume 2 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer-Verlag, Berlin, second edition, 1998.

[Gal86] Zvi Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surv.*, 18(1):23–38, mar 1986.

[zGJ03] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 2003.

[GHJ+22] 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.

[GHLS15] 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.

[GK12] Sevag Gharibian and Julia Kempe. Approximation Algorithms for QMA-Complete Problems. *SIAM Journal on Computing*, 41(4):1028–1050, 2012.

[GKSZ20] 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.

[GN13] 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.

[Goe19] 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.

[Gre14] 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.

[HNN13] 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.

[JPY88] 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.

[JS17] 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.

[Juk11] Stasys Jukna. *Extremal Combinatorics: With Applications in Computer Science*. Texts in Theoretical Computer Science. An EATCS Series. Springer, Berlin, Heidelberg, 2011.

[KO05] Richard Kenyon and Andrei Okounjov. What is... a dimer? *Notices of the AMS*, 52(3), 2005.

[Kre24] Simon-Luca Kremer. Quantum k-SAT related hypergraph problems. Bachelor Thesis, unpublished, 2024.

[Lan05] 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.

[LLM+10] 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.

[LMRV24] Joon Lee, Nicolas Macris, Jean Bernoulli Ravelomanana, and Perrine Vantalon. The PRODSAT phase of random quantum satisfiability, 2024.

[LMSS10] Laumann, C.R., Moessner, R., Scardicchio, A., and Sondhi, S. L. Phase transitions and random quantum satisfiability. *Quantum Information & Computation*, 10:1–15, 2010.

[LPR24] 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.

[MM05] 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.

[MP91] Nimrod Megiddo and Christos H. Papadimitriou. On total functions, existence theorems and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.

[MS87] 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.

[Nag08] Daniel Nagaj. Local Hamiltonians in Quantum Computation, 2008.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[Pap94] Christos H. Papadimitriou. On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994.

[Par04] K. R. Parthasarathy. On the maximal dimension of a completely entangled subspace for finite level quantum systems. *Proceedings Mathematical Sciences*, 114(4):365–374, Nov 2004.

[Pla84] David A. Plaisted. New np-hard and np-complete polynomial and integer divisibility problems. *Theoretical Computer Science*, 31(1):125–138, 1984.

[Ren92] James Renegar. On the computational complexity of approximating solutions for real algebraic formulae. *SIAM Journal on Computing*, 21(6):1008–1025, 1992.

[RGN24] Dorian Rudolph, Sevag Gharibian, and Daniel Nagaj. Quantum 2-SAT on low dimensional systems is QMA1-complete: Direct embeddings and black-box simulation, 2024.

[Sch85] Arnold Schönhage. Equation solving in terms of computational complexity. In *International Congress of Mathematicians*, pages 131–153, 1985.

[Sha74] Igor R. Shafarevich. *Basic Algebraic Geometry.* Springer Berlin Heidelberg, 1974.

[SS18] Mohab Safey El Din and Éric Schost. Bit complexity for multi-homogeneous polynomial system solving—application to polynomial minimization. *Journal of Symbolic Computation*, 87:176–206, 2018.

[Wat18] John Watrous. *The Theory of Quantum Information.* Cambridge University Press, 2018.

A Proof of Hall's Marriage Theorem for weighted hypergraphs

The proof below is a simple adaptation to the weighted case of the proof found in [Juk11].

Theorem 23. *(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 it at least one edge of X. Then (G, w) has a WSDR if and only $|V_X|_w \geq |X|$ for every $X \subseteq E(G)$.*

Proof. Assume (G, w) has a WSDR $f : E(G) \to V(G)$. Since $f(e) \in e$ for every $e \in E(G)$, then $f(X) \subseteq V_X$ and thus $\sum_{v \in V_X} |f^{-1}(v)| = |X|$ for each $X \subseteq E(G)$. Hence

$$|V_X|_w = \sum_{v \in V_X} w(v) \geq \sum_{v \in V_X} |f^{-1}(v)| = |X| \,. \tag{57}$$

Conversely, assume $|V_X|_w \geq |X|$ for every $X \subseteq E(G)$. If G has a single edge e, by assumption that edge contains a vertex v such that $w(v) \geq 1$ and the assignment $e \mapsto v$ is the required WSDR. We now work by induction on the number of edges, and assume the statement is proved for all hypergraph with less than m edges. Let $E(G) = m$. We distinguish two cases.

Case 1. Suppose that $|V_X| > |X|$ whenever $|X| < m$. Pick $e \in E(G)$ and $v \in e$ such that $w(v) \geq 1$. Let (G', w') be the weighted hypergraph such that $V(G') = V(G)$, $E(G') = E(G) \setminus \{e\}$, $w'(z) = w(z)$ if $z \in V(G) \setminus \{v\}$ and $w'(v) = w(v) - 1$. Then for every $X' \subseteq E(G')$

$$|V_{X'}|_{w'} = \sum_{v \in V_{X'}} w'(v) \geq -1 + \sum_{v \in V_{X'}} w(v) > -1 + |X'| \,. \tag{58}$$

Since necessarily $|X'| < m$, by induction we have that (G', w') has a WSDR g. Let $f : E(G) \to V(G)$ such that $f(e') = g(e')$ for every $e' \in E(G')$ and $f(e) = v$. Then f is a WSDR for (G, w).

Case 2. Suppose there exists $X \subseteq E(G)$ such that $|V_X| = |X| < m$. By induction, the weighted hypergraph (G_1, w_1) such that $V(G_1) = V(G)$, $E(G_1) = X$ and $w_1 = w$ has a WSDR f_1. Consider the weighted hypergraph (G_2, w_2) such that $V(G_2) = V(G)$, $E(G_2) = E(G) \setminus X$, and $w_2(v) = w(v) - |f_1^{-1}(v)|$ for every $v \in V(G)$. Suppose (G_2, w_2) has no WSDR. By induction, there would exist $Y \subseteq E(G_2)$ such that $|V_Y| < |Y|$. Since $w(v) = w_2(v)$ for all $v \in V_Y \setminus V_X$, this would imply

$$|V_{X \cup Y}|_w = |V_X \cup V_Y|_w = \sum_{v \in V_X} w(x) + \sum_{v \in V_Y \setminus V_X} w(v) < |X| + |Y| \tag{59}$$

which contradicts the assumption. Hence (G_2, w_2) has a WSDR $f_2 : E(G_2) \to V(G)$. Let $f : E(G) \to V(G)$ be such that $f(e) = f_1(e)$ if $e \in X$ and $f(e) = f_2(e)$ otherwise. Then

$$|f^{-1}(v)| = |f_1^{-1}(v)| + |f_2^{-1}(v)| \leq |f_1^{-1}(v)| + w_2(v) = w(v) \tag{60}$$

for all $v \in V(G)$ and thus f is a WSDR for (G, w). □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il