nature portfolio

https://doi.org/10.1038/s41586-025-09527-5

Supplementary information

Optimization by decoded quantum interferometry

In the format provided by the authors and unedited

Supplementary Information for: Optimization by Decoded Quantum Interferometry

Stephen P. Jordan*[1], Noah Shutty†[1], Mary Wootters[2], Adam Zalcman[1],
Alexander Schmidhuber[1,3], Robbie King[1,4], Sergei V. Isakov[1],
Tanuj Khattar[1], and Ryan Babbush[1]

[1] *Google Quantum AI, Venice, CA 90291*
[2] *Departments of Computer Science and Electrical Engineering, Stanford University, Stanford, CA 94305*
[3] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139*
[4] *Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125*

Organization: In §1, we state Theorem 1.1, which characterizes the performance of DQI on max-LINSAT problems in terms of the ability to solve the corresponding decoding problem. In §2 we formally define the OPI problem and apply Theorem 1.1 to predict DQI's performance on OPI. In §3 we discuss how DQI performs on unstructured sparse instances of sparse max-XORSAT using belief propagation decoding. In §4 we discuss the long line of prior work related to DQI. In §5 we explain the DQI algorithm in detail. In §6 we prove Theorem 1.1. In §7 we state and prove 7.1, which is an analogue of Theorem 1.1 for the setting where $p = 2$ and ℓ exceeds half the distance of the code $C^\perp$. In §8 we discuss several existing classical and quantum optimization algorithms and compare their performance with DQI. We follow this in §9 by showing how we construct an instance for which DQI, using belief propagation decoding, can achieve an approximate optimum that is very difficult to replicate using simulated annealing. In §10 we derive information-theoretic upper bounds on the approximate optima achievable by DQI, which depend on whether one is considering classical or quantum decoders. In §11 we generalize the max-LINSAT problem and the DQI algorithm to folded codes and extension fields in order to shed some light on DQI's potential applicability to the problem considered by Yamakawa and Zhandry in [1]. In §12 we generalize the OPI problem to multivariate polynomials. Lastly, in §13 we obtain concrete resource requirements (qubits, Clifford gates, and non-Clifford gates) to apply DQI, using the Berlekamp Massey decoder, to the OPI problem.

1 Characterizing the Performance of DQI

DQI reduces the problem of approximating max-LINSAT to the problem of decoding the linear code $C^\perp$ over $\mathbb{F}_p$ whose parity check matrix is B^T. That is,

$$C^\perp = \{\mathbf{d} \in \mathbb{F}_p^m : B^T \mathbf{d} = \mathbf{0}\}. \tag{1}$$

*stephenjordan@google.com
†shutty@google.com

This decoding problem is to be solved in superposition, such as by a reversible implementation of any efficient classical decoding algorithm. If $C^\perp$ can be efficiently decoded out to ℓ errors then, given any appropriately normalized degree-ℓ polynomial P, DQI can efficiently produce the state

$$|P(f)\rangle = \sum_{\mathbf{x} \in \mathbb{F}_p^n} P(f(\mathbf{x})) \, |\mathbf{x}\rangle \,. \tag{2}$$

Upon measuring in the computational basis one obtains a given string $\mathbf{x}$ with probability $P(f(\mathbf{x}))^2$. One can choose P to bias this distribution toward strings of large objective value. Larger ℓ allows this bias to be stronger, but requires the solution of a harder decoding problem.

More quantitatively, in §6, we prove the following theorem.

Theorem 1.1. *Given a prime p and $B \in \mathbb{F}_p^{m \times n}$, let $f(\mathbf{x}) = \sum_{i=1}^m f_i(\sum_{j=1}^n B_{ij} x_j)$ be a max-LINSAT objective function. Suppose $|f_i^{-1}(+1)| = r$ for all $i = 1, \ldots, m$ and some $r \in \{1, \ldots, p-1\}$. Given a degree-ℓ polynomial P, let $\langle s \rangle$ be the expected number of satisfied constraints for the symbol string obtained upon measuring the corresponding DQI state $|P(f)\rangle$ in the computational basis. Suppose $2\ell + 1 < d^\perp$ where $d^\perp$ is the minimum distance of the code $C^\perp = \{\mathbf{d} \in \mathbb{F}_p^m : B^T \mathbf{d} = \mathbf{0}\}$, i.e. the minimum Hamming weight of any nonzero codeword in $C^\perp$. In the limit $m \to \infty$, with ℓ/m fixed, the optimal choice of degree-ℓ polynomial P to maximize $\langle s \rangle$ yields*

$$\frac{\langle s \rangle}{m} = \left(\sqrt{\frac{\ell}{m}\left(1 - \frac{r}{p}\right)} + \sqrt{\frac{r}{p}\left(1 - \frac{\ell}{m}\right)} \right)^2 \tag{3}$$

if $\frac{r}{p} \leq 1 - \frac{\ell}{m}$ and $\frac{\langle s \rangle}{m} = 1$ otherwise.

Theorem 1.1 assumes that $2\ell + 1 < d^\perp$, which is the same as requiring that $C^\perp$ can be in principle decoded from up to ℓ worst-case errors. Further, if this decoding can be done *efficiently*, then the DQI algorithm is also efficient. In our analysis, we show how to relax these assumptions. In particular, in Theorem 7.1 we show that even when $2\ell + 1 \geq d^\perp$ and it is not possible to decode ℓ worst-case errors, an efficient algorithm that succeeds with high probability over random errors can be used in the DQI algorithm to efficiently achieve a fraction of satisfied constraints close to the one given by (3), at least for max-XORSAT problems $B\mathbf{x} \overset{\max}{=} \mathbf{v}$ with average-case $\mathbf{v}$.

For the balanced case $r \to p/2$, (3) simplifies to

$$\frac{\langle s \rangle}{m} = \frac{1}{2} + \sqrt{\frac{\ell}{m}\left(1 - \frac{\ell}{m}\right)}, \tag{4}$$

i.e. the equation of a semicircle.

DQI reduces the problem of satisfying a large number of linear constraints to the problem of correcting a large number of errors in a linear code. Decoding linear codes is also an NP-hard problem in general [2]. So, one must ask whether this reduction is ever advantageous. We next present evidence that it can be.

2 Optimal Polynomial Intersection

The problem which provides our clearest demonstration of the power of DQI is the OPI problem, as specified in Definition 2.2 of the main text. In this section, we explain how to apply DQI to OPI, and identify a parameter regime for OPI where DQI outperforms all classical algorithms known to us.

We first observe that OPI is equivalent to a special case of max-LINSAT. Let $q_0, \ldots, q_{n-1} \in \mathbb{F}_p$ be the coefficients in Q:

$$Q(y) = \sum_{j=0}^{n-1} q_j \, y^j. \tag{5}$$

Recall that a primitive element of a finite field is an element such that taking successive powers of it yields all nonzero elements of the field. Every finite field contains one or more primitive elements. Thus, we can choose γ to be any primitive element of $\mathbb{F}_p$ and re-express the OPI objective function as

$$f_{\mathrm{OPI}}(Q) = |\{i \in \{0, 1, \ldots, p-2\} : Q(\gamma^i) \in F_{\gamma^i}\}|. \tag{6}$$

Next, let

$$f_i(x) = \begin{cases} +1 & \text{if } x \in F_{\gamma^i} \\ -1 & \text{otherwise} \end{cases} \tag{7}$$

for $i = 0, \ldots, p-2$ and define the matrix B by

$$B_{ij} = \gamma^{i \times j} \quad i = 0, \ldots, p-2 \qquad j = 0, \ldots, n-1. \tag{8}$$

Then the max-LINSAT objective function is $f(\mathbf{q}) = \sum_{i=0}^{p-2} f_i(\mathbf{b}_i \cdot \mathbf{q})$ where $\mathbf{q} = (q_0, \ldots, q_{n-1})^T \in \mathbb{F}_p^n$ and $\mathbf{b}_i$ is the i^{th} row of B. But $\mathbf{b}_i \cdot \mathbf{q} = Q(\gamma^i)$, so $f(\mathbf{q}) = 2 \cdot f_{\mathrm{OPI}}(Q) - (p-1)$ which means that the max-LINSAT objective function f and the OPI objective function f_{OPI} are equivalent. Thus we have re-expressed our OPI instance as an equivalent instance of max-LINSAT with $m = p-1$ constraints.

We will apply DQI to the case where

$$|f_i^{-1}(+1)| = \lfloor p/2 \rfloor \quad \forall i = 0, \ldots, p-2. \tag{9}$$

By (9), in the limit of large p we have $|f_i^{-1}(+1)|/p \to 1/2$ and $|f_i^{-1}(-1)|/p \to 1/2$ for all i. We call functions with this property "balanced."

When B has the form (8), then $C^\perp = \{\mathbf{d} \in \mathbb{F}_p^{p-1} : B^T \mathbf{d} = 0\}$ is a Reed-Solomon code with alphabet $\mathbb{F}_p$, block length $p-1$, dimension $p-n-1$, and distance $n+1$. Note that our definition of n is inherited from the parameters of the max-LINSAT instances that we start with and hence our notations for block length and dimension unfortunately do not conform to standard notations from coding theory.

Maximum likelihood syndrome decoding for Reed-Solomon codes can be solved in polynomial time out to half the distance of the code, *e.g.* using the Berlekamp-Massey algorithm [3]. Consequently, in DQI we can take $\ell = \lfloor \frac{n+1}{2} \rfloor$. In (4) we can thus set the number of errors corrected $\ell \to \frac{n}{2}$ and the number of constraints $m \to p$, which shows that the asymptotic performance of DQI using Berlekamp-Massey is

$$\frac{\langle s \rangle_{\mathrm{DQI+BM}}}{p} = \frac{1}{2} + \sqrt{\frac{n}{2p}\left(1 - \frac{n}{2p}\right)}. \tag{10}$$

Here we have approximated $n+1$ by n and $p-1$ by p since this is an asymptotic formula anyway. For exact expressions at finite size see §6. The largest asymptotic fraction of satisfied clauses for

OPI that we know how to obtain classically in polynomial time is

$$\frac{\langle s \rangle_{\text{Prange}}}{m} = \frac{1}{2} + \frac{n}{2p}, \tag{11}$$

which is achieved by Prange's algorithm. These are plotted in Fig. 3 of the main text, where one sees that DQI+BP exceeds Prange's algorithm for all $n/p \in (0, 1)$. (See §8.3 for a description of Prange's algorithm.)

Therefore, the Optimal Polynomial Intersection problem demonstrates the power of DQI. Assuming no polynomial-time classical algorithm for this problem is found that can match this fraction of satisfied constraints, this constitutes an example of an superpolynomial quantum speedup. It is noteworthy that our quantum algorithm is not based on a reduction to an Abelian Hidden Subgroup or Hidden Shift problem. The margin of victory for the approximation fraction (0.7179 vs. 0.55) is also satisfyingly large. Nevertheless, it is also of great interest to investigate whether such a quantum speedup can be obtained for more generic constraint satisfaction problems, with less underlying structure, as we do in the next section.

Before moving on to unstructured optimization problems, we make two remarks.

Remark 2.1 (Relationship to the work of Yamakawa and Zhandry)**.** First, we note that the algorithm of Yamakawa and Zhandry [1]—which solves a version of OPI—does not apply in our setting. As discussed in §8.6, the parameters of our OPI problem are such that solutions satisfying all constraints are statistically likely to exist but these exact optima seem to be computationally intractable to find using known classical algorithms. The quantum algorithm of Yamakawa and Zhandry, when it can be used, produces a solution satisfying all constraints. However, the quantum algorithm of Yamakawa and Zhandry has high requirements on the decodability of $C^\perp$. Specifically, for the "balanced case" in which $|f_i(+1)| \simeq |f_i(-1)|$ for all i, the requirement is that $C^\perp$ can be decoded from a 1/2 fraction of random errors. For our OPI example, $C^\perp$ has rate 9/10. Shannon's noisy-channel coding theorem implies that it is not possible to reliably decode $C^\perp$ in this setting. Thus, the quantum algorithm of Yamakawa and Zhandry is not applicable.

Remark 2.2 (Classical Complexity of OPI)**.** Proving rigorous classical hardness guarantees for OPI seems like a challenging problem. OPI, and OPI-like problems, have been proposed as cryptographically hard problems. As discussed in §8.7, a version of OPI in a different parameter regime was proposed as a hardness assumption for cryptographic applications by [4]. This conjecture was broken by [5] using lattice attacks, but we demonstrate in §8.7 that these attacks do not apply in our parameter regime. Later work [6] proposed two updated hardness assumptions, which each would imply the hardness of a special case of OPI.[1] These assumptions have yet to be broken to the best of our knowledge, and DQI does not seem to be an effective attack on them in the parameter regimes of interest.

There are other problems related to OPI that are known to be computationally hard, under standard assumptions. For example, the problem of maximum-likelihood decoding for Reed-Solomon Codes—which is the case of OPI when $|f_i^{-1}(+1)| = 1$ for all i—is known to be NP-hard [7, 8]. List-decoding and bounded-distance decoding for Reed-Solomon codes to a large enough radius—also

[1] In more detail, the first problem assumed to be hard is related to bounded distance decoding for Reed-Solomon codes from random errors, which corresponds to OPI when $|f_i^{-1}(+1)| = 1$. The second problem can be viewed as a generalization of OPI to *randomly folded* Reed-Solomon codes with $|f_i^{-1}(+1)|$ larger than 1; we show in Appendix 11 that DQI can apply to folded codes, but it does not yield useful attacks in the relevant parameter regime.

related to OPI when $|f_i^{-1}(+1)| = 1$—is known to be as hard as discrete log [9, 10]. Theorem 1.1 does not provide strong performance guarantees for DQI applied to these problems. It would be very interesting to show that OPI (in a parameter regime $|f_i^{-1}(+1)| \propto p$ where Theorem 1.1 does give strong performance guarantees) is classically hard under standard cryptographic assumptions.

3 Random Sparse max-XORSAT

In this section, we consider average-case instances from certain families of bounded degree max-k-XORSAT. In a max-k-XORSAT instance with degree bounded by D, each constraint contains at most k variables and each variable is contained in at most D constraints. In other words, the matrix $B \in \mathbb{F}_2^{m \times n}$ defining the instance has at most k nonzero entries in any row and at most D nonzero entries in any column. DQI reduces this to decoding the code $C^\perp$ whose parity check matrix is B^T. Codes with sparse parity check matrices are known as Low Density Parity Check (LDPC) codes. Randomly sampled LDPC codes are known to be correctable from a near-optimal number of random errors (asymptotically as m grows) [11]. Consequently, in the limit of large m they can in principle be decoded up to a number of random errors that nearly saturates the information-theoretic limit dictated by the rate of the code. When k and D are very small, information-theoretically optimal decoding for random errors can be closely approached by polynomial-time decoders such as belief propagation [12, 13]. This makes sparse max-XORSAT a promising target for DQI.

In this section we focus on benchmarking DQI with standard belief propagation decoding (DQI+BP) against simulated annealing on max-k-XORSAT instances. We choose simulated annealing as our primary classical point of comparison because it is often very effective in practice on sparse constraint satisfaction problems and also because it serves as a representative example of local search heuristics. Local search heuristics are widely used in practice and include greedy algorithms, parallel tempering, TABU search, and many quantum-inspired optimization methods. As discussed in §8.1, these should all be expected to have similar scaling behavior with D on average-case max-k-XORSAT with bounded degree. Because of simulated annealing's simplicity, representativeness, and strong performance on average-case constraint satisfaction problems, beating simulated annealing on some class of instances is a good first test for any new classical or quantum optimization heuristic.

It is well-known that max-XORSAT instances become harder to approximate as the degree of the variables is increased [14, 15]. Via DQI, a max-XORSAT instance of degree D is reduced to a problem of decoding random errors for a code in which each parity check contains at most D variables. As D increases, with m/n held fixed, the distance of the code and hence its performance under information-theoretically optimal decoding are not degraded at all. Thus, as D grows, the fraction of constraints satisfied by DQI with information-theoretically optimal decoding would not degrade. In contrast, classical optimization algorithms based on local search yield satisfaction fractions converging toward $1/2$ in the limit $D \to \infty$ which is no better than random guessing. Thus as D grows with k/D fixed, DQI with information-theoretically optimal decoding will eventually surpass all classical local search heuristics. (See also Fig. 10.) However, for most ensembles of codes, the number of errors correctable by standard polynomial-time decoders such as belief propagation falls increasingly short of information-theoretic limits as the degree D of the parity checks increases. Thus increasing D generically makes the problem harder both for DQI+BP and for classical optimization heuristics.

Despite this challenge, we are able to find some unstructured families of sparse max-XORSAT instances for which DQI with standard belief propagation decoding finds solutions satisfying a fraction of constraints that is very difficult to replicate using simulated annealing. We do so by tuning the degree distribution of the instances. For example, in §9, we generate an example max-XORSAT instance from our specified degree distribution, which has $31,216$ variables and $50,000$ constraints, where each constraint contains an average of 53.973 variables and each variable is contained in an average of 86.451 constraints. We find that DQI with standard belief propagation decoding can find solutions in which the fraction of constraints satisfied is at least 0.831. It does so by reducing this problem to a decoding problem that can be solved by our implementation of belief propagation in 8 seconds, excluding the time needed to load and parse the instance. In contrast, our implementation of simulated annealing requires approximately 73 hours to reach this, even when allowed five cores in parallel; restricted to 8 seconds of runtime on a single core it is only able to satisfy 0.764.

Furthermore, as shown in Table 1 of the main text, DQI+BP achieves higher satisfaction fraction than we are able to obtain in a comparable number of computational steps using any of the general-purpose classical optimization algorithms that we tried. However, unlike our OPI example, we do not put this forth as an example of quantum advantage. Rather, we are able to construct a tailored classical algorithm specialized to these instances which, within seven minutes of runtime, finds solutions where the fraction of constraints satisfied is 0.88, thereby slightly beating DQI+BP.

4 Relation to Other Work

DQI is related to a family of quantum reductions that originate with the work of Aharonov, Ta-Shma, and Regev [16, 17]. In this body of work the core idea is to use the Fourier convolution theorem to obtain reductions between nearest lattice vector problems and shortest lattice vector problems. In this section we summarize the other quantum algorithms using this idea and discuss their relationship to DQI.

In [18], Chen, Liu, and Zhandry introduce a novel and powerful intrinsically-quantum decoding method that they call filtering, which can in some cases solve quantum decoding problems for which the analogous classical decoding problems cannot be solved by any known efficient classical algorithm. By combining this decoding method with a Regev-style reduction, they are able to efficiently find approximate optima to certain shortest vector problems defined using the infinity norm for which no polynomial-time classical solution is known. Due to the use of the infinity norm, the problem solved by the quantum algorithm of [18] is one of satisfying all constraints, rather than the more general problem of maximizing the number of constraints satisfied, as is addressed by DQI. Although conceptualized differently, and implemented over the space of codewords rather than the space of syndromes, the algorithm of [18] is similar in spirit to DQI and can be regarded as a foundational prior work. It is particularly noteworthy that the apparent superpolynomial quantum speedup achieved in [18], through the use of quantum decoders, is obtained for a purely geometrical problem with no algebraic structure.

In [1], Yamakawa and Zhandry define an oracle problem that they prove can be solved using polynomially many quantum queries but requires exponentially many classical queries. Their problem is essentially a class of instances of max-LINSAT over an exponentially large finite field $\mathbb{F}_{2^t}$, where the functions $f_1, \ldots, f_m$ are defined by random oracles and the matrix B has algebraic structure. In §11 we recount the exact definition of the Yamakawa-Zhandry problem and show how DQI can be extended to the Yamakawa-Zhandry problem. In the problem defined by Yamakawa and Zhandry, the truth tables for the constraints are defined by random oracles, and therefore, in the language of Theorem 1.1, $r/p = 1/2$. Furthermore, the Yamakawa-Zhandry instance is designed such that $C^{\perp}$ can be decoded with exponentially small failure probability if half of the symbols are corrupted by errors. Thus, in the language of Theorem 1.1, we can take $\ell/m = 1/2$. In this case, if we extrapolate (3) to the Yamakawa-Zhandry regime, one obtains $\langle s \rangle/m = 1$, indicating that DQI should find a solution satisfying all constraints. The quantum algorithm given by Yamakawa and Zhandry for finding a solution satisfying all constraints is different from DQI, but similar in spirit.

Extrapolation of the semicircle law is necessary because the Yamakawa-Zhandry example does not satisfy $2\ell + 1 < d^{\perp}$, and therefore the conditions of Theorem 1.1 are not met. In Theorem 7.1 we prove a variant of the semicircle law for $2\ell + 1 > d^{\perp}$ for max-XORSAT with average case $\mathbf{v}$. The Yamakawa-Zhandry problem uses a random oracle, which is the direct generalization to $\mathbb{F}_q$ of average case $\mathbf{v}$. It seems likely that the proof of Theorem 7.1 can be generalized to from $\mathbb{F}_2$ to $\mathbb{F}_q$ with arbitrary q at the cost of additional technical complications but without the need for conceptual novelty. In this case the claim that DQI encompasses the Yamakawa-Zhandry upper bound on quantum query complexity could be made rigorous.

An important difference between our OPI results and the exponential quantum query complexity speedup of Yamakawa and Zhandry is that the latter depends on $\mathbb{F}_q$ being an exponentially large finite field. This allows Yamakawa and Zhandry to obtain an information-theoretic exponential classical query-complexity lower bound, thus making their separation rigorous. In contrast, our apparent superpolynomial speedup for OPI uses a finite field of polynomial size, in which case the truth tables are polynomial size and known explicitly. This regime is more relevant to real-world optimization problems. The price we pay is that the classical query complexity in this regime is necessarily polynomial. This means that the information-theoretic argument of [1] establishing an exponential improvement in query complexity does not apply in our setting. Instead, we argue heuristically for an superpolynomial quantum speedup, by comparing to known classical algorithms.

In [19], Debris-Alazard, Remaud, and Tillich construct a quantum reduction from the approximate shortest codeword problem on a code C to the bounded distance decoding problem on its dual $C^{\perp}$. The shortest codeword problem is closely related to max-XORSAT: the max-XORSAT problem is to find the codeword in $C = \{B\mathbf{x} \,|\, \mathbf{x} \in \mathbb{F}_2^n\}$ that has smallest Hamming distance from a given bit string $\mathbf{v}$, whereas the shortest codeword problem is to find the codeword in $C \setminus \{\mathbf{0}\}$ with smallest Hamming weight. Roughly speaking, then, the shortest codeword problem is the special case of max-XORSAT where $\mathbf{v} = \mathbf{0}$ except that the trivial solution $\mathbf{0}$ is excluded. Although conceptualized quite differently, the reduction of [19] is similar to the $p = 2$ special case of DQI in that it is achieved via a quantum Hadamard transform.

Another interesting connection between DQI and prior work appears in the context of planted inference problems, such as planted kXOR, where the task is to recover a secret good assignment planted in an otherwise random kXOR instance. It has been recently shown that quantum algorithms can achieve polynomial speedups for planted kXOR by efficiently constructing a *guiding state* that has improved overlap with the planted solution [20]. Curiously, the ℓ^{th} order guiding state studied in [20] seems related to the ℓ^{th} order DQI state presented here. The key conceptual difference is that, to obtain an assignment that satisfies a large number of constraints, the DQI state in this work is measured in the computational basis, whereas the guiding state in [20] is measured in the eigenbasis of the so-called Kikuchi matrix, and subsequently rounded. It is interesting that for the large values of ℓ studied in this work, the (Fourier-transformed) Kikuchi matrix asymptotically approaches a diagonal matrix such that its eigenbasis is close to the computational basis.

Next, we would like to highlight [21] in which Chailloux and Tillich introduce a method for decoding superpositions of errors, which they use in the context of a Regev-style reduction. Their method to solve the quantum decoding problem is based on a technique called *unambiguous state discrimination* (USD). Roughly speaking, USD is a coherent quantum rotation which allows us to convert bit-flip error into erasure error. If the resulting erasure error rate is small enough, one can decode perfectly using Gaussian elimination. Interestingly, this is efficient regardless of the sparsity of the code. Curiously, however, Chailloux and Tillich find that the resulting performance of this combination of quantum algorithms yields performance for the shortest codeword problem that exactly matches that of Prange's algorithm [22].

Lastly, we note that since we posted the first version of this manuscript as an arXiv preprint in August, 2024 there have already been some works that nicely build upon DQI. In particular, we would like to highlight [23] in which Chailloux and Tillich observe that for $p > 2$ the distribution over errors on a given corrupted symbol in the decoding problem faced by DQI is non-uniform. Information-theoretically this is more advantageous than uniform errors. Furthermore, they show that this advantage can be exploited by polynomial-time decoders, thereby improving the approximation ratio efficiently achievable by DQI. Additionally, in [24], a concrete resource analysis is carried out for DQI in which the decoding of $C^\perp$ is carried out using a reversible circuit implementing information set decoding.

5 Decoded Quantum Interferometry

Here we describe in full detail the Decoded Quantum Interferometry algorithm, illustrated in Fig. 1. We start with DQI for max-XORSAT, which is the special case of max-LINSAT with $\mathbb{F} = \mathbb{F}_2$. Then we describe DQI with $\mathbb{F} = \mathbb{F}_p$ for any prime p. In each case, our explanation consists of a discussion of the quantum state we intend to create followed by a description of the quantum algorithm used to create it. The generalization of DQI to extension fields is given in §11.

In this section, we assume throughout that $2\ell + 1 < d^\perp$. In some cases we find that it is possible to decode more than $d^\perp/2$ errors with high probability, in which case we can use ℓ exceeding this bound. This contributes some technical complications to the description and analysis of the DQI algorithm, which we defer to §7.

5.1 DQI for max-XORSAT

5.1.1 DQI Quantum State for max-XORSAT

Recall that the objective function for max-XORSAT can be written in the form $f = f_1 + \ldots + f_m$, where f_i takes the value $+1$ if the i^{th} constraint is satisfied and -1 otherwise. More concretely, the function defined by

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{m} f_i(\mathbf{b}_i \cdot \mathbf{x}) && (12) \\ f_i(\mathbf{b}_i \cdot \mathbf{x}) &= (-1)^{v_i}(-1)^{\mathbf{b}_i \cdot \mathbf{x}}, && (13) \end{aligned}$$

expresses the number of constraints satisfied minus the the number of constraints unsatisfied by the bit string $\mathbf{x}$.

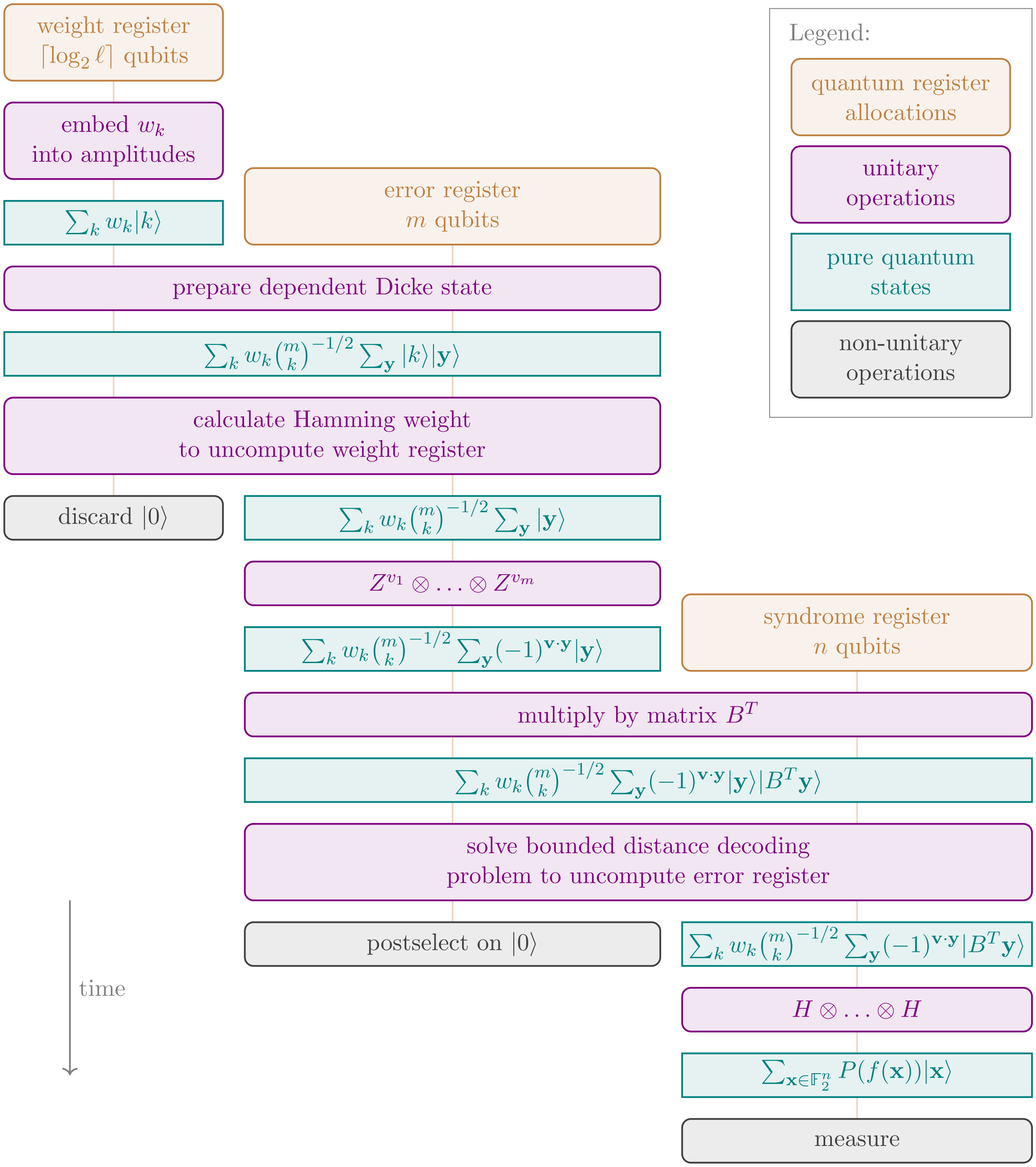
weight register $\lceil \log_2 \ell \rceil$ qubits
embed w_k into amplitudes
$\sum_k w_k |k\rangle$
error register m qubits
prepare dependent Dicke state
$\sum_k w_k \binom{m}{k}^{-1/2} \sum_{\mathbf{y}} |k\rangle |\mathbf{y}\rangle$
calculate Hamming weight to uncompute weight register
discard $|0\rangle$
$\sum_k w_k \binom{m}{k}^{-1/2} \sum_{\mathbf{y}} |\mathbf{y}\rangle$
$Z^{v_1} \otimes \ldots \otimes Z^{v_m}$
$\sum_k w_k \binom{m}{k}^{-1/2} \sum_{\mathbf{y}} (-1)^{\mathbf{v}\cdot\mathbf{y}} |\mathbf{y}\rangle$
syndrome register n qubits
multiply by matrix B^T
$\sum_k w_k \binom{m}{k}^{-1/2} \sum_{\mathbf{y}} (-1)^{\mathbf{v}\cdot\mathbf{y}} |\mathbf{y}\rangle |B^T \mathbf{y}\rangle$
solve bounded distance decoding problem to uncompute error register
postselect on $|0\rangle$
$\sum_k w_k \binom{m}{k}^{-1/2} \sum_{\mathbf{y}} (-1)^{\mathbf{v}\cdot\mathbf{y}} |B^T \mathbf{y}\rangle$
$H \otimes \ldots \otimes H$
$\sum_{\mathbf{x}\in\mathbb{F}_2^n} P(f(\mathbf{x})) |\mathbf{x}\rangle$
measure
time
Legend:
quantum register allocations
unitary operations
pure quantum states
non-unitary operations

Figure 1: Decoded Quantum Interferometry over $\mathbb{F}_2$. The algorithm begins with a computation, on a classical computer, of the principal eigenvector $(w_0, \ldots, w_\ell)^T$ of a certain matrix $A^{(m,\ell,0)}$. $P(f)$ is a degree-ℓ polynomial that enhances the probability of sampling $\mathbf{x} \in \mathbb{F}_2^n$ with high value of the objective f. Index k ranges over $\{0, \ldots, \ell\}$ and $\mathbf{y}$ over the set $\{\mathbf{y} \in \mathbb{F}_2^m : |\mathbf{y}| = k\}$ of m-bit strings of Hamming weight k. Weight register qubits may be reused for the syndrome register. If $2\ell + 1 < d^\perp$, the postselection succeeds with probability $\geq 1 - \varepsilon_\ell$, where ε_ℓ is the decoding failure rate on random weight-ℓ errors.

Given an objective of the form $f = f_1 + \ldots + f_m$ and a degree-ℓ univariate polynomial

$$P(f) = \sum_{k=0}^{\ell} \alpha_k f^k, \tag{14}$$

we can regard $P(f)$ as a degree-ℓ multivariate polynomial in $f_1, \ldots, f_m$. Since $f_1, \ldots, f_m$ are ± 1-valued we have $f_i^2 = 1$ for all i. Consequently, we can express $P(f)$ as

$$P(f) = \sum_{k=0}^{\ell} u_k P^{(k)}(f_1, \ldots, f_m), \tag{15}$$

where $P^{(k)}(f_1, \ldots, f_m)$ is the degree-k elementary symmetric polynomial, *i.e.* the unweighted sum of all $\binom{m}{k}$ products of k distinct factors from $f_1, \ldots, f_m$.

For simplicity, we will henceforth always assume that the overall normalization of P has been chosen so that the state $|P(f)\rangle = \sum_{\mathbf{x} \in \mathbb{F}_2^n} P(f(\mathbf{x})) |\mathbf{x}\rangle$ has unit norm. In analogy with equation (15), we will write the DQI state $|P(f)\rangle$ as a linear combination of $|P^{(0)}\rangle, \ldots, |P^{(\ell)}\rangle$, where

$$|P^{(k)}\rangle := \frac{1}{\sqrt{2^n \binom{m}{k}}} \sum_{\mathbf{x} \in \mathbb{F}_2^n} P^{(k)}(f_1(\mathbf{b}_1 \cdot \mathbf{x}), \ldots, f_m(\mathbf{b}_m \cdot \mathbf{x})) |\mathbf{x}\rangle. \tag{16}$$

Thus, by (13),

$$\begin{aligned} P^{(k)}(f_1, \ldots, f_m) &= \sum_{\substack{i_1, \ldots, i_k \\ \text{distinct}}} f_{i_1} \times \ldots \times f_{i_k} & (17) \\ &= \sum_{\substack{i_1, \ldots, i_k \\ \text{distinct}}} (-1)^{v_{i_1} + \ldots + v_{i_k}} (-1)^{(\mathbf{b}_{i_1} + \ldots + \mathbf{b}_{i_k}) \cdot \mathbf{x}} & (18) \\ &= \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}| = k}} (-1)^{\mathbf{v} \cdot \mathbf{y}} (-1)^{(B^T \mathbf{y}) \cdot \mathbf{x}}, & (19) \end{aligned}$$

where $|\mathbf{y}|$ indicates the Hamming weight of $\mathbf{y}$. From this we see that the Hadamard transform of $|P^{(k)}\rangle$ is

$$|\widetilde{P}^{(k)}\rangle := H^{\otimes n} |P^{(k)}\rangle = \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}| = k}} (-1)^{\mathbf{v} \cdot \mathbf{y}} |B^T \mathbf{y}\rangle. \tag{20}$$

For the set of $\mathbf{y} \in \mathbb{F}_2^n$ with $|\mathbf{y}| < d^\perp/2$ the corresponding bit strings $B^T \mathbf{y}$ are all distinct. Therefore, $|\widetilde{P}^{(0)}\rangle, \ldots, |\widetilde{P}^{(\ell)}\rangle$ form an orthonormal set provided $\ell < d^\perp/2$, and so do $|P^{(0)}\rangle, \ldots, |P^{(\ell)}\rangle$. In this case, the DQI state $|P(f)\rangle = \sum_{\mathbf{x} \in \mathbb{F}_2^n} P(f(\mathbf{x})) |\mathbf{x}\rangle$ can be expressed as

$$|P(f)\rangle = \sum_{k=0}^{\ell} w_k |P^{(k)}\rangle, \tag{21}$$

where

$$w_k = u_k \sqrt{2^n \binom{m}{k}}, \tag{22}$$

and $\langle P(f) | P(f) \rangle = \|\mathbf{w}\|^2$ where $\mathbf{w} := (w_0, \ldots, w_\ell)^T$.

5.1.2 DQI Algorithm for max-XORSAT

With these facts in hand, we now present the DQI algorithm for max-XORSAT. The algorithm utilizes three quantum registers: a *weight register* comprising $\lceil \log_2 \ell \rceil$ qubits, an *error register* with m qubits, and a *syndrome register* with n qubits.

As the first step in DQI we initialize the weight register in the state

$$\sum_{k=0}^{\ell} w_k \left| k \right\rangle . \tag{23}$$

The choice of $\mathbf{w} \in \mathbb{R}^{\ell+1}$ that maximizes the expected number of satisfied constraints can be obtained by solving for the principal eigenvector of an $(\ell+1) \times (\ell+1)$ matrix, as described in §6. Given $\mathbf{w}$, this state preparation can be done efficiently because it is a state of only $\lceil \log_2 \ell \rceil$ qubits. One way to do this is to use the method from [25] that prepares an arbitrary superposition over ℓ computational basis states using $\widetilde{O}(\ell)$ quantum gates.

Next, conditioned on the value k in the weight register, we prepare the error register into the uniform superposition over all bit strings of Hamming weight k

$$\to \sum_{k=0}^{\ell} w_k \left| k \right\rangle \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}|=k}} \left| \mathbf{y} \right\rangle . \tag{24}$$

Efficient methods for preparing such superpositions using $O(\ell m)$ quantum gates have been devised due to applications in physics, where they are known as Dicke states [26, 27]. Next, we uncompute $\left| k \right\rangle$ which can be done easily since k is simply the Hamming weight of $\mathbf{y}$. After doing so and discarding the weight register, we are left with the state

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}|=k}} \left| \mathbf{y} \right\rangle . \tag{25}$$

Next, we apply the phases $(-1)^{\mathbf{v} \cdot \mathbf{y}}$ by performing a Pauli-Z_i on each qubit for which $v_i = 1$, at the cost of $O(m)$ quantum gates

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{|\mathbf{y}|=k} (-1)^{\mathbf{v} \cdot \mathbf{y}} \left| \mathbf{y} \right\rangle . \tag{26}$$

Then, we reversibly compute $B^T \mathbf{y}$ into the syndrome register by standard matrix-vector multiplication over $\mathbb{F}_2$ at the cost $O(mn)$ quantum gates

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{|\mathbf{y}|=k} (-1)^{\mathbf{v} \cdot \mathbf{y}} \left| \mathbf{y} \right\rangle \left| B^T \mathbf{y} \right\rangle . \tag{27}$$

From this, to obtain $|\widetilde{P}^{(k)}\rangle$ we need to use the content $\mathbf{s} = B^T \mathbf{y}$ of the syndrome register to reversibly uncompute the content $\mathbf{y}$ of the error register, in superposition. This is the hardest part. We next discuss how to compute $\mathbf{y}$ from $\mathbf{s}$.

Recall that $B \in \mathbb{F}_2^{m \times n}$ with $m > n$. Thus, given $\mathbf{s}$, solving $\mathbf{s} = B^T \mathbf{y}$, for $\mathbf{y}$ is an underdetermined linear algebra problem over $\mathbb{F}_2$. It is only the constraint $|\mathbf{y}| \leq \ell$ that renders the solution unique, provided ℓ is not too large. This linear algebra problem with a Hamming weight constraint is recognizable as syndrome decoding. The kernel of B^T defines an error correcting code $C^\perp$. The string $\mathbf{s}$ is interpreted as the error syndrome, and $\mathbf{y}$ is interpreted as the error. If ℓ is less than half the minimum distance of $C^\perp$ then the problem of inferring the error from the syndrome has a unique solution. When this solution can be found efficiently, we can efficiently uncompute the content of the error register, which can then be discarded. This leaves

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{|\mathbf{y}|=k} (-1)^{\mathbf{v} \cdot \mathbf{y}} |B^T \mathbf{y}\rangle \tag{28}$$

in the syndrome register which we recognize as

$$= \sum_{k=0}^{\ell} w_k |\widetilde{P}^{(k)}\rangle . \tag{29}$$

By taking the Hadamard transform we then obtain

$$\to \sum_{k=0}^{\ell} w_k |P^{(k)}\rangle \tag{30}$$

which is the desired DQI state $|P(f)\rangle$.

5.2 DQI for General max-LINSAT

5.2.1 DQI Quantum State for General max-LINSAT

Recall that the max-LINSAT objective takes the form $f(\mathbf{x}) = \sum_{i=1}^m f_i(\mathbf{b}_i \cdot \mathbf{x})$ with $f_i : \mathbb{F}_p \to \{+1, -1\}$. In order to keep the presentation as simple as possible, we restrict attention to the situation where the preimages $F_i := f_i^{-1}(+1)$ for $i = 1, \ldots, m$ have the same cardinality $r := |F_i| \in \{1, \ldots, p-1\}$. We will find it convenient to work in terms of g_i which we define as f_i shifted and rescaled so that its Fourier transform

$$\tilde{g}_i(y) = \frac{1}{\sqrt{p}} \sum_{x \in \mathbb{F}_p} \omega_p^{yx} g_i(x) \tag{31}$$

vanishes at $y = 0$ and is normalized, *i.e.* $\sum_{x \in \mathbb{F}_p} |g_i(x)|^2 = \sum_{y \in \mathbb{F}_p} |\tilde{g}_i(y)|^2 = 1$. (Here and throughout, $\omega_p = e^{i2\pi/p}$.) In other words, rather than using f_i directly, we will use

$$g_i(x) := \frac{f_i(x) - \overline{f}}{\varphi} \tag{32}$$

where $\overline{f} := \frac{1}{p} \sum_{x \in \mathbb{F}_p} f_i(x)$ and $\varphi := \left(\sum_{y \in \mathbb{F}_p} |f_i(y) - \overline{f}|^2 \right)^{1/2}$. Explicitly, one finds

$$\bar{f} \;=\; \frac{2r}{p} - 1 \tag{33}$$

$$\varphi \;=\; \sqrt{4r\left(1 - \frac{r}{p}\right)}. \tag{34}$$

By (32), the sums

$$\begin{aligned} f(\mathbf{x}) &= f_1(\mathbf{b}_1 \cdot \mathbf{x}) + \ldots + f_m(\mathbf{b}_m \cdot \mathbf{x}) && (35) \\ g(\mathbf{x}) &= g_1(\mathbf{b}_1 \cdot \mathbf{x}) + \ldots + g_m(\mathbf{b}_m \cdot \mathbf{x}) && (36) \end{aligned}$$

are related according to

$$f(\mathbf{x}) = g(\mathbf{x})\varphi + m\overline{f}. \tag{37}$$

We transform the polynomial $P(f(\mathbf{x}))$ into an equivalent polynomial $Q(g(\mathbf{x}))$ of the same degree by substituting in this relation and absorbing the relevant powers of φ and $m\overline{f}$ into the coefficients. That is, $Q(g(\mathbf{x})) = P(f(\mathbf{x}))$. As shown in Appendix A, $Q(g(\mathbf{x}))$ can always be expressed as a linear combination of elementary symmetric polynomials. That is,

$$Q(g(\mathbf{x})) := \sum_{k=0}^{\ell} u_k P^{(k)}(g_1(\mathbf{b}_1 \cdot \mathbf{x}), \ldots, g_m(\mathbf{b}_m \cdot \mathbf{x})). \tag{38}$$

As in the case of DQI for max-XORSAT above, we write the DQI state

$$|P(f)\rangle = \sum_{\mathbf{x} \in \mathbb{F}_p^n} P(f(\mathbf{x})) |\mathbf{x}\rangle = \sum_{\mathbf{x} \in \mathbb{F}_p^n} Q(g(\mathbf{x})) |\mathbf{x}\rangle = |Q(g)\rangle \tag{39}$$

as a linear combination of $|P^{(0)}\rangle, \ldots, |P^{(\ell)}\rangle$ defined as

$$|P^{(k)}\rangle := \frac{1}{\sqrt{p^{n-k}\binom{m}{k}}} \sum_{\mathbf{x} \in \mathbb{F}_p^n} P^{(k)}(g_1(\mathbf{b}_1 \cdot \mathbf{x}), \ldots, g_m(\mathbf{b}_m \cdot \mathbf{x})) |\mathbf{x}\rangle . \tag{40}$$

By definition,

$$P^{(k)}(g_1(\mathbf{b}_1 \cdot \mathbf{x}), \ldots, g_m(\mathbf{b}_m \cdot \mathbf{x})) = \sum_{\substack{i_1, \ldots, i_k \\ \text{distinct}}} \prod_{i \in \{i_1, \ldots, i_k\}} g_i(\mathbf{b}_i \cdot \mathbf{x}). \tag{41}$$

Substituting (31) into (41) yields

$$\begin{aligned} P^{(k)}(g_1(\mathbf{b}_1 \cdot \mathbf{x}), \ldots, g_m(\mathbf{b}_m \cdot \mathbf{x})) &= \sum_{\substack{i_1, \ldots, i_k \\ \text{distinct}}} \prod_{i \in \{i_1, \ldots, i_k\}} \left(\frac{1}{\sqrt{p}} \sum_{y_i \in \mathbb{F}_p} \omega_p^{-y_i \mathbf{b}_i \cdot \mathbf{x}} \, \tilde{g}_i(y_i) \right) && (42) \\ &= \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}| = k}} \frac{1}{\sqrt{p^k}} \, \omega_p^{-(B^T \mathbf{y}) \cdot \mathbf{x}} \prod_{\substack{i=1 \\ y_i \neq 0}}^{m} \tilde{g}_i(y_i). && (43) \end{aligned}$$

From this we see that the Quantum Fourier Transform of $|P^{(k)}\rangle$ is

$$|\widetilde{P}^{(k)}\rangle := F^{\otimes n} |P^{(k)}\rangle = \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}| = k}} \left(\prod_{\substack{i=1 \\ y_i \neq 0}}^{m} \tilde{g}_i(y_i) \right) |B^T \mathbf{y}\rangle \tag{44}$$

where $F_{ij} = \omega_p^{ij}/\sqrt{p}$ with $i, j = 0, \dots, p-1$. As in the case of max-XORSAT earlier, if $|\mathbf{y}| < d^\perp/2$, then $B^T\mathbf{y}$ are all distinct. Therefore, if $\ell < d^\perp/2$ then $|\widetilde{P}^{(0)}\rangle, \dots, |\widetilde{P}^{(\ell)}\rangle$ form an orthonormal set and so do $|P^{(0)}\rangle, \dots, |P^{(\ell)}\rangle$. Thus, the DQI state $|P(f)\rangle = \sum_{\mathbf{x}\in\mathbb{F}_p^n} P(f(\mathbf{x}))\,|\mathbf{x}\rangle$ can be expressed as

$$|P(f)\rangle = \sum_{k=0}^{\ell} w_k\,|P^{(k)}\rangle \tag{45}$$

where

$$w_k = u_k\sqrt{p^{n-k}\binom{m}{k}}, \tag{46}$$

and $\langle P(f)|P(f)\rangle = \|\mathbf{w}\|^2$.

5.2.2 DQI Algorithm for General max-LINSAT

DQI for general max-LINSAT employs three quantum registers: a *weight register* comprising $\lceil \log_2 \ell \rceil$ qubits, an *error register* with $m\lceil \log_2 p \rceil$ qubits, and a *syndrome register* with $n\lceil \log_2 p\rceil$ qubits. We will consider the error and syndrome registers as consisting of m and n subregisters, respectively, where each subregister consists of $\lceil \log_2 p \rceil$ qubits[2]. We will also regard the ordered collection of least significant[3] qubits from the m subregisters of the error register as the fourth register. We will refer to this m qubit register as the *mask register*.

The task of DQI is to produce the state $|Q(g)\rangle$, which by construction is equal to $|P(f)\rangle$. We proceed as follows. First, as in the $p = 2$ case, we initialize the weight register in the normalized state

$$\sum_{k=0}^{\ell} w_k\,|k\rangle\,. \tag{47}$$

Next, conditioned on the value k in the weight register, we prepare the mask register in the corresponding Dicke state

$$\to \sum_{k=0}^{\ell} w_k\,|k\rangle\,\frac{1}{\sqrt{\binom{m}{k}}}\sum_{\substack{\boldsymbol{\mu}\in\{0,1\}^m\\ |\boldsymbol{\mu}|=k}} |\boldsymbol{\mu}\rangle\,. \tag{48}$$

Then, we uncompute $|k\rangle$ using the fact that $k = |\boldsymbol{\mu}|$, which yields

$$\to \sum_{k=0}^{\ell} w_k\frac{1}{\sqrt{\binom{m}{k}}}\sum_{\substack{\boldsymbol{\mu}\in\{0,1\}^m\\ |\boldsymbol{\mu}|=k}} |\boldsymbol{\mu}\rangle\,. \tag{49}$$

Subsequently, we will turn the content of the error register from a superposition of *bit* strings $\boldsymbol{\mu} \in \{0,1\}^m$ of Hamming weight k into a superposition of *symbol* strings $\mathbf{y} \in \mathbb{F}_p^m$ of Hamming weight k. Let G_i denote an operator acting on $\lceil \log_2 p\rceil$ qubits such that

$$G_i\,|0\rangle = |0\rangle\,, \qquad G_i\,|1\rangle = \sum_{y\in\mathbb{F}_p} \tilde{g}_i(y)\,|y\rangle\,. \tag{50}$$

[2] One can also regard each subregister as a logical p-level quantum system, *i.e.* a qudit, encoded in $\lceil \log_2 p\rceil$ qubits.

[3] Least significant qubit is the one which is in the state $|1\rangle$ when the subregister stores the value $1 \in \mathbb{F}_p$.

By reparametrizing from $f_1, \ldots, f_m$ to $g_1, \ldots, g_m$ we ensured that $\tilde{g}_i(0) = 0$ for all i, so

$$G_i \left|1\right\rangle = \sum_{y \in \mathbb{F}_p^*} \tilde{g}_i(y) \left|y\right\rangle \tag{51}$$

where $\mathbb{F}_p^* = \mathbb{F}_p \setminus \{0\}$. This guarantees that $G_i \left|0\right\rangle = \left|0\right\rangle$ is orthogonal to $G_i \left|1\right\rangle$, so we may assume that G_i is unitary. It may be realized, *e.g.* using the techniques from [25]. The fact that $\tilde{g}_i(0) = 0$ also implies that the parallel application $G := \prod_{i=1}^m G_i$ of each G_i to the respective subregister of the error register preserves the Hamming weight. More precisely, the symbol string on every term in the expansion of $G\left|\boldsymbol{\mu}\right\rangle$ in the computational basis has the same Hamming weight as $\boldsymbol{\mu}$. Indeed, using (50) and (51), we find

$$\sum_{\substack{\boldsymbol{\mu} \in \{0,1\}^m \\ |\boldsymbol{\mu}|=k}} G\left|\boldsymbol{\mu}\right\rangle = \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}\left(y_{y(1)}\right) \ldots \tilde{g}_{y(k)}\left(y_{y(k)}\right) \left|\mathbf{y}\right\rangle \tag{52}$$

where y_i denotes the i^{th} entry of $\mathbf{y}$, and $y(j)$ denotes the index of the j^{th} nonzero entry of $\mathbf{y}$. Thus, by applying G to the error register, we obtain

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}\left(y_{y(1)}\right) \ldots \tilde{g}_{y(k)}\left(y_{y(k)}\right) \left|\mathbf{y}\right\rangle . \tag{53}$$

Next, we reversibly compute $B^T\mathbf{y}$ into the syndrome register, obtaining the state

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}\left(y_{y(1)}\right) \ldots \tilde{g}_{y(k)}\left(y_{y(k)}\right) \left|\mathbf{y}\right\rangle \left|B^T\mathbf{y}\right\rangle . \tag{54}$$

The task of finding $\mathbf{y}$ from $B^T\mathbf{y}$ is the bounded distance decoding problem on $C^\perp = \{\mathbf{y} \in \mathbb{F}_p^m : B^T\mathbf{y} = \mathbf{0}\}$. Thus uncomputing the content of the error register can be done efficiently whenever the bounded distance decoding problem on $C^\perp$ can be solved efficiently out to distance ℓ.

Uncomputing disentangles the syndrome register from the error register leaving it in the state

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}\left(y_{y(1)}\right) \ldots \tilde{g}_{y(k)}\left(y_{y(k)}\right) \left|B^T\mathbf{y}\right\rangle \tag{55}$$

$$= \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_p^m \\ |\mathbf{y}|=k}} \left(\prod_{\substack{i=1 \\ y_i \neq 0}}^{m} \tilde{g}_i(y_i) \right) \left|B^T\mathbf{y}\right\rangle . \tag{56}$$

Comparing (56) with (44) shows that this is equal to

$$= \sum_{k=0}^{\ell} w_k \left|\widetilde{P}^{(k)}\right\rangle . \tag{57}$$

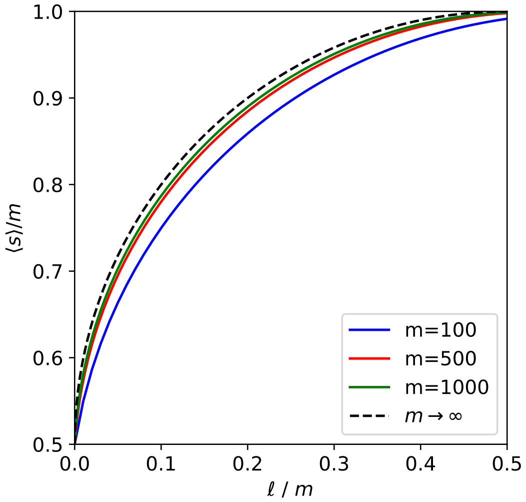

Figure 2: Here we plot the expected fraction $\langle s \rangle / m$ of satisfied constraints, as dictated by Lemma 6.2, upon measuring $|P(f)\rangle$ when P is the optimal degree-ℓ polynomial. We show the balanced case where $|f_i^{-1}(+1)| \simeq p/2$ for all i. Accordingly, the dashed black line corresponds to the asymptotic formula (3) with $r/p = 1/2$.

Thus, by applying the inverse Quantum Fourier Transform on $\mathbb{F}_p^n$ to the syndrome register we obtain

$$\rightarrow \sum_{k=0}^{\ell} w_k \, |P^{(k)}\rangle \, . \tag{58}$$

By the definition of w_k this is $|Q(g)\rangle$, which equals $|P(f)\rangle$.

6 Optimal Expected Fraction of Satisfied Constraints

In this section we prove Theorem 1.1 for the asymptotic performance of DQI as well as quantify the finite-size corrections. Before doing so, we make two remarks. First, we note that the condition $2\ell + 1 < d^{\perp}$ is equivalent to saying that ℓ must be less than half the distance of the code, which is the same condition needed to guarantee that the decoding problem used in the uncomputation step of DQI has a unique solution. This condition is met by our OPI example. It is not met in our irregular max-XORSAT example. In §7 we show that the semicircle law (3) remains a good approximation even beyond this limit, for average case $\mathbf{v}$. Second, we note that Lemma 6.2 gives an *exact* expression for the expected number of constraints satisfied by DQI at any finite size and for any choice of polynomial P, in terms of an $(\ell + 1) \times (\ell + 1)$ quadratic form. By numerically evaluating optimum of this quadratic form we find that the finite size behavior converges fairly rapidly to the asymptotic behavior of Theorem 1.1, as illustrated in Fig. 2.

6.1 Preliminaries

Recall from §5.2.1 that the quantum state $|\widetilde{P}(f)\rangle = F^{\otimes n}|P(f)\rangle$ obtained by applying the Quantum Fourier Transform to a DQI state $|P(f)\rangle = \sum_{\mathbf{x}\in\mathbb{F}_p^n} P(f(\mathbf{x}))|\mathbf{x}\rangle$ is

$$|\widetilde{P}(f)\rangle = \sum_{k=0}^{\ell} \frac{w_k}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k}} \left(\prod_{\substack{i=1 \\ y_i\neq 0}}^{m} \tilde{g}_i(y_i) \right) |B^T\mathbf{y}\rangle \tag{59}$$

This can be written more succinctly as

$$|\widetilde{P}(f)\rangle = \sum_{k=0}^{\ell} \frac{w_k}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k}} \tilde{g}(\mathbf{y})\, |B^T\mathbf{y}\rangle \tag{60}$$

by defining for any $\mathbf{y} \in \mathbb{F}_p^m$

$$\tilde{g}(\mathbf{y}) = \prod_{\substack{i=1 \\ y_i\neq 0}}^{m} \tilde{g}_i(y_i) \tag{61}$$

where we regard the product of zero factors as 1, so that $\tilde{g}_i(\mathbf{0}) = 1$. Next, we note that

$$\sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k}} |\tilde{g}(\mathbf{y})|^2 = \sum_{\substack{i_1,\ldots,i_k\in\{1,\ldots,m\} \\ \text{distinct}}} \left(\sum_{y_1\in\mathbb{F}_p} |\tilde{g}_{i_1}(y_1)|^2 \right) \ldots \left(\sum_{y_k\in\mathbb{F}_p} |\tilde{g}_{i_k}(y_k)|^2 \right) = \binom{m}{k} \tag{62}$$

where we used the fact that $\tilde{g}_i(0) = 0$ for all i. Lastly, we formally restate the following fact, which we derived in §5.2.1.

Lemma 6.1. *Let $d^\perp$ denote the minimum distance of the code $C^\perp = \{\mathbf{d} \in \mathbb{F}_p^m : B^T\mathbf{d} = \mathbf{0}\}$. If $2\ell < d^\perp$, then $\langle\widetilde{P}(f)|\widetilde{P}(f)\rangle = \|\mathbf{w}\|^2$.*

The proof of Theorem 1.1 consists of two parts. In the next subsection, we express the expected fraction $\langle s^{(m,\ell)}\rangle/m$ of satisfied constraints as a quadratic form involving a certain tridiagonal matrix. Then, we derive a closed-form asymptotic formula for the maximum eigenvalue of the matrix. We end this section by combining the two parts into a proof of Theorem 1.1. We note that [28] encountered a similar matrix in the context of proving bounds on the size of codes with good distance, and derived a similar expression for the maximum eigenvalue, using similar methods.

6.2 Expected Number of Satisfied Constraints

Lemma 6.2. *Let $f(\mathbf{x}) = \sum_{i=1}^m f_i\left(\sum_{j=1}^n B_{ij}x_j\right)$ be a max-LINSAT objective function with matrix $B \in \mathbb{F}_p^{m\times n}$ for a prime p and positive integers m and n such that $m > n$. Suppose that $|f_i^{-1}(+1)| = r$ for some $r \in \{1,\ldots,p-1\}$. Let P be a degree-ℓ polynomial normalized so that $\langle P(f)|P(f)\rangle = 1$ with $P(f) = \sum_{k=0}^{\ell} w_k P^{(k)}(f_1,\ldots,f_m)/\sqrt{p^{n-k}\binom{m}{k}}$ its decomposition as a linear combination of* *elementary symmetric polynomials. Let $\langle s^{(m,\ell)}\rangle$ be the expected number of satisfied constraints for the symbol string obtained upon measuring the DQI state $|P(f)\rangle$ in the computational basis. If $2\ell+1 < d^\perp$ where $d^\perp$ is the minimum distance of the code $C^\perp = \{\mathbf{d}\in\mathbb{F}_p^m : B^T\mathbf{d}=\mathbf{0}\}$, then*

$$\langle s^{(m,\ell)}\rangle = \frac{mr}{p} + \frac{\sqrt{r(p-r)}}{p}\mathbf{w}^\dagger A^{(m,\ell,d)}\mathbf{w} \tag{63}$$

where $\mathbf{w}=(w_0,\dots,w_\ell)^T$ and $A^{(m,\ell,d)}$ is the $(\ell+1)\times(\ell+1)$ symmetric tridiagonal matrix

$$A^{(m,\ell,d)} = \begin{bmatrix} 0 & a_1 & & & \\ a_1 & d & a_2 & & \\ & a_2 & 2d & \ddots & \\ & & \ddots & & a_\ell \\ & & & a_\ell & \ell d \end{bmatrix} \tag{64}$$

with $a_k = \sqrt{k(m-k+1)}$ and $d = \frac{p-2r}{\sqrt{r(p-r)}}$.

Proof. The number of constraints satisfied by $\mathbf{x}\in\mathbb{F}_p^n$ is

$$s(\mathbf{x}) = \sum_{i=1}^m \mathbb{1}_{F_i}(\mathbf{b}_i\cdot\mathbf{x}) \tag{65}$$

where

$$\mathbb{1}_A(x) = \begin{cases} 1 & \text{if} \quad x\in A \\ 0 & \text{otherwise} \end{cases} \tag{66}$$

is the *indicator function* of the set A. For any $v, x\in\mathbb{F}_p$, we can write

$$\mathbb{1}_{\{v\}}(x) = \frac{1}{p}\sum_{a\in\mathbb{F}_p}\omega_p^{a(x-v)} \tag{67}$$

so

$$\mathbb{1}_{F_i}(x) = \sum_{v\in F_i}\mathbb{1}_{\{v\}}(x) = \frac{1}{p}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{a(x-v)}. \tag{68}$$

Substituting into (65), we have

$$s(\mathbf{x}) = \frac{1}{p}\sum_{i=1}^m\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{a(\mathbf{b}_i\cdot\mathbf{x}-v)}. \tag{69}$$

The expected number of constraints satisfied by a symbol string sampled from the output distribution of a DQI state $|P(f)\rangle$ is

$$\langle s^{(m,\ell)}\rangle = \langle P(f)|S_f|P(f)\rangle \tag{70}$$

where $S_f = \sum_{\mathbf{x}\in\mathbb{F}_p^n} s(\mathbf{x})|\mathbf{x}\rangle\langle\mathbf{x}|$. We can express the observable S_f in terms of the *clock operator* $Z = \sum_{b\in\mathbb{F}_p} \omega_p^b |b\rangle\langle b|$ on $\mathbb{C}^p$ as

$$\begin{aligned} S_f &= \sum_{\mathbf{x}\in\mathbb{F}_p^n} s(\mathbf{x})|\mathbf{x}\rangle\langle\mathbf{x}| && (71)\\ &= \frac{1}{p}\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\sum_{\mathbf{x}\in\mathbb{F}_p^n} \omega_p^{a(\mathbf{b}_i\cdot\mathbf{x}-v)}|\mathbf{x}\rangle\langle\mathbf{x}| && (72)\\ &= \frac{1}{p}\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{-av}\prod_{j=1}^{n}\sum_{x_j\in\mathbb{F}_p}\omega_p^{aB_{ij}x_j}|x_j\rangle\langle x_j| && (73)\\ &= \frac{1}{p}\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{-av}\prod_{j=1}^{n} Z^{aB_{ij}}. && (74)\end{aligned}$$

Next, we apply the Fourier transform $F_{ij} = \omega_p^{ij}/\sqrt{p}$ with $i,j = 0,\dots,p-1$ to write $\langle s^{(m,\ell)}\rangle$ in terms of the *shift operator* $X = \sum_{b\in\mathbb{F}_p} |b+1\rangle\langle b|$ on $\mathbb{C}^p$ as

$$\begin{aligned} \langle s^{(m,\ell)}\rangle &= \frac{1}{p}\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{-av}\langle P(f)|\prod_{j=1}^{n} Z_j^{aB_{ij}}|P(f)\rangle && (75)\\ &= \frac{1}{p}\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{-av}\langle \widetilde{P}(f)|\prod_{j=1}^{n} X_j^{-aB_{ij}}|\widetilde{P}(f)\rangle. && (76)\end{aligned}$$

where we used $FZF^\dagger = X^{-1}$ and $|\widetilde{P}(f)\rangle = F^{\otimes n}|P(f)\rangle$.

Substituting (60) into (76), we get

$$\langle s^{(m,\ell)}\rangle = \frac{1}{p}\sum_{k_1,k_2=0}^{\ell}\frac{w_{k_1}^* w_{k_2}}{\sqrt{\binom{m}{k_1}\binom{m}{k_2}}}\sum_{\substack{\mathbf{y}_1,\mathbf{y}_2\in\mathbb{F}_p^m\\ |\mathbf{y}_1|=k_1\\ |\mathbf{y}_2|=k_2}}\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2)\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{-av}\langle B^T\mathbf{y}_1|\prod_{j=1}^{n}X_j^{-aB_{ij}}|B^T\mathbf{y}_2\rangle \tag{77}$$

Let $\mathbf{e}_1,\dots,\mathbf{e}_m \in \mathbb{F}_p^m$ denote the standard basis of one-hot vectors. Then

$$\prod_{j=1}^{n} X_j^{-aB_{ij}}|B^T\mathbf{y}_2\rangle = |B^T\mathbf{y}_2 - a\mathbf{b}_i^T\rangle = |B^T(\mathbf{y}_2 - a\mathbf{e}_i)\rangle \tag{78}$$

so

$$\langle s^{(m,\ell)}\rangle = \frac{1}{p}\sum_{k_1,k_2=0}^{\ell}\frac{w_{k_1}^* w_{k_2}}{\sqrt{\binom{m}{k_1}\binom{m}{k_2}}}\sum_{\substack{\mathbf{y}_1,\mathbf{y}_2\in\mathbb{F}_p^m\\ |\mathbf{y}_1|=k_1\\ |\mathbf{y}_2|=k_2}}\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2)\sum_{i=1}^{m}\sum_{v\in F_i}\sum_{a\in\mathbb{F}_p}\omega_p^{-av}\langle B^T\mathbf{y}_1|B^T(\mathbf{y}_2 - a\mathbf{e}_i)\rangle. \tag{79}$$

But $|B^T\mathbf{y}_1\rangle$ and $|B^T(\mathbf{y}_2 - a\mathbf{e}_i)\rangle$ are computational basis states, so

$$\langle B^T\mathbf{y}_1|B^T(\mathbf{y}_2 - a\mathbf{e}_i)\rangle = \begin{cases} 1 & \text{if} \quad B^T\mathbf{y}_1 = B^T(\mathbf{y}_2 - a\mathbf{e}_i)\\ 0 & \text{otherwise.}\end{cases} \tag{80}$$

Moreover,

$$B^T\mathbf{y}_1 = B^T(\mathbf{y}_2 - a\mathbf{e}_i) \iff \mathbf{y}_1 - \mathbf{y}_2 + a\mathbf{e}_i \in C^\perp \iff \mathbf{y}_1 = \mathbf{y}_2 - a\mathbf{e}_i \tag{81}$$

where we used the assumption that the smallest Hamming weight of a non-zero symbol string in $C^\perp$ is $d^\perp > 2\ell + 1 \geq k_1 + k_2 + 1$.

There are three possibilities to consider: $|\mathbf{y}_1| = |\mathbf{y}_2| - 1$, $|\mathbf{y}_2| = |\mathbf{y}_1| - 1$, and $|\mathbf{y}_1| = |\mathbf{y}_2|$. We further break up the last case into $\mathbf{y}_1 \neq \mathbf{y}_2$ and $\mathbf{y}_1 = \mathbf{y}_2$. Before simplifying (79), we examine the values of $i \in \{1, \ldots, m\}$ and $a \in \mathbb{F}_p$ for which $\langle B^T\mathbf{y}_1 | B^T(\mathbf{y}_2 - a\mathbf{e}_i)\rangle = 1$ in each of the four cases. We also compute the value of the product $\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2)$.

Consider first the case $|\mathbf{y}_1| = |\mathbf{y}_2| - 1$. Here, $a \neq 0$ and $i \in \{1, \ldots, m\}$ is the position which is zero in $\mathbf{y} := \mathbf{y}_1$ and a in $\mathbf{y}_2$. Therefore, by definition (61)

$$\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2) = |\tilde{g}(\mathbf{y})|^2 \tilde{g}_i(a). \tag{82}$$

Next, suppose $|\mathbf{y}_2| = |\mathbf{y}_1| - 1$. Then $a \neq 0$ and $i \in \{1, \ldots, m\}$ is the position which is zero in $\mathbf{y} := \mathbf{y}_2$ and $-a$ in $\mathbf{y}_1$. Thus,

$$\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2) = |\tilde{g}(\mathbf{y}_2)|^2 \tilde{g}_i^*(-a) = |\tilde{g}(\mathbf{y})|^2 \tilde{g}_i(a). \tag{83}$$

Consider next the case $|\mathbf{y}_1| = |\mathbf{y}_2|$ and $\mathbf{y}_1 \neq \mathbf{y}_2$. Here, $a \neq 0$ and $i \in \{1, \ldots, m\}$ is the position which is $z - a$ in $\mathbf{y}_1$ and z in $\mathbf{y}_2$ for some $z \in \mathbb{F}_p \setminus \{0, a\}$. Let $\mathbf{y} \in \mathbb{F}_p^m$ denote the vector which is zero at position i and agrees with $\mathbf{y}_1$, and hence with $\mathbf{y}_2$, on all other positions. Then

$$\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2) = |\tilde{g}(\mathbf{y})|^2 \tilde{g}_i^*(z - a)\tilde{g}_i(z) = |\tilde{g}(\mathbf{y})|^2 \tilde{g}_i(a - z)\tilde{g}_i(z). \tag{84}$$

Finally, when $\mathbf{y} := \mathbf{y}_1 = \mathbf{y}_2$, we have

$$\tilde{g}^*(\mathbf{y}_1)\tilde{g}(\mathbf{y}_2) = |\tilde{g}(\mathbf{y})|^2. \tag{85}$$

In this case, $a = 0$ and $i \in \{1, \ldots, m\}$.

Putting it all together we can rewrite (79) as

$$\langle s^{(m,\ell)}\rangle = \frac{1}{p}\sum_{k=0}^{\ell-1} \frac{w_k^* w_{k+1}}{\sqrt{\binom{m}{k}\binom{m}{k+1}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k}} |\tilde{g}(\mathbf{y})|^2 \sum_{\substack{i=1 \\ y_i=0}}^{m} \sum_{v\in F_i} \sum_{a\in\mathbb{F}_p^*} \omega_p^{-av}\tilde{g}_i(a) \tag{86}$$

$$+ \frac{1}{p}\sum_{k=0}^{\ell-1} \frac{w_{k+1}^* w_k}{\sqrt{\binom{m}{k+1}\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k}} |\tilde{g}(\mathbf{y})|^2 \sum_{\substack{i=1 \\ y_i=0}}^{m} \sum_{v\in F_i} \sum_{a\in\mathbb{F}_p^*} \omega_p^{-av}\tilde{g}_i(a) \tag{87}$$

$$+ \frac{1}{p}\sum_{k=1}^{\ell} \frac{|w_k|^2}{\binom{m}{k}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k-1}} |\tilde{g}(\mathbf{y})|^2 \sum_{\substack{i=1 \\ y_i=0}}^{m} \sum_{v\in F_i} \sum_{a\in\mathbb{F}_p^*} \sum_{z\in\mathbb{F}_p\setminus\{0,a\}} \omega_p^{-av}\tilde{g}_i(a - z)\tilde{g}_i(z) \tag{88}$$

$$+ \frac{1}{p}\sum_{k=0}^{\ell} \frac{|w_k|^2}{\binom{m}{k}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |\mathbf{y}|=k}} |\tilde{g}(\mathbf{y})|^2 \sum_{i=1}^{m} \sum_{v\in F_i} \sum_{a\in\{0\}} \omega_p^{-av}. \tag{89}$$

Remembering that $\tilde{g}_i(0) = 0$, we recognize the innermost sums in (86) and (87) as the inverse Fourier transform so, for $\mathbf{y}$ with $|\mathbf{y}| = k$, we have

$$\sum_{\substack{i=1\\y_i=0}}^{m} \sum_{v\in F_i} \sum_{a\in\mathbb{F}_p^*} \omega_p^{-av}\tilde{g}_i(a) = \sqrt{p}\sum_{\substack{i=1\\y_i=0}}^{m} \sum_{v\in F_i} g_i(v) = (m-k)\sqrt{r(p-r)} \tag{90}$$

where we used (33) and (34) to calculate

$$g_i(v) = \frac{1-\bar{f}}{\varphi} = \sqrt{\frac{p-r}{pr}} \tag{91}$$

for $v \in F_i$. Similarly, with $\tilde{g}_i(0) = 0$, we recognize the sums indexed by a and z in (88) as the inverse Fourier transform of the convolution of $\tilde{g}_i$ with itself, so

$$\begin{aligned}
\sum_{a\in\mathbb{F}_p} \omega_p^{-av} \sum_{z\in\mathbb{F}_p} \tilde{g}_i(a-z)\tilde{g}_i(z) &= \frac{1}{p}\sum_{a\in\mathbb{F}_p} \omega_p^{-av} \sum_{z\in\mathbb{F}_p}\sum_{x\in\mathbb{F}_p} \omega_p^{x(a-z)} g_i(x) \sum_{y\in\mathbb{F}_p} \omega_p^{yz} g_i(y) && (92)\\
&= \sum_{a,x,y\in\mathbb{F}_p} \omega_p^{a(x-v)} g_i(x) g_i(y) \frac{1}{p}\sum_{z\in\mathbb{F}_p} \omega_p^{(y-x)z} && (93)\\
&= \sum_{x\in\mathbb{F}_p} g_i(x)^2 \sum_{a\in\mathbb{F}_p} \omega_p^{a(x-v)} = p\ g_i(v)^2 && (94)
\end{aligned}$$

and, for $\mathbf{y}$ with $|\mathbf{y}| = k-1$, we have

$$\begin{aligned}
\sum_{\substack{i=1\\y_i=0}}^{m} \sum_{v\in F_i} \sum_{a\in\mathbb{F}_p^*} \sum_{z\in\mathbb{F}_p\setminus\{0,a\}} \omega_p^{-av}\tilde{g}_i(a-z)\tilde{g}_i(z) &= \sum_{\substack{i=1\\y_i=0}}^{m} \sum_{v\in F_i} \left(\sum_{a,z\in\mathbb{F}_p} \omega_p^{-av}\tilde{g}_i(a-z)\tilde{g}_i(z) - \sum_{z\in\mathbb{F}_p} |\tilde{g}_i(z)|^2 \right)\\
&= \sum_{\substack{i=1\\y_i=0}}^{m} \sum_{v\in F_i} \left(p\ g_i(v)^2 - 1\right) && (95)\\
&= \sum_{\substack{i=1\\y_i=0}}^{m} \sum_{v\in F_i} \left(\frac{p-r}{r} - 1\right) && (96)\\
&= (m-k+1)(p-2r) && (97)
\end{aligned}$$

where we used (91). Substituting back into (86), (87), and (88), we obtain

$$\langle s^{(m,\ell)} \rangle = \frac{\sqrt{r(p-r)}}{p} \sum_{k=0}^{\ell-1} w_k^* w_{k+1} \frac{m-k}{\sqrt{\binom{m}{k}\binom{m}{k+1}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |y|=k}} |\tilde{g}(\mathbf{y})|^2 \tag{98}$$

$$+ \frac{\sqrt{r(p-r)}}{p} \sum_{k=0}^{\ell-1} w_{k+1}^* w_k \frac{m-k}{\sqrt{\binom{m}{k+1}\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |y|=k}} |\tilde{g}(\mathbf{y})|^2 \tag{99}$$

$$+ \frac{p-2r}{p} \sum_{k=1}^{\ell} |w_k|^2 \frac{m-k+1}{\binom{m}{k}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |y|=k-1}} |\tilde{g}(\mathbf{y})|^2 \tag{100}$$

$$+ \frac{mr}{p} \sum_{k=0}^{\ell} |w_k|^2 \frac{1}{\binom{m}{k}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_p^m \\ |y|=k}} |\tilde{g}(\mathbf{y})|^2 \tag{101}$$

and using equation (62), we get

$$\langle s^{(m,\ell)} \rangle = \frac{\sqrt{r(p-r)}}{p} \sum_{k=0}^{\ell-1} w_k^* w_{k+1} \sqrt{(k+1)(m-k)} \tag{102}$$

$$+ \frac{\sqrt{r(p-r)}}{p} \sum_{k=0}^{\ell-1} w_{k+1}^* w_k \sqrt{(k+1)(m-k)} \tag{103}$$

$$+ \frac{p-2r}{p} \sum_{k=0}^{\ell} |w_k|^2 k + \frac{mr}{p}. \tag{104}$$

Defining

$$A^{(m,\ell,d)} = \begin{bmatrix} 0 & a_1 & & & \\ a_1 & d & a_2 & & \\ & a_2 & 2d & \ddots & \\ & & \ddots & & a_\ell \\ & & & a_\ell & \ell d \end{bmatrix} \tag{105}$$

where $a_k = \sqrt{k(m-k+1)}$ for $k = 1, \ldots, \ell$ and $d \in \mathbb{R}$, we can write

$$\langle s^{(m,\ell)} \rangle = \frac{mr}{p} + \frac{\sqrt{r(p-r)}}{p} \mathbf{w}^\dagger A^{(m,\ell,d)} \mathbf{w} \tag{106}$$

where $\mathbf{w} = (w_0, \ldots, w_\ell)^T$ and $d = \frac{p-2r}{\sqrt{r(p-r)}}$. □

6.3 Asymptotic Formula for Maximum Eigenvalue of Matrix $A^{(m,\ell,d)}/m$

Lemma 6.3. *Let $\lambda_{max}^{(m,\ell,d)}$ denote the maximum eigenvalue of the symmetric tridiagonal matrix $A^{(m,\ell,d)}$ defined in (105). If $\ell \leq m/2$ and $d \geq -\frac{m-2\ell}{\sqrt{\ell(m-\ell)}}$, then*

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lambda_{max}^{(m,\ell,d)}}{m} = \mu d + 2\sqrt{\mu(1-\mu)} \tag{107}$$

where the limit is taken as both m and ℓ tend to infinity with the ratio $\mu = \ell/m$ fixed.

Proof. First, we show that if $\ell \leq m/2$, then

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lambda_{\max}^{(m,\ell,d)}}{m} \geq \mu d + 2\sqrt{\mu(1-\mu)}. \tag{108}$$

Define vector $v^{(m,\ell)} \in \mathbb{R}^{\ell+1}$ as

$$v_i^{(m,\ell)} = \begin{cases} 0 & \text{for} \quad i = 0, 1, \dots, \ell - \lceil\sqrt{\ell}\rceil \\ \lceil\sqrt{\ell}\rceil^{-1/2} & \text{for} \quad i = \ell - \lceil\sqrt{\ell}\rceil + 1, \dots, \ell. \end{cases} \tag{109}$$

Then, $\|v^{(m,\ell)}\|_2^2 = \frac{\lceil\sqrt{\ell}\rceil}{\lceil\sqrt{\ell}\rceil} = 1$. But

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lambda_{\max}^{(m,\ell,d)}}{m} \geq \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\left(v^{(m,\ell)}\right)^T A^{(m,\ell,d)} v^{(m,\ell)}}{m} \tag{110}$$

$$= \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{1}{m\lceil\sqrt{\ell}\rceil} \sum_{k=\ell-\lceil\sqrt{\ell}\rceil+1}^{\ell} kd \right) + \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{1}{m\lceil\sqrt{\ell}\rceil} \sum_{k=\ell-\lceil\sqrt{\ell}\rceil+2}^{\ell} 2a_k \right). \tag{111}$$

The second term in (111) is bounded below by

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{1}{m\lceil\sqrt{\ell}\rceil} \sum_{k=\ell-\lceil\sqrt{\ell}\rceil+2}^{\ell} 2a_k \right) \geq \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{2a_{\ell-\lceil\sqrt{\ell}\rceil+2}(\lceil\sqrt{\ell}\rceil - 1)}{m\lceil\sqrt{\ell}\rceil} \right) \tag{112}$$

$$= \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{2\sqrt{(\ell - \lceil\sqrt{\ell}\rceil + 2)(m - \ell + \lceil\sqrt{\ell}\rceil - 1)}}{m} \cdot \frac{\lceil\sqrt{\ell}\rceil - 1}{\lceil\sqrt{\ell}\rceil} \right)$$

$$= 2\sqrt{\mu(1-\mu)} \tag{113}$$

where we used the fact that a_k increases as a function of k for $k \leq m/2$. If $d \geq 0$, then the first term in (111) is bounded below by

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{1}{m\lceil\sqrt{\ell}\rceil} \sum_{k=\ell-\lceil\sqrt{\ell}\rceil+1}^{\ell} kd \right) \geq \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lceil\sqrt{\ell}\rceil \cdot (\ell - \lceil\sqrt{\ell}\rceil + 1) \cdot d}{m\lceil\sqrt{\ell}\rceil} = \mu d \tag{114}$$

and if $d < 0$, then it is bounded below by

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \left(\frac{1}{m\lceil\sqrt{\ell}\rceil} \sum_{k=\ell-\lceil\sqrt{\ell}\rceil+1}^{\ell} kd \right) \geq \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lceil\sqrt{\ell}\rceil \cdot \ell \cdot d}{m\lceil\sqrt{\ell}\rceil} = \mu d. \tag{115}$$

Putting it all together, we get

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lambda_{\max}^{(m,\ell,d)}}{m} \geq \mu d + 2\sqrt{\mu(1-\mu)}. \tag{116}$$

Next, we establish a matching upper bound on $\lambda_{\max}^{(m,\ell,d)}$. For $k = 0, 1, \ldots, \ell$, let $R_k^{(m,\ell,d)}$ denote the sum of the off-diagonal entries in the k^{th} row of $A^{(m,\ell,d)}$ and set $a_0 = a_{\ell+1} = 0$, so that $R_k = a_k + a_{k+1}$. By Gershgorin's circle theorem, for every eigenvalue $\lambda^{(m,\ell,d)}$ of $A^{(m,\ell,d)}$, we have

$$\begin{aligned} \lambda^{(m,\ell,d)} &\leq \max_{k\in\{0,\ldots,\ell\}} (kd + R_k) && (117) \\ &= \max_{k\in\{0,\ldots,\ell\}} \left(kd + \sqrt{k(m-k+1)} + \sqrt{(k+1)(m-k)} \right). && (118) \end{aligned}$$

By assumption, $\ell < m/2$, so $\sqrt{(k+1)(m-k)} \geq \sqrt{k(m-k+1)}$ for all k in the sum. Thus (118) yields

$$\begin{aligned} \lambda^{(m,\ell,d)} &\leq \max_{k\in\{0,\ldots,\ell\}} \left(kd + 2\sqrt{(k+1)(m-k)} \right) && (119) \\ &\leq \max_{k\in\{0,\ldots,\ell\}} \left(kd + 2\sqrt{k(m-k)} + 2\sqrt{m-k} \right) && (120) \\ &\leq 2\sqrt{m} + \max_{k\in\{0,\ldots,\ell\}} \left(kd + 2\sqrt{k(m-k)} \right) && (121) \\ &= 2\sqrt{m} + \max_{k\in\{0,\ldots,\ell\}} \xi(k) && (122) \end{aligned}$$

where we define $\xi(x) = xd + 2\sqrt{x(m-x)}$. Note that $\xi''(x) = -\frac{m^2}{2[x(m-x)]^{3/2}} < 0$ for all $x \in (0, m)$, so the derivative $\xi'(x) = d + \frac{m-2x}{\sqrt{x(m-x)}}$ is decreasing in this interval. However, by assumption $d \geq -\frac{m-2\ell}{\sqrt{\ell(m-\ell)}}$, so $\xi'(x) \geq \xi'(\ell) \geq 0$ for $x \in (0, \ell]$, since $\ell < m/2$. Therefore, $\xi(x)$ is increasing on this interval and we have

$$\lambda^{(m,\ell,d)} \leq 2\sqrt{m} + \ell d + 2\sqrt{\ell(m-\ell)}. \tag{123}$$

But then

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lambda_{\max}^{(m,\ell,d)}}{m} \leq \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{2\sqrt{m} + \ell d + 2\sqrt{\ell(m-\ell)}}{m} = \mu d + 2\sqrt{\mu(1-\mu)} \tag{124}$$

which establishes the matching upper bound and completes the proof of the lemma. □

6.4 Optimal Asymptotic Expected Fraction of Satisfied Constraints

In this subsection we use Lemmas 6.2 and 6.3 to prove Theorem 1.1.

Proof. Recall from Lemma 6.1 that $\|\mathbf{w}\|_2 = 1$. Therefore, the expected number of satisfied constraints is maximized by choosing $\mathbf{w}$ in (63) to be the normalized eigenvector of $A^{(m,\ell,d)}$ corresponding to its maximal eigenvalue. This leads to

$$\frac{\langle s^{(m,\ell)} \rangle_{\text{opt}}}{m} = \rho + \sqrt{\rho(1-\rho)}\frac{\lambda_{\max}^{(m,\ell,d)}}{m} \tag{125}$$

where $\rho = \frac{r}{p}$ and $d = \frac{p-2r}{\sqrt{r(p-r)}}$. Consider first the case of $\frac{r}{p} \leq 1 - \frac{\ell}{m}$. Then $\frac{p-r}{r} \geq \frac{\ell}{m-\ell}$ and

$$d = \frac{p-2r}{\sqrt{r(p-r)}} = \sqrt{\frac{p-r}{r}} - \sqrt{\frac{r}{p-r}} \geq \sqrt{\frac{\ell}{m-\ell}} - \sqrt{\frac{m-\ell}{\ell}} = -\frac{m-2\ell}{\sqrt{\ell(m-\ell)}}. \tag{126}$$

Moreover, $\ell < (d^{\perp} - 1)/2 \leq (m-1)/2$. Therefore, Lemma 6.3 applies and we have

$$\begin{aligned}
\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\langle s^{(m,\ell)} \rangle_{\text{opt}}}{m} &= \rho + \sqrt{\rho(1-\rho)} \lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\lambda_{\max}^{(m,\ell,d)}}{m}. && (127)\\
&= \rho + \sqrt{\rho(1-\rho)}\left(\mu d + 2\sqrt{\mu(1-\mu)}\right) && (128)\\
& && (129)
\end{aligned}$$

Recalling $d = \frac{1-2\rho}{\sqrt{\rho(1-\rho)}}$ this yields

$$\begin{aligned}
\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\langle s^{(m,\ell)} \rangle_{\text{opt}}}{m} &= \mu + \rho - 2\mu\rho + 2\sqrt{\mu(1-\mu)\rho(1-\rho)} && (130)\\
&= \left(\sqrt{\mu(1-\rho)} + \sqrt{\rho(1-\mu)}\right)^2 && (131)
\end{aligned}$$

for $\rho \leq 1-\mu$. In particular, if $\rho = 1-\mu$, then $\left(\sqrt{\mu(1-\rho)} + \sqrt{\rho(1-\mu)}\right)^2 = 1$. But $\langle s^{(m,\ell)} \rangle_{\text{opt}}$ is an increasing function of r that cannot exceed 1. Consequently,

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\langle s^{(m,\ell)} \rangle_{\text{opt}}}{m} = 1 \tag{132}$$

for $\rho > 1-\mu$. Putting it all together, we obtain

$$\lim_{\substack{m,\ell\to\infty \\ \ell/m=\mu}} \frac{\langle s^{(m,\ell)} \rangle_{\text{opt}}}{m} = \begin{cases} \left(\sqrt{\mu(1-\rho)} + \sqrt{\rho(1-\mu)}\right)^2 & \text{if } \rho \leq 1-\mu \\ 1 & \text{otherwise} \end{cases} \tag{133}$$

which completes the proof of Theorem 1.1. □

7 Removing the Minimum Distance Assumption

So far, in our description and analysis of DQI we have always assumed that $2\ell + 1 < d^\perp$. This condition buys us several advantages. First, it ensures that the states $|\widetilde{P}^{(0)}\rangle, \ldots, |\widetilde{P}^{(\ell)}\rangle$, from which we construct

$$|\widetilde{P}(f)\rangle = \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v}\cdot\mathbf{y}} |B^T \mathbf{y}\rangle\,, \tag{134}$$

form an orthonormal set which implies that

$$\langle \widetilde{P}(f) | \widetilde{P}(f) \rangle = \mathbf{w}^\dagger \mathbf{w}. \tag{135}$$

Second, it allows us to obtain an *exact* expression for the expected number of constraints satisfied without needing to know the weight distributions of either the code C or $C^\perp$, namely for $p = 2$

$$\langle s \rangle = \frac{1}{2} m + \frac{1}{2} \mathbf{w}^\dagger A^{(m,\ell,0)} \mathbf{w} \tag{136}$$

where $A^{(m,\ell,d)}$ is the $(\ell+1) \times (\ell+1)$ matrix defined in (64). These facts in turn allow us to prove Theorem 1.1.

For the irregular LDPC code described in §9, we can estimate $d^\perp$ using the Gilbert-Varshamov bound[4]. By experimentally testing belief propagation, we find that for some codes it is able to correct slightly more than $(d^\perp - 1)/2$ errors with high reliability. Under this circumstance, equations (135) and (136) no longer hold exactly. Here, we prove that they continue to hold in expectation for max-XORSAT with uniformly average $\mathbf{v}$, up to small corrections due to decoding failures. The precise statement of our result is given in Theorem 7.1.

In the remainder of this section we first describe precisely what we mean by the DQI algorithm in the case of $2\ell + 1 \geq d^\perp$. (For simplicity, we consider the case $p = 2$.)

As the initial step of DQI, we perform classical preprocessing to choose $\mathbf{w} \in \mathbb{R}^{\ell+1}$, which is equivalent to making a choice of degree-ℓ polynomial P. In the case $2\ell + 1 < d^\perp$ we can exactly compute the choice of $\mathbf{w}$ that maximizes $\langle s \rangle / m$. Specifically, it is the principal eigenvector of $A^{(m,\ell,d)}$ defined in (64). Once we reach or exceed $2\ell + 1 = d^\perp$ the principal eigenvector of this matrix is not necessarily the optimal choice. But we can still use it as our choice of $\mathbf{w}$, and as we will show below, it remains a good choice.

After choosing $\mathbf{w}$, the next step in the DQI algorithm, as discussed in §5.1.2 is to prepare the state

$$\sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v}\cdot\mathbf{y}} |\mathbf{y}\rangle |B^T \mathbf{y}\rangle\,. \tag{137}$$

The following step is to uncompute $|\mathbf{y}\rangle$. When $2\ell + 1 \geq d^\perp$ this uncomputation will not succeed with 100% certainty and one cannot produce exactly the state $|\widetilde{P}(f)\rangle = \sum_{k=0}^{\ell} w_k |\widetilde{P}^{(k)}\rangle$. Instead, the goal is to produce a good approximation to $|\widetilde{P}(f)\rangle$ with high probability. This is because the number of errors ℓ is large enough so that, by starting from a codeword in $C^\perp$ and then flipping

[4] By [11] it is known that the distance of random LDPC codes drawn from various standard ensembles is well-approximated asymptotically by the Gilbert-Varshamov bound. We extrapolate that this is also a good approximation for our ensemble of random LDPC codes with irregular degree distribution.

ℓ bits, the nearest codeword (in Hamming distance) to the resulting string may be a codeword other than the starting codeword. (This is the same reason that $|\widetilde{P}^{(0)}\rangle, \ldots, |\widetilde{P}^{(\ell)}\rangle$ are no longer orthogonal and hence the norm of $|\widetilde{P}(f)\rangle$ is no longer exactly equal to $\|\mathbf{w}\|$.)

If the decoder succeeds on a large fraction of the errors that are in superposition, one simply postselects on success of the decoder and obtains a normalized state which is a good approximation to the (unnormalized) ideal state $|\widetilde{P}(f)\rangle$. The last steps of DQI are to perform a Hadamard transform and then measure in the computational basis, just as in the case of $2\ell + 1 < d^{\perp}$.

7.1 General Expressions for the Expected Number of Satisfied Constraints

Here we derive generalizations of lemmas 6.1 and 6.2 for arbitrary code distances. We restrict our attention to the max-XORSAT case where we are given $B \in \mathbb{F}_2^{m \times n}$ and $\mathbf{v} \in \mathbb{F}_2^m$ and we seek to maximize the objective function $f(\mathbf{x}) = \sum_{i=1}^m (-1)^{\mathbf{b}_i \cdot \mathbf{x} + v_i}$, where $\mathbf{b}_i$ is the i^{th} row of B and v_i is the i^{th} entry of $\mathbf{v}$.

By examining (134) one sees that if $d^{\perp} > 2\ell$ then the $B^T\mathbf{y}$ are all distinct and the norm of $|\widetilde{P}(f)\rangle$ is the norm of $\mathbf{w}$, as shown in Lemma 6.1. More generally, we have the following lemma.

Lemma 7.1. *The squared norm of $|\widetilde{P}(f)\rangle$ is*

$$\langle \widetilde{P}(f) | \widetilde{P}(f) \rangle = \mathbf{w}^{\dagger} M^{(m,\ell)} \mathbf{w}, \tag{138}$$

where $M^{(m,\ell)}$ is the $(\ell+1) \times (\ell+1)$ symmetric matrix defined by

$$M_{k,k'}^{(m,\ell)} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{\substack{\mathbf{y} \in \mathbb{F}_2^m \\ |\mathbf{y}|=k}} \sum_{\substack{\mathbf{y}' \in \mathbb{F}_2^m \\ |\mathbf{y}'|=k'}} (-1)^{(\mathbf{y}+\mathbf{y}')\cdot \mathbf{v}} \delta_{B^T\mathbf{y}, B^T\mathbf{y}'}, \tag{139}$$

for $0 \leq k, k' \leq \ell$.

Proof. This is immediate from (134) and the fact that the $|B^T\mathbf{y}\rangle$ are computational basis states. □

Note that if $d^{\perp} > 2\ell$, then $M_{k,k'}^{(m,\ell)} = \delta_{k,k'}$, in agreement with Lemma 6.1.

Lemma 7.2. *Let $|P(f)\rangle$ be the Hadamard transform of the state $|\widetilde{P}(f)\rangle$ defined in (134). Let $\langle f \rangle$ be the expected objective value for the symbol string obtained upon measuring the DQI state $|P(f)\rangle$ in the computational basis. If the weights w_k are such that $|P(f)\rangle$ is normalized, then*

$$\langle f \rangle = \mathbf{w}^{\dagger} \bar{A}^{(m,\ell)} \mathbf{w} \tag{140}$$

where $\bar{A}^{(m,\ell)}$ is the $(\ell+1) \times (\ell+1)$ symmetric matrix defined by

$$\bar{A}_{k,k'}^{(m,\ell)} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S_{k,k'}^{(i)}} (-1)^{v_i + \mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \tag{141}$$

for $0 \leq k, k' \leq \ell$, and

$$S_{k,k'}^{(i)} = \{(\mathbf{y}, \mathbf{y}') \in \mathbb{F}_2^m \times \mathbb{F}_2^m : |\mathbf{y}| = k, |\mathbf{y}'| = k', B^T(\mathbf{y} + \mathbf{y}' + \mathbf{e}_i) = \mathbf{0}\}. \tag{142}$$

for one-hot vectors $\mathbf{e}_1, \ldots, \mathbf{e}_m$ in $\mathbb{F}_2^m$.

Proof. The expected value of $f(\mathbf{x})$ in state $|P(f)\rangle$ is

$$\langle f \rangle = \langle P(f)| \, H_f \, |P(f)\rangle \,, \tag{143}$$

where

$$H_f = \sum_{i=1}^{m} (-1)^{v_i} \prod_{j:B_{ij}=1} Z_j \tag{144}$$

and Z_j denotes the Pauli Z operator acting on the j^{th} qubit. Recalling that conjugation by Hadamard interchanges Pauli X with Z, we have

$$\begin{aligned}
\langle f \rangle &= \sum_{i=1}^{m} (-1)^{v_i} \langle P(f)| \prod_{j:B_{ij}=1} Z_j \, |P(f)\rangle & (145)\\
&= \sum_{i=1}^{m} (-1)^{v_i} \langle \widetilde{P}(f)| \prod_{j:B_{ij}=1} X_j \, |\widetilde{P}(f)\rangle & (146)\\
&= \sum_{k,k'=0}^{\ell} \frac{w_k w_{k'}}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{|\mathbf{y}|=k} \sum_{|\mathbf{y}'|=k'} (-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \sum_{i=1}^{m} (-1)^{v_i} \langle B^T\mathbf{y}| \prod_{j:B_{ij}=1} X_j \, |B^T\mathbf{y}'\rangle \,, & (147)
\end{aligned}$$

where in the last line we have plugged in the definition of $|\widetilde{P}(f)\rangle$. We can rewrite this quadratic form as

$$\langle f \rangle = \sum_{k,k'=0}^{\ell} w_k \; w_{k'} \; \bar{A}^{(m,\ell)}_{k,k'}, \tag{148}$$

where

$$\bar{A}^{(m,\ell)}_{k,k'} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{|\mathbf{y}|=k} \sum_{|\mathbf{y}'|=k'} (-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \sum_{i=1}^{m} (-1)^{v_i} \langle B^T\mathbf{y}| \prod_{j:B_{ij}=1} X_j \, |B^T\mathbf{y}'\rangle \,. \tag{149}$$

For the one-hot vectors $\mathbf{e}_1, \ldots, \mathbf{e}_m \in \mathbb{F}_2^m$ we have

$$\prod_{j:B_{ij}=1} X_j \, |B^T\mathbf{y}'\rangle = |B^T(\mathbf{y}' + \mathbf{e}_i)\rangle \,. \tag{150}$$

Substituting this into (149) yields

$$\bar{A}^{(m,\ell)}_{k,k'} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{|\mathbf{y}|=k} \sum_{|\mathbf{y}'|=k'} (-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \sum_{i=1}^{m} (-1)^{v_i} \langle B^T\mathbf{y}|B^T(\mathbf{y}' + \mathbf{e}_i)\rangle. \tag{151}$$

We next note that $\langle B^T\mathbf{y}|B^T(\mathbf{y}' + \mathbf{e}_i)\rangle$ equals one when $B^T\mathbf{y} = B^T(\mathbf{y}' + \mathbf{e}_i)$ and zero otherwise. The condition $B^T\mathbf{y} = B^T(\mathbf{y}' + \mathbf{e}_i)$ is equivalent to $B^T(\mathbf{y} + \mathbf{y}' + \mathbf{e}_i) = \mathbf{0}$. Hence,

$$\bar{A}^{(m,\ell)}_{k,k'} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}')\in S^{(i)}_{k,k'}} (-1)^{v_i + \mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \tag{152}$$

where

$$S^{(i)}_{k,k'} = \{(\mathbf{y}, \mathbf{y}') \in \mathbb{F}_2^m \times \mathbb{F}_2^m : |\mathbf{y}| = k, |\mathbf{y}'| = k', B^T(\mathbf{y} + \mathbf{y}' + \mathbf{e}_i) = \mathbf{0}\}. \tag{153}$$

□

Note once again that if $d^\perp > 2\ell + 1$ then $\bar{A}^{(m,\ell)}$ simplifies to $A^{(m,\ell,0)}$ defined in (64). It is also easy to verify that $\bar{A}^{(m,\ell)}$ can be rewritten in terms of M as

$$\bar{A}^{(m,\ell)}_{k,k'} = \sqrt{k'(m-k'+1)} M^{(m,\ell)}_{k,k'-1} + \sqrt{(k'+1)(m-k')} M^{(m,\ell)}_{k,k'+1}, \tag{154}$$

where we define $M^{(m,\ell)}_{k,k'}$ according to formula (139), even if k or k' exceed $\ell + 1$.

7.2 Average-case v

Any choice of $\mathbf{w} \in \mathbb{C}^{\ell+1}$ defines a corresponding state $|P(f)\rangle = H^{\otimes n} |\widetilde{P}(f)\rangle$ via (134). (In general this will not be normalized.)

As shown in the previous subsection the general expression for the expected value of f achieved by measuring the ideal normalized DQI state $|P(f)\rangle / \| |P(f)\rangle \|$ in the computational basis, which holds even when $2\ell + 1 \geq d^\perp$, is

$$\langle f \rangle = \frac{\mathbf{w}^\dagger \bar{A}^{(m,\ell)} \mathbf{w}}{\mathbf{w}^\dagger M^{(m,\ell)} \mathbf{w}}. \tag{155}$$

In the case $2\ell + 1 \geq d^\perp$ it is impossible to precisely obtain the ideal DQI state. This is because the number of errors ℓ is large enough so that, by starting from a codeword in $C^\perp$ and then flipping ℓ bits, the nearest codeword (in Hamming distance) to the resulting string may be a codeword other than the starting codeword. Therefore, in this section, we analyze $\langle f \rangle$ in the presence of a nonzero rate of decoding failures. Specifically, we consider the case that $\mathbf{v}$ is chosen uniformly at random and calculate $\mathbb{E}_{\mathbf{v}} \langle f \rangle$, as this averaging simplifies the analysis substantially, mainly due to the following fact.

Lemma 7.3. *Let $\bar{A}^{(m,\ell)}$ be as defined in (141). Suppose $\mathbf{v}$ is chosen uniformly at random from $\mathbb{F}_2^m$. Then $\mathbb{E}_{\mathbf{v}} \bar{A}^{(m,\ell)} = A^{(m,\ell,0)}$, where $A^{(m,\ell,0)}$ is as defined in* (64)*.*

Proof. Recall that

$$\bar{A}^{(m,\ell)}_{k,k'} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S^{(i)}_{k,k'}} (-1)^{v_i + \mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \tag{156}$$

for $0 \leq k, k' \leq \ell$, and

$$S^{(i)}_{k,k'} = \{(\mathbf{y},\mathbf{y}') \in \mathbb{F}_2^m \times \mathbb{F}_2^m : |\mathbf{y}| = k, |\mathbf{y}'| = k', B^T(\mathbf{y}+\mathbf{y}'+\mathbf{e}_i) = \mathbf{0}\}. \tag{157}$$

We can express $S^{(i)}_{k,k'}$ as the union of two disjoint pieces

$$S^{(i)}_{k,k'} = S^{(i,0)}_{k,k'} \cup S^{(i,1)}_{k,k'} \tag{158}$$

$$S^{(i,0)}_{k,k'} = \{(\mathbf{y},\mathbf{y}') \in S^{(i)}_{k,k'} : \mathbf{y}+\mathbf{y}'+\mathbf{e}_i = \mathbf{0}\} \tag{159}$$

$$S^{(i,1)}_{k,k'} = \{(\mathbf{y},\mathbf{y}') \in S^{(i)}_{k,k'} : \mathbf{y}+\mathbf{y}'+\mathbf{e}_i \neq \mathbf{0}\}. \tag{160}$$

We can then write $\bar{A}^{(m,\ell)}$ as the corresponding sum of two contributions

$$\begin{aligned} \bar{A}^{(m,\ell)}_{k,k'} &= \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S^{(i,0)}_{k,k'}} (-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}'+\mathbf{e}_i)} \\ &+ \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S^{(i,1)}_{k,k'}} (-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}'+\mathbf{e}_i)}. \end{aligned} \tag{161}$$

We next average over $\mathbf{v}$. By the definition of $S_{k,k'}^{(i,0)}$, all terms in the sum have $\mathbf{y} + \mathbf{y}' + \mathbf{e}_i = \mathbf{0}$ and therefore the first term is independent of $\mathbf{v}$. This renders the averaging over $\mathbf{v}$ trivial for the first term. By the definition of $S_{k,k'}^{(i,1)}$, the second term contains exclusively contributions where $\mathbf{y} + \mathbf{y}' + \mathbf{e}_i \neq \mathbf{0}$. By the identity $\frac{1}{2^m} \sum_{\mathbf{v} \in \mathbb{F}_2^m} (-1)^{\mathbf{v} \cdot \mathbf{z}} = \delta_{\mathbf{z},\mathbf{0}}$ we see that the second term makes zero contribution to the average. Thus we obtain

$$\mathbb{E}_{\mathbf{v}} \bar{A}_{k,k'}^{(m,\ell)} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S_{k,k'}^{(i,0)}} (-1)^{\mathbf{v} \cdot (\mathbf{y} + \mathbf{y}' + \mathbf{e}_i)}. \tag{162}$$

We next observe that $S_{k,k'}^{(i,0)}$ contains the terms where $\mathbf{y} + \mathbf{y}' + \mathbf{e}_i = \mathbf{0}$ which are exactly the terms that contribute at $2\ell + 1 < d^\perp$, whereas $S_{k,k'}^{(i,1)}$ contains the terms where $\mathbf{y} + \mathbf{y}' + \mathbf{e}_i \in C^\perp \setminus \{\mathbf{0}\}$ which, prior to averaging, are the new contribution arising when ℓ exceeds this bound. Consequently, $\mathbb{E}_{\mathbf{v}} \bar{A}_{k,k'}^{(m,\ell)}$ is exactly equal to $A_{k,k'}^{(m,\ell,0)}$, as defined in (64). (One can also verify this by direct calculation.) That is, we have obtained

$$\mathbb{E}_{\mathbf{v}} \bar{A}_{k,k'}^{(m,\ell)} = A_{k,k'}^{(m,\ell,0)}, \tag{163}$$

which completes the proof of the lemma. □

7.3 Imperfect decoding

A deterministic decoder partitions the set of errors $\mathbb{F}_2^m = \mathcal{D} \cup \mathcal{F}$ into the set $\mathcal{D}$ of errors $\mathbf{y}$ correctly identified by the decoder based on the syndrome $B^T\mathbf{y}$ and the set $\mathcal{F}$ of errors misidentified. The Hamming shell $\mathcal{E}_k$ of radius k is analogously partitioned $\mathcal{E}_k = \mathcal{D}_k \cup \mathcal{F}_k$. We will quantify decoder's failure rate using $\varepsilon_k := |\mathcal{F}_k| / \binom{m}{k}$ and $\varepsilon := \max_{0 \le k \le \ell} \varepsilon_k$.

The quantum state of the error and syndrome registers after the error uncomputation step of the DQI algorithm using an imperfect decoder is

$$\sum_{k=0}^{\ell} \frac{w_k}{\sqrt{\binom{m}{k}}} \left(\sum_{\substack{\mathbf{y} \in \mathcal{D}_k \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v} \cdot \mathbf{y}} |\mathbf{0}\rangle |B^T\mathbf{y}\rangle + \sum_{\substack{\mathbf{y} \in \mathcal{F}_k \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v} \cdot \mathbf{y}} |\mathbf{y} \oplus \mathbf{y}'\rangle |B^T\mathbf{y}\rangle \right) \tag{164}$$

where $\mathbf{y} \neq \mathbf{y}'$. After uncomputing the error register, we postselect on the register being $|\mathbf{0}\rangle$. If the postselection is successful, then the syndrome register is in the quantum state proportional to the following unnormalized state vector

$$|\widetilde{P}_{\mathcal{D}}(f)\rangle := \sum_{k=0}^{\ell} \frac{w_k}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y} \in \mathcal{D}_k \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v} \cdot \mathbf{y}} |B^T\mathbf{y}\rangle. \tag{165}$$

The following theorem describes the effect that decoding failure rate ε has on the approximation ratio achieved by DQI. We do not assume that $2\ell + 1 < d^\perp$.

Theorem 7.1. *Given $B \in \mathbb{F}_2^{m \times n}$ and $\mathbf{v} \in \mathbb{F}_2^m$, let f be the objective function $f(\mathbf{x}) = \sum_{i=1}^m (-1)^{v_i + \mathbf{b}_i \cdot \mathbf{x}}$. Let P be any degree-ℓ polynomial and let $P(f) = \sum_{k=0}^{\ell} w_k P^{(k)}(f_1, \ldots, f_m) / \sqrt{2^n \binom{m}{k}}$ be the decomposition of $P(f)$ as a linear combination of elementary symmetric polynomials. Let $|P_{\mathcal{D}}(f)\rangle$ denote* *a DQI state prepared using an imperfect decoder that misidentifies $\varepsilon_k\binom{m}{k}$ errors of Hamming weight k and let $\langle f\rangle$ be the expected objective value for the symbol string resulting from the measurement of this state in the computational basis. If $\mathbf{v}\in\mathbb{F}_2^m$ is chosen uniformly at random, then*

$$\mathbb{E}_{\mathbf{v}}\langle f\rangle \geq \frac{\mathbf{w}^\dagger\left[A^{(m,\ell,0)}-2\varepsilon(m+1)\right]\mathbf{w}}{\sum_{k=0}^{\ell} w_k^2(1-\varepsilon_k)} \geq \left(\frac{\mathbf{w}^\dagger A^{(m,\ell,0)}\mathbf{w}}{\mathbf{w}^\dagger\mathbf{w}}-2\varepsilon(m+1)\right), \tag{166}$$

where $\varepsilon=\max_{0\leq k\leq\ell}\varepsilon_k$ and $A^{(m,\ell,0)}$ is the tridiagonal matrix defined in equation (64). Moreover, if $\ell\leq m/2$ and one chooses $\mathbf{w}$ to be the principal eigenvector of $A^{(m,\ell,0)}$ then (166) yields the following lower bound in the limit of large ℓ and m with the ratio $\mu=\ell/m$ fixed: For a random $\mathbf{v}\in\mathbb{F}_2^m$,

$$\lim_{\substack{m\to\infty\\ \ell/m=\mu}}\frac{\mathbb{E}_{\mathbf{v}}\langle f\rangle}{m} \geq 2\left(\sqrt{\frac{\ell}{m}\left(1-\frac{\ell}{m}\right)}-\varepsilon\right). \tag{167}$$

Before proving Theorem 7.1, we generalize lemma 7.1 and lemma 7.2 to the state $|P_{\mathcal{D}}(f)\rangle$ prepared by DQI with an imperfect decoder upon successful postselection.

Lemma 7.4. *The squared norm of $|\widetilde{P}_{\mathcal{D}}(f)\rangle$ is*

$$\langle\widetilde{P}_{\mathcal{D}}(f)|\widetilde{P}_{\mathcal{D}}(f)\rangle=\sum_{k=0}^{\ell}w_k^2(1-\varepsilon_k)\leq\mathbf{w}^\dagger\mathbf{w}. \tag{168}$$

Proof. We first observe that all syndromes $|B^T\mathbf{y}\rangle$ for $\mathbf{y}\in\mathcal{D}$ are necessarily distinct bit strings and thus orthogonal quantum states. This follows from the fact that if a deterministic decoder correctly recovers $\mathbf{y}$ from the syndrome $B^T\mathbf{y}$, then it must fail on all $\mathbf{y}'\neq\mathbf{y}$ with $B^T\mathbf{y}'=B^T\mathbf{y}$. The squared norm of $|\widetilde{P}_{\mathcal{D}}(f)\rangle$ is thus

$$\langle\widetilde{P}_{\mathcal{D}}(f)|\widetilde{P}_{\mathcal{D}}(f)\rangle=\sum_{k=0}^{\ell}\frac{w_k^2}{\binom{m}{k}}|\mathcal{D}_k|=\sum_{k=0}^{\ell}w_k^2(1-\varepsilon_k), \tag{169}$$

as claimed. □

Lemma 7.5. *For $B\in\mathbb{F}_2^{m\times n}$ and $\mathbf{v}\in\mathbb{F}_2^m$, let f be the objective function $f(\mathbf{x})=\sum_{i=1}^m(-1)^{v_i+\mathbf{b}_i\cdot\mathbf{x}}$. Let P be any degree-ℓ polynomial and let $P(f)=\sum_{k=0}^{\ell}w_kP^{(k)}(f_1,\ldots,f_m)/\sqrt{2^n\binom{m}{k}}$ be the decomposition of $P(f)$ as a linear combination of elementary symmetric polynomials. Let $\langle f\rangle$ be the expected objective value for the symbol string obtained upon measuring the imperfect DQI state $|P_{\mathcal{D}}(f)\rangle$ in the computational basis. If the weights w_k are such that $|P_{\mathcal{D}}(f)\rangle$ is normalized, then*

$$\langle f\rangle=\mathbf{w}^\dagger\bar{A}^{(m,\ell,\mathcal{D})}\mathbf{w} \tag{170}$$

where $\bar{A}^{(m,\ell,\mathcal{D})}$ is the $(\ell+1)\times(\ell+1)$ symmetric matrix defined by

$$\bar{A}_{k,k'}^{(m,\ell,\mathcal{D})}=\frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}}\sum_{i=1}^{m}\sum_{(\mathbf{y},\mathbf{y}')\in S_{k,k'}^{(i,\mathcal{D})}}(-1)^{v_i+\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \tag{171}$$

for $0 \le k, k' \le \ell$, *and*

$$S_{k,k'}^{(i,\mathcal{D})} = \{(\mathbf{y}, \mathbf{y}') \in \mathcal{D}_k \times \mathcal{D}_{k'} : B^T(\mathbf{y} + \mathbf{y}' + \mathbf{e}_i) = \mathbf{0}\} \tag{172}$$

for one-hot vectors $\mathbf{e}_1, \dots, \mathbf{e}_m$ *in* $\mathbb{F}_2^m$.

Proof. The proof is obtained from the proof of Lemma 7.2 by replacing each sum ranging over $\mathcal{E}_k$ with a sum ranging over $\mathcal{D}_k$ and the ideal DQI state $|P(f)\rangle$ with the imperfect DQI state $|P_{\mathcal{D}}(f)\rangle$. □

7.4 Average-case vith imperfect decoding

The expected objective value achieved by sampling from a normalized DQI state $|P_{\mathcal{D}}(f)\rangle$ is

$$\langle f \rangle = \frac{\mathbf{w}^\dagger \bar{A}^{(m,\ell,\mathcal{D})} \mathbf{w}}{\sum_{k=0}^{\ell} w_k^2 (1 - \varepsilon_k)} \ge \frac{\mathbf{w}^\dagger \bar{A}^{(m,\ell,\mathcal{D})} \mathbf{w}}{\mathbf{w}^\dagger \mathbf{w}}. \tag{173}$$

Next, we find the expectation of $\bar{A}^{(m,\ell,\mathcal{D})}$.

Lemma 7.6. *Let* $\bar{A}^{(m,\ell,\mathcal{D})}$ *be defined as in (171). Suppose* $\mathbf{v}$ *is chosen uniformly at random from* $\mathbb{F}_2^m$. *Then*

$$\mathbb{E}_{\mathbf{v}} \bar{A}^{(m,\ell,\mathcal{D})} = A^{(m,\ell,0)} - E^{(m,\ell,\mathcal{F})} \tag{174}$$

where $A^{(m,\ell,0)}$ *is defined as in* (64) *and*

$$E_{k,k'}^{(m,\ell,\mathcal{F})} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} |T_{k,k'}^{(i,\mathcal{F})}| \tag{175}$$

for $0 \le k, k' \le \ell$ *with*

$$T_{k,k'}^{(i,\mathcal{F})} = \{(\mathbf{y}, \mathbf{y}') \in \mathcal{E}_k \times \mathcal{F}_{k'} \cup \mathcal{F}_k \times \mathcal{E}_{k'} : \mathbf{y} + \mathbf{y}' + \mathbf{e}_i = \mathbf{0}\}. \tag{176}$$

Proof. We can partition $S_{k,k'}^{(i,\mathcal{D})}$ into two disjoint subsets

$$\begin{aligned} S_{k,k'}^{(i,\mathcal{D})} &= S_{k,k'}^{(i,0,\mathcal{D})} \cup S_{k,k'}^{(i,1,\mathcal{D})} && (177) \\ S_{k,k'}^{(i,0,\mathcal{D})} &= \{(\mathbf{y}, \mathbf{y}') \in S_{k,k'}^{(i,\mathcal{D})} : \mathbf{y} + \mathbf{y}' + \mathbf{e}_i = \mathbf{0}\} && (178) \\ S_{k,k'}^{(i,1,\mathcal{D})} &= \{(\mathbf{y}, \mathbf{y}') \in S_{k,k'}^{(i,\mathcal{D})} : \mathbf{y} + \mathbf{y}' + \mathbf{e}_i \ne \mathbf{0}\} && (179) \end{aligned}$$

so that

$$\begin{aligned} \bar{A}_{k,k'}^{(m,\ell,\mathcal{D})} &= \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S_{k,k'}^{(i,0,\mathcal{D})}} (-1)^{v_i + \mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')} \\ &+ \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}} \sum_{i=1}^{m} \sum_{(\mathbf{y},\mathbf{y}') \in S_{k,k'}^{(i,1,\mathcal{D})}} (-1)^{v_i + \mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')}. \end{aligned} \tag{180}$$

We next observe that when we average over $\mathbf{v}$, the second term in (180) vanishes, because

$$\begin{aligned}
\mathbb{E}_{\mathbf{v}}\left(\sum_{(\mathbf{y},\mathbf{y}')\in S^{(i,1,\mathcal{D})}_{k,k'}}(-1)^{v_i+\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')}\right) &= \sum_{(\mathbf{y},\mathbf{y}')\in S^{(i,1,\mathcal{D})}_{k,k'}} \mathbb{E}_{\mathbf{v}}\left((-1)^{v_i+\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')}\right) && (181)\\
&= \sum_{(\mathbf{y},\mathbf{y}')\in S^{(i,1,\mathcal{D})}_{k,k'}} \frac{1}{2^m}\sum_{\mathbf{v}\in\mathbb{F}_2^m}(-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}'+\mathbf{e}_i)} && (182)\\
&= \sum_{(\mathbf{y},\mathbf{y}')\in S^{(i,1,\mathcal{D})}_{k,k'}} \delta_{\mathbf{y}+\mathbf{y}'+\mathbf{e}_i,\mathbf{0}} && (183)\\
&= 0, && (184)
\end{aligned}$$

where the last equality follows from the definition of $S^{(i,1,\mathcal{D})}_{k,k'}$. Hence,

$$\mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell,\mathcal{D})}_{k,k'} = \mathbb{E}_{\mathbf{v}}\frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}}\sum_{i=1}^{m}\sum_{(\mathbf{y},\mathbf{y}')\in S^{(i,0,\mathcal{D})}_{k,k'}}(-1)^{v_i+\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}')}. \tag{185}$$

We next examine (162) and observe that $\mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell,\mathcal{D})}_{k,k'}$ and $\mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell)}_{k,k'}$ differ only by replacing the summation over $S^{(i,0,\mathcal{D})}_{k,k'}$ with a summation over $S^{(i,0)}_{k,k'}$. Recalling the definition of $S^{(i,0)}_{k,k'}$ from equation (159), we observe that $S^{(i,0)}_{k,k'} = S^{(i,0,\mathcal{D})}_{k,k'} \cup T^{(i,\mathcal{F})}_{k,k'}$. Hence,

$$\mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell)}_{k,k'} = \mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell,\mathcal{D})}_{k,k'} + \mathbb{E}_{\mathbf{v}}\frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}}\sum_{i=1}^{m}\sum_{(\mathbf{y},\mathbf{y}')\in T^{(i,\mathcal{F})}}(-1)^{\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}'+\mathbf{e}_i)}. \tag{186}$$

By the definition of $T^{(i,\mathcal{F})}$ one sees that all terms in the sum have $\mathbf{v}\cdot(\mathbf{y}+\mathbf{y}'+\mathbf{e}_i)=\mathbf{0}$ and hence

$$\mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell)}_{k,k'} = \mathbb{E}_{\mathbf{v}}\bar{A}^{(m,\ell,\mathcal{D})}_{k,k'} + \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}}\sum_{i=1}^{m}|T^{(i,\mathcal{F})}|. \tag{187}$$

By Lemma 7.3, this simplifies to

$$\bar{A}^{(m,\ell,0)}_{k,k'} = \mathbb{E}_{\mathbf{v}}A^{(m,\ell,\mathcal{D})}_{k,k'} + \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k'}}}\sum_{i=1}^{m}|T^{(i,\mathcal{F})}|. \tag{188}$$

By (175) we can rewrite this as

$$\bar{A}^{(m,\ell,0)}_{k,k'} = \mathbb{E}_{\mathbf{v}}A^{(m,\ell,\mathcal{D})}_{k,k'} + E^{(m,\ell,\mathcal{F})}_{k,k'}, \tag{189}$$

which rearranges to (174) completing the proof. □

The last remaining ingredient for the proof of theorem 7.1 is the following upper bound on the error term identified in Lemma 7.6.

Lemma 7.7. *Let $\varepsilon_k = |\mathcal{F}_k|/\binom{m}{k}$ and $\varepsilon = \max_{0 \leq k \leq \ell} \varepsilon_k$. Then,*

$$\|E^{(m,\ell,\mathcal{F})}\| \leq 2\varepsilon(m+1). \tag{190}$$

Proof. We note that $E^{(m,\ell,\mathcal{F})}_{k,k'} = 0$ unless $k = k' \pm 1$, so $E^{(m,\ell,\mathcal{F})}$ is tridiagonal with zeros on the diagonal. By (175) we have

$$E^{(m,\ell,\mathcal{F})}_{k,k+1} = \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k+1}}} \sum_{i=1}^{m} |T^{(i,\mathcal{F})}_{k,k+1}|. \tag{191}$$

Note that if $(\mathbf{y}, \mathbf{y}') \in T^{(i,\mathcal{F})}_{k,k+1}$, then $\mathbf{y}_i = 0$ and $\mathbf{y}'_i = 1$. Therefore, every pair $(\mathbf{y}, \mathbf{y}')$ with $\mathbf{y} \in \mathcal{F}_k$ contributes to at most $m - k$ out of the m terms in the sum above. Similarly, every pair $(\mathbf{y}, \mathbf{y}')$ with $\mathbf{y}' \in \mathcal{F}_{k+1}$ contributes to at most $k+1$ terms. Consequently,

$$\begin{aligned} E^{(m,\ell,\mathcal{F})}_{k,k+1} &\leq \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k+1}}} \left[(m-k)|\mathcal{F}_k| + (k+1)|\mathcal{F}_{k+1}|\right] && (192)\\ &= \frac{1}{\sqrt{\binom{m}{k}\binom{m}{k+1}}} \left[(m-k)\varepsilon_k \binom{m}{k} + (k+1)\varepsilon_{k+1}\binom{m}{k+1}\right] && (193)\\ &\leq (\varepsilon_k + \varepsilon_{k+1})\sqrt{(k+1)(m-k)} && (194)\\ &\leq \varepsilon(m+1) && (195) \end{aligned}$$

and by Gershgorin's circle theorem

$$\|E^{(m,\ell,\mathcal{F})}\| \leq 2\varepsilon(m+1) \tag{196}$$

completing the proof of the Lemma. □

7.5 Proof of Theorem 7.1

Finally, we prove Theorem 7.1.

Proof. Equation (173) and Lemma 7.6 imply that

$$\mathbb{E}_{\mathbf{v}}\langle f \rangle \geq \frac{\mathbf{w}^\dagger \left[A^{(m,\ell,0)} - E^{(m,\ell,\mathcal{F})}\right] \mathbf{w}}{\sum_{k=0}^{\ell} w_k^2 (1 - \varepsilon_k)} \geq \left(\frac{\mathbf{w}^\dagger A^{(m,\ell,0)} \mathbf{w}}{\mathbf{w}^\dagger \mathbf{w}} - \frac{\mathbf{w}^\dagger E^{(m,\ell,\mathcal{F})} \mathbf{w}}{\mathbf{w}^\dagger \mathbf{w}} \right) \tag{197}$$

which in light of Lemma 7.7 becomes

$$\mathbb{E}_{\mathbf{v}}\langle f \rangle \geq \frac{\mathbf{w}^\dagger \left[A^{(m,\ell,0)} - 2\varepsilon(m+1)\right] \mathbf{w}}{\sum_{k=0}^{\ell} w_k^2 (1 - \varepsilon_k)} \geq \left(\frac{\mathbf{w}^\dagger A^{(m,\ell,0)} \mathbf{w}}{\mathbf{w}^\dagger \mathbf{w}} - 2\varepsilon(m+1) \right). \tag{198}$$

This proves the first part of Theorem 7.1. The second part, equation (167), follows by substituting the asymptotic formula for the leading eigenvalue of $A^{(m,\ell,0)}$ derived in Lemma 6.3. □

Remark 7.1. In practice, we find that decoding success probability decreases as the number of errors increases. Therefore, we can use the empirical failure probability of a classical decoder on uniformly random errors of Hamming weight ℓ as an upper bound on ε, which would be the failure probability of a decoder applied to a distribution over error weights zero to ℓ determined by the vector $\mathbf{w}$.

8 Other Optimization Algorithms

In this section we survey algorithms against which DQI can be compared. In §8.1, we consider local search heuristics such as simulated annealing and greedy optimization. In §8.2 we comment in more detail on the convergence of simulated annealing when one adds more sweeps. In §8.3 we analyze Prange's algorithm in which one discards all but n of the m constraints on the n variables and then solves the resulting linear system. In §8.4, we summarize the AdvRand algorithm of [15]. In §8.5, we discuss the Quantum Approximate Optimization Algorithm (QAOA). Though necessarily not exhaustive, we believe these algorithms constitute a thorough set of general-purpose optimization strategies to benchmark DQI against on random sparse max-XORSAT. Lastly, we analyze the two classes of algebraic algorithms that pose the most plausible challenge to DQI on our OPI problem and find that they are not successful in our parameter regime. Specifically, in §8.6 we consider list-recovery algorithms, and in §8.7, we summarize the lattice-based heuristic of [5].

8.1 General Local Search Heuristics

In local search methods, one makes a sequence of local moves in the search space, such as by flipping an individual bit, and preferentially accepts moves that improve the objective function. This class of heuristics includes simulated annealing, parallel tempering, TABU search, greedy algorithms, and some quantum-inspired optimization methods. For simplicity, we will restrict our analysis to max-XORSAT and consider only local search algorithms in which at each move a single variable among $x_1, \ldots, x_n$ is flipped between 1 and 0. The generalization to single-symbol-flip updates applied to max-LINSAT is straightforward.

Let C_i be the set of constraints containing the variable x_i. In Gallager's ensemble $|C_i| = D$ for all i. For assignment $\mathbf{x} \in \mathbb{F}_2^n$ let $S_i(\mathbf{x})$ be the number of constraints in C_i that are satisfied. Consider an assignment $\mathbf{x}$ such that a fraction ϕ of the m constraints are satisfied. Then, modeling $C_1, \ldots, C_m$ as random subsets, we have

$$\Pr\left[S_i(\mathbf{x}) = s\right] = \binom{D}{s} \phi^s (1-\phi)^{D-s}. \tag{199}$$

Next, consider the move $\mathbf{x} \to \mathbf{x}'$ induced by flipping bit i. This causes all satisfied constraints in C_i to become unsatisfied and vice-versa. Hence, such a move induces the change

$$S_i(\mathbf{x}') = D - S_i(\mathbf{x}). \tag{200}$$

This will be an improvement in the objective value if and only if $S_i(\mathbf{x}) < D/2$. According to (199) the probability $P_i^{(+)}$ that the flip is an improvement is

$$P_i^{(+)} = \sum_{s=0}^{\lfloor \frac{D-1}{2} \rfloor} \binom{D}{s} \phi^s (1-\phi)^{D-s}. \tag{201}$$

By Hoeffding's inequality, we have that

$$P_i^{(+)} \leq \exp\left(-2(\phi - 1/2)^2 D\right). \tag{202}$$

For large ϕ the probability $P_i^{(+)}$ becomes very small. When the probability of making an upward move becomes extremely small, the local optimization algorithm will no longer be able to achieve further improvement in any reasonable number of steps. If the algorithm takes a total of N steps then, by the union bound and (202), the probability of finding such a move in any of the steps is upper bounded by

$$P_{\text{success}}^{\text{bound}}(\phi) = N \exp\left(-2(\phi - 1/2)^2 D\right). \tag{203}$$

From this we can see what is the highest value of ϕ for which the success probability remains significant. For example, if we set $P_{\text{success}}^{\text{bound}}(\phi_{\max}) = 1/2$ and solve for $\phi_{\max}$ the result is

$$\phi_{\max} = \frac{1}{2} + \sqrt{\frac{\log N + \log 2}{2D}} \tag{204}$$

This analysis is only approximate. Indeed, one can see that the model cannot hold indefinitely as N becomes larger, because eventually $\phi_{\max}$ becomes limited by the true optimum. Let us now compare it with computer experiment. In Fig. 3 we show best fits to the satisfaction fraction versus D with k/D fixed at $1/10$, $1/2$, and $9/10$ for simulated annealing and greedy descent (which is equivalent to zero-temperature simulated annealing). We find that for each choice of k/D, $\phi_{\max} - 1/2$ fits well to $cD^{-\nu}$ for some constant c and some power ν but that the power ν is slightly smaller than $1/2$. We believe this to be a finite-size effect, as it has been shown [29] that for an ensemble of random degree-D max-k-XORSAT instances differing only slightly from the Gallager ensemble, the exact optimum scales like $\frac{1}{2} + \frac{P_k}{\sqrt{D}}$. Thus we find that the functional form

$$\phi_{\max} = \frac{1}{2} + \frac{c}{\sqrt{D}} \tag{205}$$

predicted by this argument is in reasonably good agreement with our experimental observations.

8.2 Convergence of Simulated Annealing

We next investigate whether the N-scaling predicted by the argument in §8.1 yields a good model of the behavior of simulated annealing. This scaling cannot persist indefinitely because eventually $\phi_{\max}$ is limited by the true optimum. In fact, we find that, in contrast to the D-scaling, the N-scaling suggested by the above argument is not corroborated by experimental evidence. Instead, empirical N-scaling fits much better to power-law convergence, as illustrated in Fig. 4.

Although we do not have a strong theoretical handle on the N-scaling, we can nevertheless exploit the simple observation that increasing N has diminishing returns. Thus, in all empirical analysis relating to simulated annealing, by choosing N large one can ensure data points are on the relatively flat tail of the convergence curve and thus relatively insensitive to the specific choice of N. This renders the trends noted in figures 3, 6, 10, and 12 relatively robust to choice of sweep count. In each of these plots we also kept the number of sweeps fixed for all data points plotted in order to minimize the effect of this as a confounding factor.

We next consider the problem of comparing the performance of DQI against simulated annealing. This task is rendered complicated by the fact that the number of clauses that simulated annealing is able to satisfy depends on how long one is willing to run the anneal. In Fig. 5, we plot the convergence of simulated annealing as a function of number of sweeps for the instance defined in §9 and compare against the satisfaction fraction achieved by DQI+BP. We use the same data as in the lower panel of Fig. 4 but using a linear instead of logarithmic scale on the horizontal axis and zooming in on the region where the number of sweeps is at least 10^5.

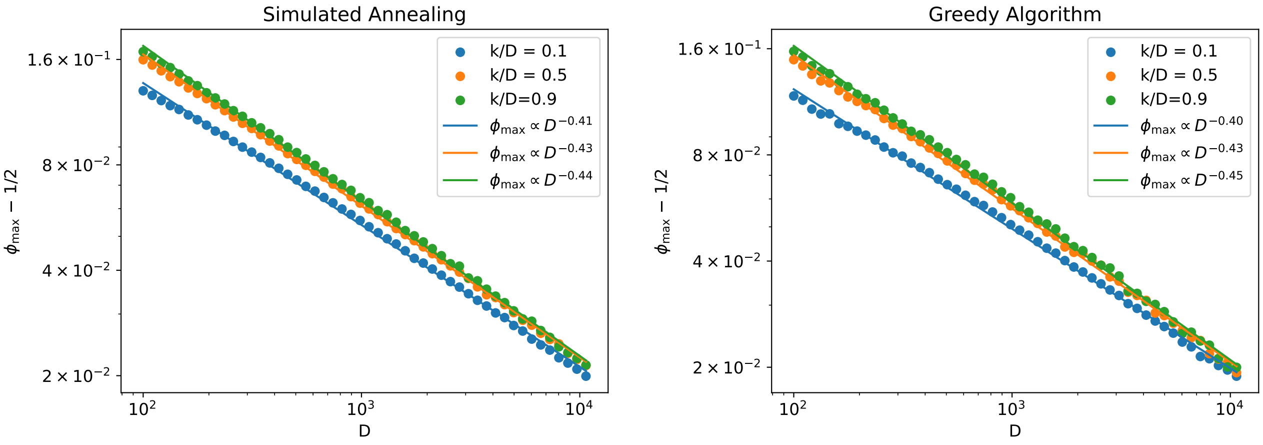

Figure 3: The approximation achieved by simulated annealing (left) and greedy optimization (right) in our computer experiments at $n = 10^5$, on a log-log scale, where each constraint contains k variables and each variable is contained in D constraints. The lines illustrates linear least-squares curve fits to the log-log data. The ensemble of max-k-XORSAT instances is formally defined in Appendix B. In these anneals we use $5,000$ sweeps with single-bit updates, and linearly increasing inverse temperature β.

In all anneals, we start with inverse temperature $\beta = 0$ and then linearly increase with each sweep to a final value of $\beta = 5$. Our results suggests this is an effective annealing schedule, although we do not claim it is precisely optimal. In each sweep, we cycle through the $n = 31,216$ bits, and for each one consider a move in which the bit is flipped, accepting the move according to the Metropolis criterion. For this instance, our implementation of simulated annealing, which is optimized C++ code, is able to execute approximately 16 sweeps (5×10^5 Metropolis moves) per second, though this varied slightly from run to run likely due to the fact that accepted Metropolis moves incur a larger computational cost than rejected moves in our implementation. Among our 1,079 annealing experiments, the largest number of sweeps we carried out was 7×10^7, and the longest anneal completed after 118 hours of runtime.

The first anneal to exceed the satisfaction fraction of 0.831 guaranteed by Theorem 7.1 for BP+DQI used 6×10^7 sweeps and ran for 73 hours. Since we ran five anneals at each number of sweeps we estimate that running five independent anneals in parallel at 6×10^7 sweeps would yield a collection of approximate optima, the best of which has a nontrivial chance of exceeding 0.831.

8.3 Prange's Algorithm

Consider an instance of max-XORSAT

$$B\mathbf{x} \overset{\max}{=} \mathbf{v} \tag{206}$$

where B is an $m \times n$ matrix over $\mathbb{F}_2$ and $m > n$. The system (206) is therefore overdetermined and we wish to satisfy as many equations as possible. In Prange's algorithm we simply throw away all but n of the linear equations from this system so that it is no longer overdetermined. Then, provided the remaining system is not singular, we can simply solve it, *e.g.* by Gaussian elimination.

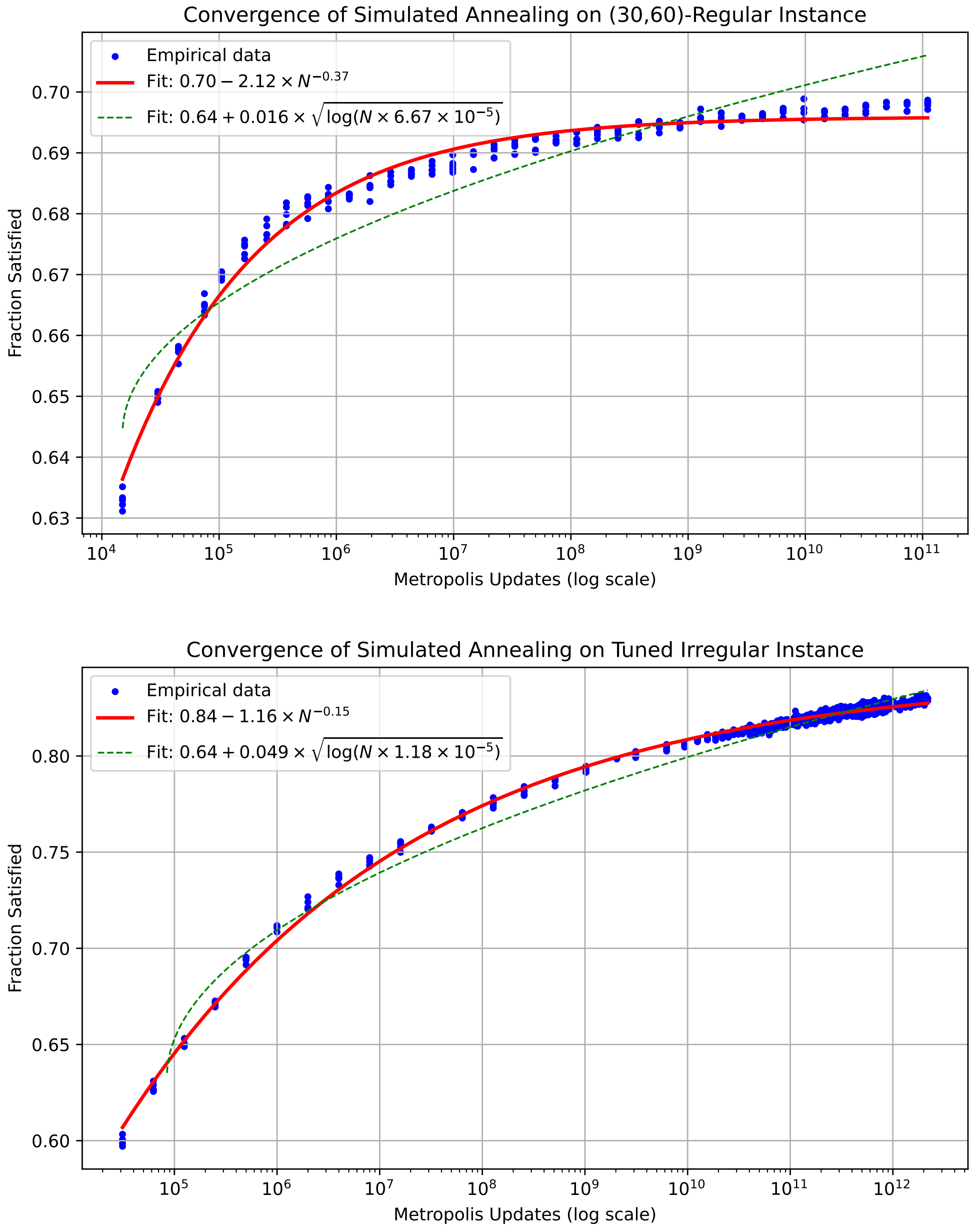

Figure 4: Here we show the dependence of the fraction of constraints satisfied ϕ on the number of Metropolis updates N used in simulated annealing. That is, N is the number of variables times the number of sweeps. We find that the functional form $\phi = a + b\sqrt{c \log N}$ suggested by the heuristic argument of §8.1 fits poorly, but the form $\phi = a - bN^{-c}$ fits reasonably well. On the top panel we consider an instance from Gallager's ensemble with $k = 30$, $D = 60$, and $n = 15,000$. On the bottom we use the irregular instance defined in §9, which has $n = 31,216$. Each data point represents the final outcome of an independent anneal, and in each anneal we vary β linearly from zero to five.

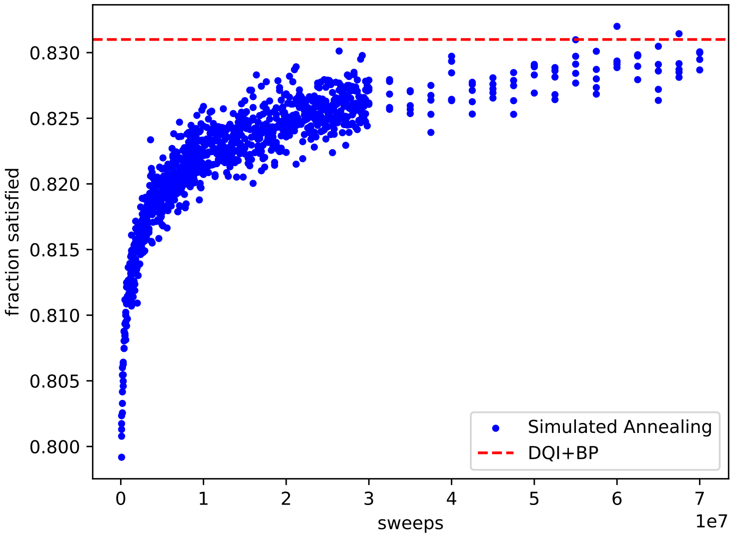

Figure 5: Here we show the satisfaction fraction achieved by simulated annealing as a function of the number of sweeps. For each number of sweeps we run five independent executions of simulated annealing with different pseudorandom seeds. At the right-hand side of the plot we incremented the number of sweeps by larger increments due to computational cost.

We thus obtain a bit string that definitely satisfies n of the original m constraints. Heuristically, one expects the remaining $m-n$ constraints each to be satisfied with probability $1/2$, independently. In other words the number of these $m-n$ constraints satisfied is binomially distributed. One reruns the above steps polynomially many times with different random choices of constraints to solve for. In this manner one can reach a logarithmic number of standard deviations onto the tail of this binomial distribution. Consequently, for max-XORSAT, the number of constraints one can satisfy using Prange's algorithm with polynomially many trials is $n + (m-n)/2 + \widetilde{\mathcal{O}}(\sqrt{m})$.

One can make the procedure more robust by bringing B^T to reduced row-echelon form rather than throwing away columns and hoping that what is left is non-singular. In this case, as long as B^T is full rank, one can find a bit string satisfying n constraints with certainty. From numerical experiments one finds that matrices from Gallager's ensemble often fall just slightly short of full rank. Nevertheless, this procedure, when applied to Gallager's ensemble, matches very closely the behavior predicted by the above argument.

This heuristic generalizes straightforwardly to max-LINSAT. First, consider the case that $|f_i^{-1}(+1)| = r$ for all $i = 1, \ldots, m$. In this case, we throw away all but n constraints, and choose arbitrarily among the r elements in the preimage $f_i^{-1}(+1)$ for each of those that remain. After solving the resulting linear system we will satisfy all of these n constraints and on average we expect to satisfy a fraction r/p of the remaining $n-m$ constraints assuming the f_i are random. Hence, with polynomially many randomized repetitions of this scheme, the number of constraints satisfied will be $n + (m-n)(r/p) + \widetilde{\mathcal{O}}(\sqrt{m})$. That is, the fraction ϕ_{PR} of the m constraints satisfied by a solution found by Prange's algorithm will be

$$\phi_{\mathrm{PR}} = \frac{r}{p} + \left(1 - \frac{r}{p}\right)\frac{n}{m} + \widetilde{\mathcal{O}}(1/\sqrt{m}). \tag{207}$$

If the preimages $f_i^{-1}(+1)$ for $i = 1, \ldots, m$ do not all have the same size, then one should choose the n constraints with smallest preimages as the ones to keep in the first step. The remaining $m - n$ are then those most likely to be satisfied by random chance.

8.4 The AdvRand Algorithm

Prompted by some successes [30] of the Quantum Approximate Optimization Algorithm (QAOA), a simple but interesting algorithm for approximating max-XORSAT was proposed by Barak *et al.* in [15], which the authors named AdvRand. Although designed primarily for the purpose of enabling rigorous average-case performance guarantees, the AdvRand algorithm can also be tried empirically as a heuristic, much like simulated annealing, and compared against DQI.

Given an instance of max-XORSAT with n variables, the AdvRand algorithm works as follows. Select two parameters $R, F \in (0, 1)$. Repeat the following sequence of steps polynomially many times. First assign Rn of the variables uniformly at random. Substitute these choices into the instance, yielding a new instance with $(1 - R)n$ variables. A constraint of degree D will become a constraint of degree $D - r$ if r of the variables it contains have been replaced by randomly chosen values. Thus, in the new instance some of the resulting constraints may have degree one. Assign the variables in such constraints to the values that render these constraints satisfied. If there are remaining unassigned variables, assign them randomly. Lastly, flip each variable independently with probability F.

In [15], formulas are given for R and F that enable guarantees to be proven about worst case performance. Here, we treat R and F as hyperparameters. We set $F = 0$ and exhaustively try all values of Rn from 0 to n, then retain the best solution found. Our results on Gallager's ensemble at $k = 3$ are displayed and compared against simulated annealing and DQI+BP in Fig. 6 for constant n and growing D.

In [15] it was proven that, given max-k-XORSAT instances, AdvRand can in polynomial time find solutions satisfying a fraction $\frac{1}{2} + \frac{e^{-\mathcal{O}(k)}}{\sqrt{D}}$ of the constraints, even in the worst case, provided each variable is contained at most D constraints. Our observed empirical performance in Fig. 6 is in good agreement with this. In [29] it was shown that at large D, for average-case degree-D max-3-XORSAT, the exact optimum concentrates at $\frac{1}{2} + \frac{0.9959}{\sqrt{D}}$, in the limit of large D. Thus, for fixed k, the functional form of the scaling of AdvRand is provably optimal, and the key metric of performance for given k is the specific value of the numerator $e^{-\mathcal{O}(k)}$. For our experiments at $n = 20,000$ with $k = 3$ we empirically observe a value of 0.31.

In the simulated annealing experiments shown in Fig. 6, we vary β linearly from 0 to 3 and apply $5,000$ sweeps through the variables, *i.e.* $5,000n$ Metropolis updates.

8.5 Quantum Approximate Optimization Algorithm

In 2014, Farhi, Goldstone, and Gutmann introduced a new quantum algorithm for optimization that they called the Quantum Approximate Optimization Algorithm (QAOA) [32]. The QAOA algorithm is parameterized by a number of rounds, p. Allowing additional rounds can only improve the approximate optima found by QAOA, but this also makes the algorithm harder to analyze theoretically. The largest p for which QAOA's performance has been analyzed on max-3-XORSAT is $p = 14$, which was achieved in [31] using nontrivial tensor network techniques, which apply to all D-regular max-3-XORSAT instances whose hypergraphs have girth greater than $2p+1$. In [31] it was found that the fraction of satisfied clauses for every D-regular large-girth hypergraph, or

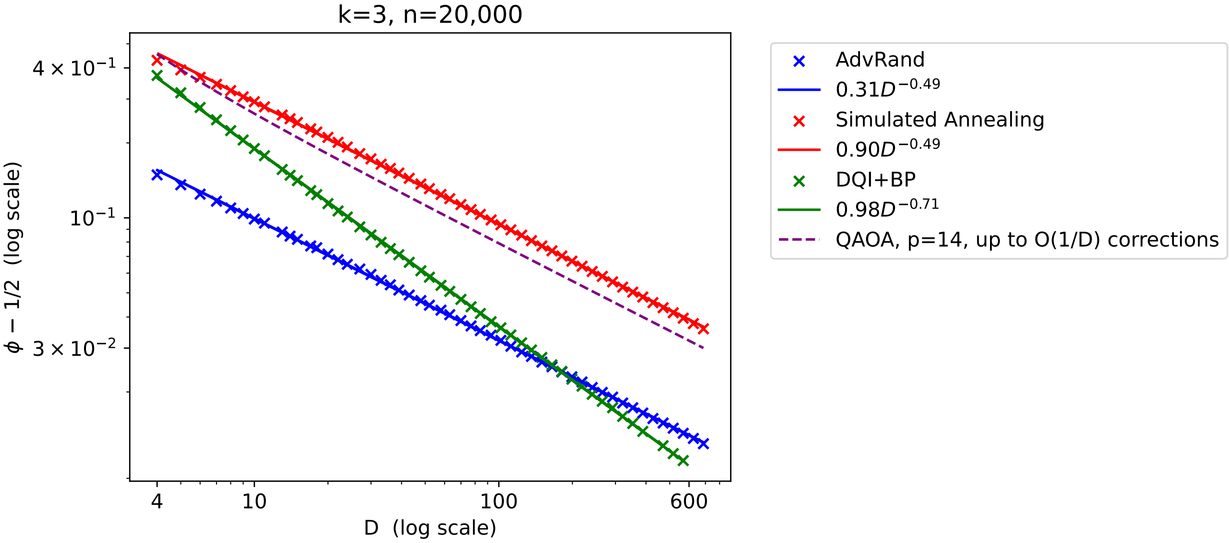

Figure 6: On Gallager's ensemble we compare the fraction ϕ of constraints satisfied in the solutions found by DQI+BP, AdvRand, and simulated annealing. We also show the approximate performance of $p = 14$ QAOA on large girth D-regular max-3-XORSAT instances, which was calculated in [31], up to $\mathcal{O}(1/D)$ corrections. We fix k, the number of variables in each constraint, at 3 and we vary D, the number of constraints that each variable is contained in, from 4 to 687. The red, blue, and green lines display the linear least-square fits to the log-log plot of $\phi - 1/2$ versus D. We keep the number of variables fixed at $n = 20,000$, thus $m = Dn/3$.

random D-regular hypergraphs in the $n \to \infty$ limit, is

$$\phi_{\mathrm{QAOA}} = \frac{1}{2} + \bar{\nu}_{14}^{[3]} \sqrt{\frac{3}{2(D-1)}} \pm \mathcal{O}(1/D), \tag{208}$$

where $\bar{\nu}_{14}^{[3]} = 0.6422$ [33].

In Fig. 6 we include a plot of the line $\phi_{\mathrm{QAOA}} = \frac{1}{2} + \bar{\nu}_{14}^{[3]} \sqrt{\frac{3}{2(D-1)}}$, alongside the empirical average-case performance on the Gallager ensemble of DQI using a standard belief propagation decoder (DQI+BP). At small D the comparison between $p = 14$ QAOA and DQI+BP is not fully conclusive due to the unknown $\mathcal{O}(1/D)$ corrections in (208), but for all D for which we can draw firm conclusions, *i.e.* large D, QAOA at $p = 14$ outperforms DQI+BP on max-3-XORSAT.

A second point of comparison between DQI and QAOA is the Sherrington-Kirkpatrick model, analyzed in [31, 34]. The analysis in [31] shows that for average-case max-2-XORSAT containing all $\binom{n}{2}$ possible constraints of the form $x_i \oplus x_j = v_k$, each with random $v_k \in \{0, 1\}$, the number of satisfied minus unsatisfied clauses achieved by QAOA scales as $\nu_p n^{3/2}$, where ν_p is a constant that depends on p, the number of rounds in the QAOA algorithm. Using Theorem 10.2 from §10 one finds that DQI, even using a classical decoder saturating the Shannon bound, would achieve at best $O(n^{3/2}/\sqrt{\log n})$. Thus, DQI is not competitive on this problem, at least using classical decoders[5].

A third point of comparison between DQI and QAOA is max-2-XORSAT where each variable is contained in exactly D constraints. MaxCut for D-regular graphs is the special case of this where $\mathbf{v}$ is the all ones vector. In [36] it was shown that QAOA with 17 rounds, when applied to the MaxCut problem on any 3-regular graph of girth at least 36 can achieve a cut fraction of 0.8971. For random 3-regular graphs, and any constant g, as the number of vertices goes to infinity, the fraction of vertices that are involved in loops of size g goes to zero. Thus for average-case instances of 3-regular MaxCut, QAOA with 17 rounds can asymptotically achieve cut fraction 0.8971. Furthermore, since the performance of QAOA on max-XORSAT is independent of $\mathbf{v}$, QAOA can also satisfy fraction 0.8971 of the constraints for 3-regular max-2-XORSAT on average-case graphs and arbitrary $\mathbf{v}$. By Theorem 10.4, the satisfaction fraction achievable by DQI with classical decoding is asymptotically upper bounded for 3-regular max-2-XORSAT by 0.75 if $\mathbf{v}$ is chosen uniformly at random. Thus, at least for random $\mathbf{v}$, DQI with classical decoders is beaten by QAOA on 3-regular max-2-XORSAT.

In [31] it was shown that when applied to MaxCut problems on large-girth D-regular graphs, QAOA achieves a cut fraction

$$\frac{1}{2} + \frac{\nu_p}{\sqrt{D-1}} \pm \mathcal{O}(1/D), \tag{209}$$

where ν_p is a constant that increases with the number of rounds p and for which $\nu_{17} = 0.6773$. As in the case of $D = 3$ described above, this implies the same asymptotic performance on average-case D-regular max-2-XORSAT. Since this analysis is only up to $\mathcal{O}(1/D)$ corrections, it cannot be quantitatively compared at finite D against DQI. Nevertheless, Theorem 10.4 shows that the approximation to average-case D-regular max-2-XORSAT achievable by DQI with classical decoders is limited to $1/2 + 1/(2D-2)$. Thus, QAOA outperforms DQI with classical decoders in the limit where $k = 2$ and D is large. To search for regimes of advantage for DQI one could instead consider increasing k together with D, or using quantum decoders, as discussed in §10.

[5] One could also consider using quantum decoders such as BPQM [35], which take advantage of the coherence of the errors and are limited only by the Holevo bound rather than the Shannon bound.

8.6 Algebraic Attacks Based on List Recovery

For our OPI instances, we believe that the most credible classical algorithms to consider as competitors to DQI must be attacks that exploit algebraic structure.

The max-LINSAT problem can be viewed as finding a codeword from C that approximately maximizes f. With DQI we have reduced this to a problem of decoding $C^\perp$ out to distance ℓ. For a Reed-Solomon code, as defined in (8), its dual is also a Reed-Solomon code. Hence, both C and $C^\perp$ can be efficiently decoded out to half their distances, which are $m-n+1$ and $n+1$, respectively. However, max-LINSAT is not a standard decoding problem, *i.e.* finding the nearest codeword to a given string under the promise that the distance to the nearest string is below some bound. In fact, exact maximum-likelihood decoding for general Reed-Solomon codes with no bound on distance is known to be NP-hard [7, 8].

The OPI problem is very similar to a problem studied in the coding theory literature known as *list-recovery*, applied in particular to Reed-Solomon codes. In list-recovery, for a code $C \subseteq \mathbb{F}_p^m$, one is given sets $F_0, F_1, \ldots, F_{m-1} \subseteq \mathbb{F}_p$ (which correspond to our sets $f_i^{-1}(+1)$), and asked to return all codewords $c \in C$ so that $c_i \in F_i$ for as many i as possible. It is easy to see that solving this problem for Reed-Solomon codes will solve the OPI problem, assuming the list of all matching codewords is small. However, existing list-recovery algorithms for Reed-Solomon codes rely on the size of the F_i being quite small (usually constant, relative to m) and do not apply in this parameter regime. In particular, the best known list-recovery algorithm for Reed-Solomon codes is the Guruswami-Sudan algorithm [37]; but this algorithm breaks down when the size of the $|F_i|$ is larger than m/n (this is the *Johnson bound* for list-recovery). In our setting, $|F_i| = p/2 \approx m/2$, which is much larger than $m/n \approx 10$. Thus, the Guruswami-Sudan algorithm does not apply. Moreover, we remark that in this parameter regime, if the f_i are random, we expect there to be exponentially many codewords satisfying all of the constraints; this is very different from the coding-theoretic literature on list-recovery, which generally tries to establish that the number of such codewords is at most polynomially large in m, so that they can all be returned efficiently. Thus, standard list-recovery algorithms are not applicable in the parameter regime we consider.

8.7 Lattice-Based Heuristics

A problem similar to our Optimal Polynomial Intersection problem—but in a very different parameter regime—has been considered before, and has been shown to be susceptible to lattice attacks. In more detail, in the work [4], Naor and Pinkas proposed essentially the same problem, but in the parameter regime where p is exponentially large compared to m, and where $|f_i^{-1}(+1)| \ll p/2$ is very small (in particular, not balanced, like we consider).[6] The work [4] conjectured that this problem was (classically) computationally difficult. This conjecture was challenged by Bleichenbacher and Nguyen in [5] using a lattice-based attack, which we describe in more detail below. However, this lattice-based attack does not seem to be effective—either in theory or in practice—against our OPI problem. Intuitively, one reason is that in the parameter regime that the attack of [5] works, a solution to the max-LINSAT problem—which will be unique with high probability—corresponds to a unique shortest vector in a lattice, which can be found via heuristic methods. In contrast,

[6] In this parameter regime, unlike ours, for random f_i it is unlikely that there are *any* solutions $\mathbf{x}$ with $f(\mathbf{x})$ appreciably large, so the problem of [4] also "plants" a solution $\mathbf{x}^*$ with $f(\mathbf{x}^*) = m$; the problem is to find this planted solution. Another difference is that DQI attains (207) for *any* functions f_i, while in the conjecture of Naor and Pinkas the f_i are random except for the values corresponding to the planted solution.

in our parameter regime, there are many optimal solutions, corresponding to many short vectors; moreover, empirically it seems that there are much shorter vectors in the appropriate lattice that do not correspond to valid solutions. Thus, these lattice-based methods do not seem to be competitive with DQI for our problem.

The target of the attack in [5] is syntactically the same as our problem, but in a very different parameter regime. Concretely, p is chosen to be much larger than m or n, while the size $r := |f_i^{-1}(+1)|$ of the set of "allowed" symbols for each $i = 1, \ldots, m$ is very small. (Here, we assume that $f_i^{-1}(+1)$ has the same size r for all i for simplicity of presentation; this can be relaxed). In [5], the f_i's are chosen as follows. Fix a planted solution $\mathbf{x}^*$, and set the f_i so that $f_i(\mathbf{b}_i \cdot \mathbf{x}^*) = 1$ for all i; thus $f(\mathbf{x}^*) = m$. Then for each i, $f_i(y)$ is set to 1 for a few other random values of $y \in \mathbb{F}_p$, and $f_i(z) = -1$ for the remaining $z \in \mathbb{F}_p$. With high probability, $\mathbf{x}^*$ is the unique vector with large objective value, and the problem is to find it.

In our setting, where $r = |f_i^{-1}(+1)| \approx p/2$, $m = p - 1$ and $n = \lceil m/10 \rfloor$, we expect there to be many vectors $\mathbf{x}$ with $f(\mathbf{x}) = m$ when the f_i are random. We have seen that DQI can find a solution with $f(\mathbf{x}) \approx 0.7179m$ (even for arbitrary f_i).

The way the attack of [5] works in our setting is the following. For $\mathbf{x} \in \mathbb{F}_p^n$, define a polynomial $P_{\mathbf{x}}(Z) = \sum_{j=1}^n x_j Z^{j-1}$. Let $F_i = f_i^{-1}(+1)$, and write $F_i = \{v_{i,1}, v_{i,2}, \ldots, v_{i,r}\} \subseteq \mathbb{F}_p$. If $f(\mathbf{x}) = m$, then $P_{\mathbf{x}}(\gamma^i) \in F_i$ for all $i = 1, \ldots, m$, where we recall from §2 that γ is a primitive element of $\mathbb{F}_p$, and $B_{i,j} = \gamma^{ij}$. Let $L_i(Z) = \prod_{j \neq i} \frac{Z - \gamma^j}{\gamma^i - \gamma^j}$. By Lagrange interpolation, we have

$$P_{\mathbf{x}}(Z) = \sum_{i=1}^{m} P_{\mathbf{x}}(\gamma^i) L_i(Z) = \sum_{i=1}^{m} \left(\sum_{j=1}^{r} \delta_{i,j}^{(P)} \, v_{i,j} \right) L_i(Z), \tag{210}$$

where $\delta_{i,j}^{(P)}$ is 1 if $P_{\mathbf{x}}(\gamma^i) = v_{i,j}$ and 0 otherwise. Since $P_{\mathbf{x}}(Z)$ has degree at most $n-1$, the coefficients on Z^k are equal to zero for $k = n, n+1, \ldots, p-1$. Hence, (210) gives us $p - n = m - n + 1$ $\mathbb{F}_p$-linear constraints on the vector $\boldsymbol{\delta} \in \{0,1\}^{rm}$ that contains the $\delta_{i,j}^{(P)}$'s. Collect these constraints in a matrix $A \in \mathbb{F}_p^{m-n+1 \times rm}$, so that $A\boldsymbol{\delta} = 0$. Consider the lattice $\Lambda := \{\boldsymbol{\delta}' \in \mathbb{Z}^{rm} \mid A\boldsymbol{\delta}' = 0 \mod p\}$. Our target vector $\boldsymbol{\delta}$ clearly lies in Λ, and moreover it has a very short ℓ_2 norm: $\|\boldsymbol{\delta}\|_2 = \sqrt{m}$. Thus, we may hope to find it using methods like LLL [38] or Schnorr's BKZ reduction [39, 40], and this is indeed the attack.[7] Upon finding a vector $\boldsymbol{\delta}$ of the appropriate structure (namely, so that $\delta_{i,j}^{(P)} = 1$ for exactly one $j \in \{1, \ldots, r\}$ for each i), we may read off the evaluations of $P_{\mathbf{x}}$ from the F_i, and hence recover $\mathbf{x}$.

Bleichenbacher and Nguyen show that in some parameter regimes, the target vector $\boldsymbol{\delta}$ with length $\sqrt{m}$ is likely to be the shortest vector in Λ. However, these parameter regimes are very different from ours. For example, their results hold for $p \approx 2^{80}$, and $r \leq 16$, with codes of rate n/m at least 0.88. In contrast, in our setting we have much larger lists, with $r \approx p/2$, and much lower-rate codes, with $n/m \approx 1/10$ (although in [5], they take $m, n \ll p$ while we take $m \approx p$, so this is not a direct comparison).

Empirically, this attack does not seem to work in our parameter regime. Indeed, the lattice heuristics do find short vectors in the lattice Λ, but these vectors are much shorter than $\sqrt{m}$ whenever m and $n_{\text{distractors}}$ are comparable to p (see Fig. 7). As a consequence, the success probability when applied to our OPI instances appears to decay exponentially with p.

[7] There are further improvements given in [5], notably passing to a sub-lattice that enforces the constraint that $\sum_{j=1}^r \delta_{i,j}^{(P)}$ is the same for all i.

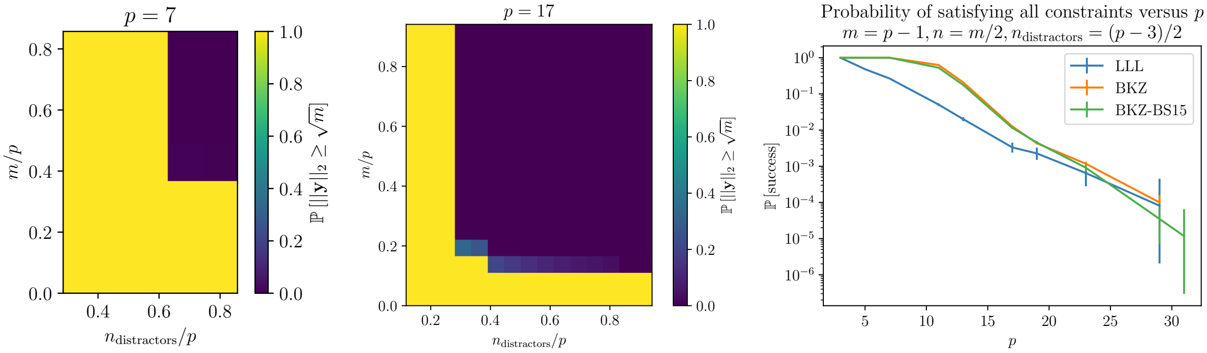

Figure 7: The attack of Bleichenbacher and Nguyen [5] is workable when the shortest nonzero vector in a particular lattice has weight $\sqrt{m}$. Above, left, we apply the BKZ algorithm [40] to find the shortest nonzero vector (under the 2-norm) in the lattices arising from random problem instances for various m and $n_{\text{distractors}}$. We observe that the shortest vector almost always has 2-norm $< \sqrt{m}$ in the regime where m and $n_{\text{distractors}}$ are both a significant fraction of p. Consequently, the success probability of the attack when applied to our OPI problem in the regime where $m = p-1, n = m/2, n_{\text{distractors}} = (p-3)/2$ appears to exhibit exponential decay with p whether LLL, BKZ with a block size of 15 ("BKZ-15") or BKZ with unlimited block size. In this regime we believe the lattice-based heuristic of [5] does not succeed.

9 Max-XORSAT Instances Advantageous to DQI Over Simulated Annealing

In this section we construct a class of max-XORSAT instances such that DQI, using belief propagation decoders, achieves a better approximation than we are able to achieve using simulated annealing if we restrict simulated annealing to a comparable number of computational steps. DQI+BP also achieves a better approximation than we obtain from any of the other general-purpose optimization algorithms that we try: greedy optimization, Prange's algorithm, and AdvRand. However, we do not claim this as an example of quantum advantage because we are able to also construct a classical heuristic tailored to the class of instances which, within reasonable runtime, beats the approximation achieved by DQI+BP. Also, as noted in the introduction, using very long anneals (up to 118 hours) we are able to reach the satisfaction fraction achieved by DQI+BP for the instance considered here. We have left the systematic investigation the scaling with n of the runtime of simulated annealing for these instances to future work.

Given a max-XORSAT instance $B\mathbf{x} \overset{\max}{=} \mathbf{v}$, the degree of a variable is the number of constraints in which it is contained. The degree of a constraint is the number of variables that are contained in it. Hence, the degree of the i^{th} constraint is the number of nonzero entries in the i^{th} row of B and the degree of the j^{th} variable is the number nonzero entries in the j^{th} column of B. For an LDPC code, the degree of a parity check is the number of bits that it contains, and the degree of a bit is the number of parity checks in which it is contained. These degrees correspond to the number of nonzero entries in the rows and columns of the parity check matrix.

Given a max-XORSAT instance, let Δ_j be the fraction of variables that have degree j. Let κ_i be the fraction of constraints that have degree i. This is illustrated in Fig. 8. Via DQI, a max-XORSAT instance with degree distribution Δ for the variables and κ for the constraints is

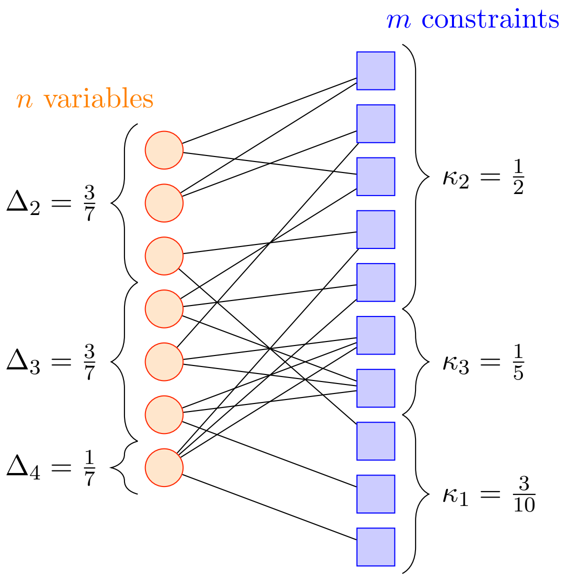

Figure 8: Tanner graph for a sparse irregular LDPC code illustrating the notation introduced in §9.

reduced to a decoding problem for a code with degree distribution Δ for the parity checks and κ for the bits. For regular LDPC codes, in which every bit has degree k and every constraint has degree D, the error rate from which belief propagation can reliably decode deteriorates as D increases. However, it has been discovered that belief propagation can still work very well for certain *irregular* codes in which the average degree $\bar{D}$ of the parity checks is large [41]. In contrast, we find that the approximate optima achieved by simulated annealing on the corresponding irregular max-XORSAT instances of average degree $\bar{D}$ are typically no better than on regular instances in which every variable has degree exactly $\bar{D}$. This allows us to find examples where DQI achieves a better approximation than simulated annealing.

If the m bits in an LDPC code have degree distribution κ the total number of nonzero entries in the parity check matrix is $m \sum_i i\kappa_i$. If the n parity checks in an LDPC code have degree distribution Δ then the total number of nonzero entries in the parity check matrix is $n \sum_j j\Delta_j$. Hence to define a valid LDPC code, a pair of degree distributions must satisfy

$$\begin{aligned} \textstyle\sum_{i=1}^m \kappa_i &= 1 && (211) \\ \textstyle\sum_{j=1}^n \Delta_j &= 1 && (212) \\ m\textstyle\sum_{i=1}^m i\kappa_i &= n\textstyle\sum_{j=1}^n j\Delta_j. && (213) \end{aligned}$$

Given any $\kappa_1, \ldots, \kappa_m$ and $\Delta_1, \ldots, \Delta_n$ satisfying these constraints, it is straightforward to sample uniformly from the set of all parity check matrices with m bits whose degree distribution is κ and n parity checks whose degree distribution is Δ. Furthermore, the maximum error rate from which belief propagation can reliably correct on codes from this ensemble can be computed in the limit of $m \to \infty$ by a method called density evolution, which numerically solves for the fixed point of a certain stochastic process [12]. Using this asymptotic maximum error rate as an objective function, one can optimize degree distributions to obtain irregular LDPC codes that outperform their regular counterparts [41].

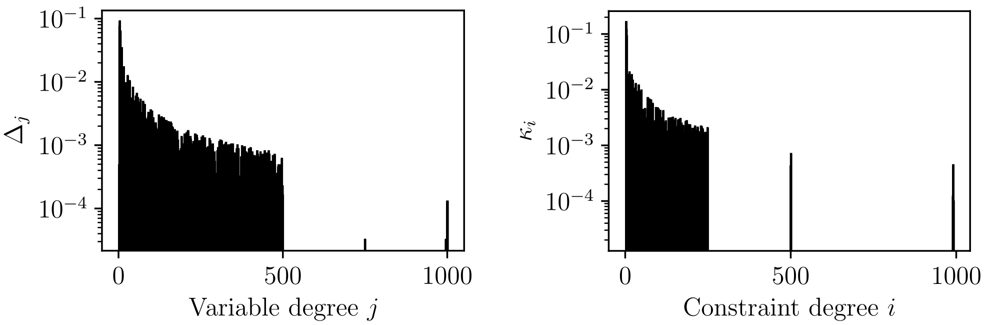

Figure 9: Degree distribution for an irregular instance sampled with $m = 50,000$ and $n = 31,216$. The full table of variable and constraint degrees is available in our Zenodo record https://doi.org/10.5281/zenodo.13327870.

As a concrete example, we consider the degree distribution shown in Fig. 9. After generating a random max-XORSAT instance with 50,000 constraints and 31,216 variables consistent with this degree distribution we find that belief propagation fails on 9 out of 10,000 trials of decoding from uniformly random errors of Hamming weight 6,350. This bit flip error rate of 6,350/50,000 is close to the asymptotic threshold of ≈ 13% predicted by density evolution. Thus for average case $\mathbf{v}$, Theorem 7.1 shows that DQI can asymptotically find solutions that satisfy at least $0.831m$ constraints with high probability. In contrast, when we run simulated annealing on this instance with 10^6 sweeps, for example, in sixteen trials the fraction of satisfied constraints ranges from 0.81024 to 0.81488. The performance of simulated annealing on this instance as a function of number of sweeps is discussed in detail in §8.2. Prange's algorithm would be predicted to achieve approximately 0.8122, and experimentally we saw 0.8124. The greedy algorithm performs far worse than simulated annealing on this instance. In sixteen trials its best satisfaction fraction was 0.666. Our trial of AdvRand achieved 0.5536.

The following classical algorithm can exceed the approximation achieved by DQI+BP on the above example, by exploiting the highly unbalanced degree distribution of its constraints. We modify simulated annealing (which is described in §8.1) by adding a β-dependent factor to each term in the objective function. Letting n_i denote the number of variables contained in constraint i, we use the objective function

$$f^{(\beta)}(\mathbf{x}) = \sum_{i=1}^{m} \max\left(0, 1 - e^{-\beta/n_i}\right) f_i\left(\sum_{j=1}^{n} B_{ij} x_j\right). \tag{214}$$

We apply 1 million sweeps (since there are 50,000 constraints this corresponds to fifty billion Metropolis updates), interpolating linearly from $\beta = 0$ to $\beta = 5$. After this, we are left with solutions that satisfy approximately $0.88m$ clauses. We call this algorithm *irregular annealing* since it takes advantage of the irregularity of the instance by prioritizing the lower-degree constraints early in the annealing process.

10 Limitations of DQI

In addition to the power of DQI for solving optimization problems it is also interesting to delineate its fundamental limits. Because DQI reduces optimization problems to decoding problems, some limitations of DQI can be deduced from information theory. In this section we use information-theoretic considerations to prove upper bounds on the performance of DQI on max-XORSAT. We first consider the case where DQI uses a classical decoding algorithm implemented reversibly, as is done throughout the rest of this manuscript. We then consider the case of intrinsically quantum decoders. Lastly, we consider the special case of max-2-XORSAT, which in many respects behaves differently from max-k-XORSAT with $k \geq 3$.

10.1 General Limitations of DQI Under Classical Decoding

DQI reduces max-XORSAT to a decoding problem for a code $C^\perp$ with m bits and n parity checks. Hence its rate R is

$$R = 1 - \frac{n}{m}. \tag{215}$$

The decoding is required to succeed with high probability when an error string $\mathbf{e}$ with Hamming weight ℓ has been added to the codeword. We can model this by an error channel where each bit is independently flipped with probability

$$p = \ell/m. \tag{216}$$

This model is called the binary symmetric channel BSC(p). Although the distribution over error weights in DQI is given by $\Pr(|\mathbf{e}| = k) = |w_k|^2$, whereas BSC($p$) has $\Pr(|\mathbf{e}| = k) = p^k(1-p)^{m-k}\binom{m}{k}$, the channels with these error distributions both asymptotically yield the same information-theoretic capacity because in both cases the probability distribution over error weights is narrowly peaked around ℓ. As shown by Shannon, the rate of a code that can reliably transmit information over BSC(p) is limited by

$$R \leq 1 - H_2(p), \tag{217}$$

where $H_2(p) = -p\log_2(p) - (1-p)\log_2(1-p)$ is the binary entropy function. Substituting (215) and (216) into (217) yields $\frac{n}{m} \geq H_2\left(\frac{\ell}{m}\right)$, which implies

$$\frac{\ell}{m} \leq H_2^{-1}\left(\frac{n}{m}\right) \tag{218}$$

where the inverse H_2^{-1} is well-defined, because $\ell/m \leq 1/2$. Substituting this into the semicircle law (4) yields the following bound

$$\frac{\langle s \rangle_{\text{Shannon}}}{m} \leq \frac{1}{2} + \sqrt{H_2^{-1}\left(\frac{n}{m}\right)\left(1 - H_2^{-1}\left(\frac{n}{m}\right)\right)}. \tag{219}$$

Since $mH_2^{-1}(n/m)$ in general exceeds $d^\perp/2$, Theorem 1.1 does not apply, though Theorem 7.1 still does. That is, this must be interpreted as a bound on the performance of DQI for max-XORSAT with average case $\mathbf{v}$.

Let's now consider this limit in relation to the ensemble of degree-D max-k-XORSAT instances in which B is chosen from the (k, D)-regular Gallager ensemble. In this case we have $n/m = k/D$, which can be substituted into (219) to yield concrete upper bounds on $\langle s \rangle_{\text{Shannon}}/m$. In Fig. 10

we compare these upper bounds on the performance of DQI with classical decoding against the empirical performance of simulated annealing as well as the known asymptotic performance of Prange's algorithm.

We note that in Fig. 10, DQI is analyzed asymptotically, whereas simulated annealing results are obtained empirically at finite n. For simulated annealing we take $n \simeq 2,000$ and use $500,000$ sweeps, as we found these parameters to achieve a good tradeoff between asymptotic informativeness and computational convenience. To improve the results of simulated annealing we run it five times for each instance with different random seeds and report the maximum number of satisfied constraints achieved by any of these five trials. This technique is referred to as "restarts," and it is often used in simulated annealing because running r repetitions of simulated annealing and keeping the best result may often outperform the equally costly procedure of running a single anneal with r times as many sweeps. We have $n \simeq 2,000$ rather than $n = 2,000$ because in the Gallager ensemble n must always be a multiple of D.

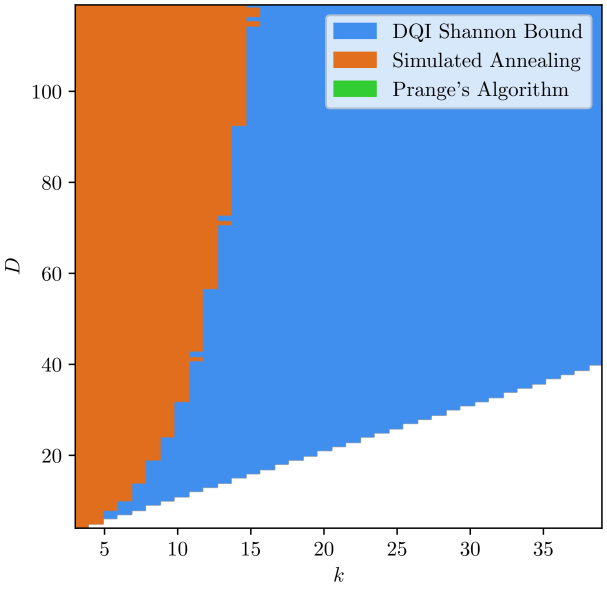

Figure 10: Here we consider degree-D max-k-XORSAT instances $B\mathbf{x} \overset{\max}{=} \mathbf{v}$ where $\mathbf{v}$ is uniformly random and B is drawn from the (k, D)-regular Gallager ensemble. In the orange region it is information-theoretically impossible for DQI with classical decoding to outperform simulated annealing on average-case instances from Gallager's ensemble. In the blue region, DQI with maximum-likelihood (*i.e.* Shannon-limit) decoding achieves a higher satisfaction fraction than simulated annealing, but realizing this advantage with polynomial-time decoders remains an open problem. Prange's algorithm does not win on any region of this plot.

Although our main focus in this section is on limitations of DQI it is also worth discussing the possibilities of DQI. In the blue region of Fig. 10 it is information-theoretically possible for DQI to achieve average-case quantum advantage using classical decoders. Based on computational experiments, we believe that this potential advantage is not realized by belief propagation decoding. This is because the number of errors that belief propagation can successfully correct falls increasingly short of the Shannon limit as the parity check matrices defining $C^\perp$ become denser. It is an open question whether efficient classical decoders can be devised that approach the Shannon limit closely enough for denser codes to allow DQI to outperform simulated annealing on average case instances from Gallager's ensemble. Efficient classical decoding of LDPC codes with denser than usual parity check matrices has so far not been a subject of intensive research but some results in this direction are obtained in [42–48].

To make (219) less unwieldy, we can use the following useful bound from [49].

Theorem 10.1. *For all $x \in [0,1]$,*

$$\frac{x}{2\log_2\left(\frac{6}{x}\right)} \leq H_2^{-1}(x) \leq \frac{x}{\log_2\left(\frac{1}{x}\right)}. \tag{220}$$

Substituting (220) into (219) yields the simpler but looser bound $\frac{\langle s \rangle}{m} \leq \frac{1}{2} + \sqrt{\frac{n/m}{\log(m/n)}}$. In summary, we have the following.

Theorem 10.2. *Consider a max-XORSAT instance with m constraints and n variables where $\mathbf{v} \in \mathbb{F}_2^m$ is chosen uniformly at random. The expected number of satisfied clauses $\langle s \rangle$ obtained by DQI in the limit of large n using a classical decoder is bounded by*

$$\frac{\langle s \rangle}{m} \leq \frac{1}{2} + \sqrt{H_2^{-1}\left(\frac{n}{m}\right)\left(1 - H_2^{-1}\left(\frac{n}{m}\right)\right)} \leq \frac{1}{2} + \sqrt{\frac{n/m}{\log(m/n)}}. \tag{221}$$

10.2 General Limitations of DQI Under Quantum Decoding

The bit flip errors that must be decoded in DQI are in coherent superposition. One can treat these errors classically by implementing a classical decoding algorithm as a reversible circuit and separately performing classical error correction on each "branch" of the superposition. This is the strategy we describe and analyze throughout this manuscript. However, this is not necessarily optimal. Information-theoretically, at least, coherent errors are more advantageous than random errors.

In the preceding section, to avoid complications arising from the details of the distribution $\Pr(|\mathbf{e}| = k) = |w_k|^2$ over error weights arising in DQI, we approximated this by the binary symmetric channel with bit flip probability $p = \ell/m$. Similarly, we may approximate our distribution over coherent errors using the following channel

$$\begin{aligned} |0\rangle &\rightarrow |0_p\rangle \\ |1\rangle &\rightarrow |1_p\rangle \end{aligned}$$

where

$$\begin{aligned} |0_p\rangle &= \sqrt{1-p}\,|0\rangle + \sqrt{p}\,|1\rangle \\ |1_p\rangle &= \sqrt{p}\,|0\rangle + \sqrt{1-p}\,|1\rangle \end{aligned}$$

and $p = \ell/m$. The capacity of this channel is limited by Holevo's bound, which states

$$R \leq \chi(p), \tag{222}$$

where

$$\chi(p) = S\left(\frac{1}{2}\left|0_p\right\rangle\left\langle 0_p\right| + \frac{1}{2}\left|1_p\right\rangle\left\langle 1_p\right|\right), \tag{223}$$

and S denotes the von Neumann entropy. By direct computation, one finds

$$\chi(p) = H_2\left(\frac{1}{2} - \sqrt{p(1-p)}\right), \tag{224}$$

where, as before, H_2 denotes the binary entropy function. When applying DQI to a max-XORSAT problem with n variables and m constraints, our decoding problem is for a code $C^\perp$ of rate $R = 1 - n/m$. Hence, the Holevo bound implies via the semicircle law (4)

$$\frac{\langle s\rangle_{\text{Holevo}}}{m} \leq \frac{1}{2} + \sqrt{\chi^{-1}\left(1 - \frac{n}{m}\right)\left(1 - \chi^{-1}\left(1 - \frac{n}{m}\right)\right)} \tag{225}$$

where the inverse χ^{-1} is well-defined, because $p = \ell/m \leq 1/2$. Using (224) and simplifying one can equivalently write (225) as

$$\frac{\langle s\rangle_{\text{Holevo}}}{m} \leq 1 - H_2^{-1}\left(1 - \frac{n}{m}\right). \tag{226}$$

Next we apply this limit to Gallager's ensemble where $n/m = k/D$. If we compare this limit to the performance of simulated annealing and Prange's algorithm for average-case instances of max-XORSAT drawn from Gallager's ensemble, analogously to in Fig. 10, we do not find any region of the (k, D)-plane in which simulated annealing or Prange's algorithm beat this upper bound on DQI's performance. Thus, the Holevo bound does not allow us to rule out quantum advantage by DQI with quantum decoding for any region of the (k, D) plane for Gallager-ensemble instances. But for other ensembles it may be successful in doing so.

In this section we consider the Holevo bound only as a tool for ruling out quantum advantage. But it also suggests that in regions where quantum advantage is not achievable using DQI with classical decoders it might be achievable using DQI with quantum decoders, if efficient quantum circuits can be found to implement them. Some exciting results on efficient quantum circuits for decoding coherent bit flip errors can be found in [18, 21, 35].

10.3 Limitations of DQI with Classical Decoding for max-2-XORSAT

The special case of max-k-XORSAT where $k = 2$ behaves somewhat differently in the context of DQI than $k > 2$ and benefits from separate analysis. The max-2-XORSAT problem is widely studied, particularly the special case of max-2-XORSAT where $\mathbf{v}$ is the all-ones vector, which is known as MaxCut. Additionally, max-2-XORSAT is the unweighted special case of the Quadratic Unconstrained Binary Optimization (QUBO) problem.

The code $C^\perp$ dual to an instance of max-2-XORSAT is one in which each bit is contained in exactly two parity checks. Such codes are sometimes referred to as cycle codes (not to be confused with cyclic codes, which are unrelated). It is known that cycle codes have minimum distance that is at most logarithmic in their block length [50]. Although decoding of adversarial errors is impossible beyond half the code distance, a large fraction of random errors of far greater Hamming weight may be decodable.

Interestingly, for cycle codes, unlike for general LDPC codes, polynomial-time decoders can achieve exact maximum-likelihood decoding, *i.e.* saturate the information-theoretic limit. A cycle code can be associated with a graph whose edges represent bits and whose vertices represent the parity checks. A given syndrome corresponds to a subset T of the vertices, and the lowest Hamming weight error (which is the maximum likelihood error for the binary symmetric channel) is given by the minimum-weight T-join, which can be found in polynomial time [51].

In [50], the following theorem is proven.

Theorem 10.3. *Consider an asymptotic family of LDPC codes in which each bit is contained in exactly two parity checks and each parity check contains exactly D bits. The rate of these codes is then $R = 1 - 2/D$. Let p_2 be the largest probability such that, if each bit is independently flipped with probability p_2, then maximum-likelihood decoding will recover the original codeword with probability converging to one in the limit of large block size. Then,*

$$p_2 \leq \frac{1}{2}\frac{(1-\sqrt{R})^2}{1+R}. \tag{227}$$

This theorem provides new information specific to $k = 2$ because, as one can easily verify, the above bound on p_2 lies below the corresponding Shannon bound $p_{\mathrm{general}} \leq H_2^{-1}(1-R)$ for general codes. We also note that our computer experiments suggest that, for $D = 3$ the bound (227) is essentially saturated by the cycle codes arising from random 3-regular graphs.

For max-2-XORSAT, we are mainly interested in the case $2\ell + 1 > d^{\perp}$ since, as noted above, $d^{\perp} = O(\log n)$. We therefore rely on Theorem 7.1, which together with Theorem 10.3 implies the following.

Theorem 10.4. *Consider an asymptotic family of max-2-XORSAT instances in which $\mathbf{v} \in \mathbb{F}_2^m$ is chosen uniformly at random and each variable is contained in exactly D constraints. In the limit of large n the performance of DQI using classical decoders is limited by*

$$\frac{\langle s \rangle}{m} \leq \frac{1}{2} + \frac{1}{2(D-1)}. \tag{228}$$

Proof. Rewriting (167) from Theorem 7.1 in terms of the expected number of constraints satisfied $\langle s \rangle$ instead of the expected objective value $\langle f \rangle$ yields

$$\frac{\langle s \rangle}{m} = \frac{1}{2} + \sqrt{\frac{\ell}{m}\left(1 - \frac{\ell}{m}\right)} - \varepsilon. \tag{229}$$

From Theorem 10.3, the information-theoretic limit of decoding cycle codes is at

$$\frac{\ell}{m} = \frac{1}{2}\frac{(1-\sqrt{R})^2}{1+R} \quad \text{and} \quad \varepsilon = 0 \tag{230}$$

Substituting (230) and $R = 1 - 2/D$ into (229) and simplifying yields (228). □

11 DQI for Folded Codes and over Extension Fields

In §5, we describe the DQI algorithm for the max-LINSAT problem over prime fields. However, DQI works over extension fields as well, and also works for so-called *folded codes*. In this section, we go through the details to extend DQI to these settings. Our main motivation is to show that DQI is applicable to the problem considered by Yamakawa and Zhandry in [1], which is similar to our OPI problem, but for folded Reed-Solomon codes. Moreover, as discussed in §4, if a variant of Theorem 1.1 applies in the regime studied by Yamakawa and Zhandry, then equation (3) implies that DQI can find a solution satisfying all constraints.

11.1 Folded max-LINSAT problem

In [1], Yamakawa and Zhandry define the following oracle problem, which they prove can be solved in polynomially many queries by a quantum computer but requires exponentially many queries for classical computers.

Definition 11.1. *Fix a prime power q and integers m, n, r such that r divides m and $m > n$. Let $\mathcal{O} : \{1, \ldots, m/r\} \times \mathbb{F}_q^r \to \{0, 1\}$ be a random function. Let $B \in \mathbb{F}_q^{m \times n}$ be a Vandermonde matrix (so that $B_{i,j} = \gamma^{ij}$ for $i \in \{0, \ldots, m-1\}$, $j \in \{0, \ldots, n-1\}$ where γ is a primitive element of $\mathbb{F}_q$), written as*

$$B = \left[\begin{array}{c} B_1 \\ \hline B_2 \\ \hline \vdots \\ \hline B_{m/r} \end{array}\right] \tag{231}$$

where $B_i \in \mathbb{F}_q^{r \times n}$. The Yamakawa-Zhandry problem is, given B and query access to $\mathcal{O}$, to efficiently find $\mathbf{x} \in \mathbb{F}_q^n$ such that $\mathcal{O}(i, B_i\mathbf{x}) = 1$ for all $i \in \{1, \ldots, m/r\}$.

The problem in Definition 11.1 has some similarities to Definition 2.1 of the main text, and especially to our OPI example in §2, but is not exactly the same, as the problem in Definition 11.1 is for *folded* Reed-Solomon codes, over an extension field $\mathbb{F}_q$. Below, we extend DQI to this setting. More precisely, we consider the following generalization of max-LINSAT.

Definition 11.2 (Folded max-LINSAT)**.** *Let $\mathbb{F}_q$ be a finite field, where q is any prime power. For $i = 1, \ldots, m/r$, let $f_i : \mathbb{F}_q^r \to \{+1, -1\}$ be arbitrary functions. Given a matrix $B \in \mathbb{F}_q^{m \times n}$ written as*

$$B = \left[\begin{array}{c} B_1 \\ \hline B_2 \\ \hline \vdots \\ \hline B_{m/r} \end{array}\right] \tag{232}$$

*with $B_i \in \mathbb{F}_q^{r \times n}$, the r-*folded max-LINSAT problem *is to find $\mathbf{x} \in \mathbb{F}_q^n$ maximizing the objective function*

$$f(\mathbf{x}) = \sum_{i=1}^{m/r} f_i(B_i\mathbf{x}). \tag{233}$$

We now describe how to adapt the presentation in §5.2 to the folded max-LINSAT problem. As before, we first discuss the properties of the DQI state $|P(f)\rangle := \sum_{\mathbf{x} \in \mathbb{F}_p^n} P(f(\mathbf{x}))|\mathbf{x}\rangle$ and then describe the algorithm for creating it.

Again, mirroring §5.2, we assume that $2\ell + 1 < d^\perp$ where $d^\perp$ is the minimum distance of the folded code $C^\perp = \{\mathbf{d} \in \mathbb{F}_q^m : B^T\mathbf{d} = \mathbf{0}\}$. Note that folding affects the definition of $d^\perp$. In a folded code, we view every codeword $\mathbf{y} \in \mathbb{F}_q^m$ as an m/r-tuple $(\mathbf{y}_1, \ldots \mathbf{y}_{m/r})$ of elements of $\mathbb{F}_q^r$ and regard each $\mathbf{y}_i$ as a symbol. Consequently, the Hamming weight $|.| : \mathbb{F}_q^m \to \{0, \ldots, m/r\}$ associated with the folded code is the number of $\mathbf{y}_i$ not equal to $\mathbf{0} \in \mathbb{F}_q^r$.

11.2 DQI Quantum State for Folded max-LINSAT

As in §5.2, we assume that no f_i is constant and that the preimages $F_i := f_i^{-1}(+1)$ have the same cardinality for all $i = 1, \ldots, m/r$. This allows us to define g_i as f_i shifted and rescaled so that its Fourier transform

$$\tilde{g}_i(\mathbf{y}) = \frac{1}{\sqrt{q^r}} \sum_{\mathbf{x} \in \mathbb{F}_q^r} \omega_p^{\mathrm{tr}(\mathbf{y} \cdot \mathbf{x})} g_i(\mathbf{x}), \tag{234}$$

where $\mathrm{tr} : \mathbb{F}_q \to \mathbb{F}_p$ given by $\mathrm{tr}(x) = x + x^p + x^{p^2} + \ldots + x^{q/p}$ is the field trace, vanishes at $\mathbf{y} = \mathbf{0} \in \mathbb{F}_q^r$ and is normalized, *i.e.* $\sum_{\mathbf{x} \in \mathbb{F}_q^r} |g_i(\mathbf{x})|^2 = \sum_{\mathbf{y} \in \mathbb{F}_q^r} |\tilde{g}_i(\mathbf{y})|^2 = 1$. More explicitly, we define

$$g_i(\mathbf{x}) := \frac{f_i(\mathbf{x}) - \overline{f}}{\varphi} \tag{235}$$

where $\overline{f} := \frac{1}{q^r} \sum_{\mathbf{x} \in \mathbb{F}_q^r} f_i(\mathbf{x})$ and $\varphi := \left(\sum_{\mathbf{y} \in \mathbb{F}_q^r} |f_i(\mathbf{y}) - \overline{f}|^2 \right)^{1/2}$. The sums $f(\mathbf{x}) = \sum_{i=1}^{m/r} f_i(B_i \mathbf{x})$ and $g(\mathbf{x}) = \sum_{i=1}^{m/r} g_i(B_i \mathbf{x})$ are related by $f(\mathbf{x}) = g(\mathbf{x})\varphi + m\overline{f}/r$. Substituting this relationship for f in $P(f)$, we obtain an equivalent polynomial $Q(g)$ which, by Lemma A.1 in Appendix A, can be expressed as a linear combination of elementary symmetric polynomials $P^{(k)}$

$$Q(g(\mathbf{x})) := \sum_{l=0}^{\ell} u_k P^{(k)} \left(g_1(B_1 \mathbf{x}), \ldots, g_{m/r}(B_{m/r} \mathbf{x}) \right). \tag{236}$$

We will write the DQI state

$$|P(f)\rangle = \sum_{\mathbf{x} \in \mathbb{F}_q^n} P(f(\mathbf{x})) |\mathbf{x}\rangle = \sum_{\mathbf{x} \in \mathbb{F}_q^n} Q(g(\mathbf{x})) |\mathbf{x}\rangle = |Q(g)\rangle \tag{237}$$

as a linear combination of $|P^{(0)}\rangle, \ldots, |P^{(\ell)}\rangle$ defined as

$$|P^{(k)}\rangle := \frac{1}{\sqrt{q^{n-rk} \binom{m/r}{k}}} \sum_{\mathbf{x} \in \mathbb{F}_q^n} P^{(k)} \left(g_1(B_1 \mathbf{x}), \ldots, g_{m/r}(B_{m/r} \mathbf{x}) \right) |\mathbf{x}\rangle . \tag{238}$$

By definition,

$$\begin{aligned}
P^{(k)}(g_1(B_1 \mathbf{x}), \ldots, g_{m/r}(B_{m/r} \mathbf{x})) &= \sum_{\substack{i_1, \ldots, i_k \\ \text{distinct}}} \prod_{i \in \{i_1, \ldots, i_k\}} g_i(B_i \mathbf{x}) && (239) \\
&= \sum_{\substack{i_1, \ldots, i_k \\ \text{distinct}}} \prod_{i \in \{i_1, \ldots, i_k\}} \left(\frac{1}{\sqrt{q^r}} \sum_{\mathbf{y}_i \in \mathbb{F}_q^r} \omega_p^{-\mathrm{tr}(\mathbf{y}_i \cdot B_i \mathbf{x})} \; \tilde{g}_i(\mathbf{y}_i) \right) && (240) \\
&= \sum_{\substack{\mathbf{y} \in \mathbb{F}_q^m \\ |\mathbf{y}| = k}} \frac{1}{\sqrt{q^{rk}}} \, \omega_p^{-\mathrm{tr}((B^T \mathbf{y}) \cdot \mathbf{x})} \prod_{\substack{i=1 \\ \mathbf{y}_i \neq 0}}^{m/r} \tilde{g}_i(\mathbf{y}_i) && (241)
\end{aligned}$$

where $|.| : \mathbb{F}_q^m \to \{0, \ldots, m/r\}$ is the Hamming weight associated with the folded code. From (241) we see that the Quantum Fourier Transform of $|P^{(k)}\rangle$ is

$$|\widetilde{P}^{(k)}\rangle := F^{\otimes n}|P^{(k)}\rangle = \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_q^m \\ |\mathbf{y}|=k}} \left(\prod_{\substack{i=1 \\ \mathbf{y}_i\neq 0}}^{m/r} \tilde{g}_i(\mathbf{y}_i) \right) |B^T\mathbf{y}\rangle. \tag{242}$$

As in §5.2, if $|\mathbf{y}| < d^{\perp}/2$, then $B^T\mathbf{y}$ are all distinct and $|P^{(0)}\rangle, \ldots, |P^{(\ell)}\rangle$ form an orthonormal set. Consequently,

$$|P(f)\rangle = \sum_{k=0}^{\ell} w_k \, |P^{(k)}\rangle \tag{243}$$

where

$$w_k = u_k \sqrt{q^{n-rk}\binom{m/r}{k}}, \tag{244}$$

and $\langle P(f)|P(f)\rangle = \|\mathbf{w}\|^2$.

11.3 DQI Algorithm for Folded max-LINSAT

Similarly to DQI for general max-LINSAT, the algorithm for folded max-LINSAT uses three quantum registers: a *weight register* comprising $\lceil \log_2 \ell \rceil$ qubits, an *error register* with $m\lceil \log_2 q\rceil$ qubits, and a *syndrome register* with $nr\lceil \log_2 q\rceil$ qubits. We will consider the error and syndrome registers as consisting of m/r and n subregisters, respectively, where each subregister consists of $r\lceil \log_2 q\rceil$ qubits. We will also regard the rightmost qubits from all subregisters of the error register as forming the *mask register* of m/r qubits. We assume that the encoding of $\mathbb{F}_q$ into each of the r components of a subregister uses a basis that contains 1, so that $1 \in \mathbb{F}_q$ is encoded as $|0, \ldots, 0, 1\rangle$.

We begin by initializing the weight register in the normalized state $\sum_{k=0}^{\ell} w_k \, |k\rangle$. Next, we prepare the mask register in the Dicke state corresponding to the weight register

$$\to \sum_{k=0}^{\ell} w_k \, |k\rangle \, \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\boldsymbol{\mu}\in\{0,1\}^{m/r} \\ |\boldsymbol{\mu}|=k}} |\boldsymbol{\mu}\rangle \tag{245}$$

and then uncompute the weight register, obtaining

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\boldsymbol{\mu}\in\{0,1\}^{m/r} \\ |\boldsymbol{\mu}|=k}} |\boldsymbol{\mu}\rangle \, . \tag{246}$$

Let G_i denote a unitary acting on $r\lceil \log_2 q\rceil$ qubits that sends $|\mathbf{0}\rangle$ to $|\mathbf{0}\rangle$ and $|0, \ldots, 0, 1\rangle$ to

$$\sum_{\mathbf{c}\in\mathbb{F}_q^r} \tilde{g}_i(\mathbf{c}) \, |\mathbf{c}\rangle = \sum_{\mathbf{c}\in\mathbb{F}_q^r\setminus\{\mathbf{0}\}} \tilde{g}_i(\mathbf{c}) \, |\mathbf{c}\rangle \, . \tag{247}$$

See §11.4 below for an implementation of G_i in the oracle setting. As in the case of DQI for general max-LINSAT, parallel application $G := \prod_{i=1}^{m/r} G_i$ of G_i to all subregisters of the error register preserves the Hamming weight, so that

$$\sum_{\substack{\boldsymbol{\mu}\in\{0,1\}^{m/r}\\ |\boldsymbol{\mu}|=k}} G\,|\boldsymbol{\mu}\rangle = \sum_{\substack{\mathbf{y}\in\mathbb{F}_q^m\\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}\left(\mathbf{y}_{y(1)}\right)\dots\tilde{g}_{y(k)}\left(\mathbf{y}_{y(k)}\right)|\mathbf{y}\rangle \tag{248}$$

where $\mathbf{y}_i$ for $i \in \{1,\dots,m/r\}$ denotes the i^{th} entry of $\mathbf{y}$, and $y(j)$ denotes the index of the j^{th} nonzero entry of $\mathbf{y}$. Consequently, by applying G to the error register, we obtain

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_q^m\\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}(\mathbf{y}_{y(1)})\dots\tilde{g}_{y(k)}(\mathbf{y}_{y(k)})\,|\mathbf{y}\rangle\,. \tag{249}$$

Next, we reversibly compute $B^T\mathbf{y}$ into the syndrome register

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_q^m\\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}(\mathbf{y}_{y(1)})\dots\tilde{g}_{y(k)}(\mathbf{y}_{y(k)})\,|\mathbf{y}\rangle\,|B^T\mathbf{y}\rangle\,. \tag{250}$$

The task of finding $\mathbf{y}$ from $B^T\mathbf{y}$ is the bounded distance syndrome decoding problem on the folded code $C^\perp = \{\mathbf{y} \in \mathbb{F}_q^m : B^T\mathbf{y} = \mathbf{0}\}$. Consequently, uncomputing the content of the error register can be done efficiently whenever the bounded distance decoding problem on $C^\perp$ can be solved efficiently out to distance ℓ.

Uncomputing disentangles the syndrome register from the error register, leaving behind

$$\to \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_q^m\\ |\mathbf{y}|=k}} \tilde{g}_{y(1)}(\mathbf{y}_{y(1)})\dots\tilde{g}_{y(k)}(\mathbf{y}_{y(k)})\,|\mathbf{y}\rangle\,|B^T\mathbf{y}\rangle \tag{251}$$

$$= \sum_{k=0}^{\ell} w_k \frac{1}{\sqrt{\binom{m/r}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_q^m\\ |\mathbf{y}|=k}} \left(\prod_{\substack{i=1\\ \mathbf{y}_i\neq 0}}^{m/r} \tilde{g}_i(\mathbf{y}_i)\right)|B^T\mathbf{y}\rangle \tag{252}$$

$$= \sum_{k=0}^{\ell} w_k\,|\widetilde{P}^{(k)}\rangle \tag{253}$$

which becomes the desired state $|P(f)\rangle$ after applying the Quantum Fourier Transform.

11.4 Oracle Access to Objective Function

In our discussion of DQI for general max-LINSAT, we assumed that the field size p is polynomial in n. In that setting, the objective functions f_i can be given explicitly by their values at all elements of $\mathbb{F}_p$ and hence the gates G_i can be realized efficiently using techniques from [25]. By contrast, in Yamakawa-Zhandry problem, the random function $\mathcal{O}$ is available via query access to an oracle.

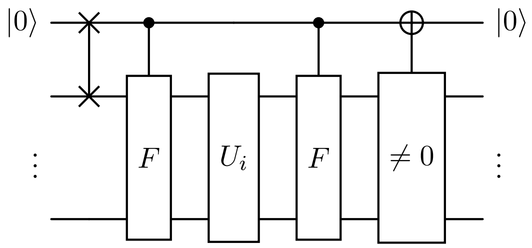

Figure 11: Quantum circuit implementing G_i gate using oracle access. The circuit takes as input a single subregister of the error register and employs an auxiliary qubit initialized in $|0\rangle$. It begins by swapping the qubit corresponding to 1 in the input with the auxiliary qubit. Then it applies the Quantum Fourier Transform (QFT) F over $\mathbb{F}_q^r$, followed by a call to the oracle U_i, followed by another QFT. The QFT gates are conditional on the auxiliary qubit. The circuit ends with the uncomputation of the auxiliary qubit by flipping it when the subregister is non-zero.

Here, we show how to realize the gates G_i in a setting where the functions $f_i(.) = \mathcal{O}(i, .)$ are provided by oracles and without assuming that the field size q is polynomial in n. For simplicity, we assume that q is a power of two, f_i is balanced, *i.e.* $|f^{-1}(+1)| = q^r/2$, and $f_i(\mathbf{0}) = +1$.

Suppose oracle U_i for f_i is defined as

$$U_i |\mathbf{x}\rangle = f_i(\mathbf{x}) |\mathbf{x}\rangle \tag{254}$$

for $\mathbf{x} \in \mathbb{F}_q^r$ and let

$$F |\mathbf{x}\rangle = \frac{1}{\sqrt{q^r}} \sum_{\mathbf{y}\in\mathbb{F}_q^r} (-1)^{\mathrm{tr}(\mathbf{x}\cdot\mathbf{y})} |\mathbf{y}\rangle \tag{255}$$

be the Quantum Fourier Transform on a subregister of the error register. Then G_i can be realized using a single auxiliary qubit as shown in Fig. 11. When the input is $|\mathbf{0}\rangle$, then all operations act as identity, so $G_i |\mathbf{0}\rangle = |\mathbf{0}\rangle$. When the input is $|0, \dots, 0, 1\rangle$, then the SWAP gate sets the auxiliary qubit to the $|1\rangle$ state and the input qubits into $|\mathbf{0}\rangle$. Subsequent three operations yield

$$FU_iF |\mathbf{0}\rangle = \sum_{\mathbf{y}\in\mathbb{F}_q^r} \tilde{g}_i(\mathbf{y}) |\mathbf{y}\rangle . \tag{256}$$

Moreover, $\tilde{g}_i(\mathbf{0}) = 0$, so the last operation uncomputes the auxiliary qubit.

12 Multivariate Optimal Polynomial Intersection

In this section we describe the application of DQI to the multivariate generalization of the OPI problem.

Definition 12.1. *Let r, u, m, q be integers where q is a prime power, $1 \leq u \leq (q-1)m$, and $1 \leq r < q$. For each $\mathbf{z} \in \mathbb{F}_q^m$ let $L(\mathbf{z}) \subset \mathbb{F}_q$ with $|L(\mathbf{z})| = r$. Given such subsets specified by explicit tables, the multivariate optimal polynomial intersection problem* mOPI(r, u, m, q) *is to find a polynomial $Q \in \mathbb{F}_q[z_1, \dots, z_m]$ of total degree at most u that maximizes the objective function f defined by*

$$f[Q] = \left|\left\{\mathbf{z} \in \mathbb{F}_q^m : Q(\mathbf{z}) \in L(\mathbf{z})\right\}\right| . \tag{257}$$

An instance of mOPI(r, u, m, q) is specified by the lists $L(\mathbf{z}) \subset \mathbb{F}_q$ for all $\mathbf{z} \in \mathbb{F}_q^m$. Specifying these requires q^{m+1} bits. Hence, throughout this section when we say "polynomial-time" we mean polynomial in q^m. The OPI problem is the special case of mOPI where $m = 1$ and q is a prime of polynomial magnitude.

We next recount some background information about Reed-Muller codes, which we will need to describe how DQI can be applied to the mOPI problem. The following exposition is based on [52].

For a given multivariate polynomial $Q(z_1, \ldots, z_m)$ over $\mathbb{F}_q$ let Eval(Q) be the symbol string in $\mathbb{F}_q^n$ obtained by evaluating Q at all possible assignments to $z_1, \ldots, z_m$ in lexicographical order. Hence,

$$n = q^m. \tag{258}$$

Let $\mathcal{P}_{q,m,u}$ be the set of polynomials in $\mathbb{F}_q[z_1, \ldots, z_m]$ of total degree at most u. In $\mathbb{F}_q$, raising a variable to power $q-1$ yields the identity. Thus, the number of distinct monomials from which elements of $\mathcal{P}_{q,m,u}$ can be constructed is

$$k = \left| \left\{ (i_1, \ldots, i_m) \ : \ 0 \leq i_j \leq q-2 \text{ and } \sum_{j=1}^{m} i_j \leq u \right\} \right|. \tag{259}$$

A polynomial in $\mathcal{P}_{q,m,u}$ is determined by choosing the coefficients from $\mathbb{F}_q$ for each of these k monomials. Hence, $|\mathcal{P}_{q,m,u}| = q^k$.

Definition 12.2. *Given a prime power q and integers u, m satisfying $1 \leq u \leq m(q-1)$, the corresponding Reed-Muller code* $\mathrm{RM}_q(u, m)$ *is* $\{\mathrm{Eval}(Q) : Q \in \mathcal{P}_{q,m,u}\}$.

We observe that $\mathrm{RM}_q(u, m)$ is an $\mathbb{F}_q$-linear code; it linearly maps from the k coefficients that define Q to the n values in Eval(Q).

As discussed in [53], the dual of a Reed-Muller code is also Reed-Muller code.

Theorem 12.1. *The dual code to* $\mathrm{RM}_q(u, m)$ *is* $\mathrm{RM}_q(u^\perp, m)$, *where*

$$u^\perp = m(q-1) - u - 1. \tag{260}$$

From [54], based on [55], we have the following.

Theorem 12.2. *Let α and β be the quotient and remainder obtained when dividing u by $q-1$. That is, $u = \alpha(q-1) + \beta$ with $0 \leq \beta < q-1$. Then the distance of* $\mathrm{RM}_q(u, m)$ *is $d = (q-\beta)q^{m-\alpha-1}$.*

As discussed in Chapter 13 of [56], the methods of [53] imply a polynomial-time classical reduction from decoding Reed-Muller codes to decoding Reed-Solomon codes, and therefore the following theorem holds.

Theorem 12.3. *The code* $\mathrm{RM}_q(u, m)$ *can be decoded from errors up to weight* $\left\lfloor \frac{d-1}{2} \right\rfloor$ *with perfect reliability by a polynomial time classical algorithm.*

From the above facts we see that DQI reduces the problem mOPI(r, u, m, q) to decoding of a Reed-Muller code $\mathrm{RM}_q(u^\perp, m)$, where $u^\perp = m(q-1) - u - 1$. Given an algorithm that can decode $\mathrm{RM}_q(u^\perp, m)$ out to ℓ errors, DQI will achieve an expected value of f given by the following theorem, which is a straightforward generalization of Lemma 6.2. (Here we include finite-size corrections since taking a limit of large problem size while keeping the ratio of constraints to variables fixed is not straightforward in the context of mOPI.)

Theorem 12.4. *Suppose we have an efficient algorithm that decodes $C^\perp = \mathrm{RM}_q(u^\perp, m)$ out to ℓ errors. Let $d^\perp$ be the distance of $C^\perp$. If $2\ell + 1 < d^\perp$, then for any instance of $\mathrm{GPR}(r, u, m, q)$, DQI produces in polynomial time samples from polynomials Q such that expected value of the objective $f[Q]$ is*

$$\langle f \rangle = q^m \frac{r}{q} + \frac{\sqrt{r(q-r)}}{q} \lambda_{\max}^{(q)} \tag{261}$$

where $\lambda_{\max}^{(q)}$ is the largest eigenvalue of the following $(\ell + 1) \times (\ell + 1)$ symmetric tridiagonal matrix

$$A^{(q)} = \begin{bmatrix} 0 & a_1 & & & \\ a_1 & d & a_2 & & \\ & a_2 & 2d & \ddots & \\ & & \ddots & & a_\ell \\ & & & a_\ell & \ell d \end{bmatrix} \tag{262}$$

with $a_k = \sqrt{k(q^m - k + 1)}$ and $d = \frac{q-2r}{\sqrt{r(q^m-r)}}$.

Together, theorems 12.4, 12.3, and 12.2 yield a strong performance guarantee for DQI applied to $\mathrm{mOPI}(r, u, m, q)$, particularly when $q \geq u$. Comparing this performance against competing classical algorithms remains for future work.

13 Resource Estimation for OPI

In this section we look at the resources required to construct a quantum circuit for syndrome decoding of Reed Solomon codes using the Berlekamp-Massey decoding algorithm [3]. Since the decoding step is the dominant cost in DQI, this gives us an estimate of the resource requirements for DQI to solve OPI. Whereas our example in §2 uses a ratio of ten constraints per variable, here we consider two constraints per variable, as this appears to be a more optimal choice for the purpose of solving classically-intractable instances of OPI using as few quantum gates and qubits as possible. That is, throughout this section, the number of constraints is $p - 1$ and the number of variables is $n \simeq p/2$.

An outline of the key steps of Berlekamp-Massey syndrome decoding algorithm is given in Algorithm 1. Readers seeking more detail can see Appendix E of [57], where the algorithm used here is explained in the context of BCH decoding. Asymptotically, the most computationally intensive step is the subroutine BerlekampMasseyLFSR, responsible for finding the shortest linear feedback shift register (LFSR). Standard irreversible implementations of this subroutine often rely on conditional branching and variable assignments that do not directly translate to efficient reversible circuits. Consequently, a naive reversible implementation of this subroutine can lead to significant overhead in terms of quantum resources, potentially undermining the overall efficiency of the DQI algorithm.

We address this challenge by presenting an optimized reversible implementation of the Berlekamp-Massey algorithm for finding the shortest LFSR in Algorithm 2. Our implementation is a generalization of the implementation given by [58], and works for any finite field $\mathbb{F}_q$. For a sequence of length n and a retroaction polynomial of maximum degree ℓ, Algorithm 2 can be implemented as a quantum circuit using $\mathcal{O}(n \cdot \ell)$ multiplications in $\mathbb{F}_q$ and using $2 \cdot (n + \ell) \cdot \lceil \log_2 q \rceil + n + \log_2 \ell$

	Subroutine Name	Subroutine Action	Toffoli	Ancilla	Reference
Arithmetic	quantum-classical addition	$\|x\rangle \to \|x+K\rangle$	n	n	[60]
	controlled quantum-classical addition	$\|c\rangle \|x\rangle \to \|c\rangle \|x+cK\rangle$	n	$2n$	[61]
	quantum-quantum addition	$\|x\rangle \|y\rangle \to \|x\rangle \|x+y\rangle$	n	n	[62]
	controlled quantum-quantum addition	$\|c\rangle \|x\rangle \|y\rangle \to \|x\rangle \|x+cy\rangle$	$2n$	n	[62]
Modular Arithmetic	modular quantum-classical addition	$\|x\rangle \to \|(x+K) \bmod P\rangle$	$2.5n$	n	[63]
	modular quantum-classical controlled addition	$\|c\rangle \|x\rangle \to \|c\rangle \|(x+cK) \bmod P\rangle$	$2.5n$	$2n$	[63]
	modular quantum-quantum addition	$\|x\rangle \|y\rangle \to \|x\rangle \|(x+y) \bmod P\rangle$	$3.5n$	n	[63]
	modular quantum-quantum controlled addition	$\|c\rangle \|x\rangle \|y\rangle \to \|c\rangle \|x\rangle \|(x+cy) \bmod P\rangle$	$4.5n$	n	[63]
	modular controlled scaled addition	$\|x\rangle \|y\rangle \to \|x\rangle \|(y+x*K) \bmod P\rangle$	$2.5n^2$	$2n$	[64]
	modular quantum-quantum multiplication	$\|x\rangle \|y\rangle \to \|x\rangle \|y\rangle \|(x*y) \bmod P\rangle$	$3n^2+2n-1$	$2n$	[65]
	modular multiplicative inverse	$\|x\rangle \to \|x^{-1} \bmod P\rangle$	$26n^2+2n$	$5n$	[65]

Table 1: Quantum circuit costs for modular arithmetic operations on n-bit operands in $\mathbb{F}_p$.

qubits. For finite fields $\mathbb{F}_p$, where p is a prime, we list the costs for performing modular arithmetic operations in Table 1. In our Zenodo record (`https://doi.org/10.5281/zenodo.13327870`) we provide an implementation of Algorithm 2 for prime fields $\mathbb{F}_p$ using Qualtran [59]. We present the resulting resource estimates in Table 2.

$RS_p(N, K)$	BerlekampMasseyLFSR$(n,\ \ell,\ \lceil \log_2 p \rceil)$	Toffoli	Clifford	Qubits
$RS_{67}(66, 34)$	$(32, 16, 7)$	999 850	5 153 696	907
$RS_{131}(130, 66)$	$(64, 32, 8)$	4 607 692	23 798 358	1912
$RS_{257}(256, 128)$	$(128, 64, 9)$	21 546 662	110 333 236	4101
$RS_{521}(520, 264)$	$(256, 128, 10)$	101 011 904	521 039 438	8850
$RS_{1031}(1030, 518)$	$(512, 256, 11)$	471 606 346	2 444 572 208	19 103
$RS_{2153}(2152, 1128)$	$(1024, 512, 12)$	2 186 280 548	11 380 033 666	41 132

Table 2: Cost of finding shortest linear feedback shift register (LFSR) using Algorithm 2, implemented and analyzed using Qualtran [59]. This is the most expensive step of Berlekamp-Massey syndrome decoding algorithm [3] for Reed Solomon codes, as presented in Algorithm 1. Here $n = N - K$ is the number of syndromes, which is equal to the length of the input sequence to Berlekamp-Massey algorithm, and $\ell = \frac{n}{2}$ is the maximum number of correctable errors, which is equal to the degree of retroaction polynomial.

As a point of comparison, we can estimate the classical cost of solving the instances in Table 2 by repeating Prange's algorithm with different size-n random subsets of the constraints until the target satisfaction fraction is reached. As an example, consider the case $p = 521$. Here, we have 260 variables and 521 constraints, of which DQI is able to satisfy 486. In a given trial of Prange's algorithm one uses Gaussian elimination to obtain a solution that is guaranteed to satisfy 260 of the constraints and which satisfies each of the remaining 261 constraints with probability $r/p = 1/2$. To beat DQI Prange's algorithm needs to satisfy at least 226 among these remaining

261 constraints. The probability of this on a given trial is

$$\frac{1}{2^{261}} \sum_{m=226}^{261} \binom{261}{m} \simeq 10^{-35}. \tag{263}$$

Thus solving this instance by the repeat-until-success version of Prange's algorithm should require on the order of 10^{35} repetitions. Therefore, assuming a CPU can execute on the order of a billion elementary operations per second this yields a total cost of at least 10^{26} CPU-seconds, even if each trial could be parallelized into a single clock-cycle.

As noted in [66], achieving such large separation between classical and quantum resource costs at reasonable problem size is only possible when the underlying source of advantage goes beyond quadratic speedups based on Grover's algorithm and its generalizations such as amplitude amplification and quantum walks. More specifically, in [67] the cost of solving random 14-SAT at the satisfiability phase transition using Grover's algorithm and quantum backtracking methods was estimated, and it was found that Grover's algorithm required lower resources, namely a T-depth of 10^{14} and a Toffoli count of 10^{19} on instances that they estimate would require 10^{10} CPU-seconds classically.

Although the repeat-until-success version of Prange's algorithm is a standard technique that provides a useful point of comparison, we do not specifically claim it to be optimal. In §8.6 and §8.7 we have surveyed all of the classical algorithms that we can find in the literature applicable to OPI and find that none yield polynomial-time solutions in the parameter regime discussed here. Nevertheless, there may be classical techniques to achieve better scaling than Prange's algorithm, while still requiring exponential time; such incremental improvements to exponential runtimes are quite common in the cryptanalysis literature (see for example [68]). We leave the investigation of this possibility to future work.

Algorithm 1 Syndrome decoding of $RS_q(\vec{\gamma}, N, K)$ using Berlekamp Massey algorithm

```
Input: A list $s = [s_1, s_2, \ldots, s_n]$ of syndromes where $n = N - K$.
Output: A list $e = [e_1, e_2, \ldots, e_N]$ of error values such that $|\mathrm{supp}(e)| \leq \ell$ where $\ell = \frac{n}{2}$
▷ Step-1: Find error locator polynomial $\sigma(Z)$ of degree $\leq \ell$, by solving for shortest linear feedback shift register (LFSR) using Berlekamp-Massey algorithm.
▷ Uses $\mathcal{O}(N^2)$ multiplications and $\mathcal{O}(N)$ inversions of elements in $\mathbb{F}_q$.
$\sigma(Z) \leftarrow \mathrm{BerlekampMasseyLFSR}(s, \ell)$
▷ Step-2: Find error evaluator polynomial $\Omega(Z)$ via fast polynomial multiplication using Number Theoretic Transform (NTT).
▷ Uses $\mathcal{O}(N \log N)$ multiplications of elements in $\mathbb{F}_q$.
$\Omega(Z) \leftarrow \sigma(Z) \times S(Z) \mod Z^{n+1}$
▷ Step-3: Find roots of $\sigma(Z)$ to determine locations of errors by evaluating $\sigma(Z)$ for all $N$ roots of unity $(\vec{\gamma})$ using NTT.
▷ Uses $\mathcal{O}(N \log N)$ multiplications of elements in $\mathbb{F}_q$.
sigma_roots $\leftarrow$ chien_search$(\sigma(Z), \vec{\gamma})$
▷ Step-4: Use evaluations of $\Omega(Z)$ and $\sigma'(Z)$ for all N roots of unity $(\vec{\gamma})$ to determine the values of errors $e$.
▷ Uses $\mathcal{O}(N \log N)$ multiplications and $\mathcal{O}(N)$ divisions of elements in $\mathbb{F}_q$.
$e \leftarrow$ forneys_algorithm$(\Omega(Z), \sigma(Z), \text{sigma\_roots}, \vec{\gamma})$
```

Algorithm 2 Reversible Berlekamp-Massey algorithm to find shortest LFSR over $\mathbb{F}_q$.

```
Input: A list $s = [s_0, s_1, \ldots, s_{n-1}]$ in $\mathbb{F}_q$ and an integer $\ell \leq n$.
Output: Retroaction polynomial $C(X) \in \mathbb{F}_q[X]$ such that $\deg(C(X)) \leq \ell$.
Storage: Register for $s$ ($n$ elements in $\mathbb{F}_q$) and $C(X)$ ($\ell$ elements in $\mathbb{F}_q$)
Garbage: Register for $L$ ($\log_2 \ell$ bits), $B(X)$ ($\ell$ elements in $\mathbb{F}_q$), $d = [d_0, d_2, \ldots, d_{n-1}]$ ($n$ elements in $\mathbb{F}_q$) and $v = [v_0, v_2, \ldots, v_{n-1}]$ ($n$ bits)
Total Qubits: $2 \times (n + \ell) \times \lceil \log_2 q \rceil + \log_2(\ell) + n$
$c \leftarrow [-1]$    ▷ $c$ and $b$ correspond to list of coefficients for $C(X)$ and $B(X)$; $C(X) = \sum_{i=0}^{\ell} c_i x^i$
$b \leftarrow [-1]$    ▷ $-1$ is the additive inverse of 1 in $F_q$
$L \leftarrow 0$
for $i$ in $0, \ldots, N-1$ do
    $d_i \leftarrow \sum_{j=0}^{\mathrm{len}(c)} s_{i-j} \times c_j$    ▷ $\leq \ell + 1$ quantum-quantum multiplications and additions
    $v_i \leftarrow (2L \leq i)$    ▷ quantum-classical comparator to decide between case 2 and case 3
    if $i < \ell$ then
        $c \leftarrow \mathrm{concatenate}(c, [0])$    ▷ Extends register for $C(X)$ upto a maximum length $\ell + 1$
        $b \leftarrow \mathrm{concatenate}([0], b)$    ▷ Equivalent to performing $B(X) \leftarrow X \cdot B(X)$
    else
        $b \leftarrow \mathrm{roll}(b, 1)$    ▷ Equivalent to performing $B(X) \leftarrow X \cdot B(X)$
    end if
    $r \leftarrow (1$ if $d_i = 0$ else $d_i)$
    $b \leftarrow b \times r$    ▷ $\leq \ell + 1$ in-place quantum-quantum multiplications
    if $d_i \neq 0$ then
        $c \leftarrow c - b$    ▷ $\leq \ell + 1$ quantum-quantum controlled subtractions
    end if
    if $d_i \neq 0$ and $v_i$ then
        $L \leftarrow i + 1 - L$
        $b \leftarrow b + c$    ▷ $\leq \ell + 1$ quantum-quantum controlled additions
    end if
    $b \leftarrow b \times r^{-1}$    ▷ $\leq \ell + 1$ in-place quantum-quantum multiplications and 1 modular inverse
end for
return c
```

A Elementary Symmetric Polynomials

In this section we prove Lemma A.1. Recall that our max-LINSAT objective function takes the form $f(\mathbf{x}) = f_1(\mathbf{b}_1 \cdot \mathbf{x}) + \ldots + f_m(\mathbf{b}_m \cdot \mathbf{x})$, where $f_j : \mathbb{F}_p \to \{+1, -1\}$. Hence $f_j^2 - 1 = 0$ for all j. Consequently, the rescaled functions $g_j(x) = (f_j(x) - \overline{f})/\varphi$ also obey a quadratic identity that does not depend on j. (Namely, $\varphi^2 g_j^2 + 2\overline{f}\varphi g + \overline{f}^2 - 1 = 0$.) Thus the conditions of Lemma A.1 are met by both $f_1, \ldots, f_m$ and $g_1, \ldots, g_m$.

Lemma A.1. *Let P be any degree-ℓ polynomial in a single variable. Let $x_1, \ldots, x_m$ be variables that each obey a quadratic identity $ax_j^2 + bx_j + c = 0$, where a, b, c are independent of j and $a \neq 0$. Then $P(x_1 + \ldots + x_m)$ can be expressed as a linear combination of elementary symmetric polynomials: $P(x_1 + \ldots + x_m) = \sum_{k=0}^{\ell} u_k P^{(k)}(x_1, \ldots, x_m)$.*

Proof. Reducing modulo the quadratic identities:

$$ax_j^2 + bx_j + c = 0 \quad j = 1, \ldots, m \tag{264}$$

brings $P(x_1 + \ldots + x_m)$ into a form where none of $x_1, \ldots, x_m$ is raised to any power greater than one. We can think of the resulting expression as a multilinear multivariate polynomial $P(x_1, \ldots, x_m)$, which is of total degree ℓ. Since the coefficients in the m identities described in (264) are independent of j, this multilinear multinomial is symmetric. That is, for any permutation $\pi \in S_m$ we have $P(x_{\pi(1)}, \ldots, x_{\pi(m)}) = P(x_1, \ldots, x_m)$. By the fundamental theorem of symmetric polynomials, multilinearity and symmetry together imply that the degree-ℓ polynomial $P(x_1, \ldots, x_m)$ can be expressed as a linear combination of elementary symmetric polynomials of degree at most ℓ. □

B Gallager's Ensemble

In [69] Gallager defined the following ensemble of matrices over $\mathbb{F}_2$. This ensemble is widely used in coding theory because when parity check matrices are drawn from Gallager's ensemble, the resulting LDPC codes have good parameters with high probability. Together with a choice of $\mathbf{v}$, a matrix B sampled from Gallager's ensemble also induces a natural ensemble of D-regular max-k-XORSAT instances.

Given parameters (k, D, b), a sample from Gallager's ensemble of matrices $B \in \mathbb{F}_2^{bk \times bD}$ is generated as follows. Let A denote the horizontal concatenation of D identity matrices, each $b \times b$, yielding $A = [I_1 I_2 \ldots I_D]$. For $i = 1, \ldots, k$ let $M_i = AP_i$ where $P_1, \ldots, P_k$ are independent uniformly random $bD \times bD$ permutation matrices. Concatenate these vertically yielding

$$B^T = \left[\begin{array}{c} M_1 \\ \hline M_2 \\ \hline \vdots \\ \hline M_k \end{array}\right] \tag{265}$$

The matrix B^T thus has $n = bk$ rows and $m = bD$ columns, with k ones in each row and D ones in each column. By choosing each function $f_1, \ldots, f_m$ independently at random to be either $f_i(x) = (-1)^x$ or $f_i(x) = -(-1)^x$ we obtain a D-regular instance of max-k-XORSAT as a special case of max-LINSAT over $\mathbb{F}_2$.

Gallager's (k, D, b) ensemble is not equivalent to sampling uniformly from all $bk \times bD$ matrices with k ones in each row and D ones in each column. It shares many properties with such a distribution but is more convenient to sample from.

C Simulated Annealing Applied to OPI

By the results of §8.1 we expect simulated annealing to yield solutions to the OPI instances of §2 where the fraction $\phi_{\max}$ of constraints satisfied scales like

$$\phi_{\max} \simeq \frac{1}{2} + \frac{c}{D^{\nu}}. \tag{266}$$

where c and ν are free parameters of the fit. According to our crude theoretical model ν should be 1/2. However, extrapolating from our empirical results with sparse instances we would expect ν to be slightly smaller than 1/2. As shown in Fig. 12, experimental results match well to this prediction with $\nu = 0.45$. In OPI every variable is contained in every constraint so the degree D is equal to the number of constraints m.

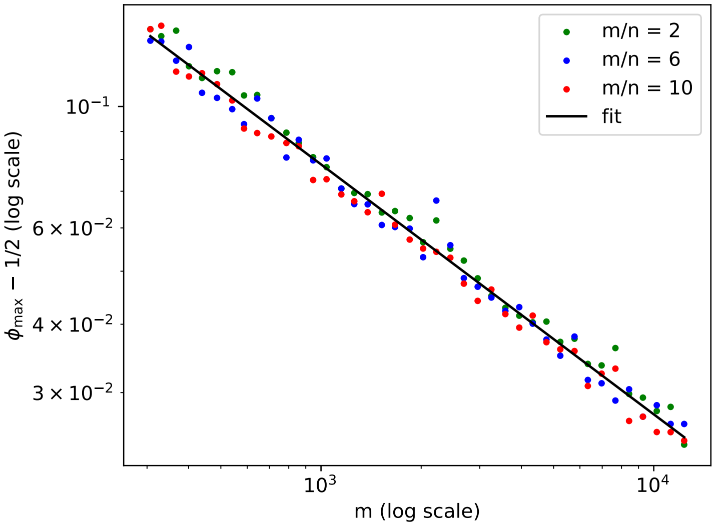

Figure 12: Here we generate OPI instances over $\mathbb{F}_p$ where p takes prime values from 307 to 12, 343. The number of constraints is $m = p - 1$. For each m we take $n \in \{m/2, m/6, m/10\}$, rounded to the nearest integer. We find that, independent of n/m, the approximation achieved by simulated annealing with 10, 000 sweeps fits well to $\phi_{\max} = 1/2 + 1.8D^{-0.45}$. Note that, in OPI the degree D equals the number of constraints m, since every variable is contained in every constraint.

References

[1] Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 69–74. IEEE, 2022.

[2] Elwyn Berlekamp, Robert McEliece, and Henk Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information theory*, 24(3):384–386, 1978.

[3] Elwyn R. Berlekamp. *Algebraic coding theory (revised edition).* World Scientific, 2015.

[4] Moni Naor and Benny Pinkas. Oblivious transfer and polynomial evaluation. In *Proceedings of the thirty-first annual ACM Symposium On Theory of Computing (STOC)*, pages 245–254, 1999.

[5] Daniel Bleichenbacher and Phong Q. Nguyen. Noisy polynomial interpolation and noisy Chinese remaindering. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 53–69. Springer, 2000.

[6] Moni Naor and Benny Pinkas. Oblivious polynomial evaluation. *SIAM Journal on Computing*, 35(5):1254–1281, 2006.

[7] Venkatesan Guruswami and Alexander Vardy. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. *IEEE Transactions on Information Theory*, 51(7):2249–2256, 2005.

[8] Venkata Gandikota, Badih Ghazi, and Elena Grigorescu. NP-hardness of Reed-Solomon decoding, and the Prouhet-Tarry-Escott problem. *SIAM Journal on Computing*, 47(4):1547–1584, 2018.

[9] Qi Cheng and Daqing Wan. On the list and bounded distance decodability of Reed-Solomon codes. *SIAM Journal on Computing*, 37(1):195–209, 2007.

[10] Qi Cheng and Daqing Wan. Complexity of decoding positive-rate Reed-Solomon codes. In *International Colloquium on Automata, Languages, and Programming*, pages 283–293. Springer, 2008.

[11] Robert Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962.

[12] Thomas J. Richardson and Rüdiger L. Urbanke. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Transactions on Information Theory*, 47(2):599–618, 2001.

[13] Mark Mézard and Andrea Montanari. *Information, Physics, and Computation.* Oxford University Press, 2009.

[14] Johan Håstad. On bounded occurrence constraint satisfaction. *Information Processing Letters*, 74(1-2):1–6, 2000.

[15] Boaz Barak, Ankur Moitra, Ryan O'Donnell, Prasad Raghavendra, Oded Regev, David Steurer, Luca Trevisan, Aravindan Vijayaraghavan, David Witmer, and John Wright. Beating the random assignment on constraint satisfaction problems of bounded degree. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, volume 40 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 110–123, 2015. arXiv:1505.03424.

[16] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 20–29, 2003. arXiv:quant-ph/0301023.

[17] Dorit Aharonov and Oded Regev. Lattice problems in NP ∩ coNP. *Journal of the ACM (JACM)*, 52(5):749–765, 2005.

[18] Yilei Chen, Qipeng Liu, and Mark Zhandry. Quantum algorithms for variants of average-case lattice problems via filtering. In *Annual international conference on the theory and applications of cryptographic techniques (EUROCRYPT)*, pages 372–401. Springer, 2022.

[19] Thomas Debris-Alazard, Maxime Remaud, and Jean-Pierre Tillich. Quantum reduction of finding short code vectors to the decoding problem. *IEEE Transactions on Information Theory*, 2023.

[20] Alexander Schmidhuber, Ryan O'Donnell, Robin Kothari, and Ryan Babbush. Quartic quantum speedups for planted inference. *Physical Review X*, 15(2):021077, 2025. arXiv:2406.19378.

[21] André Chailloux and Jean-Pierre Tillich. The quantum decoding problem. In *19th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2024)*, pages 6–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024. arXiv:2310.20651.

[22] Eugene Prange. The use of information sets in decoding cyclic codes. In *IRE Transactions on Information Theory*, volume 8, pages 5–9, 1962.

[23] André Chailloux and Jean-Pierre Tillich. Quantum advantage from soft decoders. *arXiv:2411.12553*, 2024.

[24] Natchapol Patamawisut, Naphan Benchasattabuse, Michal Hajdušek, and Rodney Van Meter. Quantum circuit design for decoded quantum interferometry. *arXiv:2504.18334*, 2025.

[25] Guang Hao Low, Vadym Kliuchnikov, and Luke Schaeffer. Trading T-gates for dirty qubits in state preparation and unitary synthesis. *Quantum*, 8:1375, 2024. arXiv:1812.00954.

[26] Andreas Bartschi and Stephan Eidenbenz. Short-depth circuits for Dicke state preparation. In *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2022. arXiv:2207.09998.

[27] Hanyu Wang, Bochen Tan, Jason Cong, and Giovanni De Micheli. Quantum state preparation using an exact CNOT synthesis formulation. *arXiv:2401.01009*, 2024.

[28] Alexander M. Barg and Dmitry Yu. Nogin. Spectral approach to linear programming bounds on codes. *Problems of Information Transmission*, 42(2):77–89, 2006.

[29] Kunal Marwaha and Stuart Hadfield. Bounds on approximating Max-k-XOR with quantum and classical local algorithms. *Quantum*, 6:757, 2022.

[30] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. *arXiv:1412.6062*, 2014.

[31] Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou. The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington-Kirkpatrick model. In François Le Gall and Tomoyuki Morimae, editors, *17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)*, volume 232 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:21, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[32] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv:1411.4028*, 2014.

[33] Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou. Performance of the QAOA on MaxCut over large-girth regular graphs. github.com/benjaminvillalonga/large-girth-maxcut-qaoa/blob/main/data.csv, 2022.

[34] Sami Boulebnane, Abid Khan, Minzhao Liu, Jeffrey Larson, Dylan Herman, Ruslan Shaydulin, and Marco Pistoia. Evidence that the quantum approximate optimization algorithm optimizes the Sherrington-Kirkpatrick model efficiently in the average case. *arXiv:2505.07929*, 2025.

[35] Christophe Piveteau and Joseph M. Renes. Quantum message-passing algorithm for optimal and efficient decoding. *Quantum*, 6:784, 2022.

[36] Edward Farhi, Sam Gutmann, Daniel Ranard, and Benjamin Villalonga. Lower bounding the MaxCut of high girth 3-regular graphs using the QAOA. *arXiv:2503.12789*, 2025.

[37] Venkatesan Guruswami and Madhu Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 28–37. IEEE, 1998.

[38] Arjen K. Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982.

[39] Claus-Peter Schnorr. A hierarchy of polynomial time lattice basis reduction algorithms. *Theoretical Computer Science*, 53(2-3):201–224, 1987.

[40] Claus-Peter Schnorr and Martin Euchner. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming*, 66:181–199, 1994.

[41] Thomas J. Richardson, Mohammad Amin Shokrollahi, and Rüdiger L. Urbanke. Design of capacity-approaching irregular low-density parity-check codes. *IEEE Transactions on Information Theory*, 47(2):619–637, 2001.

[42] Jon Feldman, Martin J. Wainwright, and David R. Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51(3):954–972, 2005.

[43] Jon Feldman, Tal Malkin, Rocco A. Servedio, Cliff Stein, and Martin J. Wainwright. LP decoding corrects a constant fraction of errors. *IEEE Transactions on Information Theory*, 53(1):82–89, 2007.

[44] Constantinos Daskalakis, Alexandros G. Dimakis, Richard M. Karp, and Martin J. Wainwright. Probabilistic analysis of linear programming decoding. *IEEE Transactions on Information Theory*, 54(8):3565–3578, 2008.

[45] Mohammad H. Taghavi and Paul H. Siegel. Adaptive methods for linear programming decoding. *IEEE Transactions on Information Theory*, 54(12):5396–5410, 2008.

[46] Alex Yufit, Asi Lifshitz, and Yair Be'ery. Efficient linear programming decoding of HDPC codes. *IEEE Transactions on Communications*, 59(3):758–766, 2010.

[47] Akin Tanatmis, Stefan Ruzika, Horst W. Hamacher, Mayur Punekar, Frank Kienle, and Norbert Wehn. Valid inequalities for binary linear codes. In *2009 IEEE International Symposium on Information Theory (ISIT)*, pages 2216–2220. IEEE, 2009.

[48] Akin Tanatmis, Stefan Ruzika, Horst W. Hamacher, Mayur Punekar, Frank Kienle, and Norbert Wehn. A separation algorithm for improved LP decoding of linear block codes. *IEEE Transactions on Information Theory*, 56(7):3277–3289, 2010.

[49] Chris Calabro. *The exponential complexity of satisfiability problems*. Phd thesis, University of California, San Diego, 2009. cseweb.ucsd.edu/~ccalabro/thesis.pdf.

[50] Laurent Decreusefond and Gilles Zémor. On the error-correcting capabilities of cycle codes of graphs. *Combinatorics, Probability and Computing*, 6(1):27–38, 1997.

[51] Jack Edmonds and Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Mathematical programming*, 5:88–124, 1973.

[52] Emmanuel Abbe, Amir Shpilka, and Min Ye. Reed–Muller codes: Theory and algorithms. *IEEE Transactions on Information Theory*, 67(6):3251–3277, 2020.

[53] Ruud Pellikaan and Xin-Wen Wu. List decoding of q-ary Reed-Muller codes. *IEEE Transactions on Information Theory*, 50(4):679–682, 2004.

[54] Philippe Delsarte, Jean-Marie Goethals, and F. J. MacWilliams. On generalized Reed-Muller codes and their relatives. *Information and Control*, 16(5):403–442, 1970.

[55] Tadao Kasami, Shu Lin, and W. Peterson. New generalizations of the Reed-Muller codes–I: Primitive codes. *IEEE Transactions on information theory*, 14(2):189–199, 1968.

[56] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. *Essential Coding Theory*. cse.buffalo.edu/faculty/atri/courses/coding-theory/book/, 2023.

[57] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM journal on computing*, 38(1):97–139, 2008.

[58] Clémence Chevignard, Pierre-Alain Fouque, and André Schrottenloher. Reducing the number of qubits in quantum information set decoding. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 299–329. Springer, 2024.

[59] Matthew P. Harrigan, Tanuj Khattar, Charles Yuan, Anurudh Peduri, Noureldin Yosri, Fionn D. Malone, Ryan Babbush, and Nicholas C. Rubin. Expressing and analyzing quantum algorithms with qualtran. *arXiv:2409.04643*, 2024.

[60] Dmytro Fedoriaka. New circuit for quantum adder by constant. *arXiv:2501.07060*, 2025.

[61] Thomas Häner, Samuel Jaques, Michael Naehrig, Martin Roetteler, and Mathias Soeken. Improved quantum circuits for elliptic curve discrete logarithms. In *Post-Quantum Cryptography*, pages 425–444. Springer, 2020. arXiv:2001.09580.

[62] Craig Gidney. Halving the cost of quantum addition. *Quantum*, 2:74, June 2018.

[63] Alessandro Luongo, Antonio Michele Miti, Varun Narasimhachar, and Adithya Sireesh. Measurement-based uncomputation of quantum circuits for modular arithmetic. *arXiv:2407.20167*, 2024.

[64] Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, April 2021.

[65] Daniel Litinski. How to compute a 256-bit elliptic curve private key with only 50 million Toffoli gates. *arXiv:2306.08585*, 2023.

[66] Ryan Babbush, Jarrod R. McClean, Michael Newman, Craig Gidney, Sergio Boixo, and Hartmut Neven. Focus beyond quadratic speedups for error-corrected quantum advantage. *PRX quantum*, 2(1):010103, 2021.

[67] Earl Campbell, Ankur Khurana, and Ashley Montanaro. Applying quantum algorithms to constraint satisfaction problems. *Quantum*, 3:167, 2019.

[68] Daniel J. Bernstein, Tung Chou, Tanja Lange, Ingo von Maurich, Rafael Misoczki, Ruben Niederhagen, Eduardo Persichetti, Christiane Peters, Peter Schwabe, Nicolas Sendrier, et al. Classic McEliece: conservative code-based cryptography. *NIST submissions*, 1(1):1–25, 2017.

[69] Robert G. Gallager. *Low-Density Parity-Check Codes.* MIT Press, 1963.