Article

Optimization by decoded quantum interferometry

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

Received: 4 December 2024

Accepted: 13 August 2025

Published online: 22 October 2025

Open access

Check for updates

Stephen P. Jordan[1]✉, Noah Shutty[1]✉, Mary Wootters[2,3], Adam Zalcman[1], Alexander Schmidhuber[1,4], Robbie King[1,5], Sergei V. Isakov[1], Tanuj Khattar[1] & Ryan Babbush[1]

Achieving superpolynomial speed-ups for optimization has long been a central goal for quantum algorithms[1]. Here we introduce decoded quantum interferometry (DQI), a quantum algorithm that uses the quantum Fourier transform to reduce optimization problems to decoding problems. When approximating optimal polynomial fits over finite fields, DQI achieves a superpolynomial speed-up over known classical algorithms. The speed-up arises because the algebraic structure of the problem is reflected in the decoding problem, which can be solved efficiently. We then investigate whether this approach can achieve a speed-up for optimization problems that lack an algebraic structure but have sparse clauses. These problems reduce to decoding low-density parity-check codes, for which powerful decoders are known[2,3]. To test this, we construct a max-XORSAT instance for which DQI finds an approximate optimum substantially faster than general-purpose classical heuristics, such as simulated annealing. Although a tailored classical solver can outperform DQI on this instance, our results establish that combining quantum Fourier transforms with powerful decoding primitives provides a promising new path towards quantum speed-ups for hard optimization problems.

NP-hardness results indicate that finding exact optima and even sufficiently good approximate optima for worst-case instances of many optimization problems is probably out of reach for polynomial-time algorithms both classical and quantum[4]. Nevertheless, there remain combinatorial optimization problems, such as the closest vector problem, for which there is a large gap between the best approximation achieved by a polynomial-time classical algorithm[5] and the strongest complexity-theoretic inapproximability result[6]. When considering average-case complexity, such gaps become more prevalent, as few average-case inapproximability results are known. These gaps present a potential opportunity for quantum computers, namely achieving in polynomial time an approximation that requires superpolynomial time to achieve using known classical algorithms.

Quantum algorithms for combinatorial optimization have been the subject of intense research over the last three decades[7–13], which has uncovered some evidence of a possible superpolynomial quantum speed-up for certain optimization problems[14–20]. Nevertheless, the problem of finding a superpolynomial quantum advantage for optimization is extremely challenging and remains largely open.

Here we propose a quantum algorithm for optimization that uses interference patterns as its main underlying principle. We call this algorithm decoded quantum interferometry (DQI). DQI uses a quantum Fourier transform to arrange that amplitudes interfere constructively on symbol strings for which the objective value is large, thereby enhancing the probability of obtaining good solutions upon measurement. Most previous approaches to quantum optimization have been Hamiltonian-based[7,8], with a notable exception being the superpolynomial speed-up due to Chen, Liu and Zhandry[16] for finding short lattice vectors, which uses Fourier transforms and can be seen as an ancestor of DQI. Whereas Hamiltonian-based quantum optimization methods are often regarded as exploiting the local structure of the optimization landscape (for example, tunnelling across barriers[21]), our approach instead exploits the sparsity that is routinely present in the Fourier spectrum of the objective functions for combinatorial optimization problems, and it can also exploit more elaborate structure in the spectrum if present.

Before presenting evidence that DQI can efficiently obtain approximate optima not achievable by known polynomial-time classical algorithms, we quickly illustrate the essence of the DQI algorithm by applying it to max-XORSAT. We use max-XORSAT as our first example because, although it is not the problem on which DQI has achieved its greatest success, it is the context in which DQI is simplest to explain.

Given an $m \times n$ matrix B with $m > n$, the max-XORSAT problem is to find an n-bit string $\mathbf{x}$ satisfying as many as possible of the m linear mod-2 equations $B\mathbf{x} = \mathbf{v}$. As we are working modulo 2, we regard all entries of the matrix B and the vectors $\mathbf{x}$ and $\mathbf{v}$ as coming from the finite field $\mathbb{F}_2$. The max-XORSAT problem can be rephrased as maximizing the objective function

$$f(\mathbf{x}) = \sum_{i=1}^{m} (-1)^{v_i + \mathbf{b}_i \cdot \mathbf{x}}, \tag{1}$$

[1]Google Quantum AI, Venice, CA, USA. [2]Department of Computer Science, Stanford University, Stanford, CA, USA. [3]Department of Electrical Engineering, Stanford University, Stanford, CA, USA. [4]Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, CA, USA. [5]Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA, USA. ✉e-mail: stephenjordan@google.com; shutty@google.com

where $\mathbf{b}_i$ is the ith row of B and v_i is the ith entry of $\mathbf{v}$. Thus, $f(\mathbf{x})$ is the number of the m linear equations that are satisfied minus the number unsatisfied.

From equation (1), one can see that the Hadamard transform of f is extremely sparse: it has m non-zero amplitudes, which are on the strings $\mathbf{b}_1, \ldots, \mathbf{b}_m$. The state $\sum_{\mathbf{x}\in\mathbb{F}_2^n} f(\mathbf{x})|\mathbf{x}\rangle$ is, thus, easy to prepare. Simply prepare the superposition $\sum_{i=1}^{m}(-1)^{v_i}|\mathbf{b}_i\rangle$ and apply the quantum Hadamard transform. (Here, for simplicity, we have omitted normalization factors). Measuring the state $\sum_{\mathbf{x}\in\mathbb{F}_2^n} f(\mathbf{x})|\mathbf{x}\rangle$ in the computational basis yields a biased sample, where a string $\mathbf{x}$ is obtained with probability proportional to $f(\mathbf{x})^2$, which slightly enhances the likelihood of obtaining strings with a large objective value relative to uniform random sampling.

To obtain stronger enhancement, DQI prepares states of the form

$$|P(f)\rangle = \sum_{\mathbf{x}\in\mathbb{F}_2^n} P(f(\mathbf{x}))|\mathbf{x}\rangle, \tag{2}$$

where P is an appropriately normalized degree-ℓ polynomial. The Hadamard transform of such a state always takes the form

$$\sum_{k=0}^{\ell} \frac{w_k}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_2^m \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v}\cdot\mathbf{y}}|B^{\mathrm{T}}\mathbf{y}\rangle, \tag{3}$$

for some coefficients $w_0, \ldots, w_\ell$. Here $|\mathbf{y}|$ denotes the Hamming weight of the bit string $\mathbf{y}$. The DQI algorithm prepares $|P(f)\rangle$ in five steps. The first step is to prepare the superposition $\sum_{k=0}^{\ell} w_k|D_{m,k}\rangle$, where

$$|D_{m,k}\rangle = \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_2^m \\ |\mathbf{y}|=k}} |\mathbf{y}\rangle \tag{4}$$

is the Dicke state of weight k. Preparing such superpositions over Dicke states can be done with $\mathcal{O}(m^2)$ quantum gates using the techniques in refs. 22,23. Second, the phase $(-1)^{\mathbf{v}\cdot\mathbf{y}}$ is imposed by applying the product $Z_1^{v_1}\otimes\ldots\otimes Z_m^{v_m}$, where Z_m is the Pauli-Z operator acting on the mth qubit. Third, the quantity $B^{\mathrm{T}}\mathbf{y}$ is computed into an ancilla register using a reversible circuit for matrix multiplication. This yields the state

$$\sum_{k=0}^{\ell} \frac{w_k}{\sqrt{\binom{m}{k}}} \sum_{\substack{\mathbf{y}\in\mathbb{F}_2^m \\ |\mathbf{y}|=k}} (-1)^{\mathbf{v}\cdot\mathbf{y}}|\mathbf{y}\rangle|B^{\mathrm{T}}\mathbf{y}\rangle. \tag{5}$$

The fourth step is to use the value $B^{\mathrm{T}}\mathbf{y}$ to infer $\mathbf{y}$, which can then be subtracted from $|\mathbf{y}\rangle$, thereby bringing it back to the all zeros state, which can be discarded. (This is known as 'uncomputation'[24]). The fifth and final step is to apply a Hadamard transform to the remaining register, yielding $|P(f)\rangle$. This sequence of steps is illustrated in Fig. 1.

The fourth step, in which $|\mathbf{y}\rangle$ is uncomputed, is not straightforward because B is a non-square matrix and, thus, inferring $\mathbf{y}$ from $B^{\mathrm{T}}\mathbf{y}$ is an underdetermined linear algebra problem. However, we also know that $|\mathbf{y}| \le \ell$. The problem of solving this underdetermined linear system with a Hamming weight constraint is precisely the syndrome decoding problem for the classical error-correcting code $C^{\perp} = \{\mathbf{d}\in\mathbb{F}_2^m : B^{\mathrm{T}}\mathbf{d} = \mathbf{0}\}$ with up to ℓ errors.

In general, syndrome decoding is an NP-hard problem[25]. However, when B is very sparse or has certain kinds of algebraic structure, the decoding problem can be solved by polynomial-time classical algorithms even when ℓ is large (for example, linear in m). By solving this decoding problem using a reversible implementation of such a classical decoder, one uncomputes $|\mathbf{y}\rangle$ in the first register. If the decoding algorithm requires T quantum gates, then the number of gates required to prepare $|P(f)\rangle$ is $\mathcal{O}(T+m^2)$.

Approximate solutions to the optimization problem are obtained by measuring $|P(f)\rangle$ in the computational basis. The higher the degree of the polynomial in $|P(f)\rangle$, the greater one can bias the measured bit strings towards solutions with a large objective value. However, this requires solving a harder decoding problem, as the maximum number of errors is equal to the degree of P. Next, we summarize how, by making an optimal choice of P and a judicious choice of decoder, DQI can be a powerful optimizer for some classes of problems.

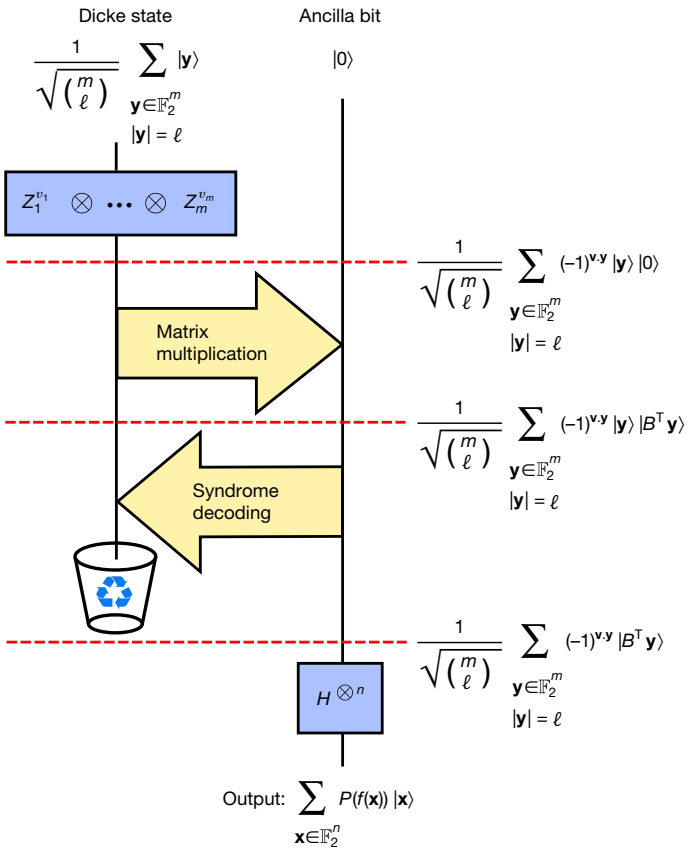

Fig. 1 | Schematic illustration of DQI. As the initial Dicke state is of weight ℓ, the final polynomial P is of degree ℓ. Here, for simplicity, we take $w_\ell = 1$ and $w_k = 0$ for all $k \neq \ell$. Recycling icon adapted from FreeSVG (https://freesvg.org) under a CC0 1.0 Universal Public Domain licence.

Although DQI can be applied more broadly, the most general optimization problem that we apply DQI to in this paper is max-LINSAT, which we define as follows.

Definition 1. Let $\mathbb{F}_p$ be a finite field and let $B \in \mathbb{F}_p^{m\times n}$. For each $i = 1, \ldots, m$, let $F_i \subset \mathbb{F}_p$ be an arbitrary subset of $\mathbb{F}_p$, which yields a corresponding constraint $\sum_{j=1}^{n} B_{ij}x_j \in F_i$. The max-LINSAT problem is to find $\mathbf{x}\in\mathbb{F}_p^n$ satisfying as many as possible of these m constraints.

We focus primarily on the case that p has at most polynomially large magnitude and the subsets $F_1, \ldots, F_m$ are given as explicit lists. The max-XORSAT problem is the special case where $p = 2$ and $|F_i| = 1$ for all i.

Consider a max-LINSAT instance where the sets $F_1, \ldots, F_m$ each have size r. Let $\langle s\rangle$ be the expected number of constraints satisfied by the symbol string sampled in the final measurement of the DQI algorithm. Suppose we have a polynomial-time algorithm that can correct up to ℓ bit flip errors on codewords from the code $C^{\perp} = \{\mathbf{d}\in\mathbb{F}_p^m : B^{\mathrm{T}}\mathbf{d} = \mathbf{0}\}$. Then, in polynomial time, DQI achieves the following approximate optimum to the max-LINSAT problem:

$$\frac{\langle s\rangle}{m} = \left(\sqrt{\frac{\ell}{m}\left(1-\frac{r}{p}\right)} + \sqrt{\frac{r}{p}\left(1-\frac{\ell}{m}\right)}\right)^2, \tag{6}$$

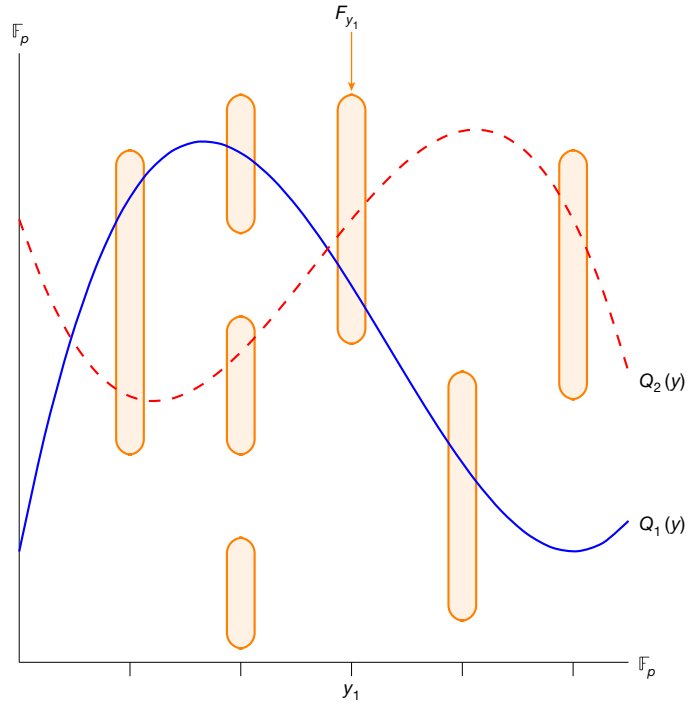

Fig. 2 | Illustration of OPI problem. A stylized example of the OPI problem. For $y_1 \in \mathbb{F}_p$, the orange set above the point y_1 represents F_{y_1}. Both polynomials $Q_1(y)$ and $Q_2(y)$ represent solutions that have a large objective value, as they each intersect all but one set F_y.

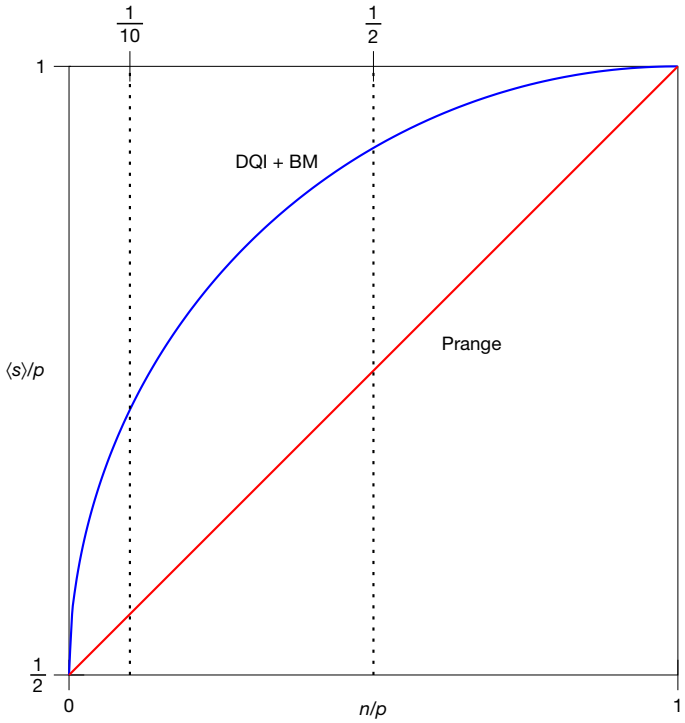

Fig. 3 | Approximate optima for OPI. Plot of the expected fraction $\langle s \rangle / p$ of satisfied constraints achieved by DQI with the Berlekamp–Massey decoder and by Prange's algorithm for the OPI problem in the balanced case $r/p = 1/2$, as a function of the ratio of variables to constraints n/p. At $n/p = 1/10$, Prange's algorithm satisfies a fraction 0.55 of the clauses whereas DQI satisfies $\langle s \rangle / p = 1/2 + \sqrt{19}/20 \approx 0.7179$. As a concrete challenge to the classical algorithms community, we propose matching or exceeding this value in polynomial time. In our concrete resource estimation, we consider $n/p = 1/2$, where OPI achieves $\langle s \rangle / p = 1/2 + \sqrt{3}/4 \approx 0.9330$ and Prange's algorithm achieves 0.75. BM, Berlekamp–Massey decoder.

if $r/p \leq 1 - \ell/m$ and $\langle s \rangle / m = 1$ otherwise. See Supplementary Theorem 1.1 for the precise statement for perfect decoding and Supplementary Theorem 7.1 for the analogous statement in the presence of decoding errors. This is achieved by a specific optimal choice of the coefficients $w_0, \ldots, w_\ell$, which can be classically precomputed in polynomial time, as described in Supplementary Information section 6.

Note that r/p is the fraction of constraints that would be satisfied if the variables were assigned uniformly at random. When $r/p = 1/2$, equation (6) becomes the equation of a semicircle. Hence, we informally refer to equation (6) as the 'semicircle law'.

From equation (6), any result on decoding a class of linear codes implies a corresponding result regarding the performance of DQI for solving a class of combinatorial optimization problems that are dual to these codes. This enables two new lines of research in quantum optimization. The first is to harvest the coding theory literature for rigorous theorems on the performance of decoders for various codes and obtain as corollaries guarantees on the approximation achieved by DQI for the corresponding optimization problems. The second is to perform computer experiments to determine the empirical performance of classical heuristic decoders, which, through equation (6), can be compared against the empirical performance of classical heuristic optimizers. In this manner, DQI can be benchmarked instance-by-instance against classical heuristics, even for optimization problems far too large to attempt on present-day quantum hardware. We next describe our results so far from each of these two lines of research.

We first use rigorous decoding guarantees to analyse the performance of DQI on the following problem.

Definition 2. Given integers $n < p - 1$ with p prime, an instance of the optimal polynomial intersection (OPI) problem is as follows. Let $F_1, \ldots, F_{p-1}$ be subsets of the finite field $\mathbb{F}_p$. Find a polynomial $Q \in \mathbb{F}_p[y]$ of degree at most $n - 1$ that maximizes $f_{\mathrm{OPI}}(Q) = |\{y \in \{1, \ldots, p-1\} : Q(y) \in F_y\}|$, that is, that intersects as many of these subsets as possible.

An illustration of this problem is given in Fig. 2.

In Supplementary Information section 2, we show that OPI is a special case of max-LINSAT over $\mathbb{F}_p$ with $m = p - 1$ constraints in which B is a Vandermonde matrix and, thus, $C^\perp$ is a Reed–Solomon code. Syndrome decoding for Reed–Solomon codes can be solved in polynomial time out to half the distance of the code, for example, using the Berlekamp–Massey algorithm[26]. Consequently, in DQI we can take $\ell = \lfloor (n+1)/2 \rfloor$. For the regime where r/p and n/p are constants and p is taken asymptotically large, the fraction of satisfied constraints achieved by DQI using the Berlekamp–Massey decoder can be obtained by substituting $\ell/m = n/2p$ into equation (6).

OPI and special cases of it have been studied in several domains. In the coding theory literature, OPI is studied under the name 'list-recovery', and in the cryptography literature it is studied under the name 'noisy polynomial reconstruction/interpolation'[27,28]. OPI can also be viewed as a generalization of the polynomial approximation problem, studied in refs. 29–31, in which each set F_i is a contiguous range of values in $\mathbb{F}_p$. In Supplementary Information section 8, we analyse the algorithms from these works in the literature and find that, for the parameter regime addressed by DQI, the best approximation achieved in polynomial time classically is $1/2 + n/2p$ using Prange's algorithm. As shown in Fig. 3, for $r/p = 1/2$ and any fixed $0 < n/p < 1$, DQI with the Berlekamp–Massey decoder exceeds the satisfaction fraction achieved by Prange's algorithm in the limit of large p. Classically, the only methods we are aware of to exceed the satisfaction fraction achieved by Prange's algorithm are brute force search or slight refinements thereof, which have exponential runtime. Thus, DQI achieves a superpolynomial speed-up for this problem, assuming no polynomial-time algorithm is found that can match the satisfaction fraction that DQI achieves.

At present, there are no results directly showing that the OPI problem in the parameter regime that we consider is classically intractable under any standard complexity-theoretic or cryptographic assumptions. However, such results are known for certain limiting cases of the OPI problem, and we propose the task of extending these results to regimes more relevant to DQI for future research. The hardness of the special case of OPI when $|f_i^{-1}(+1)| = 1$, in a certain parameter regime, has been proposed as a cryptographic assumption in ref. 32, which has not been broken to our knowledge. Finding exact optima for OPI with $|f_i^{-1}(+1)| = 1$ can be cast as maximum-likelihood decoding for Reed–Solomon codes, which is known to be NP-hard[33,34]. Finding sufficiently good approximate optima is known to be as hard as discrete log[35,36], but these hardness results do not match the parameter regime addressed by DQI.

As a concrete example, for $n \approx p/10$ and $r/p \approx 1/2$, the fraction of constraints satisfied by Prange's algorithm is 0.55, whereas DQI achieves $1/2 + \sqrt{19}/20 \approx 0.7179$. As a specific point of comparison, we challenge the algorithms community to beat this with a classical polynomial-time algorithm. Interestingly, for these parameters, one statistically expects that solutions satisfying all $p-1$ constraints exist, but they apparently remain out of reach of polynomial-time algorithms both quantum and classical.

To find classically intractable instances of OPI solvable by DQI with minimal quantum resources, we find it is advantageous to choose $n/p \approx r/p \approx 1/2$. For these parameters, DQI achieves satisfaction fraction 0.933. As discussed in Supplementary Information section 13, achieving this using classical algorithms known to us has a prohibitive computational cost for p as small as 521. The dominant cost in DQI plus the Berlekamp–Massey decoder is the reversible implementation of the subroutine to find the shortest linear-feedback shift register used in the Berlekamp–Massey algorithm. Using Qualtran[37], we find that at $p = 521$, the linear-feedback shift register can be found using approximately 1×10^8 logical Toffoli gates and 9×10^3 logical qubits.

We next use computer experiments to benchmark the performance of DQI against classical heuristics on average-case instances from certain families of max-XORSAT with sparse B. DQI reduces such problems to decoding problems on codes with sparse parity-check matrices. Such codes are known as low-density parity-check (LDPC) codes. Polynomial-time classical algorithms, such as belief propagation, can decode randomly sampled LDPC codes up to numbers of errors that nearly saturate information-theoretic limits[3,38,39]. This makes sparse max-XORSAT an enticing target for DQI. Although we use max-XORSAT as a convenient test bed for DQI, other commonly studied optimization problems, such as max-k-SAT, could be addressed similarly. Specifically, consider any binary optimization problem in which the objective function counts the number of satisfied constraints, where each constraint is a Boolean function of at most k variables. By taking the Hadamard transform of the objective function, one converts such a problem into an instance of weighted max-k-XORSAT, where the number of variables is unchanged and the number of constraints has been increased by at most a factor of 2^k.

Although we are able to analyse the asymptotic average-case performance of DQI rigorously, we do not restrict the classical competition to algorithms with rigorous performance guarantees. Instead, we choose to set a high bar by also attempting to beat the empirical performance of classical heuristics that lack such guarantees.

Through careful tuning of sparsity patterns in B, we are able to find some families of sparse max-XORSAT instances for which DQI with standard belief-propagation decoding finds solutions satisfying a larger fraction of constraints than we are able to find using a comparable number of computational steps by any of the general-purpose classical optimization heuristics that we tried, which are listed in Table 1. However, unlike our OPI example, we do not put this forth as a potential example of superpolynomial quantum advantage. Rather, we are able to construct a tailored classical algorithm specialized to these instances, which, with 7 min of runtime, finds solutions where the fraction of constraints satisfied slightly beats DQI plus belief propagation (DQI + BP). As discussed in Supplementary Information section 9, our tailored heuristic is a variant of simulated annealing that assigns temperature-dependent weights to the terms in the cost function determined by how many variables they contain.

Table 1 | Approximate optima for max-XORSAT

Algorithm	Fraction satisfied
Tailored heuristic (7 min ×1 core)	0.880
Long anneal (73 h×5 cores)	0.832
DQI + BP	≥0.831
Prange's algorithm	0.812
Short anneal (8 s×1 core)	0.764
Greedy algorithm	0.666
AdvRand	0.554

Here we compare DQI, using a standard belief-propagation decoder, against classical algorithms for a randomly generated max-XORSAT instance with irregular degree distribution specified in Supplementary Information section 9. We consider an example instance with 31,216 variables and 50,000 constraints. The classical algorithms above are defined in Supplementary Information section 8. For simulated annealing, the satisfaction fraction grows with the runtime, so we report two numbers. The first is the optimum reachable by limiting simulated annealing to the same runtime used by belief propagation to solve the problem to which the max-XORSAT instance is reduced by DQI (8 s×1 core), and the second is for the shortest anneal that matched the satisfaction fraction achieved by DQI + BP (73 h×5 cores).

The comparison against simulated annealing is complicated because, as shown in Supplementary Information section 8.2, the fraction of clauses satisfied by simulated annealing increases as a function of the duration of the anneal. Thus, there is not a unique sharply defined number indicating the maximum satisfaction fraction reachable by simulated annealing. DQI reduces our sparsity-tuned max-XORSAT problem to an LDPC decoding problem, which our implementation of belief propagation solves in approximately 8 s on a single core, excluding the time used to load and parse the instance. Thus, a natural point of comparison is the result obtained by simulated annealing with a similar runtime. By running our optimized C++ implementation of simulated annealing for 8 s, we are only able to reach 0.764. If we allow the parallel execution of several anneals and increase our runtime allowance, we are able to eventually replicate the satisfaction fraction achieved by DQI + BP using simulated annealing. The shortest anneal that achieved this used five cores and ran for 73 h, which is five orders of magnitude longer than our belief-propagation decoder. Although this is dependent on the implementation details, we can take this ratio of runtimes as a rough indicator of the ratio of computational steps. In the context of DQI, the decoder would need to be implemented as a reversible circuit and subject to an overhead due to quantum error correction, so this should not be interpreted as an indicator of the quantum versus the classical runtime.

Discussion

The idea that quantum Fourier transforms could be used to achieve reductions between problems on lattices and their duals originates in the early 2000s in the work of Regev, Aharonov and Ta-Shma[40–43]. Linear codes, as considered here, are closely analogous to lattices but over finite fields. By considering lattices with only a geometric structure, no quantum speed-ups were found using these reductions until the 2021 breakthrough of Chen, Liu and Zhandry[16], who obtained a superpolynomial speed-up for a constraint satisfaction problem by combining these ideas with an intrinsically quantum decoding method. Other recent explorations of Regev-style reductions to general unstructured codes and lattices are given in refs. 20,44,45. Here we restrict attention to codes defined by matrices that are either sparse or algebraically structured, and in the latter case, we are able to obtain

an apparent superpolynomial quantum speed-up for an optimization problem.

Recently, Yamakawa and Zhandry have also considered the application of Regev-style reductions to a problem with extra structure and obtained a quantum advantage[46]. They defined an oracle problem that they proved can be solved using polynomially many quantum queries but requires exponentially many classical queries. Their problem is essentially equivalent to max-LINSAT over an exponentially large finite field $\mathbb{F}_{2^t}$, where the sets $F_1, \ldots, F_m$ are defined by random oracles and the matrix B is obtained from a folded Reed–Solomon code. In Supplementary Information section 11, we recount the exact definition of the Yamakawa–Zhandry problem and argue that DQI can be extended to the Yamakawa–Zhandry problem and, in this case, probably yields solutions satisfying all constraints. Although problems with exponentially large $F_1, \ldots, F_m$ defined by oracles are far removed from industrial optimization problems, this limiting case provides evidence against the possibility of efficiently simulating DQI with classical algorithms and thereby 'dequantizing' it, as has happened with some previous quantum algorithms proposed as potential superpolynomial speed-ups[47]. More precisely, our argument indicates that DQI cannot be dequantized by any relativizing techniques, in the sense of ref. 48.

We conclude by noting that the work reported here initiates the exploration of quantum speed-ups through DQI but is very far from completing it. In particular, we highlight three avenues for future work: multivariate OPI, custom decoders for solving max-XORSAT by DQI, and sampling problems. First, we note that the DQI algorithm can be straightforwardly adapted to solve the multivariate generalization of OPI. As shown in Supplementary Information section 12, multivariate OPI gets reduced by DQI to the decoding of Reed–Muller codes. Known polynomial-time classical algorithms can decode all Reed–Muller codes out to half their distance[49]. (Reed–Solomon codes are the univariate special case.) Consequently, one expects a region of parameter space for which DQI achieves superpolynomial speed-up on multivariate OPI, which includes the speed-up on univariate OPI presented here as a special case. Mapping out this region of quantum advantage remains for future work.

Second, we note that our exploration of DQI applied to max-XORSAT is far from exhaustive. In particular, equation (6) enables a benchmark-driven approach to the development of tailored heuristics for decoding designed to achieve quantum speed-up on some class of optimization problems using DQI. This search can be guided by upper bounds on the performance of DQI that, through the semicircle law, follow from information-theoretic limits on decoding. Such an analysis is given in Supplementary Information section 10 and shows that for D-regular max-k-XORSAT instances, the upper bound on the possible performance of DQI with classical decoders is already exceeded by the empirical performance of simulated annealing when k is too small relative to D. Additionally, we compare the performance of DQI against the quantum approximate optimization algorithm for various ensembles of max-k-XORSAT instances at $k = 2$ and $k = 3$. On all of these, the quantum approximate optimization algorithm exceeds the upper bound on performance for DQI with classical decoders.

These limits show that, for DQI to achieve an advantage on max-k-XORSAT, one must either go to large k or move to quantum decoders that exploit the coherence of the bit flip errors. Large-k problems are reduced by DQI to decoding problems in which the parity-check matrix is denser than in typical LDPC codes. The increased density degrades the performance of belief propagation. This indicates the need for future research developing decoders to tolerate denser parity-check matrices than are typically used. Despite some progress along these lines[50–56], this remains an underexplored area compared with the decoding of codes with very sparse parity-check matrices. With quantum decoders, it remains information-theoretically possible for DQI to achieve advantage over known polynomial-time classical and quantum algorithms, even for small k. Realizing this potential advantage depends on the development of polynomial-size quantum circuits for this quantum decoding problem. Some exciting progress on this problem has been reported in refs. 16,44,57.

Third, we note that DQI produces unbiased samples, in which the probability of obtaining a given solution to an optimization problem is constant across all solutions achieving a given objective value. This guarantee of fair sampling is absent for most classical optimization algorithms and has known applications to very hard problems of approximate counting[58].

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41586-025-09527-5.

1. Abbas, A. et al. Challenges and opportunities in quantum optimization. *Nat. Rev. Phys.* **6**, 718–735 (2024).
2. Gallager, R. G. *Low-Density Parity-Check Codes* (MIT, 1963).
3. Richardson, T. J. & Urbanke, R. L. The capacity of low-density parity-check codes under message-passing decoding. *IEEE Trans. Inf. Theory* **47**, 599–618 (2001).
4. Trevisan, L. in *Paradigms of Combinatorial Optimization: Problems and New Approaches* 2nd edn (ed. Paschos, V. Th.) Ch. 13 (Wiley, 2014).
5. Ajtai, M., Kumar, R. & Sivakumar, D. A sieve algorithm for the shortest lattice vector problem. In *Proc. 33rd Annual ACM Symposium on Theory of Computing* 601–610 (ACM, 2001).
6. Moshkovitz, D. The projection games conjecture and the NP-hardness of ln *n*-approximating set-cover. *Theory Comput.* **11**, 221–235 (2015).
7. Farhi, E. et al. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292**, 472–475 (2001).
8. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm applied to a bounded occurrence constraint problem. Preprint at https://arxiv.org/abs/1412.6062 (2014).
9. Hastings, M. B. A short path quantum algorithm for exact optimization. *Quantum* **2**, 78 (2018).
10. Basso, J., Farhi, E., Marwaha, K., Villalonga, B. & Zhou, L. The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington-Kirkpatrick model. In *Proc. 17th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2022)* (eds Le Gall, F. & Morimae, T.) 7:1–7:21 (Dagstuhl, 2022).
11. Dalzell, A. M., Pancotti, N., Campbell, E. T. & Brandão, F. G. S. L. Mind the gap: achieving a super-Grover quantum speedup by jumping to the end. In *Proc. 55th Annual ACM Symposium on Theory of Computing (STOC)* (eds Saha, B. & Servedio, R. A.) 1131–1144 (ACM, 2023).
12. Kapit, E. et al. On the approximability of random-hypergraph MAX-3-XORSAT problems with quantum algorithms. Preprint at https://arxiv.org/abs/2312.06104 (2023).
13. Shaydulin, R. et al. Evidence of scaling advantage for the quantum approximate optimization algorithm on a classically intractable problem. *Sci. Adv.* **10**, eadm6761 (2024).
14. Farhi, E., Gutmann, S., Ranard, D. & Villalonga, B. Lower bounding the MaxCut of high girth 3-regular graphs using the QAOA. Preprint at https://arxiv.org/abs/2503.12789 (2025).
15. Leng, J., Zheng, Y. & Wu, X. A quantum-classical performance separation in nonconvex optimization. Preprint at https://arxiv.org/abs/2311.00811 (2023).
16. Chen, Y., Liu, Q. & Zhandry, M. Quantum algorithms for variants of average-case lattice problems via filtering. In *Proc. Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)* (eds Dunkelman, O. & Dziembowski, S.) 372–401 (Springer, 2022).
17. Pirnay, N., Ulitzsh, V., Wilde, F., Eisert, J. & Seifert, J.-P. An in-principle super-polynomial quantum advantage for approximating combinatorial optimization problems via computational learning theory. *Sci. Adv.* **10**, eadj5170 (2024).
18. Szegedy, M. Quantum advantage for combinatorial optimization problems, simplified. Preprint at https://arxiv.org/abs/2212.12572 (2022).
19. Gilyén, A., Hastings, M. B. & Vazirani, U. (Sub)exponential advantage of adiabatic quantum computation with no sign problem. In *Proc. 53rd Annual ACM SIGACT Symposium on Theory of Computing* (STOC) (eds Khuller, S. & Vassilevska Williams, V.) 1357–1369 (ACM, 2021).
20. Eldar, L. & Hallgren, S. An efficient quantum algorithm for lattice problems achieving subexponential approximation factor. Preprint at https://arxiv.org/abs/2201.13450 (2022).
21. Denchev, V. S. et al. What is the computational value of finite-range tunneling? *Phys. Rev. X* **6**, 031015 (2016).
22. Bartschi, A. & Eidenbenz, S. Short-depth circuits for Dicke state preparation. In *Proc. 2022 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2022).
23. Wang, H., Tan, B., Cong, J. & De Micheli, G. Quantum state preparation using an exact CNOT synthesis formulation. In *Proc. Design, Automation and Test in Europe (DATE)* 1–6 (IEEE, 2024).

24. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2010).
25. Berlekamp, E., McEliece, R. & van Tilborg, H. On the inherent intractability of certain coding problems. *IEEE Trans. Inf. Theory* **24**, 384–386 (1978).
26. Berlekamp, E. R. *Algebraic Coding Theory (revised edition)* (World Scientific, 2015).
27. Naor, M. & Pinkas, B. Oblivious transfer and polynomial evaluation. In *Proc. 31st Annual ACM Symposium on Theory of Computing (STOC)* 245–254 (ACM, 1999).
28. Bleichenbacher, D. & Nguyen, P. Q. Noisy polynomial interpolation and noisy Chinese remaindering. In *Proc. International Conference on the Theory and Applications of Cryptographic Techniques* (ed. Preneel, B.) 53–69 (Springer, 2000).
29. Garcia-Morchon, O., Rietman, R., Shparlinski, I. E. & Tolhuizen, L. Interpolation and approximation of polynomials in finite fields over a short interval from noisy values. *Exp. Math.* **23**, 241–260 (2014).
30. Shparlinski, I. & Winterhof, A. Noisy interpolation of sparse polynomials in finite fields. *Appl. Algebra Eng. Commun. Comput.* **16**, 307–317 (2005).
31. Shparlinski, I. E. Playing 'hide-and-seek' with numbers. In *Public-Key Cryptography: American Mathematical Society Short Course* (2005).
32. Naor, M. & Pinkas, B. Oblivious polynomial evaluation. *SIAM J. Comput.* **35**, 1254–1281 (2006).
33. Guruswami, V. & Vardy, A. Maximum-likelihood decoding of Reed-Solomon codes is NP-hard. *IEEE Trans. Inf. Theory* **51**, 2249–2256 (2005).
34. Gandikota, V., Ghazi, B. & Grigorescu, E. NP-hardness of Reed-Solomon decoding, and the Prouhet-Tarry-Escott problem. *SIAM J. Comput.* **47**, 1547–1584 (2018).
35. Cheng, Q. & Wan, D. On the list and bounded distance decodability of Reed-Solomon codes. *SIAM J. Comput.* **37**, 195–209 (2007).
36. Cheng, Q. & Wan, D. Complexity of decoding positive-rate Reed-Solomon codes. In *Proc. International Colloquium on Automata, Languages, and Programming* (ICALP) (eds Aceto, L. et al.) 283–293 (Springer, 2008).
37. Harrigan, M. P. et al. Expressing and analyzing quantum algorithms with Qualtran. Preprint at https://arxiv.org/abs/2409.04643 (2024).
38. Gallager, R. Low-density parity-check codes. *IRE Trans. Inf. Theory* **8**, 21–28 (1962).
39. Mézard, M. & Montanari, A. *Information, Physics, and Computation* (Oxford Univ. Press, 2009).
40. Aharonov, D. & Ta-Shma, A. Adiabatic quantum state generation and statistical zero knowledge. In *Proc. 35th Annual ACM Symposium on Theory of Computing (STOC)* 20–29 (ACM, 2003).
41. Regev, O. Quantum computation and lattice problems. *SIAM J. Comput.* **33**, 738–760 (2004).
42. Aharonov, D. & Regev, O. Lattice problems in NP ∩ coNP. *J. ACM* **52**, 749–765 (2005).
43. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM* **56**, 1–40 (2009).
44. Chailloux, A. & Tillich, J.-P. The quantum decoding problem. In *19th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)* (eds Magniez, F. & Grilo, A. B.) 6:1-6:14 (Schloss Dagstuhl–Leibniz-Zentrum fur Informatik, 2024).
45. Debris-Alazard, T., Remaud, M. & Tillich, J.-P. Quantum reduction of finding short code vectors to the decoding problem. *IEEE Trans. Inf. Theory*. **70**, 5323–5342 (2023).
46. Yamakawa, T. & Zhandry, M. Verifiable quantum advantage without structure. In *Proc. 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)* 69–74 (IEEE, 2022).
47. Chia, N.-H. et al. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. *J. ACM* **69**, 1–72 (2022).
48. Fortnow, L. The role of relativization in complexity theory. *Bull. Eur. Assoc. Theor. Comput. Sci.* **52**, 229–243 (1994).
49. Pellikaan, R. & Wu, X.-W. List decoding of *q*-ary Reed-Muller codes. *IEEE Trans. Inf. Theory* **50**, 679–682 (2004).
50. Feldman, J., Wainwright, M. J. & Karger, D. R. Using linear programming to decode binary linear codes. *IEEE Trans. Inf. Theory* **51**, 954–972 (2005).
51. Feldman, J., Malkin, T., Servedio, R. A., Stein, C. & Wainwright, M. J. LP decoding corrects a constant fraction of errors. *IEEE Trans. Inf. Theory* **53**, 82–89 (2007).
52. Daskalakis, C., Dimakis, A. G., Karp, R. M. & Wainwright, M. J. Probabilistic analysis of linear programming decoding. *IEEE Trans. Inf. Theory* **54**, 3565–3578 (2008).
53. Taghavi, M. H. & Siegel, P. H. Adaptive methods for linear programming decoding. *IEEE Trans. Inf. Theory* **54**, 5396–5410 (2008).
54. Yufit, A., Lifshitz, A. & Be'ery, Y. Efficient linear programming decoding of HDPC codes. *IEEE Trans. Commun.* **59**, 758–766 (2010).
55. Tanatmis, A. et al. Valid inequalities for binary linear codes. In *Proc. 2009 IEEE International Symposium on Information Theory (ISIT)* 2216–2220 (IEEE, 2009).
56. Tanatmis, A. et al. A separation algorithm for improved LP decoding of linear block codes. *IEEE Trans. Inf. Theory* **56**, 3277–3289 (2010).
57. Piveteau, C. & Renes, J. M. Quantum message-passing algorithm for optimal and efficient decoding. *Quantum* **6**, 784 (2022).
58. Sinclair, A. & Jerrum, M. Approximate counting, uniform generation and rapidly mixing Markov chains. *Inf. Comput.* **82**, 93–133 (1989).

Data availability

The problem instances that we describe and the raw data from our plots are available at Zenodo (https://doi.org/10.5281/zenodo.13327870)[59].

Code availability

The code used in our computer experiments and resource estimation is available at Zenodo (https://doi.org/10.5281/zenodo.13327870)[59].

59. Shutty, N., Jordan, S. & Khattar, T. Supporting material for: optimization by decoded quantum interferometry. *Zenodo* https://doi.org/10.5281/zenodo.13327870 (2025).

Acknowledgements We thank R. Kothari, R. O'Donnell, E. Farhi, H. Neven, K. Kechedzhi, S. Boixo, V. Smelyanskiy, Y. Lensky, D. Aharonov, O. Regev, J. McClean, M. Sudan, U. Vazirani, Y. Ishai, B. Hemenway Falk, O. Higgott, J. Azariah, O. Parekh, J. Machta, H. Katzgraber, C. Gidney, N. Yosri and D. Maslov for useful discussions. M.W.'s work on this project was funded by a grant from Google Quantum AI.

Author contributions S.P.J. and N.S. designed the quantum algorithm. A.Z. optimized it. S.P.J., A.Z., A.S. and R.K. proved the semicircle law. M.W. surveyed and analysed the algebraic codes and decoding algorithms and extended DQI to the Yamakawa–Zhandry problem. S.P.J., N.S., M.W., A.S. and A.Z. performed the computational experiments. T.K. performed resource estimation. S.V.I. devised and tested the classical optimizers. R.B. contributed efficient state preparation methods and helped to manage the collaboration. All authors contributed to writing and editing the paper.

Competing interests Google has filed a patent application pertaining to the DQI algorithm, on which S.P.J. is the inventor.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41586-025-09527-5.
Correspondence and requests for materials should be addressed to Stephen P. Jordan or Noah Shutty.
Peer review information *Nature* thanks Ashley Montanaro and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.