Limitations of Decoded Quantum Interferometry for MaxCut

Ojas Parekh
Quantum Algorithms and Applications Collaboratory, Sandia National Laboratories, Albuquerque, NM, USA

Abstract

Decoded Quantum Interferometry (DQI) is a framework for approximating special kinds of discrete optimization problems that relies on problem structure in a way that sets it apart from other classical or quantum approaches. We show that any family of instances of MaxCut on which DQI attains a nontrivial asymptotic approximation guarantee is also solvable exactly in classical polynomial time. We include a streamlined exposition of DQI tailored for MaxCut that relies on elementary graph theory instead of coding theory to motivate and explain the algorithm.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Quantum complexity theory

Keywords and phrases Decoded Quantum Interferometry, MaxCut, quantum approximation algorithms

Digital Object Identifier 10.4230/LIPIcs.TQC.2026.6

Related Version *arXiv Version*: `https://arxiv.org/abs/2509.19966`

Funding This work is supported by the U.S. Department of Energy (DOE), Office of Science, National Quantum Information Science Research Centers, Quantum Systems Accelerator (QSA). Additional support is acknowledged from DOE, Office of Science, Accelerated Research in Quantum Computing, Fundamental Algorithmic Research toward Quantum Utility (FAR-Qu). This article has been authored by an employee of National Technology & Engineering Solutions of Sandia, LLC under Contract No. DE-NA0003525 with the U.S. Department of Energy (DOE). The employee owns all right, title and interest in and to the article and is solely responsible for its contents. The United States Government retains and the publisher, by accepting the article for publication, acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this article or allow others to do so, for United States Government purposes. The DOE will provide public access to these results of federally sponsored research in accordance with the DOE Public Access Plan `https://www.energy.gov/downloads/doe-public-access-plan`.

Acknowledgements I thank Stephen Jordan and Noah Shutty for helpful discussions on DQI and feedback on this work.

1 Introduction

MaxCut seeks to find a subset of vertices $S \subseteq V$ in a graph G maximizing the number of edges crossing S (i.e., edges with exactly one endpoint in S). It can also be equivalently viewed as a canonical constraint satisfaction problem in graphs by assigning boolean values to the vertices to maximize the number of edges with different values assigned to endpoints. MaxCut has sparked diverse avenues of research and has become a testbed problem for quantum-algorithmic approaches for solving or approximating discrete optimization problems, such as quantum annealing and the Quantum Approximate Optimization Algorithm (QAOA) [6]. Although approximation guarantees can be derived for the QAOA on MaxCut, rigorous approximation advantages remain elusive despite over a decade of extensive study. A quantum advantage for approximating general instances of MaxCut seems unlikely, since we can obtain

a 0.878...-approximation in classical polynomial time [9], and a $(0.878... + \varepsilon)$-approximation is NP-hard under the Unique Games Conjecture [14]. This suggests that it is hard for even efficient quantum algorithms to improve upon this classical threshold.

For "natural" discrete optimization problems, why would we *not* expect that there is an α such that an α-approximation is possible in classical polynomial time and an $\alpha + \epsilon$-approximation is NP-hard? What kind of problems would admit a gap between classical approximability and NP-hardness of approximability, for a quantum advantage to possibly sneak in? In other words, why would a "natural" classical discrete optimization problem need a quantum algorithm to complete the story of its approximability?

A recent example of an unexpected provable exponential quantum approximation advantage is for the directed version of MaxCut [13, 12], which is arguably a natural problem. However, in this case the quantum advantage is enabled by considering space advantages in the streaming model instead of more traditional speedups. More broadly, perhaps certain kinds of restricted instances or regimes of natural discrete optimization problems are amenable to quantum approximation advantages.

Decoded Quantum Interferometry (DQI) potentially presents such an opportunity, since it relies heavily on special problem structure in a way that sets it apart from other classical or quantum approaches [11]. DQI is an application of Regev's reduction [20] designed to work on constraint satisfaction problems for which an associated decoding problem is efficiently classically solvable. Moreover, it is based on Quantum Fourier Sampling [8], which does enable provable or conjectured quantum advantages in many settings. DQI is an exciting development, and there are some examples of problems for which DQI gives a better approximation guarantee than the best classical algorithm currently known. However, the demands that DQI puts on problem structure results in problem regimes unusual and understudied for classical algorithms. The most prominent candidate for a DQI advantage is for Optimal Polynomial Intersection, and there is essentially one nontrivial classical approximation algorithm, that is decades old, currently known for it.

Contributions

Understanding the power and limitations of DQI is thus an important challenge. We make some progress in establishing limitations for DQI, for the fundamental MaxCut problem. MaxCut is perhaps the simplest interesting special case of the Max-LINSAT problem considered by Jordan et al. [11]. Focusing on MaxCut allows for a streamlined presentation and analysis of DQI.

We show that DQI for MaxCut exhibits perhaps surprising behavior: on instance families where DQI attains *any* asymptotic improvement over the trivial random assignment approximation algorithm, it is not just that the approximation guarantee can be beaten classically – such instances can be solved *exactly* in classical polynomial time. To derive this result, we concretely demonstrate how the structure necessary for DQI can also be exploited classically. Our observations and classical algorithms are relatively simple, and MaxCut may be a useful guidepost for future DQI work.

DQI is generally not guaranteed to run in polynomial time and requires both feasibility of and an efficient algorithm for a related classical decoding problem. This limits both the problems and types of instances amenable to DQI. In contrast, DQI runs efficiently on any instance of MaxCut, since the decoding problem reduces to a perfect matching problem.

The key graph parameter for analysis of DQI for MaxCut is the *girth* of the unweighted input graph, which is the length of a shortest cycle. We assume the input comes from an infinite family of graphs with g, n, and m corresponding to girth, number of vertices, and number of edges. Based on the more general analysis of DQI in [11], we first observe:

▶ **Theorem 7.** *If the expected number of edges cut in a graph by DQI is* $(\frac{1}{2}+\varepsilon)m$*, for some constant* $\varepsilon > 0$*, the girth* g *must be linear in* m*.*

The trivial random assignment algorithm for MaxCut is expected to cut $\frac{1}{2}m$ edges, so the above theorem says that any improvement in the constant factor requires instances with very high girth. We show how this can be exploited classically:

▶ **Theorem 10.** *For a connected graph with girth* g *linear in* m*, the quantity* $m - n + 1$ *is constant.*

MaxCut can be solved exactly in classical polynomial time in this regime through a variety of approaches, and even the simple algorithm of cutting every edge in a spanning tree is guaranteed to cut $m - O(1)$ edges, since $m - n + 1$ is precisely the number of edges not in the spanning tree.

Related work

Patamawisut et al. describe a circuit implementation of DQI for MaxCut, using a different decoder based on Gauss-Jordan elimination instead of the aforementioned exact decoder based on perfect matching [19]. While Jordan et al. focus on more general cases of Max-LINSAT than MaxCut, they do observe a limitation on the performance of DQI for random instances of Max-2-XORSAT, which corresponds to $\{-1, +1\}$-weighted MaxCut [11, Section 13.3]. In concurrent work with ours, Anschuetz, Gamarnik, and Lu identify obstructions for DQI on random instances of the more general Max-k-XORSAT problem, suggesting that a quantum advantage through DQI is not possible on typical random instances [1]. Another line of concurrent work, by Marwaha et al., gives evidence suggesting that the probability distribution from which DQI samples solutions may be hard to simulate classically [15]. However, even if DQI is doing something "inherently quantum" that is not efficiently simulable classically, this does not directly speak to whether efficient classical algorithms can outperform DQI. In fact MaxCut may be an example where DQI produces a distribution that is not efficiently producible classically, but as we show, independent polynomial-time classical algorithms that outperform it do exist.

Organization

The next section motivates and explains DQI for MaxCut without assuming any prior knowledge of DQI. Section 3 establishes Theorem 10 and demonstrates that MaxCut can be solved in classical polynomial time when the girth is linear. We close with a discussion in Section 4 on whether extensions of DQI might be able to provide approximation advantages for MaxCut.

2 Specializing Decoded Quantum Interferometry for MaxCut

It will be convenient to formulate MaxCut as a Boolean polynomial optimization problem. We will think of a cut as partitioning an unweighted graph, $G = (V, E)$ into two pieces by assigning either a $+1$ or -1 to each vertex, so that points $z \in \{-1, 1\}^V$ of the Boolean hypercube over V correspond to cuts. We take $n \stackrel{\text{def}}{=} |V|$, $m \stackrel{\text{def}}{=} |E|$, and g as the girth of G, which is the length of a shortest cycle. We implicitly assume that input instances G come

from an infinite family $\mathcal{G}$ so that asymptotic quantities are well defined; in particular we assume that $\lim_{m\to\infty} g/m$ exists. This is a standard assumption and is also implicit in the asymptotic analysis in [11]. The number of edges cut by a z, called the *cut value* of z, is then

$$h(z) = \frac{1}{2}\sum_{ij\in E} 1 - z_i z_j.$$

MaxCut is equivalent to the Boolean polynomial optimization problem

$$\max_{z\in\{-1,1\}^V} h(z).$$

For the incidence vector $\gamma \in \{0,1\}^U$ of a set $S \subseteq U$, we take

$$x^\gamma \stackrel{\text{def}}{=} \prod_{i\in S} x_i^{\gamma_i}.$$

We can represent a polynomial over the Boolean hybercube,

$$p(z) = \sum_{\alpha\in\{0,1\}^V} p_\alpha z^\alpha$$

as a quantum state in two different ways:

- Encode the values of p over inputs as amplitudes

$$|p\rangle = \text{unit}\left(\sum_{z\in\{-1,1\}^V} p(z)\,|z\rangle\right),$$

  where $|z\rangle \stackrel{\text{def}}{=} |\frac{1-z_1}{2}\frac{1-z_2}{2}\cdots\rangle$ and unit$(\cdot)$ outputs a normalized state, or
- Encode the coefficients of p over monomials as amplitudes

$$|\hat{p}\rangle = \text{unit}\left(\sum_{\alpha\in\{0,1\}^V} p_\alpha\,|\alpha\rangle\right).$$

These two encodings are related by the Boolean Fourier transform, also known as the Hadamard transform (see [17] for Fourier analysis of Boolean functions). Hadamard gates applied on all qubits enact this transform on a quantum computer, and we have $H^{\otimes n}|p\rangle = |\hat{p}\rangle$ and $H^{\otimes n}|\hat{p}\rangle = |p\rangle$.

Quantum Fourier Sampling

The above observation gives a simple quantum algorithm, called Quantum Fourier Sampling (QFS) [8], for trying to find a cut z with relatively large value $h(z)$:

▶ **Algorithm 1** (Quantum Fourier Sampling).

1. *Prepare* $|\hat{h}\rangle$
2. *Produce* $|h\rangle = H^{\otimes n}|\hat{h}\rangle$
3. *Measure* $|h\rangle$ *in the computational basis*

This algorithm allows sampling a cut z with probability proportional to $h^2(z)$, so that the cut value dictates the probability of picking a cut, as we would want. We will discuss Step 1 in more detail below, and for now we note that it can be executed efficiently since h is an explicit degree-2 polynomial [8, Lemma 1].

What kind of approximation guarantee does this give for MaxCut? (Spoiler alert: it's not so great). The expected number of edges cut is $\frac{1}{2}m + O(1)$. This is just a slight improvement over the expectation of $\frac{1}{2}m$ edges achieved by the random assignment where each vertex is independently assigned to a side of the cut with probability $\frac{1}{2}$.

2.1 Decoded Quantum Interferometry

What if we could sample cuts from a distribution proportional to some higher power or maybe even a higher degree polynomial of the objective h? This could amplify the probabilities of good cuts, and we might expect a better approximation guarantee.

This is where DQI steps in. Sampling from higher degree distributions is indeed possible, but the tradeoff is an increase in complexity of preparing the corresponding state in Step 1 of QFS. This leaves the questions: (i) how do we find and analyze a good polynomial of h, and (ii) can we prepare the corresponding state efficiently? DQI answers both of these questions.

For (i), the observation in [11] is that if we let $y_{ij} \stackrel{\text{def}}{=} z_i z_j$, then h is a symmetric polynomial in the variables y since it only depends on $g(y) \stackrel{\text{def}}{=} \sum_{ij\in E} y_{ij}$. The polynomials

$$g_k(y) \stackrel{\text{def}}{=} \sum_{\substack{\beta\in\{0,1\}^E \\ |\beta|=k}} y^\beta$$

are a basis for the symmetric polynomials. Hence, any degree-l polynomial of h, which is also a degree-l polynomial in the variables y, can be expressed as

$$q(y) = \sum_{k=0}^{l} \mu_k g_k(y) = \sum_{\beta\in\mathcal{E}_l} q_\beta y^\beta = \sum_{\alpha\in\{0,1\}^V} q'_\alpha z^\alpha, \tag{1}$$

where $\mathcal{E}_k \stackrel{\text{def}}{=} \{\beta \in \{0,1\}^E \mid |\beta| \le k\}$. The idea is then to find a good polynomial q and run Algorithm 1 on it to sample a cut z with probability proportional to $q^2(z)$. The coefficients μ_k of a q that maximize the expected cut value, $\mathbb{E}_{z\sim q^2}[h(z)]$, can be found classically in polynomial time by solving a maximum-eigenvector problem for an $(l+1)\times(l+1)$ tridiagonal matrix (see Lemma 5). In light of this, we can think of the input to DQI as the unweighted graph G and the degree parameter l, where the approximation guarantee of DQI monotonically increases with l.

How large does l need to be? A distribution supported only on optimal cuts is proportional to

$$h^*(y) = \prod_{i=2}^{s} (h(y) - \lambda_i),$$

where $\lambda_1 > \lambda_2 > \cdots > \lambda_s$ are the possible cut values $h(y)$ can attain. The symmetric polynomial h^*, which is proportional to $(h^*)^2$, has degree at most m since the cut value $h(y) \in [0, m]$ for all y. Hence, taking $l = m$ would solve MaxCut, and it suffices to choose $l \le m$. This also suggests that DQI should not run in polynomial time when $l = m$, since MaxCut is NP-complete, and it is widely believed that polynomial-time quantum algorithms cannot solve NP-complete problems. DQI gives a better expected approximation guarantee than the best classical algorithm currently known for certain problems in the regime where $l = \Theta(n) = \Theta(m)$ (for more general constraint satisfaction problems, n and m are the number of variables and constraints).

Preparing the initial state for QFS

We need to be able to prepare $|\hat{q}\rangle$ in polynomial time. We can do so readily in a Hilbert space where we have m qubits corresponding to E. First note that g_k can be represented using a Dicke state:

$$|\hat{g_k}\rangle_E = \frac{1}{\sqrt{\binom{m}{k}}} \sum_{\substack{\beta\in\{0,1\}^E: \\ |\beta|=k}} |\beta\rangle\,,$$

where the subscript on the ket identifies the set that the qubits correspond to. Then we obtain, as a superposition of Dicke states

$$|\hat{q}\rangle_E = \text{unit}\left(\sum_{k=0}^{l} \mu_k \sqrt{\binom{m}{k}} |\hat{g_k}\rangle\right) = \text{unit}\left(\sum_{\beta\in\mathcal{E}_l} q_\beta |\beta\rangle\right),$$

which can be prepared efficiently as described in [11]. However, in order to apply Quantum Fourier Sampling, we instead need the state

$$|\hat{q}\rangle_V = \text{unit}\left(\sum_{\alpha\in\{0,1\}^V} q'_\alpha |\alpha\rangle\right).$$

How can we construct $|\hat{q}\rangle_V$ from $|\hat{q}\rangle_E$? Since $y_{ij} = z_i z_j$, we get a natural map f from $\beta \in \{0,1\}^E$ to $\alpha \in \{0,1\}^V$ as

$$y^\beta = \prod_{\substack{ij\in E:\\ \beta_{ij}=1}} z_i z_j = z^{f(\beta)},$$

where $z_i^2 = 1$ for all i is employed to reduce the monomial $z^{f(\beta)}$ so that $f(\beta) \in \{0,1\}^V$. We think of β as an incidence vector representing a subgraph (i.e., a subset of edges) and the *parity vector*, $f(\beta)$ as indicating the parity of the degree of vertex in V in the subgraph β (vertices may have no incident edges). We can see that f is not injective in general by letting G be a cycle on three vertices, for which $f(1,1,1) = f(0,0,0) = (0,0,0)$. In other words, all vertices in V have even degree in both G itself and the empty subgraph.

▶ **Lemma 1** (specialized for MaxCut from [11]). *Suppose that G is a graph where each parity vector $\alpha \in \{0,1\}^V$ arises from at most one subgraph $\beta \in \mathcal{E}_l$. Then f is injective (on the domain $\mathcal{E}_l$), and the following algorithm may be used to produce $|\hat{q}\rangle_V$.*

1. *Prepare* $|\hat{q}\rangle_E |0\rangle_V = \text{unit}\left(\sum_{\beta\in\mathcal{E}_l} q_\beta |\beta\rangle_E |0\rangle_V\right)$
2. *Apply f coherently to get:* $\text{unit}\left(\sum_{\beta\in\mathcal{E}_l} q_\beta |\beta\rangle_E |f(\beta)\rangle_V\right)$
3. *Uncompute the E register using f^{-1} to yield:* $|\psi\rangle \stackrel{\text{def}}{=} \text{unit}\left(\sum_{\beta\in\mathcal{E}_l} q_\beta |0\rangle_E |f(\beta)\rangle_V\right)$

Proof. We see from Equation (1) that

$$q'_\alpha = \sum_{\substack{\beta\in\mathcal{E}_l:\\ f(\beta)=\alpha}} q_\beta, \tag{2}$$

with $q'_\alpha = 0$ if there is no $\beta \in \mathcal{E}_l$ such that $f(\beta) = \alpha$. By Equation (2), discarding the E register from $|\psi\rangle$ produces the desired state:

$$\text{unit}\left(\sum_{\beta\in\mathcal{E}_l} q_\beta |f(\beta)\rangle_V\right) = \text{unit}\left(\sum_{\alpha\in\{0,1\}^V} q'_\alpha |\alpha\rangle\right). \quad ◀$$

DQI uses the algorithm of Lemma 1 and thus requires f to be injective. What kind of structural assumption does this impose on the input G, l?

▶ **Lemma 2.** *Suppose we have a graph G and $l \in \mathbb{N}$. Then f is injective on $\mathcal{E}_l$ if and only if G has girth at least $2l+1$.*

Proof. First we observe that f is linear when viewed as a function on $\mathbb{F}_2^E$. Let $B \in \mathbb{F}_2^{E\times V}$ be the edge-vertex incidence matrix of G, with $B_{e,v} = 1$ if and only if the edge e is incident to the vertex v. Then $f(\beta) = \alpha$ if and only if $\alpha = B^T\beta$, with α and β taken as vectors over $\mathbb{F}_2$.

We show that if f is not injective then the girth of G is at most $2l$. Suppose there are two distinct subgraphs $\beta, \beta' \in \mathcal{E}_l$ with $f(\beta) = f(\beta')$. Then, in $\mathbb{F}_2^E$, $\gamma \stackrel{\text{def}}{=} \beta + \beta' \neq \mathbf{0}$. The parity vector of γ is $\mathbf{0}$, since $f(\gamma) = f(\beta) + f(\beta') = \mathbf{0}$. This means that the nonempty subgraph corresponding to γ has even degree at all vertices and has at most $2l$ edges. This subgraph must therefore contain a cycle of size at most $2l$. To see this, one can start at a vertex and keep moving to other vertices using previously unused edges. Since all degrees are even, it will always be possible to leave a vertex that is visited. Since the graph is finite, one must eventually revisit a vertex, resulting in a cycle.

Now we show the opposite direction. Suppose there is a cycle with at most $2l$ edges. Break the cycle into two edge-disjoint paths of length at most l, both between the same pair of vertices i and j on the cycle. Represent these paths by incidence vectors $\beta, \beta' \in \mathcal{E}_l$. Then f is not injective since $f(\beta) = f(\beta') = \alpha$, where $\alpha_i = \alpha_j = 1$ and all other entries are 0. ◀

So for MaxCut there is a simple graph-theoretic characterization of the structure DQI requires. It makes sense to pick the largest l such that f remains injective; however, we also need an efficient classical algorithm to implement f^{-1}, which may limit the choice of l. For MaxCut, it turns out that f^{-1} can be computed in classical polynomial time for any value of l, as we show in the next section. In this case l can just be taken to be the largest integer so that G has girth at least $2l + 1$. We are now in position to present the full DQI algorithm for MaxCut.

▶ **Algorithm 2** (Decoded Quantum Interferometry for MaxCut).
Input: *unweighted graph* $G = (V, E)$
1. *Compute the girth* g *of* G*, and set* $l \stackrel{\text{def}}{=} \lfloor \frac{g-1}{2} \rfloor$
2. *Solve an* $(l + 1) \times (l + 1)$ *maximum eigenvector problem to find the coefficients of an optimal degree-*l *symmetric polynomial* q *in* y
3. *Run the algorithm of Lemma 1 to prepare* $|\hat{q}\rangle_V$
4. *Run QFS (Algorithm 1) on* $|\hat{q}\rangle_V$ *to sample a cut* z *with probability proportional to* $q^2(z)$

2.2 Solving the classical portion of DQI

Solving the following problem suffices to implement the uncompute step (Step 3) of the algorithm of Lemma 1.

▶ **Problem 1** (DQI uncomputation for MaxCut).
Input: *A parameter* $l \in \mathbb{N}$*, an unweighted graph* $G = (V, E)$*, and a vector* $\alpha \in \mathbb{F}_2^V$
Output: *A subgraph with at most* l *edges, represented as* $\beta \in \mathbb{F}_2^E$*, whose degree parities match* α *(i.e.,* $|\beta| \leq l$ *and* $f(\beta) \stackrel{\text{def}}{=} B^T\beta = \alpha$*, where* $B \in \mathbb{F}_2^{E\times V}$ *is the edge-vertex incidence matrix of* G*)*
Remark: *Additionally assuming that* G *has girth at least* $2l + 1$ *guarantees a unique solution for feasible instances by Lemma 2.*

▶ Remark 3 (Classical decoding in DQI). DQI is originally framed in terms of a classical decoding problem for an error-correcting code. For MaxCut this perspective stems from the edge-incidence matrix B. The column space of B generates the *cut space* of G, which is the span over $\mathbb{F}_2$, of incidence vectors of cuts in G. Since any cycle crosses a cut an even number of times, the cut space is orthogonal to the *cycle space*, generated in the same fashion by incidence vectors of cycles. This space may be viewed as a cycle code space and is

characterized as incidence vectors of subgraphs of G which have even degree at every vertex. The parity-check matrix of this code is B^T, and the girth of G is the distance of the code. This code is usually called a graph code or cycle code.

In this picture, Problem 1 is about finding a short vector generating a given syndrome, and as observed in [11] it can also be solved as a more standard maximum-likelihood decoding problem. Finally, observe the following duality: for DQI to solve an optimization problem over cuts (MaxCut), it must solve a kind of decoding problem over the dual cycle space (Problem 1). See, e.g., Section 1.9 of [3] for more on the cut and cycle spaces of a graph.

We illustrate how Problem 1 can be cast as a T-join problem [4], which can be solved in polynomial time by a reduction to a perfect matching problem (see, e.g., [21, Theorem 29.1]).

▶ **Problem 2** (Minimum T-join)**.**
Input: *An unweighted graph $G = (V, E)$ and a subset of vertices $T \subseteq V$*
Output: *A subgraph H of G, with the minimum number of edges, such that set of odd-degree nodes in H is precisely T*

▶ **Fact 1.** *Problem 1 is polynomial-time reducible to Problem 2.*

Proof. Given an instance of Problem 1, set $T \stackrel{\text{def}}{=} \{i \in V \mid \alpha_i = 1\}$ and solve Problem 2 on G, T. If this instance of Problem 2 is infeasible, or if H has more than l edges, then Problem 1 is infeasible. Otherwise, take β as the incidence vector of H. ◀

This is the last piece in establishing:

▶ **Theorem 4.** *DQI for MaxCut (Algorithm 2) is correct and runs in polynomial time on any input graph G.*

Proof. The girth of an unweighted undirected graph can be found by running n breadth-first searches, one rooted at each vertex [10], so Steps 1 and 2 run in classical polynomial time. Since $g \geq 2l + 1$, f is injective by Lemma 2, and Step 3 correctly produces $|\hat{q}\rangle_V$. This step also runs in polynomial time since Problem 1 runs in classical polynomial time by Fact 1. QFS (Step 4) runs in polynomial time and samples from q^2 as explained in [8]. The fact that Step 2 produces an optimal symmetric degree-l polynomial q comes from the proof of [11, Lemma 9.2], although this is not necessary to prove the approximation guarantee of Algorithm 2. ◀

We now move on to considering the approximation ratio that Algorithm 2 guarantees.

2.3 Approximation guarantee

The expected number of edges cut by DQI beyond the random-assignment guarantee of $\frac{1}{2}m$ is captured by the maximum eigenvalue of a tridiagonal matrix:

▶ **Lemma 5** (Specialization of Lemma 9.2 in [11] for MaxCut)**.** *Let q be the polynomial obtained in Step 2 of Algorithm 2. Then the expected number of edges cut by Algorithm 2 is*

$$\mathbb{E}_{z\sim q^2}[h(z)] = \frac{1}{2}(m + \lambda_{\max}(A^{(m,l)})), \tag{3}$$

where

$$A^{(m,l)} \stackrel{\text{def}}{=} \begin{bmatrix} & a_1 & & & \\ a_1 & & a_2 & & \\ & a_2 & & \ddots & \\ & & \ddots & & a_l \\ & & & a_l & \end{bmatrix},$$

with $a_k = \sqrt{k(m-k+1)}$ for $1 \leq k \leq l$.

We show that DQI needs instances with $l = \Omega(m)$ in order to attain a asymptotically nontrivial approximation guarantee.

▶ **Lemma 6.** *For matrices $A^{(m,l)}$ with $l = o(m)$, $\lambda_{\max}(A^{(m,l)}) = o(m)$.*

Proof. Since $A^{(m,l)}$ is nonnegative, $\lambda_{\max}(A^{(m,l)}) \leq \sqrt{m}\lambda_{\max}(B^{(l)})$, where

$$B^{(l)} \stackrel{\text{def}}{=} \begin{bmatrix} & \sqrt{1} & & & \\ \sqrt{1} & & \sqrt{2} & & \\ & \sqrt{2} & & \ddots & \\ & & \ddots & & \sqrt{l} \\ & & & \sqrt{l} & \end{bmatrix}.$$

The matrix, $\frac{1}{\sqrt{2}}B^{(l)}$ is the $(l+1) \times (l+1)$ Jacobi matrix corresponding to the three-term recurrence relation for the normalized monic Hermite polynomials [16, 3.5(vi)]. The maximum eigenvalue of $\frac{1}{\sqrt{2}}B^{(l)}$ is consequently the largest zero of the degree-$(l+1)$ monic Hermite polynomial, which is $O(\sqrt{l})$ [16, 18.16(v)]. Thus $\lambda_{\max}(A^{(m,l)}) = O(\sqrt{ml})$. Since $l = o(m)$, we have $O(\sqrt{ml}) = o(m)$. ◀

From the two lemmas above, we have:

▶ **Theorem 7.** *If the expected number of edges cut by DQI in a graph $G \in \mathcal{G}$ is $(\frac{1}{2} + \varepsilon)m$, for some constant $\varepsilon > 0$, the girth g must be linear in m.*

Proof. If we do not have linear girth, $g = \Omega(m)$, then $g = o(m)$ since $\lim_{m\to\infty} g/m$ exists. In this case $l \leq \frac{g-1}{2} = o(m)$, and by the above lemmas, the expected number of edges cut by DQI is $\frac{1}{2}(1 + o(1))m$. ◀

▶ Remark 8. The approximation guarantee for DQI is expressed with respect to the trivial upper bound, m on the optimal cut since DQI does not leverage more refined upper bounds such as those arising from convex programs. However, this means that DQI provides a lower bound on the number of edges in a maximum cut and more generally on the number of satisfiable constraints in a constraint satisfaction problem, even when DQI is not guaranteed to run in polynomial time.

3 Classical solvability of high-girth instances

High girth has been previously exploited for solving MaxCut both classically and using the QAOA [5, 23, 7]; however, "high" in such contexts typically refers to constant or perhaps logarithmic girth, as expected in some kinds of random graphs. In contrast, in order for DQI to guarantee $(\frac{1}{2} + \varepsilon)m$ edges in expectation, we know that we must have girth $g = \Theta(n) = \Theta(m)$. We show that in this regime, MaxCut can be solved exactly in polynomial time. In fact, the approaches we describe can also solve weighted instances, while it is nontrivial to extend DQI to handle weighted instances.

▶ **Lemma 9.** *Suppose $G \in \mathcal{G}$ is a connected graph with girth $g = \Omega(m)$. Then the vertices of G can be partitioned into vertex-disjoint trees $T_1, \ldots, T_t$ where: (i) $t = O(1)$, and (ii) there is at most one edge in G between any two T_i and T_j.*

Proof. Set $d \stackrel{\text{def}}{=} \lfloor \frac{g-3}{4} \rfloor$, and find a spanning tree T of G. Then run the following algorithm.

While T has depth at least d:

1. Find the deepest leaf u in T, and let S_i be the subtree rooted at the ancestor of u at distance d from u

2. Let T_i be the subgraph of G induced by the vertices of S_i
3. Remove S_i from T and set $i = i + 1$

When the loop has terminated, set T_i to be subgraph induced by T itself. Let $t \stackrel{\text{def}}{=} i$ at this point.

We are guaranteed that each tree S_i, except possibly S_t, has depth exactly d, since we picked u to be the deepest leaf in T, and therefore in S_i, in each iteration. The last tree S_t has depth at most d.

To prove (i), since the trees $S_1, \ldots, S_{t-1}$ have depth d, these disjoint trees each have at least d edges. This gives us $t = O(1)$ since $g = \Omega(m)$ implies $d = \Omega(m)$ (the hidden constants for t and d only depend on those for g).

Next observe that each T_i can contain no edge of $G - T$ and is consequently a tree. Since each T_i is connected by edges in T and has depth at most d, if one contained some edge in $G - T$ there would be a cycle with at most $2d + 1 < g$ edges, contradicting that the girth is g. The proof of (ii) is similar to this. Two edges between distinct T_i and T_j would create a cycle of length at most $4d + 2 < g$. ◀

▶ **Theorem 10.** *For a connected graph $G \in \mathcal{G}$ with girth $g = \Omega(m)$, the quantity $\mu \stackrel{\text{def}}{=} m - n + 1$ is constant.*

Proof. Let T be a spanning tree, and construct $T_1, \ldots, T_t$ from T as specified in Lemma 9. Since each T_i is a subgraph of T, each edge in $G - T$ must go between distinct T_i and T_j. Conditions (i) and (ii) then imply that there is only a constant number of such edges. In other words, μ is constant. ◀

▶ **Corollary 11.** *MaxCut can be solved in polynomial time on a graph $G \in \mathcal{G}$ with girth $g = \Omega(m)$.*

Proof. Each component of G must also have girth $\Omega(m)$, so we can apply Theorem 10 to each component individually. MaxCut can be solved in time $O(m + n) \min\{2^{m/5}, 2^{(m-n)/2}\}$ on a connected graph [22, Theorem 2]. ◀

Other approachs for high-girth MaxCut

Theorem 10 enables other approaches for solving MaxCut in polynomial time. The fundamental cycles of a spanning tree T are the set of the μ unique cycles created by adding each of the μ non-tree edges to T. These cycles form a basis for all cycles over $\mathbb{F}_2^E$ (see Remark 3). Since μ is constant for graphs with linear girth, so are the total number of cycles, which is at most 2^μ. This allows solving positive-weighted MaxCut as a binary integer program with a constant number of constraints:

$$\begin{aligned} \max \quad & \sum_{ij \in E} w_{ij} x_{ij} \\ \text{subject to} \quad & \sum_{ij \in C} x_{ij} \leq |C| - 1, \ \forall \text{ odd cycles } C \\ & x_{ij} \in \{0, 1\}, \ \forall ij \in E, \end{aligned}$$

which can be done in polynomial time [18]. The constraints ensure that a bipartite subgraph is selected by preventing picking all the edges of any odd cycle.

Perhaps the simplest way to solve high-girth instances of MaxCut is via a dynamic program. Pick a spanning tree T. Cutting all the edges of the tree already gives a cut of size $m - O(1)$, since μ is constant. Enumerate all ways of assigning -1 or $+1$ to the endpoints of

the μ non-tree edges. For each assignment, one can solve a dynamic program that works bottom up from the leaves to find the best cut of the edges of T assuming that some nodes have been fixed by an assignment. This works for weighted MaxCut as well.

4 Discussion

The limitation presented here applies when the decoding portion of DQI is always required to succeed. A natural question is whether some extension of DQI, including quantum decoding, might be able to get around the barriers we presented. One can consider probabilistic decoding approaches for DQI that need not always succeed [11, 2]. Such approaches can conceivably tolerate some cycles of smaller girth, possibly extending the kinds of instances of MaxCut that DQI can address. However, some of the classical approaches from Section 3 work even when $\mu = \Theta(\log n)$ rather than $\mu = O(1)$. This, for instance, implies that MaxCut on any graph with a logarithmic number of cycles (of any size) can be solved efficiently. Some instances with a polynomial number of cycles may also be solvable, since it is possible to have a polynomial cycles with a fundamental cycle basis of size $\mu = \Theta(\log n)$. Thus classical approaches seem fairly robust in terms of handling cycles, and it is not clear if DQI would be able to do better.

Is MaxCut an isolated case, or are there more general ways for classical algorithms to leverage the same structure that DQI does? The particular property we leverage for MaxCut is special and seems difficult to generalize to other types of Max-LINSAT problems. The prime candidates for a DQI advantage, such as Optimal Polynomial Intersection, should be further vetted by attempts at designing better classical approximation algorithms in the regimes where running DQI is feasible and efficient.

References

1 Eric R. Anschuetz, David Gamarnik, and Jonathan Z. Lu. Decoded quantum interferometry requires structure. *arXiv*, 2025. arXiv:2509.14509.

2 André Chailloux and Jean-Pierre Tillich. Quantum advantage from soft decoders. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing (STOC 2025)*. ACM, 2025. Extended version available as arXiv:2411.12553. doi:10.1145/3717823.3718319.

3 Reinhard Diestel. *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, 6 edition, 2025. doi:10.1007/978-3-662-70107-2.

4 Jack Edmonds and Ellis L. Johnson. Matching, Euler tours and the Chinese postman. *Math. Program.*, 5:88–124, 1973. doi:10.1007/BF01580113.

5 Ahmed El Alaoui, Andrea Montanari, and Mark Sellke. Local algorithms for MaxCut and minimum bisection on locally treelike regular graphs of large degree. *Random Struct. Algorithms*, 63(3):580–645, 2023. doi:10.1002/rsa.21110.

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

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

8 Bill Fefferman and Christopher Umans. On the power of quantum fourier sampling. In *Proceedings of the 11th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2016)*, pages 1–19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.TQC.2016.1.

9 Michel X. Goemans and David P. Williamson. Improved approximation algorithms for MaxCut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145, 1995. doi:10.1145/227683.227684.

10 Alon Itai and Michael Rodeh. Finding a minimum circuit in a graph. *SIAM J. Comput.*, 7(4):413–423, 1978. doi:10.1137/0207033.

11 Stephen P. Jordan, Noah Shutty, Mary Wootters, Adam Zalcman, Alexander Schmidhuber, Robbie King, Sergei V. Isakov, Tanuj Khattar, and Ryan Babbush. Optimization by decoded quantum interferometry. *arXiv*, 2024. arXiv:2408.08292.

12 John Kallaugher, Ojas Parekh, and Nadezhda Voronova. Exponential quantum space advantage for approximating maximum directed cut in the streaming model. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC 2024)*, pages 1805–1815. ACM, 2024. doi:10.1145/3618260.3649656.

13 John Kallaugher, Ojas Parekh, and Nadezhda Voronova. How to design a quantum streaming algorithm without knowing anything about quantum computing. In *Proceedings of the 2025 Symposium on Simplicity in Algorithms (SOSA 2025)*, pages 9–45. Society for Industrial and Applied Mathematics, 2025. doi:10.1137/1.9781611977998.2.

14 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MaxCut and other 2-variable CSPs? *SIAM J. Comput.*, 37(1):319–357, 2007. doi:10.1137/S0097539705447372.

15 Kunal Marwaha, Bill Fefferman, Alexandru Gheorghiu, and Vojtech Havlicek. On the complexity of decoded quantum interferometry. *arXiv*, 2025. doi:10.48550/arXiv.2509.14443.

16 Nist digital library of mathematical functions, 2025. Release 1.2.3 of 2025-09-15, F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. URL: https://dlmf.nist.gov/.

17 Ryan O'Donnell. Analysis of Boolean functions. *arXiv*, 2021. arXiv:2105.10386.

18 Christos H. Papadimitriou. On the complexity of Integer Programming. *J. ACM*, 28(4):765–768, 1981. doi:10.1145/322276.322287.

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

20 Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005)*, pages 84–93. ACM, 2005. doi:10.1145/1060590.1060603.

21 Alexander Schrijver. *Combinatorial optimization: polyhedra and efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003. doi:10.1007/978-3-540-68279-0.

22 Alexander D. Scott and Gregory B. Sorkin. Faster algorithms for MaxCut and MaxCSP, with polynomial expected time for sparse instances. In *Proceedings of the International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM-APPROX 2003)*, pages 382–395. Springer, 2003. doi:10.1007/978-3-540-45198-1_29.

23 Jessica K. Thompson, Ojas Parekh, and Kunal Marwaha. An explicit vector algorithm for high-girth MaxCut. In *Proceedings of the SIAM Symposium on Simplicity in Algorithms (SOSA 2022)*, pages 148–159. Society for Industrial and Applied Mathematics, 2022. doi:10.1137/1.9781611977066.17.