

Quantum Max-Cut is NP-hard to approximate

Stephen Piddock

Department of Computer Science, Royal Holloway, London, UK

October 10, 2025

Abstract

We unconditionally prove that it is NP-hard to compute a constant multiplicative approximation to the QUANTUM MAX-CUT problem on an unweighted graph of constant bounded degree. The proof works in two stages: first a generic reduction to computing the optimal value of a quantum problem, from the optimal value over product states. Then an approximation preserving reduction from MAX-CUT to PRODUCT-QMC, the product state version of QUANTUM MAX-CUT.

More precisely, in the second part, we construct a PTAS reduction from MAX-CUT_k (the rank- k constrained version of MAX-CUT) to MAX-CUT_{k+1}, where MAX-CUT and PRODUCT-QMC coincide with MAX-CUT₁ and MAX-CUT₃ respectively. We thus prove that MAX-CUT_k is APX-complete for all constant k .

1 Introduction

QUANTUM MAX-CUT is a quantum generalisation of the classical MAX-CUT problem. In MAX-CUT, the aim is to assign values $s_i \in \{\pm 1\}$ for each vertex in a graph G in order to maximise the number of edges ij with different signs $s_i \neq s_j$. More precisely, given a weighted graph $G = (V, E, w)$, with $|V| = n$ and $|E| = m$

$$\text{MAX-CUT}(G) = \max_{s \in \{\pm 1\}^n} \sum_{ij \in E} w_{ij} \frac{1}{2} (1 - s_i s_j).$$

QUANTUM MAX-CUT is similar except that there is qubit at each vertex of the graph and the problem is to compute $\text{OPT}(H)$, the maximum energy $\text{Tr}(H\rho)$ over n -qubit quantum states ρ with respect to the Hamiltonian H :

$$H = \sum_{ij \in E} w_{ij} \frac{1}{4} (I - X_i X_j - Y_i Y_j - Z_i Z_j).$$

where X, Y, Z are the Pauli matrices and the subscript i denotes the action of an operator on the i th qubit. Here the weight w_{ij} is interpreted as the strength of the interaction $I - X_i X_j - Y_i Y_j - Z_i Z_j$.

Computing $\text{OPT}(H)$ exactly is equivalent to calculating the maximum eigenvalue of H , or the ground state energy of $-H$, an antiferromagnetic Heisenberg Hamiltonian. The ground state energy is the physically relevant quantity, and the “Local Hamiltonian problem” is the task of computing the ground state energy to inverse polynomial accuracy. Here we focus on the maximisation

formulation as it mirrors the classical MAX-CUT problem, and is better suited to approximation algorithms.

As the only two local qubit interaction with $SU(2)$ symmetry (i.e. it is invariant under conjugation by $U \otimes U$ for any $U \in SU(2)$), the Heisenberg interaction is in a sense one of the simplest and most natural two-qubit interactions. It is closely related to the Fermi-Hubbard model at half-filling, a simple but important model of quantum magnetism.

Due to its symmetry, the Heisenberg interaction plays a central part in the proof of the complexity classification of local Hamiltonian problems [1], and in the construction of universal qubit Hamiltonians [2]. The decision variant of QUANTUM MAX-CUT is QMA-complete [3], where the aim is to decide if $\text{OPT}(H)$ is above a or below b for thresholds $a, b \in \mathbb{R}$ with $a - b \geq 1/\text{poly}(n)$. A major drawback of this result is that the weights w_{ij} vary throughout the graph, becoming as large as $\text{poly}(n)$. This distances the model from the most physically relevant setting where the weights are all equal to 1.

Since it is hard to compute QUANTUM MAX-CUT to additive error, attention has turned to approximation algorithms [4, 5, 6, 7, 8]. These algorithms do not find the true optimal solution, but find states ρ which are guaranteed to achieve an energy $\text{Tr}(H\rho)$ that comes within a constant multiplicative factor α of the optimal value $\text{QUANTUM MAX-CUT}(G)$.

The approximation algorithm of [4] is based on the famous Goemans-Williamson approximation algorithm [9] for MAX-CUT, and its generalisation [10]. For MAX-CUT, the Goemans-Williamson algorithm is known to be optimal in the sense that, assuming the unique games conjecture, it is NP-hard to get a better approximation ratio for MAX-CUT [11]. Even without the unique games conjecture, MAX-CUT is known to be APX-complete [12], meaning it is as hard as any problem in APX, the class of classical optimisation problems with an efficiently achievable constant approximation ratio. And Håstad proved that it is NP-hard to approximate MAX-CUT for any approximation ratio better than $16/17$ [13].

For QUANTUM MAX-CUT, the situation is very different. A number of extensions and improvements have been developed since the algorithm of [4], with the best achievable approximation ratio increasing over time [5, 6, 7, 8], and the simple SDP based method of [4] proving not to be optimal.

On the hardness side, we currently have no way to extend the QMA-hardness result of [3] to the regime of constant approximation ratio algorithms, since this would require proving a quantum version of the PCP theorem, which appears to be beyond the reach of current tools. Although QUANTUM MAX-CUT is introduced as a generalisation of MAX-CUT, we do not have any way to directly embed MAX-CUT instances into QUANTUM MAX-CUT instances, and so we do not inherit any of the NP-hardness results from MAX-CUT. Recently, there has been an attempt [14] at upper bounding the approximation ratio for QUANTUM MAX-CUT assuming the unique games conjecture, analogous to [11].

Is QUANTUM MAX-CUT hard when the interaction strengths are $O(1)$, or even uniform? Is it hard to approximate up to a constant approximation ratio? Can we prove a more direct reduction from MAX-CUT to QUANTUM MAX-CUT? We remark that the first and third question here are the essence of open questions 3.2 and 3.3 from Gharibian's recent survey paper on quantum versions of NP [15].

We resolve these questions in our main result:

Theorem 1. *There is a constant $\alpha < 1$ such that it is NP-hard to compute a value that approximates the optimal QUANTUM MAX-CUT value within approximation ratio α . This holds even when the graph G is unweighted ($w_{ij} = 1$ for all $ij \in E$) and each vertex has bounded (constant) degree.*

We believe that this result is the first hardness of approximation result for a family of quantum local Hamiltonian problems that does not require any conjectures or include a known hard-to-approximate classical CSP in that family.

Theorem 1 is proven by reducing from MAX-CUT to QUANTUM MAX-CUT, although the reduction is not completely direct. As an intermediate problem, we consider the product state variant of QUANTUM MAX-CUT.

First we show that approximating $\text{QUANTUM MAX-CUT}(G) = \text{OPT}(H)$ is at least as hard as approximating $\text{PRODUCT-QMC}(G) = \text{OPT}_{\text{prod}}(H)$, the maximal value over product states:

$$\text{PRODUCT-QMC}(G) = \text{OPT}_{\text{prod}}(H) = \max_{\rho = \rho_1 \otimes \rho_2 \otimes \dots \otimes \rho_n} \text{Tr}(H\rho)$$

We do this by taking a graph G , and replacing each vertex i with a cloud of vertices T_i where $|T_i| = T$, with edges between all vertices in T_i and T_j iff $ij \in E$. The new graph G' has the same rough structure as G , but now each vertex has high degree, and since each qubit cannot be highly entangled with its neighbours, product states are now close to optimal.

In fact the same idea applies generically to any k -local Hamiltonian and we prove the following theorem:

Theorem 2. *Let $H = \sum_{\mathbf{i} \in I} h_{\mathbf{i}}$ be a k -local qubit Hamiltonian with $h_{\mathbf{i}} \geq 0$ for all $\mathbf{i} \in I$. Then one can construct H' using the interactions $\{h_{\mathbf{i}}\}$ such that*

$$T^k \text{OPT}_{\text{prod}}(H) \leq \text{OPT}(H') \leq (T + 2)^k \text{OPT}_{\text{prod}}(H)$$

We remark that Theorem 2 has a very similar flavour to the results of Brandao and Harrow [16]. Indeed, their Corollary 4 shows that for a Hamiltonian with an interaction graph of high degree, there is a product state which is nearly optimal. With an additional argument relating the optimal product state for H' and the optimal product state for H , one could prove a variant of Theorem 2 using their result.

However we instead provide a novel proof of Theorem 2 based on the representation theory of $SU(2)$ and an inequality due to Lieb [17]. This has the advantage of being simple, easily extends the results to $k > 2$ and achieves better constant factors. The disadvantage of this method is that it apparently cannot be generalised to systems of qudits¹.

The recent work of [18] classifies the complexity of the product state problem for $\mathcal{S}$ -Hamiltonians, the family of Hamiltonians consisting of interactions from a fixed set $\mathcal{S}$ (with positive and negative weights allowed). Therefore as a corollary of Theorem 2, we have:

Corollary 1. *Let $\mathcal{S}$ be a set of two qubit interactions. Then the Local Hamiltonian problem for $\mathcal{S}$ -Hamiltonians (i.e. determining if the ground state energy is below a or above b for $b - a \geq 1/\text{poly}(n)$) is:*

1. *in P, if all elements of $\mathcal{S}$ are 1-local.*
2. *NP-hard otherwise. Furthermore, this holds even when the interaction strengths are $O(1)$.*

Note that this corollary is weaker than the classification of [1], except that here the interaction strengths are constant.

¹Qudits are quantum subsystems with a complex state space of dimension d . For example, qubits are qudits of dimension $d = 2$.

Armed with Theorem 2, our focus in the second part of the paper is to prove hardness of approximation for finding the optimal product state $\rho_1 \otimes \rho_2 \otimes \dots \otimes \rho_n$ for $\text{PRODUCT-QMC}(G)$. By decomposing a single qubit density matrix in the Pauli basis as $\rho = \frac{1}{2}(I + x_1X + x_2Y + x_3Z)$, we can identify ρ with the 3-dimensional unit vector of Pauli coefficients $\mathbf{x}$, and the product state $\rho_1 \otimes \rho_2 \otimes \dots \otimes \rho_n$ with the list of unit vectors $\vec{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$.

The rank- k constrained MAX-CUT problem, which we denote MAX-CUT_k is a generalisation of MAX-CUT, where each vertex is assigned a unit vector in $\mathbb{R}^k$, i.e. an element of the sphere S_{k-1} .

$$\text{MAX-CUT}_k(G) = \max_{\vec{\mathbf{x}}=(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n) \in (S_{k-1})^n} \sum_{ij \in E} \frac{1}{2}(1 - \mathbf{x}_i \cdot \mathbf{x}_j)$$

Therefore, as has been observed previously (for example in [4]), $\text{MAX-CUT}_3(G) = 2 \times \text{PRODUCT-QMC}(G)$; and note that $\text{MAX-CUT} = \text{MAX-CUT}_1$. Briët, de Oliveira Filho and Vallentin showed how the Goeman-Williams approximation algorithm for MAX-CUT can be generalised to MAX-CUT_k for every k [10]. Strong evidence was provided in [14] that the approximation algorithms of [10] are optimal: they proved that, assuming the unique games conjecture, and a newly conjectured inequality referred to as the vector-valued Borel inequality, it is NP-hard to approximate MAX-CUT_k to any better approximation ratio.

However, without these assumptions, the hardness of MAX-CUT_k is less well understood. While MAX-CUT_1 was on Karp's original list of NP-complete problems [19], and Lovász and others have conjectured that MAX-CUT_k is hard for other values of k (see e.g. p236 of [20]), it is only very recently that MAX-CUT_3 was proven to be NP-complete for additive inverse polynomial accuracy [18], and no other hardness results are known for $k \neq 1, 3$.

Furthermore no hardness of approximation results are known without additional assumptions, except for $k = 1$, where it is known that MAX-CUT is APX-complete [12]. We extend these results to all $k > 1$ in Theorem 3.

Theorem 3. *There is a PTAS reduction from MAX-CUT_k to MAX-CUT_{k+1} . Therefore MAX-CUT_k is APX-complete for all (constant) $k \geq 1$.*

We remark that $\text{MAX-CUT}_2 = 2 \times \text{PRODUCT-QMC}_{XY}(G)$, the problem of finding the optimal product state for the Hamiltonian H_{XY} :

$$H_{XY} = \sum_{ij \in E} w_{ij} \frac{1}{4}(I - X_i X_j - Y_i Y_j)$$

and thus we also get a hardness of approximation result for $\text{PRODUCT-QMC}_{XY}(G)$.

To reduce from MAX-CUT_k to MAX-CUT_{k+1} , it is necessary to find a way to effectively reduce the available dimensions at each vertex in a MAX-CUT_{k+1} instance. One natural way to do this is to look for a gadget which forces two vertices to be orthogonal (or approximately orthogonal). Then by selecting a reference vertex 0, and applying this gadget to all pairs $i, 0$, one can ensure that $\mathbf{x}_i$ is orthogonal to $\mathbf{x}_0$ for all i , and thus the $\mathbf{x}_i$ are effectively in a lower dimensional space. Wright [21] had suggested a gadget based on a long odd length cycle for this task.

Our insight is to recognise that it is not necessary to make $\mathbf{x}_i$ and $\mathbf{x}_0$ orthogonal, but it is enough to make sure that $\mathbf{x}_i$ and $\mathbf{x}_0$ are at a fixed angle from each other for all i . For this task, almost any gadget will do, as long as it is not a bipartite graph, and we focus on the simplest example of a triangle. In a triangle graph, the optimal assignment for MAX-CUT_k has an angle of $2\pi/3$ between each of the vertices on the triangle.

We sketch a simplified version of our construction based on this idea. Take an instance G of MAX-CUT_k and add a vertex c_i for each i and an extra vertex 0 . Include the original edges of G , and add heavily weighted edges between the vertices of the triangle $0, i, c_i$ for all i to create a new graph G' . The heavily weighted triangle gadgets mean that any assignment $\vec{x}$ close to optimal must have $\mathbf{x}_i = \cos(\theta_i)\mathbf{x}_0 + \sin(\theta_i)\tilde{\mathbf{x}}_i$ for some $\tilde{\mathbf{x}}_i$ orthogonal to $\mathbf{x}_0$, and $\theta_i \approx 2\pi/3$. Since the $\tilde{\mathbf{x}}_i$ are orthogonal to $\mathbf{x}_0$, we can interpret them living in $\mathbb{R}^k$, rather than $\mathbb{R}^{k+1}$.

Therefore the optimal value $\text{MAX-CUT}_{k+1}(G')$ will be approximately equal to a simple function of $\text{MAX-CUT}_k(G)$. Furthermore by projecting the vector $\mathbf{x}_i$ onto the space orthogonal to $\mathbf{x}_0$ and renormalising for all i , we have a map $g : (S_k)^{n+1} \rightarrow (S_{k-1})^n$ which maps good solutions for $\text{MAX-CUT}_{k+1}(G')$ to good solutions for $\text{MAX-CUT}_k(G)$. This is essential for showing that the construction is a PTAS reduction.

While this sketch can be formalised into a PTAS reduction and hence prove MAX-CUT_k is APX-hard, there are a couple of drawbacks of this simple approach. Firstly, the heavily weighted terms are undesirable: although they only need to be large constant values (not scaling with n , the number of vertices), we want a construction where all weights are equal to 1. Second, the reference vertex 0 has very high degree as it is connected to all other vertices.

We overcome these difficulties by replacing the single reference vertex 0 with a large bipartite expander graph of additional vertices. The optimal assignment for a bipartite graph is to have all vertices on one side of the bipartition set to the same vector $\mathbf{z}$ and all vertices on the other side of the partition set to $-\mathbf{z}$. Rather than have a heavily weighted triangle between i and 0 , we instead have multiple unweighted triangles between i and different vertices from one side of the bipartite graph. This then reproduces the behaviour of the simple reduction but without any weighted edges or high degree vertices.

We briefly contrast our proof method to that of [18], where in their Theorem 1.5, they prove that it is NP-hard to determine the optimal value of MAX-CUT_3 up to additive inverse polynomial accuracy. Rather than reduce from MAX-CUT , their method instead reduces from 3-COLORING . The construction involves taking an instance G of 3-COLORING , and replacing each edge ij with a clique of four vertices, then taking that new graph and replacing each edge with a clique of three vertices. Finally an additional vertex is added and connected to some of the vertices. This final vertex plays a similar role to the vertex 0 described in the sketch of our construction above, in that it provides a reference state to fix a basis for the other vertices.

Our proof method is arguably simpler, provides a direct connection to MAX-CUT , and obtains the stronger result of APX-completeness for MAX-CUT_3 .

2 Preliminaries

2.1 $\text{SU}(2)$ representation theory

The Lie group $\text{SU}(2)$ is the group of unitary 2×2 matrices with determinant 1. The associated Lie algebra $\mathfrak{su}(2)$ is the space of traceless anti-hermitian 2×2 matrices. Equivalently, any element $A \in \mathfrak{su}(2)$ can be decomposed as $A = a_1\sigma^{(1)} + a_2\sigma^{(2)} + a_3\sigma^{(3)}$ for real numbers a_1, a_2, a_3 and where the $\sigma^{(j)}$ are the Pauli matrices:

$$\sigma^{(1)} = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^{(2)} = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^{(3)} = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

For convenience, we will also write $\sigma^{(0)}$ for the 2×2 identity matrix $\sigma^{(0)} = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$, so that $\{\sigma^{(j)}\}_{j=0}^4$ is a basis for all hermitian 2×2 matrices.

Note that $\mathfrak{su}(2)$ is closed under the action of the matrix commutator (sometimes referred to as the Lie bracket) $[A, B] = AB - BA$.

A *representation* R of dimension d is a map $R : \mathfrak{su}(2) \rightarrow M(\mathbb{C}^d)$, where $M(\mathbb{C}^d)$ is the space of complex square matrices of dimension d , such that

$$[R(A), R(B)] = [A, B] \quad \text{for all } A, B \in \mathfrak{su}(2).$$

The representation R is *irreducible* if there is no non-trivial subspace $W \subset \mathbb{C}^d$ for which $R(A)$ maps W to itself for all $A \in \mathfrak{su}(2)$.

For each positive integer d , there exists a unique (up to isomorphism) irreducible representation of $\mathfrak{su}(2)$ of dimension d . We denote this d -dimensional irreducible representation R_J where $J = \frac{d-1}{2}$, and will refer to it as the *spin- J representation*, matching the terminology common in physics.

The Lie algebra $\mathfrak{su}(2)$ is *semi-simple* which means that any representation R is isomorphic to a direct sum of irreducible representations. Concretely, for a representation R there is a unitary basis change U on $\mathbb{C}^d$ such that:

$$R(A) = U \left(\bigoplus_J R_J(A) \right) U^\dagger \quad \text{for all } A \in \mathfrak{su}(2). \quad (1)$$

Given two representations R_1 and R_2 one can define the *tensor product representation* $R_1 \otimes R_2$ on $\mathbb{C}^{d_1} \otimes \mathbb{C}^{d_2}$:

$$R_1 \otimes R_2(A) = R_1(A) \otimes I_{d_2} + I_{d_1} \otimes R_2(A)$$

We are particularly interested in the representation $\hat{R}$ produced by taking the tensor product of T copies of the spin- $\frac{1}{2}$ representation. This acts on the space $(\mathbb{C}^2)^{\otimes T}$, which we can identify with the space of T qubits:

$$\hat{R}(A) = \sum_{t=1}^T A_t$$

where A_t denotes the operator A acting on qubit t , and identity elsewhere.

The decomposition of $\hat{R}$ into irreducible representations as in equation (1) is very well understood. In fact, it is best described via Schur-Weyl duality, which connects to the representation theory of the symmetric group. However all we need is that there exists a unitary U such that

$$\hat{R}(A) = U \left(\bigoplus_{J \in \mathcal{T}} R_J(A) \otimes I_{g_J} \right) U^\dagger \quad \text{for all } A \in \mathfrak{su}(2). \quad (2)$$

where I_{g_J} is the identity matrix on a space of dimension g_J and $\mathcal{T} = \{T/2, T/2 - 1, T/2 - 2, \dots\}$ (continuing down to 0 if T is even, or $1/2$ if T is odd).

2.2 Local Hamiltonians and Lieb's inequality

Let H be a k -local Hamiltonian on n qubits. Let I be a subset of $[n]^k$ so that H can be written as $H = \sum_{\mathbf{i} \in I} h_{\mathbf{i}}$ for interactions $h_{\mathbf{i}}$ which act non trivially only on the qubits in $\mathbf{i} = (i_1, i_2, \dots, i_k)$. Each $h_{\mathbf{i}}$ can be written in the Pauli basis as

$$h_{\mathbf{i}} = \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} \sigma_{i_1}^{(a_1)} \sigma_{i_2}^{(a_2)} \dots \sigma_{i_k}^{(a_k)} \quad (3)$$

for some order k tensor $M(\mathbf{i})$.

To introduce Lieb's inequality, we define $H(\mathbf{J})$ and $\tilde{H}(\mathbf{J})$ two variants of H , both of which depend on a vector $\mathbf{J} = (J_1, J_2, \dots, J_n) \in (\mathbb{Z}_{\geq 0}/2)^n$.

First to define $H(\mathbf{J})$, we imagine that qubit i is replaced by a spin- J_i particle (i.e. a qudit of dimension $d_i = 2J_i + 1$). Then $H(\mathbf{J})$ looks the same as H , except that each $\sigma_i^{(a)}$ is replaced with $R_{J_i}(\sigma^{(a)})_i$.

$$H(\mathbf{J}) = \sum_{\mathbf{i} \in I} \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} R_{J_{i_1}}(\sigma^{(a)})_{i_1} R_{J_{i_2}}(\sigma^{(a)})_{i_2} \dots R_{J_{i_k}}(\sigma^{(a)})_{i_k} \quad (4)$$

The other variant $\tilde{H}(\mathbf{J})$ still acts on n qubits and has the same structure as the original Hamiltonian H , but each $\sigma_i^{(a)}$ is replaced by $2J_i \sigma_i^{(a)}$. Equivalently $\tilde{H}(\mathbf{J})$ can be defined as:

$$\tilde{H}(\mathbf{J}) = \sum_{\mathbf{i} \in I} \left(\prod_{l=1}^k 2J_{i_l} \right) h_{\mathbf{i}}. \quad (5)$$

An inequality of Lieb (equation 5.6 of [17]) tells us that

$$\text{OPT}(H(\mathbf{J})) \leq \text{OPT}_{\text{prod}}(\tilde{H}(\mathbf{J} + \mathbf{1})) \quad (6)$$

where $\mathbf{1}$ represents the all ones vector.²

2.3 The APX class and PTAS reductions

An NP optimisation (NPO) problem is a triple (I, S, m) where:

1. I is a set of instances.
2. Given an instance $x \in I$, the set $S(x)$ is the set of valid solutions to x .
3. Given an instance $x \in I$ and a valid solution $y \in S(x)$, the value achieved by y is $m(x, y) \in \mathbb{R}$.

The sets I and $S(x)$ are subsets of $\{0,1\}^*$ which can be recognised in polynomial time, and the function m is computable in polynomial time.

An NP optimisation problem can be either a maximisation problem or a minimisation problem. Given an instance x of a maximisation NPO problem, the aim is to find a valid solution y which maximises the value $m(x, y)$. An algorithm that, for any instance x , finds a y such that $m(x, y)$ is at least α times the optimal value, is said to have approximation ratio α .

An NP optimisation problem is in the class APX if there exists a constant α and a polynomial time algorithm T that achieves an approximation ratio α . The class PTAS is a subset of APX, and consists of all NPO problems for which there exists a polynomial time approximation scheme. That is, a problem is in PTAS if there exists an algorithm T such that for any instance x and approximation ratio α , $T(x, \alpha)$ returns a valid solution y that achieves the approximation ratio α , and T runs in time $q_r(|x|)$ for some polynomial q_r .

²The notation in [17] does not match our notation here. The Hamiltonian $H(\mathbf{J})$ is the main object of study in [17], and their equation 5.6 is expressed in terms of the ground state energy (lowest eigenvalue), which they denote E^Q , so the inequality appears in the reverse direction. Their notation E^C is for the lowest energy of the classical Hamiltonian which we may think of as the product state energy of $\tilde{H}(\mathbf{J})$

When proving that a problem is hard for the class APX, it is necessary to use PTAS reductions, which are a type of approximation-preserving reduction which preserves containment in the class PTAS. One reason for this is that we want APX-hardness to imply that a problem does not have a polynomial time approximation scheme (assuming that $\text{APX} \neq \text{PTAS}$).

Formally a PTAS reduction from problem A to problem B is a tuple of functions (f, g, c) such that, for any instance x of A and $\alpha < 1$:

1. $f(x, \alpha)$ is an instance of B computable in polynomial time in $|x|$.
2. for any valid solution y to $f(x, \alpha)$, then $g(x, y, \alpha)$ is a valid solution of x , computable in polynomial time in $|x|$ and $|y|$.
3. for any valid solution y to $f(x, \alpha)$, if y achieves an approximation ratio at least $c(\alpha)$ for the instance $f(x, \alpha)$, then $g(x, y, \alpha)$ achieves an approximation ratio at least α for the instance x .

Therefore a PTAS reduction provides a map between instances of A and B with related optimal values, but also map back such that ‘good’ solutions of the B instance are mapped back to ‘good’ solutions of the A instance. Note that a PTAS reduction from A to B provides an algorithm for A with approximation ratio α , if there is an algorithm for B which achieves an approximation ratio $c(\alpha)$. If c is not the identity function, then it may be that the approximation ratio $c(\alpha)$ required for B is a lot higher than the desired approximation ratio α for A.

Other (simpler) kinds of approximation-preserving reductions are often used when proving APX-completeness results, such as L-reductions or AP-reductions, see [22] for more. However the reduction we use in Lemma 1 constructs a different instance G' depending on a parameter η which is chosen to achieve particular approximation ratios α . That is, the function $f(x, \alpha)$ in the PTAS reduction really does depend on α , which is not allowed in the other types of approximation-preserving reductions.

3 Reduction from product state problem

In this section we prove Theorem 2, which relates the optimal product state value of a local Hamiltonian H to the optimal value of another Hamiltonian H' , constructed from the same interactions as H . First we describe how to construct H' from H .

Using the notation introduced in Section 2.2, we write $H = \sum_{\mathbf{i} \in I} h_{\mathbf{i}}$ for interactions $h_{\mathbf{i}}$ which act non trivially only on the qubits in $\mathbf{i} = (i_1, i_2, \dots, i_k)$, and can be expressed in the Pauli basis as in equation (3), so that

$$H = \sum_{\mathbf{i} \in I} \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} \sigma_{i_1}^{(a_1)} \sigma_{i_2}^{(a_2)} \dots \sigma_{i_n}^{(a_k)} \quad (7)$$

We construct H' by replacing each qubit of H with a cluster of T qubits for some parameter T which is used to control the quality of approximation. We label the nT qubits by (i, t) for some $i \in [n]$ and $t \in [T]$. Each term $h_{\mathbf{i}}$ in H is replaced with the same interaction applied to qubits $(i_1, t_1), (i_2, t_2), \dots, (i_k, t_k)$ for all combinations of $t_1, t_2, \dots, t_k$. The overall Hamiltonian H' is therefore:

$$H' = \sum_{\mathbf{t} \in [T]^k} \sum_{\mathbf{i} \in I} \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} \sigma_{(i_1, t_1)}^{(a_1)} \sigma_{(i_2, t_2)}^{(a_2)} \dots \sigma_{(i_k, t_k)}^{(a_k)}. \quad (8)$$

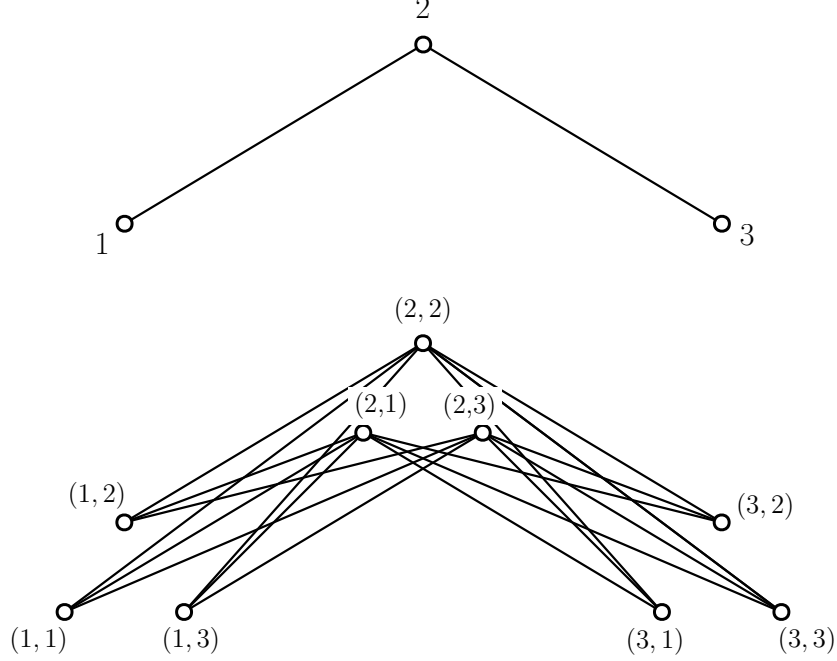


Figure 1: Example interaction graph of H' . At the top of the figure is a simple interaction graph of a 2-local Hamiltonian H on $n = 3$ qubits labelled 1, 2, 3. Beneath is the interaction graph of the new Hamiltonian H' . Each qubit of H has been replaced by a collection of $T = 3$ qubits in H' .

Theorem 2. Let $H = \sum_{\mathbf{i} \in I} h_{\mathbf{i}}$ be a k -local qubit Hamiltonian with $h_{\mathbf{i}} \geq 0$ for all $\mathbf{i} \in I$. Let H' be constructed from H as above. Then

$$T^k \text{OPT}_{\text{prod}}(H) \leq \text{OPT}(H') \leq (T + 2)^k \text{OPT}_{\text{prod}}(H)$$

Proof. First we prove the lower bound: let $\rho = \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_n$ be the product state that achieves $\text{Tr}(H\rho) = \text{OPT}_{\text{prod}}(H)$. Extend this to a product state ρ' on nT qubits where qubit (i, t) is in state ρ_i for all t . Then

$$\text{OPT}(H') \geq \text{Tr}(H'\rho') = \sum_{\mathbf{t} \in [T]^k} \text{Tr}(H\rho) = T^k \text{OPT}_{\text{prod}}(H).$$

It remains to prove the upper bound. Let $\hat{S}_i^{(a)} = \sum_{t \in T} \sigma_{(i,t)}^{(a)}$ and observe that

$$\begin{aligned} H' &= \sum_{\mathbf{i} \in I} \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} \sum_{\mathbf{t} \in [T]^k} \sigma_{(i_1, t_1)}^{(a_1)} \sigma_{(i_2, t_2)}^{(a_2)} \cdots \sigma_{(i_k, t_k)}^{(a_k)} \\ &= \sum_{\mathbf{i}} \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} \left(\sum_{t_1 \in [T]} \sigma_{(i_1, t_1)}^{(a_1)} \right) \left(\sum_{t_2 \in [T]} \sigma_{(i_2, t_2)}^{(a_2)} \right) \cdots \left(\sum_{t_k \in [T]} \sigma_{(i_k, t_k)}^{(a_k)} \right) \\ &= \sum_{\mathbf{i} \in I} \sum_{\mathbf{a} \in \{0,1,2,3\}^k} M(\mathbf{i})_{a_1, a_2, \dots, a_k} \hat{S}_{i_1}^{(a_1)} \hat{S}_{i_2}^{(a_2)} \cdots \hat{S}_{i_k}^{(a_k)}. \end{aligned}$$

The operator $\hat{S}_i^{(a)}$ is the tensor product representation $\hat{R}$ (of T copies of the spin $\frac{1}{2}$ representation) applied to $\sigma^{(a)}$, and it acts on the set of T qubits $\{(i, t)\}_{t \in [T]}$ which replaced the original qubit i in H . Therefore, as in equation (2) the tensor product representation $\hat{R}$ decomposes into a direct sum of irreducible representations:

$$\hat{S}^{(a)} = U \left(\bigoplus_{J \in \mathcal{T}} R_J(\sigma^{(a)}) \otimes I_{g_J} \right) U^\dagger$$

for some unitary U , and where the sum is over all $J \in \mathcal{T} = \{T/2, T/2-1, T/2-2, \dots\}$ (continuing down to 0 if T is even, or $1/2$ if T is odd).

Therefore

$$H' = U^{\otimes n} \left(\bigoplus_{\mathbf{J} \in \mathcal{T}^n} H(\mathbf{J}) \otimes I_{g_{\mathbf{J}}} \right) (U^\dagger)^{\otimes n}$$

where for each $\mathbf{J} = (J_1, J_2, \dots, J_n) \in \mathcal{T}^n$, $I_{g_{\mathbf{J}}}$ is the identity matrix on a space of dimension $g_{\mathbf{J}} = g_{J_1} \times g_{J_2} \cdots \times g_{J_n}$, and $H(\mathbf{J})$ is the variant of H where each qubit i is replaced by a spin J_i particle, as defined in equation (4).

So the optimal energy for H' is the maximum out of the optimal energies $\text{OPT}(H(\mathbf{J}))$ out of all possible values of $\mathbf{J} \in \mathcal{T}^n$

$$\text{OPT}(H') = \max_{\mathbf{J} \in \mathcal{T}^n} \text{OPT}(H(\mathbf{J})). \quad (9)$$

Applying Lieb's inequality (equation (6) above) to this gives:

$$\text{OPT}(H') \leq \max_{\mathbf{J} \in \mathcal{T}^n} \text{OPT}_{\text{prod}}(\tilde{H}(\mathbf{J} + \mathbf{1})) \quad (10)$$

where $\mathbf{1}$ is the all ones vector and $\tilde{H}(\mathbf{J})$ is as defined in equation (5).

Since $h_{\mathbf{i}} \geq 0$ for all $\mathbf{i}$, increasing the value of J_i for some i can only increase the value of $\text{OPT}_{\text{prod}}(\tilde{H}(\mathbf{J} + \mathbf{1}))$. Therefore the maximum in equation (10) is obtained when each J_i is maximised, i.e. at $\mathbf{J}^* = (T/2, T/2, \dots, T/2) \in \mathcal{T}^n$. For this $\mathbf{J}^*$, we have $\tilde{H}(\mathbf{J}^* + \mathbf{1}) = (T+2)^k H$ and so

$$\text{OPT}(H') \leq \text{OPT}_{\text{prod}}(\tilde{H}(\mathbf{J}^* + \mathbf{1})) = (T+2)^k \text{OPT}_{\text{prod}}(H).$$

□

4 Rank constrained Max-Cut

In this section we prove Theorem 3, that MAX-CUT_k is APX-complete, by demonstrating a PTAS reduction from MAX-CUT_k to MAX-CUT_{k+1} .

We first introduce some notation for the cost function which is maximised in MAX-CUT_k . Let $F_k^G : (S_{k-1})^n \rightarrow \mathbb{R}$ be defined by:

$$F_k^G(\vec{\mathbf{x}}) = \sum_{ij \in E} w_{ij} \frac{1}{2} (1 - \mathbf{x}_i \cdot \mathbf{x}_j)$$

so that $\text{MAX-CUT}_k(G) = \max_{\vec{\mathbf{x}}} F_k^G(\vec{\mathbf{x}})$.

In analogy to QUANTUM MAX-CUT, we sometimes refer to $F_k^G(\vec{\mathbf{x}})$ as the energy of $\vec{\mathbf{x}}$. For the remainder we now take $w_{ij} = 1$.

The main technical lemma which is used to prove Theorem 3 is Lemma 1:

Lemma 1. *Given an instance G of MAX-CUT_k and an error parameter η , we construct an instance G' of MAX-CUT_{k+1} which is only polynomially larger than G and prove that there exist constants c, C such that:*

1. $\text{MAX-CUT}_{k+1}(G') \geq C\eta m + \frac{3}{4}\text{MAX-CUT}_k(G)$
2. *there is a map $g : (S_k)^{n+6\eta m} \rightarrow (S_{k-1})^n$ such that for any $\vec{y} \in (S_k)^{n+6\eta m}$:*

$$F_{k+1}^{G'}(\vec{y}) \leq C\eta m + \frac{cm}{\eta} + \frac{3}{4}F_k^G(g(\vec{y})).$$

We first prove how Theorem 3 follows from Lemma 1, delaying the proof of Lemma 1 until Section 4.4.

Theorem 3. *There is a PTAS reduction from MAX-CUT_k to MAX-CUT_{k+1} . Therefore MAX-CUT_k is APX-complete for all (constant) $k \geq 1$.*

Proof. To prove that the construction of Lemma 1 is a PTAS reduction, we need to show that for any desired $\beta < 1$, we can choose η and α such that:

$$F_{k+1}^{G'}(\vec{y}) \geq \alpha \text{MAX-CUT}_{k+1}(G') \quad \Rightarrow \quad F_k^G(g(\vec{y})) \geq \beta \text{MAX-CUT}_k(G)$$

So suppose that $F_{k+1}^{G'}(\vec{y}) \geq \alpha \text{MAX-CUT}_{k+1}(G')$ for some α to be chosen later. Using both parts of Lemma 1, we have

$$\begin{aligned} C\eta m + \frac{cm}{\eta} + \frac{3}{4}F_k^G(g(\vec{y})) &\geq F_{k+1}^{G'}(\vec{y}) \\ &\geq \alpha \text{MAX-CUT}_{k+1}(G') \\ &\geq \alpha(C\eta m + \frac{3}{4}\text{MAX-CUT}_k(G)) \end{aligned}$$

Rearranging this gives

$$\begin{aligned} F_k^G(g(\vec{y})) &\geq \alpha \text{MAX-CUT}_k(G) + [(\alpha - 1)C\eta - c/\eta]4m/3 \\ &\geq \left[\alpha + \frac{8}{3}(\alpha - 1)C\eta - \frac{8c}{3\eta} \right] \text{MAX-CUT}_k(G) \end{aligned}$$

where we've used the fact that $\text{MAX-CUT}_k(G) \geq m/2$ and $[(\alpha - 1)C\eta - c/\eta] < 0$. Therefore it suffices to choose

$$\eta \geq \frac{16c}{3(1 - \beta)} \quad \text{and} \quad \alpha = \frac{\frac{1+\beta}{2} + \frac{8C}{3}}{1 + \frac{8C}{3}}$$

so that $\alpha + \frac{8}{3}(\alpha - 1)C\eta = \frac{1+\beta}{2}$ and $\frac{8c}{3\eta} \leq (1 - \beta)/2$. □

Before we prove Lemma 1, we describe the construction of G' from G and study the separate parts of G' , in order to build up enough understanding to complete the proof of Lemma 1 in Section 4.4.

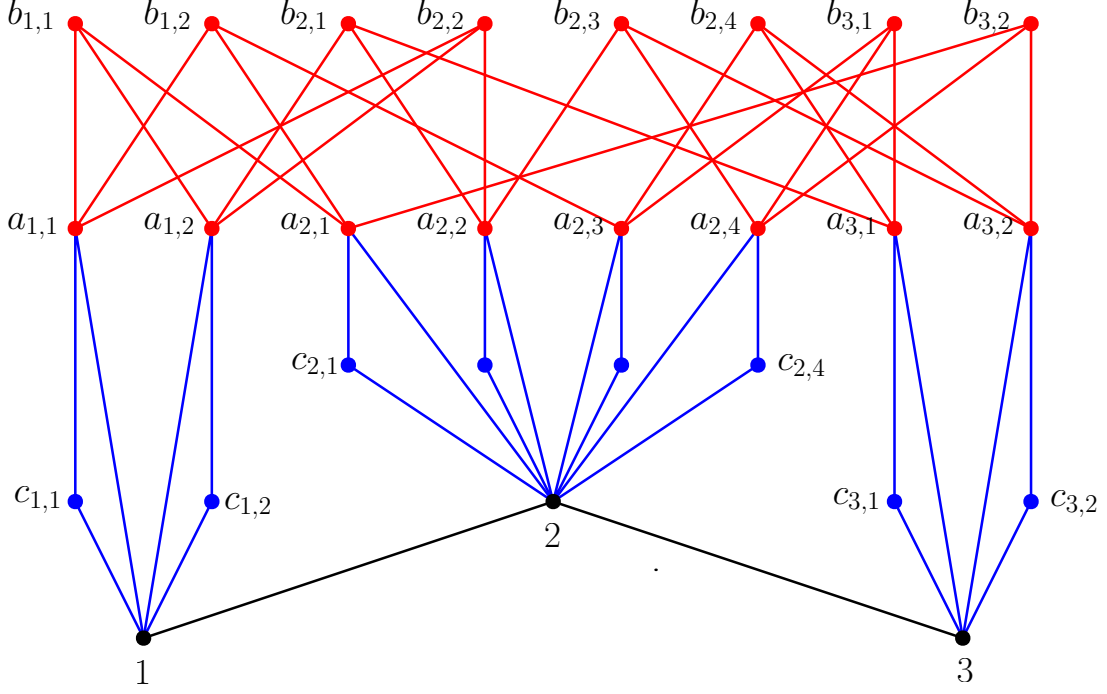


Figure 2: Example of the construction of G' described in Section 4.1. Here, the original graph G is a path graph on three vertices $\{1, 2, 3\}$ and the parameter $\eta = 2$. The original edges and vertices of G are coloured black; the edges in the triangle gadgets are coloured blue, and the edges in the d -regular bipartite graph are coloured red.

4.1 Construction

Given a graph $G = (V, E)$ with $n = |V|$ vertices and $m = |E|$ edges, we extend G to a graph G' on vertex set $V \cup V'$ where $|V'| = 6\eta m$ for some parameter η which controls the quality of approximation. For each vertex $i \in V$ of degree d_i , we add $3\eta d_i$ vertices $a_{i,\alpha}, b_{i,\alpha}, c_{i,\alpha}$ for $\alpha \in [\eta d_i]$. There are three types of edges in E' :

1. Edges in E from the original graph G between the vertices in V .
2. There is a bipartite expander graph between the ‘ a ’ vertices in $A = \{a_{i,\alpha}\}$ and the ‘ b ’ vertices $B = \{b_{i,\alpha}\}$. This is chosen to be a d -regular expander graph for constant d , with constant spectral gap in the adjacency matrix.
3. For each i, α , there is a triangle of edges between vertices $i, a_{i,\alpha}, c_{i,\alpha}$.

4.2 Triangle gadget

First we consider a triangle gadget, consisting of three vertices $0, 1, 2$ with edges between all pairs.

Lemma 2. *Let $G_\Delta = (V, E)$ where $V = \{0, 1, 2\}$ and $E = \{01, 12, 02\}$. Let $\vec{\mathbf{x}} = (\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2)$ be an assignment of unit vectors to the vertices, and suppose $\mathbf{x}_0 \cdot \mathbf{x}_1 = \cos \theta$ for some angle $\theta \in [0, \pi)$. Then*

$$\max_{\mathbf{x}_2} F_k^{G_\Delta}(\vec{\mathbf{x}}) = \frac{9}{4} - \frac{1}{4}(2 \cos \frac{\theta}{2} - 1)^2 \leq \frac{9}{4} - \frac{9}{16\pi^2}(\theta - \frac{2\pi}{3})^2$$

Proof.

$$\begin{aligned} F_k^{G\Delta}(\vec{\mathbf{x}}) &= \frac{1}{2} \sum_{uv \in E} 1 - \mathbf{x}_u \cdot \mathbf{x}_v \\ &= \frac{1}{2} (3 - \mathbf{x}_2 \cdot (\mathbf{x}_0 + \mathbf{x}_1) - \mathbf{x}_0 \cdot \mathbf{x}_1) \end{aligned}$$

This expression is maximised by choosing $\mathbf{x}_2$ in the $-(\mathbf{x}_0 + \mathbf{x}_1)$ direction. i.e. by setting $\mathbf{x}_2 = -\frac{\mathbf{x}_0 + \mathbf{x}_1}{\|\mathbf{x}_0 + \mathbf{x}_1\|}$. Substituting this in above gives

$$\max_{\mathbf{x}_2} F_k^{G\Delta}(\vec{\mathbf{x}}) = \frac{1}{2} (3 + \|\mathbf{x}_0 + \mathbf{x}_1\| - \mathbf{x}_0 \cdot \mathbf{x}_1)$$

Note that the above substitution is only possible if $\mathbf{x}_0 + \mathbf{x}_1 \neq 0$. If $\mathbf{x}_0 + \mathbf{x}_1 = 0$, then $F_k^{G\Delta}(\vec{\mathbf{x}})$ is independent of $\mathbf{x}_2$, and the above expression for $\max_{\mathbf{x}_2} F_k^{G\Delta}(\vec{\mathbf{x}})$ still holds.

Now use the fact that $\|\mathbf{x}_0 + \mathbf{x}_1\|^2 = (\mathbf{x}_0 + \mathbf{x}_1) \cdot (\mathbf{x}_0 + \mathbf{x}_1) = 2 + 2\mathbf{x}_0 \cdot \mathbf{x}_1$ to get

$$\max_{\mathbf{x}_2} F_k^{G\Delta}(\vec{\mathbf{x}}) = \frac{1}{2} (3 + \sqrt{2 + 2\mathbf{x}_0 \cdot \mathbf{x}_1} - \mathbf{x}_0 \cdot \mathbf{x}_1).$$

Substitute in $\mathbf{x}_0 \cdot \mathbf{x}_1 = \cos \theta$ and rearrange to show

$$\max_{\mathbf{x}_2} F_k^{G\Delta}(\vec{\mathbf{x}}) = \frac{1}{2} (3 + \sqrt{2 + 2\cos \theta} - \cos \theta) = \frac{9}{4} - \frac{1}{4} (\sqrt{2 + 2\cos \theta} - 1)^2$$

Finally, substitute in the half-angle formula $\sqrt{2 + 2\cos \theta} = 2\cos \frac{\theta}{2}$ to obtain the equality stated in the lemma.

To prove the inequality, it remains to show that $(2\cos \frac{\theta}{2} - 1)^2 \geq (1 - \frac{3\theta}{2\pi})^2$ for all $\theta \in [0, \pi)$.

Note that $2\cos \frac{\theta}{2} - 1$ and $1 - \frac{3\theta}{2\pi}$ are equal at $\theta = 0$ and $\theta = 2\pi/3$. Since $\frac{d^2}{d\theta^2}(2\cos \frac{\theta}{2} - 1) = -\frac{1}{2}\cos \frac{\theta}{2} \leq 0$ for $\theta \in [0, \pi)$ and $1 - \frac{3\theta}{2\pi}$ is linear, it must be the case that

$$2\cos \frac{\theta}{2} - 1 \geq 1 - \frac{3\theta}{2\pi} \geq 0 \quad \text{for } \theta \in [0, 2\pi/3]$$

$$2\cos \frac{\theta}{2} - 1 \leq 1 - \frac{3\theta}{2\pi} \leq 0 \quad \text{for } \theta \in [2\pi/3, \pi]$$

and therefore $(2\cos \frac{\theta}{2} - 1)^2 \geq (1 - \frac{3\theta}{2\pi})^2$ for all $\theta \in [0, \pi)$ as required. $\square$

4.3 Bipartite graphs

Lemma 3. *Let G be a d -regular connected, bipartite graph with edges across the bipartition $A \cup B = V$. Let $\lambda_{\max}$ be the largest eigenvalue of the Laplacian L of G , and suppose the gap between the largest and second largest eigenvalue is at least Δ .*

Let $\vec{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$ be an assignment of k -dimensional vectors to the n vertices of G . Let $\mathbf{z} = (\sum_{i \in A} \mathbf{x}_i - \sum_{i \in B} \mathbf{x}_i)/n$, and for each $i \in A$ let ϵ_i be the angle between $\mathbf{z}$ and $\mathbf{x}_i$. Then

$$F_k^G(\vec{\mathbf{x}}) \leq \frac{\lambda_{\max} n}{4} - \frac{\Delta}{4\pi^2} \sum_{i \in A} \epsilon_i^2$$

In the construction for Lemma 1, Lemma 3 will be applied to a d -regular bipartite expander graph, for d constant and with constant spectral gap. Since the spectrum of the adjacency matrix of a bipartite graph is symmetric about 0, this implies that the largest eigenvalue $\lambda_{\max}$ of the Laplacian L is $2d = O(1)$ and the gap Δ to the second largest eigenvalue is also constant.

Proof. The total energy is

$$F_k^G(\vec{\mathbf{x}}) = \sum_{ij \in E} \frac{1}{2}(1 - \mathbf{x}_i \cdot \mathbf{x}_j) = \sum_{i,j \in V} \frac{1}{4} L_{ij} \mathbf{x}_i \cdot \mathbf{x}_j.$$

Let the coordinates of $\mathbf{x}_i$ be x_{il} for $l \in [k]$, so that $\mathbf{x}_i$ can be identified with the i th row of the matrix with entries $\{x_{il}\}_{i \in [n], l \in [k]}$. Define the vector $\hat{\mathbf{x}}_l$ to be the l th column of this matrix, i.e. the n -dimensional vector with coordinates x_{il} for $i \in [n]$

$$\sum_{i,j \in V} \frac{1}{4} L_{ij} \mathbf{x}_i \cdot \mathbf{x}_j = \sum_{l=1}^k \sum_{i,j \in V} \frac{1}{4} x_{il} L_{ij} x_{jl} = \sum_{l=1}^k \frac{1}{4} \hat{\mathbf{x}}_l^\top L \hat{\mathbf{x}}_l.$$

If G is connected then the all ones vector is the unique smallest eigenvector of L , with eigenvalue 0. Since G bipartite across the split $V = A \cup B$, the spectrum of the adjacency matrix $\text{Adj}(G)$ is symmetric about 0. To see this note that the diagonal matrix with entry at i, i on the diagonal equal to $+1$ for $i \in A$, and -1 for $i \in B$, maps eigenvectors of $\text{Adj}(G)$ of eigenvalue λ to eigenvectors of eigenvalue $-\lambda$. Since G is d -regular, the spectrum of $L = dI - \text{Adj}(G)$ is therefore symmetric about d and L has a unique largest eigenvalue $\lambda_{\max} = 2d$ with corresponding eigenvector $\mathbf{e}^*$:

$$e_i^* = \begin{cases} 1 & \text{for } i \in A \\ -1 & \text{for } i \in B. \end{cases}$$

Decompose $\hat{\mathbf{x}}_l$ in terms of $\mathbf{e}^*$ and some $\hat{\mathbf{e}}_l$ orthogonal to $\mathbf{e}^*$

$$\hat{\mathbf{x}}_l = z_l \mathbf{e}^* + \hat{\mathbf{e}}_l. \tag{11}$$

The vector $\mathbf{e}^*$ is the unique eigenvector of L corresponding to the largest eigenvalue $\lambda_{\max}$ and so $\mathbf{e}^{*\top} L \mathbf{e}^* = \lambda_{\max} \|\mathbf{e}^*\|^2$. Since $\hat{\mathbf{e}}_l$ is orthogonal to $\mathbf{e}^*$, it has overlap with no eigenvector of eigenvalue larger than the second largest eigenvalue $\lambda_{\max} - \Delta$, so $\hat{\mathbf{e}}_l^\top L \hat{\mathbf{e}}_l \leq (\lambda_{\max} - \Delta) \|\hat{\mathbf{e}}_l\|^2$. Therefore

$$\begin{aligned} F_k^G(\vec{\mathbf{x}}) &= \sum_{l=1}^k \frac{1}{4} \hat{\mathbf{x}}_l^\top L \hat{\mathbf{x}}_l = \frac{1}{4} \sum_{l=1}^k \left(z_l^2 \mathbf{e}^{*\top} L \mathbf{e}^* + \hat{\mathbf{e}}_l^\top L \hat{\mathbf{e}}_l \right) \\ &\leq \frac{1}{4} \sum_{l=1}^k \left(z_l^2 \lambda_{\max} \|\mathbf{e}^*\|^2 + (\lambda_{\max} - \Delta) \|\hat{\mathbf{e}}_l\|^2 \right) \\ &= \frac{1}{4} \left(\lambda_{\max} \left(\sum_{l=1}^k \|\hat{\mathbf{x}}_l\|^2 \right) - \Delta \left(\sum_{l=1}^k \|\hat{\mathbf{e}}_l\|^2 \right) \right) \end{aligned}$$

where in the final equality we have used the fact that $\mathbf{e}^*$ and $\hat{\mathbf{e}}_l$ are orthogonal to equate $\|\hat{\mathbf{x}}_l\|^2 = z_l^2 \|\mathbf{e}^*\|^2 + \|\hat{\mathbf{e}}_l\|^2$. Since each $\mathbf{x}_i$ has unit size, we have:

$$\sum_{l=1}^k \|\hat{\mathbf{x}}_l\|^2 = \sum_{i=1}^n \sum_{l=1}^k x_{il}^2 = \sum_{i=1}^n \|\mathbf{x}_i\|^2 = n.$$

Similarly let the coordinates of $\hat{\mathbf{e}}_l$ be e_{il} for $i \in [n]$, so that $\hat{\mathbf{e}}_l$ can be identified with the l th column of the matrix with entries $\{e_{il}\}_{i \in [n], l \in [k]}$. Define the vector $\mathbf{e}_i$ to be the i th row of this matrix, i.e. the k -dimensional vector with coordinates x_{il} for $l \in [k]$.

Observe that $\sum_{l=1}^k \|\hat{\mathbf{e}}_l\|^2 = \sum_{l=1}^k \sum_{i=1}^n e_{il}^2 = \sum_{i=1}^n \|\mathbf{e}_i\|^2$, so we have

$$F_k^G(\vec{\mathbf{x}}) \leq \frac{1}{4} \left(\lambda_{\max} n - \Delta \sum_{i=1}^n \|\mathbf{e}_i\|^2 \right).$$

Noting that $\|\mathbf{e}_i\|^2 \geq 0$ for all $i \in B$, all that remains is to show that $\|\mathbf{e}_i\|^2 \geq \epsilon_i^2/\pi^2$ for all $i \in A$. We actually prove the slightly stronger bound of

$$\|\mathbf{e}_i\| \geq \begin{cases} \sin(\epsilon_i) & \text{if } \epsilon_i \in [0, \pi/2] \\ 1 & \text{if } \epsilon_i \in [\pi/2, \pi]. \end{cases}$$

Since $\mathbf{e}^*$ and $\hat{\mathbf{e}}_1$ are orthogonal, we can take the inner product of equation (11) with $\mathbf{e}^*$ to get:

$$z_l \|\mathbf{e}^*\|^2 = \hat{\mathbf{x}}_1 \cdot \mathbf{e}^* = \sum_{i \in A} x_{il} - \sum_{i \in B} x_{il}.$$

So $\mathbf{z} = (\sum_{i \in A} \mathbf{x}_i - \sum_{i \in B} \mathbf{x}_i)/n$ is the k -dimensional vector with coordinates z_l . Looking at the i th coordinate of $\hat{\mathbf{x}}_1$ in equation (11) for $i \in A$, we have:

$$x_{il} = z_l + e_{il} \quad \Rightarrow \quad \mathbf{x}_i = \mathbf{z} + \mathbf{e}_i.$$

The angle ϵ_i between $\mathbf{z}$ and $\mathbf{x}_i$ satisfies $\mathbf{z} \cdot \mathbf{x}_i = \|\mathbf{z}\| \|\mathbf{x}_i\| \cos(\epsilon_i) = \|\mathbf{z}\| \cos(\epsilon_i)$, and so

$$\|\mathbf{e}_i\|^2 = (\mathbf{x}_i - \mathbf{z}) \cdot (\mathbf{x}_i - \mathbf{z}) \tag{12}$$

$$= \mathbf{x}_i \cdot \mathbf{x}_i - 2\mathbf{x}_i \cdot \mathbf{z} + \mathbf{z} \cdot \mathbf{z} \tag{13}$$

$$= 1 - 2\|\mathbf{z}\| \cos(\epsilon_i) + \|\mathbf{z}\|^2. \tag{14}$$

If $\epsilon_i \in [\pi/2, \pi]$, then $\cos(\epsilon_i) \leq 0$, and so $\|\mathbf{e}_i\|^2 \geq 1$. Otherwise, if $\epsilon_i \in [0, \pi/2]$, we can complete the square in equation (14) to get

$$\|\mathbf{e}_i\|^2 = 1 - \cos^2(\epsilon_i) + (\|\mathbf{z}\| - \cos(\epsilon_i))^2 \geq \sin^2(\epsilon_i).$$

□

4.4 Proof of Lemma 1

We are now ready to prove Lemma 1.

Proof. For part 1. of Lemma 1, let $\vec{\mathbf{x}}$ be the assignment that achieves $F_k^G(\vec{\mathbf{x}}) = \text{MAX-CUT}_k(G)$. Embed each vector $\mathbf{x}_i$ into $\mathbb{R}^{k+1}$ by padding with an extra 0 and call the resulting vector $\mathbf{x}'_i \in \mathbb{R}^{k+1}$. Let $\mathbf{z} = (0, 0, \dots, 0, 1) \in \mathbb{R}^{k+1}$ be the new direction.

Define $\vec{\mathbf{y}} \in (S_{k+1})^{V'}$ by:

$$\begin{aligned}\mathbf{y}_i &= \cos(2\pi/3)\mathbf{z} + \sin(2\pi/3)\mathbf{x}'_i \\ \mathbf{y}_{c_{i,\alpha}} &= \cos(4\pi/3)\mathbf{z} + \sin(4\pi/3)\mathbf{x}'_i \\ \mathbf{y}_{a_{i,\alpha}} &= \mathbf{z} \quad \text{and} \quad \mathbf{y}_{b_{i,\alpha}} = -\mathbf{z}.\end{aligned}$$

This assignment achieves the maximum possible $9/4$ for each triangle $i, a_{i,\alpha}, c_{i,\alpha}$. The expander graph includes $4\eta m$ vertices in total, and so $\vec{\mathbf{y}}$ achieves the maximum of $\lambda_{\max}\eta m$ for the expander graph. For the edges in G' corresponding to the original edges of G :

$$\frac{1}{2}[1 - \mathbf{y}_i \cdot \mathbf{y}_j] = \frac{1}{2}[1 - \cos^2(2\pi/3) - \sin^2(2\pi/3)\mathbf{x}_i \cdot \mathbf{x}_j] = \frac{3}{4} \times \frac{1}{2}[1 - \mathbf{x}_i \cdot \mathbf{x}_j].$$

So in total:

$$F_{k+1}^{G'}(\vec{\mathbf{y}}) = \frac{9}{4}2\eta m + \lambda_{\max}\eta m + \frac{3}{4}F_k^G(\vec{\mathbf{x}}).$$

Noting that $\text{MAX-CUT}_{k+1}(G') \geq F_{k+1}^{G'}(\vec{\mathbf{y}})$ and $F_k^G(\vec{\mathbf{x}}) = \text{MAX-CUT}_k(G)$, we have

$$\text{MAX-CUT}_{k+1}(G') \geq C\eta m + \frac{3}{4}\text{MAX-CUT}_k(G)$$

where $C = 9/2 + \lambda_{\max}$.

The remainder of the proof focuses on part 2. of Lemma 1. Note that we reuse the variable names $\mathbf{x}, \mathbf{y}, \mathbf{z}$, although they are not the same as in part 1. We start by describing the map $g : (S_k)^{n+6\eta m} \rightarrow (S_{k-1})^n$ which maps inputs for $F_{k+1}^{G'}$ to inputs for F_k^G .

Let $\vec{\mathbf{y}} \in (S_k)^{n+6\eta m}$ and set $\mathbf{z} = \sum_{a \in A} \mathbf{y}_a - \sum_{b \in B} \mathbf{y}_b / \|\sum_{a \in A} \mathbf{y}_a - \sum_{b \in B} \mathbf{y}_b\|$. Now change basis so that $\mathbf{z} = (0, 0, \dots, 0, 1) \in \mathbb{R}^{k+1}$. For each $i \in V$, let $\mathbf{x}_i$ be the unit vector you get by taking the first k entries of $\mathbf{y}_i$ in this basis and renormalising. Let $g(\vec{\mathbf{y}}) = \vec{\mathbf{x}} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)$.

We write $\mathbf{x}'$ for the embedding of $\mathbf{x}$ into $\mathbb{R}^{k+1}$ by padding with an extra zero; or equivalently $\mathbf{x}'_i$ is the (renormalised) projection of $\mathbf{y}_i$ onto the space orthogonal to $\mathbf{z}$. Since $\mathbf{y}_i, \mathbf{z}$ and $\mathbf{x}'_i$ are all unit vectors for all $i \in V$, we have:

$$\mathbf{y}_i = \cos(\theta_i)\mathbf{z} + \sin(\theta_i)\mathbf{x}'_i$$

for some angle θ_i between $\mathbf{z}$ and $\mathbf{y}_i$.

We can now bound the contribution from different parts of G' . First consider the expander graph across A and B . Let $\epsilon_{i,\alpha}$ be the angle between $\mathbf{y}_{a_{i,\alpha}}$ and $\mathbf{z}$. By Lemma 3, the contribution to $F_{k+1}^{G'}(\vec{\mathbf{y}})$ from the bipartite graph is at most

$$\lambda_{\max}\eta m - \frac{\Delta}{4\pi^2} \sum_{i,\alpha} \epsilon_{i,\alpha}^2.$$

Second, consider the triangle gadgets. The angle between $\mathbf{y}_i$ and $\mathbf{y}_{a_{i,\alpha}}$ is in $[\theta_i - \epsilon_{i,\alpha}, \theta_i + \epsilon_{i,\alpha}]$. Let $\phi_i = \frac{2\pi}{3} - \theta_i$ so that by Lemma 2, the contribution from the triangle $i, a_{i,\alpha}, c_{i,\alpha}$ is at most

$$\frac{9}{4} - \frac{9}{16\pi^2}(\phi_i + \epsilon_{i,\alpha})^2.$$

Third, consider the original edges from G between the vertices in V . For an edge $ij \in E$, if $\theta_i = \theta_j = 2\pi/3$ (i.e. $\phi_i = 0 = \phi_j$), then the contribution of that edge to $F_{k+1}^{G'}(\vec{\mathbf{y}})$ is

$$\frac{3}{8}(1 - \mathbf{x}_i \cdot \mathbf{x}_j).$$

If ϕ_i and ϕ_j are non-zero, this can change the angle ξ_i between $\mathbf{y}_i$ and $\mathbf{y}_j$ by at most $|\phi_i| + |\phi_j|$. Since $\mathbf{y}_i \cdot \mathbf{y}_j = \cos(\xi_i)$, the rate of change with respect to ξ_i is at most $|\frac{d}{d\xi_i} \cos(\xi_i)| = |\sin(\xi_i)| \leq 1$. Therefore the edge ij can contribute at most

$$\frac{3}{8}(1 - \mathbf{x}_i \cdot \mathbf{x}_j) + |\phi_i| + |\phi_j|.$$

Adding all these contributions together gives:

$$\begin{aligned} F_{k+1}^{G'}(\vec{\mathbf{y}}) &\leq \lambda_{\max} \eta m - \frac{\Delta}{4\pi^2} \sum_{i,\alpha} \epsilon_{i,\alpha}^2 + \sum_{i,\alpha} \left(\frac{9}{4} - \frac{9}{16\pi^2} (\phi_i + \epsilon_{i,\alpha})^2 \right) \\ &\quad + \sum_{ij \in E} \left(\frac{3}{8}(1 - \mathbf{x}_i \cdot \mathbf{x}_j) + |\phi_i| + |\phi_j| \right) \\ &= C\eta m + \frac{3}{4} F_k^G(\vec{\mathbf{x}}) - \sum_{i,\alpha} (c_1 \epsilon_{i,\alpha}^2 + c_2 (\phi_i + \epsilon_{i,\alpha})^2) + \sum_{i \in V} d_i |\phi_i| \end{aligned}$$

where $C = 9/2 + \lambda_{\max}$, $c_1 = \frac{\Delta}{4\pi^2}$ and $c_2 = \frac{9}{16\pi^2}$. If the bipartite expander graph is chosen to be low degree and with good spectral expansion such that $\lambda_{\max}$ and Δ are constant, then C, c_1, c_2 are all constant.

Looking at the terms with ϵ_{i,α_i} and rearranging we have

$$c_1 \epsilon_{i,\alpha}^2 + c_2 (\phi_i + \epsilon_{i,\alpha})^2 = (c_1 + c_2) \left(\epsilon_{i,\alpha} + \frac{c_2}{c_1 + c_2} \phi_i \right)^2 + \frac{c_1 c_2}{c_1 + c_2} \phi_i^2 \geq c' \phi_i^2$$

where $c' = \frac{c_1 c_2}{c_1 + c_2}$. Substituting this back in and summing over α , we get:

$$F_{k+1}^{G'}(\vec{\mathbf{y}}) \leq C\eta m + \frac{3}{4} F_k^G(\vec{\mathbf{x}}) + \sum_{i \in V} d_i (-\eta c' \phi_i^2 + |\phi_i|).$$

To upper bound $-\eta c' \phi_i^2 + |\phi_i|$, note that it is symmetric about zero, and for $\phi_i \geq 0$, one can complete the square $-\eta c' \phi_i^2 + \phi_i = -\eta c' (\phi_i - \frac{1}{2\eta c'})^2 + \frac{1}{4\eta c'} \leq \frac{1}{4\eta c'}$. Putting this back in leads to

$$F_{k+1}^{G'}(\vec{\mathbf{y}}) \leq C\eta m + \frac{3}{4} F_k^G(\vec{\mathbf{x}}) + \frac{1}{2c'\eta} m.$$

□

References

- [1] Toby Cubitt and Ashley Montanaro. Complexity classification of local hamiltonian problems. *SIAM Journal on Computing*, 45(2):268–316, 2016.

- [2] Toby S. Cubitt, Ashley Montanaro, and Stephen Piddock. Universal quantum Hamiltonians. *Proceedings of the National Academy of Sciences*, 115(38):9497–9502, aug 2018.
- [3] Stephen Piddock and Ashley Montanaro. The complexity of antiferromagnetic interactions and 2d lattices. *Quantum Information & Computation*, 17(7-8):636–672, 2017.
- [4] Sevag Gharibian and Ojas Parekh. Almost optimal classical approximation algorithms for a quantum generalization of max-cut. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019.
- [5] Ojas Parekh and Kevin Thompson. An optimal product-state approximation for 2-local quantum hamiltonians with positive terms. *arXiv preprint arXiv:2206.08342*, 2022.
- [6] Eunou Lee. Optimizing quantum circuit parameters via SDP, 2022.
- [7] Robbie King. An Improved Approximation Algorithm for Quantum Max-Cut, 2022.
- [8] Eunou Lee and Ojas Parekh. An improved quantum max cut approximation via maximum matching. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, pages 105–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.
- [9] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.
- [10] Jop Briët, Fernando Mário de Oliveira Filho, and Frank Vallentin. The positive semidefinite grothendieck problem with rank constraint. In *International Colloquium on Automata, Languages, and Programming*, pages 31–42. Springer, 2010.
- [11] Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O’Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable CSPs? *SIAM Journal on Computing*, 37(1):319–357, 2007.
- [12] Christos Papadimitriou and Mihalis Yannakakis. Optimization, approximation, and complexity classes. In *Proceedings of the twentieth annual ACM symposium on Theory of computing*, pages 229–234, 1988.
- [13] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001.
- [14] Yeongwoo Hwang, Joe Neeman, Ojas Parekh, Kevin Thompson, and John Wright. Unique Games hardness of Quantum Max-Cut, and a conjectured vector-valued Borell’s inequality, 2022.
- [15] Sevag Gharibian. Guest column: The 7 faces of quantum np. *ACM SIGACT News*, 54(4):54–91, December 2023.
- [16] Fernando G.S.L. Brandao and Aram W. Harrow. Product-state approximations to quantum ground states. In *Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing, STOC ’13*, page 871–880, New York, NY, USA, 2013. Association for Computing Machinery.

- [17] Elliott H Lieb. The classical limit of quantum spin systems. *Communications in Mathematical Physics*, 31:327–340, 1973.
- [18] John Kallaugher, Ojas Parekh, Kevin Thompson, Yipu Wang, and Justin Yirka. Complexity classification of product state problems for local hamiltonians, 2024.
- [19] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.
- [20] László Lovász. *Graphs and geometry*, volume 65. American Mathematical Soc., 2019.
- [21] John Wright. Personal communication.
- [22] P. Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of Computational Complexity. Twelfth Annual IEEE Conference*, pages 262–273, 1997.