Epsilon-net method for optimizations over separable states.

Yaoyun Shi\textsuperscript{a}, Xiaodi Wu\textsuperscript{b,*}

\textsuperscript{a}Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105, USA.
\textsuperscript{b}Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA.

Abstract

We give algorithms for the optimization problem: \( \max_\rho \langle Q, \rho \rangle \), where \( Q \) is a Hermitian matrix, and the variable \( \rho \) is a bipartite separable quantum state. This problem lies at the heart of several problems in quantum computation and information, such as the complexity of QMA(2). While the problem is NP-hard, our algorithms are better than brute force for several instances of interest. In particular, they give PSPACE upper bounds on promise problems admitting a QMA(2) protocol in which the verifier performs only a logarithmic number of elementary gates on both proofs, as well as the promise problem of deciding if a bipartite local Hamiltonian has a large or small ground energy. For \( Q \geq 0 \), our algorithm runs in time exponential in \( \|Q\|_F \). While the existence of such an algorithm was first proved recently by Brandão, Christandl and Yard [Proceedings of the 43rd annual ACM Symposium on Theory of Computation, 343–352, 2011], our algorithm is conceptually simpler.

Keywords: Epsilon-net, separable states, QMA(2), PSPACE, local Hamiltonian problem

1. Introduction

Entanglement is an essential ingredient in many ingenious applications of quantum information processing. Understanding and exploiting entanglement remains a central theme in quantum information processing research [23]. Denote by SepD(\( A_1 \otimes A_2 \)) the set of separable density operators over the space \( A_1 \otimes A_2 \). The weak membership problem for separability that is to decide, given a classical description of \( \rho \in \text{SepD}(A_1 \otimes A_2) \), whether the state \( \rho \) is inside or \( \epsilon \) far away in trace distance from SepD(\( A_1 \otimes A_2 \)), turns out to be NP-hard when \( \epsilon \) is inverse exponential \((21)\) (or even inverse polynomial \( [26, 17] \)) in the dimension of \( A_1 \otimes A_2 \).

In this paper we study a closely related problem, namely the linear optimization problem over separable states below where \( \langle A, B \rangle \) denotes the Hilbert-Schmidt inner product of \( A \) and \( B \).

**Problem 1.** Given a Hermitian matrix \( Q \) over \( A_1 \otimes A_2 \) (of dimension \( d \times d \)), compute the optimum value, denoted by OptSep(\( Q \)), of the optimization problem

\[
\max \langle Q, X \rangle \text{ subject to } X \in \text{SepD}(A_1 \otimes A_2).
\]

It is a standard fact in convex optimization [19] Figure 4.1] that the weak membership problem and the weak linear optimization, a special case of Problem [1] over certain convex set, such as

---

\*Corresponding author. This work was completed when the author was a student at Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48105, USA.

Email addresses: shiyy@eecs.umich.edu (Yaoyun Shi), xiaodiwu@mit.edu (Xiaodi Wu)
SepD \((A_1 \otimes A_2)\), are equivalent up to a polynomial loss in precision and a polynomial-time overhead. Thus it is NP-hard to compute OptSep\((Q)\) with an inverse polynomial additive error. Besides the connection mentioned above, Problem 1 can also be understood from various aspects. Firstly, Problem 1 can be viewed as finding the minimum energy of some physical system that is achieved by separable states. Secondly, in the study of the tensor product space \([13]\), the value OptSep\((Q)\) is precisely the injective norm of \(Q\) in \(\mathcal{L}(A_1) \otimes \mathcal{L}(A_2)\), where \(\mathcal{L}(A)\) denotes the Banach space of operators on \(A\) with the operator norm. Finally, one may be equally motivated from the study in operations research (e.g., “Bi-Quadratic Optimization over Unit Spheres” \([32]\)).

Another motivation to study Problem 1 is due to the recent interest about the complexity class QMA(2). Originally the class QMA was defined \([30]\) as the quantum counterpart of the complexity class NP. While the extension of NP to allow multiple provers trivially reduces to NP itself, the power of QMA(2), the extension of QMA with multiple unentangled provers, remains far from being well understood. The study of the multiple-prover model was initiated in \([29]\), where QMA(k) denotes the complexity class for the k-prover case. Much attention was attracted to this model because of the surprising discovery that NP admits logarithmic-size unentangled quantum proofs \([6]\), comparing with the fact that single prover quantum logarithm-size proofs only characterize BQP \([33]\). It seems adding one unentangled prover increases the power of the model substantially. There are several subsequent works on refining the initial protocol either with improved completeness and soundness \([4, 11, 20]\), or with less powerful verifiers \([10]\). Recently it was proved that QMA(2)=QMA(poly) \([24]\) by using the so-called product test protocol that determines whether a multipartite state is a product state when two copies of it are given. Also, variants of QMA(2), such as BellQMA and LOCCQMA with restricted verifiers that perform only nonadaptive or adaptive local measurements respectively, were defined in \([1]\) and studied in \([7, 8]\).

Despite much effort, any nontrivial upper bound of QMA(2) remains elusive. The best known upper bound QMA(2) \(\subseteq\) NEXP follows trivially by nondeterministically guessing the two proofs. It would be surprising if QMA(2) = NEXP. Thus it is reasonable to seek a better upper bound like EXP or even PSPACE. It is not hard to see that simulating QMA(2) amounts to distinguishing between two promises of OptSep\((Q)\), although one is free to choose the appropriate \(Q\).

**Our contributions.** In this paper we provide efficient algorithms for Problem 1 in either time or space for several \(Qs\) of interest. Our idea is to enumerate via epsilon-nets more “cleverly” with the help of certain structures of \(Q\).

Now we briefly describe our strategy of obtaining space-efficient algorithms. When the total number of points to enumerate is not large, one can represent, and hence enumerate each point in polynomial space. If the additional computation for each point can also be done in polynomial space, one immediately gets a polynomial-space implementation for the whole algorithm by composing those two components naturally. We make use of the relation NC(poly)=PSPACE \([5]\) to obtain space-efficient implementation for the additional computation, which in our case basically includes the following two parts. The first part assures that the enumeration procedure functions correctly because these epsilon-nets of interest are not standard. This part turns out to be a simple application of the so-called multiplicative matrix weight update (MMW) method \([2, 39, 28]\) to computing a min-max form. The second part only contains fundamental matrix operations, which usually admit efficient parallel algorithms \([16]\). As a result, both parts of the additional computation admit efficient parallel algorithms, and therefore can be implemented in polynomial space.

We summarize below the main results obtained by applying the above ideas.

1. The first property exploited is the so-called decomposability of \(Q\) which refers to whether \(Q\)
can be decomposed in the form $Q = \sum_{i=1}^{M} Q_i^1 \otimes Q_i^2$ with small $M$. Intuitively, if one substitutes this $Q$’s decomposition into $\langle Q, \rho_1 \otimes \rho_2 \rangle$ and treat $\langle Q_1^1, \rho_1 \rangle, \ldots, \langle Q_{M_1}^1, \rho_1 \rangle, \langle Q_2^1, \rho_2 \rangle, \ldots, \langle Q_{M_2}^2, \rho_2 \rangle$ as variables, the optimization problem becomes quadratic and $M$ is the number of second-order terms in the objective function. If we plug the values of $\langle Q_1^1, \rho_1 \rangle, \ldots, \langle Q_{M_1}^1, \rho_1 \rangle$ into the objective function, then the optimization problem reduces to an efficiently solvable semidefinite program. Hence by enumerating all possible values of $\langle Q_1^1, \rho_1 \rangle, \ldots, \langle Q_{M_1}^1, \rho_1 \rangle$ one can efficiently solve the original problem when $M$ is small. Since this approach naturally extends to the $k$-partite case for $k \geq 2$, we obtain the following general result.

**Theorem 1** (Informal. See Section 3). Given any Hermitian $Q$ (of dimension $d$) and its decomposition $Q = \sum_{i=1}^{M} Q_i^1 \otimes \cdots \otimes Q_i^k$, $\text{OptSep}(Q)$ can be approximated with additive error $\delta$ in quasi-polynomial time\(^1\) in $d$ and $1/\delta$ if $kM$ is $O(\text{poly-log}(d))$.

By exploiting the space-efficient algorithm design strategy above, this algorithm can also be made space-efficient. To facilitate the later applications to complexity classes, we choose the input size to be some $n$ such that $d = \exp(\text{poly}(n))$.

**Corollary 1** (Informal. See Section 3). If $kM/\delta \in O(\text{poly}(n))$, the quantity $\text{OptSep}(Q)$ can be approximated with additive error $\delta$ in $\text{PSPACE}$.

As a direct application, we prove the following variant of QMA(2) belongs to PSPACE where QMA(2)[poly($n$), $O(\text{log}(n))$] refers to the model in which the verifier only performs $O(\text{log}(n))$ elementary gates that act on both proofs at the same time and a polynomial number of other elementary gates.

**Corollary 2.** QMA(2)[poly($n$), $O(\text{log}(n))$] $\subseteq$ PSPACE.

This result establishes the first PSPACE upper bound for a variant of QMA(2) where the verifier is allowed to generate some quantum entanglement between two proofs. In contrast, previous results are all about variants with nonadaptive or adaptive local measurements, such as BellQMA(2) or LOCCQMA(2).

We also study Problem\(^1\) when $Q$ is a local Hamiltonian over $k$ parties. Recall that a promise version of this problem in the one party case, namely the local-Hamiltonian problem, is QMA-complete [30]. Our definition extends the original local Hamiltonian problem to its $k$-partite version, which, however, is no longer necessarily QMA(k)-complete. Indeed, our result supports this fact in the algorithmic aspect. An independent work of Chailloux and Sattath [12], which complements our result, shows that the 2-partite local Hamiltonian problem defined above lies in QMA.

**Corollary 3** (Informal. See Section 5). Given some local Hamiltonian $Q$ over $k$ parties, $\text{OptSep}(Q)$ can be approximated with additive error $\delta$ in quasi-polynomial time in $d, 1/\delta$; the $k$-partite local Hamiltonian problem is in $\text{PSPACE}$.

2. The second structure made use of is the eigenspace of $Q$ of large eigenvalues, where we establish an algorithm in time exponential in $\|Q\|_F$.

**Theorem 2.** For $Q \geq 0$, $\text{OptSep}(Q)$ can be approximated with additive error $\delta$ in time $\exp(O(\text{log}(d) + \delta^{-2} \|Q\|_F^2 \ln(\|Q\|_F/\delta)))$.

---

\(^1\)Quasi-polynomial time is upper bounded by $2^{O((\text{log}$ $n)^c)}$ for some fixed $c$, where $n$ is the input size.
A similar running time \( \exp(O(\log^2 d)\delta^{-2}\|Q\|_F^2) \) was obtained in [8] using some advanced results (i.e., the semidefinite programming hierarchy for finding symmetric extension [14] and an improved quantum de Finetti-type bound) in quantum information theory. In contrast, our algorithm only uses fundamental operations of matrices and epsilon-nets. To approximate with precision \( \delta \), it suffices to consider the eigenspace of \( Q \) of eigenvalues greater than \( \delta \) whose dimension is bounded by \( \|Q\|_F^2/\delta^2 \). Nevertheless, naively enumerating density operators over that subspace does not work since one cannot detect the separability of those density operators. We circumvent this difficulty by making nontrivial use of the Schmidt decomposition of bipartite pure states.

We note, however, that other results in [8] do not follow from our algorithm, and our method cannot be seen as a replacement of the kernel technique therein. Furthermore, our method does not extend to the \( k \)-partite case, as there is no Schmidt decomposition in that case.

### Subsequent development:

There are a few recent results on this topic after the first appearance of this paper. In [18], a new QMA(2)-complete problem is proposed, which is related to entanglement detection in a certain circuit model. In [9], an epsilon-net algorithm is proposed to serve as an alternative to the semidefinite programming hierarchy in [8, 14], while achieving comparable parameters. In [25], an improved semidefinite programming hierarchy is proposed with exponential improvement on the error dependence (c.f. [8, 14]).

### Organizations:

The necessary background knowledge on the epsilon-nets is introduced in Section 2. The main algorithm based on the decomposability of \( Q \) is illustrated in Section 3, which is followed by the simulation of variants of QMA(2) in Section 4 and the local Hamiltonian case in Section 5. Finally, the demonstration of an algorithm with running time exponential in \( \|Q\|_F \) for Problem 1 can be found in Section 6.

### Notations:

We assume familiarity with standard concepts from quantum information [35, 30, 38]. Let \( \mathcal{A}, \mathcal{B} \) denote complex Euclidean spaces and \( \mathcal{L}(\mathcal{A}), \text{Herm}(\mathcal{A}), \text{D}(\mathcal{A}) \) denote the linear, Hermitian and density operators over \( \mathcal{A} \) respectively. We denote the trace norm of operator \( Q \) by \( \|Q\|_\text{tr} \), i.e. \( \|Q\|_\text{tr} = \text{Tr}(Q^*Q)^{1/2} \) where \( Q^* \) stands for the conjugate transpose of \( Q \). The Frobenius norm is denoted by \( \|Q\|_F \) and the operator norm is denoted by \( \|Q\|_\text{op} \). The \( \ell_1 \) norm of vector \( x \in \mathbb{C}^n \) is denoted by \( \|x\|_1 = \sum_{i=1}^n |x_i| \) and its \( \ell_\infty \) norm is denoted by \( \|x\|_\infty = \max_{i=1,\ldots,n} |x_i| \). We use \( \|\cdot\| \) to denote the Euclidean norm. The unit ball of \( \mathbb{C}^n \) under a certain norm \( \|\cdot\| \) is denoted by \( \mathcal{B}(\mathbb{C}^n, \|\cdot\|) \).

### 2. Epsilon Net

The epsilon-net (or \( \varepsilon \)-net) is an important concept in several mathematical topics. For our purpose, we adopt the following definition of \( \varepsilon \)-net.

**Definition 1 (\( \varepsilon \)-net).** Let \( (X,d) \) be any metric space and let \( \varepsilon > 0 \). A subset \( \mathcal{N}_\varepsilon \) is called an \( \varepsilon \)-net of \( X \) if for each \( x \in X \), there exists \( y \in \mathcal{N}_\varepsilon \) with \( d(x,y) \leq \varepsilon \).

---

2 We will abuse the notation later where the metric \( d \) is replaced by the norm from which the metric is induced.
Now we turn to the particular ε-net in this paper. Let \( \mathcal{H} \) be any Hilbert space of dimension \( d \) and \( Q = Q(M, w) = (Q_1, Q_2, \ldots, Q_M) \) be a sequence of operators on \( \mathcal{H} \) with \( \|Q_i\|_{op} \leq w \), for all \( i \). Define the \( Q \)-space, denoted by \( \text{SP}(Q) \), as

\[
\text{SP}(Q) = \{(\langle Q_1, \rho \rangle, \langle Q_2, \rho \rangle, \ldots, \langle Q_M, \rho \rangle) : \rho \in \mathcal{D}(\mathcal{H})\} \subseteq \mathbb{C}^M.
\]

The set is convex and compact, and a (possibly proper) subset of \( \text{Raw-}(M, w) = \{(q_1, q_2, \ldots, q_M) : \forall i, q_i \in \mathbb{C}, |q_i| \leq w\} \). In the following, we construct an ε-net of the metric space \( (\text{SP}(Q), \ell_1) \) by first generating an ε-net of \( (\text{Raw-}(M, w), \ell_1) \) via a standard procedure and then selecting those points close to \( Q \)-space.

**Selection process**

The selection process determines if some point \( \vec{p} \) in \( \text{Raw-}(M, w) \) is close to \( \text{SP}(Q) \). Denote by \( \text{dis}(\vec{p}) \) the distance of \( \vec{p} \in \mathbb{C}^M \) to \( \text{SP}(Q) \), i.e.,

\[
\text{dis}(\vec{p}) = \min_{\vec{q} \in \text{SP}(Q)} \|\vec{p} - \vec{q}\|_1.
\]

The distance \( \text{dis}(\vec{p}) \) can be efficiently computed in time by casting the problem as a semidefinite program (e.g., see appendix Appendix A.1). Although semidefinite programs admit polynomial time solutions (when \( Q \) has a concise description, then this corresponds to exponential time solutions), it is generally unknown whether these polynomial time solutions can also be made space-efficient (i.e., in poly-logarithmic space). In our case in which \( Q \) has a concise description, space-efficient solutions correspond to a PSPACE upper bound. Thus we need to develop our own space-efficient algorithm for this problem. Due to the duality of the \( \ell_1 \) norm, one has

\[
\text{dis}(\vec{p}) = \min_{\rho \in \mathcal{D}(\mathcal{H})} \max_{\vec{z} \in \mathcal{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \Re \langle \vec{p} - \vec{q}(\rho), \vec{z} \rangle,
\]

where \( \vec{q}(\rho) = (\langle Q_1, \rho \rangle, \langle Q_2, \rho \rangle, \ldots, \langle Q_M, \rho \rangle) \in \mathbb{C}^M \). By rephrasing \( \text{dis}(\vec{p}) \) in the above form, one shows the quantity \( \text{dis}(\vec{p}) \) is actually an equilibrium value. This follows from the well-known extensions of von’ Neumann’s Min-Max Theorem [37] [15]. One can easily verify that the density operator set \( \mathcal{D}(\mathcal{H}) \) and the unit ball of \( \mathbb{C}^M \) under \( \ell_\infty \) norm are convex and compact and the objective function is a bilinear form over the two sets.

\[
\min_{\rho \in \mathcal{D}(\mathcal{H})} \max_{\vec{z} \in \mathcal{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \Re \langle \vec{p} - \vec{q}(\rho), \vec{z} \rangle = \max_{\vec{z} \in \mathcal{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \min_{\rho \in \mathcal{D}(\mathcal{H})} \Re \langle \vec{p} - \vec{q}(\rho), \vec{z} \rangle. \tag{1}
\]

Fortunately, there is an efficient algorithm in both time and space (in terms of \( d, M, w, 1/\epsilon \)) to approximate \( \text{dis}(\vec{p}) \) with additive error \( \epsilon \). The main tool used here is the so-called matrix multiplicative weight update method [2] [28] [39]. Similar min-max forms also appeared before in a series of work on quantum complexity [27] [40] [22]. The algorithm presented here is another simple application of this powerful method. For the sake of completeness, we provide the proof of the following lemma in Appendix Appendix A.2.

**Lemma 1.** Given any point \( \vec{p} \in \text{Raw-}(M, w) \) and \( \epsilon > 0 \), there is an algorithm that approximates \( \text{dis}(\vec{p}) \) with additive error \( \epsilon \) in \( \text{poly}(d, M, w, 1/\epsilon) \) time. Furthermore, if \( d \) is considered as the input size and \( M, w, 1/\epsilon \in O(\text{poly-log}(d)) \), this algorithm is also efficient in parallel, namely, it is inside NC.
Construction of the $\epsilon$-net

Given any $Q(M, \omega)$ and $\epsilon > 0$, we construct the $\epsilon$-net of $\text{SP}(Q)$ as follows.

- Construct the $\epsilon$-net of the set $\text{Raw}-(M, \omega)$ with the metric induced from the $\ell_1$ norm. Denote such an $\epsilon$-net by $\mathcal{R}_\epsilon$.
- For each point $\vec{p} \in \mathcal{R}_\epsilon$, determine $\text{dis}(\vec{p})$ and select it to $\mathcal{N}_\epsilon$ if $\text{dis}(\vec{p}) \leq \epsilon$. We claim $\mathcal{N}_\epsilon$ is the $\epsilon$-net of $(\text{SP}(Q), \ell_1)$.

The construction for the first step is rather routine. Creating an $\epsilon'$-net $T'_\epsilon$ over a bounded complex region $\{z \in \mathbb{C} : \|z\| \leq \omega\}$ is simple: we can place a 2D grid over the complex plane to cover the disk $\|z\| \leq \omega$. Simple argument shows $|T'_\epsilon| \in O(\frac{M}{\epsilon^2})$. Then $\mathcal{R}_\epsilon$ can be obtained by cross-producing $T'_\epsilon$ for $M$ times. To ensure the closeness in the $\ell_1$ norm, we will choose $\epsilon' = \epsilon / M$.

**Theorem 3.** The $\mathcal{N}_\epsilon$ constructed above is indeed an $\epsilon$-net of $(\text{SP}(Q), \ell_1)$ with cardinality at most $O((\frac{wM^2}{\epsilon^2})^M)$. For any point $\vec{n} \in \mathcal{N}_\epsilon$, we have $\text{dis}(\vec{n}) \leq \epsilon$.

**Proof.** First we show $\mathcal{R}_\epsilon$ is indeed an $\epsilon$-net of $(\text{Raw}-(M, \omega), \ell_1)$. To that end, consider any point $\vec{p} \in \text{Raw}-(M, \omega)$. From the construction of $\mathcal{R}_\epsilon$, there is some point $\vec{q} \in \mathcal{R}_\epsilon$ such that $\|\vec{p} - \vec{q}\|_\infty \leq \epsilon'$. Then we have $\|\vec{p} - \vec{q}\|_1 \leq M\|\vec{p} - \vec{q}\|_\infty \leq Me' \leq \epsilon$. Since $\mathcal{N}_\epsilon \subseteq \mathcal{R}_\epsilon$, one has $|\mathcal{N}_\epsilon| \leq |\mathcal{R}_\epsilon| \in O((\frac{wM^2}{\epsilon^2})^M)$.

In order to show $\mathcal{N}_\epsilon$ is the required $\epsilon$-net, consider any point $\vec{p} \in \text{SP}(Q)$. Since $\text{SP}(Q) \subseteq \text{Raw}-(M, \omega)$, there exists a point $\vec{p}' \in \mathcal{R}_\epsilon$ such that $\|\vec{p} - \vec{p}'\|_1 \leq \epsilon$. Hence we have $\text{dis}(\vec{p}') \leq \epsilon$ and the point $\vec{p}'$ will be selected, namely $\vec{p}' \in \mathcal{N}_\epsilon$. Finally, it is a simple consequence of the selection process that every point $\vec{n} \in \mathcal{N}_\epsilon$ has $\text{dis}(\vec{n}) \leq \epsilon$.

**Remarks.** If one choose $Q$ to be $Q(d^2, 1) = \{|i\rangle\langle j| : i, j = 1, \cdots, d\}$, one can generate the $\epsilon$-net of the density operator set with the $\ell_1$ norm in the method described above. It is akin to generating an $\epsilon$-net for every entry of the density operator. At the other extreme, one can also efficiently generate the $\epsilon$-net of a small size $\text{SP}(Q)$ even when the space dimension $d$ is relatively large.

3. The Main Algorithm

Without loss of generality, we assume $A_1, A_2$ are identical, and of dimension $d$ in Problem 1. Moreover, our algorithm will deal with the set of product states rather than separable states. Namely, we consider the following problem.

$$\max_{\rho} \quad \langle Q, \rho \rangle,$$

subject to: $\rho = \rho_1 \otimes \rho_2, \rho_1 \in \text{D}(A_1), \rho_2 \in \text{D}(A_2)$.

It is easy to see these two optimization problems are equivalent since product states are extreme points of the set of separable states. Our algorithm works for both maximization and minimization of the objective function and can be extended naturally to the $k$-partite version.

**Problem 2** (k-partite version). Given any Hermitian matrix $Q$ over $A_1 \otimes \cdots \otimes A_k$ ($k \geq 2$), compute the optimum value $\text{OptSep}(Q)$ with additive error $\delta$.

$$\max_{\rho} \quad \langle Q, \rho \rangle,$$

subject to: $\rho = \rho_1 \otimes \cdots \otimes \rho_k, \forall \rho_i \in \text{D}(A_i)$.
Let \( Q_t(M, w_t) = (Q^t_1, Q^t_2, \cdots, Q^t_M) \) for \( t=1, \ldots, k-1 \). Let \( W = \Pi_{i=1}^k w_i \). Generate the \( \epsilon_t \)-net (by Theorem 3) of \( (\mathbf{SP}(Q_t), \ell_1) \) for each \( t=1, \ldots, k-1 \) with \( \epsilon_t = w_i \delta / (k-1)W \) and denote such a set by \( N^t_{\epsilon_t} \). Also let OPT store the optimum value.

2. For each point \( \bar{q} = (\bar{q}^1, \bar{q}^2, \cdots, \bar{q}^{k-1}) \in N^1_{\epsilon_1} \times N^2_{\epsilon_2} \times \cdots \times N^{k-1}_{\epsilon_{k-1}} \), let \( Q^k \) be

\[
Q^k = \sum_{i=1}^M q^i_1 q^i_2 \cdots q^{i-1}_i Q^k_i,
\]

and calculate \( \bar{Q}^k = \frac{1}{k}(Q^k + Q^{k\ast}) \). Then compute the maximum eigenvalue of \( \bar{Q}^k \), denoted by \( \lambda_{\text{max}}(\bar{q}) \). Update OPT as follows: \( \text{OPT} = \max\{\text{OPT}, \lambda_{\text{max}}(\bar{q})\} \).

3. Return \( \text{OPT} \).

Figure 1: The main algorithm with precision \( \delta \).

Before describing the algorithm, we need some terminology about the decomposability of a multipartite operator. Any Hermitian operator \( Q \) is called \( M \)-decomposable if there exists \( (Q^1_t, Q^2_t, \cdots, Q^M_t) \in \mathcal{L}(\mathcal{A}_t)^M \) for each \( t \) such that

\[
Q = \sum_{i=1}^M Q^1_i \otimes Q^2_i \otimes \cdots \otimes Q^{k-1}_i \otimes Q^k_i.
\]

To facilitate the use of \( \epsilon \)-net, we adopt a slight variation of the decomposability above. Let \( \bar{w} \in \mathbb{R}^k \) denote the widths of operators over each \( \mathcal{A}_t \). Any \( Q \) is called \( (M, \bar{w}) \)-decomposable if \( Q \) is \( M \)-decomposable and the widths of those operators in the decomposition are bounded in the sense that \( \max_i \|Q^i_t\|_\text{op} \leq w_t \) for each \( t \). It is noteworthy to mention that the decomposability defined above is related to the concept tensor rank. However, given the representation \( Q \) as input, it is hard in general to compute the tensor rank of \( Q \) or its corresponding decomposition. Therefore, for any \( (M, \bar{w}) \)-decomposable \( Q \) we assume its corresponding decomposition is also a part of the input to our algorithm.

**Theorem 4.** Let \( Q \) be some \((M, \bar{w})\)-decomposable Hermitian over \( \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k \) (each \( \mathcal{A}_i \) is of dimension \( d \)) and \( \delta > 0 \). Also let \((Q^1_t, Q^2_t, \cdots, Q^M_t), t=1, \ldots, k\) be the operators in the corresponding decomposition of \( Q \). The algorithm shown in Fig. 1 approximates \( \text{OptSep}(Q) \) of Problem 2 with additive error \( \delta \) in \( O\left(\frac{(k-1)^2W^2M^2}{\delta^2}\right) \times \text{poly}(d, M, k, W, 1/\delta) \) time where \( W = \Pi_{i=1}^k w_i \).

**Proof.** By substituting the identity \( Q = \sum_{i=1}^M Q^1_i \otimes Q^2_i \otimes \cdots \otimes Q^{k-1}_i \otimes Q^k_i \), the optimization problem becomes

\[
\text{max: } \left\langle \sum_{i=1}^M p^i_1 p^i_2 \cdots p^{i-1}_i Q^k_i, \rho_k \right\rangle
\]

subject to: \( \forall t \in \{1, \cdots, k-1\}, \bar{p}_t \in \mathbf{SP}(Q_t(M, w_t)), \) and \( \rho_k \in \mathbf{D}(\mathcal{A}_k) \).

Thus, solving the optimization problem amounts to first enumerating \( \bar{p}_t \in \mathbf{SP}(Q_t(M, w_t)) \) for each \( t \), and then solving the optimization problem over \( \mathbf{D}(\mathcal{A}_k) \).
Consider any point $\tilde{p} = (p^1, p^2, \cdots, p^{k-1}) \in \text{SP}(Q_i)^{k-1}$ where $\text{SP}(Q_i)^{k-1}$ denotes $\text{SP}(Q_1) \times \cdots \times \text{SP}(Q_{k-1})$. Due to Theorem 3, there is at least one point $\tilde{q} = (q^1, q^2, \cdots, q^{k-1}) \in \{N_{t_i}^j\}^{k-1}$ where $\{N_{t_i}^j\}^{k-1}$ denotes $N_{t_i}^1 \times N_{t_i}^2 \times \cdots \times N_{t_i}^{k-1}$ such that $\|\tilde{q} - \tilde{p}\|_1 \leq \epsilon_i$ for $t=1,\ldots,k-1$. The choice of $\tilde{Q}^k$ is to symmetrize $Q^k$. With $\tilde{Q}^k$ being Hermitian, it is clear that $\lambda_{\max}(\tilde{q}) = \lambda_{\max}(\tilde{p}, \rho_k)$. Now let’s analyze how much error will be induced in this process.

Let $P^k(\tilde{p}) = \sum_{i=1}^M p_i^1 p_i^2 \cdots p_i^{k-1} Q_i^k$ and $\tilde{p}^k = \frac{1}{\lambda}(P^k + P^{k*})$. It is not hard to see that $P^k = \tilde{p}^k$. The error bound is achieved by applying a chain of triangle inequalities as follows. Firstly, one has

$$\|\tilde{p}^k - Q^k\|_{\text{op}} = \left\| \frac{1}{2} (P^k - Q^k) + \frac{1}{2} (P^{k*} - Q^{k*}) \right\|_{\text{op}} \leq \|P^k - Q^k\|_{\text{op}}.$$ 

Substitute the expressions for $P^k, Q^k$ and apply the standard hybrid argument.

$$\|P^k - Q^k\|_{\text{op}} = \left\| \sum_{i=1}^M (p_i^1 p_i^2 \cdots p_i^{k-1} - q_i^1 q_i^2 \cdots q_i^{k-1}) Q_i^k \right\|_{\text{op}}$$

which is immediately upper bounded by the sum of the following terms,

$$\sum_{i=1}^M |p_i^1 - q_i^1| \|p_i^2 \cdots p_i^{k-1}\|_{\text{op}}, \cdots, \sum_{i=1}^M |q_i^1 \cdots q_i^{k-2}| \|p_i^{k-1} - q_i^{k-1}\|_{\text{op}}.$$ 

As the $t^{th}$ term above can be upper bounded by $\epsilon_i W / w_t$ for each $t$, we have,

$$\|\tilde{p}^k - Q^k\|_{\text{op}} \leq \epsilon_1 W / w_1 + \epsilon_2 W / w_2 + \cdots + \epsilon_{k-1} W / w_{k-1} = \frac{\delta}{k-1} + \cdots + \frac{\delta}{k-1} = \delta.$$ 

Hence the optimum value for any fixed $\tilde{p}$ won’t differ too much from the one for its approximation $\tilde{q}$ in the $\epsilon$-net. This is because

$$\max_{\rho_k \in \text{D}(A_k)} \left\langle \tilde{p}^k, \rho_k \right\rangle = \max_{\rho_k \in \text{D}(A_k)} \left\langle Q^k, \rho_k \right\rangle + \left\langle \tilde{p}^k - Q^k, \rho_k \right\rangle.$$ 

By Hölder Inequalities we have $|\left\langle \tilde{p}^k - Q^k, \rho_k \right\rangle| \leq \|\tilde{p}^k - Q^k\|_{\text{op}}\|\rho_k\|_{\text{tr}} \leq \delta,$

$$\lambda_{\max}(\tilde{q}) - \delta \leq \max_{\rho_k \in \text{D}(A_k)} \left\langle \tilde{p}^k(\tilde{p}), \rho_k \right\rangle \leq \lambda_{\max}(\tilde{q}) + \delta.$$ 

We now optimize $\tilde{p}$ over $\text{SP}(Q_i)^{k-1}$ and the corresponding $\tilde{q}$ will run over the $\epsilon$-net $\{N_{t_i}^j\}^{k-1}$. As every point $\tilde{q} \in \{N_{t_i}^j\}^{k-1}$ is also close to $\text{SP}(Q_i)^{k-1}$ in the sense that $\text{dis}(\tilde{q}) \leq \epsilon_i$ for each $t$, we have

$$\max_{\tilde{q} \in \{N_{t_i}^j\}^{k-1}} \lambda_{\max}(\tilde{q}) - \delta \leq \max_{\tilde{p} \in \text{SP}(Q_i)^{k-1}} \max_{\rho_k \in \text{D}(A_k)} \left\langle \tilde{p}^k(\tilde{p}), \rho_k \right\rangle \leq \max_{\tilde{q} \in \{N_{t_i}^j\}^{k-1}} \lambda_{\max}(\tilde{q}) + \delta.$$ 

Finally, it is not hard to see that $\text{OPT} = \max_{\tilde{q} \in \{N_{t_i}^j\}^{k-1}} \lambda_{\max}(\tilde{q})$ and therefore

$$\text{OPT} - \delta \leq \text{OptSep}(Q) \leq \text{OPT} + \delta.$$
Now let us analyze the efficiency of this algorithm. The total number of points in the ε-net \(\mathcal{N}_\epsilon\) is upper bounded by \(O((\frac{(k-1)^2W^2M^2}{\epsilon^2})(k-1)M)\). The generation of each point \(\bar{q}\) will cost time polynomial in \(d, M, W, 1/\delta\) (See Lemma 1). Afterward, one needs to calculate \(Q^k\) and its maximum eigenvalue for each point, which can be done in time polynomial in \(d, k, M\). Thus, the total running time is bounded by \(O((\frac{(k-1)^2W^2M^2}{\epsilon^2})(k-1)M) \times \text{poly}(d, M, k, W, 1/\delta)\). \(\square\)

**Remarks.** All operations in the algorithm described in Fig. 1 can be implemented efficiently in parallel in some situation. This is because fundamental operations of matrices can be done in NC and the calculation of \(\text{dis}(\bar{p})\) can be done in NC (See Lemma 1) when \(M, W, k, 1/\delta\) are in nice forms of \(d\).

**Corollary 4.** Let \(n\) be the input size such that \(d = \exp(\text{poly}(n))\), if \(W/\delta \in O(\text{poly}(n))\), \(kM \in O(\text{poly}(n))\), then \(\text{OptSep}(Q)\) can be approximated with additive error \(\delta\) in \(\text{PSPACE}\).

**Proof.** Given \(Q\) and its decomposition, consider the following algorithm

1. Enumerate each point \(\bar{p} = (\bar{p}_1, \cdots, \bar{p}_{k-1})\) in the raw set \(\mathcal{R}^1_{\epsilon_1} \times \cdots \times \mathcal{R}^{k-1}_{\epsilon_{k-1}}\).
2. Compute \(\text{dis}(\bar{p}_t)\) for each \(t=1,\ldots,k-1\). If \(\bar{p}\) is a valid point in the epsilon-net, then we continue with the rest part in Step 2 of the algorithm in Fig. 1.
3. Compare the values obtained by each point \(\bar{p}\) and keep the optimum one.

Given the condition \(W/\delta \in O(\text{poly}(n))\), \(kM \in O(\text{poly}(n))\), the first part of the algorithm can be done in polynomial space. This is because in this case each point in the raw set can be represented by polynomial space and therefore enumerated in polynomial space. The second part is more difficult. Computing \(\text{dis}(\bar{p}_t)\) for each \(t=1,\ldots,k-1\) can be done in \(\text{NC}(\text{poly}(n))\) as shown in Lemma 1. Step 2 in the main algorithm only contains fundamental operations of matrices and the spectrum decomposition. Thus, it also admits a parallel algorithm in \(\text{NC}(\text{poly}(n))\). One can easily compose the two circuits and get a polynomial space implementation by the relation \(\text{NC}(\text{poly})=\text{PSPACE}\). The third part can obviously be done in polynomial space. Thus, by composing these three polynomial-space implementable parts, one proves the whole algorithm can be done in \(\text{PSPACE}\). \(\square\)

### 4. Simulation of several variants of QMA(2)

This section illustrates the use of the algorithm shown in Section 3 to simulate some variants of the complexity class QMA(2). The idea is to show for those variants, the corresponding POVM matrices of acceptance are \((M, \bar{w})\)-decomposable with small Ms. Recall the definition of the complexity class QMA(2).

**Definition 2.** A language \(L\) is in \(QMA(2)\) if there exists a polynomial-time generated family of quantum verification circuits \(Q = \{Q_n\}_{n \in \mathbb{N}}\) such that for any input \(x\) of size \(n\), the circuit \(Q_n\) implements a two-outcome measurement \(\{Q^\text{acc}_x, \mathbb{I} - Q^\text{acc}_x\}\). Furthermore,

- **Completeness:** If \(x \in L\), there exist \(|\psi_1\rangle \in A_1, |\psi_2\rangle \in A_2\), each of \(m\) qubits,
  \[\langle Q^\text{acc}_x, |\psi_1\rangle \langle \psi_1 | \otimes |\psi_2\rangle \langle \psi_2 | \geq c.\]

- **Soundness:** If \(x \notin L\), then for any states \(|\psi_1\rangle \in A_1, |\psi_2\rangle \in A_2\),
  \[\langle Q^\text{acc}_x, |\psi_1\rangle \langle \psi_1 | \otimes |\psi_2\rangle \langle \psi_2 | \leq s.\]
We call QMA(2)=QMA(2)_{poly(n),2/3,1/3}. It is easy to see that simulating the complexity class QMA(2) amounts to distinguishing between the two promises of the maximum acceptance probability (i.e. OptSep(\(Q^{acc}\_x\)).

The first example is the variant with only logarithm-size proofs, namely QMA(2)_{O(log(n)),2/3,1/3}. It is not hard to find out the corresponding POVMs of acceptance (i.e. \(Q^{acc}\)) need to be (poly(n),\(\vec{w}\))-decomposable where \(\vec{w}=(1,1)\) since \(A_1, A_2\) are only of polynomial dimension. Thus, it follows directly from Corollary 4 that OptSep(\(Q^{acc}\_x\)) can be approximated in polynomial space. Namely,

\[
QMA(2)_{O(log(n)),2/3,1/3} \subseteq \text{PSPACE}.
\]

The next example is slightly less trivial. Before moving on, we need some terminology about the quantum verification circuits \(Q\). Assume the input \(x\) is fixed from now on. Let \(A_1, A_2\) be the Hilbert space of size \(d_A\) for the two proofs and let \(V\) be the ancillary space of size \(d_V\). Then the quantum verification process will be carried out on the space \(A_1 \otimes A_2 \otimes V\) with some initial state \(|\psi_1\rangle \otimes |\psi_2\rangle \otimes |\vec{0}\rangle\) where \(|\psi_1\rangle, |\psi_2\rangle\) are provided by the provers. The verification process is also efficient in the sense that the whole circuit only consists of polynomial elementary gates. Without loss of generality, we can fix one universal gate set for the verification circuits. Particularly, we choose the universal gate set to be single qubit gates plus the CNOT gates \([35]\). One can also choose other universal gate sets without any change of the main result.

We categorize all elementary gates in the verification circuits into two types. A gate is of type-I if it only affects the qubits within the same space (i.e, \(A_1, A_2,\) or, \(V\)). Otherwise, this gate is of type-II. It is easy to see single qubit gates are always type-I gates. The only type-II gates are CNOT gates whose control qubit and target qubit sit in different spaces. Let \(p, r : \mathbb{N} \rightarrow \mathbb{N}\) be polynomial-bounded functions. A polynomial-time generated family of quantum verification circuits \(Q\) is called \(Q[p, r]\) if each \(Q_n\) only contains \(p(n)\) type-I elementary gates and \(r(n)\) type-II elementary gates.

**Definition 3.** A language \(L\) is in QMA(2)\(_{m,c,s}\)[\(p, r\)] if \(L\) is in QMA(2)\(_{m,c,s}\) with some \(Q[p, r]\) verification circuit family.

It is easy to see that QMA(2) = QMA(2)[poly, poly] from our definition.

**Lemma 2.** For any family of verification circuits \(Q[p, r]\), the POVM \(Q^{acc}\_x\) is \((4^{c(n)}, (1, 1))-decomposable for any \(n \in \mathbb{N}\) and input \(x\). Moreover, this decomposition can be calculated in parallel with \(O(t(n)4^{c(n)}) \times poly(n)\) time.

**Proof.** Fix the input \(x\), let us denote the whole unitary that the verification circuit applies on the initial state by \(U = U_tU_{t-1} \cdots U_1\) where each \(U_i\) corresponds to one elementary gate and \(t = p + r\). Without loss of generality, we assume the output bit is the first qubit in the space \(V\) and the verification accepts when that qubit is 1. Let \(\bar{V}\) be the space \(V\) without the first qubit, then we have

\[
Q^{acc}\_x = \text{Tr}_V \left( I_{A_1A_2} \otimes |\vec{0}\rangle \langle 0| \left( U^+ I_{A_1A_2} \otimes I_{\bar{V}} \otimes |1\rangle \langle 1| U \right) I_{A_1A_2} \otimes |\vec{0}\rangle \langle 0| \right).
\]

Let \(P_{t+1} = I_{A_1A_2} \otimes I_{\bar{V}} \otimes |1\rangle \langle 1|\) and \(P_\tau = U_{\tau}P_{t+1}U_{\tau}^{-1}\) for \(\tau = t, t-1, ..., 1\). It is easy to see \(P_1 = U^* (I_{A_1A_2} \otimes I_{\bar{V}} \otimes |1\rangle \langle 1|) U\). Also it is straightforward to verify that \(P_{t+1}\) is 1-decomposable. Now let us observe how the decomposability of \(P_\tau\) changes with \(\tau\).

For each \(\tau\), the unitary \(U_\tau\) either corresponds to a type-I or type-II elementary gate. In the former case, applying \(U_\tau\) won’t change the decomposability. Thus, \(P_\tau\) is \(M\)-decomposable if \(P_{t+1}\)
is. In the latter case, applying $U_\tau$ will potentially change the decomposability in the following way. For any such CNOT gate one has $U_\tau = |0\rangle \langle 0| \otimes I + |1\rangle \langle 1| \otimes X$ where $X$ is the Pauli matrix for the flip. And one can show
\[
P_\tau = (|0\rangle \langle 0| \otimes I)P_{\tau+1}(|0\rangle \langle 0| \otimes I) + (|0\rangle \langle 0| \otimes I)P_{\tau+1}(|1\rangle \langle 1| \otimes X)
+ (|1\rangle \langle 1| \otimes X)P_{\tau+1}(|0\rangle \langle 0| \otimes I) + (|1\rangle \langle 1| \otimes X)P_{\tau+1}(|1\rangle \langle 1| \otimes X).
\]

Thus in general we can only say $P_\tau$ is $4M$-decomposable if $P_{\tau+1}$ is $M$-decomposable. As there are $r(n)$ type-II gates, one immediately has $P_1$ is $4^r(n)$-decomposable. Moreover, each operator appearing in the decomposition is a multiplication of unitaries, $|0\rangle \langle 0|, |1\rangle \langle 1|$ and $X$ in some order, which implies the operator norm of those operators is bounded by 1. Therefore we have $P_1$ is $(4^r(n), (1, 1))$-decomposable.

Finally, it is not hard to verify that multiplications with $I_{A_1, A_2} \otimes |0\rangle \langle 0|$ and partial trace over $\mathcal{V}$ won’t change the decomposability of $P_1$. Namely, we have $Q_\text{acc}^x$ is $(4^r(n), (1, 1))$-decomposable. The above proof can also be considered as the process to compute the decomposition of $Q_\text{acc}^x$. Each multiplication of matrices can be done in $\text{NC}(\text{poly}(n))$. And the total number of multiplications is upper bounded by $O(t(n)4^r(n))$. Therefore, the total parallel running time is upper bounded by $O(t(n)4^r(n)) \times \text{poly}(n)$.

We will show that when the number of type-II gates is relatively small, one can simulate this complexity model efficiently by the algorithm in Fig. 1.

**Corollary 5.** QMA(2)[poly($n$), $O(\log(n))$] $\subseteq$ PSPACE.

**Proof.** This is a simple consequence of Lemma 4 and Corollary 5. For any fixed $x$ of length $n$, one can first compute the decomposition of $Q_\text{acc}^x$ in parallel with $O(t(n)4^r(n)) \times \text{poly}(n)$ time, which is parallel polynomial time in $n$ when $r(n) = O(\log(n))$ and $t(n) \in \text{poly}(n)$. Hence the first step can be done in polynomial space via the relation $\text{NC}(\text{poly})=\text{PSPACE}$ [5].

Then one can invoke the parallel algorithm in Corollary 5 to approximate OptSep($Q_\text{acc}^x$) to sufficient precision $\delta$ such that one can distinguish between the two promises. Precisely in this case, we choose those parameters as follows,

$$k = 2, W = 1, M = 4^{O(\log(n))} = \text{poly}(n), 1/\delta = \text{poly}(n).$$

Thus the whole algorithm can be done in polynomial space.

**Remarks.** Although the proof of the result is not too technical, it establishes the first non-trivial upper bound (PSPACE in this case) for variants of QMA(2) that allow quantum operations acting on both proofs at the same time.

However, our results are hard to extend to the most general case of QMA(2). This is because SWAP-test operation uses many more type-II gates than what is allowed in our method. And SWAP-test seems to be inevitable if one wants to fully characterize the power of QMA(2).

### 5. Quasi-polynomial algorithms for local Hamiltonian cases

In this section, we illustrate that if $Q$ appears in the objective function that is a local Hamiltonian then the optimal value OptSep($Q$) can be efficiently computed by our main algorithm. Consider any $k$-partite space $A_1 \otimes A_2 \otimes \cdots \otimes A_k$ where each $A_i$ contains $n$ qubits (i.e., of dimension $2^n$).
Definition 4. Any Hermitian $Q$ over $A_1 \otimes \cdots \otimes A_k$ is a 1-local Hamiltonian if $Q$ is expressible as $Q = \sum_{i=1}^{r} H_i$ where each term is a Hermitian operator acting on at most $l$ qubits among $k$ parties.

Hamiltonians are widely studied in physics since they usually characterize the energy of a physical system. Local Hamiltonians are of particular interest since they refer to the energy of many interesting models in low-dimension systems. Our algorithm can be considered as a way to find the minimum energy of some physical system achieved by separable states.

Local Hamiltonians are also appealing to computational complexity theorists since the discovery of the promise 5-local Hamiltonian problem \[30\] which turns out to be QMA-complete. Precisely, it refers to the following promise problem when $k = 1, l = 5$.

Problem 3 (k-partite l-local Hamiltonian problem). Take the expression $Q = \sum_{i=1}^{r} H_i$ for any l-local Hamiltonian over $A_1 \otimes \cdots \otimes A_k$ as input\[4\] where $\|H_i\|_{op} \leq 1$ for each $i$. Let $\text{OptSep}(Q)$ denote the minimum value of $\langle Q, \rho \rangle$ achieved for some $\rho \in \text{SepD}(A_1 \otimes \cdots \otimes A_k)$. The goal is to tell between the following two promises: either $\text{OptSep}(Q) \geq a$ or $\text{OptSep}(Q) \leq b$ for some $a > b$ with inverse polynomial gap.

When $k = 1$, the promise problem defined above is exactly the original l-local Hamiltonian problem. Subsequent results demonstrate that it remains QMA-complete even when $l = 3, 4, 5$ \[3\] \[31\] \[36\]. Our definition of the promise problem naturally extends to the $k$-partite case. We refer to Chapter 14 in \[30\] for technical details. It is not hard to see that $k$-partite l-local Hamiltonian problem belongs to QMA(k) by applying similar techniques in the original proof. However, it does not remain as QMA(k)-complete problems. Indeed, Chailloux and Sattath \[12\] proved that the $k$-partite local Hamiltonian problem defined above lies in QMA for constant $k$.

Lemma 3. Any l-local Hamiltonian $Q$ over $A_1 \otimes \cdots \otimes A_k$ such that $Q = \sum_{i=1}^{r} H_i$ and $\|H_i\|_{op} \leq w$ is ($O((4nk)^l), w$)-decomposable.

Proof. Since $Q$ is a l-local Hamiltonian, it is easy to see $r \leq \binom{kn}{l}$. For each $H_i$ with $\|H_i\|_{op} \leq w$, since it acts only on at most $l$ qubits, it must be $(4^l, w)$-decomposable. Thus $Q$ is $(r4^l, w)$-decomposable. In terms of only $n, k, l$, we have $Q$ is ($O((4nk)^l), w$)-decomposable.

Corollary 6. Take the expression $Q = \sum_{i=1}^{r} H_i$ of any l-local Hamiltonian over $A_1 \otimes \cdots \otimes A_k$ (each $A_i$ is of dimension $d = 2^l$) such that $\|H_i\|_{op} \leq w$ for each $i$ as input. Assuming $k, l = O(1)$, the quantity $\text{OptSep}(Q)$ can be approximated to precision $\delta$ in quasi-polynomial time in $d, w, 1/\delta$.

If $n$ is considered as the input size and $w/\delta = O(\text{poly}(n))$, then $\text{OptSep}(Q)$ can be approximated to precision $\delta$ in PSPACE.

Proof. The proof of the first part follows directly from Lemma\[3\] and Theorem\[4\]. Recall the proof of Lemma\[3\] also provides a way to compute the decomposition of $Q$ given the expression $Q = \sum_{i=1}^{r} H_i$ as input. It is easy to verify that $O(r4^l)$ time (upper bounded by $O((4k \log d)^l)$) is sufficient to complete this computation. After that, one may directly invoke the algorithm in Fig.\[1\] and make use of Theorem\[4\]. Now we substitute the following identities into our main algorithm. Note $k, l = O(1)$ and we have $M = O(\log^{O(1)} d), W = w^{O(1)}$. One immediately gets the total running time bounded by

$$\exp(O(\log^{O(1)} d(\log \log d + \log w/\delta))) \times \text{poly}(d, w, 1/\delta),$$

\[3\] It is noteworthy to mention that the input size of local Hamiltonian problems can be only poly-logarithm in the dimension of the space where $Q$ sits in.
1. Compute the spectral decomposition of $Q = \sum_t \lambda_t |\Psi_t\rangle\langle \Psi_t|$. Choose $\epsilon = \delta/2$ and $\Gamma_\epsilon = \{t : \lambda_t \geq \epsilon\}$ Also let OPT store the optimum value.

2. Generate the $\epsilon$-net of the unit ball of $C|\Gamma_\epsilon|$ under the Euclidean norm with $\epsilon = \frac{\delta}{\|Q\|_F}$. Denote such set by $N_\epsilon$. Then for each point $\alpha \in N_\epsilon$,

   (a) Compute $|\phi_\alpha\rangle = \sum_{t \in \Gamma_\epsilon} a^*_t \sqrt{\lambda_t} |\Psi_t\rangle$ and compute the Schmidt decomposition of $|\phi_\alpha\rangle$, i.e.
   
   $|\phi_\alpha\rangle = \sum_i \mu_i |u_i\rangle |v_i\rangle$,

   where $\mu_1 \geq \mu_2 \geq \cdots$ and $\{u_i\}, \{v_i\}$ are orthogonal bases.

   (b) Update OPT as follows: OPT = max\{OPT, $\mu_1$\}.

3. Return OPT.

---

Figure 2: The algorithm runs in time exponential in $\|Q\|_F/\delta$.

which is quasi-polynomial time in $d, w, 1/\delta$.

For the second part when $n$ is considered as the input size, it is easy to see the computation of the decomposition of $Q$ according to Lemma 3 can be done in NC(poly), henceforth in polynomial space. (Note $M = O(\text{poly}(n))$.) Then by composing with the polynomial-space algorithm implied by Corollary 1, one proves the whole algorithm can be implemented in polynomial space.

Remarks. It is a direct consequence of Corollary 6 that Problem 3 is inside PSPACE.

6. **Exponential running time algorithm in $\|Q\|_F$**

In this section we demonstrate another application of the simple idea “enumeration” by epsilon-net to Problem 3. As a result, we obtained an algorithm with running time exponential in $\|Q\|_F$ (or $\|Q\|_{\text{LOCC}}$ [34]) for computing OptSep($Q$) with additive error $\delta$. A similar running time $\exp(O(\log^2(d)\delta^{-2}\|Q\|_F^2))$ was obtained in [3] using some known results in quantum information theory.

Theorem 5. Given any positive semidefinite $Q$ over $A_1 \otimes A_2$ (of dimension $d \times d$) and $\delta > 0$, the algorithm in Fig. 2 approximates OptSep($Q$) with additive error $\delta$ with running time $\exp(O(\log(d) + \delta^{-2}\|Q\|_F^2 \ln(\|Q\|_F/\delta)))$.

Proof. We first prove the correctness of the algorithm. The analysis will mainly be divided into two parts. Let $S_\epsilon = \text{span}\{|\Psi_t\rangle | t \in \Gamma_\epsilon\}$. The first part shows it suffices to only consider vectors inside the subspace $S_\epsilon$ for approximating OptSep($Q$) with additive error $\delta$. The second one demonstrates that our algorithm in Fig. 2 approximates the optimal value obtained by only considering vectors

\[\|Q\|_F = O(\|Q\|_{\text{LOCC}})\]  

where $\|Q\|_{\text{LOCC}}$ stands for the LOCC norm of the operator $Q$.
we replace the unit ball by its $\epsilon$-net. Then we have
\[
\langle Q, |u\rangle \langle u \otimes |v\rangle \langle v \rangle = \sum_{t \in \Gamma} \lambda_t |\beta_t|^2 + \sum_{t \notin \Gamma} \lambda_t |\beta_t|^2,
\]
where the term (II) is obviously bounded by $\delta/2$ (i.e., $\sum_{t \notin \Gamma} \lambda_t |\beta_t|^2 \leq \delta/2$). For the term (I), it is equivalent to $\text{OptSep}(\bar{Q})$ where $\bar{Q} = \sum_{t \in \Gamma} \lambda_t |\Psi_t\rangle \langle \Psi_t|$. Namely, small eigenvalues are truncated in $\bar{Q}$. Now observe the following identity.
\[
\max_{|u\rangle \langle u|} \langle \bar{Q}, |u\rangle \langle u \otimes |v\rangle \langle v \rangle \rangle = \max_{|u\rangle \langle u|} \sum_{t \in \Gamma} \lambda_t \langle u | \langle v | \Psi_t \rangle \rangle^2 = \max_{|u\rangle \langle u|} \gamma^{|u,v|} \gamma^{|u,v|} = \max_{a \in B(\mathbb{C}^{||v||})} \sum_{t \in \Gamma} \alpha_t^* \sqrt{\lambda_t} \langle u | \langle v | \Psi_t \rangle \rangle^2 = \max_{a \in B(\mathbb{C}^{||v||})} \max_{|u\rangle \langle u|} \langle u | \langle v | \phi_a \rangle \rangle^2 = \max_{a \in B(\mathbb{C}^{||v||})} \max_{|u\rangle \langle u|} \langle u | \langle v | \phi_a \rangle \rangle^2,
\]
where $\gamma^{|u,v|} \in C^{||v||}$ and $\gamma^{|u,v|} = \sqrt{\lambda_t} \langle u | \langle v | \Psi_t \rangle \rangle$ for each $t \in \Gamma$. The second line comes from the duality of the Euclidean norm (i.e., $||y|| = \max_{||z|| \leq 1} |\langle z | y \rangle|$). The third line comes by exchanging positions of the two maximizations. We then make use of the following well-known fact.

**Fact.** For any bipartite vector $|\psi\rangle$ with the Schmidt decomposition
\[
|\psi\rangle = \sum_i \mu_i |u_i\rangle |v_i\rangle,
\]
where $\mu_1 \geq \mu_2 \geq \cdots$ and $\{u_i\}, \{v_i\}$ are orthogonal bases. Then $\max_{|u\rangle \langle u|} \langle u | \langle v | \psi \rangle \rangle = \mu_1$ and the maximum value is obtained by choosing $|u\rangle \langle u| \langle u \rangle \langle u \rangle$ to be $|u_1\rangle \langle u_1|$. It is not hard to see that our algorithm computes exactly the term on the third line except that we replace the unit ball by its $\epsilon$-net. However, this won’t incur too much extra error. For any $a \in B(\mathbb{C}^{||v||})$, there exists $\tilde{a} \in \mathcal{N}_\epsilon$, such that $||a - \tilde{a}|| \leq \epsilon$. Thus, the extra error incurred is $|| \langle u | \langle v | \phi_a \rangle \rangle^2 - \langle u | \langle v | \phi_{\tilde{a}} \rangle \rangle^2 ||$ and can be bounded by
\[
(\|\langle \phi_{\tilde{a}} \rangle \| + \|\phi_{\tilde{a}} \|) \langle u | \langle v | \phi_{\tilde{a}} - \phi_{\tilde{a}} \rangle \rangle \rangle \leq 2 \max_{\|\phi_{\tilde{a}} \| \leq 1} \|\phi_{\tilde{a}} \| \max_{\|\phi_{\tilde{a}} - \phi_{\tilde{a}} \| \leq \epsilon} \|\phi_{\tilde{a}} \| = 2 \sqrt{||Q||_F^2} \times \epsilon \sqrt{||Q||_F} \leq \delta/2,
\]
where $\max_{\|\phi_{\tilde{a}} \| \leq \epsilon} \|\phi_{\tilde{a}} \| \leq \epsilon' \sqrt{||Q||_F^2}$ for any $\epsilon' > 0$ can be verified directly and therefore the total additive error is bounded by $\delta/2 + \delta/2 = \delta$.

Finally, let us turn to the analysis of the efficiency of this algorithm. The spectrum decomposition in the first step takes polynomial time in $d$, so is the same with calculation of $|\psi_a\rangle$. The generation of the $\epsilon$-net of the unit ball is standard and can be done in $O((1 + \frac{\epsilon}{\delta})^{||v||}) \times \text{poly}(||\Gamma\rangle)|$. The last operation, finding the Schmidt decomposition, is equivalent to singular value decompositions, and thus can be done in polynomial time in $d$ as well. Also note $||\Gamma\rangle| \leq \min\{d^2, ||Q||_F^2 / \delta^2 \}$. To sum up, the total running time of the algorithm is upper bounded by $O((1 + \frac{\epsilon}{\delta})^{||v||}) \times \text{poly}(d)$, or equivalently $\exp(O(\log(d) + \delta^{-2} ||Q||_F^2 \ln(||Q||_F / \delta)))$. □
Acknowledgement

We thank Zhengfeng Ji and John Watrous for helpful discussions and anonymous reviewers for helpful comments on the manuscript. This research was supported in part by National Basic Research Program of China Awards 2011CBA00300 and 2011CBA00301, and by NSF of United States Award 1017335.

[1] Aaronson, S., Beigi, S., Drucker, A., Fefferman, B., Shor, P.: The Power of Unentanglement. Theory of Computing, Volume 5, 1–42 (2009).

[2] Arora, S., Hazan, E., Kale, S.: The multiplicative weights update method: a meta algorithm and applications. Theory of Computing, Volume 8, pp. 121-164 (2012).

[3] Aharonov, D., Gottesman, D., Irani, S., Kempe, J.: The power of quantum systems on a line. Communications in Mathematical Physics, 287(1):41–65 (2009).

[4] Beigi, S.: NP vs QMA_{log(2)}. Quantum Information and Computation, 54(1&2):0141–0151 (2010).

[5] Borodin, A.: On relating time and space to size and depth. SIAM Journal on Computing, 6(4):733–744 (1977).

[6] Blier, H., Tapp, A.: All languages in NP have very short quantum proofs. In: Proceedings of the ICQNM, 34–37 (2009).

[7] Brandão, F.: Entanglement Theory and the Quantum Simulation of Many-Body Physics. PhD thesis, Imperial College (2008).

[8] Brandão, F., Christandl, M., Yard, J.: A quasipolynomial-time algorithm for the quantum separability problem. In: Proceedings of the 43rd annual ACM Symposium on Theory of Computation (STOC’11), pp. 343 (2011).

[9] Brandão, F., Harrow, A.: Enumeration over nets algorithm for optimizing over separable quantum states and related problems, personal communication, 2014.

[10] Chen, J., Drucker, A.: Short multi-prover quantum proofs for SAT without entangled measurements. arXiv:1011.0716 (2010).

[11] Chiesa, A., Forbes, M.: Improved Soundness for QMA with Multiple Provers. Chicago Journal of Theoretical Computer Science, 2013(1):Article 1, January 2013. arXiv:1108.2098 [quant-ph].

[12] Chailloux, A., Sattath, O.: The Complexity of the Separable Hamiltonian Problem. In: Proceedings of the 27rd Annual IEEE Conference on Computational Complexity (CCC 2012).

[13] Defant, A., Floret, K.: Tensor norms and operator ideals. North Holland (1992).

[14] Doherty, A., Parrilo, P., Spedalieri, F.: A complete family of separability criteria. Physical Review A, 69:022308 (2004).

[15] Fan, K.: Minimax theorems. In: Proceedings of the National Academy of Sciences, 39:42–47 (1953).
[16] Gathen, J.: Parallel linear algebra. In: Synthesis of Parallel Algorithms, Morgan Kaufmann Publishers (1993).

[17] Gharibian, S.: Strong NP-hardness of the quantum separability problem. Quantum Information and Computation, 10:343 (2010).

[18] Gutoski, G., Hayden P., Milner, K., Wilde, M.-M.: Quantum interactive proofs and the complexity of separability testing. arXiv:1308.5788 [quant-ph].

[19] Grötschel, M., Lovász, L., Schrijver, A.: Geometric algorithms and combinatorial optimization, Springer-Verlag (1993).

[20] Gall, F., Nakagawa, S., Nishimura, H.: On QMA Protocols with Two Short Quantum Proofs. Quantum Information and Computation, 12(7&8):589–600, 2012. arXiv:1108.4306 [quant-ph].

[21] Gurvits, L.: Classical complexity and quantum entanglement. Journal of Computer and System Sciences, 69:448 (2004).

[22] Gutoski, G., Wu, X.: Parallel approximation of min-max problems with applications to classical and quantum zero-sum games. Computational Complexity, 22(2):385-428, the special issue of CCC 2012.

[23] Horodecki, R., Horodecki, P., Horodecki, M., Horodecki, K.: Quantum entanglement. Review of Modern Physics, 81:865 (2009).

[24] Harrow, A., Montanaro, A.: Testing Product States, Quantum Merlin-Arthur Games and Tensor Optimization. Journal of the ACM (JACM), vol 60(1), article 3, February 2013.

[25] Harrow, A., Natarajan, A., Wu, X.: An improved semidefinite programming hierarchy for testing entanglement, unpublished results, 2014.

[26] Ioannou, L.: Computational complexity of the quantum separability problem. Quantum Information and Computation, 7:335 (2007).

[27] Jain, R., Watrous, J.: Parallel approximation of non-interactive zero-sum quantum games. In: Proceedings of the 24th IEEE Conference on Computational Complexity, 243–253 (2009).

[28] Kale, S.: Efficient algorithms using the multiplicative weights update method. PhD thesis, Princeton University (2007).

[29] Kobayashi, H., Matsumoto, K., Yamakami, T.: Quantum Merlin-Arthur Proof Systems: Are Multiple Merlins More Helpful to Arthur? Chicago Journal of Theoretical Computer Science, vol 2009. Article 3 (2009).

[30] Kitaev, A., Shen, A., Vyalyi, M.: Classical and Quantum Computation. American Mathematical Society (2002).

[31] Kempe, J., Kitaev, A., Regev, O: The complexity of the local Hamiltonian problem. SIAM Journal on Computing, 35(5):1070–1097 (2006).

[32] Ling, C., Qi, L., Nie, J., Ye, Y.: Bi-Quadratic Optimization over Unit Spheres and Semidefinite Programming Relaxations. SIAM Journal on Optimization, Vol. 20, No. 3, pp.1286–1310 (2009).
[33] Marriott, C., Watrous, J.: Quantum Arthur-Merlin Games. Computational Complexity, 14(2):122-152 (2005).

[34] Matthews, W., Wehner, S., Winter, A.: Distinguishability of quantum states under restricted families of measurements with an application to quantum data hiding. Comm. Math. Phys., 291 (2009).

[35] Nielsen, M., Chuang, I.: Quantum Computation and Quantum Information. Cambridge University Press (2000).

[36] Oliveira, R., Terhal, B.: The complexity of quantum spin systems on a twodimensional square lattice. Quantum Information and Computation, 8(10):0900–0924 (2008).

[37] Neumann, J.: Zur theorie der gesellschaftsspiele. Mathematische Annalen, 100:295–320 (1928).

[38] Watrous, J.: Lecture Notes on Theory of Quantum Information (2008).

[39] Warmuth, M., Kuzmin, D.: Online variance minimization. In: Proceedings of the 19th Annual Conference on Learning Theory, volume 4005 of Lecture Notes in Computer Science, pages 514–528, Springer (2006).

[40] Wu, X.: Equilibrium value method for the proof of QIP=PSPACE. arXiv:1004.0264v4 (2010).

Appendix A. Appendix

Appendix A.1. Semidefinite program that computes dis(\vec{p})

In this section, we provide a simple SDP that computes dis(\vec{p}).

$$\min \sum_{i=1}^{M} t_i$$ \hspace{1cm} (A.1)

subject to: \[
\begin{pmatrix}
 t_i & \vec{p}_i - \vec{q}(\rho)_i \\
\vec{p}_i^* - \vec{q}(\rho)^*_i & t_i \\
\end{pmatrix} \succeq 0, \quad \forall i = 1, \ldots, M. \hspace{1cm} (A.2)
\]

$$\rho \in \mathcal{D}(\mathcal{H}). \hspace{1cm} (A.3)$$

The correctness of the SDP comes from that the positive semidefinite constraint in Eq. 1.2 implies \( t_i^2 \geq \| \vec{p}_i - \vec{q}(\rho)_i \|^2 \) for each \( i=1,\ldots,M \). For any fixed \( \rho \), the minimization over \( t_i \)s will give \( \| \vec{p} - \vec{q}(\rho) \|_1 \). And then the minimization over \( \rho \) gives the desired answer.

Appendix A.2. Proof of Lemma 1

**Theorem 6** (Multiplicative weights update method—see Ref. [28, Theorem 10]). Fix \( \gamma \in (0, 1/2) \). Let \( N^{(1)}, \ldots, N^{(T)} \) be arbitrary \( d \times d \) “loss” matrices with \( 0 \preceq N^{(t)} \preceq \alpha I \). Let \( W^{(1)}, \ldots, W^{(T)} \) be \( d \times d \) “weight” matrices given by

\[ W^{(1)} = I \] \[ W^{(t+1)} = \exp(-\gamma(N^{(1)} + \cdots + N^{(t)})) \].

17
1. Let $\gamma = \frac{\epsilon}{8Mw}$ and $T = \left\lceil \frac{\ln d}{\gamma^2} \right\rceil$. Also let $W^{(1)} = I_{\mathcal{X}}$, $d = \dim(\mathcal{X})$.
2. Repeat for each $t = 1, \ldots, T$:
   
   (a) Let $\rho^{(t)} = W^{(t)} / \text{Tr} W^{(t)}$ and compute $\bar{q}(\rho^{(t)})$. One can then rewrite the vector $\bar{p} - \bar{q}(\rho^{(t)})$ in the polar form $(c_1^{(t)} e^{i\phi_1^{(t)}}, c_2^{(t)} e^{i\phi_2^{(t)}}, \ldots, c_M^{(t)} e^{i\phi_M^{(t)}})$ and choose $\bar{z}^{(t)} = (e^{-i\phi_1^{(t)}}, e^{-i\phi_2^{(t)}}, \ldots, e^{-i\phi_M^{(t)}})$. It is not hard to see such $\bar{z}^{(t)}$ maximizes $\text{Re} \left\langle \bar{p} - \bar{q}(\rho^{(t)}), \bar{z} \right\rangle$.
   
   (b) Choose $N^{(t)}$ to be
   
   $$N^{(t)} = \text{Re} \left\langle \bar{p}, \bar{z}^{(t)} \right\rangle I_{\mathcal{X}} - \frac{1}{2}(Q^{(t)} + Q^{(t)\ast}) + 2MwI_{\mathcal{X}},$$

   where $Q^{(t)} = \sum_{i=1}^{M} e^{i\phi_i^{(t)} Q_i}$.
   
   (c) Update the weight matrix as follows: $W^{(t+1)} = \exp(-\gamma \sum_{t=1}^{T} N^{(t)})$.
3. Return $\bar{d} = \frac{1}{T} \sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} - 2MwI_{\mathcal{X}} \right\rangle$.

Figure A.3: An algorithm that approximates the $d(\bar{p})$ with additive error $\epsilon$.

Let $\rho^{(1)}, \ldots, \rho^{(T)}$ be density operators obtained by normalizing each $W^{(1)}, \ldots, W^{(T)}$ so that $\rho^{(t)} = W^{(t)} / \text{Tr} W^{(t)}$. For all density operators $\rho$ it holds that

$$\frac{1}{T} \sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} \right\rangle \leq \left\langle \rho, \frac{1}{T} \sum_{t=1}^{T} N^{(t)} \right\rangle + \alpha(\gamma + \frac{\ln d}{\gamma^2}).$$

Note Theorem 6 holds for all choices of loss matrices $N^{(1)}, \ldots, N^{(T)}$, including those for which each $N^{(t)}$ is chosen adversarially based upon $W^{(1)}, \ldots, W^{(T)}$. This adaptive selection of loss matrices is typical in implementations of the MMW. Consider the algorithm shown in Fig. A.3.

**Proof.** The algorithm is a typical application of the matrix multiplicative weight update method. In order to make use of Theorem 6, we need first to show $N^{(t)}$ is bounded for each $t$. Since $\bar{p} \in \text{Raw-} (M, w)$ and $\|\bar{z}^{(t)}\|_{\infty} \leq 1$, by Cauchy-Schwartz inequality we have

$$|\text{Re} \left\langle \bar{p}, \bar{z}^{(t)} \right\rangle| \leq \|\bar{p}\|_1 \|\bar{z}^{(t)}\|_{\infty} \leq M\|\bar{p}\|_{\infty} \|\bar{z}^{(t)}\|_{\infty} = Mw.$$

Furthermore we have $\|Q\|_{op} = \|\sum_{i=1}^{M} e^{-i\phi_i^{(t)} Q_i}\|_{op} \leq \sum_{i=1}^{M} \|Q_i\|_{op} \leq Mw$. Thus by triangle inequality, one can easily find $0 \leq N^{(t)} \leq 4MwI_{\mathcal{X}}$. Then we can make use of Theorem 6. Immediately, for any $\rho \in D(\mathcal{X})$, we have

$$\frac{1}{T} \sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} \right\rangle \leq \left\langle \rho, \frac{1}{T} \sum_{t=1}^{T} N^{(t)} \right\rangle + \alpha(\gamma + \frac{\ln d}{\gamma^2}).$$

Substitute $\alpha = 4Mw, \gamma = \frac{\epsilon}{8Mw}$ and $T = \left\lceil \frac{\ln d}{\gamma^2} \right\rceil$. Also consider the identity $\left\langle \rho^{(t)}, N^{(t)} - 2MwI_{\mathcal{X}} \right\rangle =$
\[ \text{Re}\left( \vec{p} - \vec{q}(\rho(t)), \vec{z}(t) \right). \] Then we have for any \( \rho \in D(\mathcal{X}), \]
\[ \tilde{d} = \frac{1}{T} \sum_{t=1}^{T} \left\langle \rho^{(t)}, N^{(t)} - 2MwI_{\mathcal{X}} \right\rangle \leq \text{Re}\left( \vec{p} - \vec{q}(\rho), \frac{1}{T} \sum_{t=1}^{T} \vec{z}(t) \right) + \epsilon. \quad (A.4) \]

Consider the equilibrium value form of \( \text{dis}(\vec{p}) \) in Equ. (1). For each \( \rho^{(t)} \), we always find the \( \vec{z}(t) \) that maximizes \( \text{Re}\left( \vec{p} - \vec{q}(\rho^{(t)}), \vec{z} \right) \). Hence, \( \text{dis}(\vec{p}) \leq \tilde{d} \). Let \( \rho^* \) be any equilibrium point of the equilibrium value in Equ. (1). By substituting such \( \rho^* \) into Equ. (A.4) we have
\[ \tilde{d} \leq \text{Re}\left( \vec{p} - \vec{q}(\rho^*), \frac{1}{T} \sum_{t=1}^{T} \vec{z}(t) \right) + \epsilon \leq \text{dis}(\vec{p}) + \epsilon. \]

So far we complete the proof of the correctness of this algorithm. Note that each step in the algorithm only contains fundamental operations of matrices and vectors, which can be done in polynomial time in \( M, d \). Also there are totally \( O(T) = \text{poly}(\ln d, M, w, 1/\epsilon) \) steps, thus the whole algorithm can be executed in \( \text{poly}(d, M, w, 1/\epsilon) \) time. Moreover, given the fact that fundamental operations of matrices and vectors also admit efficient algorithms in parallel (i.e., NC algorithm), one can easily compose these NC circuits of each step and obtain a NC algorithm as a whole if the total number of steps \( T \) is not too large. Precisely, if \( M, w, 1/\epsilon \in O(\text{poly-log}(d)) \) and \( \tilde{d} \) is considered as the input size, this algorithm is also efficient in parallel. □