A Direct Proof of $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$

by

Alexandre Laplante

A research paper
presented to the University of Waterloo
in fulfillment of the
research paper requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2013

© Alexandre Laplante 2013

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

For s and c such that $\frac{1}{2} \leq s < c \leq 1$, and the gap between s and c being at most the inverse of a polynomial in the size of the input, the complexity class containment $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1] \subseteq \mathsf{PSPACE}$ is known through the chain of containments $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{QIP}(2) \subseteq \mathsf{PSPACE}$. The main result of this paper is a PSPACE algorithm based on the "Matrix Multiplicative Weights Update" (MMWU) method, which is used to solve a certain exponentially large Semidefinite Program (SDP) whose optimal value is the acceptance bias of an $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol, thus giving a direct, and self-contained proof that $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$.

We observe that the complexity class $\oplus\mathsf{MIP}^*[2,1]$ can have its maximum acceptance probability characterized by an SDP in super-operator form involving only positive semidefinite matrices. However, due to the difference between the super-operator form and the standard inequality form of SDPs, the recent parallel algorithms of Jain and Yao [13] cannot easily be used to show containment in PSPACE.

Contents

1 Introduction

The complexity class $\oplus\mathsf{MIP}^*[2,1]$ was first defined by Cleve, Høyer, Toner, and Watrous. [6] Informally, it is the class of problems that can be decided by a polynomial time verifier performing an interactive protocol with two computationally unbounded, non-communicating provers who may share entanglement. The protocol has the further restriction that there is only one round of interaction with the provers, and they respond with exactly one bit each, and the verifier receives only the XOR of the prover's bits. There is a result due to Håstad that for all $\varepsilon \in \left(0, \frac{1}{16}\right)$, if $s = \frac{11}{16} + \varepsilon$ and $c = \frac{12}{16}$ then $\oplus\mathsf{MIP}_{\mathsf{c,s}}[2,1] = \mathsf{NEXP}$ [1, 9]. Here, $\oplus\mathsf{MIP}_{\mathsf{c,s}}[2,1]$ is defined in the same way as $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$, except the provers do not share any entanglement.

Cleve et al. formulated the acceptance probability of $\oplus\mathsf{MIP}^*[2,1]$ as an exponential sized SDP, as we will do in Section 3.2. They conclude that $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{EXP}$ because of the existence of efficient algorithms for solving SDPs (See [4] for a reference on SDPs). This tells us something about the power of entanglement. The $\oplus\mathsf{MIP}[2,1]$ protocol seems to lose its power when the provers are allowed to share entanglement. The question of exactly where $\oplus\mathsf{MIP}^*[2,1]$ fits in with known complexity classes is still open, but some advances have been made.

Cleve, Gavinski, and Jain have also shown [5] that for $\varepsilon \in \left(0, \frac{1}{4}\right)$, and with $c = 1 - \varepsilon$, $s = \frac{1}{2} + \varepsilon$, that $\mathsf{NP} \subseteq \oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$.

The containment in PSPACE comes from the combination of two results. Wehner showed that $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{QIP}(2)$ [21], while Jain, Upadhyay, and Watrous. showed that $\mathsf{QIP}(2) \subseteq \mathsf{PSPACE}$ [11]. Putting these two proofs together gives us the containment $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$.

Jain et al. formulate the maximum acceptance probability of a $\mathsf{QIP}(2)$ protocol as an exponentially sized SDP, and present an efficient parallel algorithm based on the "Matrix Multiplicative Weights Update" method to approximate its value in PSPACE.

In this paper, we present a direct proof of the containment $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$, where the maximum acceptance probability of an $\oplus\mathsf{MIP}^*[2,1]$ is first written as an exponentially large SDP, and then solved using an MMWU algorithm based on the algorithm from Jain et al. In order to write the probability as an SDP, our method requires an amount of preprocessing which can be done in PSPACE. Note that since this algorithm runs in PSPACE, we can't store an explicit description of the SDP without taking too much space. The SDP is implicitly defined by the protocol, and any bit of it can be accessed using a polynomial amount of space.

The $\mathsf{QIP}(2)$ class is potentially much larger than $\oplus\mathsf{MIP}^*[2,1]$, so a direct proof

of the containment $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$ could potentially be a lot more simple, giving a better understanding of the containment, and perhaps leading to a tighter bound. This was the motivation behind this work.

The MMWU technique is presented in detail in Satyen Kale's PhD Thesis [15]. It has the property that it is "width"-bounded. That is to say, there is some parameter, which Kale et al. have called the width, which must be small enough in order for the algorithm to run efficiently. The MMWU technique is not an algorithm itself, but a meta-algorithm, or a technique for creating algorithms. Because of this, the width parameter varies from algorithm to algorithm and it must be independently argued for each algorithm, and the class of SDPs which it solves, that the width is low enough to be efficient.

Briefly, the MMWU technique works by binary searching over guesses for the objective value of the SDP. In a given step of the binary search, suppose our guess for the optimal solution is α. We start with $\rho_0 = \frac{\mathbb{I}}{\mathrm{Tr}(\mathbb{I})}$, a scaled identity matrix, as a guess for the optimal solution. The algorithm proceeds in iterations, where at each iteration t, ρ_t is based on ρ_{t-1} in a way that improves its feasibility. If we complete T iterations where T is an appropriately defined amount, we can prove that ρ_T is approximately feasible with objective value α, and so the true optimal value is at least α. If at any point we fail to find an improved ρ_t, we can conclude that the true optimal value is at most α.

The MMWU technique has seen lot of use in quantum complexity theory. It has been used to show that $\mathsf{QRG}(1) \subseteq \mathsf{PSPACE}$ [12], $\mathsf{QIP}(2) \subseteq \mathsf{PSPACE}$ [11], $\mathsf{QIP} = \mathsf{PSPACE}$ [10] and $\mathsf{QRG}(2) = \mathsf{PSPACE}$ [8], in that order, over the course of 5 years.

In all of the above papers, the authors first characterize the given quantum complexity class as an SDP, and then use a MMWU algorithm to solve that SDP in PSPACE. These proofs all used the MMWU technique, meaning that they had to independently show that the "width" parameter in the problem that they were considering was low enough to show containment in PSPACE.

Recently, Jain and Yao [13, 14] as well as Peng and Tangwonsan [17] have given algorithms which can solve exponentially large SDPs in PSPACE which are not based on the MMWU method and are independent of any width parameter. The class of SDPs to which these algorithms apply are "mixed packing and covering SDPs", of which "positive SDPs" are a special case. For a definition of positive SDPs, see Section 2.4. Jain and Yao claim that their algorithm can be directly applied to the SDP formulations of $\mathsf{QIP}(2)$ and $\mathsf{QRG}(1)$ to show containment in PSPACE. [13] However, this relies on an assumption that there is a simple transformation from the super-operator form of the SDP formulations of $\mathsf{QIP}(2)$ and $\mathsf{QRG}(1)$, and the standard inequality form of SDPs used by Jain and Yao. We will show that the assumed simple transformation does not exist, so if the algorithm

does apply to these SDPs, it does not apply directly.

The motivation behind this work was a better understanding of SDP approximating algorithms, with the hope of possibly finding a tighter bound than PSPACE. It seems unlikely that using the techniques in this paper, specifically the MMWU, that anything tighter than PSPACE can be shown. The algorithms depend on matrix operations such as matrix multiplication and exponentiation. These operations, when performed on exponential sized inputs, belong to the complexity class NC (poly) which is equal to PSPACE. In the case of matrix exponentiation, the operation can only be approximated in NC (poly). Therefore, it seems unlikely that any refinement of these techniques could yield a tighter bound than PSPACE.

Rahul Jain, John Watrous, and Sarvagya Upadhyay discovered the more direct proof of $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$ while working on their proof of $\mathsf{QIP}(2) \subseteq \mathsf{PSPACE}$ [11]. Their algorithm was unpublished because the result is also implied by the chain of containments $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{QIP}(2) \subseteq \mathsf{PSPACE}$. The proof was partially rediscovered by the author of this paper.

2 Mathematical Preliminaries

2.1 Linear Algebra

Definition. If $\mathcal{X} = \mathbb{C}^N$ is a complex vector space, we denote the set of linear operators (matrices) on $\mathcal{X}$ by $\mathrm{L}(\mathcal{X}) = \mathbb{C}^{N \times N} = \mathbb{M}_{N \times N}(\mathbb{C})$.

Definition. We use the notation $\lambda(X)$ to denote the vector of eigenvalues of X, sorted by size such that $\lambda_1(X) \leq \lambda_2(X) \leq \cdots \leq \lambda_N(X)$.

Definition. The *trace* of a matrix $A \in \mathrm{L}(\mathcal{X})$ is defined as $\mathrm{Tr}(A) = \sum_i A_{ii}$. Note also that $\mathrm{Tr}(A) = \sum_i \lambda_i(A)$. We will use $\langle A, B \rangle$ to mean $\mathrm{Tr}(AB)$, and call this operation the *inner product*. Notice that the inner product is bilinear and symmetric. Note also that $\mathrm{Tr}(AB) = \sum_{ij} A_{ij} B_{ij}$. If $A \in \mathrm{L}(\mathcal{X})$ and $B \in \mathrm{L}(\mathcal{Y})$, the partial trace with respect to $\mathcal{X}$ is defined as the unique linear operator $\mathrm{Tr}_{\mathcal{Y}} : \mathrm{L}(\mathcal{X} \otimes \mathcal{Y}) \to \mathrm{L}(\mathcal{X})$ such that $\mathrm{Tr}_{\mathcal{Y}}(A \otimes B) = \mathrm{Tr}(A)\, B$.

Definition. The *conjugate transpose*, or *adjoint* of a matrix A is denoted A^* and it means taking the complex conjugate of all entries of A^{T}, which is the transpose of A.

Definition. A *Unitary matrix* is a matrix U such that $U^*U = UU^* = \mathbb{I}$. Closed quantum systems evolve under unitary transformations.

Definition. We will use the notation E_{ij} to denote the matrix with a 1 in position i, j and 0 elsewhere, the dimension of E_{ij} depending on the context.

Definition. A *Hermitian matrix* is a matrix H such that $H = H^*$. Hermitian matrices are diagonalizable by unitary matrices, in other words, if H is Hermitian, then $H = UDU^*$ for some diagonal matrix D and unitary matrix U.

Definition. A Hermitian matrix is called *positive semidefinite* if all of its eigenvalues are non-negative. It is called *positive definite* if all of its eigenvalues are positive. We write $X \succeq 0$ to mean X is positive semidefinite, $X \succ 0$ to mean X is positive definite, and $X \succeq Y$ to mean $X - Y \succeq 0$.

Some properties of positive (semi)definite matrices:

1. X is positive semidefinite if and only if X is the Gram matrix of some set of vectors, i.e. $X_{ij} = \langle x_i, x_j \rangle$ for some set of vectors $\{x_i\}$.
2. X is positive semidefinite if and only if $\langle X, Y \rangle \geq 0$ for all Y positive semidefinite.
3. X is positive definite if and only if it is positive semidefinite and invertible.

Lemma 1. *The space $\mathbb{C}^N \times \mathbb{C}^N$ has a basis of positive semidefinite matrices.*

Proof. For example, consider the matrices

$$B_{ij} = \begin{cases} E_{ii} & \text{if } i = j \\ E_{ii} + E_{jj} + E_{ij} + E_{ji} & \text{if } i < j \\ E_{ii} + E_{jj} + iE_{ij} - iE_{ji} & \text{if } i > j \end{cases},$$

then the set $\{B_{ij}\}_{1 \leq i,j \leq N}$ is a basis of $\mathbb{M}_{N \times N}(\mathbb{C})$ consisting of only positive semidefinite matrices. □

Lemma 2. *The space $\mathbb{M}_{N \times N}(\mathbb{C})$ does not have an orthogonal basis of positive semidefinite matrices.*

Proof. A basis $\{Z_i\}_{1 \leq i \leq N}$ is orthogonal if and only if $\forall_{i \neq j} \ \langle Z_i, Z_j \rangle = 0$.

Every non-zero positive semidefinite matrix has a positive eigenvalue, and hence a positive trace. So every non-zero positive semidefinite matrix has a non-zero diagonal element. Since there are N diagonal elements, there are at most N orthogonal positive semidefinite matrices. However, a basis of $\mathbb{M}_{N \times N}(\mathbb{C})$ must contain N^2 matrices. □

Definition. We can extend the domain of functions $f : \mathbb{C} \to \mathbb{C}$ to diagonal matrices with the definition $f \begin{pmatrix} D_{11} & 0 & \cdots & 0 \\ 0 & D_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & D_{nn} \end{pmatrix} = \begin{pmatrix} f(D_{11}) & 0 & \cdots & 0 \\ 0 & f(D_{22}) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & f(D_{nn}) \end{pmatrix}$.
We further extend this definition to all diagonalizable matrices A with the rule that $f(A) = U f(D) U^*$. We use this definition for the matrix exponential, and matrix square roots.

Definition. The *trace norm* $\|\cdot\|_1$ is the Schatten p-norm with $p = 1$, and for any matrix A, $\|A\|_1 = \mathrm{Tr}\left(\sqrt{A^* A}\right)$, or equivalently, the sum of its singular values. For a positive semidefinite matrix, its trace norm is equal to its trace.

Definition. The *infinity norm* $\|\cdot\|_\infty$ is the Schatten p-norm as $p \to \infty$, and for $A \succeq 0$, $\|A\|_\infty$ corresponds to the largest eigenvalue of A, $\lambda_N(A)$.

Lemma 3. For all Hermitian A, B, the following special case of of Hölder's inequality holds,

$$\langle A, B \rangle \leq \|A\|_\infty \|B\|_1 .$$

Lemma 4. *[11] For $R_0, R_1 \succeq 0$,*

$$\|R_0 - R_1\|_1 \leq \sqrt{2\,\mathrm{Tr}(R_0)^2 + 2\,\mathrm{Tr}(R_1)^2 - 4\mathrm{F}(R_0, R_1)^2},$$

where F is the *fidelity* function, defined as $\mathrm{F}(A, B) = \left\| \sqrt{A}\sqrt{B} \right\|_1$.

Definition. The ith *Gershgorin disk* $D_i(A)$ of a matrix $A \in \mathbb{M}_{N\times N}(\mathbb{C})$ is a ball in the complex plane centered at A_{ii} with radius $\sum_{j\neq i} |A_{ij}|$.

Theorem. *(Gershgorin disk theorem) [7] The eigenvalues of A lie within $\bigcup_{i=0}^{n} D_i(A)$.*

Corollary. *If A is Hermitian and all of its Gershgorin disks lie in the positive half of the complex plane, then $A \succ 0$.*

2.2 Quantum Information

In quantum information, an n-qubit *register* is a physical system whose state is described as a vector in $\mathbb{C}^{2^n}$. When we say a register corresponds to space $\mathcal{X}$, we mean $\mathcal{X} = \mathbb{C}^{2^n}$ for a register of size n.

The *state* of an n-qubit register corresponding to the space $\mathcal{X}$ is a density matrix $\rho \in \mathrm{L}(\mathcal{X}) = \mathbb{C}^{2^n} \times \mathbb{C}^{2^n}$. If $\rho^2 = \rho$, or equivalently, if ρ has rank 1, then we can write $\rho = |\psi\rangle\langle\psi|$, or simply $|\psi\rangle$ for some unit vector $|\psi\rangle \in \mathcal{X} = \mathbb{C}^{2^n}$. We call such a $|\psi\rangle$ a *pure state*. If we have two pure states on different registers $\rho_0 \in \mathrm{L}(\mathcal{X})$, $\rho_1 \in \mathrm{L}(\mathcal{Y})$, then their combined state, treating the two registers as one big register, is $\rho_0 \otimes \rho_1 \in \mathrm{L}(\mathcal{X} \otimes \mathcal{Y})$, where $\otimes$ denotes the Kronecker product. If the combined state of two registers $\mathcal{X}$ and $\mathcal{Y}$ is $\rho \in \mathrm{L}(\mathcal{X} \otimes \mathcal{Y})$, the *reduced state* on $\mathcal{X}$ is $\rho_{\mathcal{X}} = \mathrm{Tr}_{\mathcal{Y}}(\rho)$.

A *measurement* on a space $\mathcal{X}$ is described by a sequence $\{P_i \in L(\mathcal{X})\}$ of positive semidefinite matrices such that $\sum_i P_i = \mathbb{I}$. The matrix P_i in the sequence corresponds to the measurement outcome i, and the probability of observing the outcome corresponding to P_i when measuring the state $\rho \in \mathrm{L}(\mathcal{X})$ is $\mathrm{Tr}(P_i\rho)$. If all of $P_i^2 = P_i$ this is called a *projective measurement*, furthermore, if for the outcomes of each P_i we associate a real number λ_i, then $A = \sum_i \lambda_i P_i$ is the *observable* corresponding to this measurement.

For a more complete understanding of the formalism of quantum information we refer the reader to a textbook by Nielsen and Chuang. [16]

2.3 Definitions of $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ and $\oplus\mathsf{MIP}^*[2,1]$

We begin by formally defining the complexity class $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$. Consider the following interactive protocol.

Two *provers* and a *verifier* receive input x. The verifier has three functions, $\sigma_x : \left[2^R\right] \to S$, $\tau_x : \left[2^R\right] \to T$ and $f_x : \left[2^R\right] \to \{0,1\}$, where $R \in \text{poly}(|x|)$, $|S|, |T| \in \mathcal{O}\left(2^{\text{poly}(|x|)}\right)$, and (σ_x, τ_x, f_x) depend on x in a polynomial-time uniform way, and are computable in time polynomial in $|x|$. The prover generates a secret string $r \in \left[2^R\right]$ using R uniformly random bits and computes $\sigma_x(r) = s$ and $\tau_x(r) = t$. The values s and t are called the *questions* to the provers, and the verifier sends one question to each prover. The provers are computationally unbounded but obey the laws of quantum physics. They do not communicate but they may share an entangled quantum state $|\Phi\rangle$. After receiving their input bits, the provers measure their shared quantum state with a measurement possibly depending on their input, in order to correlate their responses. The provers respond with one bit each, a and b. The verifier receives $a \oplus b$, the XOR of the two response bits, and accepts if and only if $a \oplus b = f_x(r)$. Note that with this definition the provers can always achieve a 50% probability of acceptance. We do not allow post-processing of the type: "always reject 75% of the time".

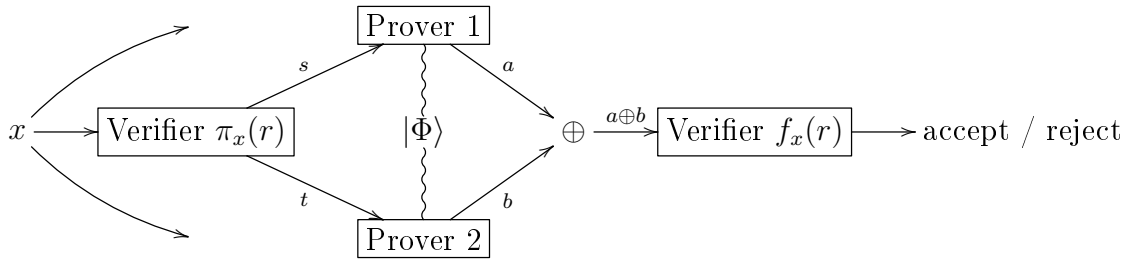

Figure 1: An illustration of an arbitrary $\oplus\mathsf{MIP}^*[2,1]$ protocol.

Suppose the provers share the state $|\Phi\rangle\langle\Phi| \in \mathrm{L}(\mathcal{X} \otimes \mathcal{Y})$. Let $\Pi^1_x : S \to \mathrm{L}(\mathcal{X})$, $\Pi^2_x : T \to \mathrm{L}(\mathcal{Y})$ be the strategies of the provers. I.e. when given x and $s \in S$, $t \in T$, the provers perform the measurements corresponding to $\{\Pi^1_x(s), \mathbb{I} - \Pi^1_x(s)\}$ and $\{\Pi^2_x(t), \mathbb{I} - \Pi^2_x(t)\}$, respectively. The complexity class $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ is defined as the set of languages L for which there exists a uniform family of polynomial time constructible pairs $(\pi_x,\ f_x)$ such that after running a protocol of this form, we have

$$x \in L \Rightarrow \exists \Pi^1_x, \Pi^2_x,\ \Pr\left[\text{Verifier Accepts}\right] \geq c$$
$$x \notin L \Rightarrow \forall \Pi^1_x, \Pi^2_x,\ \Pr\left[\text{Verifier Accepts}\right] \leq s.$$

The values c and s are called the *soundness* and *completeness* of the protocol, respectively.

Suppose, for example, the verifier is trying to decide if a formula is satisfiable. She may expect the provers to provide a proof of satisfiability. In the case that $x \in L$, this is certainly a strategy that will allow her to achieve a high completeness. However, if $x \notin L$ then the provers need not "stick to the script", and they may behave arbitrarily in order to cause the verifier to accept anyway.

$\oplus\mathsf{MIP}^*[2,1]$ is the union of all $\oplus\mathsf{MIP}^*_{\mathsf{c},\mathsf{s}}[2,1]$ for which $\frac{1}{2} \leq s < c \leq 1$, and $c - s \in \Omega\Big(\frac{1}{\mathrm{poly}(|x|)}\Big)$. This is intuitively the union of all reasonable $\oplus\mathsf{MIP}^*_{\mathsf{c},\mathsf{s}}[2,1]$ protocols.

2.4 Semidefinite Programming

For $\mathcal{X} = \mathbb{C}^N$, $\mathcal{Y} = \mathbb{C}^M$, A semidefinite program (SDP) in the *super-operator form* is an optimization problem of the form

	Primal		Dual
max	$\mathrm{Tr}(AX)$	min	$\mathrm{Tr}(BY)$
s.t.	$\Phi(X) \preceq B$	s.t.	$\Phi^*(Y) \succeq A$
	$X \succeq 0$, $X \in \mathrm{L}(\mathcal{X})$		$Y \succeq 0$, $Y \in \mathrm{L}(\mathcal{X})$

where where $A \in \mathrm{L}(\mathcal{X})$, $B \in \mathrm{L}(\mathcal{Y})$ are Hermitian matrices and $\Phi : \mathrm{L}(X) \to \mathrm{L}(Y)$ is a linear map which maps Hermitian operators to Hermitian operators. With X, Y positive semidefinite and A, B Hermitian, the objective values are real, so the maximization and minimizations make sense. Also note that this formulation is equivalent to the standard formulation in the literature. [20]

If both the primal and the dual have Slater points, that is, there exists $X \succ 0$ such that $\Phi(X) \preceq B$ and there exists $Y \succ 0$ such that $\Phi^*(Y) \succeq A$, then the strongest form of a property called *strong duality* holds. When this property holds, the optimal value of the primal is equal to the optimal value of the dual, and both are achieved. All of the SDPs which we consider in this paper admit Slater points in their primal and dual.

In general, an SDP which admits Slater points can have its optimal value approximated to an additive factor ε in time polynomial in the dimensions of the matrices involved, using standard convex optimization techniques e.g. the ellipsoid method. [4] The SDPs with which we will concern ourselves have dimensions exponential in the size of our input, and so these standard techniques do not suffice to show containment in PSPACE, becuase, for instance, they require storing large matrices explicitly. Explicitly storing any of our matrices would require more than a polynomial amount of space, so in what follows matrices are not stored explicitly, but any of their entries can be accessed using polynomial space.

Definition. A *positive SDP* is an SDP in the *standard inequality form*

Primal

$$\begin{array}{lll} \min & & \mathrm{Tr}(CY) \\ \text{s.t.} & \forall_{i\in[N]} & \mathrm{Tr}(A_i Y) \geq b_i \\ & & Y \succeq 0 \end{array}$$

Dual

$$\begin{array}{lll} \max & & \sum_{i\in[M]} b_i x_i \\ \text{s.t.} & & \sum_{i\in[M]} x_i A_i \preceq C \\ & \forall_{i\in[M]} & x_i \geq 0 \end{array}$$

where for all i, $A_i, C \succeq 0$, $b_i \geq 0$, and $[N] = \{1, \ldots N\}$, $[M] = \{1, \ldots M\}$.

Jain and Yao [13, 14], and Peng and Tangwonsan [17] have shown that positive semidefinite programs of this form can be approximated to a multiplicative factor $(1+\varepsilon)$ in parallel time

$$\log^c(M,N)\,\frac{1}{\varepsilon^4}\log\left(\frac{1}{\varepsilon}\right),$$

for some constant c, with M, N as in the definition of the SDP. That is, if the true optimal objective value of the SDP is α, these algorithms will output α' such that $\alpha \leq \alpha' \leq (1+\varepsilon)\,\alpha$. They can thus be parallelized and implemented as a poly-logarithmic depth circuit.

If we have a positive SDP with $M = N = \mathcal{O}(2^n)$ for some $n \in \mathcal{O}(|x|^c)$ and some constant c, we call this an *exponential size positive SDP*.

If the accuracy ε required is $\Omega\left(\frac{1}{\mathrm{poly}(n,m)}\right)$, exponential size positive SDPs can be solved in parallel time

$$\log^c(2^n)\,\frac{1}{\varepsilon^4}\log\left(\frac{1}{\varepsilon}\right) \in \mathcal{O}(n^c)\,,$$

for some c. In other words, they can be parallelized and implemented as a polynomial depth circuit. The complexity class of languages which can be decided by polynomial depth circuits is called $\mathsf{NC}\,(\mathrm{poly})$, and it is known that $\mathsf{NC}\,(\mathrm{poly}) = \mathsf{PSPACE}$. [3]

It is a common mistake that SDPs in the standard inequality form and the super-operator form are equivalent. Most properties are shared between them because it is simple to convert from one form to the other. However, it is not trivial to convert an SDP in super-operator form with all positive semidefinite matrices into a positive-SDP in standard inequality form. In Jain and Yao [13], it was assumed that these forms were equivalent and that their algorithm for solving positive-SDPs in standard inequality form thus also solved the super-operator form SDP formulation of the complexity class $\mathsf{QIP}(2)$, and others. However, the transformation from super-operator form to standard inequality form depended on the hidden assumption of the existence of an orthogonal positive semidefinite basis. Lemma 2 proves that such a basis does not exist, and that there is therefore no straightforward way to convert one SDP form to another, while preserving positivity.

3 Positive SDP characterization of $\oplus\mathsf{MIP}^*[2,1]$

3.1 Overview

We will follow Cleve et al. [6] in defining $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ as an SDP. We will then see how to write this SDP in the form of a positive SDP as defined in Section 2.4. From this, we will be able to conclude that $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$.

3.2 SDP formulation of $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$

In an $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol, the provers are computationally unbounded, but since each one only has one of (s,t) and part of $|\Phi\rangle$, we can reduce each prover's computation to a measurement on $|\Phi\rangle$ depending on x and one of (s,t).

Let $|\Phi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ be the state which is shared between the provers, where $\mathcal{X}$ is the space of prover 1, and $\mathcal{Y}$ is the state of prover 2. Let $\{\Pi^1_x(s), \mathbb{I} - \Pi^1_x(s)\}$ be the measurement used by the first prover on her space $\mathcal{X}$, with $\Pi^1_x(s)$ corresponding to the output $a = 0$ and $\mathbb{I} - \Pi^1_x(s)$ corresponding to output $a = 1$. Define $\{\Pi^2_x(t), \mathbb{I} - \Pi^2_x(t)\}$ similarly for b. Note that in what follows, the dependence on x is no longer explicitly mentioned to simplify notation.

We can define observables $A_s = 2\Pi^1_x(s) - \mathbb{I}$ and $B_t = 2\Pi^2_x(t) - \mathbb{I}$ with outcomes $(-1)^a$ and $(-1)^b$, respectively.

Then,

$$(-1)^{a\oplus b} \Pr[a,b|s,t] = \langle\Phi| A_s \otimes B_t |\Phi\rangle .$$

Furthermore,

$$2\Pr[\text{Verifier Accepts } x] - 1 = \sum_r \frac{1}{2^R} (-1)^{f_x(r)} \langle\Phi| A_{\sigma(r)} \otimes B_{\tau(r)} |\Phi\rangle . \tag{1}$$

Where the verifier uses R uniformly random bits in the random seed r, while $\sigma(r)$ and $\tau(r)$ are the values of s and t respectively, given random seed r.

If L is a language in $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$, then the quantity $2\Pr[\text{Verifier Accepts } x] - 1$ is the bias in the probability of acceptance, and it is $\geq 2c - 1$ when $x \in L$ and $\leq 2s - 1$ when $x \notin L$.

Tsirelson proved[18] that given observables A_s on $\mathcal{X}$ and B_t on $\mathcal{Y}$, there exists unit vectors $u_s, v_t \in \mathbb{R}^N$ where $N = \min(|S|, |T|)$, such that for all $s \in S$, $t \in T$, $|\Phi\rangle \in \mathcal{X} \otimes \mathcal{Y}$, we have

$$\langle u_s, v_t\rangle = \langle\Phi| A_s \otimes B_t |\Phi\rangle .$$

Conversely, given $u_s, v_t \in \mathbb{R}^N$, $\dim(\mathcal{X}) = \dim(\mathcal{Y}) = 2^{\lceil N/2 \rceil}$, then for $|\Phi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ a maximally entangled state, there exist observables A_s on $\mathcal{X}$, B_t on $\mathcal{Y}$,

$$\langle u_s,\, v_t \rangle = \langle \Phi |\, A_s \otimes B_t\, | \Phi \rangle\,.$$

This lets us simplify Eq. (1) as

$$\text{bias} = \sum_r \frac{1}{2^R} (-1)^{f_x(r)} \left\langle u_{\sigma(r)},\, v_{\tau(r)} \right\rangle. \tag{2}$$

We now formulate the bias of an $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol as a semidefinite program. The highest bias that the provers can achieve is given by the maximization problem

$$\max_{\{u_s, v_t\}} \sum_r \frac{1}{2^R} (-1)^{f_x(r)} \left\langle u_{\sigma(r)},\, v_{\tau(r)} \right\rangle, \tag{3}$$

such that $u_s, v_t \in \mathbb{R}^N$, $\|v_t\| = \|u_S\| = 1$.

For notation, let us assume that $s \in S$ are numbers from 1 to $|S|$, and that $t \in T$ are numbers from $|S|+1$ to $|S|+|T|$. Consider the matrix G which is the Gram matrix of the set of vectors $u_s, v_t \in \mathbb{R}^N$. In other words, for any $s_1, s_2 \in S$ and $t_1, t_2 \in T$, we have $G_{s_1 t_1} = \langle u_{s_1},\, v_{t_1} \rangle$, $G_{s_1 s_2} = \langle u_{s_1},\, u_{s_2} \rangle$, and $G_{t_1 t_2} = \langle v_{t_1},\, v_{t_2} \rangle$ with G being of dimension $|S| + |T|$.

Let the matrix P be such that

$$P_{s,t} = \sum_r g_{s,t}(r)\,, \text{ where } g_{s,t}(r) = \begin{cases} \frac{1}{2^R} (-1)^{f_x(r)} & \text{if } (\sigma(r)\,, \tau(r)) = (s,t) \\ 0 & \text{otherwise.} \end{cases}$$

In other words, $P_{s,t}$ is the the expected value of $(-1)^{f_x(r)}$, weighted by the probability of (s,t). Note that $P_{ab} = 0$ when both $a, b \in S$, or both $a, b \in T$.

Now we may rewrite Eq. (3) as the following maximization problem over matrix variable G of dimension $N = |S| + |T|$.

$$\begin{aligned} \max \quad & \operatorname{Tr}(PG) \\ \text{s.t.} \quad \forall i \in [N] \quad & \operatorname{Tr}(E_{ii} G) = 1 \\ & G \succeq 0. \end{aligned} \tag{4}$$

The condition $G \succeq 0$ ensures that the matrix G is the Gram matrix to some set of vectors, while the condition that $\forall i \in [N]\ \operatorname{Tr}(E_{ii} G) = 1$ ensures that these are unit vectors. The quantity being maximized is

$$\begin{aligned} \operatorname{Tr}(PG) &= \sum_{s,t} P_{s,t} G_{s,t} \\ &= \sum_r \frac{1}{2^R} (-1)^{f_x(r)} \left\langle u_{\sigma(r)},\, v_{\tau(r)} \right\rangle. \end{aligned}$$

In order for our objective function matrix to be Hermitian, we rewrite our optimization problem with $H = \begin{pmatrix} 0 & \frac{1}{2}P \\ \frac{1}{2}P^{\mathrm{T}} & 0 \end{pmatrix}$, thus making it real and symmetric. Using H instead of P increases the dimensions of our matrices by a factor of 2. Our optimization problem then becomes the following SDP, which is equivalent to the form described in Section 2.4, using standard arguments which can be found in [20].

$$\begin{array}{llll} & \underline{\text{Primal}} & & \\ \max & & \mathrm{Tr}(HX) & \\ \text{s.t.} & \forall_i \in [2N] & \mathrm{Tr}(E_{ii}X) = 1 & \\ & & X \succeq 0. & (5) \end{array} \qquad \begin{array}{ll} & \underline{\text{Dual}} \\ \min & \sum_i y_i \\ \text{s.t.} & \sum_i y_i E_{ii} \succeq \mathbb{I} \\ & \forall i \in [2N]\,,\, y_i \geq 0. \end{array}$$

3.3 Uniformity of Questions in $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$

We here present a property of $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ which will prove to be important in more than one algorithm. This argument is part of unpublished work due to Rahul Jain, Sarvagya Upadhyay, and John Watrous.

Consider the value

$$\pi_\sigma(s) \equiv \sum_r g_s\left(r\right), \text{ where } g_s\left(r\right) = \begin{cases} \frac{1}{2^R} & \text{if } \sigma(r) = s \\ 0 & \text{otherwise} \end{cases}. \tag{6}$$

That is, $\pi_\sigma\left(s\right)$ is the probability that the verifier chooses the question s. We define $\pi_T\left(t\right)$ similarly.

We will show that when simulating an $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol in PSPACE, we can assume without loss of generality that our protocol possesses the following property.

Property (1)

$$\begin{aligned} \forall s \in S \qquad & \pi_\sigma(s) \leq \frac{2}{|S|} \\ \forall t \in T \qquad & \pi_\tau(t) \leq \frac{2}{|T|}. \end{aligned}$$

We will replace the given $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol with another $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ having the same probability of acceptance on each input x, but possessing property (1). We accomplish this with a padding argument. In this way, instead of having an implicit SDP for the bias of original protocol, we have an implicit SDP for the bias of a new protocol. The bias will be the same, so as long as the entries of the matrices of the new protocol can be accessed using a polynomial amount of space, solving

the new SDP with a PSPACE algorithm will still show $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1] \subseteq \mathsf{PSPACE}$.

Without modification to the protocol, we know both that

$$0 \leq \pi_\sigma(s) \leq 1,$$

and

$$\sum_{s \in S} \pi_\sigma(s) = 1.$$

We now present the conversion procedure to generate a new protocol:

For each $s \in S$, we make k_s unique copies (s,i) with $1 \leq i \leq k_s$ of s, in our new protocol, where $k_s = \lceil \pi_\sigma(s) |S| \rceil$. So if $k_s = 3$ our protocol will contain $(s,1), (s,2), (s,3) \in S'$.

Now we consider our random seed to be $r' = (r_1, r_2)$. s and t are chosen according to r_1 while the padding of s is chosen uniformly at random, using r_2. Because r_1 and r_2 are independent of each other, the prover gain no knowledge from the padding she receives.

Now,

$$\pi_{\sigma'}(s') = \pi_\sigma(s) \Pr\left(\mathrm{pad}(s, r_2)\right) = \frac{\pi_\sigma(s)}{k_s} \leq \frac{1}{|S|}.$$

The new number of questions is

$$\sum_s k_s \leq \sum_s \left(\pi_\sigma(s) |S| + 1\right) \leq 2 |S|.$$

So, we have $\pi_{\sigma'}(s') \leq \frac{2}{|S'|}$. Repeating this argument for questions $t \in T$ to the second prover, we have a new protocol possessing property (1).

We should note that this conversion procedure depends on knowing $\pi_\sigma(s), \pi_\tau(t)$ for each $s \in S, t \in T$. These can be computed in PSPACE by summing over all $r \in R$.

3.4 Positive Semidefinite Objective Function Matrix

If we replace H with $Q = H + \frac{3}{N}\mathbb{I}$, in our SDP 5, we are instead maximizing

$$\begin{aligned} &\mathrm{Tr}\left[\left(H + \frac{3}{N}\mathbb{I}\right) X\right] \\ &= \mathrm{Tr}[HX] + \frac{3}{N}\mathrm{Tr}[X] \\ &= \langle H, X \rangle + 3 \end{aligned} \tag{7}$$

If we solve this SDP, we simply subtract 3 from our answer and have the solution to the SDP for H. Let N be the dimension of H.

Claim. $H + \frac{3}{N}\mathbb{I}$ is positive definite.

Proof. The centers of its Gershgorin disks are $\frac{3}{N}$, and the sum of the absolute values of the off diagonal terms are $\pi_\sigma(s)$ or $\pi_\tau(t)$, which are at most $\frac{2}{N}$ for each row, by Property (1). $H + \frac{3}{N}\mathbb{I}$ is also real and symmetric, so it is Hermitian and has real eigenvalues, which, by the Gershgorin disk theorem are positive.

□

We may now write our SDP as

$$\begin{array}{lll} & \underline{\text{Primal}} & \\ \max & \operatorname{Tr}(QX) & \\ \text{s.t.} \quad \forall_i & \operatorname{Tr}(E_{ii}X) = 1 & \\ & X \succeq 0 & \end{array} \qquad \begin{array}{lll} & \underline{\text{Dual}} & \\ \min & \sum_i y_i & \\ \text{s.t.} & \sum_i E_{ii} y_i \succeq Q & \\ \forall_i & y_i \geq 0. & \end{array} \tag{8}$$

We will rewrite the SDP 8 in a more convenient form. A few definitions:
Define Δ as the completely depolarizing channel, i.e.

$$(\Delta(X))_{ij} = \begin{cases} X_{ii} & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Define Φ and its dual Φ^* as $\Phi(X) = \Delta\big(Q^{-1/2} X Q^{-1/2}\big)$ and $\Phi^*(X) = Q^{-1/2}\Delta(X)\,Q^{-1/2}$.
We now present the new form of the SDP.

$$\begin{array}{ll} \underline{\text{Primal}} & \\ \max & \operatorname{Tr}(X) \\ \text{s.t.} & \Phi(X) \preceq \mathbb{I} \\ & X \succeq 0 \end{array} \qquad \begin{array}{ll} \underline{\text{Dual}} & \\ \min & \operatorname{Tr}(Y) \\ \text{s.t.} & \Phi^*(Y) \succeq \mathbb{I} \\ & Y \succeq 0 \end{array} \tag{9}$$

This SDP can be seen as a reformulation of SDP 8. If we take a feasible solution X' to SDP 8, and compute $X = Q^{1/2} X' Q^{1/2}$, then X will be feasible for SDP 9, with the same objective value. Similarly, given X feasible to SDP 9, $X' = Q^{-1/2} X Q^{-1/2}$ is feasible to SDP 8. These transformations are possible because Q is positive definite.

We've also replaced the equality constraint with an inequality constraint. We can assume that any optimal solution to SDP 9 will achieve $\Phi(X) = \mathbb{I}$ because

of the following argument. Suppose X is optimal and $\Phi(X) \preceq \mathbb{I}$ but $\Phi(X) \neq \mathbb{I}$. Consider $B \neq \mathbf{0}$ such that $B \succeq 0$ and $\Phi(B) \preceq \mathbb{I} - \Phi(X)$. Then $\Phi(X+B) \preceq \mathbb{I}$ and $\mathrm{Tr}(X+B) > \mathrm{Tr}(X)$. Such a B exists because Φ is linear, $\Phi(B) \neq \mathbf{0}$ for all $B \succeq 0$, and $\mathbb{I} - \Phi(X) \succeq 0$, but $\mathbb{I} - \Phi(X) \neq \mathbf{0}$, so X can not be optimal unless $\Phi(X) = \mathbb{I}$.

On input x, the optimal value of this SDP is the bias for the given $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol. So if $c > s$, then

$$\begin{aligned} \text{opt} > 2s - 1 &\Rightarrow x \in L \\ \text{opt} < 2c - 1 &\Rightarrow x \notin L. \end{aligned}$$

If we can compute the maximum bias with which the provers can cause a verifier to accept, we need not simulate the $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol, but simply compare the optimum of the SDP, or a suitable approximation, with $2s-1$ and $2c-1$, to deduce whether $x \in L$ or $x \notin L$. If we solve this problem to an accuracy of $\text{opt} \pm \delta$, then as long as δ is smaller than half the gap between $2c-1$ and $2s-1$, we will never be wrong about accepting or rejecting. The positive SDP algorithms instead give a multiplicative $(1+\varepsilon)$ approximation of the bias, which is at most 1. So we set $\varepsilon = \delta = c - s$.

By lemma 2, there is no straightforward way to convert this SDP into the form of a positive SDP as defined in Section 2.4. However, the matrices involved in 9 are all positive semidefinite, so if there were an algorithm to solve arbitrary positive SDPs in super-operator form, we could conclude that for $\frac{1}{2} \leq s < c \leq 1$ and $c - s \in \Omega\Big(\frac{1}{\mathrm{poly}(|x|)}\Big)$, $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$.

4 Direct MMWU Algorithm to show $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$

4.1 Computational Task

SDP 8 is specified implicitly, as described in Section 3.3. To write down the entire SDP would take an exponential amount of space. However, we can access individual bits of the SDP in PSPACE. We will develop an algorithm based on the MMWU method, to approximate the value of SDP 8, which runs in PSPACE.

Our computational task is to approximate the value of SDP 8. On input x, the optimal value of this SDP, call it opt, is 3 plus the bias for the given $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol. See Eqn. (7) to understand why we add 3. So if $c > s$, then

$$\begin{aligned} \text{opt} > (2s-1) + 3 &\Rightarrow x \in L \\ \text{opt} < (2c-1) + 3 &\Rightarrow x \notin L. \end{aligned}$$

So comparing the optimal value of this SDP to $(2c-1)+3 = 2c+2$ and $(2s-1)+3 = 2s+2$ will tell us whether to accept or reject.

The algorithm presented below follows the MMWU method. We being with an initial guess for the solution, $\frac{I}{\mathrm{Tr}(I)}$, and in successive iterations we improve our guess. We continue until either the algorithm halts early, in which case we show that there exists a primal feasible solution to SDP 8 with objective value $> 2s-2$, so opt $> 2s-2$. Or we complete all iterations without halting, in which case we conclude that there exists a dual feasible solution with objective value $< 2c-2$, so by strong duality, opt $< 2c-2$.

For any choice of constants ε and γ, when the algorithm below accepts, there exists a primal feasible solution with objective value $> (1-\varepsilon)\gamma$. When it rejects, there exists a dual feasible solution with objective value $< (1+\varepsilon)\gamma$. Since we want to show either opt $> 2c+2$ or opt $< 2s+2$, we can set $\gamma = c+s+2$, and $\varepsilon = \frac{c-s}{c+s+2}$. If this algorithm accepts, there exists a primal feasible solution with objective value $> \left(1-\frac{c-s}{c+s+2}\right)(c+s+2) = 2s+2$. If it rejects, there exists a dual feasible solution with objective value $< \left(1+\frac{c-s}{c+s+2}\right)(c+s+2) = 2c+2$.

Algorithm 1 MMWU Algorithm for $\oplus\mathsf{MIP}^*[2,1]$

1. Let $\delta = \dfrac{\varepsilon^2}{8\left(\kappa(Q)\right)^2}$, $T = \left\lceil \dfrac{24\ln(N)}{\varepsilon^3\gamma^3\delta} \right\rceil$, $W_0 = \mathbb{I}$ and $\rho_0 = \dfrac{W_0}{\mathrm{Tr}(W_0)}$.
2. For $t = 0, \cdots, T-1$ do:
   (a) Let $S = \left\{ j : \Phi(\rho_t)_{jj} > \frac{1}{\gamma} \right\}$ and $s = \sum_{j\in S} \Phi(\rho_t)_{jj}$.
   (b) If $s \leq \delta \left\| Q^{-1} \right\|_\infty$, then halt and accept.
   (c) Let $Y_t = \sum_{j\in S} \frac{1}{s} E_{jj}$, $W_{t+1} = \exp\left(-\dfrac{\varepsilon\gamma\delta}{2} \Phi^*(Y_0 + \cdots + Y_t) \right)$, $\rho_{t+1} = \frac{W_{t+1}}{\mathrm{Tr}(W_{t+1})}$.
3. Halt and reject.

As is typical in MMWU algorithms, this algorithm makes iterative improvements to its guess for an optimal ρ. S can be thought of as the directions in which ρ_t does not satisfy the primal feasibility condition, and we are iteratively scaling back $\Phi(\rho)$ in the directions of S.

The algorithm is modeled after the algorithm by Jain et al. used to show $\mathsf{QIP}(2) \subseteq \mathsf{PSPACE}$. [11] The algorithms are very similar, but the feasibility condition is more simple in SDP 8, so the algorithm above has a more simple acceptance condition, which involves computing diagonal entries of $\Phi(\rho_t)$, instead of eigenval-

ues of $\Phi(\rho_t)$.

We can think of S as the set of directions in which our current guess ρ_t violates the feasibility condition $\Phi(\rho_t) \preceq \mathbb{I}$, and s as the amount by which the condition is violated. We iteratively reduce s until it is below a certain threshold. When $s \leq \delta \left\|Q^{-1}\right\|_\infty$, then ρ_t is approximately feasible and we show how to construct a feasible solution from ρ_t. While $s > \delta \left\|Q^{-1}\right\|_\infty$, our update rule has the effect of scaling down the directions in which which ρ_t violates feasibility.

When we accept, there is a sense in which ρ_t is "close" to being feasible. The analysis shows how to come up with a feasible solution from ρ_t with a higher enough objective value.

When we reject, we know that we have been iteratively improving our guess over T iterations. We prove a bound on the average over all of our Y_t's which gives us a dual feasible solution with low enough objective value.

4.1.1 Algorithm Accepts

If we accept at $\rho = \rho_t$, then let R_0 be $Q^{-1/2}\rho Q^{-1/2}$. So that $\Delta(R_0) = \Phi(\rho)$. We want to find some other matrix X close to ρ such that $\Phi(X)$ is feasible and $\operatorname{Tr}(X) \geq (1-\varepsilon)\gamma$. Define R_1 to be a positive semidefinite matrix with the same diagonal entries as R_0 but whose values are capped at $\frac{1}{\gamma}$. More precisely, Let $R_1 = \Gamma R_0 \Gamma$ for some positive semidefinite Γ such that

$$(R_1)_{jj} \;=\; \begin{cases} \frac{1}{\gamma} & \text{for } j \in S \\ (R_0)_{jj} & \text{for } j \notin S. \end{cases}$$

If we take

$$\Gamma_{jj} \;=\; \begin{cases} \frac{1}{\sqrt{\gamma (R_0)_{jj}}} & \text{for } j \in S \\ 1 & \text{for } j \notin S \end{cases}$$

a diagonal matrix, then $\Gamma \geq 0$, so $R_1 = \Gamma R_0 \Gamma \geq 0$ (Since $R_0 \geq 0$).

Let's first note:

$$\begin{aligned} \langle Q, R_0 - R_1 \rangle &= \operatorname{Tr}(QR_0) - \operatorname{Tr}(QR_1) \\ &= \operatorname{Tr}\big(QQ^{-1/2}\rho Q^{-1/2}\big) - \operatorname{Tr}\big(Q^{1/2}R_1Q^{1/2}\big) \\ &= \operatorname{Tr}(\rho) - \operatorname{Tr}\big(Q^{1/2}R_1Q^{1/2}\big) \\ &= 1 - \operatorname{Tr}\big(Q^{1/2}R_1Q^{1/2}\big) \end{aligned} \tag{10}$$

Because $Q \geq 0$ and ρ is a density matrix.

Since we have R_1,R_2 positive semidefinite, lemma 4 holds,

$$\|R_0 - R_1\|_1 \leq \sqrt{2\operatorname{Tr}(R_0)^2 + 2\operatorname{Tr}(R_1)^2 - 4\mathrm{F}(R_0, R_1)^2}.$$

We know that $\operatorname{Tr}(R_1) \leq \operatorname{Tr}(R_0)$, Since R_1 has the same diagonal as R_0, except with entries capped below $\frac{1}{\gamma}$. So,

$$\begin{aligned}
\|R_0 - R_1\|_1 &\leq \sqrt{4\operatorname{Tr}(R_0)^2 - 4\mathrm{F}(R_0, R_1)^2} \\
&= \sqrt{4\operatorname{Tr}(R_0)^2 - 4\left(\operatorname{Tr}\left(\sqrt{\sqrt{R_0}R_1\sqrt{R_0}}\right)\right)^2} \\
&= \sqrt{4\operatorname{Tr}(R_0)^2 - 4\left(\operatorname{Tr}\left(\sqrt{\sqrt{R_0}\Gamma R_0\Gamma\sqrt{R_0}}\right)\right)^2} \\
&= \sqrt{4\operatorname{Tr}(R_0)^2 - 4\left(\operatorname{Tr}\left(\sqrt{R_0}\Gamma\sqrt{R_0}\right)\right)^2} \\
&= \sqrt{4\operatorname{Tr}(R_0)^2 - 4\left(\operatorname{Tr}(R_0\Gamma)\right)^2} \\
&= \sqrt{4\operatorname{Tr}(R_0)^2 - 4\langle R_0, \Gamma\rangle^2}.
\end{aligned}$$

Since $\Gamma_{jj} = 1$ for $j \notin S$,

$$\begin{aligned}
\langle R_0, \Gamma\rangle &\geq \sum_{j\notin S} (R_0)_{jj} \\
&= \sum_j (R_0)_{jj} - \sum_{j\in S} (R_0)_{jj} \\
&= \operatorname{Tr}(R_0) - s \\
&\geq \operatorname{Tr}(R_0) - \delta \left\|Q^{-1}\right\|_\infty.
\end{aligned}$$

So,

$$\begin{aligned}
\|R_0 - R_1\|_1 &\leq \sqrt{4\operatorname{Tr}(R_0)^2 - 4\left(\operatorname{Tr}(R_0) - \delta\|Q^{-1}\|_\infty\right)^2} \\
&= \sqrt{8\operatorname{Tr}(R_0)\,\delta\|Q^{-1}\|_\infty - 4\delta^2\|Q^{-1}\|_\infty^2}.
\end{aligned}$$

Note that,

$$\begin{aligned}
\operatorname{Tr}(R_0) &= \operatorname{Tr}\left(Q^{-1/2}\rho Q^{-1/2}\right) \\
&= \operatorname{Tr}\left(Q^{-1}\rho\right) \\
&= \left\langle Q^{-1}, \rho\right\rangle \\
&\leq \left\|Q^{-1}\right\|_\infty \|\rho\|_1 \\
&= \left\|Q^{-1}\right\|_\infty.
\end{aligned}$$

The inequality above follows from lemma 3. So,

$$\begin{aligned}\|R_0 - R_1\|_1 &\leq \sqrt{8\delta \|Q^{-1}\|_\infty^2 - 4\delta^2 \|Q^{-1}\|_\infty^2} \\ &\leq \sqrt{8\delta \|Q^{-1}\|_\infty^2} \\ &= \sqrt{8\delta} \left\|Q^{-1}\right\|_\infty \\ &= \frac{\varepsilon}{\|Q\|_\infty}.\end{aligned}$$

Where in the last step we substitute in $\delta = \frac{\varepsilon^2}{8(\kappa(Q))^2}$.

Substituting this back into Eqn. (10), we get

$$\begin{aligned}1 - \operatorname{Tr}\left(Q^{1/2} R_1 Q^{1/2}\right) &= \langle Q, R_0 - R_1 \rangle \\ &\leq \|Q\|_\infty \|R_0 - R_1\|_1 \\ &\leq \varepsilon.\end{aligned}$$

Or,

$$\operatorname{Tr}\left(Q^{1/2} R_1 Q^{1/2}\right) \geq 1 - \varepsilon.$$

Let us take $X = \gamma \left(Q^{1/2} R_1 Q^{1/2}\right)$,

$$\begin{aligned}\operatorname{Tr}(X) &= \gamma \operatorname{Tr}\left(Q^{1/2} R_1 Q^{1/2}\right) \\ &\geq (1 - \varepsilon)\gamma.\end{aligned}$$

Thus, X has objective value $\geq (1 - \varepsilon)\gamma$.

Now to show that X is also primal feasible.

First, $X \geq 0$ since $Q, R_1 \geq 0$.

Finally, X satisfies the SDP 9 primal constraint.

$$\begin{aligned}\Phi(X) &= \Delta\left(Q^{-1/2} X Q^{-1/2}\right) \\ &= \Delta\left(Q^{-1/2} \gamma \left(Q^{1/2} R_1 Q^{1/2}\right) Q^{-1/2}\right) \\ &= \gamma \Delta(R_1) \\ &\preceq \mathbb{I}.\end{aligned}$$

We have thus proved that if the algorithm accepts, there exists a positive semidefinite X such that $\operatorname{Tr}(X) \geq (1 - \varepsilon)\gamma$, $\Phi(X) \preceq \mathbb{I}$. So X is feasible and has objective value $\geq (1 - \varepsilon)\gamma$.

4.1.2 Algorithm Rejects

Suppose our algorithm rejects.

Take

$$Y = \frac{1+\varepsilon}{T}\left(Y_0 + \cdots + Y_{T-1}\right),$$

where Y_t is the variable defined in line 2.(c) at iteration t. We prove 3 properties of our Y's then adapt the proof from [11].

First property:

For any $0 \leq t \leq T-1$,

$$\begin{aligned}
\mathrm{Tr}(Y_t) &= \sum_{j \in S} \frac{1}{s} \\
&< \frac{1}{s} \sum_{j \in S} \gamma \left(\Phi(\rho_t)\right)_{jj} \\
&= \gamma.
\end{aligned}$$

So,

$$\begin{aligned}
\mathrm{Tr}(Y) &= \frac{1+\varepsilon}{T}\left(\mathrm{Tr}(Y_0) + \cdots + \mathrm{Tr}(Y_{T-1})\right) \\
&< \frac{1+\varepsilon}{T}\left(\gamma + \cdots + \gamma\right) \\
&= (1+\varepsilon)\,\gamma. \qquad (11)
\end{aligned}$$

We have thus proven the following lemma.

Lemma 5. *If the algorithm rejects, Y as defined above satisfies* $\mathrm{Tr}(Y) < (1+\varepsilon)\,\gamma$.

Second property:

$$\begin{aligned}
\left\|\Phi^*(Y_t)\right\|_\infty &= \left\|Q^{-1/2}\Delta(Y_t)\,Q^{-1/2}\right\|_\infty \\
&\leq \left\|Q^{-1/2}\right\|_\infty \left\|Y_t\right\|_\infty \left\|Q^{-1/2}\right\|_\infty \\
&= \left\|Q^{-1}\right\|_\infty \left\|Y_t\right\|_\infty \\
&= \left\|Q^{-1}\right\|_\infty \left\|\sum_{j \in S} \frac{1}{s} E_{jj}\right\|_\infty \\
&= \frac{\left\|Q^{-1}\right\|_\infty}{s} \\
&< \frac{\left\|Q^{-1}\right\|_\infty}{\delta \left\|Q^{-1}\right\|_\infty} \text{ (since we reject, } s > \left\|Q^{-1}\right\|_\infty) \\
&= \frac{1}{\delta}. \qquad (12)
\end{aligned}$$

Third property:

$$\begin{aligned}\langle \rho_t, \Phi^*(Y_t)\rangle &= \langle \Phi(\rho_t)\,, Y_t\rangle \\ &= \left\langle \Phi(\rho_t)\,, \sum_{j\in S} \frac{1}{s} E_{jj} \right\rangle \\ &= \frac{1}{s}\sum_{j\in S} \Phi(\rho_t)_{jj} \\ &= 1. \end{aligned} \tag{13}$$

Take $\eta = \frac{\varepsilon\gamma}{2}$, now,

$$\begin{aligned}\mathrm{Tr}(W_{t+1}) &= \mathrm{Tr}[\exp(-\eta\delta\Phi^*(Y_0 + \cdots + Y_t))] \\ &\leq \mathrm{Tr}[\exp(-\eta\delta\Phi^*(Y_0 + \cdots + Y_{t-1}))\exp(-\eta\delta\Phi^*(Y_t))] \\ &= \mathrm{Tr}[W_t \exp(-\eta\delta\Phi^*(Y_t))]\,.\end{aligned}$$

The inequality above is follows from the linearity of Φ^* and the Golden-Thompson inequality, which says for Hermitian matrices A, B, $\mathrm{Tr}(\exp(A+B)) \leq \mathrm{Tr}(\exp(A)\exp(B))$. [2]

The following lemma is a straightforward consequence of a standard scalar inequality: For $0 \preceq P \preceq \mathbb{I}$, $\eta > 0$, $\exp(-\eta P) \leq \mathbb{I} - \eta\exp(-\eta)\,P$.

Since $\|\delta\Phi^*(Y_t)\| \leq 1$ (from 12) this lemma implies

$$\exp(-\eta\delta\Phi^*(Y_t)) \leq \mathbb{I} - \eta\delta\exp(-\eta)\,\Phi^*(Y_t)\,.$$

So,

$$\begin{aligned}\mathrm{Tr}(W_{t+1}) &\leq \mathrm{Tr}[W_t \exp(-\eta\delta\Phi^*(Y_t))] \\ &= \langle W_t, \exp(-\eta\delta\Phi^*(Y_t))\rangle \\ &= \mathrm{Tr}(W_t)\,\langle \rho_t, \exp(-\eta\delta\Phi^*(Y_t))\rangle \\ &\leq \mathrm{Tr}(W_t)\,\langle \rho_t, \mathbb{I} - \eta\delta\exp(-\eta)\,\Phi^*(Y_t)\rangle \\ &= \mathrm{Tr}(W_t)\,(\mathrm{Tr}(\rho_t) - \eta\delta\exp(-\eta)\,\langle \rho_t,\ \Phi^*(Y_t)\rangle) \\ &= \mathrm{Tr}(W_t)\,(1 - \eta\delta\exp(-\eta)) \quad \text{(from 13)} \\ &\leq \mathrm{Tr}(W_t)\exp(-\eta\delta\exp(-\eta))\,.\end{aligned}$$

By induction, substituting $\mathrm{Tr}(W_0) = \mathrm{Tr}(\mathbb{I}) = N$, we have

$$\mathrm{Tr}(W_T) \leq N\exp(-\eta\delta T\exp(-\eta))\,. \tag{14}$$

We also have

$$\begin{aligned}\mathrm{Tr}(W_T) &= \mathrm{Tr}[\exp(-\eta\delta\Phi^*(Y_0+\cdots+Y_{T-1}))] \\ &\geq \exp(-\eta\delta\lambda_N(\Phi^*(Y_0+\cdots+Y_{T-1})))\,. \end{aligned} \tag{15}$$

This last inequality follow from the fact that $\mathrm{Tr}(\exp(A)) = \sum_k e^{\lambda_k(A)} \geq e^{\lambda_N(A)}$, where $\lambda_k(A)$ is the k'th eigenvalue of A, and $\lambda_N(A)$ largest eigenvalue of A.

Combining Eqn. (14) and Eqn. (15), we have

$$\exp(-\eta\delta\lambda_N(\Phi^*(Y_0+\cdots+Y_{T-1}))) \leq \mathrm{Tr}(W_T) \leq N\exp(-\eta\delta T\exp(-\eta))\,, \tag{16}$$

Simplifying and substituting T and η, Eqn. (16) implies

$$\begin{aligned}
& -\eta\delta\lambda_N(\Phi^*(Y_0+\cdots+Y_{T-1})) \leq \ln(N) - \eta\delta T\exp(-\eta) \\
\implies \quad & \lambda_N(\Phi^*(Y_0+\cdots+Y_{T-1})) \geq -\frac{\ln(N)}{\eta\delta} + T\exp(-\eta) \\
\implies \quad & \lambda_N\left(\Phi^*\left(\frac{(1+\varepsilon)}{(1+\varepsilon)}\frac{Y_0+\cdots+Y_{T-1}}{T}\right)\right) \geq -\frac{\ln(N)}{\eta\delta T} + \exp(-\eta) \\
\implies \quad & \frac{1}{(1+\varepsilon)}\lambda_N(\Phi^*(Y)) \geq -\frac{2\varepsilon^3\gamma^3\delta\ln(N)}{\varepsilon\gamma\delta 24\ln(N)} + \exp(-\eta) \\
\implies \quad & \frac{1}{(1+\varepsilon)}\lambda_N(\Phi^*(Y)) \geq -\frac{\varepsilon^2\gamma^2}{12} + \exp(-\eta) \\
\implies \quad & \frac{1}{(1+\varepsilon)}\lambda_N(\Phi^*(Y)) \geq -\frac{\eta^2}{3} + \exp(-\eta) \\
\implies \quad & \lambda_N(\Phi^*(Y)) \geq (1+\varepsilon)(1-\eta) \\
\implies \quad & \lambda_N(\Phi^*(Y)) \geq (1+\varepsilon)\left(1-\frac{\varepsilon\gamma}{2}\right) \\
\implies \quad & \lambda_N(\Phi^*(Y)) \geq 1.
\end{aligned}$$

The last line follows because $0 \leq \gamma, \varepsilon \leq 1$. So, all eigenvalues of $\Phi^*(Y)$ are greater or equal to 1, so

$$\Phi^*(Y) \succeq \mathbb{I}.$$

Thus, If the algorithm rejects, there exists a Y which is dual feasible and, from lemma 5, has objective value $< (1+\varepsilon)\gamma$.

4.2 Precision

All matrix operations in the algorithm from Section 1 can be performed in NC except for matrix exponentiation. [19] The matrix exponential can, however, be approximated in NC and we will show that this is good enough for our purposes.

In the case that the algorithm accepts, none of our analysis depends on taking a matrix exponential. We do not need to think of the ρ_t which caused acceptance as "approximately" something else, the mere fact that it caused acceptance allows us to construct a feasible X.

In the case that the algorithm rejects, we proved 3 properties about Y_t: 11, 12 and 13. Namely,

$$\mathrm{Tr}(Y_t) < \gamma, \qquad \|\Phi^*(Y_t)\|_\infty < \tfrac{1}{\delta}, \qquad \langle \rho_t, \Phi^*(Y_t)\rangle = 1.$$

The first and second properties still hold with an approximate exponential function. The third property should be replaced with $\langle \rho_t, \Phi^*(Y_t)\rangle = 1 - \alpha$ for some accuracy parameter α at our discretion.

If we do the same analysis as before with some $\alpha < \frac{\eta^2}{12}$, we will again conclude that

$$\Phi^*(Y) \geq \mathbb{I}.$$

Therefore, we may implement the algorithm using exact and approximate matrix operations in NC.

4.3 Run-Time Analysis

The number of iterations of the algorithm in Section 1 is $T = \mathcal{O}\left(\frac{(\kappa(Q))^2 \ln(N)}{\varepsilon^5 \gamma^3}\right)$. Recall that ε and γ are constants depending on the completeness c and soundness s of the $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol. N is the dimension of the matrices of SDP 9, so it is $\mathcal{O}(2^n)$ where n is the size of the input x. The condition number $\kappa(Q)$ defined as $\|Q\|_\infty \|Q^{-1}\|_\infty$.

The Gershgorin centers of Q are all $\frac{3}{N}$, while the radii are at most $\frac{2}{N}$, by Property (1). So by the Gershgorin disk theorem the eigenvalues of Q lie between $\frac{1}{N}$ and $\frac{5}{N}$. So $\|Q\|_\infty \in \left(\frac{1}{N}, \frac{5}{N}\right)$. Therefore, $\|Q^{-1}\|_\infty \in \left(\frac{N}{5}, N\right)$. So the condition number is constant.

Thus, T is polynomial in the size of the input x. All of our matrix operations can be computed in NC, including approximating the matrix exponential (see Section 4.2). [19] Combining these two facts, it is possible to implement the algorithm as a polynomial depth circuit. Since $\mathsf{NC}\,(\mathrm{poly}) = \mathsf{PSPACE}$, [3] we have concluded that we can decide between $x \in L$ and $x \notin L$ for an $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1]$ protocol in PSPACE. From this we conclude that when $c > s$ and $c - s \in \Omega\left(\frac{1}{\mathrm{poly}(|x|)}\right)$, $\oplus\mathsf{MIP}^*_{\mathsf{c,s}}[2,1] \subseteq \mathsf{PSPACE}$. So,

$$\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}.$$

5 Conclusion

We have proven that $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{PSPACE}$ using a direct MMWU algorithm, giving a proof which does not depend on the result $\oplus\mathsf{MIP}^*[2,1] \subseteq \mathsf{QIP}(2)$. It is worth noting that that algorithm in Section 1 is more simple than most MMWU algorithms, including the one due to Jain et al.[11] upon which it is based. The "Oracle" step consists only of comparing elements on the diagonal of $\Phi(\rho_t)$ with $\frac{1}{\gamma}$, instead of doing something more complicated like taking spectral decompositions as in the algorithm upon which it is based.

The "width" parameter of the algorithm we have given can be considered to be the condition number $\kappa(Q)$ of the objective function matrix Q. The SDP formulation of $\oplus\mathsf{MIP}^*[2,1]$ happens to have a low enough width that our algorithm is feasible. In order to bound the width, it was important to assume the uniformity of the questions in the $\oplus\mathsf{MIP}^*[2,1]$ protocol, and then ensuring that we have a positive semidefinite Q. It is not clear that a fast parallel algorithm could exist for the general form of SDP 5 where the objective function matrix is an arbitrary Hermitian matrix. Our algorithm requires a positive definite objective function matrix, so when the matrix H from SDP 5 is an arbitrary Hermitian matrix, we need to add a large multiple of the identity to ensure H is positive definite. This results in too high a runtime to show containment in PSPACE.

A width independent algorithm may exist for SDPs in super-operator form containing only positive semidefinite matrices, such as the one characterizing $\oplus\mathsf{MIP}^*[2,1]$ or the ones characterizing $\mathsf{QIP}(2)$ or $\mathsf{QRG}(1)$, but as lemma 2 shows, it is not a straightforward application of the fast parallel algorithms of Jain and Yao or Peng and Tangwonsan. Finding a width-independent algorithm for positive SDPs in super-operator form, or a transformation into standard inequality form which preserves positivity is left as an open problem.

References

[1] Bellare, M., Goldreich, O., and Sudan, M. Free bits, PCPs, and nonapproximability—towards tight results. *SIAM Journal on Computing 27*, 3 (June 1998), 804–915. 1

[2] Bhatia, R. *Matrix Analysis*. Springer-Verlag, New York, 1997. 20

[3] Borodin, A. On relating time and space to size and depth. *SIAM J. Comput. 6*, 4 (1977), 733–744. 8, 22

[4] Boyd, S., and Vandenberghe, L. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. 1, 7

[5] Cleve, R., Gavinsky, D., and Jain, R. Entanglement-resistant two-prover interactive proof systems and non-adaptive pir's. *Quantum Info. Comput. 9*, 7 (July 2009), 648–656. 1

[6] Cleve, R., Hoyer, P., Toner, B., and Watrous, J. Consequences and limits of nonlocal strategies. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity* (2004), pp. 236–249. 1, 9

[7] Geršgorin, S. Uber die abgrenzung der eigenwere einer matrix. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na 6* (1931), 749–754. 5

[8] Gutoski, G., and Wu, X. Parallel approximation of min-max problems with applications to classical and quantum zero-sum games. In *Computational Complexity (CCC), 2012 IEEE 27th Annual Conference on* (2012), pp. 21–31. 2

[9] Håstad, J. Some optimal inapproximability results. *J. ACM 48*, 4 (July 2001), 798–859. 1

[10] Jain, R., Ji, Z., Upadhyay, S., and Watrous, J. QIP = PSPACE. In *Proceedings of the 42nd ACM symposium on Theory of computing* (New York, NY, USA, 2010), STOC '10, ACM, pp. 573–582. 2

[11] Jain, R., Upadhyay, S., and Watrous, J. Two-message quantum interactive proofs are in pspace. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (Washington, DC, USA, 2009), FOCS '09, IEEE Computer Society, pp. 534–543. 1, 2, 3, 5, 15, 19, 23

[12] Jain, R., and Watrous, J. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity* (Washington, DC, USA, 2009), CCC '09, IEEE Computer Society, pp. 243–253. 2

[13] Jain, R., and Yao, P. A parallel approximation algorithm for positive semidefinite programming. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on* (2011), pp. 463–471. iii, 2, 8

[14] Jain, R., and Yao, P. A parallel approximation algorithm for mixed packing and covering semidefinite programs. *CoRR abs/1201.6090* (2012). 2, 8

[15] Kale, S. Efficient algorithms using the multiplicative weights update method. Tech. rep., Princeton University, 2006. 2

[16] Nielsen, M. A., and Chuang, I. L. *Quantum Computation and Quanum Information*. Cambridge University Press, 2000. 5

[17] Peng, R., and Tangwongsan, K. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. In *Proceedinbgs of the 24th ACM symposium on Parallelism in algorithms and architectures* (New York, NY, USA, 2012), SPAA '12, ACM, pp. 101–108. 2, 8

[18] Tsirel'son, B. Quantum analogues of the Bell inequalities: The case of two spatially separated domains. *Journal of Soviet Mathematics 36* (1987), 557–570. 9

[19] von zur Gathen, J. Parallel linear algebra. In *John H. Reif, editor, Synthesis of Parallel Algorithms* (San Francisco, CA, USA, 1993), Morgan Kaufmann Publishers Inc. 21, 22

[20] Watrous, J. CS 766/QIC 820 Theory of Quantum Information (Fall 2011). https://cs.uwaterloo.ca/ watrous/CS766/, 2011. 7, 11

[21] Wehner, S. Entanglement in interactive proof systems with binary answers. In *Proceedings of the 23rd Annual conference on Theoretical Aspects of Computer Science* (Berlin, Heidelberg, 2006), STACS'06, Springer-Verlag, pp. 162–171. 1