Parallelized Solution to Semidefinite Programmings in Quantum Complexity Theory

Xiaodi Wu *

Institute for Quantum Computing, University of Waterloo, Ontario, Canada,

Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, USA

July, 2010

Abstract

In this paper we present an equilibrium value based framework for solving SDPs via the multiplicative weight update method which is different from the one in Kale's thesis [Kal07]. One of the main advantages of the new framework is that we can guarantee the convertibility from approximate to exact feasibility in a much more general class of SDPs than previous result. Another advantage is the design of the oracle which is necessary for applying the multiplicative weight update method is much simplified in general cases. This leads to an alternative and easier solutions to the SDPs used in the previous results QIP(2)$\subseteq$PSPACE [JUW09] and QMAM=PSPACE [JJUW09]. Furthermore, we provide a generic form of SDPs which can be solved in the similar way. By parallelizing every step in our solution, we are able to solve a class of SDPs in NC. Although our motivation is from quantum computing, our result will also apply directly to any SDP which satisfies our conditions.

In addition to the new framework for solving SDPs, we also provide a novel framework which improves the range of equilibrium value problems that can be solved via the multiplicative weight update method. Before this work we are only able to calculate the equilibrium value where one of the two convex sets needs to be the set of density operators. Our work demonstrates that in the case when one set is the set of density operators with further linear constraints, we are still able to approximate the equilibrium value to high precision via the multiplicative weight update method.

*The work was completed when the author was visiting the Institute for Quantum Computing , University of Waterloo.

1 Introduction

Semidefinite programming (SDP) is a relatively new field of optimization which grew up in the 1990s [Ali95, BV04, Lov03, VB96, dK02]. Despite of the short time since its introduction, SDP has proved useful in many different contexts. Especially, there are many applications of SDPs in the field of theoretical computer science, like the design of approximation algorithm [Vaz01] and the recent application to Unique Game Conjecture [Ste10, Rag08] and simulating quantum complexity classes [Gut05, GW05, GW07, JJUW09, JUW09, KW00, Wat09b].

SDPs are in fact a special case of conic programming and there exists polynomial algorithms to solve any SDP instance (like the interior point method [Ali95, dK02]). Thus, any problem which can be modeled or approximated as a SDP is considered to have efficient solutions. However, on the other side, our understanding of SDPs is far less than our understanding of Linear programming(LP), another typical optimization method used a lot in practice. One thing is about the running time : although any SDP instance can be solved in polynomial time theoretically, it is much slower than LP's solution in practice. Moreover, the generic algorithm for SDP are always used as a black box. Hence it is rarely seen that we can employ the duality or the structure of SDPs while the duality of LPs inspires lots of new algorithm designs.

A generic *primal-dual* method for SDP problem (under certain conditions) was introduced by Arora *et.al.* [AHK05a, AHK05b, AK07] to overcome the difficulties mentioned above. The generic method exploits a generic framework (or meta-algorithm) called the *multiplicative weight update method*. Similar frameworks were studied [Fle00, FS99, GK98, PST91, Kha04, WK06, You95] for different purposes before. This new generic method turns out to be very useful and successful in several contexts. In the paper [AK07] (see also Kale's Phd thesis [Kal07]) where it was originally proposed, this generic method improves the upper bounds of running time of many approximation algorithms. In addition to that, this generic method was introduced by Watrous *et. al.* to the field of quantum computation and successfully proved QIP(2)⊆PSPACE [JUW09] and QIP=PSPACE [JJUW09].

A *semidefinite program* over $\mathcal{X}$ and $\mathcal{Y}$ (shown below) is specified by a triple (Ψ, A, B) where $\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is a Hermiticity preserving super-operator and $A \in \mathrm{Herm}(\mathcal{X})$ and $B \in \mathrm{Herm}(\mathcal{Y})$. This form which was obtained in [Wat08] is somehow different from but equivalent to the standard form. Let α, β be the optimum values of the primal and dual programs respectively. One important property of the semidefinite program, called the *duality*, implies that $\alpha \leq \beta$ and the equality will hold in some situation.

Primal problem		Dual problem	
maximize:	$\langle A, X\rangle$	minimize:	$\langle B, Y\rangle$
subject to:	$\Psi(X) \leq B$,	subject to:	$\Psi^*(Y) \geq A$,
	$X \in \mathrm{Pos}(\mathcal{X})$.		$Y \in \mathrm{Pos}(\mathcal{Y})$.

In practice we usually consider the SDPs whose optimum values are within a small range. For those SDPs, instead of directly calculating the optimum value, we usually consider the *feasibility problem* first. Once the feasibility problem is solved, we could use the binary search in that range to find the optimum value. For any instance of SDP and a guess value c , the feasibility problem is defined to be

Feasibility Problem	
ask whether:	$\langle A, X\rangle \geq c$
subject to:	$\Psi(X) \leq B$,
	$X \in \mathrm{Pos}(\mathcal{X})$.

Intuitively, the primal-dual method in Kale's Thesis [Kal07] for the feasibility problem contains the following three ingredients. First, update via the multiplicative weight update method. Second, the existence of an *efficient width-bounded* oracle $\mathcal{O}_1$. Finally, this method will generate an approximately feasible solution Y to the dual problem such that $\langle B, Y\rangle \leq c$. By approximate feasibility, we mean

$$\forall \rho \in \mathrm{D}(\mathcal{X}), \langle \Psi^*(Y) - A, \rho\rangle \geq -\epsilon \tag{1}$$

where the $\mathrm{D}(\mathcal{X})$ denotes the set of density operators over the space $\mathcal{X}$ and ϵ is some small constant. In order to solve the feasibility problem, we need to get an exact dual feasible solution $\tilde{Y}$ such that $\langle B, \tilde{Y} \rangle \leq (1+\varepsilon)c$ from this. We will refer this as the *convertibility from approximate to exact feasibility*. After executing a combination of those three ingredients, this method will either return a feasible solution $\tilde{X}$ to the primal problem with object function value at least c or a feasible solution $\tilde{Y}$ such that $\langle B, \tilde{Y} \rangle \leq (1+\varepsilon)c$. The latter case will imply $\alpha \leq \beta \leq (1+\epsilon)c$ by the duality of SDPs. A detailed description of this procedure can be found in Appendix A.3.

Another important value which can be calculated via the multiplicative weight update method is the *equilibrium value* of zero-sum games [vN28] and its generalizations (like, [Haz06]). Particulary, we consider the value λ,

$$\lambda = \min_{x \in X} \max_{y \in Y} f(x,y) = \max_{y \in Y} \min_{x \in X} f(x,y)$$

for some *convex-concave* function f (see definition in Appendix A.3) over $X \times Y$ where X, Y are convex compact sets. Again, this method involves the update via the multiplicative weight update method and an *efficient width-bounded* but functionally different oracle $\mathcal{O}_2$ as main ingredients. However, for equilibrium value, there is no requirement for the convertibility from approximate to exact feasibility. Under several other conditions as well (see details in Theorem 2 and Appendix A.3), such value λ can be approximated to high precision efficiently. It should be noted here the multiplicative weight update method plays a quite different role from the one in the solution to SDPs above. This difference has led to an alternative and easier proof of QIP=PSPACE [Wu10]. It was also applied to the proof of QRG(1)$\subseteq$PSPACE [JW09].

One important advantage of the solutions to SDPs or equilibrium value based on the multiplicative weight update method is that we can easily implement the algorithm in parallel. Precisely, this is because the fundamental operations of matrices and singular value decomposition of matrices [Gat93] can be implemented with high accuracy in NC. This trick was widely exploited in the recent progress of quantum complexity theory [Wu10, JJUW09, JUW09, JW09].

In this paper we will demonstrate how the equilibrium value can be related to the feasibility problem (then, SDPs). It will then suffice to make use of the solution to the equilibrium value to solve the feasibility problem. Following this idea, we will provide an alternative and easier solution to the SDPs used for QIP(2) [JUW09] and QMAM [JJUW09]. Moreover, we provide a generic form and conditions under which any feasibility problem can be solved in the same way. By parallelizing each step in the solution, we are able to show any feasibility problem can be solved in NC as well. Precisely, we consider the feasibility problem as follows.

Feasibility Problem

$$\begin{aligned} \text{ask whether:} \quad & \langle A, X \rangle \geq c \\ \text{subject to:} \quad & \Psi(X) \leq B, \\ & X \in \mathrm{D}(\mathcal{X}). \end{aligned}$$

This feasibility problem is very similar to the most general version above. The only difference is we replace the condition $X \in \mathrm{Pos}(\mathcal{X})$ by $X \in \mathrm{D}(\mathcal{X})$. This change corresponds to the trace bound $\mathrm{Tr}\, X = R$ for some const R which commonly appears in applications of SDPs. Under certain conditions our constraint on $X \in \mathrm{D}(\mathcal{X})$ is equivalent to the trace bound $\mathrm{Tr}\, X \leq R$. This implies the feasibility problem in our consideration is very general and could cover many instances in practical use. Although our original motivation is from quantum computation, our result also works well for any SDPs which can be converted to the form in our consideration.

The concept of equilibrium values will be related if we imagine a two-player game to solve the feasibility problem. Assume there is a primal player who wants to provide you a feasible solution X to prove the original problem is feasible. On the contrary, the dual player who wants to disprove the feasibility will try to find where the constraints on X are violated. This is different from Kale's method where the disproof of the feasibility is by getting some feasible solution to the dual problem and then making use of the duality of SDPs. Precisely, we define the following convex-concave function f to capture the two-player game.

Framework 1. *Let function f be*

$$f(X,\Pi) = \left\langle \begin{pmatrix} c - \langle A, X\rangle & \\ & \Psi(X) - B \end{pmatrix}, \Pi \right\rangle \tag{2}$$

over the set $\mathrm{D}(\mathcal{X}) \times T$ where $T = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{Y}\oplus\mathbb{C}}\}$. Let the equilibrium value $\lambda^\star$ be

$$\lambda^\star = \min_{X\in\mathrm{D}(X)} \max_{\Pi\in T} f(X,\Pi) = \max_{\Pi\in T} \min_{X\in\mathrm{D}(X)} f(X,\Pi)$$

The relation between the equilibrium value $\lambda^\star$ and the feasibility of the original problem is captured by the following theorem.

Theorem 1. *The original problem is feasible if and only if $\lambda^\star \leq 0$.*

This formulation is inspired by similar formulations [Haz06] used for convex optimization and Theorem 1 follows easily from the argument in [Haz06]. Given the fact that equilibrium value $\lambda^\star$ can only be calculated approximately, we still need to do the conversion from approximately to exactly feasible solutions. Due to this reason, we make an important change to the old formulation in [Haz06]. Namely, we choose T to be $\{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{Y}\oplus\mathbb{C}}\}$ rather than the set of density operators over the space $\mathcal{Y} \oplus \mathbb{C}$. Assume we approximate $\lambda^\star$ to precision ϵ and the return value implies $\lambda^\star$ is in an interval containing 0. Or equivalently, we encounter the situation

$$\forall \quad \Pi \in T, \left\langle \begin{pmatrix} c - \langle A, X\rangle & \\ & \Psi(X) - B \end{pmatrix}, \Pi \right\rangle \leq \epsilon \tag{3}$$

and we want to convert X into some exact feasible solution without changing the object function a lot. This is very similar to the situation captured by Equation [1]. However, the difficulty in making this conversion happen is different. By applying the approximate feasible result in Equation [1], we can only expect to solve the problems [JUW09, JJUW09] when $A = \mathbb{1}_{\mathcal{X}}$ efficiently in general. This is because the density operator ρ in Equation [1] implies the approximate feasibility only holds in term of $\mathcal{L}_\infty$ norm. On the other side, in order to remain the object function which is a inner product almost unchanged, the approximate feasibility needs to hold in a stronger sense like $\mathcal{L}_1$ norm. Otherwise only trivial cases like $A = \mathbb{1}_{\mathcal{X}}$ can be solved.

Our new approximate feasibility result in Equation [3] overcomes such difficulty and provides a method to do the conversion from the approximate to exact feasibility in much more general cases. Especially we will demonstrate when the super-operator Ψ is *partial trace* or its generalizations, our approximate feasibility result works very well to make the conversion happen. This kind of constraints is very powerful because it is the only type of constraints we need in many SDPs in quantum computation. In addition to that, we still have the freedom to choose T in order to meet the requirement of new types of constraints. Such freedom is a huge advantage over the primal-dual method since Equation [1] is the only result can be expected from the primal-dual method.

Another advantage of the Framework 1 is that the oracle $\mathcal{O}_2$ which will be required to compute the equilibrium value $\lambda^\star$ can be easily designed. Precisely, since $T = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{Y}\oplus\mathbb{C}}\}$, we can simply get the other part's spectral decomposition and let T be the projection onto the positive eigenspace of it. Finally, under the conditions of Theorem 2, we can implement the whole algorithm in NC.

Besides the application to the feasibility problem, the idea of Framework 1 can also improve our ability on calculating equilibrium over some non-density operator set. Consider some convex-concave function h over $X \times Y$ where X is the set of density operators but with some constraint Φ. Precisely,

Framework 2. *If we consider the equilibrium value μ,*

$$\mu = \min_{x\in X} \max_{y\in Y} h(x,y) = \max_{y\in Y} \min_{x\in X} h(x,y)$$

where $X = \{\rho : \rho \in \mathrm{D}(\mathcal{X}), \Phi(\rho) \leq B\}$ and Y is any other compact convex set. It will be useful to consider the equilibrium value μ' as follows

$$\mu' = \min_{x\in X'} \max_{\{y,\Pi\}\in Y'} h(x,y) + \alpha \left\langle \Phi(\rho) - B, \Pi \right\rangle = \max_{\{y,\Pi\}\in Y'} \min_{x\in X'} h(x,y) + \alpha \left\langle \Phi(\rho) - B, \Pi \right\rangle$$

where $X = \mathrm{D}(\mathcal{X})$, $Y' = Y \times T$ ($T = \{\Pi : 0 \leq \Pi \leq \mathbb{1}\}$) and α is any factor.

It is easy to see that the idea of Framework 2 or especially the term $\langle \Phi(\rho) - B, \Pi \rangle$ is to penalize when ρ does not satisfy the constraint $\Phi(\rho) \leq B$. Furthermore, the penalization is weighted according to the factor α. By using the game value of one restricted model of one-round quantum refereed game as an example, we can demonstrate when there are two promise values of μ with large gap, such gap can be transferred to the new value μ'. Thus, it will be sufficient to calculate the value of μ' in order to distinguish between two promises of the original value μ. This improves the range of problems which can be solved by the multiplicative weight update method since so far we can only calculate the equilibrium value when X is the set of density operators (up to a factor). This will also give a binary search method for calculating the equilibrium value to some precision by artificially assuming two promise values.

The rest of the paper is organized as follows. Most of the preliminaries can be found in Appendix A. We also leave the lemmas and theorems which will be directly used in Section 2. The two examples for Framework 1 will be demonstrated in Section 3 (QIP(2)) and in Appendix B (QMAM) respectively. One restricted version of one-round quantum refereed game will be discussed in Section 4 to demonstrate the power of Framework 2. We will conclude the paper with further discussions and open problems in Section 5.

There are two points to make clear before the readers move on to the next section. First, when we consider the feasibility problem or equilibrium value directly, we always refer the size of the SDP or the function with equilibrium value as the input size. However, when we consider the quantum complexity classes, the SDP or the function with equilibrium value will have exponential size in term of its actual input size $|x|$. Second, we will not take care of the precision issues with the NC implementation in the main part of this paper. Instead, we will assume such implementation can be made exactly and deal with them in Appendix C.

2 Preliminaries

In order to make this paper self-contained, we try to provide brief surveys on each topic related to our paper. However, most of them will be put in the appendix due to limited space. Precisely, we will introduce the fundamentals of quantum information in Appendix A.1. Useful facts on NC and parallel matrix operations are stated in Appendix A.2. The multiplicative weight update method and its application to calculating the equilibrium value and SDPs (the primal-dual method) are surveyed in Appendix A.3.

2.1 Useful Lemma and Facts in Quantum Information

The following are some important lemmas about purification and fidelity which are useful in our proof later. The proofs of following results are put in Appendix A.1.

Lemma 1. *Given any two density operators ρ_1, ρ_2 over the space $\mathcal{A}$, and another density operator σ_1 over the space $\mathcal{A} \otimes \mathcal{B}$ such that $\mathrm{Tr}_B \sigma_1 = \rho_1$, then there exists another density operator σ_2 over the space $\mathcal{A} \otimes \mathcal{B}$ for which that $\mathrm{Tr}_B \sigma_2 = \rho_2$ and $\mathcal{F}(\rho_1, \rho_2) = \mathcal{F}(\sigma_1, \sigma_2)$.*

Lemma 2. *In addition to the result in Lemma 1, we can compute the classical representation of σ_2 as required above given the classical representations of ρ_1, ρ_2 and σ_1 in* NC *where the input size refers to the size of the matrices.*

Lemma 3. *Given two density operators $\rho_1, \rho_2 \in \mathrm{D}(\mathcal{A})$, and their purifications $\sigma_1, \sigma_2 \in \mathrm{D}(\mathcal{A} \otimes \mathcal{B})$ in space $\mathcal{A} \otimes \mathcal{B}$ respectively, for which $\mathcal{F}(\rho_1, \rho_2) = \mathcal{F}(\sigma_1, \sigma_2)$, let*

$$s = \frac{1}{2}\|\rho_1 - \rho_2\|_1 \text{ and } t = \frac{1}{2}\|\sigma_1 - \sigma_2\|_1$$

Then we have the following inequalities

$$1 - s \leq \sqrt{1 - t^2} \text{ and } 1 - t \leq \sqrt{1 - s^2}$$

Lemma 4. *Given two density operators $\rho_1, \rho_2 \in \mathrm{D}(\mathcal{A} \otimes \mathcal{B})$ where ρ_1 represents a pure state, there exists an admissable quantum channel $\Phi : \mathrm{L}(\mathcal{A}) \to \mathrm{L}(\mathcal{A})$ such that $\Phi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{B})}(\rho_1) = \rho_2$ if and only if $\mathrm{Tr}_{\mathcal{A}}(\rho_1) = \mathrm{Tr}_{\mathcal{A}}(\rho_2)$.*

1. Let $\varepsilon = \frac{\delta}{4r}$ and $T = \left\lceil \frac{16r^2 \ln D}{\delta^2} \right\rceil$. Also let $W^{(1)} = \mathbb{1}_{\mathcal{X}}$, $D = \dim(\mathcal{X})$.
2. Repeat for each $t = 1, \ldots, T$:
   (a) Let $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}$ and let $\Pi^{(t)}$ be the projection onto the positive eigenspace of $S(\rho^{(t)})$.
   (b) Let $M^{(t)} = (N(\Pi^{(t)}) + r\mathbb{1}_{\mathcal{X}})/2r$ and update the weight matrix as follows:
$$W^{(t+1)} = exp(-\varepsilon \sum_{\tau=1}^{t} M^{(\tau)})$$
3. Choose $\bar{\rho} = \frac{1}{T}\sum_{\tau=1}^{T} \rho^{(t)}$ and let $\bar{\Pi}$ be the projection onto the positive eigenspace of $S(\bar{\rho})$. Return $(\bar{\rho}, \bar{\Pi})$ as the approximate equilibrium point and $\langle S(\bar{\rho}), \bar{\Pi}\rangle$ as the approximate equilibrium value.

Figure 1: An algorithm that computes the approximate value and point to precision δ.

Lemma 5. *Given any Hermitian operator A such that* $\operatorname{Tr} A = 0$*, then we have*

$$\max_{\Pi:0\le\Pi\le\mathbb{1}} \langle A, \Pi\rangle = \frac{1}{2}\|A\|_1 \text{ and } \min_{\Pi:0\le\Pi\le\mathbb{1}} \langle A, \Pi\rangle = -\frac{1}{2}\|A\|_1$$

2.2 Multiplicative Weights Update Method

The detailed discussion of the multiplicative weight update method is provided in Appendix A.3. However, we will demonstrate the particular algorithm to calculate the equilibrium value in the form of Equation [2] in this section. Precisely, we consider the

$$\lambda^\star = \min_{\rho\in \mathrm{D}(\mathcal{X})} \max_{\Pi\in T} \left\langle \begin{pmatrix} c - \langle A, \rho\rangle & \\ & \Psi(\rho) - B \end{pmatrix}, \Pi \right\rangle \tag{4}$$

The existence of the equilibrium value is implied before. Thus we only need to see when a NC algorithm will exist to calculate the λ approximately to high precision. To ease the description of the algorithm, let $S(\rho)$ defined to be

$$S(\rho) = \begin{pmatrix} c - \langle A, \rho\rangle & \\ & \Psi(\rho) - B \end{pmatrix}$$

for any $\rho \in \mathrm{D}(\mathcal{X})$. Choose $N(\Pi)$ to be the raw loss matrix such that for any $\rho \in \mathrm{D}(\mathcal{X})$ and $\Pi = \begin{pmatrix} p & \\ & P \end{pmatrix} \in T$, we have

$$\langle S(\rho), \Pi\rangle = \langle \rho, N(\Pi)\rangle \tag{5}$$

It is easy to see that we can choose $N(\Pi)$ to be

$$N(\Pi) = -pA + \Psi^*(P) + (pc - \langle B, P\rangle)\mathbb{1}_{\mathcal{X}} \tag{6}$$

Theorem 2. *Let the input size, denoted by $|x|$, be the size of any instance in Equation [4]. If for any $\Pi \in T$ the $N(\Pi)$ defined above satisfies $\|N(\Pi)\|_\infty \le r$ where $r = O(polylog(|x|))$ and $\Psi(\rho)$ can be calculated in* NC *for any $\rho \in \mathrm{D}(\mathcal{X})$, then by using the algorithm in Figure 1, we can approximate the equilibrium value to precision $\delta = \Omega(polylog(|x|))$ in* NC*.*

We will leave the proof of Theorem 2 in Appendix A.3.

3 QIP(2) Case

Now it is our turn to consider a real instance of semidefinite program and apply our framework to solve it. Our first candidate is the quantum interactive proof system with two-messages. In this system, after the input x is given, the polynomial-time bounded quantum verifier will send one quantum message to an all powerful quantum prover and get another quantum message back. Then the verifier will decide whether to accept or to reject based on the message sent back from the prover and the qubits kept at his side. The only constraint on the all powerful quantum prover is that the prover must operate an admissable quantum operation on the quantum message sent to him. The complexity class QIP(2) denotes all the languages which can be recognized by the procedure above. Precisely, we have

Definition 1. *Any language L is inside* QIP(2) *if and only if*

- *If $x \in L$, there exists a prover such that the verifier will* accept *with probability at least $c(|x|)$.*
- *If $x \notin L$, for any prover the verifier will* accept *with probability at most $s(|x|)$.*

where $c(|x|) - s(|x|) = \Omega(1/poly(|x|))$.

It is known that QIP(2)⊆PSPACE [JUW09] by following Kale's way [Kal07] to solve SDPs. By contrast, we will demonstrate here how our **Framework 1** can be applied to this problem. Namely, we provide an alternative proof of the result QIP(2)⊆PSPACE. The main difference between our approach and the previous approach is that we formulate the problem using the density operators instead of quantum channels and we solve the semidefinite program using the new framework.

Let $\mathcal{M}$ denote the message's space between the prover and the verifier and $\mathcal{V}$ denote the verifier's private space. Let us assume the input x is fixed for the following discussion. Without lost of generality, let the pure state $\rho_1 \in \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$ be the initial state for the input x, namely the state that the verifier prepares given input x. The prover will then operate an admissable quantum channel $\Phi : \mathrm{L}(\mathcal{M}) \to \mathrm{L}(\mathcal{M})$ on part of the state ρ_1 and it will result another state $\rho_2 = \Phi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V})}(\rho_1)$. The verifier will performance a POVM measurement on ρ_2 to decide whether to accept or to reject. Let R be the POVM which corresponds to the case where the verifier accepts. In order to decide whether $x \in L$, it suffices to solve the optimization problem

$$\max_{\Phi} \langle R, \rho_2 \rangle \text{ s.t. } \rho_2 = \Phi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V})}(\rho_1)$$

where the optimum value is the maximum probability that the verifier accepts given the input x.

Because of Lemma 4, we have ρ_1 and ρ_2 are connected by an admissable quantum operation if and only if $\mathrm{Tr}_{\mathcal{M}} \rho_1 = \mathrm{Tr}_{\mathcal{M}} \rho_2$. Thus the above optimization problem is equivalent to the following SDP, denoted by SDP (I).

SDP Problem		Feasibility Problem	
maximize:	$\langle R, \rho_2 \rangle$	ask whether:	$\langle R, \rho_2 \rangle \geq c$
subject to:	$\mathrm{Tr}_{\mathcal{M}}(\rho_2) \leq \mathrm{Tr}_{\mathcal{M}}(\rho_1)$,	subject to:	$\mathrm{Tr}_{\mathcal{M}}(\rho_2) \leq \mathrm{Tr}_{\mathcal{M}}(\rho_1)$,
	$\rho_2 \in \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$.		$\rho_2 \in \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$.

3.1 Solution to the Feasibility Problem

Following the Framework 1, we consider the feasibility problem above. Precisely, we define

$$f_1(\rho, \Pi) = \left\langle \begin{pmatrix} c - \langle R, \rho \rangle & \\ & \mathrm{Tr}_{\mathcal{M}}(\rho) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \end{pmatrix}, \Pi \right\rangle \tag{7}$$

where $\rho \in T_1 = \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$ and $\Pi \in T_2 = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{M} \oplus \mathbb{C}}\}$. Let $\lambda_1^\star$ be the equilibrium value of function f_1, namely,

$$\lambda_1^\star = \min_{\rho \in T_1} \max_{\Pi \in T_2} f_1(\rho, \Pi) = \max_{\Pi \in T_2} \min_{\rho \in T_1} f_1(\rho, \Pi)$$

Base on Theorem 1, the value of $\lambda_1^\star$ will imply whether the original problem is feasible. In addition to that, we will demonstrate how to convert any approximately feasible solution to exactly feasible solution without changing the value of the object function a lot.

Lemma 6. *Assume we can calculate the approximate equilibrium value and point of the function f_1 to the precision δ. Let the $\bar{\lambda_1}$ and $\{\bar{\rho}, \bar{\Pi}\}$ be the approximate equilibrium value and point returned by algorithm in Figure 1, $\lambda_1^\star$ be the actual equilibrium value. Then we have*

- *if $\bar{\lambda_1} > \delta$, then the original problem is infeasible.*
- *if $\bar{\lambda_1} \leq \delta$, then there exists a feasible solution $\tilde{\rho}$ such that $\langle R, \tilde{\rho} \rangle \geq c - \sqrt{2\delta - \delta^2}$.*

Proof.

- If $\bar{\lambda_1} > \delta$, namely, $\lambda_1^\star \geq \bar{\lambda_1} - \delta > 0$, then due to Theorem 1, the original problem is feasible.
- Otherwise, due to Lemma 5, we have

$$h_1(\bar{\rho}, \bar{P}) = \max\{c - \langle R, \bar{\rho} \rangle, 0\} + \frac{1}{2} \| \mathrm{Tr}_{\mathcal{M}}(\bar{\rho}) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \|_1 \leq \delta \tag{8}$$

By Lemma 1 and 2, we can compute $\tilde{\rho}$ such that $\mathcal{F}(\mathrm{Tr}_{\mathcal{M}}(\bar{\rho}), \mathrm{Tr}_{\mathcal{M}}(\rho_1)) = \mathcal{F}(\tilde{\rho}, \rho_1)$ and $\mathrm{Tr}_{\mathcal{M}}(\tilde{\rho}) = \mathrm{Tr}_{\mathcal{M}}(\rho_1)$. Let $s = \frac{1}{2} \| \mathrm{Tr}_{\mathcal{M}}(\bar{\rho}) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \|_1$ and $t = \frac{1}{2} \| \tilde{\rho} - \bar{\rho} \|_1$. Then we have

$$\begin{aligned} \langle R, \tilde{\rho} \rangle - c &= \langle R, \bar{\rho} \rangle + \langle R, \tilde{\rho} - \bar{\rho} \rangle - c \\ &\geq \frac{1}{2} \| \mathrm{Tr}_{\mathcal{M}}(\bar{\rho}) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \|_1 - \delta - \frac{1}{2} \| \tilde{\rho} - \bar{\rho} \|_1 \\ &= s - t - \delta \geq s - \sqrt{2s - s^2} - \delta \\ &\geq \delta - \sqrt{2\delta - \delta^2} - \delta = -\sqrt{2\delta - \delta^2} \end{aligned}$$

where the first inequality is due to Equation [8] and Lemma 5 and the second inequality comes from Lemma 3. The last inequality is because $s - \sqrt{2s - s^2}$ is decreasing when $0 \leq s \leq 0.2$ and by Equation [8] $s \leq \delta$.

□

Theorem 3. *For any guess c ($0 \leq c \leq 1$) for the feasibility problem, and let the input size denoted by $|x|$ be the size of function f_1, there is a* NC *algorithm to solve the feasibility problem by returning either case in Lemma 6.*

Proof. The algorithm basically follows the Theorem 2 and Lemma 6. Since the particular feasibility problem in our consideration has $\Psi(\cdot) = \mathrm{Tr}_{\mathcal{M}}(\cdot)$ and $A = R, B = \mathrm{Tr}_{\mathcal{M}}(\rho_1)$, we have

$$N(\Pi) = -pR + \mathbb{1}_{\mathcal{M}} \otimes P + (pc - \langle \mathrm{Tr}_{\mathcal{M}}(\rho_1), P \rangle) \mathbb{1}_{\mathcal{M} \otimes \mathcal{V}}$$

according to the definition in Equation [6] where $\Pi = \begin{pmatrix} p & \\ & P \end{pmatrix} \in T_2$ in our problem. It is easy to see that

$$\| N(\Pi) \|_\infty \leq \| -pR \|_\infty + \| \mathbb{1}_{\mathcal{M}} \otimes P \|_\infty + \| (pc - \langle \mathrm{Tr}_{\mathcal{M}}(\rho_1), P \rangle) \mathbb{1}_{\mathcal{M} \otimes \mathcal{V}} \|_\infty \leq 1 + 1 + 1 = 3$$

for any $\Pi \in T_2$ and $\Psi(\rho) = \mathrm{Tr}_{\mathcal{M}}(\rho)$ can be calculated in NC for any $\rho \in T_1$. Thus, by Theorem 2, we can compute the approximate equilibrium value and point in Lemma 6 to precision $\delta = \Omega(1/polylog(|x|))$ in NC. Based on the two cases discussed in Lemma 6, we can either claim the original problem is infeasible or calculate the $\tilde{\rho}$ by a NC algorithm according to Lemma 2. Compose all the NC circuits above, then we have a NC algorithm for the feasibility problem. □

3.2 Solution to the Promised Version and General Case

We are ready to apply the result of Theorem 3 to simulate QIP(2) or more general cases. Recall the definition of QIP(2), there will be two promises with gap $\Delta = c(|x|) - s(|x|) = \Omega(1/poly(|x|))$. Thus,

Corollary 1. QIP(2) $\subseteq$ PSPACE

Proof. For any input x, we simply compose the following circuits.

- For any specific x, compute the corresponding initial state ρ_1 and the function f_1. This can be done in NC(poly) because it only involves the computation of the product of a polynomial number of exponential-size matrices that corresponds to the quantum circuits used by the verifier.
- Choose the guess value $c = \frac{1}{2}(c(|x|) + s(|x|))$ and precision $\delta = \frac{1}{18}\Delta^2$. Then use the NC algorithm implied by Theorem 3 to calculate the equilibrium value $\lambda_1^\star$ to the precision δ.
- Finally, based on the two cases in Lemma 6, we can claim either the optimum value of the SDP is less than c or at least $c - \sqrt{2\delta - \delta^2} \geq c - \frac{1}{3}\Delta$. Then we are able to tell whether $x \in L$.

Due to the facts in Appendix A.2, all the circuits can be composed in NC(poly). Because of the fact NC(poly)=PSPACE [Bor77], we have QIP(2)$\subseteq$PSPACE. □

Furthermore, for the case where no such promise exists we can develop a binary search to approximate the optimum value to high precision.

Theorem 4. *Let x be any instance of SDP (I) and α be the optimum value of SDP (I). There exists a* NC *algorithm which can calculate α to precision $\delta = \Omega(1/polylog(|x|))$. Furthermore, there is a* NC *algorithm to compute ρ_2 such that $\langle R, \rho_2 \rangle \geq \alpha - \delta$.*

Proof. The proof follows from the binary search based on Lemma 6. Start with guess value c, by Theorem 3, we have a NC algorithm to claim either $\alpha < c$ or $\alpha \geq c - \delta$. Thus, by using binary search, we can calculate α to precision δ. Since $\delta = \Omega(1/polylog(|x|))$, there will be at most polynomial-logarithm iterations in the binary search. Therefore, all the circuits above can be composed in NC. □

4 One-round Product QRG

In this section, we demonstrate how the Framework 2 can be applied to real problems. Here we consider the simplified version of quantum refereed game with one round (two turns). The upper bound of the complexity class recognized by the latter model, denoted by QRG(2), becomes more and more interesting after the proof QIP=PSPACE [JJUW09, Wu10]. Particular, it is interesting to see whether QRG(2)=PSPACE while its classical counterpart RG(2) equals PSPACE [FK97].

The general one-round quantum refereed game works as follows. After receiving some input x, the verifier then prepares some quantum messages and send them to both Yes and No provers. After both provers reply with quantum messages, the verifier will base on all the quantum states at his hand to decide whether to accept x or not. Now, let us consider a simplified case where the messages sent to the Yes prover and No prover are product states. We denote all the languages recognized by this procedure by product-QRG(2). At the first sight, this might seem to be a very restricted complexity class. However, by using the techniques from the recent result [BSW10], we can prove the product-QRG(2) contains all the languages which can recognized by the most general model of one-round quantum refereed game except the message sent to Yes prover is only of poly-logarithm size.

Let us formulate the product one-round quantum refereed game in the following way. Since the messages sent to both provers are product states, let $\mathcal{V}_Y$ denote the verifier's private space when interacts with the Yes prover and $\mathcal{Y}$ denote the message space between the verifier and the Yes prover. Similarly, let $\mathcal{V}_N$ denote the verifier's private space when interacts with the No prover and $\mathcal{N}$ denote the message space between the verifier and the No prover. Without lose of generality, we can assume the pure states

$\rho_Y \in \mathrm{D}(\mathcal{V}_Y \otimes \mathcal{Y}), \rho_N \in \mathrm{D}(\mathcal{V}_N \otimes \mathcal{N})$ are the initial states of the verifier given some input x. We will assume the input x is fixed in the following discussion. Then the Yes and No provers will apply some admissable quantum operations Φ_{Yes}, Φ_{No} respectively on part of the density operators ρ_Y, ρ_N. The resultant states will be $\sigma_Y = \Phi_{Yes} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V}_Y)}(\rho_Y), \sigma_N = \Phi_{No} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V}_N)}(\rho_N)$. Finally, the verifier will make some POVM measurement on $\sigma_Y \otimes \sigma_N$ to decide whether to accept or reject. Let R ($0 \leq R \leq \mathbb{1}$) be the POVM which corresponds to the case where Yes prover wins, then *Game Value* $GV(R)$ is defined to be

$$GV(R) = \max_{\sigma_Y \in T_1} \min_{\sigma_N \in T_2} \langle R, \sigma_Y \otimes \sigma_N \rangle = \min_{\sigma_N \in T_2} \max_{\sigma_Y \in T_1} \langle R, \sigma_Y \otimes \sigma_N \rangle$$

where

$$T_1 = \{\sigma \in \mathrm{D}(\mathcal{V}_Y \otimes \mathcal{Y}) : \exists \text{ admissable } \Phi_{Yes} : \mathrm{L}(\mathcal{Y}) \to \mathrm{L}(\mathcal{Y}), \sigma = \Phi_{Yes} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V}_Y)}(\rho_Y)\}$$

and

$$T_2 = \{\sigma \in \mathrm{D}(\mathcal{V}_N \otimes \mathcal{N}) : \exists \text{ admissable } \Phi_{No} : \mathrm{L}(\mathcal{N}) \to \mathrm{L}(\mathcal{N}), \sigma = \Phi_{No} \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V}_N)}(\rho_N)\}$$

Since ρ_Y, ρ_N are pure states, by Lemma 4, we can simplify the definition of T_1, T_2 to

$$T_1 = \{\sigma_Y \in \mathrm{D}(\mathcal{V}_Y \otimes \mathcal{Y}) : \mathrm{Tr}_{\mathcal{Y}}(\sigma_Y) = \mathrm{Tr}_{\mathcal{Y}}(\rho_Y)\}, T_2 = \{\sigma_N \in \mathrm{D}(\mathcal{V}_N \otimes \mathcal{N}) : \mathrm{Tr}_{\mathcal{N}}(\sigma_N) = \mathrm{Tr}_{\mathcal{N}}(\rho_N)\}$$

Definition 2. *Any language L is inside this* product-QRG(2) *if and only if for any input x,*

- *If $x \in L$, then $GV(R) \geq c(|x|)$*
- *If $x \notin L$, then $GV(R) \leq s(|x|)$*

where $c(|x|) - s(|x|) = \Omega(1/poly(|x|))$.

Let $b = \frac{1}{2}(c(|x|) + s(|x|))$ and $\Delta = c(|x|) - s(|x|)$. By applying Framework 2, we define the convex-concave function h_1 as follows.

$$h_1(\{\sigma_Y, \Pi\}, \sigma_N) = \langle R, \sigma_Y \otimes \sigma_N \rangle - b + \frac{2}{\Delta} \langle \Pi, \mathrm{Tr}_{\mathcal{N}}(\sigma_N) - \mathrm{Tr}_{\mathcal{N}}(\rho_N) \rangle$$

where Π comes from the set $T = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{V}_N}\}$. The function h_1 is actually the weighted sum where the factor α is chosen to be $2/\Delta$. Then we will consider the new equilibrium value of function h_1 instead. Precisely, we define

$$\mu^\star = \max_{\{\sigma_Y, \Pi\} \in T_1 \times T} \min_{\sigma_N \in \mathrm{D}(\mathcal{V}_N \otimes N)} h_1(\{\sigma_Y, \Pi\}, \sigma_N) = \min_{\sigma_N \in \mathrm{D}(\mathcal{V}_N \otimes N)} \max_{\{\sigma_Y, \Pi\} \in T_1 \times T} h_1(\{\sigma_Y, \Pi\}, \sigma_N)$$

Then we are ready to show the gap between two promises in Definition 2 can be transferred to the equilibrium value $\mu^\star$.

Theorem 5. *Given the two promises in the Definition 2, we have for any input x,*

- *If $x \in L$, then $\mu^\star \geq \frac{1}{4}\Delta$*
- *If $x \notin L$, then $\mu^\star \leq -\frac{1}{2}\Delta$*

Proof. • If $x \in L$, choose $(\{\sigma_Y^\star, \Pi^\star\}, \sigma_N^\star)$ to be the equilibrium point. By Lemma 5, we have

$$\mu^\star = \langle R, \sigma_Y^\star \otimes \sigma_N^\star \rangle - b + \frac{1}{\Delta} \| \mathrm{Tr}_{\mathcal{N}}(\sigma_N^\star) - \mathrm{Tr}_{\mathcal{N}}(\rho_N) \|_1$$

By Lemma 3, there exists some $\tilde{\sigma_N}$ such that $\mathrm{Tr}_{\mathcal{N}}(\tilde{\sigma_N}) = \mathrm{Tr}_{\mathcal{N}}(\rho_N)$ and $\mathcal{F}(\tilde{\sigma_N}, \sigma_N^\star) = \mathcal{F}(\mathrm{Tr}_{\mathcal{N}}(\sigma_N^\star), \mathrm{Tr}_{\mathcal{N}}(\rho_N))$. Let $s = \frac{1}{2}\|\tilde{\sigma_N} - \sigma_N^\star\|_1$ and $t = \frac{1}{2}\|\mathrm{Tr}_{\mathcal{N}}(\sigma_N^\star) - \mathrm{Tr}_{\mathcal{N}}(\rho_N)\|_1$, thus

$$\begin{aligned}
\mu^\star &= \langle R, \sigma_Y^\star \otimes \tilde{\sigma_N}\rangle - b + \langle R, \sigma_Y^\star \otimes (\sigma_N^\star - \tilde{\sigma_N})\rangle + \frac{1}{\Delta}\|\mathrm{Tr}_{\mathcal{N}}(\sigma_N^\star) - \mathrm{Tr}_{\mathcal{N}}(\rho_N)\|_1 \\
&\geq \frac{1}{2}\Delta - \frac{1}{2}\|\sigma_N^\star - \tilde{\sigma_N}\|_1 + \frac{1}{\Delta}\|\mathrm{Tr}_{\mathcal{N}}(\sigma_N^\star) - \mathrm{Tr}_{\mathcal{N}}(\rho_N)\|_1 = \frac{\Delta}{2} - s + \frac{2}{\Delta}t \\
&\geq \frac{\Delta}{2} - s + \frac{2}{\Delta}(1 - \sqrt{1-s^2}) \geq \frac{\Delta}{2} + \frac{2}{\Delta} - \sqrt{1 + \frac{4}{\Delta^2}} \\
&= \frac{\Delta}{2} + \frac{1 - \sqrt{1 + (\frac{\Delta}{2})^2}}{\frac{\Delta}{2}} \geq \frac{1}{4}\Delta
\end{aligned}$$

where the first inequality comes from Lemma 5 and the second inequality is due to Lemma 3. The third inequality is due to the fact that $\min_{0\leq s\leq 1} \frac{1}{x} - s - \frac{1}{x}\sqrt{1-s^2} = \frac{1}{x} - \sqrt{1 + \frac{1}{x^2}}$. The last inequality comes from the fact $1 - \sqrt{1+x^2} \geq -\frac{1}{2}x^2$ for any $0 \leq x \leq 1$.

- If $x \notin L$, by Definition 2 , there exists a $\tilde{\sigma_N}$ satisfying $\mathrm{Tr}_{\mathcal{N}}(\tilde{\sigma_N}) \leq \mathrm{Tr}_{\mathcal{N}}(\rho_N)$ and for any $\sigma_Y \in T_1$, we have $\langle R, \sigma_Y \otimes \tilde{\sigma_N}\rangle - b \leq -\frac{\Delta}{2}$. Thus,

$$\mu^\star \leq \max_{\{\sigma_Y, \Pi\} \in S_1 \times T} h_1(\{\sigma_Y, \Pi\}, \tilde{\sigma_N}) \leq -\frac{\Delta}{2}$$

□

Thus in order to tell whether x is inside L, it suffices to compute the value of $\mu^\star$ and make the decision according to Theorem 5. In the following proof, we will implicitly make use of the result in Theorem 4 recursively at each iteration as the solution to the oracle.

Corollary 2. product-QRG(2)$\subseteq$ PSPACE

Proof. The proof of this corollary is quite similar to Corollary 1. Whenever an input x is given, we can calculate R in NC(poly) and approximate the equilibrium value $\mu^\star$. By composing all these circuits, we prove the whole circuit is in NC(poly) and thus in PSPACE.

The only difference is the function h_1 is not in the same form as we discussed in Theorem 2. However, by applying the general framework for the equilibrium value (see Appendix A.3), we are able to calculate equilibrium value $\mu^\star$ to high precision in NC as well. Particularly, for each $\sigma_N^{(t)}$ generated, we will choose $\sigma_Y^{(t)}$ to be the return value by Theorem 4 where the POVM for SDP (I) is $\mathrm{Tr}_{\mathcal{V}_N \otimes N}((\mathbb{1}_{\mathcal{V}_Y \otimes Y} \otimes \sqrt{\sigma_N^{(t)}})R(\mathbb{1}_{\mathcal{V}_Y \otimes Y} \otimes \sqrt{\sigma_N^{(t)}}))$. Furthermore, we will choose $\Pi^{(t)}$ to be the projection onto the positive eignspace of $\mathrm{Tr}_{\mathcal{N}}(\sigma_N^{(t)}) - \mathrm{Tr}_{\mathcal{N}}(\rho_N)$. Similarly to the algorithm in Figure 1, we will choose

$$N(\sigma_Y, \Pi) = \mathrm{Tr}_{\mathcal{V}_Y \otimes Y}((\sqrt{\sigma_Y} \otimes \mathbb{1}_{\mathcal{V}_N \otimes N})R(\sqrt{\sigma_Y} \otimes \mathbb{1}_{\mathcal{V}_N \otimes N})) + \frac{2}{\Delta}\Pi \otimes \mathbb{1}_{\mathcal{N}} - (\frac{2}{\Delta}\langle \Pi, \mathrm{Tr}_{\mathcal{N}}(\rho_N)\rangle + b)\mathbb{1}_{\mathcal{V}_N \otimes N}$$

Since $\Delta = \Omega(1/poly(|x|))$, we have $\|N(\sigma_Y, \Pi)\|_\infty$ is always bounded by $O(poly(|x|))$. Thus the whole algorithm can be accomplished in NC(poly).

□

5 Conclusions and Open Problems

In this paper we demonstrate how the Framework 1 can be used to solve a class of SDPs. Moreover, by the examples of QIP(2) and QMAM, we demonstrate how the conversion from approximate to exact feasibility

can be done in our framework. The generic form in Theorem 2 also illustrates the potential of our framework to solve other SDPs. In addition, our example of product-QRG(2) illustrates how the Framework 2 can be used to calculate the equilibrium value of more complicated form.

However, there are several limits and unknown facts about our two frameworks. As mentioned in [JJUW09], it might be impossible to solve any SDPs in NC. Thus, we cannot hope to include all possible SDPs into our framework. Understanding what kind of constraints for SDPs can be solved via our framework is a major open problem. So far, we have positive results when the constraints are partial trace and its simple combination. It can be easily verified that the constraint $\Psi = \mathrm{Tr}_{\mathcal{A}}(U \cdot U^*)$ for some unitary U can also be solved in the similar way.

Another open problem is whether any generalization of the multiplicative weight update method can be found to improve the results based on the multiplicative weight update method. The recent survey paper [Haz10] could provide some insights into that.

Finally, it is still open whether QRG(2)=PSPACE while its classical counterpart RG(2)=PSPACE.

Acknowledgement

The author is grateful to Gus Gutoski, Zhengfeng Ji, Rahul Jain, Richard Cleve, Sarvagya Upadhyay and John Watrous for helpful discussions. Particularly, the penalization idea in Framework 2 is inspired by discussion with Gus Gutoski and the weight factor in the penalization is suggested by John Watrous. The fact that the constraint $\Psi = \mathrm{Tr}_{\mathcal{A}}(U \cdot U^*)$ for some unitary U can also be solved in the similar way is noticed by Zhengfeng Ji. The author is grateful to Gus Gutoski and Zhengfeng Ji for helpful comments on the manuscript. The author also wants to thank the hospitality and invaluable guidance of John Watrous when he was visiting the Institute for Quantum Computing, University of Waterloo. The research was completed during this visit and was supported by the Canadian Institute for Advanced Research (CIFAR).

References

[Ali95] F. Alizadeh. Interior point methods in semidefinite programming with applications to combinatorial optimization. *SIAM Journal on Optimization*, 5(1):13–51, 1995.

[AHK05a] S. Arora, E. Hazan, and S. Kale. Fast algorithms for approximate semidefinite programming using the multiplicative weights update method. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 339–348, 2005.

[AHK05b] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta algorithm and applications. 2005

[AK07] S. Arora and S. Kale. A combinatorial, primal-dual approach to semidefinite programs. In *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, pages 227–236, 2007.

[Bab85] L. Babai. Trading group theory for randomness. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 421–429, 1985.

[BSW10] S. Beigi, P.W. Shor and J. Watrous Quantum interative proofs with short messages availabe at arXiv:1004.0411, 2010.

[BCP83] A. Borodin, S. Cook, and N. Pippenger. Parallel computation for well-endowed rings and space-bounded probabilistic machines. *Information and Control*, 58:113–136, 1983.

[BGH82] A. Borodin, J. von zur Gathen, and J. Hopcroft. Fast parallel matrix and GCD computations. In *Proceedings of the 23rd Annual IEEE Symposium on Foundations of Computer Science*, pages 65–71, 1982.

[Bha97] R. Bhatia. *Matrix Analysis*. Springer, 1997.

[Bor77] A. Borodin. On relating time and space to size and depth. *SIAM Journal on Computing*, 6:733–744, 1977.

[BV04] S. Boyd and L. Vandenberghe *Convex Optimization*. Cambridge University Press, 2004

[dK02] E. de Klerk. *Aspects of Semidefinite Programming – Interior Point Algorithms and Selected Applications*, volume 65 of *Applied Optimization*. Kluwer Academic Publishers, Dordrecht, 2002.

[FK97] U. Feige and J. Kilian. Making games short. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 506–516, 1997.

[Fle00] L.K. Fleischer. Approximating fractional multicommodity flow independent of the number of commodities. *SIAM J. Discret. Math.*, 13(4):505ÍC520, 2000

[FS99] Y. Freund and R.E. Schapire. Adaptive game playing using multiplicative weights. *Games and Economic Behavior* , 29:79–103, 1999.

[Haz06] E. Hazan Approximate Convex Optimization by Online Game Playing. available at arxiv cs.CC/0610119, 2006.

[Haz10] E. Hazan The convex optimization approach to regret minimization. manuscript, 2010.

[Gat93] J. von zur Gathen. Parallel linear algebra. In J. Reif, editor, *Synthesis of Parallel Algorithms*, chapter 13. Morgan Kaufmann Publishers, Inc., 1993.

[GK98] N. Garg and J. Könemann. Faster and simpler algorithms for multicommodity flow and other fractional packing problems. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 300–309, 1998

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 291–304, 1985.

[Gut05] G. Gutoski Upper bounds for quantum interactive proofs with competing provers. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, pages 334–343, 2005.

[GW05] G. Gutoski and J. Watrous Quantum interactive proofs with competing provers. In *Proceedings of the 22th Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer 2005.

[GW07] G. Gutoski and J. Watrous Toward a general theory of quantum games. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 565–574, 2007.

[JJUW09] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP = PSPACE. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, 2010

[JUW09] R. Jain, S. Upadhyay, and J. Watrous. Two-message quantum interactive proofs are in PSPACE. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, 2009.

[JW09] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 24th IEEE Conference on Computational Complexity*, pages 243–253, 2009.

[Kal07] S. Kale. *Efficient algorithms using the multiplicative weights update method*. PhD thesis, Princeton University, 2007.

[Kha04] R. Khandekar *Lagrangian Relaxation based Algorithms for Convex Programming Problems*. PhD thesis, Indian Institute of Technology, Delhi, 2004

[KW00] A. Kitaev and J. Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof system. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 608–617, 2000.

[KSV02] A. Kitaev, A. Shen, M. Vyalyi Classical and Quantum Computation American Mathematical Society, 2002

[Lov03] L. Lovasz. Semidefinite programs and combinatorial optimization. *Recent Advances in Algorithms and Combinatorics*, 2003.

[NC00] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[PST91] S.A. Plotkin, D.B. Shmoys, and T. Tardos. Fast approximation algorithm for fractional packing and covering problems. In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computer Science*, pages 495–504, 1991.

[Ste10] David Steurer. Fast SDP Algorithms for Constraint Satisfaction Problems. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, 2010.

[Rag08] P. Raghavendra. Optimal algorithms and inapproximability results for every CSP? In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 245–254, 2008.

[vN28] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1928):295–320, 1928

[VB96] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, 1996.

[Vaz01] V. Vazirani. *Approximation Algorithms* Springer-Verlag, Berlin, 2001

[Wat08] J. Watrous. Lecture Notes for Theory of Quantum Information Fall 2008.

[Wat09b] J. Watrous. Semidefinite programs for completely bounded norms. *Theory of Computing 5: 11*, 2009.

[WK06] M. Warmuth and D. Kuzmin. 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.

[Wu10] Xiaodi Wu. Equilibrium Value Method for the Proof of QIP=PSPACE availabe at arxiv: 1004.0264

[You95] N.E. Young Randomized rounding without solving the linear program. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 170–178, 1995.

A Extended Preliminaries

A.1 Fundamentals of Quantum Information

In this section, we will provide a summary of the fundamental notations and facts in quantum information. We assume the readers are familiar with these knowledge, and most part of this section is meant to make clear the terminology and well-known facts used in this paper. For those readers who are not familiar with these concepts, we recommend them to refer to [Bha97, KSV02, NC00, Wat08]. Our notation basically follows the notation in Watrous's lecture notes [Wat08].

A *quantum register* refers to a collection of qubits, usually represented by a complex Euclidean spaces of the form $\mathcal{X} = \mathbb{C}^{\Sigma}$ where Σ refers to some finite non-empty set of the possible states.

For any two complex Euclidean spaces $\mathcal{X}, \mathcal{Y}$, let $\mathrm{L}(\mathcal{X}, \mathcal{Y})$ denote the space of all linear mappings(or operators) from $\mathcal{X}$ to $\mathcal{Y}$ ($\mathrm{L}(\mathcal{X})$ short for $\mathrm{L}(\mathcal{X}, \mathcal{X})$). An operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ is a *linear isometry* if $A^*A = \mathbb{1}_{\mathcal{X}}$ where A^* denotes the adjoint(or conjugate transpose) of A. The set of linear isometries is denoted by $\mathrm{U}(\mathcal{X}, \mathcal{Y})$ then.

An operator $A \in \mathrm{L}(\mathcal{X})$ is *Hermitian*, the set of which is denoted by $\mathrm{Herm}(\mathcal{X})$, if $A = A^*$. The eigenvalues of a Hermitian operator are always real. For $n = \dim \mathcal{X}$, we write $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_n(A)$ to denote the eigenvalues of A sorted from largest to smallest. An operator $P \in \mathrm{L}(\mathcal{X})$ is *positive semidefinite*, the set of which is denoted by $\mathrm{Pos}(\mathcal{X})$, if P is Hermitian and all of its eigenvalues are nonnegative, namely $\lambda_n(P) \geq 0$. An operator $\rho \in \mathrm{Pos}(\mathcal{X})$ is a *density operator*, the set of which is denoted by $\mathrm{D}(\mathcal{X})$, if it has trace equal to 1. A density operator $\rho \in \mathrm{D}(\mathcal{X})$ is said to be *pure* if it has rank equal to one. An operator $\Pi \in \mathrm{Pos}(\mathcal{X})$ is a *projection* if Π projects onto some subspace of $\mathcal{X}$. Furthermore, such operators only have eigenvalues of 0 or 1.

The Hilbert-Schmidt inner product on $\mathrm{L}(\mathcal{X})$ is defined by

$$\langle A, B \rangle = \mathrm{Tr}\, A^* B$$

for all $A, B \in \mathrm{L}(\mathcal{X})$.

A *super-operator*(or quantum channel) is a linear mapping of the form

$$\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$$

A super-operator Ψ is said to be *positive* if $\Psi(X) \in \mathrm{Pos}(\mathcal{Y})$ for any choice of $X \in \mathrm{Pos}(\mathcal{X})$, and is *completely positive* if $\Psi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{Z})}$ is positive for any choice of a complex vector space $\mathcal{Z}$. The super-operator Ψ is said to be *trace-preserving* if $\mathrm{Tr}\,\Psi(X) = \mathrm{Tr}\, X$ for all $X \in \mathrm{L}(\mathcal{X})$. A super-operator Ψ is *admissable* if it is completely positive and trace-preserving. Admissable super-operators represent the discrete-time changes in quantum systems that, in principle, can be physically realized.

We refer to *measurements*, or precisely POVM-type measurements as a collection of positive semidefinite operators

$$\{P_a : a \in \Sigma\} \subset \mathrm{Pos}(\mathcal{X})$$

satisfying the constraint $\sum_{a \in \Sigma} P_a = \mathbb{1}_{\mathcal{X}}$. Here Σ refers to a finite, nonempty set of *measurement outcomes*. If a quantum state represented by $\rho \in \mathrm{D}(\mathcal{X})$ is measured with respect to this measurement, then each outcome $a \in \Sigma$ will be observed with probability $\langle P_a, \rho \rangle$.

The *trace norm* of an operator $A \in \mathrm{L}(\mathcal{X})$ is denoted by $\|A\|_1$ and defined to be

$$\|A\|_1 = \mathrm{Tr}\sqrt{A^*A}$$

When A is Hermitian, we have

$$\|A\|_1 = \max\{\langle P_0 - P_1, A \rangle : P_0, P_1 \in \mathrm{Pos}(\mathcal{X}), P_0 + P_1 = \mathbb{1}_{\mathcal{X}}\} \tag{9}$$

The *spectral norm* of an operator $A \in \mathrm{L}(\mathcal{X})$ is defined to be

$$\|A\|_\infty = \max\{\|Au\| : u \in \mathcal{X}, \|u\| = 1\}$$

Given two positive semidefinite operators $P, Q \in \mathrm{Pos}(\mathcal{X})$, we define the *fidelity* between P and Q as

$$\mathcal{F}(P, Q) = \|\sqrt{P}\sqrt{Q}\|_1$$

When P, Q are density operators, due to Fuchs-van de Graaf inequality, we have

$$1 - \frac{1}{2}\|P - Q\|_1 \leq \mathcal{F}(P, Q) \leq \sqrt{1 - \frac{1}{4}\|P - Q\|_1^2} \tag{10}$$

Suppose $\rho \in \mathrm{D}(\mathcal{X})$ is a density operator, the *purification* of ρ in $\mathcal{X} \otimes \mathcal{Y}$ is any pure density operator $uu^* \in \mathrm{D}(\mathcal{X} \otimes \mathcal{Y})$ for which $\mathrm{Tr}_{\mathcal{Y}}(uu^*) = \rho$.

The following contains the proof of lemmas shown in Section 2.1

Proof of Lemma 1 (Please note that this lemma was originally proved in many places. The following proof follows the one in [JUW09]. The only reason to include this proof is because we will use it to prove Lemma 2.)

Proof. First, by the monotonicity of the fidelity function under partial trace, we have for any $\sigma_2 \in \mathrm{D}(\mathcal{A}\otimes\mathcal{B})$ such that $\mathrm{Tr}_B\,\sigma_2 = \rho_2$ the inequality $\mathcal{F}(\rho_1,\rho_2) \geq \mathcal{F}(\sigma_1,\sigma_2)$ always holds. Thus, it suffices to show that equality can be achieved.

Let $V \in \mathrm{U}(\mathcal{A})$ such that $\sqrt{\rho_1}\sqrt{\rho_2}V$ is positive semidefinite. Since for fidelity function we have $\mathcal{F}(\rho_1,\rho_2) = \|\sqrt{\rho_1}\sqrt{\rho_2}\|_1$, then for such a V it holds that $\mathcal{F}(\rho_1,\rho_2) = \mathrm{Tr}(\sqrt{\rho_1}\sqrt{\rho_2}V)$. Now let $\mathcal{C} = \mathcal{A}\otimes\mathcal{B}$ and $|u_1\rangle \in \mathcal{A}\otimes\mathcal{B}\otimes\mathcal{C}$ be the purification of σ_1, in particular, $|u_1\rangle$ is chosen to be

$$|u_1\rangle = \mathrm{vec}(\sqrt{\sigma_1})$$

By rearranging the coefficients we can find a $X \in \mathrm{L}(\mathcal{B}\otimes\mathcal{C},\mathcal{A})$ such that $\mathrm{vec}(X) = |u_1\rangle$. Since $|u_1\rangle$ is also a purification of ρ_1, there must exist a linear isometry $U \in \mathrm{U}(\mathcal{A},\mathcal{B}\otimes\mathcal{C})$ such that,

$$X = \sqrt{\rho_1}U^*$$

Finally, let $|u_2\rangle = \mathrm{vec}(\sqrt{\rho_2}VU^*) \in \mathcal{A}\otimes\mathcal{B}\otimes\mathcal{C}$ where V, U are obtained above respectively. It is easy to see that $|u_2\rangle$ is a purification of ρ_2 in the space $\mathcal{A}\otimes\mathcal{B}\otimes\mathcal{C}$. Thus, we choose $\sigma_2 = \mathrm{Tr}_{\mathcal{C}}(|u_2\rangle\langle u_2|)$ and it will hold that

$$\mathcal{F}(\sigma_1,\sigma_2) \geq |\langle \mathrm{vec}(\sqrt{\rho_1}U^*), \mathrm{vec}(\sqrt{\rho_2}VU^*)\rangle| = |\langle \sqrt{\rho_1}U^*, \sqrt{\rho_2}VU^*\rangle| = \mathrm{Tr}(\sqrt{\rho_1}\sqrt{\rho_2}V) = \mathcal{F}(\rho_1,\rho_2)$$

□

Proof of Lemma 2

Proof. The proof of the Lemma 1 actually gives you a way to construct such a σ_2 given ρ_1,ρ_2,σ_1. Let us review the important steps in the proof again with more attention to the computation of each intermediate quantity.

In the first step, we need to calculate a $V \in \mathrm{U}(\mathcal{A})$ such that $\sqrt{\rho_1}\sqrt{\rho_2}V$ is positive semidefinite. This can be done by calculating the singular value decomposition of $\sqrt{\rho_1}\sqrt{\rho_2}$ and let $V = \mathbb{1} - 2P$ where P is the projection onto the subspace with negative singular values.

The second step calculates X such that $\mathrm{vec}(X) = |u_1\rangle = \mathrm{vec}(\sqrt{\sigma_1})$. This can be done by simply rearranging the coefficients in the entries of $\sqrt{\sigma_1}$. In order to get $U \in \mathrm{U}(\mathcal{A},\mathcal{B}\otimes\mathcal{C})$, we can calculate the singular value decomposition of $\sqrt{\rho_1}$ and get the inverse(or pseudo-inverse) of $\sqrt{\rho_1}$. Then $U = X^*(\sqrt{\rho_1}^{-1})^*$.

Once we have U and V, we can easily calculate σ_2 by using the formula

$$\sigma_2 = \mathrm{Tr}_{\mathcal{C}}(\mathrm{vec}(\sqrt{\rho_2}VU^*)\,\mathrm{vec}(\sqrt{\rho_2}VU^*)^*)$$

Due to the fact that fundamental operations of matrix and the singular value decomposition can be done in NC(see Fact 1 and Fact 3) and the fact we can compose these NC circuits easily, then we can conclude that σ_2 can be calculated in NC given the classical representations of ρ_1,ρ_1 and σ_1 as input. □

Proof of Lemma 3

Proof. This is only a simple application of the Fuchs-van de Graaf Inequalities(Eq[10]). Namely, we have

$$1-s \leq \mathcal{F}(\rho_1,\rho_2) \leq \sqrt{1-s^2} \text{ and } 1-t \leq \mathcal{F}(\sigma_1,\sigma_2) \leq \sqrt{1-t^2}$$

Given the fact $\mathcal{F}(\rho_1,\rho_2) = \mathcal{F}(\sigma_1,\sigma_2)$, it easily follows that

$$1-s \leq \sqrt{1-t^2} \text{ and } 1-t \leq \sqrt{1-s^2}$$

□

Proof of Lemma 4

Proof. This lemma is a standard fundamental fact of quantum information that follows from the unitary equivalence of purifications and the fact ρ_1 is a pure state. □

Proof of Lemma 5

Proof. Given the fact about the trace norm of any Hermitian operator A in Equation 9, we have

$$\begin{aligned} \|A\|_1 &= \max_{\Pi:0\leq\Pi\leq\mathbb{1}} \langle A, \Pi - (\mathbb{1}-\Pi)\rangle \\ &= \max_{\Pi:0\leq\Pi\leq\mathbb{1}} 2\langle A, \Pi\rangle - \mathrm{Tr}(A) \\ &= \max_{\Pi:0\leq\Pi\leq\mathbb{1}} 2\langle A, \Pi\rangle \end{aligned}$$

Thus,

$$\max_{\Pi:0\leq\Pi\leq\mathbb{1}} \langle A, \Pi\rangle = \frac{1}{2}\|A\|_1$$

For the other equation, choose $A' = -A$, and apply the equation above, we have

$$\max_{\Pi:0\leq\Pi\leq\mathbb{1}} \langle A', \Pi\rangle = \max_{\Pi:0\leq\Pi\leq\mathbb{1}} \langle -A, \Pi\rangle = \frac{1}{2}\|A'\|_1 = \frac{1}{2}\|A\|_1$$

Thus

$$\min_{\Pi:0\leq\Pi\leq\mathbb{1}} \langle A, \Pi\rangle = -\frac{1}{2}\|A\|_1$$

□

A.2 Facts on NC and parallel matrix computations

We denote by NC the class of promise problems computed by the logarithmic-space uniform Boolean circuits with poly-logarithmic depth. Furthermore, we denote by NC(poly) the class of promise problems computed by the polynomial-space uniform Boolean circuits with polynomial depth. Since it holds that NC(poly)= PSPACE [Bor77], thus in order to simulate the algorithm above in PSPACE, it suffices to prove that we can simulate the algorithm in NC(poly).

There are a few facts about these classes which are useful in our discussion. The first fact is the functions in these classes compose nicely. It is clear that if $f \in$ NC(poly) and $g \in$ NC, then their composition $g \circ f$ is in NC(poly), which follows from the most obvious way of composing the families of circuits. Another useful fact is that many computations involving matrices can be performed by NC algorithms (Please refer to the survey [Gat93] which describes NC algorithms for these tasks). Especially, we will make use of the fact that matrix exponentials and singular value decompositions can be approximated to high precision in NC. We will directly cite the well-prepared form of these facts in [JJUW09].

Fact 1. *Fundamental operations like addition, multiplication of matrices can be done in* NC

Fact 2. *Matrix exponentials: there exists* NC *algorithms such that*

Input: *An $n \times n$ matrix M, a positive rational number η, and an integer k expressed in unary notation (i.e., 1^k).*
Promise: $\|M\| \leq k$.
Output: *An $n \times n$ matrix X such that $\|\exp(M) - X\| < \eta$.*

Fact 3. *Singular value decompositions: there exists* NC *algorithms such that*

Input: *An $m \times n$ matrix M and a positive rational number η.*
Output: *An $m \times m$ unitary matrix U ,$n \times n$ unitary matrix V and an $m \times n$ real diagonal matrix Λ such that*

$$\|M - U\Lambda V^*\| < \eta.$$

1. Initialization: Pick a fixed $\varepsilon \leq \frac{1}{2}$, and let $W^{(1)} = \mathbb{1}_{\mathcal{X}} \in \mathrm{L}(\mathcal{X})$, $D = \dim \mathcal{X}$.
2. Repeat for each $t = 1, \ldots, T$:
   (a) Let the density operator $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}$
   (b) Observe the loss matrix $M^{(t)} \in \mathrm{L}(\mathcal{X})$ which satisfies $-\mathbb{1}_{\mathcal{X}} \leq M^{(t)} \leq 0$ or $0 \leq M^{(t)} \leq \mathbb{1}_{\mathcal{X}}$, update the weight matrix as follows:

$$W^{(t+1)} = exp(-\varepsilon \sum_{\tau=1}^{t} M^{(\tau)})$$

Figure 2: The Matrix Multiplicative Weights Update method.

A.3 Multiplicative Weights Update Method

The *multiplicative weights update method* introduced in Section 1 is a framework for algorithm design(or meta-algorithm) that works as the one shown in Fig 2. This kind of framework involves lots of technical details and we refer the curious reader to the survey and the PhD thesis [Kal07] mentioned in the introduction. However, for the sake of completeness, we provide the main result which will be useful in our proof. It should be noticed that $\{M^{(t)}\}$ is the freedom we have in this framework.

Theorem 6. *Assume* $0 \leq M^{(t)} \leq \mathbb{1}$ *for all* t*, after* T *rounds, the algorithm in Fig 2 guarantees that, for any* $\rho^* \in \mathrm{D}(\mathcal{X})$*, we have*

$$(1-\epsilon) \sum_{t=1}^{T} \left\langle \rho^{(t)}, M^{(t)} \right\rangle \leq \left\langle \rho^*, \sum_{t=1}^{T} M^{(t)} \right\rangle + \frac{lnD}{\epsilon} \tag{11}$$

The proof can be found in Kale's thesis [Kal07] or the appendix of [Wu10]. We will then discuss how this method can used to solve the feasibility problem and equilibrium value following the way in Kale's thesis [Kal07] with more details.

In order to solve any feasibility problem of general form, the primal-dual method will generate a series of candidate solutions $X^{(1)}, X^{(2)}, \cdots, X^{(T)}$ for T rounds. For any $X^{(t)}$ in the round t, we require an oracle $\mathcal{O}_1$ to solve the following problem

$$\text{find } Y^{(t)} \text{ s.t. } \left\langle \Psi^*(Y^{(t)}) - A, X^{(t)} \right\rangle \geq 0, \left\langle B, Y^{(t)} \right\rangle \leq c, Y^{(t)} \in \mathrm{Pos}(\mathcal{Y}) \tag{12}$$

The oracle $\mathcal{O}_1$ will return such a $Y^{(t)}$ or claim such a $Y^{(t)}$ does not exist. If such a $Y^{(t)}$ exists, the primal-dual method will generate the $X^{(t+1)}$ via the multiplicative weight update method by choosing $M^{(t)}$ to be a renormalized version of $\Psi^*(Y^{(t)}) - A$. Otherwise, it stops to claim the rescaled $X^{(t)}$ is feasible to the primal problem and $\left\langle A, X^{(t)} \right\rangle \geq c$. If the method does not stop for T rounds, the multiplicative weight update method will generate an approximate dual feasible solution Y. Under certain conditions, such a Y can be converted into exact dual feasible solution $\tilde{Y}$ such that $\left\langle B, \tilde{Y} \right\rangle \leq (1+\epsilon)c$. Due to the duality of SDPs, this implies that $\alpha \leq \beta \leq (1+\epsilon)c$.

In addition to the convertibility from approximate to exact feasibility, another difficulty in applying this generic method is the design of the oracle $\mathcal{O}_1$. Efficient solution to the oracle $\mathcal{O}_1$ is necessary to guarantee an efficient algorithm for the feasibility problem. We will refer this requirement as the *efficient solvability*. In addition, we need the spectrum of $\Psi^*(Y^{(t)}) - A$ of the oracle $\mathcal{O}_1$ is bounded within a small range. We will refer this requirement as *width-boundness*.

The generic framework to calculate the equilibrium value is different in the sense of the design of the oracle and the use of Theorem 6. Consider the value λ,

$$\lambda = \min_{x \in X} \max_{y \in Y} f(x,y) = \max_{y \in Y} \min_{x \in X} f(x,y)$$

for some *convex-concave* function over $X \times Y$ where X, Y are convex compact sets. By convex-concave, we mean

Definition 3. *A function f on $X \times Y$ is* convex-concave *if for every $y \in Y$ the function $\forall x \in X, f_y(x) \triangleq f(x, y)$ is convex on X and for every $x \in X$ the function $\forall y \in Y, f_x(y) \triangleq f(x, y)$ is concave on Y.*

Consider the case when X is the set of density operators up to some factor. Again, we will generate a series of $x^{(1)}, x^{(2)}, \cdots, x^{(T)} \in X$ for T rounds. For each $x^{(t)}$ in the round t, we require another oracle $\mathcal{O}_2$ to find approximate solution, denoted by $y^{(t)}$, to the following optimization problem,

$$\max_{y \in Y} f(x^{(t)}, y) \tag{13}$$

Then $x^{(t+1)}$ will be generated via the multiplicative weight update method. After T rounds, we can claim $\frac{1}{T}\sum_{t=1}^{T} f(x^{(t)}, y^{(t)})$ is the approximate value for λ. Similarly, the oracle $\mathcal{O}_2$ is required to be *efficient solvable* and *width-bounded*. By the efficient solvability, we mean the oracle $\mathcal{O}_2$ can be solved efficiently. By the width-boundness, we mean the $\mathcal{L}_\infty$ norm of $y^{(t)}$ is bounded in some sense.

We will conclude this section with the proof of Theorem 2.

Proof of Theorem 2

Proof. First note that $\|N^{(t)}\|_\infty \leq r$ for any $1 \leq t \leq T$. Thus,

$$0 \leq M^{(t)} = (N^{(t)} + r\mathbb{1}_{\mathcal{X}})/2r \leq \mathbb{1}_{\mathcal{X}}$$

Then it is easy to see this is a typical multiplicative weights update method. Due to the Theorem 6, we have:

$$(1-\varepsilon)\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, M^{(\tau)}\right\rangle \leq \left\langle \rho^*, \sum_{\tau=1}^{T} M^{(\tau)}\right\rangle + \frac{\ln D}{\varepsilon} \tag{14}$$

for any density operator $\rho^* \in \mathrm{D}(\mathcal{X})$. Substitute $M^{(t)} = (N^{(t)} + r\mathbb{1}_{\mathcal{X}})/2r$ into Equation [14] and divide both side by T, note that $\left\langle \rho^{(t)}, M^{(t)}\right\rangle \leq 1$, then we have

$$\frac{1}{T}\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, N^{(\tau)}\right\rangle \leq \frac{1}{T}\left\langle \rho^*, \sum_{\tau=1}^{T} N^{(\tau)}\right\rangle + 2r\varepsilon + \frac{2r\ln D}{\varepsilon T} \tag{15}$$

By choosing $\varepsilon = \frac{\delta}{4r}$ and $T = \left\lceil \frac{16r^2 \ln D}{\delta^2} \right\rceil$, we have

$$\frac{1}{T}\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, N^{(\tau)}\right\rangle \leq \frac{1}{T}\left\langle \rho^*, \sum_{\tau=1}^{T} N^{(\tau)}\right\rangle + \delta$$

According to the definition of $N^{(t)}, 1 \leq t \leq T$, we have

$$\frac{1}{T}\sum_{\tau=1}^{T}\left\langle S(\rho^{(t)}), \Pi^{(t)}\right\rangle \leq \frac{1}{T}\sum_{\tau=1}^{T}\left\langle S(\rho^*), \Pi^{(t)}\right\rangle + \delta \tag{16}$$

Choose $(\rho^\star, \Pi^\star)$ to be the *equilibrium point* and substitute $\rho^\star$ into Equation [16], we have

$$\frac{1}{T}\sum_{\tau=1}^{T}\left\langle S(\rho^{(t)}), \Pi^{(t)}\right\rangle \leq \frac{1}{T}\sum_{\tau=1}^{T}\left\langle S(\rho^\star), \Pi^{(t)}\right\rangle + \delta \leq \lambda^\star + \delta \tag{17}$$

where the last inequality comes from Equation [4].

Since $\bar{\rho} = \frac{1}{T}\sum_{\tau=1}^{T}\rho^{(\tau)}$ and by definition of $\bar{\Pi}$, we have

$$\langle \bar{\rho}, \bar{\Pi} \rangle = \frac{1}{T}\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, \bar{\Pi} \right\rangle \leq \frac{1}{T}\sum_{\tau=1}^{T}\left\langle \rho^{(\tau)}, \Pi^{(\tau)} \right\rangle \leq \lambda^{\star} + \delta$$

where the first inequality is due to the definition of $\Pi^{(t)}$ and the second inequality comes from Equation [16]. On the other side, $\langle \bar{\rho}, \bar{\Pi} \rangle \geq \lambda^{\star}$ by the definition of $\bar{\Pi}$ and Equation [4]. Thus, we have

$$\lambda^{\star} \leq \langle \bar{\rho}, \bar{\Pi} \rangle \leq \lambda^{\star} + \delta$$

Finally we need to show that this algorithm can actually run in NC if $\delta_2 = O(1/polylog(|x|))$. Consider every iteration of the algorithm. The only operations involved are the fundamental operation of matrices, the singular value decomposition and the exponentials of matrices. Due to the Fact 1,2,3, they all can be computed in NC and the circuits for the computation can be easily composed. Since there are only $T = \left\lceil \frac{16r^2 \ln N}{\delta^2} \right\rceil$ iterations and $r = O(polylog(|x|)), \delta = O(1/polylog(|x|))$, thus there will be at most poly-logarithm iterations respect to the input size. Therefore, the whole circuit will be NC. □

B QMAM case

In this section, we will demonstrate how the Framework 1 can be applied to the SDP of QMAM. Due to space limit, we will directly describe the SDP used in [JJUW09] and denote it by SDP (II).

SDP Problem		Feasibility Problem	
maximize:	$\langle R, \rho \rangle$	ask whether:	$\langle R, \rho \rangle \geq c$
subject to:	$\mathrm{Tr}_{\mathcal{Y}}(\rho) \leq \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma$,	subject to:	$\mathrm{Tr}_{\mathcal{Y}}(\rho) \leq \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma$,
	$\rho \in \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y}), \sigma \in \mathrm{D}(\mathcal{X})$		$\rho \in \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y}), \sigma \in \mathrm{D}(\mathcal{X})$

where R ($0 \leq R \leq \mathbb{1}_{\mathcal{X}}$) is a POVM measurement and the space $\mathcal{A}$ is of dimension 2. We need to design an algorithm to distinguish between the following two promises. Let α be the optimum value of the SDP (II).

Definition 4. *Any language L is inside* QMAM *if and only if*

- *If $x \in L$, $\alpha \geq c(|x|)$.*
- *If $x \notin L$, $\alpha \leq s(|x|)$.*

where $c(|x|) - s(|x|) = \Omega(1/poly(|x|))$.

Following the Framework 1, we consider the feasibility problem above. Precisely, we define

$$f_2(\{\rho, \sigma\}, \Pi) = \left\langle \begin{pmatrix} c - \langle R, \rho \rangle & \\ & \mathrm{Tr}_{\mathcal{Y}}(\rho) - \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma \end{pmatrix}, \Pi \right\rangle \tag{18}$$

where $\{\rho, \sigma\} \in T_1 = \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y}) \times \mathrm{D}(\mathcal{X})$ and $\Pi \in T_2 = \{\Pi : 0 \leq \Pi \leq \mathbb{1}_{\mathcal{A}\otimes\mathcal{X}\oplus\mathbb{C}}\}$. Let $\lambda_2^{\star}$ be the equilibrium value of function f_2, namely,

$$\lambda_2^{\star} = \min_{\{\rho,\sigma\}\in T_1}\max_{\Pi\in T_2} f_2(\{\rho, \sigma\}, \Pi) = \max_{\Pi\in T_2}\min_{\{\rho,\sigma\}\in T_1} f_2(\{\rho, \sigma\}, \Pi)$$

Base on the Theorem 1, the value of $\lambda_1^{\star}$ will imply whether the original problem is feasible. In order to tell the two promises in Definition 4, we will choose the guess value $c = \frac{1}{2}(c(|x|) + s(|x|))$.

Lemma 7. *Given the two promises in Definition 4, we have*

- *If $x \in L$, then $\lambda_2^{\star} \leq 0$.*

- *If $x \notin L$, then $\lambda_2^\star \geq \frac{1}{8}\Delta^2$.*

where $\Delta = c(|x|) - s(|x|)$.

Proof.
- If $x \in L$, then there exists a $\rho \in \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y}), \sigma \in \mathrm{D}(\mathcal{X})$ such that $\langle \rho, R \rangle \geq c$ and $\mathrm{Tr}_{\mathcal{Y}}(\rho) \leq \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma$. This implies $\lambda_2^\star \leq 0$.
- Otherwise, let $(\{\rho^\star, \sigma^\star\}, \Pi^\star)$ be the equilibrium point. Due to Lemma 5, we have

$$\lambda_2^\star = f_2(\{\rho^\star, \sigma^\star\}, \Pi^\star) = \max\{c - \langle R, \rho^\star \rangle, 0\} + \frac{1}{2}\|\mathrm{Tr}_{\mathcal{Y}}(\rho^\star) - \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma\|_1 \tag{19}$$

By Lemma 1, there exists a $\tilde{\rho} \in \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y})$ such that $\mathcal{F}(\frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma, \mathrm{Tr}_{\mathcal{Y}}(\rho^\star)) = \mathcal{F}(\tilde{\rho}, \rho^\star)$ and $\mathrm{Tr}_{\mathcal{Y}}(\tilde{\rho}) = \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma$. Let $s = \frac{1}{2}\|\mathrm{Tr}_{\mathcal{Y}}(\rho^\star) - \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma\|_1$ and $t = \frac{1}{2}\|\tilde{\rho} - \rho^\star\|_1$. Then if $t \leq \frac{1}{2}\Delta$, we have

$$\begin{aligned}
\lambda_2^\star &\geq c - \langle R, \tilde{\rho} \rangle + \langle R, \tilde{\rho} - \rho^\star \rangle + s \\
&\geq \frac{1}{2}\Delta - t + s \\
&\geq \frac{1}{2}\Delta - t + 1 - \sqrt{1 - t^2} \\
&\geq \frac{1}{2}\Delta - \frac{1}{2}\Delta + 1 - \sqrt{1 - \frac{1}{4}\Delta^2} \geq \frac{1}{8}\Delta^2
\end{aligned}$$

where the first inequality is due to Equation [19], the second inequality comes from Lemma 5 and the third inequality comes from Lemma 3. The last inequality is because $t + \sqrt{1 - t^2}$ is increasing when $0 < t < \frac{1}{2}$ and $1 - \sqrt{1 - x^2} \geq \frac{1}{2}x^2$ for any $0 < x < 1$. On the other side, if $t \geq \frac{1}{2}\Delta$, by Equation [19],

$$\lambda_2^\star \geq s \geq 1 - \sqrt{1 - t^2} \geq \frac{1}{2}t^2 \geq \frac{1}{8}\Delta^2$$

Finally, we have $\lambda_2^\star \geq \frac{1}{8}\Delta^2$ in this case.

□

The only part left is to prove that we can calculate the equilibrium value $\lambda_2^\star$ to high precision in NC. As the readers might notice, the set T_1 is no longer a simple set of density operators but a cross product of two sets of density operators. However, we are still able to use a modified version of the algorithm in Figure 1 to solve the problem.

Precisely, we claim the algorithm in Figure 3 will be able to calculate $\lambda_2^\star$ to precision δ in NC. The proof is almost the same as the proof for Theorem 2. The only difference is that we need to update $\rho^{(t)}, \sigma^{(t)}$ independently and get two inequalities from each update. Then we combine them to get the final result. This can be done because

$$f_2(\{\rho, \sigma\}, \Pi) = \langle S_1(\rho), \Pi \rangle + \langle S_2(\sigma), \Pi \rangle$$

where

$$S_1(\rho) = \begin{pmatrix} c - \langle R, \rho \rangle & 0 \\ 0 & \mathrm{Tr}_{\mathcal{Y}}(\rho) \end{pmatrix} \text{ and } S_2(\sigma) = \begin{pmatrix} 0 & 0 \\ 0 & -\frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma \end{pmatrix}$$

Let $\Pi = \begin{pmatrix} p & \\ & P \end{pmatrix}$ again, we can choose

$$N_1(\Pi) = -pR + P \otimes \mathbb{1}_{\mathcal{Y}} + pc\mathbb{1}_{\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y}} \text{ and } N_2(\Pi) = -\frac{1}{2}\,\mathrm{Tr}_{\mathcal{A}}\, P$$

It is easy to verify that $\|N_1(\Pi)\|_\infty, \|N_2(\Pi)\|_\infty$ is bounded by 3. Thus given the gap of the equilibrium value $\lambda_2^\star$ between two promises in Lemma 7, we can distinguish them in NC(poly) namely PSPACE.

1. Let $\varepsilon = \frac{\delta}{4r}$ and $T = \left\lceil \frac{16r^2 \ln D}{\delta^2} \right\rceil$. Also let $W^{(1)} = \mathbb{1}_{\mathcal{A}\otimes\mathcal{X}\otimes\mathcal{Y}}$, $V^{(1)} = \mathbb{1}_{\mathcal{X}}$.
2. Repeat for each $t = 1, \ldots, T$:
   (a) Let $\rho^{(t)} = W^{(t)} / \operatorname{Tr} W^{(t)}, \sigma^{(t)} = V^{(t)} / \operatorname{Tr} V^{(t)}$ and let $\Pi^{(t)}$ be the projection onto the positive eigenspace of $S_1(\rho^{(t)}) + S_2(\sigma^{(t)})$.
   (b) Let $M_1^{(t)} = (N_1(\Pi^{(t)}) + r\mathbb{1}_{\mathcal{A}\otimes\mathcal{X}\otimes\mathcal{Y}})/2r$, $M_2^{(t)} = (N_2(\Pi^{(t)}) + r\mathbb{1}_{\mathcal{X}})/2r$ and update the weight matrix as follows:

$$W^{(t+1)} = exp(-\varepsilon \sum_{\tau=1}^{t} M_1^{(\tau)})$$
$$V^{(t+1)} = exp(-\varepsilon \sum_{\tau=1}^{t} M_2^{(\tau)})$$

3. Return $\frac{1}{T}\sum_{\tau=1}^{T} \left\langle S_1(\rho^{(t)}) + S_2(\sigma^{(t)}), \Pi^{(t)} \right\rangle$ as the approximate equilibrium value of $\lambda_2^\star$.

Figure 3: An algorithm that computes the approximate value of $\lambda_2^\star$ to precision δ.

C Precision Issue

The discussions on precision issue about the parallel implementations in our paper are quite similar to the arguments used in [JJUW09, JUW09, JW09, Wu10]. Again, we will make use the three facts in Appendix A.2 and truncate the computation to sufficient precision for each step. It should be noticed that because of our new framework we will only use three types of operations of matrices (namely, the three facts in Appendix A.2). The analysis of the precision issue in our paper then should be easier than the one in [JJUW09] since the latter one involves other types of operations, like the inversion of matrices.