On Applications of the Equilibrium Value Method

Serena Booth

December 12th, 2014

Abstract

In 2010, Jain et al. [JJUW10] provided the first proof that QIP = PSPACE. Jain et al. used a constructive argument which applied the matrix multiplicative weight update method to a semidefinite programming problem formulation of a QMAM protocol to achieve this proof. The proof that QIP = PSPACE was alternately characterized by Xiaodi Wu [Wu10] that same year; Xiaodi Wu's method, however, was simplified through the application of the equilibrium value method. In this survey, we evaluate the equilibrium value method and its flexibility as a proof technique. We consider Gutoski and Watrous's proof that QRG = RG = EXP [GW07]. We look at Gutoski and Wu's proof that QRG(2) = PSPACE [GW10, GW12], which uses the equilibrium value method. Finally, we reason about the feasibility of providing an alternate characterization that QRG = RG = EXP using the equilibrium value method.

1 Introduction

Watrous first extended the class IP, which consists of those languages which can be decided by an interactive proof system and was shown to be equal to PSPACE [Sha92], to its quantum analog, QIP, in 1999 [Wat03]. The complexity upper bound on this class was unknown until 2010, when Jain et al. [JJUW10] and Wu [Wu10] provided proofs that QIP = IP = PSPACE. This result is extremely enticing as it demonstrates that for interactive proof systems, the power of quantum computation is no greater than the power of classical computation.

Similarly, for the domain of refereed games, quantum computation offers no additional power over classical computation. Refereed games consist of the languages which can be decided by an alternate interactive proof system with two competing provers. Gutoski and Watrous demonstrated that the class of quantum refereed games, QRG, is exactly characterized by QRG = RG = EXP [GW07].

In this paper, we compare the two versions of the proof QIP = PSPACE. We postulate that it is possible to provide a characterization of QRG = EXP using the equilibrium value method used in Wu's proof that QIP = PSPACE. We provide evidence of this by discussing Gutoski and Wu's proof that QRG(2) = PSPACE [GW10], which does indeed use the equilibrium value method. In short, in this survey we discuss the application of the equilibrium value method as a proof technique for simplified applications of the matrix multiplicative weight update method.

2 Definitions

In this section, we provide formal definitions of the complexity classes QIP and QRG, as well as the QIP-complete problem of quantum circuit distinguishability (QCD). We also discuss semidefinite programming, the matrix multiplicative weight update method (MMW), and the equilibrium value method.

Definition 2.1 QIP
We define a verifier $V : \Sigma^* \rightarrow \Sigma^*$ to be a quantum circuit which can compute a quantum polynomial-time algorithm; hence V decides a language L' in BQP.

We define a prover $P : x \in \Sigma^* \rightarrow l(|x|)$, where $l(|x|)$ is defined to be a tuple of quantum circuits. The prover P is said to have unbounded computational power.

Let all provers P and the verifier V each exchange m-messages. A language L can be decided in QIP if and only if the following completeness and soundness conditions hold:

$$\begin{gathered}(\text{Completeness})\ x \in L \Longrightarrow \exists P,\ \Pr[(P,V)(x) = 1] \geq 2/3 \\ (\text{Soundness})\ x \notin L \Longrightarrow \forall P,\ \Pr[(P,V(x) = 1] \leq 1/3\end{gathered} \tag{1}$$

Given the restriction of this protocol to m-messages, this protocol is said to belong to the class QIP(m). QIP(1) is exactly the class QMA.

The full complexity class QIP is restricted to $k(m)$-messages, where $k(m)$ is polynomial in m.

Parallelization and amplification of QIP protocols were demonstrated by Kitaev and Watrous [KW00].

Definition 2.2 QRG
The definition of the complexity class QRG is extremely similar to that of QIP; the difference is that QRG has two provers, P_Y and P_N, both of unbounded computational power, whereby the "yes" prover,

P_Y, aims to cause the verifier to accept, while the "no" prover P_N aims to cause the verifier to reject. A language L can be decided in QRG if the following completeness and soundness conditions hold:

$$\text{(Completeness) } x \in L \Longrightarrow \exists P_Y \forall P_N,\ \Pr[(P_Y, P_N, V)(x) = 1] \geq \frac{2}{3} \tag{2}$$
$$\text{(Soundness) } x \notin L \Longrightarrow \exists P_N \forall P_Y,\ \Pr[(P_N, P_Y, V(x) = 1] \leq \frac{1}{3}$$

Definition 2.3 QCD The *quantum circuit distinguishability* (or QCD) problem for $0 < \epsilon < 1$, $\text{QCD}_{2-\epsilon,\epsilon}$ is QIP-complete [RW05]. This problem is most generally parametrized by two constants a and b, where $0 \leq b < a \leq 2$. The diamond norm, a measure of the distance between two quantum channels A and B, is denoted as $||A - B||_\diamond$. The problem $\text{QCD}_{a,b}$ is defined as follows:

$$\Pi = (\Pi_Y, \Pi_N) \tag{3}$$

On input (Q_0, Q_1) where Q_0, Q_1 are mixed-state quantum circuits:

$$\Pi_Y : ||Q_0 - Q_1||_\diamond \geq a$$
$$\Pi_N : ||Q_0 - Q_1||_\diamond \leq b$$

Semidefinite Programming

The concept of *semidefinite programming*, the optimization of a linear objective function over the entries of a positive semidefinite matrix, is central to the proof sketches contained in this survey. We cover this material with brevity; an interested reader may refer to Vandenberghe and Boyd's exposition of the topic [VB96] or to Jain et al. [JJUW10]. The syntax presented here is similar to Arora and Kale [AK07].

For a vector space $\mathbf{V}$, let the set of *linear mappings* of that operator be denoted by $\text{L}(\mathbf{V})$.

An operator $P \in \text{L}(\mathbf{V})$ is said to be *positive semidefinite* if the following conditions are true:

- P is Hermitian
- P has only non-negative eigenvalues.

Let the space of *positive semidefinite operators* be denoted by $\text{Pos}(\mathbf{V})$.

Taking Σ to be a finite, nonempty set of possible measurement outcomes, define the quantum state as a density operator ρ, where A is a register. For the positive semidefinite operator collection:

$$\{P_a | a \in \Sigma\} \subseteq \text{Pos}(A)$$

We define a measurement to be:

$$\sum_{a \in \Sigma} P_a = \mathbb{I}_A$$

Define the relation $A \succcurlyeq B$ to mean $A - B$ is positive semidefinite.

Let $\mathbf{V}$ be the complex vector spaces $\mathbb{C}^\Sigma$. Define m to be the number of constraints. Let A be some $n \times n$ matrix. An example semidefinite program can be denoted as follows:

$$\begin{aligned} \text{For: } & X \in \text{Pos}(\mathbf{V}) \\ \text{Maximize: } & \langle C | X \rangle \\ \text{Subject to: } & \forall d \in [m] : \langle A_d | X \rangle \leq B_d \\ \text{Where: } & X \succcurlyeq 0 \end{aligned} \tag{4}$$

The *dual* semidefinite program (to the semidefinite program above) is denoted as follows:

$$\begin{aligned} \text{Minimize: } & B \cdot Y \\ \text{Subject to: } & \sum_{d=1}^{m} A_d Y_d \succcurlyeq C \\ \text{Where: } & Y \geq 0 \end{aligned} \tag{5}$$

Notice that the matrix B consists of all constraints, and the matrix Y consists of all dual variables.

Matrix Multiplicative Weights Update Method

The *matrix multiplicative weights update method* (or MMW) is a meta-algorithm which tracks costs and weights with applications to solving semidefinite programs; both Jain et al. and Wu use this meta-algorithm in their respective proofs of QIP = PSPACE [JJUW10, Wu10]. A full exposition of the general multiplicative weights update method was published by Arora et al. in 2012. In this survey, we do not discuss the analysis which determines the acceptance probabilities for this algorithm. For such a discussion, we refer the reader to Arora et al. in 2012 [AHK12].

The goal of using the MMW update method is to sample possible solutions to an SDP – either from an oracle or from a distribution – and evaluate the helpfulness of the weight of each possible solution. The algorithm is iterative, and each iteration refines the weight of each solution. However, for certain classes of SDPs, the algorithm is parallelizable in NC(*poly*). This parallelizable class of SDPs which exhibit weak duality, or can be expressed as feasibility problems, is of particular use to us.

For an operator $A \in \mathrm{L}(\mathbf{V})$, where $\mathbf{V} = \mathbb{C}^{\Sigma}$ (as above), the *expontential* of A is defined to be

$$\exp(A) = 1 + A + \frac{A^2}{2} + \frac{A^3}{6} + \ldots$$

Let M be a cost matrix in $\mathbb{R}^{n \times n}$. For each decision $v \in [-1,1]$ made by our algorithm's evaluation, the cost is $v^{\dagger} M v$. This cost is used in analyzing the efficacy of the algorithm.

In a traditional multiplicative weight update approach, our algorithm would be advised by some form of prover. In this case, however, we look for a solution among distributions. A relatively general sketch of the algorithm follows.

```
Pre-processing:
1. Fix some $\eta \leq 1/2$.
2. Initialize a weight matrix $W = \mathbb{I}_X$. This matrix is updated with each iteration of the algorithm.
3. $N = \dim(\mathbf{V})$.
Processing:
For $t = 1,2,...,T$:
1. Take the density matrix $\rho_t = \frac{W_t}{\psi_t}$, for $\psi_t = \mathrm{Tr}(W_t)$.
2. Evaluate $M_t$.
3. Update the matrix $W$ to reflect the observed weights, $W_{t+1} = \exp(-\eta \sum_{k=1}^{t} M_k)$.
After $T$ rounds, return $\frac{1}{T}\sum_{t=1}^{T} \rho_t$.
```

In order to use this method to solve a semidefinite program, we consider how it would fare on the semidefinite program (and its dual) described above. Note that while this efficiently parallelizable approach is able to provide solutions to semidefinite programs of duality, it is not necessarily able to do so for semidefinite programs which do not take this form.

Assume the existence, for some ϵ, of a density matrix for which all values $\langle A_j | X' \rangle \geq \epsilon$. Define $\rho_t = \max_t ||A_t||$. For each iteration of the algorithm, set the density matrix X to be ρ_t. With the value of X set to be this, we then can evaluate whether the conditions of the semidefinite program are satisfied. If so, we have found an accepting condition. Otherwise, we continue with our search. If there is a constraint d for which

$\langle A_d|X\rangle \leq 0$, then set the cost matrix to be $M_t = \frac{1}{\rho_d}A_d$.

Theorem 2.1 *For the MMW algorithm sketched above, the algorithm after T rounds, the error margin for any decision v is:*

$$\sum_{t=0}^{T}\langle M_t|\rho_t\rangle \leq \sum_{t=0}^{T} v^\dagger M_t v + \eta\sum_{t=0}^{T}\langle M_t^2|\rho_t\rangle + \frac{\ln X}{\eta}$$

For a proof of this theorem, see Arora et al.'s survey of the multiplicative weight update method [AHK12].

Equilibrium Value Method

The *equilibrium value method* is essentially a result of the Minimax theorem which was first presented in 1928 by J. v. Neumann, and of the Nash equilibrium which was first presented by John Nash in 1958 [Nas50]. The equilibrium value method is a systematic approach to converting semidefinite programs with feasibility problem representations to zero-sum games. The concept of the Nash equilibrium is key to interpreting strategies in zero-sum or non-cooperative games; it thus appears to lend itself well to the class of quantum refereed games, though a proof of its general applicability is yet unknown. The definition here is similar to Wu's argument. [Wu10].

For some vector spaces $X_0 \otimes X_1$, $X_2 \otimes X_3$, define $L(X_0 \otimes X_1) \to L(X_2 \otimes X_3)$. Let Ξ be the linear operator mapping the function L. Define, in addition, $\Gamma = \{\Pi | 0 \leq \Pi \leq 1_{X_2 \otimes X_3}\}$.

The *equilibrium value* is:

$$\hat{\lambda}(\Xi) = \min_{\rho \in D(X_0 \otimes X_1)} \max_{\Pi \in \Gamma}\langle \Pi|\Xi(\rho)\rangle = \max_{\Pi \in \Gamma} \min_{\rho \in D(X_0 \otimes X_1)}\langle \Pi|\Xi(\rho)\rangle$$

An *equilibrium point* for this value is:

$$\langle \hat{\Pi}|\Xi(\hat{\rho})\rangle = \min_{\rho \in D(X_0 \otimes X_1)}\langle \hat{\Pi}|\Xi(\rho)\rangle = \max_{\Pi \in \Gamma}\langle \Pi|\Xi(\hat{\rho})\rangle$$

3 Quantum interactive proofs

We discuss two proofs of the theorem QIP = PSPACE. The first proof uses the matrix multiplicative weight update method; the second uses MMW as well as the equilibrium value method. This presentation demonstrates that the latter method is significantly simpler, and, as such, this presentation opens the question of where else the equilibrium value method may aid in proof simplification.

3.1 Semidefinite Formulation of QIP = PSPACE

The first step in Jain et al.'s proof QIP = PSPACE is to formulate the class QMAM, or QIP(3), in terms of a semidefinite program and its dual [JJUW10]. The argument then demonstrates that applying the MMW update meta-algorithm to this solve this semidefinite program results in an NC(*poly*) = PSPACE algorithm. Given reference to Jain et al.'s earlier result that QIP(2) ⊆ PSPACE [JUW09] (a result that also uses a semidefinite program formulation), this proves QIP = PSPACE. Note that the QIP(3) protocol acts as a QIP-complete language; this fact is of interest as it illustrates a similarity between Jain et al. and Wu's formulations of QIP = PSPACE.

3.1.1 Semidefinite Program for a QIP(3) Protocol

The derivation of the following semidefinite program and its dual is contained in Appendix B; ψ, X, W, and Y are all defined within that appendix.
Primal semidefinite program:

$$\begin{aligned} \text{Given: } & X \in \text{Pos}(X \otimes W \otimes Y), \\ & \sigma \in D(W) \\ \text{Maximize: } & \text{Tr}(X) \\ \text{Subject to: } & \psi(X) \leq \mathbb{I}_X \otimes \sigma \end{aligned} \tag{6}$$

And its dual:

$$\begin{aligned} \text{Given: } & Y \in \text{Pos}(X \otimes W), \\ \text{Minimize: } & ||\text{Tr}_X(Y)|| \\ \text{Subject to: } & \psi^\dagger(Y) \geq \mathbb{I}_{X \otimes W \otimes Y} \end{aligned} \tag{7}$$

3.1.2 Matrix Multiplicative Weight Update Method to Solve QIP(3) Protocol

The application of the matrix multiplicative weight update method to the dual semidefinite programs defined for the $QIP(3)$ protocol above is exceedingly intricate. This intricateness is one of the motivations for this paper; using the equilibrium value method as we discuss in section 3.2, we can drastically simplify this proof. Using the semidefinite program and its dual defined above, we present an algorithm which accepts when an optimal solution which is greater than $\frac{7}{8}$ exists, and which rejects when an optimal solution is less than $\frac{5}{8}$.

```
Pre-processing:
1. $N = \dim(X \otimes W \otimes Y)$,
2. $M = \dim(W)$
3. $\Pi \in \text{Pos}(X \otimes W)$
4. Define a weight matrix: $W = \mathbb{I}_{X \otimes W \otimes Y}$
5. $\rho = W/N$; $\rho_t = W_t/N$; $\rho \in D(X \otimes W \otimes Y)$
6. Define another weight matrix: $Z = \mathbb{I}_W$
7. $\zeta = Z/M$; $\zeta_t = Z_t/M$; $\zeta \in D(W)$
Processing:
For $t = 1,2,...,T$:
1. Let $\epsilon = 1/64$, $\delta = \frac{\epsilon}{2||Q^{-1}||}$, $T = \lceil \frac{4\log(N)}{\epsilon^2 \delta} \rceil$.
2. Set $\beta_t = \langle \Pi_t | \psi(\rho_t) \rangle$, where $\Pi_t$ represents the projection onto the positive eigenspaces of $\psi(\rho_t) - \frac{4}{3}\mathbb{I}_{X \otimes \zeta_t}$.
3. If $\beta_t \leq \epsilon$: accept.
4. Else:
        $$W_{t+1} = \exp\left( -\epsilon\delta \sum_{j=1}^{t} \psi^\dagger(\frac{\Pi_j}{\beta_j}) \right)$$
        $$Z_{t+1} = \exp\left( \epsilon\delta \sum_{j=1}^{t} \text{Tr}(\frac{\Pi_j}{\beta_j}) \right)$$
   Update $\rho$ and $\zeta$ accordingly.
5. If, after $T$ rounds, we have not yet accepted, reject.
```

Grisly analysis of the acceptance and rejecting probabilities of this application of the matrix multiplicative weight update method to the semidefinite formulations of a QIP(3) protocol indeed confirms that the algorithm accepts when an optimal solution which is greater than $\frac{7}{8}$ exists, and rejects when an optimal solution is less than $\frac{5}{8}$. This analysis is achieved by using Theorem 2.1. For a full treatment of this analysis, refer to Jain et al. [JJUW10].

The intuition for this algorithm is as follows. We sample density matrices from a some distribution, and use these matrices as starting points from which to approximate solutions to the constraints of the semidefinite program. With each iteration, we project these density matrices onto the positive eigenspaces of the constraint space. Using these projections and density matrices, we update our cost and weight matrices. With each iteration, we attempt to narrow the gap between the value of the projection and the chosen value for ϵ. If a solution's projection approximately satisfies the system constraints, then the algorithm accepts the density matrices as a solution to the system and accepts.

3.2 Equilibrium Value Method Formulation of QIP = PSPACE

This section discusses Wu's [Wu10] proof of QIP = PSPACE; the goal of this section is to demonstrate the simplification provided by this presentation of the result. In order to apply the equilibrium value method to demonstrate QIP = PSPACE, we first characterize the class by considering the QIP-complete problem QCD, or quantum circuit differentiability. It must be said, though, that we could equally apply the equilibrium value method to the QIP(3) protocol above; we choose to use the class QCD for additional simplicity. Using this categorization, we apply the equilibrium value method to again reformulate the problem. The equilibrium values can then be approximated by applying the matrix multiplicative weight update algorithm.

3.2.1 Equilibrium Value formulation of QCD$_{a,b}$

Recall our definition of the equilibrium value method.

For some vector spaces $X_0 \otimes X_1$, $Z \otimes Q$, where X_0, X_1 are isomorphic copies of the space X, define $L(X_0 \otimes X_1) \to L(Z \otimes Q)$. Let Ξ be the linear operator mapping the function L.

The equilibrium value $\hat{\lambda}(\Xi)$ for QCD$_{a,b}$ is:

$$\begin{aligned} \hat{\lambda}(\Xi) &\leq \frac{\sqrt{(4-a^2)}}{2}, ||Q_0 - Q_1||_\diamond \geq a \\ \hat{\lambda}(\Xi) &\geq \frac{2-b}{2}, ||Q_0 - Q_1||_\diamond \leq b \end{aligned} \tag{8}$$

The constraints, $\frac{\sqrt{(4-a^2)}}{2}$, $\frac{2-b}{2}$, on the equilibrium value are determined in part by analyzing the fidelity of the quantum states. The algebraic manipulation which achieves this proof is discussed at length in Wu's original paper [Wu10]. We note that the operator Ξ is constructed efficiently in parallel (in NC(*poly*)), a requirement for the application of this proof technique to PSPACE problems.

3.2.2 Converting the Equilibrium Value Formulation to a Semidefinite Program

Formulating a semidefinite program and its dual from this equilibrium value representation of the quantum circuit distinguishability problem is immediate. Because the equilibrium point is determined through the application of the Minimax theorem, the minimization and maximization criteria are explicitly spelled out. This is a significant difference from the earlier demonstration of formulating a QIP(3) protocol as a semidefinite program.
For coverage of how to convert an instance of the Minimax theorem to a semidefinite program to which the matrix multiplicative weight update method is relevant, see Kale's thesis [Kal07]. Here, we simply observe that any problem which can be formulated as a Minimax problem also has strong duality between semidefinite programs, and thus the matrix multiplicative weight update algorithm can be used to parallelize the optimization.

3.2.3 Matrix Multiplicative Weight Solution to the Semidefinite Program

Given the Minimax program defined above, we can apply the matrix multiplicative weight algorithm in the usual way so as to determine the equilibrium value, and, by doing so, we can then resolve the $QCD_{a,b}$ protocol in NC(*poly*) = PSPACE. The following algorithm updates the cost matrix M in a non-standard way.

```
Pre-processing:
1. Fix some $\epsilon = \frac{\delta}{4}$. Fix $T = \lceil \frac{16 \log N}{\delta^2} \rceil$.
2. Initialize a weight matrix $W = \mathbb{I}_X$.
3. $N = \dim(X)$.
Processing:
For $t = 1, 2, ..., T$:
1. Take the density matrix $\rho_t = \frac{W_t}{\psi_t}$, for $\psi_t = \mathrm{Tr}(W_t)$.
2. Compute $\Xi(\rho_t)$. Let $\Pi_t$ be the projection onto the positive eigenspaces of $\Xi(\rho_t)$.
3. Evaluate
       $M_t = \frac{\Xi^\dagger(\Pi_t) + \mathbb{I}_X}{2}$
4. Update the matrix $W$ to reflect the observed weights, $W_{t+1} = \exp(-\epsilon \sum_{k=1}^{t} M_k)$.
After $T$ rounds, return $\hat{\lambda}(\Xi) = \frac{1}{T} \sum_{t=1}^{T} \langle \Pi_t | \Xi(\rho_t) \rangle$.
```

Applying Theorem 2.1 to analyze this algorithm indeed demonstrates that this protocol solves the semidefinite program defining the QIP-complete problem of quantum circuit distinguishability. For a detailed analysis of the algorithm, including details of the promise separation, see Wu's original paper.
As Ξ can be generated in NC(*poly*), and as the matrix multiplicative weight update algorithm described above operates in NC(*poly*), given that this protocol is for a QIP-complete problem, this demonstrates QIP = PSPACE.

4 Quantum Refereed Games

So far, this survey has discussed two different albeit similar approaches to proving QIP = PSPACE. The goal of this discussion has been to demonstrate the simplifications made possible by applying the equilibrium value method. In comparing these approaches, a natural next step is to see where else the equilibrium value method can be used as a proof technique. In this section, we discuss quantum refereed games, a complexity class which is an abstraction above QIP. As QIP = PSPACE, similarly QRG = EXP, whereby adding quantum computational power does not increase the overall power of the complexity class. The technique used to prove QRG = EXP is very similar to Jain et al.'s proof of QIP = PSPACE. Again, it requires the manipulation of semidefinite programs along with the application of the matrix multiplicative weight update algorithm to resolve these programs. The question of whether the equilibrium value method can be used to simplify this proof remains open; this section lays the groundwork for approaching that problem, first by discussing the known proof of QRG = EXP, and then by discussing how the equilibrium value method has been effectively used to show QRG(2) = PSPACE. This latter proof is particularly interesting as it requires a recursive implementation of the matrix multiplicative weight update algorithm

4.1 Semidefinite Formulation of QRG = RG = EXP

In this section, we follow Gutoski and Watrous's [GW07] demonstration that the class QRG can be expressed in terms of a semidefinite program which is polynomial in the size of the input string, and, as the particular semidefinite program generated is similar to a convex programming semidefinite program, it can be solved in polynomial time. This then demonstrates QRG $\subseteq$ EXP. EXP $\subseteq$ QRG follows immediately from RG $\subseteq$ QRG, combined with the result that RG = EXP.

We assume the definition of a strategy, quantum or otherwise. For more information about strategy choices, see Gutoski and Watrous's original paper [GW07]. The syntax employed here is to say that for $S_n(X_{1...n}, W_{1...n})$, the strategy set S_n takes $X_{1...n}$ as input and gives $W_{1...n}$ as output.

The following theorem is stated and proven in Gutoski and Watrous's paper [GW07]. We refer to this result in constructing our semidefinite program, and so we repeat it here:

Theorem 4.1 *Let $n \geq 1$, let $X_1,...,X_n$, $Y_1,...,Y_n$ be complex Euclidean spaces, and let $\{Q_a | a \in \Sigma\}$ represent an n-turn measuring strategy with input spaces $X_1,...,X_n$ and output spaces $Y_1,...,Y_n$. Then for each $a \in \Sigma$, the maximum probability with which this strategy can be forced to output a, maximized over all choices of compatible co-strategies is given by:*

$$\min\{p \in [0,1] | Q_a \leq pR \text{ for some } R \in S_n(X_{1...n}, Y_{1...n})$$

With the same result applying to measuring co-strategies.

Now we return to our construction of a semidefinite program for QRG.

- Let m be the number of turns the "yes" prover and the "no" prover are allotted. Let the verifier $V = \{V_y, V_n\}$.
- Let the yes-prover (defined in the definition of QRG) have a strategy $Y \in S_n(Y_{1...m}, C_{1...m})$ and let the no-prover have a strategy $N \in S_n(N_{1...m}, D_{1...m})$.
- From the strategies defined here, we construct linear super operators Ω_Y and Ω_N:

$$\Omega_{V_y}(Y) = \mathrm{Tr}_{Y_{1...m} \otimes C_{1...m}}\left((Y \otimes \mathbb{I}_{D_{1...m} \otimes N_{1...m}}) V_y \right)$$
$$\Omega_{V_n}(Y) = \mathrm{Tr}_{N_{1...m} \otimes D_{1...m}}\left((Y \otimes \mathbb{I}_{D_{1...m} \otimes N_{1...m}}) V_n \right)$$

- Assuming the provers conform to their chosen strategies, the verifier outputs 1 (a "yes" response) with probability $\langle Y \otimes N | V_y \rangle = \langle N | \Omega_{V_y}(Y) \rangle$.
- The verifier outputs 0 (a "no" response) with probability $\langle Y \otimes N | V_n \rangle = \langle N | \Omega_{V_n}(Y) \rangle$.

In order to decide an input x, we compute the maximum probability that the no-prover can win the game.

$$\Pr[\text{Prover outputs } 0] = \max \langle N | \Omega_{V_n}(Y) \rangle$$

Which, using Theorem 4.1, is:

$$\min\{p \geq 0 | \Omega_{V_n}(Y) \leq pQ \text{ for some } Q \in \text{co-}S_n(N_{1...m}, D_{1...m})\}$$

Which gives the semidefinite program:

$$\begin{aligned} \text{Minimize: } & p \\ \text{Subject to: } & \Omega_{V_n}(Y) \leq pQ \\ \text{And: } & Y \in S_n(Y_{1...m}, C_{1...m}) \\ \text{And: } & Q \in \text{co-}S_n(N_{1...m}, D_{1...m}) \end{aligned} \tag{9}$$

Note that this semidefinite program can be written in terms of linear and semidefinite constraints. With this observation, the formulation of the semidefinite program is complete, and the result QRG $\subseteq$ EXP is secured.

4.2 Equilibrium Value Method Proof QRG(2) = PSPACE

Restricting QRG to only allow two rounds of interactions between the "yes" and "no" provers and the verifier, we can use the equilibrium value method to show QRG(2) $\subseteq$ PSPACE. This result is of particular interest, as it's the first instance in which the equilibrium value method has resolved an open question before a traditional formulation of a semidefinite program followed by the matrix multiplicative weight update algorithm [GW10, GW12]. This discussion is of particular interest also because it requires the use of a recursive implementation of the matrix multiplicative weight update method. The inclusion of this second fact is evidence of the flexibility of the equilibrium value method as a proof technique.

Due to space constraints, we briefly discuss the application of the matrix multiplicative weight update method to an approximation of the equilibrium values for QRG. We assume the following four conditions can be achieved:

1. There exist equilibrium values bounded in the usual way which we can approximate to solve QRG(2).
2. We have the ability to apply the MMW update method to k-tuples of matrices simultaneously.

3. We have the ability to ignore some k-tuples of matrices.
4. We have the ability to obtain consistent quantum states using a rounding lemma.

From these three assumptions, we can apply the matrix multiplicative weight update method in a recursive form to approximate the equilibrium values for QRG(2). The recursive form is used to select first an optimum quantum strategy for the "yes" prover, and at the next level to select an optimum quantum strategy for the "no" prover. As the equilibrium value constraints can be determined in NC(*poly*), and as the recursive implementation of the matrix multiplicative weight update method can run in NC(*poly*), this demonstrates QRG(2) $\subseteq$ PSPACE.

5 Open: Equilibrium Value Method Proof QRG = RG = EXP

In this paper, we have looked at the use of the equilibrium value method as an alternate characterization of the proof of QIP = IP = PSPACE. We then looked at the matrix multiplicative weight proof of QRG = RG = EXP, and we briefly discussed how the equilibrium value method was used to prove QRG(2) = PSPACE. Between these results, it seems probably that the equilibrium value method could be used again to provide an alternate characterization of the proof that QRG = RG = EXP. The equilibrium value method appears to be flexible and robust as a proof technique, and hence we are hopeful that its application could be used for further proof simplification and for proof resolution for a number of results in the coming years.

6 Acknowledgements

Thanks to Scott Aaronson for a great class on quantum complexity theory!

References

[AHK12] S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory of Computing*, 8(1):121–164, 2012.

[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. ACM, 2007.

[GW07] G. Gutoski and J. Watrous. Toward a general theory of quantum games. In *Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing*, STOC '07, pages 565–574, New York, NY, USA, 2007. ACM.

[GW10] G. Gutoski and X. Wu. Parallel approximation of min-max problems. *arXiv preprint arXiv:1011.2787*, 2010.

[GW12] Gus Gutoski and Xiaodi Wu. 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*, pages 21–31. IEEE, 2012.

[JJUW10] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP = PSPACE. *Commun. ACM*, 53(12):102–109, December 2010.

[JUW09] R. Jain, S. Upadhyay, and J. Watrous. Two-message quantum interactive proofs are in pspace. In *Foundations of Computer Science, 2009. FOCS '09. 50th Annual IEEE Symposium on*, pages 534–543, Oct 2009.

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

[KW00] A. Kitaev and J. Watrous. Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 608–617. ACM, 2000.

[Nas50] J. Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

[RW05] B. Rosgen and J. Watrous. On the hardness of distinguishing mixed-state quantum computations. In *Computational Complexity, 2005. Proceedings. Twentieth Annual IEEE Conference on*, pages 344–354. IEEE, 2005.

[Sha92] A. Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, October 1992.

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

[Wat03] J. Watrous. PSPACE has constant-round quantum interactive proof systems. *Theoretical Computer Science*, 292(3):575 – 588, 2003. Algorithms in Quantum Information Prcoessing.

[Wu10] X. Wu. Equilibrium value method for the proof of QIP = PSPACE. *arXiv preprint arXiv:1004.0264*, 2010.

Appendix A

For an input x, a QIP(3) model operates in the following manner:
The prover (defined earlier) communicates a quantum register W to the verifier. The verifier then generates a random binary bit. The prover then sends an additional quantum register to the verifier. This completes the number of rounds of communication allotted to this protocol. The verifier then applies a binary measurement to the register (W,Y). Defining operators P_0 and P_1 as accepting and $\mathbb{I}-P_0$, $\mathbb{I}-P_1$ as rejecting, the outcome of the measurement is

$$\{P_a,\mathbb{I}-P_b\}\subset \mathrm{Pos}(W\otimes Y)$$

We define an additional vector space

$$X=\mathbb{C}^{\{0,1\}}$$

We use this vector space to reason about the maximum accepting probability of the semidefinite program. Take:

$$Q=\frac{1}{2}|0\rangle\langle 0|\otimes P_0+\frac{1}{2}|1\rangle\langle 1|\otimes P_1$$

Clearly:

$$Q\in \mathrm{Pos}(X\otimes W\otimes Y)$$

Take:

$$X=|0\rangle\langle 0|\otimes \rho_0+|1\rangle\langle 1|\otimes \rho_1$$

The accepting probability is then:

$$\langle Q|X\rangle=\frac{1}{2}\langle P_0|\rho_0\rangle+\frac{1}{2}\langle P_1|\rho_1\rangle$$

Constraining ρ_0 and ρ_1 such that:

$$\mathrm{Tr}_Y(\rho_0)=\mathrm{Tr}_Y(\rho_1)=\sigma$$

With this setup, we can take the semidefinite program to be:

$$\begin{aligned}\text{Given: } & X\in \mathrm{Pos}(X\otimes W\otimes Y),\\ & \sigma\in D(W)\\ \text{Maximize: } & \langle Q|X\rangle\\ \text{Subject to: } & \mathrm{Tr}_Y(X)\leq \mathbb{I}_X\otimes\sigma\end{aligned} \tag{10}$$

The dual of this semidefinite program is then:

$$\begin{aligned}\text{Given: } & Y\in \mathrm{Pos}(X\otimes W),\\ \text{Minimize: } & ||\mathrm{Tr}_X(Y)||\\ \text{Subject to: } & Y\otimes \mathbb{I}_Y\geq Q\end{aligned} \tag{11}$$

Defining the linear mapping:

$$\begin{aligned}&\psi: L(X\otimes W\otimes Y)\to L(X\otimes W)\\ &\psi(X)=\mathrm{Tr}_Y(Q^{-\frac{1}{2}}XQ^{-\frac{1}{2}})\\ &\psi^\dagger(Y)=Q^{-\frac{1}{2}}(Y\otimes \mathbb{I}_Y)Q^{-\frac{1}{2}}\end{aligned} \tag{12}$$

We can then rewrite the semidefinite program while preserving the optimal values as:

$$\begin{aligned}\text{Given: } & X\in \mathrm{Pos}(X\otimes W\otimes Y),\\ & \sigma\in D(W)\\ \text{Maximize: } & \mathrm{Tr}(X)\\ \text{Subject to: } & \psi(X)\leq \mathbb{I}_X\otimes\sigma\end{aligned} \tag{13}$$

And its dual:

$$\begin{aligned}\text{Given: } & Y\in \mathrm{Pos}(X\otimes W),\\ \text{Minimize: } & ||\mathrm{Tr}_X(Y)||\\ \text{Subject to: } & \psi^\dagger(Y)\geq \mathbb{I}_{X\otimes W\otimes Y}\end{aligned} \tag{14}$$

The above revision is helpful in that it allows the application of the matrix multiplicative weight update method to solve the SDP in the same vein as the solution which earlier demonstrated $\mathrm{QIP}(2) \subseteq \mathrm{PSPACE}$ [JUW09].