Upper Bounds for Quantum Interactive Proofs with Competing Provers

Gus Gutoski
Institute for Quantum Information Science and
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada

Abstract

Refereed games are interactive proof systems with two competing provers: one that tries to convince the verifier to accept and another that tries to convince the verifier to reject. In quantum refereed games, the provers and verifier may perform quantum computations and exchange quantum messages. One may consider games with a bounded or unbounded number of rounds of messages between the verifier and provers.

In this paper, we prove classical upper bounds on the power of both one-round and many-round quantum refereed games. In particular, we use semidefinite programming to show that many-round quantum refereed games are contained in NEXP*. It then follows from the symmetric nature of these games that they are also contained in* coNEXP*. We also show that one-round quantum refereed games are contained in* EXP *by supplying a separation oracle for use with the ellipsoid method for convex feasibility.*

1. Introduction

A *refereed game* consists of an interaction between a computationally bounded verifier and two computationally unbounded provers regarding some input string x. The two provers use their unbounded computational power to compete with each other: one prover, called the *yes-prover*, attempts to convince the verifier to accept x, while the other prover, called the *no-prover*, attempts to convince the verifier to reject x. At the end of the interaction, the verifier decides whether to accept or reject the input x, effectively deciding which of the provers wins the game. Such games represent games of incomplete information because the messages exchanged between one prover and the verifier are considered to be hidden from the other prover.

A language L is said to have a refereed game if, for each string $x \in L$, there exists a yes-prover that can always convince the verifier to accept x with probability at least $3/4$, regardless of the no-prover's strategy. Similarly, for each $x \notin L$, there must also exist a no-prover that can always convince the verifier to reject x with probability at least $3/4$, regardless of the yes-prover's strategy. A *round* of messages between the verifier and provers consists of one message from the verifier to each of the provers, followed by a response from each of the provers to the verifier. One may consider refereed games with a bounded or unbounded number of rounds.

The refereed games model is based on the interactive proof system model [10, 3, 4, 5], which has a rich history that we will not survey here. The refereed games model and variations thereof were considered in the classical case in Refs. [21, 8, 7, 18, 9, 6], among others. Much of what is known about the complexity-theoretic aspects of classical refereed games is due to Feige and Kilian [6]. The complexity class of languages having classical refereed games with any polynomial number of rounds coincides with EXP (deterministic time $2^{p(|x|)}$ for some polynomial p). The simulation of EXP by a polynomial-round refereed game is due to Feige and Kilian [6] and is based on the arithmetization technique developed by Lund, Fortnow, Karloff and Nisan [19] and used in proofs of $\mathsf{IP} = \mathsf{PSPACE}$ [22, 23]. The containment of this class in EXP is due to Koller and Megiddo [18] and relies upon convex programming algorithms such as the ellipsoid method [13, 11] to solve an exponential-size linear program with a doubly exponential number of constraints.

On the other hand, the class of languages having one-round refereed games coincides with PSPACE [6]. Little is known about the expressive power of classical refereed games intermediate between these two extremes. For instance, games with a constant number of rounds may correspond to PSPACE, EXP, or some complexity class between the two.

Similar to the classical case, quantum refereed games are based on the quantum interactive proof system model [25, 16]. Quantum refereed games and quantum interactive proof systems differ from their classical counterparts in that the provers and the verifier may perform quantum computations and exchange quantum messages. It is known that the class QIP of languages having quantum interactive proof systems satisfies PSPACE $\subseteq$ QIP $\subseteq$ EXP. The lower bound follows trivially from the fact that IP $=$ PSPACE and the upper bound is achieved via a reduction to an exponential-size semidefinite program [16, 14, 15] and relies upon efficient algorithms for semidefinite programming [13, 11, 20, 2, 24].

In this paper, we formalize the reduction outlined in Ref. [14] and use it to show that any language with a many-round quantum refereed game is contained in NEXP (the nondeterministic analogue of EXP). That such a language is also in coNEXP follows immediately from the symmetric nature of quantum refereed games. Hence, quantum refereed games are strictly contained in NEXP unless NEXP $=$ coNEXP.

Quantum refereed games were also studied in Ref. [12], in which it was shown that every language in QIP has a one-round quantum refereed game, provided that the verifier is permitted to process the yes-prover's response before sending a message to the no-prover. That paper also raises the question of how one-round quantum refereed games of this special form relate to EXP.

In the present paper, we answer that question by proving the containment of these short quantum games in EXP. This containment is obtained via an interesting combination of the semidefinite programming approach for quantum interactive proof systems and the convex programming approach for classical refereed games.

These two upper bounds serve to tighten the possible ranges of power for one- and many-round quantum refereed games. In consideration with known lower bounds, one-round quantum refereed games lay between QIP and EXP and many-round quantum refereed games lay between EXP and NEXP $\cap$ coNEXP. Our results also shed some light on the effect of quantum information on the complexity of finding strategies for two-player games: PSPACE versus an upper bound of EXP in the one-round case and EXP versus an upper bound of NEXP $\cap$ coNEXP in the many-round case.

The remainder of this paper is organized as follows. We begin by defining quantum refereed games in Section 2. In Section 3 we prove the containment of many-round quantum refereed games in NEXP $\cap$ coNEXP. The developments in that section facilitate the work of Section 4, in which we prove that one-round quantum refereed games are no more powerful than EXP. We conclude with Section 5, which includes a diagram of relationships between complexity classes discussed in this paper.

2. Preliminaries

2.1. Quantum Information and Quantum Circuits

In this paper, a *Hilbert space* is the vector space $\mathbb{C}^N$ for some positive integer N endowed with the inner product $\langle v, w\rangle = v^* w$ for any column vectors $v, w \in \mathbb{C}^N$. Let $\mathcal{H}$ be any Hilbert space with dimension N, let $\mathbf{L}(\mathcal{H})$ denote the vector space of all $N \times N$ matrices, and let $\langle A, B\rangle = \operatorname{tr}(A^* B)$ be an inner product on $\mathbf{L}(\mathcal{H})$ for any $A, B \in \mathbf{L}(\mathcal{H})$. For any vector $v \in \mathcal{H}$, $\|v\| = \sqrt{\langle v, v\rangle}$ denotes the Euclidean norm of v. For any matrix $A \in \mathbf{L}(\mathcal{H})$, the spectral norm of A, denoted $\|A\|$, is given by

$$\|A\| = \sup_{v \in \mathcal{H}\setminus\{0\}} \frac{\|Av\|}{\|v\|}.$$

Let $I_{\mathcal{H}} \in \mathbf{L}(\mathcal{H})$ denote the $N \times N$ identity matrix and let $|0_{\mathcal{H}}\rangle \in \mathcal{H}$ denote the standard basis vector with a 1 in the first entry and all other entries equal to zero. We may write I and $|0\rangle$ when the Hilbert space $\mathcal{H}$ is clear from the context. Note that use of the Dirac notation in this paper is limited to the vector $|0\rangle$, where $|0\rangle^*$ is denoted by $\langle 0|$. We also let $\mathbf{Pos}(\mathcal{H}) \subset \mathbf{L}(\mathcal{H})$ denote the set of all positive semidefinite matrices in $\mathbf{L}(\mathcal{H})$.

A *qubit* is a fundamental unit of quantum information described as follows. Any collection of n qubits has a corresponding Hilbert space $\mathcal{F}$ of dimension 2^n. Any *state* of those qubits is completely described by some $X \in \mathbf{Pos}(\mathcal{F})$ satisfying $\operatorname{tr}(X) = 1$. Conversely, any such X describes some physically realizable state of those n qubits, so it makes sense to refer to the matrix X as a "state". Furthermore, if $X = vv^*$ for some vector $v \in \mathcal{F}$ then X is called a *pure state*. As X has unit trace, it must be the case that $\|v\| = 1$. Because X is completely described by v, it makes sense to refer to any unit vector v as a "pure state" of those n qubits.

The model for quantum computation that provides a basis for quantum refereed games is the quantum circuit model. All quantum circuits in this paper are assumed to be composed of a finite number of gates, each of which is chosen from some finite universal set of quantum gates. Thus, for any quantum circuit Q acting on n qubits with corresponding Hilbert space $\mathcal{F}$, there is a unitary matrix $U \in \mathbf{U}(\mathcal{F})$ associated with Q, where $\mathbf{U}(\mathcal{F}) \subset \mathbf{L}(\mathcal{F})$ denotes the set of all unitary matrices in $\mathbf{L}(\mathcal{F})$. Furthermore, for any unitary matrix $V \in \mathbf{U}(\mathcal{F})$ and any $\varepsilon > 0$, there is a quantum circuit Q with associated unitary matrix $U \in \mathbf{U}(\mathcal{F})$ such that $\|U - V\| < \varepsilon$.

This associated unitary matrix U models the action of Q upon its input qubits in the state $X \in \mathbf{Pos}(\mathcal{F})$ so that the state of those n qubits after Q is applied is $UXU^* \in \mathbf{Pos}(\mathcal{F})$. If $X = uu^*$ for some pure state $u \in \mathcal{F}$ then the resulting state is the pure state $Uu \in \mathcal{F}$. It is important to note that we lose no generality by allowing only unitary quantum circuits because arbitrary admissible quantum operations, including measurements, can be simulated by unitary circuits as described in Ref. [1].

2.2. Quantum Refereed Games

Throughout the paper we assume all strings are over the binary alphabet $\Sigma = \{0, 1\}$. For any string $x \in \Sigma^*$, $|x|$ denotes the length of x.

An *r-round prover P* is a mapping on input strings $x \in \Sigma^*$ where

$$P(x) = (P_1, \ldots, P_r)$$

is an r-tuple of quantum circuits. Each of the circuits in this r-tuple acts upon the same number of qubits and these qubits are partitioned into two sets: one set of qubits is private to that prover and the other is shared with the verifier. These shared qubits act as a quantum channel between the verifier and that prover. No restrictions are placed on the complexity of the prover's circuits, which captures the notion that the prover has unlimited computational power—each of the prover's circuits can be viewed as an arbitrary unitary operation on its input qubits.

We require that any prover must be either a *yes-prover* or a *no-prover*. This distinction is purely a notational convenience and is defined as follows: the Hilbert spaces corresponding to the private and shared qubits of a yes-prover are denoted $\mathcal{Y}$ and $\mathcal{M}_Y$ respectively. Similarly, the Hilbert spaces corresponding to the private and shared qubits of a no-prover are denoted $\mathcal{N}$ and $\mathcal{M}_N$ respectively.

An *r-round verifier V* is a mapping on input strings $x \in \Sigma^*$ where

$$V(x) = (V_0, \ldots, V_r)$$

is an $(r+1)$-tuple of quantum circuits. Each of the circuits in this $(r+1)$-tuple acts upon the same number of qubits and these qubits are partitioned into three sets: one set, with corresponding Hilbert space $\mathcal{V}$, is private to the verifier and the two remaining sets have corresponding Hilbert spaces $\mathcal{M}_Y$ and $\mathcal{M}_N$ and are shared with the yes- and no-provers respectively. We require that the verifier's $(r+1)$-tuple of circuits $V(x)$ be generated by a polynomial-time Turing machine on input x. This uniformity constraint captures the notion that the verifier's computational power is limited. We implicitly identify provers and verifiers with the unitary matrices associated with their quantum circuits.

A *quantum refereed game* has a verifier V, a yes-prover Y, and a no-prover N. For any input string $x \in \Sigma^*$ we create a composite circuit $(V, Y, N)(x)$ by concatenating the circuits

$$V_0, N_1, Y_1, V_1, \ldots, V_{r-1}, N_r, Y_r, V_r$$

in sequence, each circuit acting only on the sets of qubits stipulated in the above definitions. Such a circuit is illustrated in Figure 1 for the case $r = 3$. The Hilbert space corresponding to the qubits upon which $(V, Y, N)(x)$ acts is denoted

$$\mathcal{S} = \mathcal{Y} \otimes \mathcal{M}_Y \otimes \mathcal{V} \otimes \mathcal{M}_N \otimes \mathcal{N}.$$

For later convenience, we also define

$$\mathcal{S}_{\mathcal{N}} = \mathcal{Y} \otimes \mathcal{M}_Y \otimes \mathcal{V} \otimes \mathcal{M}_N,$$
$$\mathcal{S}_{\mathcal{Y},\mathcal{N}} = \mathcal{M}_Y \otimes \mathcal{V} \otimes \mathcal{M}_N.$$

The quantum refereed game is implemented by applying the circuit $(V, Y, N)(x)$ to the initial pure state $|0_{\mathcal{S}}\rangle \in \mathcal{S}$. One of the verifier's private qubits is designated as the *output qubit*. After $(V, Y, N)(x)$ has been applied, acceptance is dictated by a measurement of the output qubit in the computational basis.

We now define the complexity class QRG based on quantum refereed games. The class QRG consists of all languages $L \subseteq \Sigma^*$ for which there exists an r-round verifier V such that

1. There exists an r-round yes-prover Y such that, for all r-round no-provers N and all $x \in L$, $(V, Y, N)(x)$ rejects x with probability at most $1/4$.
2. There exists an r-round no-prover N such that, for all r-round yes-provers Y and all $x \notin L$, $(V, Y, N)(x)$ accepts x with probability at most $1/4$.

The quantity $1/4$ in this definition is called the *error*.

The class QRG is robust with respect to error in the sense that our choice of the quantity $1/4$ in the previous definition is arbitrary and may be replaced with any ε in the interval $(0, 1/2)$ without affecting the power of the quantum refereed game model of computation.

To see this fact, it suffices to note that the error of any quantum refereed game can be made as small as desired by repeating the game many times in succession and accepting based on a majority vote of all the repetitions. By Chernoff bounds, the error of the new repeated game decreases exponentially in the number of repetitions of the old game.

Later in this paper we will restrict our attention to a specific class of quantum refereed games that we call *short quantum games*. A short quantum game has a two-round verifier V, a one-round yes-prover Y, and a one-round no-prover N. In these games, the composite circuit $(V, Y, N)'(x)$ is created by concatenating the circuits V_0, Y_1, V_1, N_1, V_2 in sequence. In other words, short quantum games are one-round quantum refereed games in which the verifier is permitted to process the yes-prover's response before sending a message to the no-prover. We let SQG denote the complexity class of languages that have short quantum games. This class is known to be at least partially robust with respect to error [12].

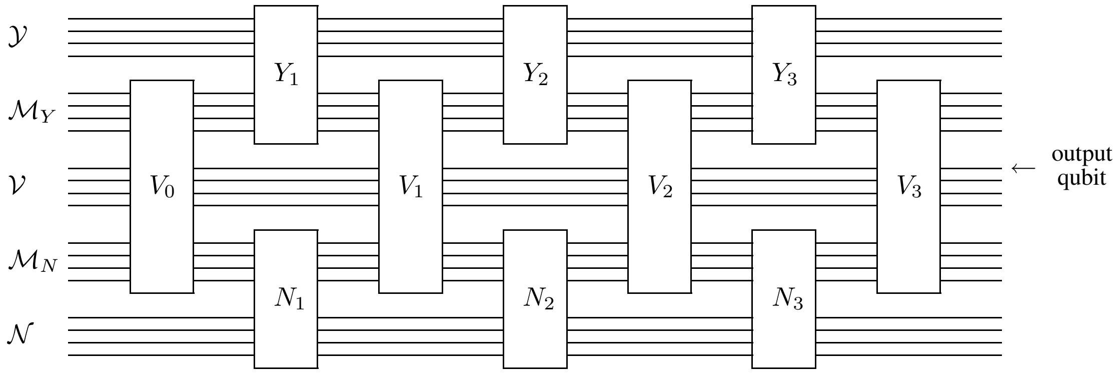

Figure 1. Example of a three-round quantum refereed game

It is clear that the class QRG is closed under complementation, so that coQRG $=$ QRG. However, the protocol for short quantum games is asymmetric in that the yes-prover exchanges messages with the verifier before the no-prover. The class coSQG of languages whose complement is in SQG can be defined in a manner similar to that of SQG. The only difference is that the composite circuit $(V, Y, N)''(x)$ is created by concatenating the circuits V_0, N_1, V_1, Y_1, V_2 in sequence, so that the no-prover exchanges messages with the verifier before the yes-prover. It is not known whether SQG and coSQG coincide.

3. A Classical Upper Bound for QRG

In this section we prove that QRG $\subseteq$ NEXP. As mentioned in Section 1, this upper bound is achieved via a reduction to semidefinite programming. Kitaev and Watrous used a specialized reduction to simulate three-message quantum interactive proof systems in deterministic exponential time [16]. That QIP $\subseteq$ EXP follows from the fact that the verifier and prover in such a proof system need only ever exchange at most three messages [16]. Kitaev later gave a generalized reduction that directly simulates many-message quantum interactive proof systems [14].

We use this generalized reduction in a straightforward manner: nondeterministically guess the unitary matrices for one of the provers and then simulate the induced quantum interactive proof system. The formalization of that idea in this section serves as a convenient warm-up for the work in Section 4.

3.1. The $\mathrm{opt}()$ Function

Often in this paper we multiply matrices acting on a certain Hilbert space with matrices or vectors from a larger Hilbert space. In these cases we implicitly assume that the smaller matrix is extended to the larger Hilbert space by tensoring with the identity. For example, if $\mathcal{F}$ and $\mathcal{G}$ are Hilbert spaces, $A \in \mathbf{L}(\mathcal{F})$, $B \in \mathbf{L}(\mathcal{F} \otimes \mathcal{G})$, and $v \in \mathcal{F} \otimes \mathcal{G}$ then AB and Av always mean $(A \otimes I_{\mathcal{G}})B$ and $(A \otimes I_{\mathcal{G}})v$ respectively.

Let V be an r-round verifier and let Y and N be r-round yes- and no-provers. Fix any string $x \in \Sigma^*$ and let $V_0, \ldots, V_r \in \mathbf{U}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ denote the unitary matrices associated with $V(x)$. Similarly, let $Y_1, \ldots, Y_r \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$ and $N_1, \ldots, N_r \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$ denote the unitary matrices associated with $Y(x)$ and $N(x)$. In Section 2.2 we stipulated that the circuit $(V, Y, N)(x)$ is applied to input qubits in the pure state $|0_{\mathcal{S}}\rangle \in \mathcal{S}$. Thus, the pure state of the system after $(V, Y, N)(x)$ is applied is precisely

$$V_r Y_r N_r V_{r-1} \cdots V_1 Y_1 N_1 V_0 |0_{\mathcal{S}}\rangle.$$

We also stipulated in Section 2.2 that acceptance is dictated by a measurement of the output qubit. In particular, $(V, Y, N)(x)$ rejects x with probability

$$\left\| \Pi_{\mathrm{rej}} V_r Y_r N_r V_{r-1} \cdots V_1 Y_1 N_1 V_0 |0_{\mathcal{S}}\rangle \right\|^2$$

where $\Pi_{\mathrm{rej}} \in \mathbf{L}(\mathcal{V})$ denotes the projection onto the states for which the verifier's output qubit is in the pure state $|0\rangle$.

If the verifier V happens to witness the fact that some language L is in QRG then it follows from the definition of QRG that

1. If $x \in L$ then there exist unitary matrices $Y_1, \ldots, Y_r \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$ such that

   $$\left\| \Pi_{\mathrm{rej}} V_r Y_r N_r V_{r-1} \cdots V_1 Y_1 N_1 V_0 |0\rangle \right\|^2 \leq 1/4$$

   for all unitary matrices $N_1, \ldots, N_r \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$.

2. If $x \notin L$ then there exist unitary matrices $N_1, \ldots, N_r \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$ such that

   $$\left\| \Pi_{\mathrm{rej}} V_r Y_r N_r V_{r-1} \cdots V_1 Y_1 N_1 V_0 |0\rangle \right\|^2 \geq 3/4$$

   for all unitary matrices $Y_1, \ldots, Y_r \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$.

In light of this fact, we define the $\mathrm{opt}()$ function as follows. For any Hilbert spaces $\mathcal{F}$, $\mathcal{G}$, and $\mathcal{H}$ and any matrices $A_0, \ldots, A_r \in \mathbf{L}(\mathcal{F} \otimes \mathcal{G})$, we let $\mathrm{opt}(A_r, \ldots, A_0)$ denote

$$\max \left\{ \left\| A_r U_r A_{r-1} \cdots A_1 U_1 A_0 |0_{\mathcal{F}\otimes\mathcal{G}\otimes\mathcal{H}}\rangle \right\|^2 \right\} \qquad (1)$$

where the maximum is over all $U_1, \ldots, U_r \in \mathbf{U}(\mathcal{G} \otimes \mathcal{H})$. Using this notation, we have

$$\begin{aligned} x \in L \quad &\Leftrightarrow \quad \exists\, Y_1, \ldots, Y_r \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y) \\ &\qquad : \mathrm{opt}\left(\Pi_{\mathrm{rej}} V_r Y_r, \ldots, V_1 Y_1, V_0\right) \leq 1/4 \\ x \notin L \quad &\Leftrightarrow \quad \forall\, Y_1, \ldots, Y_r \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y) \\ &\qquad : \mathrm{opt}\left(\Pi_{\mathrm{rej}} V_r Y_r, \ldots, V_1 Y_1, V_0\right) \geq 3/4 \end{aligned}$$

where we take $\mathcal{F} = \mathcal{Y} \otimes \mathcal{M}_Y \otimes \mathcal{V}$, $\mathcal{G} = \mathcal{M}_N$, and $\mathcal{H} = \mathcal{N}$ in the definition of $\mathrm{opt}()$.

These implications suggest an obvious nondeterministic algorithm for QRG, provided that $\mathrm{opt}()$ can be computed in polynomial-time. Indeed, computing this function is precisely the task achieved by Kitaev's reduction to semidefinite programming [14].

3.2. A Characterization of $\mathrm{opt}()$

For any Hilbert spaces $\mathcal{F}$ and $\mathcal{G}$, the *partial trace* is a trace-preserving linear map $\mathrm{tr}_\mathcal{G} : \mathbf{L}(\mathcal{F} \otimes \mathcal{G}) \to \mathbf{L}(\mathcal{F})$ defined as $\mathrm{tr}_\mathcal{G}(X \otimes W) = \mathrm{tr}(W)X$ for any $X \in \mathbf{L}(\mathcal{F})$ and $W \in \mathbf{L}(\mathcal{G})$ and extending to all of $\mathbf{L}(\mathcal{F} \otimes \mathcal{G})$ by linearity. The partial trace is *completely positive*, a consequence of which is that $\mathrm{tr}_\mathcal{G}(Z) \in \mathbf{Pos}(\mathcal{F})$ whenever $Z \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$.

For any $X \in \mathbf{Pos}(\mathcal{F})$ and any $v \in \mathcal{F} \otimes \mathcal{G}$, v is a *purification* of X if $\mathrm{tr}_\mathcal{G}(vv^*) = X$. Such a purification exists if and only if $\dim(\mathcal{G}) \geq \mathrm{rank}(X)$. Different purifications of X are *unitarily equivalent* in the sense that if $u, v \in \mathcal{F} \otimes \mathcal{G}$ are purifications of X then there exists a unitary matrix $U \in \mathbf{U}(\mathcal{G})$ such that $(I_\mathcal{F} \otimes U)u = v$.

We say that $X_1, \ldots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$ are *consistent with* $A_0, \ldots, A_{r-1} \in \mathbf{L}(\mathcal{F} \otimes \mathcal{G})$ if

$$\mathrm{tr}_\mathcal{G}\left(X_{i+1}\right) = \mathrm{tr}_\mathcal{G}\left(A_i X_i A_i^*\right) \qquad (2)$$

for every $i \in \{0, \ldots, r-1\}$ where X_0 always denotes the matrix $|0_{\mathcal{F}\otimes\mathcal{G}}\rangle\langle 0_{\mathcal{F}\otimes\mathcal{G}}|$. We call Eq. (2) the *consistency criterion*. The following lemma characterizes $\mathrm{opt}()$.

Lemma 1. For any $A_0, \ldots, A_r \in \mathbf{L}(\mathcal{F} \otimes \mathcal{G})$, $\mathrm{opt}(A_r, \ldots, A_0)$ is precisely the maximum of $\langle A_r^* A_r, X_r\rangle$ over all $X_1, \ldots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$ consistent with $A_0, \ldots, A_{r-1}$.

Proof. First we show that

$$\langle A_r^* A_r, X_r\rangle \leq \mathrm{opt}(A_r, \ldots, A_0)$$

for any $X_1, \ldots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$ consistent with $A_0, \ldots, A_{r-1}$. Let $u_0, \ldots, u_r \in \mathcal{F} \otimes \mathcal{G} \otimes \mathcal{H}$ be purifications of $X_0, \ldots, X_r$ with $u_0 = |0\rangle$. It follows that $A_i u_i$ is a purification of $A_i X_i A_i^*$ for every $i \in \{0, \ldots, r-1\}$. The consistency criterion and the unitary equivalence of purifications imply that there exists some unitary matrix $U_{i+1} \in \mathbf{U}(\mathcal{G}\otimes\mathcal{H})$ such that $u_{i+1} = U_{i+1} A_i u_i$, from which it follows that $u_r = U_r A_{r-1} \cdots A_1 U_1 A_0 |0\rangle$. We have

$$\begin{aligned} &\mathrm{opt}\left(A_r, \ldots, A_0\right) \\ &\geq \left\| A_r U_r A_{r-1} \cdots A_1 U_1 A_0 |0\rangle \right\|^2 = \left\| A_r u_r \right\|^2 \qquad (3) \\ &= u_r^* A_r^* A_r u_r = \mathrm{tr}\left(A_r^* A_r u_r u_r^*\right) \\ &= \mathrm{tr}\left(A_r^* A_r \,\mathrm{tr}_\mathcal{H}\left(u_r u_r^*\right)\right) = \mathrm{tr}\left(A_r^* A_r X_r\right) \\ &= \langle A_r^* A_r, X_r\rangle\,. \end{aligned}$$

Next, we show that the maximum is attained for some $X_1, \ldots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$. Choose any $U_1, \ldots, U_r \in \mathbf{U}(\mathcal{G} \otimes \mathcal{H})$ that attain the maximum in Eq. (1). Let $u_0, \ldots, u_r \in \mathcal{F}\otimes\mathcal{G}\otimes\mathcal{H}$ be defined by $u_{i+1} = U_{i+1} A_i u_i$ for every $i \in \{0, \ldots, r-1\}$ with $u_0 = |0\rangle$. Let $X_0, \ldots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$ be defined by $X_i = \mathrm{tr}_\mathcal{H}(u_i u_i^*)$ for every $i \in \{0, \ldots, r\}$. Eq. (3) now holds with equality by our choice of $U_1, \ldots, U_r$.

It remains to show that $X_1, \ldots, X_r$ are consistent with $A_0, \ldots, A_{r-1}$. For each $i \in \{0, \ldots, r-1\}$ we have

$$\begin{aligned} &\mathrm{tr}_\mathcal{G}\left(X_{i+1}\right) \\ &= \mathrm{tr}_{\mathcal{G}\otimes\mathcal{H}}\left(u_{i+1} u_{i+1}^*\right) = \mathrm{tr}_{\mathcal{G}\otimes\mathcal{H}}\left(U_{i+1} A_i u_i u_i^* A_i^* U_{i+1}^*\right) \\ &= \mathrm{tr}_{\mathcal{G}\otimes\mathcal{H}}\left(A_i u_i u_i^* A_i^*\right) = \mathrm{tr}_\mathcal{G}\left(A_i \,\mathrm{tr}_\mathcal{H}\left(u_i u_i^*\right) A_i^*\right) \\ &= \mathrm{tr}_\mathcal{G}\left(A_i X_i A_i^*\right). \end{aligned}$$

□

3.3. A Semidefinite Program for $\mathrm{opt}()$

In this subsection we formalize the reduction from $\mathrm{opt}()$ to the semidefinite program found in Ref. [14]. We start by briefly reviewing the relevant aspects of semidefinite programming (see for example Refs. [2, 24]). For any Hilbert space $\mathcal{H}$ we let $\mathbf{H}(\mathcal{H}) \subset \mathbf{L}(\mathcal{H})$ denote the set of all Hermitian matrices in $\mathbf{L}(\mathcal{H})$. The semidefinite programming problem that we use is stated as follows.

Problem (SDP).

Input. A Hermitian matrix $H \in \mathbf{H}(\mathcal{H})$, matrices $A_1, \ldots, A_m \in \mathbf{L}(\mathcal{H})$, scalars $a_1, \ldots, a_m \in \mathbb{C}$, a feasible solution $X_{\mathrm{init}} \in \mathbf{Pos}(\mathcal{H})$, a positive real number b, and an accuracy parameter $\varepsilon > 0$.

Assumptions. $X_{\mathrm{init}} \in \mathbf{Feas}(\mathcal{H})$ and $\| X \| \leq b$ for all $X \in \mathbf{Feas}(\mathcal{H})$ where the feasible solution set $\mathbf{Feas}(\mathcal{H})$ consists of all $X \in \mathbf{Pos}(\mathcal{H})$ satisfying $\langle A_i, X\rangle = a_i$ for every $i \in \{1, \ldots, m\}$.

Output. $X \in \mathbf{Feas}(\mathcal{H})$ such that $\langle H, X\rangle > \langle H, W\rangle - \varepsilon$ for every $W \in \mathbf{Feas}(\mathcal{H})$.

This problem can be solved in time polynomial in the bit length of the input data using interior point methods [20]. This variant of SDP is not a standard one, but it is also used in Refs. [16, 15]. The problem that we reduce to SDP is stated as follows.

Problem (OPT).

Input. Matrices $A_0, \dots, A_r \in \mathbf{L}(\mathcal{F} \otimes \mathcal{G})$ and an accuracy parameter $\varepsilon > 0$.

Output. A real number α satisfying

$$|\alpha - \operatorname{opt}(A_r, \dots, A_0)| < \varepsilon$$

and matrices $X_1, \dots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$ consistent with $A_0, \dots, A_{r-1}$ such that $\langle A_r^* A_r, X_r\rangle = \alpha$.

For both SDP and OPT it is assumed that the real and imaginary parts of all input numbers are rational numbers represented in binary notation. The rest of this subsection is devoted to proving the following theorem.

Theorem 2. OPT can be solved in time polynomial in the bit length of the input data.

We start with some notation. For any Hilbert space $\mathcal{H}$ and any matrix $B \in \mathbf{L}(\mathcal{H})$ we let $B[i,j]$ denote the $[i,j]$ entry of B. We also let $E^{i,j}_{\mathcal{H}} \in \mathbf{L}(\mathcal{H})$ denote the matrix with all entries equal to zero except for a 1 in the $[i,j]$ entry. Note that $B[i,j] = \langle E^{i,j}_{\mathcal{H}}, B\rangle$.

For any positive integer n we let $n\mathcal{H}$ denote the Hilbert space with dimension $n \cdot \dim(\mathcal{H})$. For any matrices $B_1, \dots, B_n \in \mathbf{L}(\mathcal{H})$, we let $(B_1, \dots, B_n) \in \mathbf{L}(n\mathcal{H})$ denote the block diagonal matrix

$$\begin{pmatrix} B_1 & & 0 \\ & \ddots & \\ 0 & & B_n \end{pmatrix}.$$

Our goal is to develop a set of linear constraints for SDP that characterizes consistency with $A_0, \dots, A_{r-1}$. Letting $\mathcal{R} = (r+1)(\mathcal{F} \otimes \mathcal{G})$, we start by describing constraints that ensure every feasible solution $X \in \mathbf{Pos}(\mathcal{R})$ is a block diagonal matrix of the form $(X_0, \dots, X_r)$ for some $X_0, \dots, X_r \in \mathbf{Pos}(\mathcal{F} \otimes \mathcal{G})$. For this task, the "brute force" method of simply forcing every off-block-diagonal entry to zero works just fine. In other words, we require that

$$\left\langle E^{i,j}_{\mathcal{R}}, X \right\rangle = 0$$

for all suitably chosen i and j. Using this same brute force technique, we set every entry of X_0 to indicate the matrix $|0\rangle\langle 0|$.

Next, we concentrate on the consistency criterion defined in Eq. (2). More notation: we define

$$\Xi_k : \mathbf{L}(2(\mathcal{F} \otimes \mathcal{G})) \to \mathbf{L}(\mathcal{R})$$

for all $k \in \{0, \dots, r-1\}$ so that, given $C \in \mathbf{L}(2(\mathcal{F} \otimes \mathcal{G}))$, $\Xi_k(C)$ is the matrix with all entries equal to zero except that C is embedded in $\Xi_k(C)$ in such a way that, if $X = (X_0, \dots, X_r) \in \mathbf{Pos}(\mathcal{R})$ is a block diagonal matrix, then

$$\langle \Xi_k(C), X\rangle = \langle C, (X_k, X_{k+1})\rangle .$$

We also define

$$T^{i,j} : \mathbf{L}(\mathcal{F} \otimes \mathcal{G}) \to \mathbf{L}(2(\mathcal{F} \otimes \mathcal{G}))$$

for all $i, j \in \{1, \dots, \dim(\mathcal{F})\}$ so that, given $A \in \mathbf{L}(\mathcal{F} \otimes \mathcal{G})$, $T^{i,j}(A)$ is the block diagonal matrix

$$\left(A^* \left(E^{i,j}_{\mathcal{F}} \otimes I_{\mathcal{G}} \right) A, -E^{i,j}_{\mathcal{F}} \otimes I_{\mathcal{G}} \right).$$

We have the following lemma.

Lemma 3. Let $X = (X_0, \dots, X_r) \in \mathbf{Pos}(\mathcal{R})$ be a block diagonal matrix with $X_0 = |0\rangle\langle 0|$. Then $X_1, \dots, X_r$ are consistent with $A_0, \dots, A_{r-1}$ if and only if X satisfies

$$\left\langle \Xi_k(T^{i,j}(A_k)), X \right\rangle = 0$$

for all $i, j \in \{1, \dots, \dim(\mathcal{F})\}$ and all $k \in \{0, \dots, r-1\}$.

Proof. We have

$$\begin{aligned}
&\left\langle \Xi_k(T^{i,j}(A_k)), X \right\rangle \\
&= \left\langle T^{i,j}(A_k), (X_k, X_{k+1}) \right\rangle \\
&= \left\langle A_k^* \left(E^{i,j}_{\mathcal{F}} \otimes I_{\mathcal{G}} \right) A_k, X_k \right\rangle - \left\langle E^{i,j}_{\mathcal{F}} \otimes I_{\mathcal{G}}, X_{k+1} \right\rangle \\
&= \left\langle E^{i,j}_{\mathcal{F}} \otimes I_{\mathcal{G}}, A_k X_k A_k^* \right\rangle - \left\langle E^{i,j}_{\mathcal{F}} \otimes I_{\mathcal{G}}, X_{k+1} \right\rangle \\
&= \left\langle E^{i,j}_{\mathcal{F}}, \operatorname{tr}_{\mathcal{G}} \left(A_k X_k A_k^* \right) \right\rangle - \left\langle E^{i,j}_{\mathcal{F}}, \operatorname{tr}_{\mathcal{G}} \left(X_{k+1} \right) \right\rangle \\
&= \operatorname{tr}_{\mathcal{G}}(A_k X_k A_k^*)[i,j] - \operatorname{tr}_{\mathcal{G}}(X_{k+1})[i,j].
\end{aligned}$$

Of course, $\operatorname{tr}_{\mathcal{G}}(A_k X_k A_k^*) = \operatorname{tr}_{\mathcal{G}}(X_{k+1})$ if and only if their entrywise difference is zero, from which the lemma follows. □

The constraints of Lemma 3 are based on similar constraints found in Ref. [15]. We have thus established a polynomial number of equality constraints that characterize the consistency criterion.

We complete the proof of Theorem 2 by providing the remaining inputs to SDP. The objective matrix $H \in \mathbf{H}(\mathcal{R})$ is the block diagonal matrix $(0, \dots, 0, A_r^* A_r)$. The initial feasible solution $X_{\text{init}} \in \mathbf{Pos}(\mathcal{R})$ can be taken to be the block diagonal matrix $(X_0, \dots, X_r)$ where $X_{i+1} = A_i X_i A_i^*$ for every $i \in \{0, \ldots, r-1\}$. The bound b for all feasible solutions can be taken to be

$$b = 1 + \sum_{i=0}^{r-1} \prod_{j=0}^{i} \| A_j \|^2 .$$

Note that if $A_0, \ldots, A_{r-1}$ are unitary then we may take $b = r + 1$ as one might expect.

3.4. Putting it All Together

Now that we have established a polynomial-time solution to OPT, most of the work towards proving $\mathsf{QRG} \subseteq \mathsf{NEXP}$ is done and only two minor issues remain. The first issue is that the provers' quantum circuits are unbounded. In order to store in memory the unitary matrices corresponding to these circuits, we require that they act on at most a polynomial number of qubits.

Fortunately, it is easy to show that any prover can be simulated by another prover who uses no more qubits than the verifier. To see this fact it suffices to note that, at any point in the protocol, the verifier's qubits purify the state of the private qubits of each of the provers. Hence, each of those states is supported by a Hilbert space with dimension no higher than $\dim(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$. This issue is also discussed in Refs. [15] and [17, Theorem 10].

The other issue is that of numerical error introduced by finite-precision approximations of continuous quantities. In this paper we often depend upon the fact that some matrix A can be approximated "accurately enough" for some purpose by a finite-precision matrix $\tilde{A}$. In every such instance, we make the implicit assumption that $\tilde{A}$ can be computed in time polynomial in some appropriate parameters, such as a desired error bound ε or the dimensions of A.

We now formalize the main result of this section as a corollary of Theorem 2.

Corollary 4. $\mathsf{QRG} \subseteq \mathsf{NEXP}$.

Proof. Let $L \in \mathsf{QRG}$, let $V = (V_0, \ldots, V_r)$ be an r-round verifier witnessing this fact, and fix any input $x \in \Sigma^*$. We now specify a nondeterministic exponential-time Turing machine M that decides L:

1. Nondeterministically guess matrix approximations $\tilde{Y}_1, \ldots, \tilde{Y}_r$ of unitary matrices $Y_1, \ldots, Y_r \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$.
2. Compute matrix approximations $\tilde{V}_0, \ldots, \tilde{V}_r$ of $V_0, \ldots, V_r$.
3. Compute a real approximation α of $\mathrm{opt}(\Pi_{\mathrm{rej}} \tilde{V}_r \tilde{Y}_r, \ldots, \tilde{V}_1 \tilde{Y}_1, \tilde{V}_0)$ by solving OPT. If $\alpha < 1/2$ then accept, otherwise reject.

To see that M decides L, we note that it suffices to compute approximations accurate enough so that

$$|\alpha - \mathrm{opt}(\Pi_{\mathrm{rej}} V_r Y_r, \ldots, V_1 Y_1, V_0)| < 1/4.$$

To see that M runs in time exponential in $|x|$, we note that, as mentioned previously, we may assume that $\dim(\mathcal{Y}) \leq \dim(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$. Hence, the input matrices to OPT have size at most exponential in $|x|$. The result follows from Theorem 2. □

The next corollary follows by virtue of the fact that QRG is closed under complementation.

Corollary 5. $\mathsf{QRG} \subseteq \mathsf{NEXP} \cap \mathsf{coNEXP}$.

It is worth mentioning that even if an exponential number of rounds are permitted, the corresponding semidefinite program for OPT still has only exponential size, provided that each of the verifier's circuits is of polynomial size as usual.

4. A Classical Upper Bound for SQG

In this section we prove that $\mathsf{SQG} \subseteq \mathsf{EXP}$. We hinted in Section 1 that this containment is achieved via a reduction to convex feasibility with exponential-size semidefinite matrices. The intuition behind this reduction is as follows.

Because the provers in a short quantum game only play once and because the yes-prover in such a game plays first, the actions of the yes-prover are completely characterized by the state of the verifier's qubits upon receipt of the yes-prover's message. Given a candidate matrix X_Y for that state, we can compute the optimal response X_N for the no-prover via the polynomial-time solution to OPT from Section 3. Using X_N, we show how to compute a matrix B such that the quantity $\langle B, X_Y \rangle$ is small when X_Y indicates a winning yes-prover and large when X_N is a response that wins against X_Y.

In the latter case, B defines a hyperplane that separates X_Y from the (possibly empty) bounded convex set of winning yes-provers. This hyperplane can be used by the ellipsoid method to create a refined candidate X'_Y. The ellipsoid method submits at most a polynomial number of refinements in this manner before it deduces whether or not the set of winning yes-provers is empty [13, 11].

An algorithm that computes a separating hyperplane for a given candidate in this way is called a *separation oracle*. This section is devoted to constructing an efficient separation oracle for the aforementioned bounded convex set of winning yes-provers.

We begin by describing the set of winning yes-provers for a short quantum game as an intersection of half-spaces. This description provides a general form for the separating hyperplanes that we compute for use with the ellipsoid method.

Lemma 6. Let $A_0, A_1, A_2 \in \mathbf{L}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ and let $\beta \in \mathbb{R}$. The following are equivalent:

1. There exists a unitary matrix $Y \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$ such that $\operatorname{opt}(A_2, A_1 Y A_0) \leq \beta$.
2. There exists a positive semidefinite matrix $X \in \mathbf{Pos}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ consistent with A_0 such that
$$\langle A_1^* N^* A_2^* A_2 N A_1, X\rangle \leq \beta$$
for every unitary matrix $N \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$.

Proof. Assume that item 1 holds and let $Y \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$ witness this fact. Let

$$X = \operatorname{tr}_{\mathcal{Y}} \left(Y A_0 |0\rangle\langle 0| A_0^* Y^* \right),$$

which is clearly positive semidefinite and consistent with A_0. Choose any unitary matrix $N \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$. We have

$$\begin{aligned}\beta &\geq \operatorname{opt}(A_2, A_1 Y A_0) \geq \| A_2 N A_1 Y A_0 |0\rangle \|^2 \\ &= \langle A_1^* N^* A_2^* A_2 N A_1, Y A_0 |0\rangle\langle 0| A_0^* Y^*\rangle \\ &= \langle A_1^* N^* A_2^* A_2 N A_1, X\rangle .\end{aligned}$$

Now assume that item 2 holds and let $X \in \mathbf{Pos}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ witness this fact. Let $u \in \mathcal{S}_{\mathcal{N}}$ be any purification of X. Because X is consistent with A_0, it follows from the unitary equivalence of purifications that there exists some unitary matrix $Y \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$ such that $Y A_0 |0\rangle = u$ and therefore

$$X = \operatorname{tr}_{\mathcal{Y}}(uu^*) = \operatorname{tr}_{\mathcal{Y}} \left(Y A_0 |0\rangle\langle 0| A_0^* Y^* \right).$$

Choose any unitary matrix $N \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$. We have

$$\begin{aligned}\beta &\geq \langle A_1^* N^* A_2^* A_2 N A_1, X\rangle \\ &= \langle A_1^* N^* A_2^* A_2 N A_1, Y A_0 |0\rangle\langle 0| A_0^* Y^*\rangle \\ &= \| A_2 N A_1 Y A_0 |0\rangle \|^2 .\end{aligned}$$

Because N was chosen arbitrarily, it follows that $\operatorname{opt}(A_2, A_1 Y A_0) \leq \beta$. □

Let $L \in \mathsf{SQG}$ and let $V = (V_0, V_1, V_2)$ be a verifier with error $\beta \in (0, 1/2)$ witnessing this fact. We wish to decide whether there exists a yes-prover Y such that $\operatorname{opt}(\Pi_{\text{rej}} V_2, V_1 Y V_0) \leq \beta$. By Lemma 6, this task is equivalent to ascertaining the emptiness of the convex set $\mathbf{Yes}(V, \beta)$ of all $X \in \mathbf{Pos}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ consistent with V_0 such that

$$\left\langle V_1^* N^* V_2^* \Pi_{\text{rej}}^* \Pi_{\text{rej}} V_2 N V_1, X \right\rangle \leq \beta$$

for all $N \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$. Note that any such X has spectral norm at most 1 and so $\mathbf{Yes}(V, \beta)$ is bounded.

We already intimated that the ellipsoid method can decide the emptiness of a given bounded convex set of semidefinite matrices in polynomial time, provided access to an efficient separation oracle for that set. In particular, the separation oracle we seek for $\mathbf{Yes}(V, \beta)$ solves the following problem (compare with Ref. [11, Theorem 3.2.1]).

Problem (SEP(V, β)).

Input. A matrix $X \in \mathbf{Pos}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ consistent with V_0 and an error parameter $\delta > 0$.

Output. One of the following:

1. A Hermitian matrix $B \in \mathbf{H}(\mathcal{S}_{\mathcal{Y},\mathcal{N}})$ such that $\langle B, W\rangle < \langle B, X\rangle + \delta$ for every $W \in \mathbf{Yes}(V, \beta)$.
2. An assertion that there exists $X' \in \mathbf{Yes}(V, \beta)$ with $\| X - X' \| < \delta$.

Some remarks: first, as with SDP and OPT, we assume that the real and imaginary parts of all input numbers (including β and the entries of V_0, V_1, V_2) are rational numbers represented in binary notation. Second, we assume that the input matrix X is positive semidefinite and consistent with V_0 so that we may concentrate on the more interesting aspects of our separation oracle. This assumption is justified because more conventional methods can easily yield a separating hyperplane when X does not satisfy these criteria.

Finally, SEP implements a *weak* separation oracle in the following sense. Case 1 above allows us to reject a "good" yes-prover if it is close to a "bad" yes-prover. Conversely, case 2 implies that we may accept a "bad" yes-prover so long as it is close to a "good" yes-prover. We will see in the proof of Corollary 8 that the leeway afforded to us by the bounded-error requirement for quantum refereed games permits this convenient relaxation.

We now prove the following theorem.

Theorem 7. SEP(V, β) can be solved in time polynomial in the bit lengths of (V, β) and the input data.

Proof. The following is a polynomial-time algorithm for SEP(V, β):

1. Let $Y \in \mathbf{U}(\mathcal{Y} \otimes \mathcal{M}_Y)$ be a unitary matrix satisfying
$$\operatorname{tr}_{\mathcal{Y}} \left(Y V_0 |0\rangle\langle 0| V_0^* Y^* \right) = X.$$
Compute a matrix approximation $\tilde{Y}$ of Y.
2. Compute a real approximation α of $\operatorname{opt}(\Pi_{\text{rej}} V_2, V_1 \tilde{Y} V_0)$ by solving OPT. If $\alpha \leq \beta$ then halt and output case 2.
3. Otherwise, OPT has computed a matrix $Z \in \mathbf{Pos}(\mathcal{S}_{\mathcal{N}})$ consistent with $V_1 \tilde{Y} V_0$ such that
$$\left\langle V_2^* \Pi_{\text{rej}}^* \Pi_{\text{rej}} V_2, Z \right\rangle = \alpha > \beta. \tag{4}$$

Let $N \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$ be a unitary matrix satisfying

$$\mathrm{tr}_{\mathcal{N}} \left(N V_1 \tilde{Y} V_0 |0\rangle\langle 0| V_0^* \tilde{Y}^* V_1^* N^* \right) = Z.$$

Compute a matrix approximation $\tilde{N}$ of N. Halt and output case 1 with

$$B = V_1^* \tilde{N}^* V_2^* \Pi_{\mathrm{rej}}^* \Pi_{\mathrm{rej}} V_2 \tilde{N} V_1.$$

We now verify the correctness of this algorithm. Existence of the unitary matrices Y and N in steps 1 and 3 follows from the unitary equivalence of purifications and the consistency of X and Z with V_0 and $V_1 \tilde{Y} V_0$ respectively.

Suppose first that the halting condition is satisfied in step 2. Provided that our approximations $\tilde{Y}$ and α are accurate enough, it must be the case that $\mathrm{opt}(\Pi_{\mathrm{rej}} V_2, V_1 Y V_0)$ is smaller than $\beta + \delta$. It follows from Lemma 6 that

$$\left\langle V_1^* N^* V_2^* \Pi_{\mathrm{rej}}^* \Pi_{\mathrm{rej}} V_2 N V_1, X \right\rangle < \beta + \delta$$

for all $N \in \mathbf{U}(\mathcal{M}_N \otimes \mathcal{N})$. In other words, X is at least "close" to $\mathbf{Yes}(V, \beta)$ as required by output case 2.

Next, suppose that the algorithm proceeds to step 3. Once again, if $\tilde{Y}$, α, and $\tilde{N}$ are accurate enough then

$$\begin{aligned} &\left\langle V_2^* \Pi_{\mathrm{rej}}^* \Pi_{\mathrm{rej}} V_2, Z \right\rangle \\ &= \left\langle V_1^* N^* V_2^* \Pi_{\mathrm{rej}}^* \Pi_{\mathrm{rej}} V_2 N V_1, \tilde{Y} V_0 |0\rangle\langle 0| V_0^* \tilde{Y}^* \right\rangle \\ &< \langle B, X \rangle + \delta/2. \end{aligned}$$

Furthermore, for any $W \in \mathbf{Yes}(V, \beta)$ we have

$$\beta \geq \left\langle V_1^* N^* V_2^* \Pi_{\mathrm{rej}}^* \Pi_{\mathrm{rej}} V_2 N V_1, W \right\rangle > \langle B, W \rangle - \delta/2.$$

It follows from Eq. (4) that $\langle B, W \rangle < \langle B, X \rangle + \delta$ as required by output case 1.

That this algorithm runs in polynomial time follows from the existence of a polynomial-time solution to OPT (Theorem 2). □

We now formalize the main result of this section as a corollary of Theorem 7.

Corollary 8. SQG $\subseteq$ EXP.

Proof. Let $L \in$ SQG, let $V = (V_0, V_1, V_2)$ be a verifier with error $1/4$ witnessing this fact, and fix any input $x \in \Sigma^*$. We now specify a deterministic exponential-time Turing machine M that decides L:

1. Compute matrix approximations $\tilde{V} = (\tilde{V}_0, \tilde{V}_1, \tilde{V}_2)$ of V_0, V_1, V_2.

2. Use the ellipsoid method with an oracle for SEP$(\tilde{V}, 1/3)$ to decide the emptiness of $\mathbf{Yes}(\tilde{V}, 1/3)$. If it is empty then reject, otherwise accept.

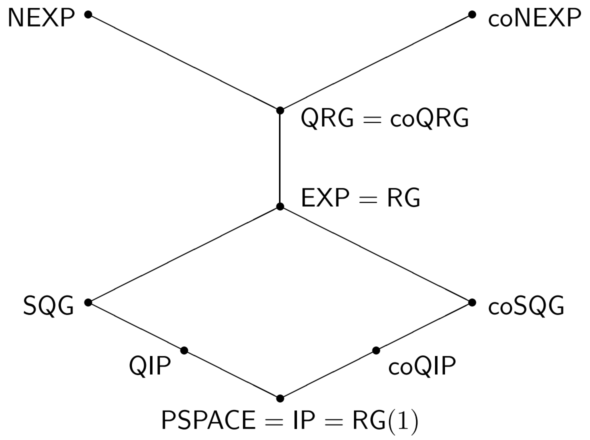

Figure 2. Relationships between complexity classes discussed in this paper

That M decides L follows from the correctness of the ellipsoid method and our solution to SEP. The reason that M solves SEP$(\tilde{V}, \beta)$ with $\beta = 1/3$ in step 2 is to allow for finite-precision error and an overly picky separation oracle that might otherwise reject every candidate yes-prover when $\beta = 1/4$, even in the case $x \in L$.

That M runs in time exponential in $|x|$ follows from the polynomiality of the ellipsoid method, Theorem 7, and the fact that all matrices have size at most exponential in $|x|$. □

5. Conclusion

Figure 2 summarizes the relationships between the complexity classes considered in this paper where RG denotes the class of languages with polynomial-round classical refereed games and RG(1) denotes the class of languages with one-round classical refereed games as defined in Ref. [6].

It is interesting to note that the techniques of Section 4 extend easily to quantum refereed games in which the verifier exchanges multiple messages with the yes-prover and then multiple messages with the no-prover. Indeed, even if an exponential number of messages are exchanged in this manner, the game can still be decided in deterministic exponential time provided that each of the verifier's circuits is of polynomial size as usual. The complexity class corresponding to games of this strange form might not be closed under complementation, yet it is trivially seen to contain both QIP and coQIP—a property that SQG is not known to satisfy.

At first, it might seem as though the ellipsoid method could be applied recursively to put all of QRG inside EXP. However, it is unclear how to generate a separating hyperplane for messages beyond the first round. It is nonetheless curious that $\mathsf{EXP} \subseteq \mathsf{QRG} \subseteq \mathsf{NEXP} \cap \mathsf{coNEXP}$. Is there evidence to suggest that $\mathsf{QRG} = \mathsf{EXP}$?

6. Acknowledgements

The author is grateful to John Watrous for valuable discussions concerning semidefinite programming and to several anonymous referees for helpful suggestions and for pointing out the existence of Ref. [14]. This research was supported in part by Canada's NSERC and by a graduate student scholarship from the Province of Alberta.

References

[1] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 20–30, 1998.

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

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

[4] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer and System Sciences*, 36(2):254–276, 1988.

[5] M. Ben-Or, S. Goldwasser, J. Kilian, and A. Wigderson. Multi-prover interactive proofs: how to remove intractability assumptions. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing*, pages 113–131, 1988.

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

[7] U. Feige and A. Shamir. Multi-oracle interactive protocols with constant space verifiers. *Journal of Computer and System Sciences*, 44(2):259–271, 1992.

[8] U. Feige, A. Shamir, and M. Tennenholtz. The noisy oracle problem. In *Advances in Cryptology—Proceedings of CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 284–296. Springer-Verlag, 1990.

[9] J. Feigenbaum, D. Koller, and P. Shor. A game-theoretic classification of interactive complexity classes. In *Proceedings of the 10th Annual IEEE Conference on Structure in Complexity Theory*, pages 227–237, 1995.

[10] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer-Verlag, second corrected edition, 1988.

[12] G. Gutoski and J. Watrous. Quantum interactive proofs with competing provers. In *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer-Verlag, 2005. cs.CC/0412102.

[13] L. Khachiyan. A polynomial time algorithm in linear programming. *Soviet Mathematics Doklady*, 20:191–194, 1979.

[14] A. Kitaev. Quantum coin-flipping. MSRI lecture. Transparencies available at http://www.msri.org, 2002.

[15] A. Kitaev and J. Watrous. Quantum interactive proof systems. In preparation. A preliminary version appeared as Ref. [16].

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

[17] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *Journal of Computer and System Sciences*, 66(3), 2003. cs.CC/0102013.

[18] D. Koller and N. Megiddo. The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, 4(4):528–552, 1992.

[19] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM*, 39(4):859–868, 1992.

[20] Y. Nesterov and A. Nemirovski. Interior point polynomial algorithms in convex programming. *SIAM Studies in Applied Mathematics*, 13, 1994.

[21] J. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984.

[22] A. Shamir. IP = PSPACE. *Journal of the ACM*, 39(4):869–877, 1992.

[23] A. Shen. IP = PSPACE: simplified proof. *Journal of the ACM*, 39(4):878–880, 1992.

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

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