Space-efficient Simulations of Quantum Interactive Proofs

by

Xiaodi Wu

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in the University of Michigan
2013

Doctoral Committee:

Associate Professor Yaoyun Shi, Chair
Professor John Patrick Hayes
Professor Igor L. Markov
Professor Kim A. Winick

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF SYMBOLS

$\mathcal{X}, \mathcal{Y}$:	complex Euclidean spaces $\mathcal{X}, \mathcal{Y}$.
$A^\star$:	the conjugate transpose of operator A.
$\langle A, B \rangle$:	the inner-product of operators A and B.
$\mathrm{L}(\mathcal{X}, \mathcal{Y})$**:**	the space of all linear mappings (or *operators*) from $\mathcal{X}$ to $\mathcal{Y}$ ($\mathrm{L}(\mathcal{X})$ short for $\mathrm{L}(\mathcal{X}, \mathcal{X})$).
$\mathrm{Herm}(\mathcal{X})$:	the set of Hermitian operators over space $\mathcal{X}$.
$\mathrm{Pos}(\mathcal{X})$:	the set of positive semideinite operators over space $\mathcal{X}$.
$\mathrm{D}(\mathcal{X})$:	the set of density operators over space $\mathcal{X}$.
$\|A\|_p$:	Schatten p-norm of operator A.
$\|A\|_\infty$:	the spectral norm of operator A.
$\|A\|_\mathrm{F}$:	the Frobenius norm of operator A.
$\|A\|_\mathrm{Tr}$:	the trace norm of operator A.
$\mathrm{F}(\sigma, \rho)$:	the fidelity between quantum states σ and ρ.
$J(\Psi)$:	the Choi-Jamiolkowski representation of super-operator Ψ.
$\|\Psi\|_\diamond$:	the diamond norm of super-operator Ψ.

ABSTRACT

Interactive proof systems form an important complexity model that has been central to many prominent results in computational complexity theory, such as those on probabilistically checkable proofs, hardness of approximation, and fundamental cryptographic primitives. In this thesis, we study the *quantum-enhanced* version of interactive proof systems, in which each party has access to quantum computing resources. We focus on its power and limitations, which lead us to the following results.

- We provide an alternative and conceptually simpler proof of Jain et al.'s recent breakthrough result, which demonstrates that the expressive power of quantum interactive proof systems coincides with that of its classical counterpart, and therefore PSPACE.
- We determine the complexity of several variants of quantum competing-prover interactive proof systems, which introduce zero-sum games into interactive proof systems. In particular, we prove that the complexity class of two-turn quantum refereed games coincides with PSPACE, answering an important open problem of Jain et al.'s. Our result suggests that a more general model called *double quantum interactive proof systems* coincides with PSPACE, which subsumes and unifies all previous results on short refereed games.
- We contribute to the study of two-prover quantum Merlin-Arthur games (QMA(2)), which exhibit an intriguing tradeoff between an intrinsic quantum ingredient, *entanglement*, and computational power. We prove a PSPACE upper bound for a variant of QMA(2) that is to date the most general one known in PSPACE. We also pro-

vide a quasi-polynomial time algorithm for optimizing linear functions over separable states, giving an alternative to the one proposed by Brandao et al.

Our main technical contribution behind the above results is the *equilibrium value method* for obtaining space-efficient algorithms for a class of optimization problems that arise naturally in quantum computation.

We also contribute to QMA-complete problems, where we demonstrate that the "Non-Identity Check" problem remains QMA-complete on circuits of poly-logarithmic depths, improving upon polynomial depths from previous results.

CHAPTER 1

Introduction

1.1 Quantum Information and Computation

Quantum mechanics: created at the beginning of the last century, quantum mechanics describes physical systems that are otherwise indescribable by classical physics at an atomic scale. At the heart of quantum mechanics are a few counterintuitive postulates that are fundamentally different from the classical world. One particular postulate is that the state of a quantum system can be in a superposition of many different classical states, which can exhibit interference during its evolutions. Another marvelous property is that entanglement shared among spatially separated systems can display "nonlocal" effects. Immediately after its inception, this theory has been at the center of exciting developments in various disciplines of science. The last three decades have witnessed a huge body of research focused on exploiting quantum mechanical features to perform computational, communication, cryptographic, and information-theoretic tasks. Research in this perspective also reinforces our understanding of quantum mechanics and the nature.

Quantum computation: in early 1980s, Benioff [16] and Feynman [42] studied the task of simulating quantum mechanical phenomena on existing models of computation such as Turing machines. It turned out that a new computational model based on the laws of quantum mechanics is necessary to perform such tasks because there seem to be forbidding

difficulties to execute simulations on classical computers. This observation led to the seminal work of David Deutsch [35], in which the concept of the quantum computer was made precise in terms of quantum Turing machines. In the same paper, Deutsch showed that a quantum computer can solve a black box problem substantially faster than any classical computer. Our understanding about quantum computation was improved in the subsequent decade; Bernstein and Vazirani [17] formalized the quantum complexity theory; Deutsch and Jozsa [36] showed that a quantum computer can solve a certain type of black-box problem *exponentially* faster than a classical computer. One of the most important discoveries in this field is Shor's efficient quantum algorithms [104] for two important problems (i.e., integer factorization and discrete logarithms) that are not known to admit efficient solutions on a classical computer.

The rapid development of quantum information and computation has already affected various areas in the theory of computation. In the algorithm aspect, besides the famous Shor's algorithm for integer factorization and discrete logarithms, other quantum algorithms (e.g., Grover's algorithm for unstructured database search and quantum random walk algorithms) are known to out-perform their classical counter-parts. Such an impact also exists in the field of cryptography, in which we are most likely to see practical applications in the near future. The peculiar properties of quantum mechanics, especially the non-cloning property, enable the existence of an unconditionally secure quantum key distribution protocol for the first time. On the other hand, as Shor's algorithm renders most modern cryptography systems based on the hardness of integer factorization insecure, it is of great interest to develop post-quantum cryptography systems that are secure against quantum adversaries.

Interactions with other fields: one of the most important concepts discovered during the study of quantum information was *quantum entanglement*. Moreover, the study of dif-

ferent types of quantum correlations and local Hamiltonian systems has also shed light on the recent progress in the study of condensed matter physics. For theoretical computer science, arguments that arise from quantum information have proven instrumental in tackling seemingly unrelated classical problems. One of the applications of quantum arguments in theoretical computer science is to prove the first exponential lower bound on two-query Locally Decodable Codes (LDCs). It is also known that quantum arguments can lead to simplifications of proofs for important results [1, 33]. Other examples include lower bounds in communication complexity and polynomial approximation.

1.2 Quantum Computational Complexity

In this section, we provide a comprehensive introduction of the main topic of this dissertation, *quantum computational complexity*, and the particular model we focus on, *efficient verification of proofs*.

Computational complexity theory studies the inherent difficulty, or hardness, of computational problems. Hardness is typically measured in terms of the resources required to solve a given problem, such as the number of steps of a deterministic Turing machine. Typically, computational models and resource constraints are *physically* motivated. A prominent example of this is the class of polynomial-time computable functions, whose relevance is ultimately derived from physical considerations.

However, it took the community significant efforts to realize that quantum mechanics should have implications to computational complexity theory. We mentioned some facts about the history of formal models of quantum computation in the previous section. It is only through these remarkable discoveries and ideas of several researchers that this potential has become evident. In particular, Shor's polynomial-time quantum factoring and discrete-logarithm algorithms have provided strong support for the conjecture that

quantum and classical computers yield different notions of computational hardness. Other quantum complexity-theoretic concepts, such as efficient verification of quantum proofs, suggest a wider extent to which quantum mechanics could influence computational complexity.

The principal aim of quantum computational complexity theory is to understand the implications of quantum physics to classical computational complexity theory. In this dissertation, we focus on the influence of quantum mechanics on computational complexity in the context of *efficient verification of proofs*, or equivalently the *interactive proof system* model. Interactive proof systems form an important complexity model that has been central to many prominent results in computational complexity theory, such as those on probabilistically checkable proofs, hardness of approximation, and fundamental cryptographic primitives. The famous complexity class NP appears to be the simplest one in this model.

In the following we provide a formal introduction (including the definition and a brief survey of prior results) of this model and its variants, which contain the standard interactive proof systems either with a single prover or with competing provers. Moreover, we also study the quantum analogue of NP called *Quantum Merlin-Arthur* (QMA) games. In particular, we are interested in two-prover quantum Merlin-Arthur games (QMA(2)), which exhibit a counter-intuitive and quantum-unique phenomenon that the enforcement of no entanglement potentially increases computational power.

1.2.1 Efficient Verification of Proofs

Efficient proof verification implicitly involves two players, the *prover* and the *verifier*, and that the verification procedure is assumed to be efficient. It has become one of the most widely studied notions in theoretical computer science, with broad applications. One of the

motivations to investigate this model is to study the difference between the following two fundamental tasks in computational complexity: proof generation and proof verification.

Let $(L_{\text{yes}}; L_{\text{no}})$ be a promise decision problem and x be an input string. The aim is to establish that $x \in L_{\text{yes}}$. The task of efficient proof generation is to design an efficient (polynomial-time) algorithm that outputs a proof such that if the claim is true, then the algorithm accepts the proof and if the claim is false, then the algorithm rejects the proof. For example, the complexity class P is a class of decision problems that admit efficient proof generation procedures.

The task of efficient proof verification is to design an efficient algorithm that verifies the correctness of a *given* proof of the claim $x \in L_{\text{yes}}$. In other words, the algorithm takes in two inputs, the string x and the proof, and then efficiently verifies the correctness of the claim. For instance, the complexity class NP is such a class of decision problems that admit efficient proof verifications. In this sense, the problem about the intrinsic complexity of the two tasks, proof generation and proof verification, is equivalent to the famous P-*versus*-NP problem, which asks whether the two classes are the same.

We remark that the above separation demonstrates the difference between proof generation and proof verification in traditional problem-solving procedures. In this way, we generalize the conventional concept of mathematical proofs that are *static* and are verified in a *deterministic* way to a convincing procedure between an all-powerful but untrustful prover and a computationally bounded verifier. The new model allows us to introduce various ingredients that would lead to more powerful complexity models and classes. Among the most important ones are *interaction* and *randomness*. Intuitively, interactions (i.e., two-way communication between the prover and the verifier) will increase the chance of any efficient verifier to catch a cheating prover. Similarly, randomness could make the verifier's questions more difficult to answer and thus potentially increases the computational

power of the entire proof system. Of course, any proof verification system still needs to be efficient and satisfy the following three properties.

- *Completeness*. For every $x \in L_{\text{yes}}$, there exists a behavior of the prover that causes the verifier to accept x with high probability.
- *Soundness*. For every $x \in L_{\text{no}}$, regardless of the behavior of the prover, the verifier will reject x with high probability.
- *Efficiency*. The complete verification procedure, including all rounds of communication between the prover and the verifier, and the verifier's internal computation should run in polynomial time in terms of the size of the input.

The efficient interactive proof systems proposed above turn out to be an important computational model, as it provides alternative and insightful characterizations of more natural complexity classes defined in terms of consumed time or space. The famous complexity class NP is one such example. The generalized single-prover interactive proofs turn out to characterize polynomial space [101, 100, 86] and the multi-prover version provides the same expressive power as nondeterministic exponential time [15]. All these results shed light on alternative characterizations of conventional computational resources, and are considered among the most important results in theoretical computer science since Cook-Levin's theorem.

The proof verification scenario also deepens our understanding about proofs themselves. One prominent example in this regard is the *zero-knowledge* proof in cryptography, in which the untrustful prover tries to convince the verifier that a given statement is true, without conveying any additional information more than the mere fact that the statement is indeed true. This is very useful in cryptographic scenarios where the ability to prove the statement requires some secret information from the prover. Zero-knowledge proofs, thus, prevent the verifier to prove the statement to anyone else. Another important exam-

ple is the *probabilistically checkable proofs* (i.e., PCP theorem) that certify every decision problem in NP with a constant number of queries into the proofs and a logarithmic amount of randomness. The PCP theorem serves as the cornerstone of the study of the hardness of approximation, which investigates the inherent obstacles in designing efficient approximation algorithms for various optimization problems.

In the following, we shall elaborate on the models and their variants studied in this dissertation. All the models extend easily to quantum setting, in which provers and verifiers have access to quantum computers. Also, classical messages (or proofs) will be replaced by quantum messages (or proofs). The requirement on completeness, soundness, and efficiency, however, remains the same.

1.2.2 Quantum Interactive Proof Systems

Interactive proof systems evolve from the standard efficient verification of proofs model with extra ingredients (i.e., interaction and randomness). In this model (denoted IP), a computationally bounded verifier exchanges messages of at most polynomial length with an all-powerful prover. For any input x, the prover tries to convince the verifier to accept, whereas the verifier will make its own decision based on the interacting process.

The expressive power of this interactive proof system model is completely characterized [101, 100, 86] by the well-known relationship

$$\text{IP=PSPACE}$$

through the technique commonly known as *arithmetization*. Many variants of the interactive proof system model have been studied, such as public-coin interactive proofs [8, 9, 48], multi-prover interactive proofs [15], and zero-knowledge interactive proofs [47], as well as competing-prover interactive proofs [40].

The *quantum interactive proof system* (denoted QIP) is defined [74, 110] in a similar

way to its classical counterparts except that the verifier and the prover have access to quantum resources and exchange quantum messages. Similarly, several variants of quantum interactive proof systems have also been studied, including the ordinary quantum interactive proofs [110, 74], public-coin quantum interactive proofs [88], zero-knowledge quantum interactive proofs [112, 76, 54], and multi-prover quantum interactive proofs [68, 77]. The complexity class QIP, known as the set of languages admitting quantum interactive proof systems, satisfies [74]

$$\text{PSPACE=IP} \subseteq \text{QIP} \subseteq \text{EXP}.$$

Pinning down the complexity class QIP between EXP and PSPACE was open ever since and was considered one of the most important problems in quantum computational complexity.

There are a few interesting properties about quantum interactive proof systems, among the most important ones of which is that it only requires three-turn interactions. Let QIP(r) denote the subset of QIP where the interaction is r-turn. This implies that QIP(3)=QIP, which is contrary to the classical case where constant-turn interactions lead to the complexity class AM, weaker than the whole IP. We can also build similar connections for $r = 0, 1$, where QIP(0)=BQP and QIP(1)=QMA. However, the complexity class QIP(2) remains mysterious because no clear characterization exists for that class.

Recently, Jain et al. [60] solved the complexity of QIP and concluded that the expressive power of quantum interactive proof systems coincides with its classical counterpart, i.e., QIP=IP=PSPACE. Their result is crucially based on the fact that three-turn quantum Merlin-Arthur games (denoted QMAM) have the same expressive power as quantum interactive proof systems. This observation leads to a simplified formulation of QIP as a semidefinite program over density operators. Then they applied Arora-Kale's [67] primal-dual approach for solving semidefinite programs through the use of the matrix multiplica-

tive weight update method. The later one admits space-efficient implementations. However, the resultant proof is still quite involved and less accessible to non-expert readers.

Our contributions: we find out that Jain et al. did not make full use of the properties of quantum interactive proof systems in the sense that they oversaw the possibility to use a QIP-complete problem as a start point. Moreover, the choice of Arora-Kale's primal dual method for solving semidefinite programs is sufficient but not elegant. In particular, Arora-Kale's algorithm, when applied, incurs a few unnecessary technical difficulties.

Our proof starts with one QIP-complete problem called the *close images* problem [74]. Our crucial observation is to imagine this promise problem as a zero-sum game between two well-designed players. Thus, in order to solve this promise problem, it suffices to approximate the equilibrium value of that particular zero-sum game. This separates our proof from Jain et al's which formulates QIP as semidefinite programs. Then we resort to the well-known application of the multiplicative update method to approximate equilibrium values of zero-sum games. We apply the same matrix version of the multiplicative weight update method, however, in a quite different way from Arora-Kale's approach, as their approach is for semidefinite programs. These observations lead to a conceptually very simple and modularized proof of QIP=PSPACE.

Extending our observations above to concrete mathematics, we develop a framework called the *equilibrium value method*. This general idea is simple and might be known before. Our contributions are technical solutions to an interesting class of problems that arise naturally in quantum computation. In the following, we will see how this framework could be applied in more sophisticated settings and lead to stronger results.

1.2.3 Quantum Refereed Games

In this section we introduce the variant of interactive proof systems induced from competitive (or zero-sum) games. Competitive two-player games are often modeled as either a table of payouts (*normal form*) or a game tree (*extensive form*). The extensive form model is equivalent to the *refereed games* model wherein the game is specified by a referee who exchanges messages with the players and declares a winner at the end of the interaction. In this terminology, normal form games correspond to the very restricted class of *one-turn* refereed games in which there is no communication from the referee to the players.

Despite this restriction, the problem of computing the exact value of a normal form game is logspace-hard for P [44, 41]. This hardness result is striking when juxtaposed with the existence of deterministic polynomial-time algorithms for arbitrary, multi-turn games [81, 80]. For succinct games where the game setting is specified implicitly by circuits, exponential-time algorithm exists for finding the exact value [80, 81] and it is also EXP-hard to approximate the game value [44, 41]. The situation is much different for shorter games, where succinct two-turn games admit polynomial-space approximation scheme and are also PSPACE-hard to approximate [40]. Approximating one-turn games is known to be S_2^P-complete [44].

For each competitive game model, one can analogously define the corresponding competing provers interactive proofs (also called *refereed games*), where players become competing provers who are trying to convince the polynomial-time bounded verifier to either accept or reject on the input x. Note this complexity class includes the standard *interactive proof systems* as special cases when the verifier simply ignores one of the provers. Let RG(k) denote the complexity class of problems that admit classical refereed games of k-turns and RG be short for RG(poly). Thus the above algorithmic results imply RG=EXP and RG(2)=PSPACE.

Quantum refereed games are defined similarly except that the referee is a polynomial time quantum computer who exchanges quantum messages with the provers. The class of problems that admit quantum refereed games is denoted QRG.

The polynomial-time algorithm for quantum games implies QRG$\subseteq$EXP [52]. Prior work on classical refereed games then implies QRG=RG=EXP that is the competing-prover analogy of the well-known collapse QIP=IP=PSPACE for single-prover interactive proofs [86, 100, 60, 113]. One can analogously define QRG(k) as the quantum counterpart of RG(k). The parallel algorithm for one-turn quantum games immediately implies QRG(1)$\subseteq$PSPACE [62].

For both classical and quantum refereed games, it is an interesting long standing open question as to whether there is a space-efficient algorithm for approximating k-turn games for some $k \geq 2$.

Our contributions: we completely answer the above question for $k = 2$ by showing that quantum two-turn refereed games coincide with polynomial space, i.e., QRG(2)=PSPACE. Our result is more than that. We define and study a much more general class called *double quantum interactive proof systems* and prove that its expressive power is also PSPACE. This class is so general that our result subsumes and unifies all previously known results about short classical or quantum refereed games.

Our results also lead to two main by-products. First, we observe the difference between public-coin and private-coin models in refereed games. In contrast to single-prover interactive proof systems, in which public-coin and private-coin models are roughly identical, we demonstrate that public-coin refereed games are strictly weaker than private-coin refereed games unless PSPACE=EXP. Second, part of our results can also be treated as parallel algorithms for a very general class of semidefinite programs.

1.2.4 Quantum Merlin-Arthur Games: Single vs Multiple Provers

In this section we discuss about the simplest quantum model in proof verification and its surprisingly interesting variant.

QMA and its complete problems: the complexity class QMA (Quantum Merlin-Arthur games, also known as *quantum proofs*) was defined [73] as the quantum counterpart of the complexity class NP. Discovered by Kitaev [73], the *local hamiltonian* problem is the first known QMA-complete problem, which naturally generalizes the Boolean Satisfiability problem and serves as the quantum analog of the Cook-Levin theorem [31, 82]. Subsequently, a series of parameter improvements have been made on the local Hamiltonian problem [65, 69, 59, 92, 3], one of which suggests that the problem remains QMA-complete even for 1-D local Hamiltonian. Other QMA-complete problems also exist, such as the local consistency problem and its variants [83, 84, 105], and the quantum clique problem [13].

We are interested in one QMA-complete problem called the *Non-Identity Check* problem that is closely connected to the implementation of quantum computation. Much of the difficulty in implementing quantum computation is due to the decoherence effect of quantum mechanics, which happens in very short time. Short-depth quantum circuits, as a result, seem to be an easier model to implement. Thus, analyzing the power of short-depth quantum circuits is of significant importance.

The Non-Identity Check problem is to decide whether or not a quantum circuit is far away from the identity circuit, given a classical description of the circuit. More generally, one can ask whether two quantum circuits U and V are equivalent or not. However, it is easy to see that the non-equivalence problem can be reduced to the non-identity check problem of the circuit $UV^\dagger$. Similar problems [19, 98, 117] that determine whether two given classical circuits are equivalent or not are known to admit efficient randomized al-

gorithms. In contrast, the quantum Non-Identity Check problem was shown to be QMA-complete [65], which suggests the hardness of distinguishing between quantum circuits.

The original version of the quantum Non-Identity Check problem [65] is about quantum circuits of polynomial sizes. However, all known realizable quantum circuits, which need to overcome the decoherence effect in some way, are circuits of short depths. Thus, it is reasonable and interesting to ask about the complexity of the quantum Non-Identity Check problem when restricting to short-depth circuits.

If the complexity is still high, say, remains QMA-complete, this might suggest even short-depth quantum circuits still preserve enough amount of "quantumness" in the sense that it is sufficient to generate QMA-hardness. On the other side, if the complexity is reduced, this might be a signal that short-depth quantum circuits reduce to classical circuits somehow and might not be useful to perform truly quantum tasks.

Our contributions: we determine the complexity of the Non-Identity Check problem for poly-logarithmic depth quantum circuits and show that it is still QMA-complete. Namely, even poly-logarithmic depth quantum circuits are still "quantum enough" to be hard to simulate.

QMA with multiple provers: the multiple-prover variant of QMA attracts much attention recently because it exhibits a counter-intuitive phenomenon that the enforcement of no entanglement could increase computational power. This is a quantum-unique phenomenon as its classical counterpart trivially reduces to the one-prover setting. The study of QMA(k), where k denotes the number of provers, was initialized by Kobabyashi *et al.* [78, 79]. Much attention was attracted to this model because of the discovery that NP admits *logarithmic*-size unentangled quantum proofs [18]. This result was surprising because single prover quantum logarithm-size proofs only characterize BQP [88]. Adding one unentangled prover seems to increase the power of the model substantially. The ini-

tial protocol [18] has been subsequently improved either with better completeness and soundness [12, 2, 27, 38] or with less powerful verifiers [26].

Despite many efforts, any non-trivial upper bound of QMA(2) remains elusive. The best known upper bound QMA(2)$\subseteq$NEXP follows trivially by nondeterministically guessing the two proofs. Although NP admits logarithmic-size unentangled quantum proofs, the protocol dose not scale to show NEXP is inside QMA(2). Indeed, it would be surprising if QMA(2)=NEXP. Nevertheless, several partial results that attack on related problems or simplified models have been obtained recently. In [55], Harrow et al. proved that QMA(2)=QMA(poly) by using the so-called *product test* protocol that determines whether a multipartite pure state is a product state when two copies of the multipartite state are given. Another line of research studies the power of unentangled quantum proofs with restricted verifiers. Two complexity classes BellQMA and LOCC-QMA, referring to the restricted verifiers that perform only nonadaptive or adaptive local measurements respectively, were defined in [2] and studied in [21, 22]. It has been shown [22] that LOCC-QMA(m) is equivalent to QMA for constant m.

It is a challenging problem to seek a better upper bound of QMA(2) itself or of a more powerful variant than LOCC-QMA or BellQMA.

Our contributions: we provide a PSPACE upper bound for a variant of QMA(2) that allows the verifier to perform restricted but entangled measurements over the two proofs. Note that neither LOCC-QMA nor BellQMA allows any entangled measurement. Our variant is thus the most general one considered up to date, and one step closer to QMA(2).

Our main observation is to cleverly enumerate over a structured solution space. As a result, we can minimize enumeration and replace it by efficient computation. Again, the final ingredient is our equilibrium value method, which leads to space-efficient solutions.

1.3 Organization

The rest of the dissertation is organized as follows. In Chapter 2, we introduce necessary backgrounds about quantum computation, computational complexity classes and semidefinite programs. In Chapter 3, we formulate our main technique contribution, the *equilibrium value method*, and deal with various issues about this framework. In Chapter 4, we apply the equilibrium value method to characterize the expressive power of quantum interactive proof systems. Then we proceed to the problems requiring more sophisticated uses of the equilibrium value method. In Chapter 5 and Chapter 6, we demonstrate how to simulate quantum refereed games and quantum Merlin-Arthur games of multiple provers in this way. In Chapter 7, we illustrate that the Non-Identity Check problem remains QMA-complete even for poly-logarithmic depth quantum circuits.

This dissertation is mainly devoted to our work on quantum computational complexity, especially on quantum interactive proof systems. We will briefly review the author's work on other topics during the Phd period and conclude this dissertation in Chapter 8.

CHAPTER 2

Preliminaries

This chapter serves as an introduction to those concepts, mathematical objects and our notation used throughout this dissertation. It is not meant to be comprehensive. We only summarize fundamental notions that are directly related to our topics here. Relatively more advanced notions shall be introduced right before their use.

In Section 2.1, we provide mathematical formulations of objects in quantum computation. In Section 2.2, we provide formal definitions of those classical and quantum computational complexity classes that we are interested in. Finally, in Section 2.3, we introduce semidefinite programs, another important mathematical object in our dissertation.

2.1 Mathematical Formulations of Quantum Computation

We refer readers who are unfamiliar with fundamental quantum computation and information concepts to [73, 91, 111]. The following is meant to clarify notation used through this dissertation.

Operators: 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.

- An operator $A \in \mathrm{L}(\mathcal{X})$ is *Hermitian* 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 in descending order.

- 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 $\Pi \in \mathrm{Pos}(\mathcal{X})$ is a *projection* if Π is a Hermitian and satisfies $\Pi^2 = \Pi$. Note 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 $A, B \in \mathrm{L}(\mathcal{X})$.

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

A *quantum state* of a quantum register $\mathcal{X}$ is represented by a *density operator* ρ, in which $\rho \in \mathrm{Pos}(\mathcal{X})$ and $\mathrm{Tr}(\rho) = 1$. The set of density operators is denoted by $\mathrm{D}(\mathcal{X})$. When ρ's rank is one, such a state is called a *pure* state; otherwise it is called a *mixed* state. A mixed density operator describes a quantum system in a statistical ensemble of several quantum states, in contrast to a pure state. The density operator is the quantum-mechanical analogue to a phase-space probability measure (probability distribution of position and momentum) in classical statistical mechanics.

Quantum Measurements: 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 $\langle P_a, \rho\rangle$.

Tensor-product Spaces: The tensor product $\mathcal{X} \otimes \mathcal{Y}$ of vector space $\mathcal{X} = \mathbb{C}^{\Sigma}$ and $\mathcal{Y} = \mathbb{C}^{\Gamma}$ is associated with the space $\mathbb{C}^{\Sigma\times\Gamma}$. The tensor product of operators $A \in \mathrm{L}\,(\mathcal{X})$ and $B \in \mathrm{L}\,(\mathcal{Y})$ is defined to be the unique linear mapping that satisfies $(A \otimes B)(x \otimes y) = (Ax) \otimes (By)$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.

For spaces $\mathcal{X}$ and $\mathcal{Y}$, one can define the *partial trace* $\mathrm{Tr}_{\mathcal{Y}} : \mathrm{L}\,(\mathcal{X} \otimes \mathcal{Y}) \to \mathrm{L}\,(\mathcal{X})$ to be the unique linear mapping that satisfies $\mathrm{Tr}_{\mathcal{Y}}(A \otimes B) = (\mathrm{Tr}\,A)B$ for all $A \in \mathrm{L}\,(\mathcal{X})$ and $B \in \mathrm{L}\,(\mathcal{Y})$.

Norms of Operators: Many interesting and useful norms can be defined on spaces of operators. We restrict our interest to a single family of norms called Schatten p-norms. This family includes the three most commonly used norms in quantum information theory: the *spectral norm*, the *Frobenius norm*, and the *trace norm*.

For any operator $A \in \mathrm{L}\,(\mathcal{X}, \mathcal{Y})$ and any real number $p \geq 1$, one defines the Schatten p-norms of A as

$$\|\,A\,\|_p = [\mathrm{Tr}((A^{\star}A)^{p/2})]^{1/p}.$$

Denoting by $\|\cdot\|$ the Euclidean norm, we define

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

which happens to coincide with $\lim_{p\to\infty}\|\,A\,\|_p$.

For a given non-zero operator $A \in \mathrm{L}(\mathcal{X}, \mathcal{Y})$ with rank $r \geq 1$, let the vector $s(A)$ denote the singular values of A. For each real number $p \in [1, \infty]$, it holds that the Schatten p-norm of A coincides with the ordinary (vector) p-norm of $s(A)$:

$$\|A\|_p = \|s(A)\|_p.$$

- The *spectral norm* of an operator $A \in \mathrm{L}(\mathcal{X})$, also called the *operator norm*, is defined by

  $$\|A\|_\infty = \max\{\|Ax\| : x \in \mathcal{X}, \|x\| \leq 1\}.$$

  It is easy to see that $\|A\|$ is actually the maximum singular value of A.

- The *Frobenius norm* of a matrix A is defined by

  $$\|A\|_{\mathrm{F}} = [\mathrm{Tr}(A^*A)]^{1/2} = \sqrt{\langle A, A\rangle}.$$

  It is therefore the norm defined by the Hilbert-Schmidt inner product on $\mathrm{L}(\mathcal{X}, \mathcal{Y})$, which leads to an alternative formulation:

  $$\|A\|_{\mathrm{F}} = \|\mathrm{vec}(A)\| = \sqrt{\sum_{i,j} |A(i,j)|^2},$$

  where i and j range over the indices of the matrix representation of A.

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

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

  When A is Hermitian, we have

  $$\|A\|_{\mathrm{Tr}} = \max\{\langle P_0 - P_1, A\rangle : P_0, P_1 \in \mathrm{Pos}(\mathcal{X}), P_0 + P_1 = \mathbb{1}_{\mathcal{X}}\}. \tag{2.1.1}$$

Distance Metrics: Two distance measures between quantum states are commonly used, i.e., the *trace distance* and the *fidelity*.

- Given any two quantum states $\rho, \sigma \in \mathrm{D}(\mathcal{X})$, the *trace distance* between ρ and σ is simply,

$$\|\rho - \sigma\|_{\mathrm{Tr}},$$

which ranges between 0 and 2.

- Given any two quantum states $\rho, \sigma \in \mathrm{D}(\mathcal{X})$, the *fidelity* between ρ and σ is

$$\mathrm{F}(\rho, \sigma) = \left\| \sqrt{\rho}\sqrt{\sigma} \right\|_{\mathrm{Tr}},$$

which ranges between 1 and 0.

There is an important relation between the above two measures, known as Fuchs-van de Graaf inequality.

Lemma 2.1.1 (Fuchs-van de Graaf)**.** *For any $\rho, \sigma \in \mathrm{D}(\mathcal{X})$, we have*

$$1 - \frac{1}{2}\|\rho - \sigma\|_{\mathrm{Tr}} \leq \mathrm{F}(\rho, \sigma) \leq \sqrt{1 - \frac{1}{4}\|\rho - \sigma\|_{\mathrm{Tr}}^2}. \tag{2.1.2}$$

Super-operators: 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 *admissible* if it is completely positive and trace-preserving. Admissible super-operators represent the discrete-time changes in quantum systems that, in principle, can be physically realized.

One can also define the *adjoint* super-operator of Ψ, denoted by

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

to be the unique linear mapping that satisfies

$$\langle B, \Psi(A) \rangle = \langle \Psi^*(B), A \rangle ,$$

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

Let $\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ be any given super-operator, the *Choi-Jamiolkowski representation* of Ψ is denoted

$$J(\Psi) = \sum_{i,j \in \Sigma} \Psi(|i\rangle \langle j|) \otimes |i\rangle \langle j| \in \mathrm{L}(\mathcal{Y} \otimes \mathcal{X}) .$$

For an admissible super-operator Ψ, the $J(\Psi)$ is positive semidefinite and $\mathrm{Tr}_{\mathcal{Y}} J(\Psi) = \mathbb{1}_{\mathcal{X}}$.

The *Stinespring representations* of super-operators goes as follows. For any super-operator Ψ, there is some auxiliary space $\mathcal{Z}$ and $A, B \in \mathrm{L}(\mathcal{X}, \mathcal{Y} \otimes \mathcal{Z})$ such that

$$\Psi(X) = \mathrm{Tr}_{\mathcal{Z}} AXB^*$$

for all $X \in \mathrm{L}(\mathcal{X})$. When Ψ is admissible, we have $A = B$ and A is a linear isometry.

The *diamond norm* of a super-operator $\Psi : \mathrm{L}(\mathcal{X}) \to \mathrm{L}(\mathcal{Y})$ is defined to be

$$\| \Psi \|_{\diamond} = \max_{\|X\|_1 \leq 1} \| \Psi \otimes \mathbb{1}_{\mathcal{X}}(X) \|_{\mathrm{Tr}}.$$

Because of taking into account the effort of using auxiliary entanglement , the diamond norm serves as a good measure of the distinguishability between quantum operations.

2.2 Computational Complexity Classes: Classical and Quantum

We assume fundamental knowledge of the deterministic Turing machine and its non-deterministic and probabilistic variant. In the following, we have briefly summarized those basic complexity classes that appear in this dissertation and then provide formal definitions of interactive proof systems and their variants, both classical and quantum.

2.2.1 Basic Classical Complexity Classes

The following complexity classes are directly related to this dissertation.

- P : stands for the collection of promise problems $A = (A_{\text{yes}}; A_{\text{no}})$ such that for every A, there exists a *polynomial-time deterministic Turing machine* that accepts if $x \in A_{\text{yes}}$ and rejects if $x \in A_{\text{no}}$.
- BPP : stands for the collection of promise problems $A = (A_{\text{yes}}; A_{\text{no}})$ such that for every A, there exists a *polynomial-time probabilistic Turing machine* that accepts with probability at least $2/3$ if $x \in A_{\text{yes}}$ and rejects with probability at least $2/3$ if $x \in A_{\text{no}}$.
- NP : stands for the collection of promise problems $A = (A_{\text{yes}}; A_{\text{no}})$ such that for every A, there exists a *polynomial-time nondeterministic Turing machine* that accepts if $x \in A_{\text{yes}}$ and rejects if $x \in A_{\text{no}}$. Alternatively, the class NP can be defined as follows. For every A, there exists a polynomial-time deterministic Turing machine that takes the input x and some witness string w whose length $|w| = \text{poly}(|x|)$. So that if $x \in A_{\text{yes}}$, then there exists a w to make the machine accept; otherwise, if $x \in A_{\text{no}}$, then the machine rejects no matter what the witness string w is.
- PSPACE : stands for the collection of promise problems $A = (A_{\text{yes}}; A_{\text{no}})$ for which there exists a *polynomial-space deterministic Turing machine* that accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.
- EXP : stands for the collection of promise problems $A = (A_{\text{yes}}; A_{\text{no}})$ such that there exists an *exponential-time deterministic Turing machine* accepts every string $x \in A_{\text{yes}}$ and rejects every string $x \in A_{\text{no}}$.
- NEXP : stands for the collection of promise problems $A = (A_{\text{yes}}; A_{\text{no}})$ such that for every A, there exists a *exponential-time nondeterministic Turing machine* that

accepts if $x \in A_{\text{yes}}$ and rejects if $x \in A_{\text{no}}$. Similar to NP, this class has an alternative definition using the notion of witness strings except that the length now becomes exponential.

These above complexity classes are related to each other via the following relationships:

$$\text{P} \subseteq \text{NP} \subseteq \text{PSPACE} \subseteq \text{EXP} \subseteq \text{NEXP and P} \subseteq \text{BPP}.$$

2.2.2 Interactive Proof Systems

In the introduction we briefly survey interactive proof systems. Here we formally define those complexity classes. The introduction already includes the definition of interactive proof systems. We will focus on definitions of quantum interactive proof systems, quantum Merlin-Arthur games, and classical and quantum refereed games.

Definition 2.2.1 (QIP). Let $L = (L_{\text{yes}}, L_{\text{no}})$ be a promise problem, let m be a polynomial-bounded function, and let $a, b : \mathbb{N} \to [0,1]$ be polynomial-time computable functions. Then $L \in \text{QIP}(m,a,b)$ if and only if there exists an m-message quantum verifier V with the following properties:

1. *Completeness.* For all $x \in L_{\text{yes}}$, there exists a quantum prover P that causes V to accept x with probability at least $a(|x|)$.
2. *Soundness.* For all $x \in L_{\text{no}}$, every quantum prover P causes V to accept x with probability at most $b(|x|)$.

Also define $\text{QIP}(m) = \text{QIP}(m, 2/3, 1/3)$ for each polynomial-bounded function m and define $\text{QIP} = \bigcup_m \text{QIP}(m)$, where the union is over all polynomial-bounded functions m.

As a special case of quantum interactive proof systems, in which there is only one-way communication from the prover to the verifier, one-turn quantum interactive proof systems (i.e., QIP(1)) have another name called *quantum Merlin-Arthur games* (QMA).

It is considered as the quantum counterpart of the complexity class NP. Due to inherent randomness in quantum computation, it would be more accurate to consider it the quantum counterpart of the complexity class MA, which stands for Merlin-Arthur games and is the randomized version of NP.

Definition 2.2.2 (QMA)**.** A language L is in QMA if there is a family of circuits $\{U_x, x \in \Sigma^*\}$ generated in polynomial-time together with a polynomial m such that U_x acts on $m+k$ qubits and the following holds:

1. If $x \in L$, there exists an $m(|x|)$-qubit state $|\psi\rangle$, $\Pr\left[U_x \text{ accepts } |\psi\rangle \otimes |0\rangle^{\otimes k(|x|)}\right] \geq 2/3$;
2. If $x \notin L$, for all $m(|x|)$-qubit state $|\psi\rangle$, $\Pr\left[U_x \text{ accepts } |\psi\rangle \otimes |0\rangle^{\otimes k(|x|)}\right] \leq 1/3$.

We shall encounter multiple-prover quantum Merlin-Arthur games in Chapter 6 in which we will provide the precise definition for the multiple-prover version. In the following, we formally define the classical refereed games (RG) and its quantum variant (QRG).

Definition 2.2.3 (**RG**)**.** A promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ is in RG if and only if it has a classical interactive proof system with two competing provers. The completeness and soundness conditions for such a proof system are replaced by the following conditions:

1. For every $x \in L_{\text{yes}}$, there exists a yes-prover P_{yes} that convinces the referee to *accept* with probability at least 2/3, regardless of the strategy employed by the no-prover P_{no}.
2. For every $x \in L_{\text{no}}$, there exists a no-prover P_{no} that convinces the referee to *reject* with probability at least 2/3, regardless of the strategy employed by the yes-prover P_{yes}.

Quantum refereed games are defined similarly except that the referee is a polynomial time quantum computer who exchanges quantum messages with the provers. The class of

problems that admit quantum refereed games is denoted QRG.

Definition 2.2.4 (QRG). A promise problem $L = (L_{\text{yes}}, L_{\text{no}})$ is in QRG (*quantum refereed games*) if and only if it has a quantum interactive proof system with two competing provers. The completeness and soundness conditions for such a proof system are analogous to RG.

2.3 Semidefinite Programs

A *semidefinite program* (SDP) over $\mathcal{X}$ and $\mathcal{Y}$ (shown below) is specified by a triple (Ψ, A, B) where $\Psi : \mathrm{L}(\mathcal{X}) \rightarrow \mathrm{L}(\mathcal{Y})$ is a Hermiticity preserving super-operator and $A \in \mathrm{Herm}(\mathcal{X})$ and $B \in \mathrm{Herm}(\mathcal{Y})$.

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

The reason why SDP is intimately related to quantum information and computation is that quantum states are represented by semidefinite positive operators with unit trace. Therefore, if one considers any optimization problem involving quantum states, SDP should be the most nature tool to formulate the problem.

In practice we usually consider SDPs whose optimum values are within a constant range. For those SDPs, it suffices to consider the *feasibility problem* defined below. Once the feasibility problem is solved, it suffices to use binary search to find the optimum value. For any instance of SDP and a guess value c , the feasibility problem is defined to be

Feasibility Problem

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

CHAPTER 3

Equilibrium Value Method

In this chapter, we formulate our main technical contribution in this dissertation, namely the so-called *equilibrium value method*, for obtaining PSPACE solutions to a class of quantum optimization problems. We illustrate our approach of reformulating these optimization problems as zero-sum games and highlight the main two technical difficulties in doing so: the first one is that one can only hope to approximate equilibrium values of zero-sum games rather than to solve them exactly, and the second one is that we need an extra rounding theorem to convert approximate solutions to exact solutions without incurring too much error. We discuss how the matrix multiplicative weight update method applies in our method and point out potential technical limits.

The rest of this chapter is organized as follows. In Section 3.1, we formulate the equilibrium value method. In Section 3.2, we introduce the matrix multiplicative weight update method and show how it approximates a specific form of equilibrium values of zero-sum games. In Section 3.3, we demonstrate a warm-up example in which our equilibrium value method out-performs known approaches. In Section 3.4, we compare our use of the matrix multiplicative weight update method with Arora-Kale's primal-dual approach. Finally, in Section 3.5, we address how our solutions can be made space-efficient and also deal with their possible precision issues.

3.1 Formulation

This equilibrium value method first converts the optimization problem into a zero-sum game, and then applies the matrix multiplicative weight update method to approximate the equilibrium value of that particular zero-sum game. Note that in such conversions, a small error in approximating the equilibrium value might blow up significantly for the original optimization problem. Among all possible conversions of this kind, our conversion is optimal in the sense that it guarantees the efficiency in implementing this conversion, and also minimizes the error induced by the conversion.

Let us demonstrate this framework more carefully. Recall that for any instance of semidefinite programs and a guess value c , the feasibility problem[1] is defined to be

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

The equilibrium value method converts any feasibility problem into a zero-sum game as follows. Imagine that there is a primal player who wants to provide you a feasible solution X to prove that 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. Thus, it is conceivable that the ability to determine the winner of this zero-sum game is equivalent to the ability to determine the feasibility of the optimization problem.

However, this intuition cannot be carried out directly due to two technical constraints.

- First, we cannot calculate equilibrium values, or who wins the game, perfectly. All known techniques only approximate equilibrium values. Moreover, for our purpose,

[1] We replace $\mathrm{Pos}(\mathcal{X})$ by $\mathrm{D}(\mathcal{X})$ due to technical reasons. However, imposing $\mathrm{D}(\mathcal{X})$ constraint (i.e., bounding the width of X) is valid in many applications.

we need to efficiently approximate equilibrium values of zero-sums, not only in time but also in space. This is a further requirement that we do not have much prior knowledge.

- Second, we need an extra rounding theorem to convert approximate solutions to exact solutions without incurring too much error. Such a rounding theorem will depend on specific cases.

The main challenge is to address these two issues at the same time. For technical reasons, we might want to approximate equilibrium values of some relaxations of original zero-sum games, which might in turn require special handling in the rounding theorem.

We define the following bilinear function f to capture the zero-sum game mentioned above.

Framework 3.1.1. *Let function f be*

$$f(X,\Pi) = \left\langle \begin{pmatrix} c - \langle A, X\rangle & \\ & \Psi(X) - B \end{pmatrix}, \Pi \right\rangle \tag{3.1.1}$$

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 $\check{\lambda}(f)$ be

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

The relation between the equilibrium value $\check{\lambda}(f)$ and the feasibility of the original problem is captured by the following theorem.

Theorem 3.1.2. *The original problem is feasible if and only if $\check{\lambda}(f) \leq 0$.*

We remark that Framework 3.1.1 is flexible in the sense that it admits manipulations on the specific form of the function f. For example, one could assign different weights to different constraints; or one can split the variable X into two density operators X_1, X_2 and

1. Initialization: Pick a fixed $\varepsilon \leq \frac{1}{2}$, and let $W^{(1)} = \mathbb{1}_{\mathcal{X}} \in \mathrm{L}(\mathcal{X})$, $N = \dim \mathcal{X}$.
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 $0 \leq M^{(t)} \leq \mathbb{1}_{\mathcal{X}}$, update the weight matrix as follows:
$$W^{(t+1)} = \exp(-\varepsilon \sum_{\tau=1}^{t} M^{(\tau)}).$$

Figure 3.1: The Matrix Multiplicative Weight Update method.

so on. What appears in Eq. (3.1.1) only conveys the spirit of this framework and can be manipulated for different problems.

In the next section, we will introduce the matrix multiplicative weight update method which leads to algorithms that approximate $\check{\lambda}(f)$ within satisfiable precision under mild technical conditions. Moreover, because the matrix multiplicative weight update method only contains fundamental operations of matrices, the resultant algorithms are usually also efficient in parallel (i.e., in NC). By invoking the well-known relation between parallel efficiency and space efficiency (i.e., NC(poly)=PSPACE [20]), we could also obtain space-efficient solutions. This is vital for our later purpose, namely to prove PSPACE upper bounds of quantum computational complexity classes.

3.2 Matrix Multiplicative Weight Update Method

Fortunately, for many interesting instances, our equilibrium value method can be applied with the help of the matrix multiplicative weight update method. This method is a well-known framework that originates in various fields, such as machine learning and operations research. This particular variant (shown in Fig. 3.1) in our application was developed and discussed in a survey paper [6] and also in the PhD thesis of Kale [67]. Note that $\{M^{(t)}\}$ is the freedom we have in this framework.

Theorem 3.2.1. *Assume* $0 \leq M^{(t)} \leq \mathbb{1}$ *for all t, after T rounds, the algorithm in Fig 3.1*

guarantees that, for any $\rho^* \in \mathrm{D}(\mathcal{X})$, we have

$$(1-\epsilon)\sum_{t=1}^{T}\left\langle \rho^{(t)}, M^{(t)}\right\rangle \leq \left\langle \rho^*, \sum_{t=1}^{T} M^{(t)}\right\rangle + \frac{\ln N}{\epsilon}. \tag{3.2.1}$$

One important and direct application of this method is to efficiently approximate a class of equilibrium values. One of the main advantages of this method is that any time efficiency in this framework can be easily extended to space efficiency. This is because the matrix multiplicative weight update method is explicit and only contains fundamental operations of matrices, which usually admit space-efficient solutions [108]. We defer discussions for space-efficiency to Section 3.5.

Let us consider the following class of equilibrium values. Given any convex matrix set $\mathbf{P}$ of bounded width (say, $\exists\, w > 0$ s.t. $\forall P \in \mathbf{P}, \|P\|_\infty \leq w$), and any *explicit*[2] Hermiticity preserving super-operator Ξ, define the equilibrium value $\lambda(\mathbf{P}, \Xi)$ by

$$\lambda(\mathbf{P}, \Xi) = \min_{X\geq 0, \mathrm{Tr}(X)=1} \max_{P\in\mathbf{P}} \langle \Xi(X), P\rangle = \max_{P\in\mathbf{P}} \min_{X\geq 0, \mathrm{Tr}(X)=1} \langle \Xi(X), P\rangle . \tag{3.2.2}$$

A direct consequence following from [6] is:

Corollary 3.2.2. *If the optimization problem $\max_{P\in\mathbf{P}} \langle \Xi(X^*), P\rangle$ for any X^* can be efficiently approximated within an additive error δ in space, then there is a space-efficient algorithm to approximate λ with an inverse poly-logarithmic additive error plus δ.*

3.3 Example: QIP(2)⊆PSPACE

Now it is our turn to consider a real instance of semidefinite programs and apply our framework to solve it. Our candidate is the quantum interactive proof systems with two messages. In this model, on input x, 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

[2] For example, if the super-operator can be calculated by a constant number of steps of fundamental matrix operations.

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 admissible 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.

It is known that QIP(2)$\subseteq$PSPACE [61] by following Arora-Kale's way [67] to solve SDPs. Here we demonstrate how our **Framework 3.1.1** can be applied, thus contributing an alternative and simple proof of the same result. There are two main differences between the two proofs. The first one is that we choose a better formulation of this model in terms of density operators rather than quantum channels. The second one is that we apply our equilibrium value method rather than Arora-Kale's approach to solve the resultant semidefinite programs.

Let $\mathcal{M}$ denote the message's space between the prover and the verifier and $\mathcal{V}$ denote the verifier's private space. Fix input x for the following discussion. Without loss of generality, let the pure state $\rho_1 \in \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$ be the initial state. The prover then applies an admissible quantum channel $\Phi : \mathrm{L}(\mathcal{M}) \to \mathrm{L}(\mathcal{M})$ on part of the state ρ_1 and obtains state $\rho_2 = \Phi \otimes \mathbb{1}_{\mathrm{L}(\mathcal{V})}(\rho_1)$. The verifier then measures on ρ_2 to decide whether to accept or to reject. Let R be the POVM corresponding to the acceptance case. To simulate this model, it suffices to solve the following 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 on input x.

By Lemma 3.3.3, 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 semidefinite program,

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:	$\mathrm{Tr}_{\mathcal{M}}(\rho_2) \leq \mathrm{Tr}_{\mathcal{M}}(\rho_1)$,
	$\rho_2 \in \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$.		$\rho_2 \in \mathrm{D}(\mathcal{M} \otimes \mathcal{V})$.

Before we proceed to the solution to the above feasibility problem, we present several useful properties about purification and fidelity as follows.

Lemma 3.3.1. *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 $\mathcal{F}(\rho_1, \rho_2) = \mathcal{F}(\sigma_1, \sigma_2)$.*

(Please note that this lemma was originally proved in many places. The following proof follows the one in [61]. The only reason to include this proof is for later use in proving Lemma 3.3.2.)

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 $\mathcal{F}(\rho_1, \rho_2) = \|\sqrt{\rho_1}\sqrt{\rho_2}\|_{\mathrm{Tr}}$, 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 $|u_2\rangle = \mathrm{vec}(\sqrt{\rho_2} V U^*) \in \mathcal{A} \otimes \mathcal{B} \otimes \mathcal{C}$ 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 holds that

$$\begin{aligned} \mathrm{F}(\sigma_1, \sigma_2) &\geq |\langle \mathrm{vec}(\sqrt{\rho_1} U^*), \mathrm{vec}(\sqrt{\rho_2} V U^*) \rangle| = |\langle \sqrt{\rho_1} U^*, \sqrt{\rho_2} V U^* \rangle| \\ &= \mathrm{Tr}(\sqrt{\rho_1}\sqrt{\rho_2} V) = \mathrm{F}(\rho_1, \rho_2). \end{aligned}$$

□

Lemma 3.3.2. *In addition to the result in Lemma 3.3.1, 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.*

Proof. The proof of Lemma 3.3.1 actually gives one a way to construct 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 $S = \mathrm{span}\{|\psi\rangle : \langle\psi| \sqrt{\rho_1}\sqrt{\rho_2} |\psi\rangle \leq 0\}$.

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

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

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

Lemma 3.3.3. *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 admissible 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 $\mathrm{Tr}_{\mathcal{A}}(\rho_1) = \mathrm{Tr}_{\mathcal{A}}(\rho_2)$.*

Lemma 3.3.4. *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 $\mathrm{F}(\rho_1, \rho_2) = \mathrm{F}(\sigma_1, \sigma_2)$, let*

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

Then we have the following inequalities

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

Proof. This is only a simple application of Fuchs-van de Graaf Inequality. Namely, we have

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

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

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

□

Solution to the feasibility problem

Following Framework 3.1.1, we define

$$f_1(\rho, \Pi) = \left\langle \begin{pmatrix} c - \langle R, \rho \rangle & \\ & \mathrm{Tr}_{\mathcal{M}}(\rho) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \end{pmatrix}, \Pi \right\rangle, \tag{3.3.1}$$

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 $\check{\lambda_1}$ be the equilibrium value of function f_1, namely,

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

By Theorem 3.1.2, the value of $\check{\lambda_1}$ determines whether the original problem is feasible. In addition, we demonstrate a rounding theorem that converts any approximately feasible solution to exactly feasible solution below.

Lemma 3.3.5. *Let $\bar{\lambda_1}$ be the approximate equilibrium value of the function f_1 to precision δ, and $\check{\lambda_1}$ 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, $\check{\lambda_1} \geq \bar{\lambda_1} - \delta > 0$, then due to Theorem 3.1.2, the original problem is feasible.
- Otherwise, we have

$$\max\{c - \langle R, \bar{\rho} \rangle, 0\} + \frac{1}{2} \| \mathrm{Tr}_{\mathcal{M}}(\bar{\rho}) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \|_{\mathrm{Tr}} \leq \delta, \tag{3.3.2}$$

where $\bar{\rho}$ is an approximate equilibrium point. By Lemma 3.3.1 and 3.3.2, we can compute $\tilde{\rho}$ such that $\mathrm{F}(\mathrm{Tr}_{\mathcal{M}}(\bar{\rho}), \mathrm{Tr}_{\mathcal{M}}(\rho_1)) = \mathrm{F}(\tilde{\rho}, \rho_1)$ and $\mathrm{Tr}_{\mathcal{M}}(\tilde{\rho}) = \mathrm{Tr}_{\mathcal{M}}(\rho_1)$. Let $s = \frac{1}{2} \| \mathrm{Tr}_{\mathcal{M}}(\bar{\rho}) - \mathrm{Tr}_{\mathcal{M}}(\rho_1) \|_{\mathrm{Tr}}$ and $t = \frac{1}{2} \| \tilde{\rho} - \bar{\rho} \|_{\mathrm{Tr}}$. Then we have

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

where the first inequality is due to Eq. (3.3.2) and Eq. (2.1.1) and the second inequality comes from Lemma 3.3.4. The last inequality is because $s - \sqrt{2s - s^2}$ is decreasing when $0 \leq s \leq 0.2$ and by Eq. (3.3.2) $s \leq \delta$.

□

To complete our algorithm, it suffices to notice that our $\check{\lambda}_1$ is a special form of $\lambda(\mathbf{P}, \Xi)$ in Eq. (3.2.2) where $\mathbf{P} = \{\Pi : 0 \leq \Pi \leq \mathbb{1}\}$ and Ξ is defined in Eq. (3.3.1). Furthermore, we notice that the optimization problem (i.e., $\max_{P \in \mathbf{P}} \langle \Xi(X^*), P \rangle$) required in Corollary 3.2.2 reduces to the projection onto that positive eignspace of $\Xi(X^*)$, which follows directly from the spectral decomposition of $\Xi(X^*)$. Therefore all conditions of Corollary 3.2.2 are satisfied and we could make use of it to approximate $\check{\lambda}_1$ to desired precision.

We are ready to apply our result to simulate QIP(2). Recall the definition of QIP(2), there will be two promises with gap $\Delta = c(|x|) - s(|x|) = \Omega(1/poly(|x|))$. It suffices to choose the right precision and then apply our above result.

Corollary 3.3.6. 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 Corollary 3.2.2 to calculate the equilibrium value $\check{\lambda}_1$ to precision δ.
- Finally, based on the two cases in Lemma 3.3.5, 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 $x \in L$.

All the circuits can be composed in NC(poly). Because NC(poly)=PSPACE [20], we have QIP(2)$\subseteq$PSPACE. □

3.4 Comparison with Arora-Kale's Approach

It is interesting to compare our equilibrium value method with the *primal-dual* method introduced by Arora-Kale [6, 7] to solve semidefinite programs. The main advantages of our method are a better guarantee of rounding theorems (i.e., incur the minimum amount of extra error) and a simpler oracle design (i.e., a simplification of one technical step required by the matrix multiplicative weight update method).

First of all, these two methods share a lot of similarities in making use of the matrix multiplicative weight update method. They both generate a series of candidate solutions $X^{(1)}, X^{(2)}, \cdots, X^{(T)}$ for T rounds using the matrix multiplicative weight update method. For each round, special designed operators serve as the loss matrices in the framework. To obtain these loss matrices, they both require to finish some extra computation, which we abstract as *oracles*. In general, the design of an efficient oracle depends on the specific problem and might need to handle specific technical difficulties. After T rounds, both methods obtain an approximate solutions for the original problem, however, under different guarantees of approximation. A crucial step afterwards is to prove such an approximate solution could be converted to an exact solution without incurring too much extra error, namely, a *rounding* theorem. It is conceivable that the quality of any rounding theorem shall highly depend on the approximation guarantee achieved.

We now briefly compare the performance of these two methods with respect to the two considerations discussed above.

- **Oracles**. In the primal-dual approach, for any feasibility problem of semidefinite programs, and for any $X^{(t)}$ in the round t, we require an oracle $\mathcal{O}_1$ to solve the following problem

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

The oracle $\mathcal{O}_1$ will return such a $Y^{(t)}$ or claim such a $Y^{(t)}$ does not exist. Note that this oracle is a semidefinite program in a restricted form. However, the problem is still so general that it is not known to admit an efficient solution automatically. In fact, a significant effort in using the primal-dual approach in previous works is to design an efficient oracle $\mathcal{O}_1$.

To approximate equilibrium values, we need a different oracle $\mathcal{O}_2$. For any equilibrium value λ in the form:

$$\lambda = \min_{X \geq 0, \mathrm{Tr}\, X = 1} \max_{y \in Y} f(x, y) = \max_{y \in Y} \min_{X \geq 0, \mathrm{Tr}\, X = 1} f(x, y),$$

oracle $\mathcal{O}_2$ only needs to solve the following optimization problem:

$$\max_{y \in Y} f(X^{(t)}, y).$$

In Framework 3.1.1, we choose $Y = \{\Pi : 0 \leq \Pi \leq \mathbb{1}\}$. This implies oracle $\mathcal{O}_2$ is equivalent to projection onto the positive eigenspace, which is a standard problem and known to admit space-efficient solutions. Namely, oracle design in our equilibrium value method is no longer an issue.

- **Approximation guarantees**. For the same additive error ϵ governed by the matrix multiplicative weight update method, the approximation guarantees for these two methods are different. For the primal-dual method, it means,

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

On the other side, our Framework 3.1.1 guarantees that

$$\forall \quad \Pi \in \{\Pi : 0 \leq \Pi \leq \mathbb{1}\}, \left\langle \begin{pmatrix} c - \langle A, X \rangle & \\ & \Psi(X) - B \end{pmatrix}, \Pi \right\rangle \leq \epsilon. \tag{3.4.3}$$

In examples [61, 60] using the primal-dual approach, the fact $A = \mathbb{1}_{\mathcal{X}}$ is crucial because of the guarantee in Eq. (3.4.2). Intuitively, this is because Eq. (3.4.2) roughly implies that $\Psi^*(Y) - A \gtrapprox 0$ only holds in terms of ℓ_∞ norm. On the other side, our approximation guarantee (Eq. (3.4.3)) is stronger in the sense that the candidate solution X only violates the constraints a bit so that the ℓ_1 norm of the violations is tiny, i.e., at most ϵ. This is one of the main reasons why we can handle sophisticated constraints later in Chapter 4 and Chapter 5. In addition, we have the freedom to choose a different T to generate a different Eq. (3.4.3), and thus to meet the requirement of new constraints, which is another advantage over the primal-dual approach.

3.5 NC and Precision Issues

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 of polynomial depths. Since it holds that NC(poly)= PSPACE [20], thus in order to simulate any algorithm 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. First, 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 [108] which de-

scribes 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.

Fact 3.5.1. Fundamental operations like addition, multiplication of matrices are in NC [108].

Fact 3.5.2. Matrix exponentials: there exist 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 3.5.3. Singular value decompositions: there exist 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

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

Precision Issues. The analysis made in the main part has assumed that all computations performed by the algorithm are exact. However, in order to implement our algorithm, some steps of the computations, such as positive eigenspace projections and matrix exponentials, must be approximate. An elaborated analysis on these issues can be found in [60, 62] for specific problems. We will basically follow that type of analysis and provide a sketch of the analysis to our problem.

First, it must be made clear which part of the algorithm can be made exact and which part must be made approximate. We will use the same convention of storing complex numbers as the one in [60]. Once the input x is given and stored in memory, all elementary

matrix operations (in this case: addition, multiplication, and computation of traces or partial traces) can be implemented exactly in NC [108]. However, matrix exponentials and positive eigenspace projections cannot be exact since these operations will generate irrational numbers and the precision must be truncated somewhere. Fortunately, Watrous et al. [60] provided a way to approximate these two operations to high precision in NC.

Fact 3.5.4. Given an $n \times n$ matrix M (whose operator norm is bounded by k) and a positive rational number η, the computation of $n \times n$ matrix X such that $\|exp(M) - X\| < \eta$ can be done in NC.

Fact 3.5.5. Given an $n \times n$ Hermitian matrix H and a positive rational number η, the computation of an $n \times n$ positive semidefinite matrix $\Delta \leq \mathbb{1}$ such that $\|\Delta - \Lambda\| < \eta$ for Λ being the projection operator onto the positive eigenspace of H can be done in NC.

Before we move on to the analysis of precision issues, it helps to introduce the following convention. We will represent the actual matrices generated during the algorithm by placing a tilde over the variables that represent idealized values. As we discussed above, there are mainly two types of operations where the accuracy will be lost. Further investigation tells us that matrix exponentials are always necessary to the multiplicative weight update method while positive eigenspace projections are special for our application. For generality of the analysis, we will first discuss the general form of Theorem 3.2.1 when the computation is only approximate.

Consider the scheme in Fig 3.1 and keep the notation convention in mind. The $\tilde{\rho}^{(t)}$ will be the actual generated density operator for round t and $W^{(t+1)} = \exp(-\epsilon \sum_{\tau=1}^{t} \tilde{M}^{(t)})$. The latter one is exact simply because $W^{(t)}$ is only a notation and not stored in the memory at all. Fact 3.5.4 implies that $\|\tilde{\rho}^{(t)} - W^{(t)} / \operatorname{Tr} W^{(t)}\| < \delta_1 / N$ for every t where δ_1 is some constant for our purpose. The situation for $\tilde{M}^{(t)}$ is tricky in the sense that there is no

idealized value for $M^{(t)}$ in general. By going through the proof of Theorem 3.2.1 again, we can easily obtain the following fact.

Fact 3.5.6. If the computation can only be performed approximately, the inequality in Theorem 3.2.1 becomes

$$(1-\epsilon)\sum_{t=1}^{T}\left\langle \tilde{\rho}^{(t)}, \tilde{M}^{(t)} \right\rangle \leq \left\langle \rho^{*}, \sum_{t=1}^{T}\tilde{M}^{(t)} \right\rangle + \frac{\ln N}{\epsilon} + \frac{1}{2}T\delta_1.$$

Thus, approximate scenario only incurs an affordable additive error. A similar analysis also applies to approximate eigenspace projections, which we leave to readers as a simple exercise.

CHAPTER 4

Quantum Interactive Proof Systems

This chapter is based on work [113, 114].

In this chapter, we provide alternative proofs of Jain et al.'s result of QIP=PSPACE [60]. The containment of PSPACE inside QIP follows automatically from the well-known relation IP=PSPACE. Thus, our contribution is to upper bound QIP by PSPACE.

In the following we provide two different proofs of QIP=PSPACE. The first one takes a totally different approach from Jain et al.'s. It starts with a QIP-complete problem, known as *close images*, and converts this problem into an equilibrium value problem in which Corollary 3.2.2 readily applies. This conversion only requires fundamental manipulations in quantum computation, and therefore leads to a much simplified proof of QIP=PSPACE. The second one is another demonstration of our standard equilibrium value method to solve semidefinite programs. We follow Jain et al.'s approach and try to solve the semidefinite programs corresponding to the complexity class QMAM. We depart from Jain et al.'s approach by using our own equilibrium value method rather than Arora-Kale's method to solve that particular semidefinite program. We note that this approach does not simplify the original proof much. However, it is meant to illustrate the generality of our equilibrium value method.

Before we proceed to the main technical part, let us clarify some pre and post processing procedures. First, it is clear that, given input x, one can easily calculate (through

fundamental matrix operations) an explicit representation of quantum circuits or states by definition. It is worth mentioning that representations of quantum circuits or states are of exponential sizes in terms of $|x|$. This step shall correspond to the calculation of the instance of *close images* problem and the instance of SDPs that correspond to QMAM. Second, for the purpose of distinguishing between yes and no instances, it suffices to approximate the corresponding problem to inverse polynomial precision in terms of $|x|$, which is inverse poly-logarithm in terms of the size of representations of quantum states. We thus claim that Corollary 3.2.2 is sufficient for our purpose in this step. As a routine in our framework, one can finally compose all these circuits in NC(poly) in terms of $|x|$ and then invoke the relation NC(poly)=PSPACE [20] to show the PSPACE upper bound.

In the rest of this chapter, we demonstrate our first approach in Section 4.1, followed by our second approach in Section 4.2. Finally, as a corollary of the whole chapter, we have

$$\text{QIP} = \text{PSPACE}.$$

4.1 Close Images Approach

To prove the containment of a certain complexity class inside another, it suffices to prove such containment of the complete problems of that complexity class. We thus start with one QIP-complete problem and prove its containment inside PSPACE. The particular complete problem we exploit is the *close images* problem defined below.

Definition 4.1.1 (Close Images, QIP-complete [74])**.** Given any two constants $a, b \in [0,1]$ with $b < a$, and two mixed quantum circuits (Q_0, Q_1), we want to distinguish between the following two promises.

- **Yes:** There exist quantum states ρ_0 and ρ_1 such that $\mathrm{F}(Q_0(\rho_0), Q_1(\rho_1)) \geq a$.
- **No:** For every choice of quantum states ρ_0 and ρ_1, $\mathrm{F}(Q_0(\rho_0), Q_1(\rho_1)) \leq b$.

By standard manipulations in quantum computation, we can convert the above problem into an equilibrium value problem as follows.

Theorem 4.1.2. *There exists an explicit Hermiticity preserving super-operator Ξ (which can be efficiently calculated from Q_0, Q_1) and a convex set $\mathbf{P}$ such that*

$$\lambda(\mathbf{P}, \Xi) = \min_{X \geq 0, \mathrm{Tr}(X)=1} \max_{P \in \mathbf{P}} \langle \Xi(X), P \rangle = \max_{P \in \mathbf{P}} \min_{X \geq 0, \mathrm{Tr}(X)=1} \langle \Xi(X), P \rangle$$

satisfies $\lambda(\mathbf{P}, \Xi) \leq \sqrt{1-a^2}$ if $x \in L$ and $\lambda(\mathbf{P}, \Xi) \geq 1 - b$ if $x \notin L$.

Proof. This theorem follows from simple conversions between different distance measures of quantum states. By Fuchs-van de Graaf inequality (Lemma 2.1.1), we have

- if $x \in L$, then $\exists \rho_0, \rho_1$ such that $\mathrm{F}(Q_0(\rho_0), Q_1(\rho_1)) \geq a$ and thus $\| Q_0(\rho_0) - Q_1(\rho_1) \|_{\mathrm{Tr}} \leq 2\sqrt{1-a^2}$.
- if $x \notin L$, then $\forall \rho_0, \rho_1$, we have $\mathrm{F}(Q_0(\rho_0), Q_1(\rho_1)) \leq b$ and thus $\| Q_0(\rho_0) - Q_1(\rho_1) \|_{\mathrm{Tr}} \geq 2(1-b)$.

Given any $\rho = (\rho_0, \rho_1)$ as the input state[1], let $\Xi(\rho) = Q_0(\rho_0) - Q_1(\rho_1)$. Also by Eq. (2.1.1), we can rephrase the trace norm $\| \Xi(\rho) \|_{\mathrm{Tr}}$ as,

$$\| \Xi(\rho) \|_{\mathrm{Tr}} = 2 \max_{\Pi : 0 \leq \Pi \leq \mathbb{1}} \langle \Xi(\rho), \Pi \rangle .$$

Thus, we can let $\mathbf{P} = \{\Pi : 0 \leq \Pi \leq \mathbb{1}\}$ and define $\lambda(\mathbf{P}, \Xi)$. It follows easily from the above two observations that: if $x \in L$, then $\lambda(\mathbf{P}, \Xi) \leq \sqrt{1-a^2}$; and if $x \notin L$, then $\lambda(\mathbf{P}, \Xi) \geq 1 - b$. □

A direct consequence of Theorem 4.1.2 is that it suffices to approximate $\lambda(\mathbf{P}, \Xi)$ to sufficient precision to distinguish between yes and no instances. For example, if we choose $a = 0.9, b = 0.1$, then it suffices to distinguish between

$$\lambda(\mathbf{P}, \Xi) \leq 0.44 \quad \text{and} \quad \lambda(\mathbf{P}, \Xi) \geq 0.9.$$

[1] One can simply image ρ_0 and ρ_1 as reduced states of ρ on different parts.

It is also from Theorem 4.1.2 (also similar to our QIP(2) example) that all the conditions of Corollary 3.2.2 are met. In particular, the optimization problem (i.e., $\max_{P\in\mathbf{P}} \langle \Xi(\rho^*), P\rangle$) required in Corollary 3.2.2 reduces to the projection onto that positive eignspace of $\Xi(\rho^*)$. Therefore by invoking Corollary 3.2.2, we prove the containment of the close images problem inside PSPACE.

4.2 QMAM Approach

In this section, we demonstrate how Framework 3.1.1 can be applied to the SDPs of QMAM. Since our goal is to illustrate the generality of our equilibrium value method, we directly cite the SDPs of QMAM in [60] as follows:

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:	$\mathrm{Tr}_{\mathcal{Y}}(\rho) \leq \frac{1}{2}\mathbb{1}_{\mathcal{A}} \otimes \sigma$
	$\rho \in \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y})$		$\rho \in \mathrm{D}(\mathcal{A} \otimes \mathcal{X} \otimes \mathcal{Y})$
	$\sigma \in \mathrm{D}(\mathcal{X})$,		$\sigma \in \mathrm{D}(\mathcal{X})$,

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

Definition 4.2.1. 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 $c(|x|) - s(|x|) = \Omega(1/\mathrm{poly}(|x|))$.

Following Framework 3.1.1, we define

$$f_2(\{\rho,\sigma\},\Pi) = \left\langle \begin{pmatrix} c-\langle R,\rho\rangle & \\ & \mathrm{Tr}_{\mathcal{Y}}(\rho)-\frac{1}{2}\mathbb{1}_{\mathcal{A}}\otimes\sigma \end{pmatrix}, \Pi \right\rangle, \tag{4.2.1}$$

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 $\check{\lambda}_2$ be the equilibrium value of function f_2, namely,

$$\check{\lambda}_2 = \min_{\{\rho,\sigma\}\in T_1}\max_{\Pi\in T_2} f_2(\{\rho,\sigma\},\Pi) = \max_{\Pi\in T_2}\min_{\{\rho,\sigma\}\in T_1} f_2(\{\rho,\sigma\},\Pi).$$

Based on Theorem 3.1.2, the value of $\check{\lambda}_2$ determines whether the original problem is feasible. We note that it suffices to choose $c = \frac{1}{2}(c(|x|)+s(|x|))$ to distinguish between the two promises.

Lemma 4.2.2. *Given the two promises in Definition 4.2.1, let $\Delta = c(|x|) - s(|x|)$, then we have*

- *If $x \in L$, then $\check{\lambda}_2 \leq 0$.*
- *If $x \notin L$, then $\check{\lambda}_2 \geq \frac{1}{8}\Delta^2$.*

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 $\check{\lambda}_2 \leq 0$.
- Otherwise, let $(\{\check{\rho},\check{\sigma}\},\check{\Pi})$ be any equilibrium point. Due to Eq. (2.1.1), we have

$$\check{\lambda}_2 = f_2(\{\check{\rho},\check{\sigma}\},\check{\Pi}) = \max\{c-\langle R,\check{\rho}\rangle, 0\} + \frac{1}{2}\|\mathrm{Tr}_{\mathcal{Y}}(\check{\rho}) - \frac{1}{2}\mathbb{1}_{\mathcal{A}}\otimes\sigma\|_{\mathrm{Tr}}. \tag{4.2.2}$$

By Lemma 3.3.1, there exists a $\tilde{\rho} \in \mathrm{D}(\mathcal{A}\otimes\mathcal{X}\otimes\mathcal{Y})$ such that $\mathrm{F}(\frac{1}{2}\mathbb{1}_{\mathcal{A}}\otimes\sigma, \mathrm{Tr}_{\mathcal{Y}}(\check{\rho})) = \mathrm{F}(\tilde{\rho},\check{\rho})$ and $\mathrm{Tr}_{\mathcal{Y}}(\tilde{\rho}) = \frac{1}{2}\mathbb{1}_{\mathcal{A}}\otimes\sigma$. Let $s = \frac{1}{2}\|\mathrm{Tr}_{\mathcal{Y}}(\check{\rho}) - \frac{1}{2}\mathbb{1}_{\mathcal{A}}\otimes\sigma\|_{\mathrm{Tr}}$ and $t = \frac{1}{2}\|\tilde{\rho}-\check{\rho}\|_{\mathrm{Tr}}$. Then if $t \leq \frac{1}{2}\Delta$, we have

$$\begin{aligned}\check{\lambda}_2 &\geq c - \langle R,\tilde{\rho}\rangle + \langle R,\tilde{\rho}-\check{\rho}\rangle + s \geq \frac{1}{2}\Delta - t + s \\ &\geq \frac{1}{2}\Delta - t + 1 - \sqrt{1-t^2} \geq \frac{1}{2}\Delta - \frac{1}{2}\Delta + 1 - \sqrt{1-\frac{1}{4}\Delta^2} \geq \frac{1}{8}\Delta^2,\end{aligned}$$

where the first inequality is due to Eq. (4.2.2), the second inequality comes from Eq. (2.1.1) and the third inequality comes from Lemma 3.3.4. 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 Eq. (4.2.2),

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

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

□

The above lemma establishes the fact approximating $\check{\lambda_2}$ to appropriate precision (say, $O(\Delta^2)$) is sufficient to determine whether $x \in L$. We then invoke Corollary 3.2.2 again to implement this approximation in PSPACE. There is one slight difference here as T_1 is defined to be over $\{\rho, \sigma\}$ instead of a single density operator. We note that such a variance can be easily handled by simultaneously running two matrix multiplicative weight update methods.

CHAPTER 5

Quantum Refereed Games

This chapter is based on a joint work with Gus Gutoski [53].

In this chapter we demonstrate how our equilibrium value method extends to a much more sophisticated situation to simulate short *quantum refereed games*. As we mentioned in Chapter 3, two main difficulties in applying the equilibrium value method are first to approximate equilibrium values and second to convert approximate solutions into exact solutions through a rounding theorem. Here we encounter both difficulties at the same time. We overcome both difficulties with the help of a penalization idea with well-designed penalties and a recursive rounding theorem that makes non-trivial use of the Bures angle. We package our findings as an abstract mathematical result, which also finds other applications, for example, parallel algorithms for a class of semidefinite programs.

Applying our result to quantum refereed games, we determine the complexity of all short-turn quantum refereed games. In particular, we define a new complexity model called *double quantum interactive proof systems*, which can be simulated in PSPACE by invoking our result, and thus has the equal expressive power as PSPACE. This complexity model is so general that our result subsumes and unifies all previous results about short refereed games. Our findings also imply a crucial difference between public and private randomness in refereed games.

The rest of this chapter is organized as follows. We summarize our abstract math-

ematical result in Section 5.1, together with its two main applications and a sketch of techniques. A brief preliminary on quantum fidelity and the Bures angle is provided in Section 5.2, followed by our rounding theorem in Section 5.3. We present our main algorithm in Section 5.4 and then illustrate how to use this algorithm to simulate double quantum interactive proof systems in Section 5.5. We conclude with some extensions of the main results in Section 5.6

5.1 Introduction

This chapter presents a parallel approximation scheme for a new class of min-max problems with applications to classical and quantum zero-sum games and interactive proofs. In order to describe this class of min-max problems let us begin by considering a semidefinite program (SDP) of the form

$$\begin{array}{ll} \text{minimize} & \operatorname{Tr}(X_k P) \\ \text{subject to} & \operatorname{Tr}_{\mathbb{M}_n}(X_{i+1}) = \Phi_i(X_i) \text{ for } i = 1, \ldots, k-1 \\ & \operatorname{Tr}(X_1) = 1 \\ & 0 \preceq X_1, \ldots, X_k \in \mathbb{M}_{mn}. \end{array} \tag{5.1.1}$$

Here $\mathbb{M}_d$ denotes the space of all $d \times d$ complex matrices and $\operatorname{Tr}_{\mathbb{M}_n}$ is the *partial trace*—the unique linear map from matrices to matrices satisfying

$$\operatorname{Tr}_{\mathbb{M}_n} : \mathbb{M}_{mn} \to \mathbb{M}_m : A \otimes B \mapsto \operatorname{Tr}(B) A$$

for every choice of $A \in \mathbb{M}_m$ and $B \in \mathbb{M}_n$. An SDP (5.1.1) is specified by arbitrary choices of a positive semidefinite matrix $P \geq 0$ with $\|P\|_\infty \leq 1$ and completely positive and trace-preserving supper-operators

$$\Phi_1, \ldots, \Phi_{k-1} : \mathbb{M}_{mn} \to \mathbb{M}_m.$$

Let $\mathbf{A}$ denote the feasible region of the SDP (5.1.1) (which is always non-empty) and let $\mathbf{P} \subset \mathbb{M}_{mn}$ be a non-empty compact convex subset of positive semidefinite matrices having spectral norm at most 1. We are concerned with the following min-max problem, which is a generalization of the SDP (5.1.1):

$$\lambda(\mathbf{A}, \mathbf{P}) \stackrel{\text{def}}{=} \min_{(X_1,\ldots,X_k)\in\mathbf{A}} \max_{P\in\mathbf{P}} \operatorname{Tr}(X_k P). \tag{5.1.2}$$

The ordering of minimization and maximization is immaterial, as implied by well-known extensions of von Neumann's Min-Max Theorem [107, 39] given the fact that $\mathbf{A}, \mathbf{P}$ are convex compact sets and $\operatorname{Tr}(X_k P)$ is a bilinear form over the two sets.

Our main result is an efficient parallel oracle-algorithm for finding approximate solutions to the min-max problem (5.1.2) and for approximating the quantity $\lambda(\mathbf{A}, \mathbf{P})$, given an oracle for optimization over the set $\mathbf{P}$. We also describe parallel implementations of this oracle for certain sets $\mathbf{P}$, yielding an unconditionally efficient parallel approximation scheme for the min-max problem (5.1.2) for those choices of $\mathbf{P}$. This result is stated formally below as Theorem 5.1.3. Before stating this theorem let us clarify about our terminology. We refer readers to Section 3.5 on backgrounds about parallel computation.

An *oracle-algorithm* is an algorithm endowed with the ability to get instantaneous answers to questions that fall within the scope of some specific *oracle*. In our case, we assume an oracle for optimization over $\mathbf{P}$, which instantly solves problems of the form

Problem 5.1.1 (Optimization over $\mathbf{P}$)**.**

Input: *A matrix* $X \succeq 0$ *with* $\operatorname{Tr}(X) = 1$ *and an accuracy parameter* $\delta > 0$.

Output: *A near-optimal element* $P^\star \in \mathbf{P}$ *such that* $\operatorname{Tr}(XP^\star) \geq \operatorname{Tr}(XP) - \delta$ *for all* $P \in \mathbf{P}$.

An oracle is incorporated into the circuit model of computation by supplementing a standard gate set (such as $\{\text{AND}, \text{OR}, \text{NOT}\}$) with a special *oracle gate*. This oracle

gate has many input bits (describing the question) and many output bits (describing the answer). As with standard gates, each oracle gate contributes unit cost to circuit size and run time.

An *approximation scheme* refers to an algorithm that computes one or more quantities to a given precision δ and whose run time is efficient for each fixed choice of $\delta > 0$ but does not necessarily scale well with δ. In the circuit model (and other models, too) this property is encapsulated by defining the underlying problem so that the accuracy parameter $\delta = 1/s$ is specified in unary as 1^s, thus forcing the bit length of the input to be proportional to $1/\delta$ instead of $1/\log(\delta)$. The choice to specify the accuracy parameter in unary allows parallel approximation schemes to be described neatly by log-space uniform circuits with polylog depth.

The following is a formal statement of the problem solved by our algorithm.

Problem 5.1.2 (Approximation of $\lambda(\mathbf{A}, \mathbf{P})$)**.**

Input: *Completely positive and trace-preserving linear maps* $\Phi_1, \ldots, \Phi_{k-1}$ *specifying the feasible region* $\mathbf{A}$ *of an SDP of the form* (5.1.1)*. An accuracy parameter* $\delta > 0$*.*

Oracle: *Optimization over* $\mathbf{P}$ *(Problem 5.1.1).*

Output: *Near-optimal elements* $(X_1^\star, \ldots, X_k^\star) \in \mathbf{A}$ *and* $P^\star \in \mathbf{P}$ *such that*

$$\mathrm{Tr}(X_k^\star P) \leq \lambda(\mathbf{A}, \mathbf{P}) + \delta \text{ for all } P \in \mathbf{P}$$

$$\mathrm{Tr}(X_k P^\star) \geq \lambda(\mathbf{A}, \mathbf{P}) - \delta \text{ for all } (X_1, \ldots, X_k) \in \mathbf{A}$$

and a quantity $\tilde{\lambda}$ *with* $\left|\tilde{\lambda} - \lambda(\mathbf{A}, \mathbf{P})\right| \leq \delta$*.*

Theorem 5.1.3 (Main result)**.** *There is a parallel oracle-algorithm for Problem 5.1.2 (Approximation of* $\lambda(\mathbf{A}, \mathbf{P})$*) with run time bounded by a polynomial in* k*,* $1/\delta$*, and* $\log(mn)$*. This algorithm is efficient if* $k, 1/\delta$ *are promised to scale as a polynomial in* $\log(mn)$*.*

5.1.1 Application: Parallel Approximation of Semidefinite Programs

The SDP (5.1.1) is recovered from (5.1.2) in the special case where $\mathbf{P} = \{P\}$ is a singleton set. Thus, a special case of Theorem 5.1.3 is a parallel approximation scheme for SDPs of the form (5.1.1).

We restricted attention to SDPs for which $\|P\|, \mathrm{Tr}(X_1) \leq 1$ because this restriction does not interfere with our application to quantum interactive protocols and because the run time of our parallel algorithm scales polynomially with the largest eigenvalue of P and with the trace of X_1, so it is only efficient when these quantities are bounded by a fixed polynomial in the logarithm of the bit length of the input $P, \Phi_1, \ldots, \Phi_{k-1}$. (In keeping with convention, one can think of these quantities as the *width* of the SDPs we consider. Our algorithm is efficient only for *width-bounded* SDPs.)

It has long since been known that the problem of approximating the optimal value of an arbitrary SDP is logspace-hard for P [99, 89], so there cannot be a parallel approximation scheme for *all* SDPs unless $\mathrm{NC} = \mathrm{P}$. The precise extent to which SDPs admit parallel solutions is not known. This special case of our result adds considerably to the set of such SDPs, subsuming all prior work in the area at the time it was made public. (Since that time parallel approximation schemes have been found for some SDPs of unbounded width that are not covered by our scheme [63, 93, 64].)

Some of what is known about SDPs in this respect is inherited knowledge from linear programs (LPs). For example, Luby and Nisan describe a parallel approximation scheme for so-called *positive* LPs of the form

$$\text{minimize } xp^* \text{ subject to } Cx \geq q \text{ and } x \geq 0$$

where each entry of the matrix C and vectors p, q is a nonnegative real number [85]. Young provides a generalization of Luby-Nisan to arbitrary mixed packing and covering problems

[116]. By contrast, Trevisan and Xhafa show that it is P-hard to find *exact* solutions for positive LPs [106].[1]

The notion of a positive instance of an LP can be generalized to SDPs as follows. An SDP of the form

$$\text{minimize } \mathrm{Tr}(XP) \text{ subject to } \Psi(X) \succeq Q \text{ and } X \succeq 0$$

is said to be *positive* if $P, Q \succeq 0$ and Ψ is a positive map. Of course, P-hardness of exact solutions for positive LPs implies P-hardness of exact solutions for positive SDPs. Jain and Watrous give a parallel approximation scheme for width-bounded positive SDPs [62]. Subsequent improvements extend to all positive SDPs [63, 93], and even to mixed packing and covering SDPs [64].

The Jain-Watrous algorithm for positive SDPs is derived from a correspondence between positive SDPs and one-turn quantum games and can therefore be recovered as a special case of the work in this chapter. In their proof of QIP = PSPACE, Jain *et al.* give a parallel algorithm for a specific SDP based on quantum interactive proofs [60]. It is not difficult to see that their SDP can be written in the form (5.1.1) considered in this chapter.

As mentioned above, our algorithm is not efficient when used for SDPs of unbounded width, leaving the recent works of Jain and Yao [63, 64] and Peng and Tangwongsan [93] on mixed packing and covering SDPs as the only known parallel SDP approximation schemes that are not subsumed by the present work. These recent works do not subsume our results, as neither the SDP instance used in Ref. [60] to prove QIP = PSPACE nor its generalization (5.1.1) in this chapter are mixed packing and covering SDPs.

[1] For clarification, a polynomial-time algorithm finds an *exact* solution to an LP or SDP if it finds solutions that are within ε of optimal in time polynomial in the bit length of ε—that is, $\log(1/\varepsilon)$. By contrast, an *approximation scheme* for LPs or SDPs finds solutions that are within ε of optimal with run time that depends super-polynomially in the bit length of ε—typically $1/\varepsilon$.

5.1.2 Application: Refereed Games

We refer to Section 2.2.2 for the formal definition of the complexity class RG. Here we consider an interesting subclass of RG obtained by placing restrictions upon the number and timing of messages in the interaction between the verifier and provers. Precisely, we introduce one such subclass based upon interactions of the following form:

1. The verifier exchanges several messages with only the yes-prover.
2. After processing this interaction with the yes-prover, the verifier exchanges several additional messages with only the no-prover.
3. After further processing, the referee declares acceptance or rejection.

Interactive proofs of this form shall be called *double interactive proofs*: the verifier in such a protocol executes a standard single-prover interactive proof with the yes-prover followed by a second single-prover interactive proof with the no-prover. The class of problems that admit double interactive proofs shall be called DIP.

By contrast to RG, it is not immediately clear that the definition of DIP is robust with respect to the choice of parameters c, s. But it follows from our result that DIP is, in fact, robust with respect to the choice of c, s. Also, whereas RG is trivially closed under complement, the protocol for double interactive proofs is asymmetric and so it is not immediately clear that DIP is closed under complement. Again, it follows from our result that DIP is closed under complement.

Another example of an interesting subclass of RG is the family of *bounded-turn* classes. For each positive integer k the class $\mathrm{RG}(k)$ consists of those problems that admit an interactive proof with competing provers in which the verifier exchanges no more than k messages with each prover. It is understood that messages are exchanged with the provers in parallel so that $\mathrm{RG}(k)$, like RG, is trivially closed under complement.

Quantum interactive proofs with competing provers are defined similarly except that the verifier is a polynomial-time quantum computer who exchanges quantum information with the provers. The analogous complexity classes are denoted QRG, DQIP, and QRG(k).

Prior Work. As noted in Refs. [41, 40], the results of Koller and Meggido [80] and Koller, Megiddo, and von Stengel [81] imply that RG $\subseteq$ EXP. The reverse containment was proven by Feige and Kilian [40], yielding the characterization RG $=$ EXP. It was proven in Ref. [52] that QRG $\subseteq$ EXP, from which one obtains

$$\text{QRG} = \text{RG} = \text{EXP},$$

which is the competing-provers version of the well-known collapse QIP $=$ IP $=$ PSPACE for single-prover interactive proofs [86, 100, 60].

For bounded-turn classes, the results of Fortnow et al. tell us that RG(1) is essentially a randomized version of S_2^{P} [44]. Feige and Kilian proved RG(2) $=$ PSPACE [40].[2] For bounded-turn quantum classes, [62] proved QRG(1) $\subseteq$ PSPACE. The complexity of QRG(2) is an open question of [60] that is solved in this chapter. The exact complexity of RG(k) and QRG(k) for all other k is not known.

Bounded-turn double quantum interactive proofs have been studied previously under the name *short quantum games*; the associated complexity class has been called SQG. In an effort to unify notation let DQIP(k,l) denote the class consisting of problems that admit a double quantum interactive proof with competing provers in which the verifier exchanges no more than k messages with the yes-prover followed by no more than l messages with the no-prover. The class SQG was first defined in Ref. [51] to be equal to DQIP$(1,2)$, wherein it was shown that this class contains QIP $=$ DQIP$(\text{poly},0)$. The importance of short quantum games has been diminished by the proof of QIP $=$ PSPACE, as containment

[2] The class we call RG(2) is called RG(1) by Feige and Kilian [40]. This conflict in notation stems from the fact that we measure the length of an interaction in *turns* (*i.e.* messages per prover), whereas those authors measure an interaction in *rounds* of messages. This switch of notation was instigated by Jain and Watrous, who required a convenient symbol for one-turn interactions [62].

of QIP is no longer such a peculiar property. However, the containment of PSPACE inside $\mathrm{DQIP}(1,2)$ is still interesting, as it is not known whether PSPACE is contained in $\mathrm{DIP}(1,2)$, the classical version of this class.

Our contribution

As we explain in Section 5.5, the oracle-algorithm of Theorem 5.1.3—together with a parallel implementation of a suitably chosen oracle—implies that near-optimal strategies for the provers in a double quantum interactive proof can be computed efficiently in parallel. The following containment then follows from a standard argument (summarized in Section 5.5.4).

Theorem 5.1.4. $\mathrm{DQIP} \subseteq \mathrm{PSPACE}$.

This containment, when combined with the trivial containments $\mathrm{IP} \subseteq \mathrm{DIP} \subseteq \mathrm{DQIP}$ and the well-known fact that $\mathrm{PSPACE} \subseteq \mathrm{IP}$ [86, 100], yields the following characterization.

Corollary 5.1.5. $\mathrm{DQIP} = \mathrm{DIP} = \mathrm{PSPACE}$.

As a special case of Corollary 5.1.5 we obtain the solution to an open problem of [60]:

Corollary 5.1.6.

$$\mathrm{QRG}(2) = \mathrm{PSPACE}.$$

Another special case of our result is a direct polynomial-space simulation of multi-message quantum interactive proofs, resulting in a first-principles proof of $\mathrm{QIP} = \mathrm{PSPACE}$.

Corollary 5.1.7.

$$\mathrm{QIP} = \mathrm{PSPACE}$$

via direct polynomial-space simulation of multi-message quantum interactive proofs.

By contrast, all other known proofs [60, 113] rely upon the fact that the verifier can be assumed to exchange only three messages with the prover [74]. The original proof of Jain *et al.* [60] also relies on the additional fact that the verifier's only message to the prover can be just a single classical coin flip [88].

Of course, every other competing-provers complexity class whose protocol can be cast as a double interactive proof also collapses to PSPACE, such as the aforementioned class $\mathrm{DQIP}(1,2)$ based on short quantum games.

It follows from the collapse of DQIP and DIP to PSPACE that these classes are closed under complement and that they are robust with respect to the choice of parameters c, s. (Indeed, it may be assumed that $c = 1$ and $s \leq 2^{-q}$ for any desired polynomially-bounded function $q(|x|)$—see Section 5.6.3.)

Prior to our work polynomial-space algorithms were known only for two-turn classical interactive proofs with competing provers ($\mathrm{RG}(2)$), for one-turn quantum interactive proofs with competing provers ($\mathrm{QRG}(1)$), and for single-prover quantum interactive proofs (QIP). Our result unifies and subsumes all of these algorithms. It also demonstrates for the first time the existence of a polynomial-space algorithm for a competing-prover interaction (classical or quantum) in which one prover reacts adaptively to the other.

Finally, our results illustrate a difference in the effect of public randomness between *single*-prover interactive proofs and *competing*-prover interactive proofs. Any classical interactive proof with single prover can be simulated by another *public-coin* interactive proof where the verifier's messages to the prover consist entirely of uniformly random bits and the verifier uses no other randomness [48]. Extending the notion of public-coin interaction to competing-prover interactions, it is easy to see that any such interaction with a public-coin verifier can be simulated by a double interactive proof.[3] We therefore have

[3] *Proof sketch:* As the verifiers's questions to each prover are uniformly random, they cannot depend on prior responses from the other prover and can therefore be reordered so that all messages with one prover are exchanged before any messages with the other.

that the public-coin version of RG is a subset of DIP, which we now know is equal to PSPACE. Thus, by contrast to the single-prover case where public-coin-IP $=$ IP, in the competing-prover case we establish the following.

Corollary 5.1.8. public-coin-RG $\neq$ RG *unless* PSPACE $=$ EXP.

5.1.3 Summary of Techniques

Equilibrium Value Method

We mentioned earlier about the difficulties we will encounter in the use of the equilibrium value method. The first one is that we cannot directly apply Corollary 3.2.2 to approximate equilibrium values in Eq. (5.1.2). This is because, in its unaltered form, the MMW can only be used to solve min-max problems over the domain of *density operators*—positive semidefinite matrices X with $\mathrm{Tr}(X) = 1$. We introduce a new extension to this method for min-max problems over the domain $\mathbf{A}$ defined in the SDP (5.1.1)—a domain consisting of *k-tuples of density operators* lying within a *strict subspace* of the affine space associated with k-tuples of density operators. The high-level approach of our method is as follows:

1. **Extend the domain from a single density matrix to a k-tuple of density matrices.**
   This step is straightforward: the MMW can be applied without complication to all k density matrices at the same time. (Equivalently, k density matrices may be viewed as a single, larger, block-diagonal density matrix.)

2. **Restrict the domain to a strict subspace of k-tuples of density matrices.**
   This step is more difficult. It is accomplished by relaxing the problem so as to allow *all* k-tuples, with an additional *penalty term* to remove incentive for the players to use inconsistent transcripts.

3. **Round strategies in the relaxed problem to strategies in the original protocol.**
   For this step one must prove a "rounding" theorem (Theorem 5.3.1), which estab-

lishes that near-optimal, fully admissible strategies can be obtained from near-optimal strategies in the unrestricted domain with penalty terms.

It is also crucial to design an appropriate rounding theorem. The first challenge is to find good penalty terms such that these terms can be naturally incorporated into the min-max forms and at the same time are powerful enough to round approximate solutions to exact ones. The second challenge lies in the proof of such a rounding theorem. Since we are dealing with polynomially many quantum messages at the same time, where each message has dependence on the previous one, we need to invoke a recursive structure in the proof. To avoid unaffordable losses during the recursion, we make non-trivial use of the Bures angle.

Finding optimal strategies for the provers in a double quantum interactive proof

In Section 5.5 we observe that the verifier in a double quantum interactive proof induces a min-max problem of the form (5.1.2) in which elements of $\mathbf{A}$ correspond to strategies for the yes-prover and elements of $\mathbf{P}$ correspond to strategies for the no-prover. Thus, the parallel oracle-algorithm of Theorem 5.1.3—together with a parallel implementation of the oracle for optimization over $\mathbf{P}$—can be used to find optimal strategies for the provers in a double quantum interactive proof.

Our implementation of this oracle is itself a special case of the algorithm of Theorem 5.1.3, so that the overall algorithm employs the MMW method *twice* in a two-level recursive fashion. At the top level the MMW is used to iteratively converge toward an optimal strategy for the yes-prover; at the bottom level the MMW is used again to solve an SDP for "best responses" for the no-prover to a given strategy for the yes-prover.

The central challenge in using the MMW to find optimal strategies for parties in a quantum interaction is to find a representation for strategies that is amenable to the MMW

method. In Kitaev's *transcript* representation [72] the actions of a prover in a double quantum interactive proof are represented by a list $X_1, \ldots, X_k$ of density matrices that satisfy a special consistency condition that is captured by the definition of the feasible region $\mathbf{A}$ of the SDP (5.1.1). Intuitively, these density matrices correspond to "snapshots" of the state of the verifier's qubits at various times during the interaction. (See Figure 5.3 on page 77.)

The key property of double quantum interactive proofs that we exploit is the ability to draw a "temporal line" in the interaction before which only the yes-prover acts and after which only the no-prover acts. Given a transcript $X_1, \ldots, X_k$ for the yes-prover, the actions of the no-prover can then be represented by another transcript $Y_1, \ldots, Y_\ell$. By optimizing over all such transcripts one obtains an oracle for "best responses" for the no-prover to a given strategy of the yes-prover as required by the MMW method.

Comparison of proofs of QIP = PSPACE

Unlike our proof, the original proof of QIP = PSPACE due to Jain *et al.* [60] does not take advantage of the transcript representation for arbitrary multi-turn strategies. Instead, as mentioned earlier, those authors derive a special SDP by invoking several nontrivial facts about quantum interactive proofs. Admittedly, their SDP does bear a resemblance to Kitaev's transcript conditions, but this resemblance is only superficial and their solution applies only to a very restricted subset of transcripts. Indeed, their derivation breaks down without the assumption that the verifier sends only classical messages to the prover.

In Chapter 4, we presented two simplified proofs of QIP = PSPACE that, like the work in this chapter, employs Kale's algorithmic min-max theorem [67] instead of the primal-dual approach for SDPs that was used in the original proof by Jain et al. [60]. Since those proofs all employ non-trivial facts about quantum interactive proofs, they do not require the penalization method introduced in this chapter nor an attendant rounding theorem.

The Bures angle

Finally, it is noteworthy that the proof of our rounding theorem (Theorem 5.3.1) contains an interesting and nontrivial application of the Bures angle, which is a distance measure for quantum states that is defined in terms of the more familiar fidelity function.

Properties of the trace norm, which captures the physical distinguishability of quantum states, are sufficient for most needs in quantum information. When some property of the fidelity is also required one uses the Fuchs-van de Graaf inequalities to convert between the trace norm and fidelity [45]. (See Eq. (2.1.2) in Section 2.1.)

However, every such conversion incurs a quadratic slackening of relevant accuracy parameters. Our study calls for repeated conversions, which would incur an unacceptable exponential slackening if done naively via Fuchs-van de Graaf. Instead, we make only a *single* conversion between the trace norm and the Bures angle and then repeatedly exploit the simultaneous properties of (i) the triangle inequality, (ii) contractivity under quantum channels, and (iii) preservation of subsystem fidelity.

Although conversion inequalities between the trace norm and Bures metric are implied by Fuchs-van de Graaf, to our knowledge explicit conversion inequalities have not yet appeared in published literature. The required inequalities are derived in this chapter (Proposition 5.2.2).

5.2 Fidelity and the Bures Angle

5.2.1 Preservation of Subsystem Fidelity

Consider the following property of the fidelity function, which we call the *preservation of subsystem fidelity*: if ρ, ξ are states of a quantum system with fidelity $F(\rho, \xi)$ and ρ' is any state of a larger system consistent with ρ then it is always possible to find ξ' consistent with ξ such that $F(\rho', \xi') = F(\rho, \xi)$. We exploit this property in our proof of QIP(2)$\subseteq$PSPACE

in Section 3.3.

Proposition 5.2.1 (Preservation of subsystem fidelity: Lemma 3.3.1 and Lemma 3.3.2)**.** *Let $\rho, \xi \in \mathbb{M}_m$ and $\rho' \in \mathbb{M}_{mn}$ be density matrices with $\mathrm{Tr}_{\mathbb{M}_n}(\rho') = \rho$. There exists a density matrix $\xi' \in \mathbb{M}_{mn}$ with $\mathrm{Tr}_{\mathbb{M}_m}(\xi') = \xi$ and $F(\rho', \xi') = F(\rho, \xi)$. Moreover ξ' can be computed efficiently in parallel given ρ, ξ, ρ'.*

5.2.2 The Bures Angle

The *Bures angle* or simply the *angle* $A(\rho, \xi)$ between quantum states ρ, ξ is defined by

$$A(\rho, \xi) \stackrel{\text{def}}{=} \arccos F(\rho, \xi).$$

The angle is a metric on quantum states, meaning that it is nonnegative, equals zero only when $\rho = \xi$, and obeys the triangle inequality [91]. Moreover, the angle is *contractive*, so that

$$A(\Phi(\rho), \Phi(\xi)) \leq A(\rho, \xi)$$

for any quantum channel Φ. The Fuchs-van de Graaf inequalities establish a relationship between the fidelity and trace norm [45]. The inequalities are

$$1 - F(\rho, \xi) \leq \frac{1}{2} \left\| \rho - \xi \right\|_{\mathrm{Tr}} \leq \sqrt{1 - F(\rho, \xi)^2}. \tag{5.2.1}$$

These inequalities can be used to derive a relationship between $A(\rho, \xi)$ and $\left\| \rho - \xi \right\|_{\mathrm{Tr}}$. For example,

Proposition 5.2.2 (Relationship between trace norm and Bures angle)**.** *For all density matrices ρ, ξ it holds that*

$$\frac{1}{2} \left\| \rho - \xi \right\|_{\mathrm{Tr}} \leq A(\rho, \xi) \leq \sqrt{\frac{\pi}{2} \left\| \rho - \xi \right\|_{\mathrm{Tr}}}.$$

Proof. The lower bound on $A(\rho, \xi)$ follows immediately from Fuchs-van de Graaf:

$$\frac{1}{2} \left\| \rho - \xi \right\|_{\mathrm{Tr}} \leq \sqrt{1 - \cos A(\rho, \xi)^2} = \sin A(\rho, \xi) \leq A(\rho, \xi),$$

where we used the identity $\sin x \leq x$ for all $x \geq 0$.

To obtain the upper bound on $A(\rho, \xi)$ we employ the identity $\cos x \leq 1 - x^2/\pi$ for $x \in [0, \pi/2]$, which can be verified using basic calculus. Then we have

$$\frac{1}{2} \|\rho - \xi\|_{\mathrm{Tr}} \geq 1 - \cos A(\rho, \xi) \geq \frac{A(\rho, \xi)^2}{\pi}$$

from which the proposition follows. □

5.3 Rounding Theorem for a Relaxed Min-Max Problem

In this section we define a new min-max expression $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ that approximates the desired quantity $\lambda(\mathbf{A}, \mathbf{P})$ from (5.1.2) in the limit as ε approaches zero. This new expression is a relaxation of $\lambda(\mathbf{A}, \mathbf{P})$ that is more amenable to the MMW. We prove a "rounding theorem" (Theorem 5.3.1) by which near-optimal points for $\lambda(\mathbf{A}, \mathbf{P})$ are efficiently obtained from near-optimal points for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$. Our use of the Bures angle occurs in the proof of Lemma 5.3.4, which is used in the proof of our rounding theorem.

Define the relaxation $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ of $\lambda(\mathbf{A}, \mathbf{P})$ by

$$\begin{aligned}
\mu_\varepsilon(\mathbf{A}, \mathbf{P}) &\stackrel{\text{def}}{=} \min_{(\rho_1, \ldots, \rho_k)} \max_{\substack{P \in \mathbf{P} \\ (\Pi_1, \ldots, \Pi_{k-1})}} \langle \rho_k, P \rangle + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \langle \mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i), \Pi_i \rangle \\
&= \min_{(\rho_1, \ldots, \rho_k)} \max_{P \in \mathbf{P}} \langle \rho_k, P \rangle + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \frac{1}{2} \|\mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i)\|_{\mathrm{Tr}}.
\end{aligned}$$

Here the minimum is taken over all density operators $\rho_1, \ldots, \rho_k \in \mathbb{M}_{mn}$ and the maximum over all $P \in \mathbf{P}$ and over all measurement operators $\Pi_1, \ldots, \Pi_{k-1} \in \mathbb{M}_m$. The second equality follows immediately from Eq. (2.1.1) in Section 2.1.

Notice that the minimum in the definition of $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ is taken over *all* k-tuples $(\rho_1, \ldots, \rho_k)$ of density operators, not just those in $\mathbf{A}$. Each term in the summation serves to penalize any violation of the conditions required for membership in $\mathbf{A}$ by adding the magnitude of that violation to the objective function. The k/ε factor amplifies the penalty so as to

remove incentive to select an element outside of $\mathbf{A}$. Indeed, it is clear that

$$\lim_{\varepsilon \to 0} \mu_\varepsilon(\mathbf{A}, \mathbf{P}) = \lambda(\mathbf{A}, \mathbf{P}).$$

The following "rounding" theorem establishes a specific rate of convergence for this limit. A subsequent extension of this theorem (Proposition 5.3.3) provides a means by which near-optimal points for $\lambda(\mathbf{A}, \mathbf{P})$ are efficiently computed from near-optimal points for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$.

Theorem 5.3.1 (Rounding theorem)**.** *For any $\varepsilon > 0$ it holds that*

$$\lambda(\mathbf{A}, \mathbf{P}) \geq \mu_\varepsilon(\mathbf{A}, \mathbf{P}) > \lambda(\mathbf{A}, \mathbf{P}) - \varepsilon.$$

Proof. The first inequality is easy: let $(\rho_1, \ldots, \rho_k)$ be optimal for $\lambda(\mathbf{A}, \mathbf{P})$ and let $(P, \Pi_1, \ldots, \Pi_{k-1})$ be optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$. Then we have

$$\lambda(\mathbf{A}, \mathbf{P}) \geq \langle \rho_k, P \rangle = \langle \rho_k, P \rangle + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \langle \mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i), \Pi_i \rangle \geq \mu_\varepsilon(\mathbf{A}, \mathbf{P}).$$

(The first inequality is because $(\rho_1, \ldots, \rho_k)$ is optimal for $\lambda(\mathbf{A}, \mathbf{P})$. The equality follows because $(\rho_1, \ldots, \rho_k) \in \mathbf{A}$, so each term in the sum is zero. The final inequality is because $(P, \Pi_1, \ldots, \Pi_{k-1})$ is optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$.)

The second inequality is more difficult. We invoke the following lemma, the proof of which appears later in this section.

Lemma 5.3.2 (Rounding lemma)**.** *For any $\varepsilon > 0$ and any states $\rho_1, \ldots, \rho_k \in \mathbb{M}_{mn}$ there exists $(\rho'_1, \ldots, \rho'_k) \in \mathbf{A}$ such that*

$$\frac{1}{2} \left\| \rho_k - \rho'_k \right\|_{\mathrm{Tr}} < \varepsilon + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \frac{1}{2} \left\| \mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i) \right\|_{\mathrm{Tr}}.$$

Moreover, $\rho'_1, \ldots, \rho'_k$ can be computed efficiently in parallel given $\rho_1, \ldots, \rho_k$.

Let $(\rho_1, \ldots, \rho_k)$ be optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$, let $(\rho'_1, \ldots, \rho'_k)$ be the density operators obtained by invoking Lemma 5.3.2, and let $P \in \mathbf{P}$ be optimal for $\lambda(\mathbf{A}, \mathbf{P})$. Because $(\rho_1, \ldots, \rho_k)$ is optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ we have

$$\mu_\varepsilon(\mathbf{A}, \mathbf{P}) \geq \langle \rho_k, P \rangle + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \frac{1}{2} \left\| \mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i) \right\|_{\mathrm{Tr}}. \tag{5.3.1}$$

Employing the identity (2.1.1), the quantity $\langle \rho_k, P \rangle$ becomes

$$\langle \rho_k, P \rangle = \langle \rho'_k, P \rangle + \langle \rho_k - \rho'_k, P \rangle \geq \langle \rho'_k, P \rangle - \frac{1}{2} \left\| \rho_k - \rho'_k \right\|_{\mathrm{Tr}}.$$

Substituting the bound on $\frac{1}{2} \left\| \rho_k - \rho'_k \right\|_{\mathrm{Tr}}$ from Lemma 5.3.2, we see that the summation of trace norms in (5.3.1) is canceled, leaving

$$\mu_\varepsilon(\mathbf{A}, \mathbf{P}) > \langle \rho'_k, P \rangle - \varepsilon \geq \lambda(\mathbf{A}, \mathbf{P}) - \varepsilon$$

as desired. (The final inequality is because P is optimal for $\lambda(\mathbf{A}, \mathbf{P})$.) □

Proposition 5.3.3 (Construction of near-optimal strategies)**.** *The following hold for any* $\delta, \varepsilon > 0$*:*

1. *If* $(\rho_1, \ldots, \rho_k)$ *is* δ*-optimal for* $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ *then there is an efficient parallel algorithm to compute* $(\rho'_1, \ldots, \rho'_k) \in \mathbf{A}$ *that is* $(\delta + \varepsilon)$*-optimal for* $\lambda(\mathbf{A}, \mathbf{P})$.
2. *If* $(P, \Pi_1, \ldots, \Pi_{k-1})$ *is* δ*-optimal for* $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ *then* P *is also* $(\delta + \varepsilon)$*-optimal for* $\lambda(\mathbf{A}, \mathbf{P})$.

Proof of item 1. Let $(\rho_1, \ldots, \rho_k)$ be δ-optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$, let $(\rho'_1, \ldots, \rho'_k) \in \mathbf{A}$ be obtained by invoking Lemma 5.3.2, and let $P \in \mathbf{P}$. We have

$$\begin{aligned}
\langle \rho'_k, P \rangle &\leq \langle \rho_k, P \rangle + \frac{1}{2} \left\| \rho_k - \rho'_k \right\|_{\mathrm{Tr}} \\
&\leq \langle \rho_k, P \rangle + \varepsilon + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \frac{1}{2} \left\| \mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i) \right\|_{\mathrm{Tr}} \\
&\leq \mu_\varepsilon(\mathbf{A}, \mathbf{P}) + \varepsilon + \delta \leq \lambda(\mathbf{A}, \mathbf{P}) + \varepsilon + \delta.
\end{aligned}$$

(The first inequality follows from (2.1.1); the second from Lemma 5.3.2; the third because $(\rho_1, \ldots, \rho_k)$ is δ-optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$; and the fourth because $\mu_\varepsilon(\mathbf{A}, \mathbf{P}) \leq \lambda(\mathbf{A}, \mathbf{P})$.) It therefore follows that $(\rho'_1, \ldots, \rho'_k)$ is $(\delta + \varepsilon)$-optimal for $\lambda(\mathbf{A}, \mathbf{P})$. □

Proof of item 2. Let $(P, \Pi_1, \ldots, \Pi_{k-1})$ be δ-optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$. For any $(\rho_1, \ldots, \rho_k) \in \mathbf{A}$ we have

$$\begin{aligned} \langle \rho_k, P \rangle &= \langle \rho_k, P \rangle + \frac{k}{\varepsilon} \sum_{i=1}^{k-1} \langle \mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i), \Pi_i \rangle \\ &\geq \mu_\varepsilon(\mathbf{A}, \mathbf{P}) - \delta > \lambda(\mathbf{A}, \mathbf{P}) - \varepsilon - \delta. \end{aligned}$$

(The equality is because $(\rho_1, \ldots, \rho_k) \in \mathbf{A}$ so each term in the sum is zero. The first inequality is because $(P, \Pi_1, \ldots, \Pi_{k-1})$ is δ-optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$. The final inequality is because $\mu_\varepsilon(\mathbf{A}, \mathbf{P}) > \lambda(\mathbf{A}, \mathbf{P}) - \varepsilon$.) It therefore follows that P is $(\delta + \varepsilon)$-optimal for $\lambda(\mathbf{A}, \mathbf{P})$. □

We now prove Lemma 5.3.2, the statement of which appeared in the proof of Theorem 5.3.1. Given any states $\rho_1, \ldots, \rho_k$ this lemma asserts that these states can be "rounded" to an element $(\rho'_1, \ldots, \rho'_k) \in \mathbf{A}$ in such a way that the distance between the final states ρ_k and ρ'_k is bounded by a function of the extent to which $(\rho_1, \ldots, \rho_k)$ violate the conditions required for membership in $\mathbf{A}$. Let us re-state Lemma 5.3.2 in terms of the Bures angle.

Lemma 5.3.4 (Rounding lemma)**.** *For any $\varepsilon > 0$ and any states $\rho_1, \ldots, \rho_k \in \mathbb{M}_{mn}$ there exists $(\rho'_1, \ldots, \rho'_k) \in \mathbf{A}$ such that*

$$A(\rho_k, \rho'_k) \leq \sum_{i=1}^{k-1} A\left(\mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}), \Phi_i(\rho_i)\right).$$

Moreover, $\rho'_1, \ldots, \rho'_k$ can be computed efficiently in parallel given $\rho_1, \ldots, \rho_k$.

Proof. Define $\rho'_1, \ldots, \rho'_k$ recursively as follows. Let $\rho'_1 = \rho_1$. For each $i = 1, \ldots, k-1$ by the preservation of subsystem fidelity (Proposition 5.2.1) there exists ρ'_{i+1} (which can

be computed efficiently in parallel) with

$$\mathrm{Tr}_{\mathbb{M}_n}(\rho'_{i+1}) = \Phi_i(\rho'_i)$$

and

$$A(\rho_{i+1}, \rho'_{i+1}) = A\left(\mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}), \Phi_i(\rho'_i)\right).$$

By the triangle inequality this quantity is at most

$$A\left(\mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}), \Phi_i(\rho_i)\right) + A\left(\Phi_i(\rho_i), \Phi_i(\rho'_i)\right).$$

By contractivity of the Bures angle under channels, the summand on the right is at most $A(\rho_i, \rho'_i)$. The lemma now follows inductively from the fact that $A(\rho_1, \rho'_1) = 0$. □

It is easy to recover Lemma 5.3.2 from Lemma 5.3.4: it follows immediately from Lemma 5.3.4 and Proposition 5.2.2 (Relationship between trace norm and Bures angle) that

$$\frac{1}{2}\left\|\rho_k - \rho'_k\right\|_{\mathrm{Tr}} \leq \sum_{i=1}^{k-1} \sqrt{\frac{\pi}{2}\left\|\mathrm{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i)\right\|_{\mathrm{Tr}}}.$$

Lemma 5.3.2 then follows from the fact that $\sqrt{\frac{\pi}{2}x} < \frac{1}{2\delta}x + \delta$ for all $x \geq 0$ and all $\delta > 0$.

5.4 A Parallel Oracle-algorithm for a Min-Max Problem

In this section we prove Theorem 5.1.3 (Main result) by exhibiting an efficient parallel oracle-algorithm based on our equilibrium value method for finding approximate solutions to the min-max problem (5.1.2). We formally introduce the matrix multiplicative weight update method in Chapter 3. For technical convenience, we introduce the following variant that can be readily used in later proof.

Theorem 5.4.1 (Multiplicative weights update method [67, Theorem 10])**.** *Fix $\gamma \in (0, 1/2)$ and $\alpha > 0$. Let $M^{(1)}, \ldots, M^{(T)}$ be arbitrary $d \times d$ "loss" matrices with $0 \preceq M^{(t)} \preceq \alpha I$.*

Let $W^{(1)}, \ldots, W^{(T)}$ be $d \times d$ "weight" matrices given by

$$W^{(1)} = I \qquad\qquad W^{(t+1)} = \exp\left(-\gamma\left(M^{(1)} + \cdots + M^{(t)}\right)\right).$$

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

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

Let us establish some notation before stating our algorithm. Let $\varepsilon > 0$ and consider the linear mapping $f_{\mathbf{A},\varepsilon}$ with the property that

$$\langle f_{\mathbf{A},\varepsilon}(\rho_1, \ldots, \rho_k), (P, \Pi_1, \ldots, \Pi_{k-1})\rangle = \langle \rho_k, P\rangle + \frac{k}{\varepsilon}\sum_{i=1}^{k-1}\langle \operatorname{Tr}_{\mathbb{M}_n}(\rho_{i+1}) - \Phi_i(\rho_i), \Pi_i\rangle$$

so that we may write

$$\mu_\varepsilon(\mathbf{A}, \mathbf{P}) = \min_{(\rho_1,\ldots,\rho_k)} \max_{\substack{P\in\mathbf{P} \\ (\Pi_1,\ldots,\Pi_{k-1})}} \langle f_{\mathbf{A},\varepsilon}(\rho_1, \ldots, \rho_k), (P, \Pi_1, \ldots, \Pi_{k-1})\rangle.$$

It is clear that the mapping $f_{\mathbf{A},\varepsilon}$ is given by

$$f_{\mathbf{A},\varepsilon} : (\rho_1, \ldots, \rho_k) \mapsto \left(\rho_k, \frac{k}{\varepsilon}\left[\operatorname{Tr}_{\mathbb{M}_n}(\rho_2) - \Phi_1(\rho_1)\right], \ldots, \frac{k}{\varepsilon}\left[\operatorname{Tr}_{\mathbb{M}_n}(\rho_k) - \Phi_{k-1}(\rho_{k-1})\right]\right).$$

It is tedious but straightforward to verify that the adjoint mapping $f^*_{\mathbf{A},\varepsilon}$ is given by

$$f^*_{\mathbf{A},\varepsilon} = \left(f^*_{\mathbf{A},\varepsilon,1}, \ldots, f^*_{\mathbf{A},\varepsilon,k}\right)$$

where

$$\begin{aligned}
f^*_{\mathbf{A},\varepsilon,1} &: (P, \Pi_1, \ldots, \Pi_{k-1}) \mapsto -\frac{k}{\varepsilon}\Phi_1^*(\Pi_1) \\
f^*_{\mathbf{A},\varepsilon,i} &: (P, \Pi_1, \ldots, \Pi_{k-1}) \mapsto \frac{k}{\varepsilon}\left[\Pi_{i-1}\otimes I - \Phi_i^*(\Pi_i)\right] \qquad \text{for } i = 2, \ldots, k-1 \\
f^*_{\mathbf{A},\varepsilon,k} &: (P, \Pi_1, \ldots, \Pi_{k-1}) \mapsto P + \frac{k}{\varepsilon}\Pi_{k-1}\otimes I
\end{aligned}$$

Note that for any $(P, \Pi_1, \ldots, \Pi_{k-1})$ it holds that

$$
\begin{aligned}
-\frac{k}{\varepsilon} I &\preceq f^*_{\mathbf{A},\varepsilon,1}(P, \Pi_1, \ldots, \Pi_{k-1}) \preceq 0 \\
-\frac{k}{\varepsilon} I &\preceq f^*_{\mathbf{A},\varepsilon,i}(P, \Pi_1, \ldots, \Pi_{k-1}) \preceq \frac{k}{\varepsilon} I \qquad \text{for } i = 2, \ldots, k-1 \\
0 &\preceq f^*_{\mathbf{A},\varepsilon,k}(P, \Pi_1, \ldots, \Pi_{k-1}) \preceq \left(1 + \frac{k}{\varepsilon}\right) I \preceq \frac{2k}{\varepsilon} I
\end{aligned}
\tag{5.4.1}
$$

The statement of our MMW algorithm in Figure 5.1 employs these formulae for the adjoint. We are now ready to prove Theorem 5.1.3.

Proof of Theorem 5.1.3. We argue that the theorem is established by the oracle-algorithm presented in Figure 5.1. To this end, note that each loss matrix $M_i^{(t)} \in \mathbb{M}_{mn}$ satisfies $0 \preceq M_i^{(t)} \preceq \frac{1}{k} I$—a fact that follows immediately from their definition in step 2d and the bounds (5.4.1) on the adjoint mapping $f^*_{\mathbf{A},\varepsilon}$.

For each $i = 1, \ldots, k$ it is clear that the construction of the density operators $\rho_i^{(t)}$ in terms of the loss matrices $M_i^{(t)}$ presented in Figure 5.1 are as defined in Theorem 5.4.1. It therefore follows that for any density operator $\rho_i^\star \in \mathbb{M}_{mn}$ we have

$$
\frac{1}{T} \sum_{t=1}^{T} \left\langle \rho_i^{(t)}, M_i^{(t)} \right\rangle \leq \left\langle \rho_i^\star, \frac{1}{T} \sum_{t=1}^{T} M_i^{(t)} \right\rangle + \frac{1}{k} \left(\gamma + \frac{\ln(mn)}{\gamma T} \right).
$$

Summing these inequalities over all i we find that for any density operators $(\rho_1^\star, \ldots, \rho_k^\star)$ it holds that

$$
\begin{aligned}
&\frac{1}{T} \sum_{t=1}^{T} \left\langle \left(\rho_1^{(t)}, \ldots, \rho_k^{(t)}\right), \left(M_1^{(t)}, \ldots, M_k^{(t)}\right) \right\rangle \\
&\leq \left\langle (\rho_1^\star, \ldots, \rho_k^\star), \frac{1}{T} \sum_{t=1}^{T} \left(M_1^{(t)}, \ldots, M_k^{(t)}\right) \right\rangle + \left(\gamma + \frac{\ln(mn)}{\gamma T} \right).
\end{aligned}
$$

Substituting the definition of the loss matrices $M_i^{(t)}$ from step 2d and simplifying, we

1. Let $\varepsilon = \delta/3$, let $\gamma = \frac{\varepsilon\delta}{12k^2}$, and let $T = \left\lceil \frac{\ln(mn)}{\gamma^2} \right\rceil$. Let

$$W_i^{(1)} = I \in \mathbb{M}_{mn}$$

for each $i = 1, \ldots, k$.

2. Repeat for each $t = 1, \ldots, T$:

   (a) For $i = 1, \ldots, k$: Compute the updated density operators $\rho_i^{(t)} = W_i^{(t)} / \operatorname{Tr}(W_i^{(t)})$.

   (b) For $i = 1, \ldots, k-1$: Compute the projection $\Pi_i^{(t)} \in \mathbb{M}_m$ onto the positive eigenspace of

$$\operatorname{Tr}_{\mathbb{M}_n}(\rho_{i+1}^{(t)}) - \Phi_i(\rho_i^{(t)}).$$

   (c) Use the oracle to obtain a $\delta/3$-optimal solution $P^{(t)} \in \mathbb{M}_{mn}$ to the optimization problem for $\mathbf{P}$ (Problem 5.1.1) on input $\rho_k^{(t)}$.

   (d) Compute the loss matrices

$$\left(M_1^{(t)}, \ldots, M_k^{(t)}\right) = \frac{\varepsilon}{2k^2}\left[f_{R,\varepsilon}^*\left(P^{(t)}, \Pi_1^{(t)}, \ldots, \Pi_{k-1}^{(t)}\right) + \frac{k}{\varepsilon}(I, \ldots, I, 0).\right]$$

   (e) Update each weight matrix according to the standard MMW update rule:

$$W_i^{(t+1)} = \exp\left(-\gamma\left(M_i^{(1)} + \cdots + M_i^{(t)}\right)\right).$$

3. Return

$$\tilde{\lambda} = \frac{1}{T}\sum_{t=1}^{T}\left\langle f_{R,\varepsilon}\left(\rho_1^{(t)}, \ldots, \rho_a^{(t)}\right), \left(P^{(t)}, \Pi_1^{(t)}, \ldots, \Pi_{k-1}^{(t)}\right)\right\rangle$$

as the δ-approximation to $\lambda(\mathbf{A}, \mathbf{P})$.

4. Compute

$$(\rho_1, \ldots, \rho_k) = \frac{1}{T}\sum_{t=1}^{T}(\rho_1^{(t)}, \ldots, \rho_k^{(t)})$$

$$(P, \Pi_1, \ldots, \Pi_{k-1}) = \frac{1}{T}\sum_{t=1}^{T}(P^{(t)}, \Pi_1^{(t)}, \ldots, \Pi_{k-1}^{(t)}),$$

the pair of which are $\frac{2}{3}\delta$-optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$. Compute $(\rho_1', \ldots, \rho_k')$ from $(\rho_1, \ldots, \rho_k)$ as described in item 1 of Proposition 5.3.3. Return $(\rho_1', \ldots, \rho_k')$ and P as the δ-optimal point for $\lambda(\mathbf{A}, \mathbf{P})$.

Figure 5.1: An parallel oracle-algorithm for finding approximate solutions to $\lambda(\mathbf{A}, \mathbf{P})$ (Problem 5.1.2) used in the proof of Theorem 5.1.3.

obtain

(5.4.2)

$$\tilde{\lambda} = \frac{1}{T}\sum_{t=1}^{T}\left\langle \left(\rho_1^{(t)},\ldots,\rho_k^{(t)}\right), f_{R,\varepsilon}^{*}\left(P^{(t)},\Pi_1^{(t)},\ldots,\Pi_{k-1}^{(t)}\right)\right\rangle$$
$$\leq \left\langle (\rho_1^{\star},\ldots,\rho_k^{\star}), \frac{1}{T}\sum_{t=1}^{T} f_{R,\varepsilon}^{*}\left(P^{(t)},\Pi_1^{(t)},\ldots,\Pi_{k-1}^{(t)}\right)\right\rangle + \underbrace{\frac{2k^2}{\varepsilon}\left(\gamma+\frac{\ln(mn)}{\gamma T}\right)}_{\text{error term}}.$$

Substituting the choice of γ, T from step 1 we see that the error term on the right side is at most $\delta/3$. Since this inequality holds for any choice of $(\rho_1^{\star},\ldots,\rho_k^{\star})$ it certainly holds for the optimal choice, from which it follows that the right side is at most $\mu_\varepsilon(\mathbf{A},\mathbf{P})+\delta/3$. By construction each $(P^{(t)},\Pi_1^{(t)},\ldots,\Pi_{k-1}^{(t)})$ is a $\delta/3$-best response to $(\rho_1^{(t)},\ldots,\rho_k^{(t)})$ so it must be that the left side of this inequality is at least $\mu_\varepsilon(\mathbf{A},\mathbf{P})-\delta/3$. It then follows from Theorem 5.3.1 (Rounding theorem) and the choice $\varepsilon=\delta/3$ that $\left|\tilde{\lambda}-\lambda(\mathbf{A},\mathbf{P}))\right| < \frac{2}{3}\delta$ as desired.

Next we argue that the point $(\rho_1',\ldots,\rho_k')$ returned in step 4 is δ-optimal for $\lambda(\mathbf{A},\mathbf{P})$. By item 1 of Proposition 5.3.3 it suffices to argue that $(\rho_1,\ldots,\rho_k)$ is $\frac{2}{3}\delta$-optimal for $\mu_\varepsilon(\mathbf{A},\mathbf{P})$. To this end, choose any $(P^{\star},\Pi_1^{\star},\ldots,\Pi_a^{\star})$. Since each $(P^{(t)},\Pi_1^{(t)},\ldots,\Pi_{k-1}^{(t)})$ is a $\delta/3$-best response to $(\rho_1^{(t)},\ldots,\rho_k^{(t)})$ it holds that the inner product

$$\left\langle \left(\rho_1^{(t)},\ldots,\rho_k^{(t)}\right), f_{R,\varepsilon}^{*}\left(P^{(t)},\Pi_1^{(t)},\ldots,\Pi_{k-1}^{(t)}\right)\right\rangle$$

can increase by no more than $\delta/3$ when $(P^{\star},\Pi_1^{\star},\ldots,\Pi_{k-1}^{\star})$ is substituted for $(P^{(t)},\Pi_1^{(t)},\ldots,\Pi_{k-1}^{(t)})$. It then follows from (5.4.2) that

$$\left\langle \frac{1}{T}\sum_{t=1}^{T}\left(\rho_1^{(t)},\ldots,\rho_k^{(t)}\right), f_{R,\varepsilon}^{*}\left(P^{\star},\Pi_1^{\star},\ldots,\Pi_{k-1}^{\star}\right)\right\rangle \leq \tilde{\lambda}+\delta/3 \leq \mu_\varepsilon(\mathbf{A},\mathbf{P})+\tfrac{2}{3}\delta$$

and hence $(\rho_1,\ldots,\rho_k)$ is $\frac{2}{3}\delta$-optimal for $\mu_\varepsilon(\mathbf{A},\mathbf{P})$ as desired.

Next we argue that the operator P returned in step 4 is δ-optimal for $\lambda(\mathbf{A},\mathbf{P})$. By item 2 of Proposition 5.3.3 it suffices to argue that $(P,\Pi_1,\ldots,\Pi_{k-1})$ is $\frac{2}{3}\delta$-optimal for $\mu_\varepsilon(\mathbf{A},\mathbf{P})$.

To this end, choose any $(\rho_1^\star, \ldots, \rho_k^\star)$. It follows from (5.4.2) that

$$\left\langle (\rho_1^\star, \ldots, \rho_k^\star), f_{R,\varepsilon}^*\left(P, \Pi_1, \ldots, \Pi_{k-1}\right)\right\rangle \geq \tilde{\lambda} - \delta/3 \geq \mu_\varepsilon(\mathbf{A}, \mathbf{P}) - \tfrac{2}{3}\delta$$

and hence $(P, \Pi_1, \ldots, \Pi_{k-1})$ is $\frac{2}{3}\delta$-optimal for $\mu_\varepsilon(\mathbf{A}, \mathbf{P})$ as desired.

The efficiency of this algorithm is not difficult to argue. Each individual step consists only of matrix operations that are known to admit an efficient parallel implementation. Efficiency then follows from the observation that the number T of iterations is polynomial in k, $1/\delta$, and $\log(mn)$. □

5.5 Double Quantum Interactive Proofs

In this section we prove DQIP $\subseteq$ PSPACE by means of Theorem 5.1.3. Specifically, in Section 5.5.2 we argue that the verifier in a double quantum interactive proof induces a min-max problem of the form (5.1.2) in which elements of $\mathbf{A}$ correspond to strategies for the yes-prover, elements of $\mathbf{P}$ correspond to strategies for the no-prover, and the value $\lambda(\mathbf{A}, \mathbf{P})$ corresponds to the probability with which the verifier rejects when both provers act optimally.

Thus, the parallel oracle-algorithm of Theorem 5.1.3—together with a parallel implementation of the oracle for optimization over $\mathbf{P}$—can be used to compute this probability to sufficient accuracy so as to determine which prover has the winning strategy. In Section 5.5.3 we provide a parallel implementation of the oracle required by Theorem 5.1.3. Finally, in Section 5.5.4 we recite the argument by which the existence of a parallel algorithm for approximating $\lambda(\mathbf{A}, \mathbf{P})$ leads to the containment of DQIP inside PSPACE. First, we briefly introduce new notation in Section 5.5.1.

5.5.1 Notation

Until now we have used the symbol $\mathbb{M}_n$ to denote the space of complex $n \times n$ matrices. This notation is ideal when only one or two distinct quantum systems are under consideration. However, discussion henceforth deals with many different systems (called *registers*) and so we adopt the convention that distinct finite-dimensional complex vector spaces of the form $\mathbb{C}^d$ shall be denoted with calligraphic letters ($\mathcal{X}, \mathcal{Y}, \ldots$). We also adopt the following notation:

$\mathcal{XY}$	Shorthand for the Kronecker product $\mathcal{X} \otimes \mathcal{Y}$. If $\mathcal{X} = \mathbb{C}^d$ and $\mathcal{Y} = \mathbb{C}^{d'}$ then $\mathcal{XY} = \mathbb{C}^{dd'}$.
$\mathbb{M}_{\mathcal{X}}$	The complex space of all linear operators (matrices) acting on $\mathcal{X}$.
$I_{\mathcal{X}} \in \mathbb{M}_{\mathcal{X}}$	The identity operator acting on $\mathcal{X}$.
$\mathrm{Tr}_{\mathcal{X}} : \mathbb{M}_{\mathcal{XY}} \to \mathbb{M}_{\mathcal{Y}}$	The partial trace over $\mathcal{X}$.

5.5.2 Characterization of Strategies for the Yes-prover

The verifier in a double quantum interactive proof can be assumed to act upon two quantum registers: an m-qubit register M that is shared with the provers for the purpose of exchanging messages and a v-qubit register V that serves as a private memory for the verifier. Associated with the registers M, V are complex Euclidean spaces $\mathcal{M} = \mathbb{C}^{2^m}, \mathcal{V} = \mathbb{C}^{2^v}$, respectively. A verifier who exchanges a rounds of messages with the yes-prover followed by b rounds of messages with the no-prover is completely specified by a tuple $V = (|\psi\rangle, V_1, \ldots, V_{a+b-1}, \Pi)$ where

1. $|\psi\rangle \in \mathcal{MV}$ is a pure state.
2. $V_1, \ldots, V_{a+b-1} \in \mathbb{M}_{\mathcal{MV}}$ are unitary operators.
3. $\Pi \in \mathbb{M}_{\mathcal{MV}}$ is a projective measurement operator.

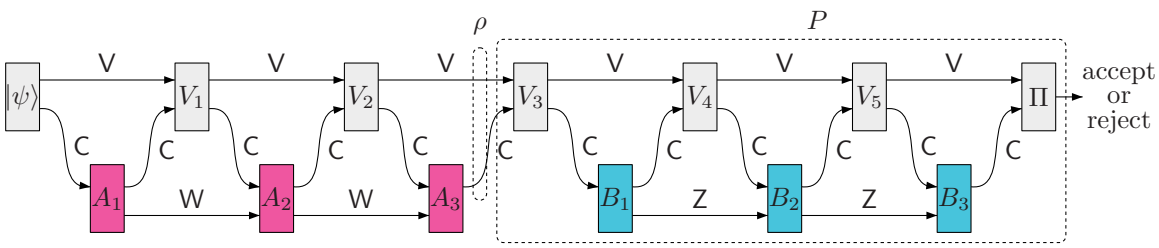

Figure 5.2: An illustration of a double quantum interactive proof in which the verifier $V = (|\psi\rangle, V_1, \ldots, V_5, \Pi)$ exchanges $a = 3$ rounds of messages with the yes-prover followed by $b = 3$ rounds of messages with the no-prover before performing the measurement $\{\Pi, I - \Pi\}$ that dictates acceptance or rejection. Any choice of (A_1, A_2, A_3) and (B_1, B_2, B_3) induces a state ρ and a measurement operator P as indicated. The probability of rejection is given by $\langle \rho, P \rangle = \mathrm{Tr}(\rho P)$.

The yes-prover acts upon the shared communication register M and a private memory register W with associated space $\mathcal{W}$. The actions of the yes-prover are specified by unitaries $A_1, \ldots, A_a \in \mathbb{M}_{\mathcal{M}\mathcal{W}}$. Similarly, the no-prover acts upon the shared communication register M and a private memory register Z with associated space $\mathcal{Z}$. The actions of the no-prover are specified by unitaries $B_1, \ldots, B_b \in \mathbb{M}_{\mathcal{M}\mathcal{Z}}$. The interaction proceeds as suggested by Figure 5.2 with measurement outcome Π indicating rejection.

Basic quantum formalism tells us that if the yes- and no-provers act according to $\vec{A} = (A_1, \ldots, A_a)$ and $\vec{B} = (B_1, \ldots, B_b)$, respectively, then the probability of rejection is given by

(5.5.1)

$$\Pr\left[\text{reject} \mid \vec{A}, \vec{B}\right] = \left\| \Pi B_b V_{a+b-1} B_{b-1} \cdots B_1 V_a A_a V_{a-1} A_{a-1} \cdots A_2 V_1 A_1 |\psi\rangle \right\|^2 .$$

(For clarity we have suppressed numerous tensors with identity and the initial states $|0\rangle$ of the provers' private memory registers.)

For any $\vec{A}$ let ρ be the reduced state of the verifier's registers (M, V) immediately after A_a is applied so that the actions of the yes-prover are completely represented by the state ρ.

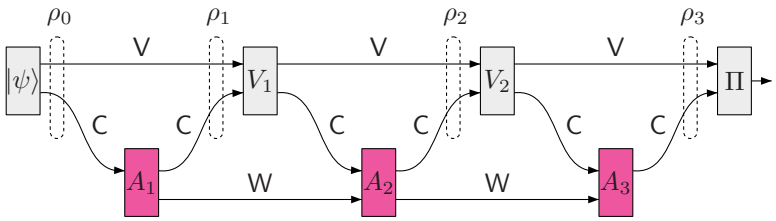

Figure 5.3: The states ρ_1, ρ_2, ρ_3 are a transcript of the referee's conversation with the yes-prover. It follows easily from the unitary equivalence of purifications that a triple (ρ_1, ρ_2, ρ_3) is a valid transcript if and only if it obeys the recursive relation $\mathrm{Tr}_{\mathcal{M}_i}(\rho_i) = \mathrm{Tr}_{\mathcal{A}_i}(V_{i-1}\rho_{i-1}V_{i-1}^*)$ for $i = 1, 2, 3$ where $V_0 = I$.

Similarly, for any $\vec{B}$ let P be the measurement operator on (M, V) obtained by bundling the verifier–no-prover interaction into a single measurement operator as suggested by Figure 5.2. The expression (5.5.1) for the probability of rejection can be rewritten in terms of ρ, P as

$$\Pr[\text{reject} \mid \vec{A}, \vec{B}] = \langle \rho, P \rangle.$$

By definition, the no-prover wishes to maximize this quantity while the yes-prover wishes to minimize it. Let $\lambda(V)$ denote the verifier's probability of rejection when both provers act optimally. For a verifier with completeness c and soundness s, or goal is to determine whether $\lambda(V)$ is closer to $1 - c$ or to $1 - s$.

Let $\mathbf{Y}(V) \subset \mathbb{M}_{\mathcal{MV}}$ denote the set of states of (M, V) obtainable by the yes-prover and let $\mathbf{P}(V) \subset \mathbb{M}_{\mathcal{MV}}$ denote the set of measurement operators on (M, V) obtainable by the no-prover. Then the desired quantity $\lambda(V)$ is given by the min-max problem

$$\lambda(V) = \min_{\rho \in \mathbf{Y}(V)} \max_{P \in \mathbf{P}(V)} \langle \rho, P \rangle. \tag{5.5.2}$$

What can be said of the sets $\mathbf{Y}(V), \mathbf{P}(V)$? Let us begin by considering the set $\mathbf{Y}(V)$. As suggested by Figure 5.3, each element of $\mathbf{Y}(V)$ can be viewed as the final entry ρ_a in a *transcript* $(\rho_1, \ldots, \rho_a)$ of the verifier's conversation with the yes-prover. Moreover, it is straightforward to use the unitary equivalence of purifications to characterize those

a-tuples of density matrices which constitute valid transcripts. This characterization was first noted by Kitaev [72].

Proposition 5.5.1 (Kitaev's consistency conditions [72])**.** *Let $V = (|\psi\rangle, V_1, \ldots, V_{a+b-1}, \Pi)$ be a verifier and let $\mathbf{Y}(V)$ be the set of admissible states for the yes-prover. A given state ρ is an element of $\mathbf{Y}(V)$ if and only if there exist density matrices $\rho_1, \ldots, \rho_a \in \mathbb{M}_{\mathcal{MV}}$ with $\rho_a = \rho$ and*

$$\mathrm{Tr}_{\mathcal{M}}(\rho_i) = \mathrm{Tr}_{\mathcal{M}}(V_{i-1}\rho_{i-1}V_{i-1}^*) \quad \textit{for } i = 1, \ldots, a$$

where we have written $V_0 = I$ and $\rho_0 = |\psi\rangle\langle\psi|$ for convenience.

With these observations in mind we consider completely positive and trace-preserving linear maps

$$\Phi_0, \ldots, \Phi_{a-1} : \mathbb{M}_{\mathcal{MV}} \to \mathbb{M}_{\mathcal{V}}$$

defined by

$$\Phi_0 : X \mapsto \mathrm{Tr}(X)\,\mathrm{Tr}_{\mathcal{M}}(|\psi\rangle\langle\psi|)$$

$$\Phi_i : X \mapsto \mathrm{Tr}_{\mathcal{M}}(V_i X V_i^*) \quad \text{for } i = 1, \ldots, a-1.$$

These maps specify the feasible region $\mathbf{A}(V)$ of an SDP of the form (5.1.1) from Section 5.1. Moreover, it follows from Kitaev's consistency conditions (Proposition 5.5.1) that $(\rho_0, \ldots, \rho_a) \in \mathbf{A}(V)$ if and only if $\rho_a \in \mathbf{Y}(V)$. Thus, the min-max problem (5.5.2) for $\lambda(V)$ can equivalently be written

$$\lambda(V) = \min_{(\rho_0,\ldots,\rho_a)\in\mathbf{A}(V)} \max_{P\in\mathbf{P}(V)} \langle \rho_a, P\rangle\,. \tag{5.5.3}$$

We have not yet shown that the set $\mathbf{P}(V)$ of measurement operators for the no-prover is compact and convex. But if we assume for the moment that it is then we may already apply Theorem 5.1.3 so as to obtain a parallel oracle-algorithm for approximating $\lambda(V)$ on input $\Phi_0, \ldots, \Phi_{a-1}$ given an oracle for optimization over $\mathbf{P}(V)$.

5.5.3 Implementation of the Oracle for Best Responses of the No-prover

In order to complete the description of our parallel algorithm for double quantum interactive proofs it remains only to describe the implementation of the oracle for optimization for $\mathbf{P}(V)$ (Problem 5.1.1). In this section we establish the following.

Proposition 5.5.2. *Let $V = (|\psi\rangle, V_1, \ldots, V_{a+b-1}, \Pi)$ be a verifier and let $\mathbf{P}(V)$ be the set of admissible measurement operators for the no-prover. There is a parallel algorithm for optimization over $\mathbf{P}(V)$ (Problem 5.1.1) with run time bounded by a polynomial in b, $1/\delta$, and $\log(\dim(\mathcal{MV}))$.*

It follows that the algorithm of Figure 5.1 yields an unconditionally efficient parallel algorithm for approximating $\lambda(V)$ given an explicit matrix representation of the verifier V.

As mentioned earlier, this instance of optimization over $\mathbf{P}(V)$ (Problem 5.1.1) will be rephrased as an SDP of the form (5.1.1) (plus some post-processing) so that the algorithm of Section 5.4 can be reused in the implementation of our oracle.

To this end choose any state $\rho \in \mathbb{M}_{\mathcal{MV}}$ and suppose that a (possibly cheating) yes-prover was somehow able to make it so that the registers (M, V) after the interaction with the yes-prover are in state ρ. Let W be a register large enough to admit a purification of ρ and let $|\varphi\rangle \in \mathcal{WMV}$ be any such purification. If the no-prover acts according to $(B_1, \ldots, B_b)$ then the probability of rejection (as per Eq. (5.5.1)) is

$$\Pr[\text{reject} \mid \rho, (B_1, \ldots, B_b)] = \|\Pi B_b V_{a+b-1} B_{b-1} \cdots B_1 V_a |\varphi\rangle\|^2.$$

Notice that this quantity also represents the probability of rejection in a different, single-prover interactive proof with a verifier V' whose initial state is $V_a|\varphi\rangle$. (Formally, the verifier V' exchanges b rounds of messages with one of the provers and zero messages with the other.) The unitaries $B_1, \ldots, B_b$ could specify actions for either the yes-prover

or the no-prover—a choice that depends only upon how we label the components of the verifier V'.

Since our goal is to reduce optimization over $\mathbf{P}(V)$ (which is a maximization problem) to an SDP of the form (5.1.1) (which is a minimization problem), it befits us to view $B_1, \ldots, B_b$ as actions for the yes-prover in the interactive proof with verifier V'. Let us write

$$V' = (V_a \left|\varphi\right\rangle, V_1', \ldots, V_{b-1}', \Pi')$$

where $V_1', \ldots, V_{b-1}', \Pi' \in \mathbb{M}_{\mathcal{MVW}}$ are given by

$$\begin{aligned} V_i' &= V_{a+i} \otimes I_{\mathcal{W}} \qquad \text{for } i = 1, \ldots, b-1 \\ \Pi' &= (I - \Pi) \otimes I_{\mathcal{W}}. \end{aligned}$$

The private memory register V' of the new verifier V' is identified with the registers (V, W) and communication register M' of the new verifier is identified with M.

Each choice of unitaries $(B_1, \ldots, B_b)$ induces both a measurement operator $P \in \mathbf{P}(V)$ and a state $\xi \in \mathbf{Y}(V')$ with

$$\langle \rho, P \rangle = \left\| \Pi B_b V_{a+b-1} B_{b-1} \cdots B_1 V_a \left|\varphi\right\rangle \right\|^2 = 1 - \left\langle \xi, \Pi' \right\rangle$$

and therefore

$$\max_{P \in \mathbf{P}(V)} \langle \rho, P \rangle = 1 - \lambda(V') = 1 - \min_{\xi \in \mathbf{Y}(V')} \left\langle \xi, \Pi' \right\rangle.$$

Moreover, $P \in \mathbf{P}(V)$ achieves the maximum on the left side if and only if the unitaries $(B_1, \ldots, B_b)$ that induce P also induce a state $\xi \in \mathbf{Y}(V')$ that achieves the minimum on the right side.

Incidentally, by identifying elements of $\mathbf{P}(V)$ with elements of $\mathbf{A}(V')$ we have established that the set $\mathbf{P}(V)$ is compact and convex as required by Theorem 5.1.3. We are now ready to prove Proposition 5.5.2.

Proof of Proposition 5.5.2. Consider the following algorithm for optimization over $\mathbf{P}(V)$:

1. Use the algorithm of Figure 5.1 to find $\xi \in \mathbf{Y}(V')$ minimizing $\langle \xi, \Pi' \rangle$.
2. Find the unitaries $(B_1, \ldots, B_b)$ that induce ξ. These unitaries also induce a measurement operator $P \in \mathbf{P}(V)$ maximizing $\langle \rho, P \rangle$. Compute P using $(B_1, \ldots, B_b)$ via standard matrix multiplication.

We already saw how the algorithm of Figure 5.1 can be used to accomplish step 1 given an oracle for optimization over $\mathbf{P}(V')$. In this case $\mathbf{P}(V') = \{\Pi'\}$ is a singleton set and thus the oracle for optimization over $\mathbf{P}(V')$ admits a trivial implementation by returning the only element.

It remains only to fill in the details for step 2. Recall that the algorithm of Figure 5.1 finds a near-optimal transcript $(\xi_0, \ldots, \xi_b) \in \mathbf{A}(V')$, meaning that

$$\begin{aligned} \mathrm{Tr}_{\mathcal{M}}(\xi_1) &= \mathrm{Tr}_{\mathcal{M}}(V_a \left|\varphi\right\rangle \left\langle\varphi\right| V_a^*) \\ \mathrm{Tr}_{\mathcal{M}}(\xi_{i+1}) &= \mathrm{Tr}_{\mathcal{M}}(V_i' \xi_i V_i'^*) \qquad \text{for each } i = 1, \ldots, b-1. \end{aligned}$$

(Here ξ_0 is an arbitrary density matrix that is not used in our construction. The presence of this matrix is an artifact of the identification of $\mathbf{Y}(V')$ with $\mathbf{A}(V')$.) The following algorithm finds the unitaries $(B_1, \ldots, B_b)$:

1. Let $\mathcal{Z}$ be a space large enough to admit purifications of $\xi_1, \ldots, \xi_b$. Write $\left|\alpha_0\right\rangle = \left|\varphi\right\rangle \left|0_{\mathcal{Z}}\right\rangle$ and $V_0' = V_a$.
2. For each $i = 1, \ldots, b$:
   (a) Compute a purification $\left|\alpha_i\right\rangle \in \mathcal{Z}\mathcal{M}\mathcal{V}\mathcal{W}$ of ξ_i.
   (b) Compute a unitary $B_i \in \mathbb{M}_{\mathcal{Z}\mathcal{M}}$ that maps $V_{i-1}' \left|\alpha_{i-1}\right\rangle$ to $\left|\alpha_i\right\rangle$.
3. Return the desired unitaries $(B_1, \ldots, B_b)$.

Correctness of this construction is straightforward (though notationally cumbersome). Let us argue that each individual step consists only of matrix operations that are known to admit an efficient parallel implementation, from which it follows that the entire construction is efficient.

Step 2a requires that we compute a purification $|\alpha\rangle$ of a given mixed state ξ. This can be achieved by computing a spectral decomposition

$$\xi = \sum_i \mu_i |\phi_i\rangle \langle\phi_i|$$

of ξ; the purification $|\alpha\rangle$ is then given by

$$|\alpha\rangle = \sum_i \sqrt{\mu_i} |\phi_i\rangle |\phi_i\rangle .$$

Given two pure states $|\alpha\rangle, |\alpha'\rangle \in \mathcal{Z}\mathcal{M}\mathcal{V}\mathcal{W}$ with

$$\mathrm{Tr}_{\mathcal{Z}\mathcal{M}}(|\alpha\rangle \langle\alpha|) = \mathrm{Tr}_{\mathcal{Z}\mathcal{M}}(|\alpha'\rangle \langle\alpha'|),$$

step 2b requires that we compute a unitary $B \in \mathbb{M}_{\mathcal{Z}\mathcal{M}}$ that maps $|\alpha\rangle$ to $|\alpha'\rangle$. This can be achieved by computing Schmidt decompositions

$$|\alpha\rangle = \sum_i s_i |\phi_i\rangle |\psi_i\rangle \qquad\qquad |\alpha'\rangle = \sum_i s'_i |\phi'_i\rangle |\psi_i\rangle$$

with respect to the partition $\mathcal{Z}\mathcal{M} \otimes \mathcal{V}\mathcal{W}$. (Schmidt decompositions on vectors are equivalent to singular value decompositions on matrices and hence can be implemented in parallel.) The desired unitary is then given by straightforward matrix multiplication and summation:

$$B = \sum_i |\phi'_i\rangle \langle\phi_i| .$$

□

5.5.4 Containment of DQIP inside PSPACE

The argument by which a parallel algorithm for double quantum interactive proofs leads to a proof of DQIP $\subseteq$ PSPACE is by now a familiar one. (See Section 3 of Ref. [60] for a good exposition of this type of argument.)

Proof of Theorem 5.1.4. For each decision problem $L \in$ DQIP we must prove that there is a polynomial space algorithm for L. To this end consider a "scaled up" version of NC known as $\mathrm{NC}(\mathrm{poly})$, which consists of all functions computable by polynomial-space uniform Boolean circuits of polynomial depth. It has long since been known that $\mathrm{NC}(\mathrm{poly})$ algorithms can be simulated in polynomial space [20], so in order to prove $L \in$ PSPACE it suffices to give an $\mathrm{NC}(\mathrm{poly})$ algorithm for L.

Let V be a verifier with completeness c, soundness s, and polynomial-bounded p with $c - s \geq 1/p$ witnessing the membership of L in DQIP. Let x be any input string and consider the following algorithm for deciding whether x is a yes-instance or a no-instance of L:

1. Compute the matrix representation of the verifier $V = (|\psi\rangle, V_1, \ldots, V_{a+b-1}, \Pi)$ on input x. As argued earlier, this representation specifies sets $\mathbf{A}(V), \mathbf{P}(V)$ for a min-max problem of the form (5.1.2).
2. Compute a δ-approximation of $\lambda(V)$ for the choice $\delta = (c-s)/3$ so as to determine which of the two provers has a winning strategy. Accept or reject accordingly.

The dimension $\dim(\mathcal{MV}) = 2^{m+v}$ of the matrix representation of a verifier on input x might grow exponentially in the bit length of x. Nevertheless, as argued in Ref. [60] for ordinary quantum interactive proofs, it is not difficult to see that step 1 admits a straightforward implementation in $\mathrm{NC}(\mathrm{poly})$ via standard matrix multiplication.

Earlier in this section we argued that the parallel oracle-algorithm of Theorem 5.1.3

can be used to compute the desired approximation of $\lambda(V)$. We also presented a parallel implementation of the oracle for optimization over $\mathbf{P}(V)$ required by Theorem 5.1.3. To see that this parallel algorithm is *efficient* it suffices to observe that the number of rounds $a+b$ and the inverse of the accuracy parameter $1/\delta$ both scale as a polynomial in $|x|$ and hence also in $\log(\dim(\mathcal{MV}))$.

Thus, the above algorithm computes the composition of a function in $\mathrm{NC}(\mathrm{poly})$ with another function in NC. As $\mathrm{NC}(\mathrm{poly})$ is closed under such compositions, it follows that the above algorithm admits an $\mathrm{NC}(\mathrm{poly})$ implementation and hence also a polynomial-space implementation. It follows that $L \in$ PSPACE and hence DQIP $\subseteq$ PSPACE. $\square$

5.6 Consequences and Extensions

5.6.1 A Direct Polynomial-space Simulation of QIP

As mentioned in the introduction, a special case of our result is a direct polynomial-space simulation of multi-message quantum interactive proofs, resulting in a first-principles proof of QIP $\subseteq$ PSPACE. Recall that an ordinary, single-prover quantum interactive proof is a double quantum interactive proof in which the verifier exchanges zero messages with the no-prover. We already observed in Section 5.5.3 that such a verifier induces an SDP of the form (5.1.1) in which elements of the feasible region $\mathbf{A}$ are identified with strategies for the prover. In this case, Theorem 5.1.3 yields an efficient parallel algorithm for finding optimal strategies for the prover in a single-prover quantum interactive proof with no need to specify an oracle.

5.6.2 Finding Near-optimal Strategies

The algorithm of Figure 5.1 not only approximates the value $\lambda(\mathbf{A},\mathbf{P})$ of the min-max problem (5.1.2), but it also finds near-optimal points $(\rho_1,\ldots,\rho_k) \in \mathbf{A}$ and $P \in \mathbf{P}$. By contrast, in Section 5.5 we were primarily concerned with the problem of approximating

only the value $\lambda(V)$ of the min-max problem (5.5.3). This quantity is the verifier's probability of rejection when both provers act optimally; approximating it suffices to prove DQIP $\subseteq$ PSPACE.

However, our result readily extends to the related search problem of *finding* near-optimal strategies for the provers. Indeed, step 4 of the algorithm of Figure 5.1 returns a transcript $(\rho_0, \ldots, \rho_a) \in \mathbf{A}(V)$ and a measurement operator $P \in \mathbf{P}(V)$, both of which are δ-optimal for $\lambda(V)$. The unitaries $(A_1, \ldots, A_a)$ for the yes-prover can be recovered from the transcript $(\rho_0, \ldots, \rho_a)$ via the method described in Section 5.5.3 with no additional complication.

It is only slightly more difficult to recover the no-prover's unitaries $(B_1, \ldots, B_b)$ from P. Our definition of Problem 5.1.1 (Optimization over $\mathbf{P}$) specifies only that a solution produce a near-optimal measurement operator $P \in \mathbf{P}$ for a given state ρ. But the algorithm for Problem 5.1.1 described in Section 5.5.3 for optimization over $\mathbf{P}(V)$ produces its output P by first constructing the associated unitaries $(B, \ldots, B_b)$. It is a simple matter to modify our definition of Problem 5.1.1 so as to also return those unitaries in addition to P.

The near-optimal measurement operator P returned in step 4 of the algorithm of Figure 5.1 is given by

$$P = \frac{1}{T} \sum_{t=1}^{T} P^{(t)},$$

which indicates a strategy for the no-prover that selects $t \in \{1, \ldots, T\}$ uniformly at random and then acts according to $(B_1^{(t)}, \ldots, B_b^{(t)})$. It is a simple matter to construct unitaries $(B_1, \ldots, B_b)$ that implement this probabilistic strategy by sampling the integer t during the first round, recording that integer in the no-prover's private memory (which must be enlarged slightly to make room for it), and controlling the operation in subsequent turns on the contents of that integer. All of the matrix operations required to construct $(B_1, \ldots, B_b)$

from each $(B_1^{(t)}, \ldots, B_b^{(t)})$ in this way can be implemented efficiently in parallel.

5.6.3 Robustness with Respect to Error

In Section 5.1.2 we noted that it is not immediately obvious that the classes DIP and DQIP are robust with respect to completeness and soundness parameters c, s. Because of this we defined the classes to be inclusive as possible, allowing any verifier for which $c - s \geq 1/p$ for some polynomial-bounded function $p(|x|)$.

Nevertheless, it follows from the collapse of these classes to PSPACE that they are indeed robust with respect to completeness and soundness. In particular, classical interactive proofs for PSPACE [86, 100] imply that if a decision problem L admits a double (quantum) interactive proof with $c - s \geq 1/p$ then L also admits a double (quantum) interactive proof with $c = 1$ and $s \leq 2^{-q}$ for any desired polynomial-bounded function $q(|x|)$.

However, the method by which the original verifier is transformed into the low-error verifier is very circuitous: the original verifier must be simulated in polynomial space according to Theorem 5.1.4 and then that polynomial-space computation must be converted back into an interactive proof with perfect completeness and exponentially small soundness according to proofs of IP = PSPACE. It would be nice to know whether a more straightforward transformation such as parallel repetition followed by a majority vote could be used to reduce error for double quantum interactive proofs and other bounded-turn interactive proofs with competing provers.

5.6.4 Arbitrary Payoff Observables

In the study of interactive proofs attention is generally restricted to the *accept-reject* model wherein the verifier's measurement $\{\Pi, I - \Pi\}$ indicates only acceptance or rejection without specifying a payout to the provers. From a game-theoretic perspective, one might wish to consider a more general verifier whose final measurement $\{\Pi_a\}_{a \in \Sigma}$ could have

outcomes belonging to some arbitrary finite set Σ. In this case, the verifier awards *payouts* to the provers according to a *payout function* $v : \Sigma \to \mathbb{R}$ where $v(a)$ denotes the payout to the yes-prover in the event of outcome a. (Since the game is zero-sum, the no-prover's payout must be $-v(a)$.)

Jain and Watrous describe a simple transformation by which their algorithm for one-turn quantum games can be used to approximate the expected payout in this more general setting [62]. Their transformation extends without complication to double quantum interactive proofs.

In our case, the expected payout to the yes-prover when she and the no-prover play according to $(A_1, \ldots, A_a)$ and $(B_1, \ldots, B_b)$, respectively, is given by

$$\sum_{a \in \Sigma} v(a) \langle \phi | \Pi_a | \phi \rangle = \langle \phi | \Pi_\Sigma | \phi \rangle$$

where

$$|\phi\rangle = B_b V_{a+b-1} B_{b-1} \cdots B_1 V_a A_a V_{a-1} A_{a-1} \cdots A_2 V_1 A_1 |\psi\rangle$$

is the final state of the system and the Hermitian operator $\Pi_\Sigma = \sum_{a \in \Sigma} v(a) \Pi_a$ denotes the *payout observable* induced by the verifier. The expected payout of this interaction can be computed simply by translating and rescaling Π_Σ so as to obtain a measurement operator $0 \preceq \Pi \preceq I$ and then running our algorithm for double quantum interactive proofs with verifier $V = (|\psi\rangle, V_1, \ldots, V_{a+b-1}, \Pi)$. The expected payout of the original protocol is then obtained by inverting the scaling and translation operations by which Π was obtained from Π_Σ. As noted by Jain and Watrous, this transformation has the effect of inflating the additive approximation error δ by a factor of $\| \Pi_\Sigma \|$, which is the maximum absolute value of any given payout.

CHAPTER 6

Quantum Merlin-Arthur Games: Multiple Provers

This chapter is based on a joint work with Yaoyun Shi [102].

In this chapter we provide a PSPACE upper bound for a variant of QMA(2) that allows the verifier to perform restricted but entangled measurements over the two proofs. We introduced QMA(2) in Section 1.2.4 that exhibits an intriguing quantum phenomenon that the enforcement of absence of entanglement might increase its computational power. It is, however, a challenging problem to beat the trivial upper bound of QMA(2). In the following, we demonstrate how a clever use of enumeration through epsilon-nets can lead to a PSPACE upper bound for a variant of QMA(2) that is the most general one considered up to date. Our results are also applications of the equilibrium value method.

6.1 Background and Main Results

Entanglement is an essential ingredient in many ingenious applications of quantum information processing. Understanding and exploiting entanglement remains a central theme in quantum information processing research [57]. Denote by $\mathrm{SepD}\left(\mathcal{A}_1 \otimes \mathcal{A}_2\right)$ the set of separable density operators over the space $\mathcal{A}_1 \otimes \mathcal{A}_2$. The *weak membership* problem for separability that is to decide, given a classical description of $\rho \in \mathrm{SepD}\left(\mathcal{A}_1 \otimes \mathcal{A}_2\right)$, whether the state ρ is inside or ϵ far away in trace distance from $\mathrm{SepD}\left(\mathcal{A}_1 \otimes \mathcal{A}_2\right)$, turns

out to be NP-hard when ϵ is inverse exponential [50] (or even inverse polynomial [58, 46]) in the dimension of $\mathcal{A}_1 \otimes \mathcal{A}_2$. In this chapter we study a closely related problem, namely the linear optimization problem over separable states below.

Problem 6.1.1. *Given a Hermitian matrix Q over $\mathcal{A}_1 \otimes \mathcal{A}_2$ (of dimension $d \times d$), compute the optimum value, denoted by OptSep(Q), of the optimization problem*

$$\max \langle Q, X \rangle \text{ subject to } X \in \mathrm{SepD}\left(\mathcal{A}_1 \otimes \mathcal{A}_2\right).$$

It is a standard fact in convex optimization [49, Figure 4.1] that the weak membership problem and the weak linear optimization, a special case of Problem 6.1.1, over a certain convex set, such as $\mathrm{SepD}\left(\mathcal{A}_1 \otimes \mathcal{A}_2\right)$, are equivalent up to a polynomial loss in precision and a polynomial-time overhead. Thus it is NP-hard to compute OptSep(Q) with an inverse polynomial additive error.

Besides the connection mentioned above, Problem 6.1.1 can also be understood from various aspects. Firstly, Problem 6.1.1 can be viewed as finding the minimum energy of some physical system that is achieved by separable states. Secondly, in the study of the tensor product space [34], the value OptSep(Q) is precisely the *injective norm* of Q in $\mathcal{L}(\mathcal{A}_1) \otimes \mathcal{L}(\mathcal{A}_2)$, where $\mathcal{L}(\mathcal{A})$ denotes the Banach space of operators on $\mathcal{A}$ with the operator norm. Finally, one may be equally motivated from the study in operations research (e.g.,"Bi-Quadratic Optimization over Unit Spheres" [24]).

Another motivation to study Problem 6.1.1 is due to the recent interest about the complexity class QMA(2). While the extension of NP to allow multiple provers trivially reduces to NP itself, the power of QMA(2), the extension of QMA with multiple *unentangled* provers, remains far from being well understood. The study of the multiple-prover model was initiated in [78], in which QMA(k) denotes the complexity class for the k-prover case. Much attention was attracted to this model because of the surprising discovery that NP ad-

mits *logarithmic* size unentangled quantum proofs [18], comparing with the fact that single prover quantum logarithmic size proofs only characterize BQP [87]. It seems adding one unentangled prover increases the power of the model substantially. There are several subsequent works on refining the initial protocol either with improved completeness and soundness [12, 2, 27, 38], or with less powerful verifiers [26]. Recently it was proved that QMA(2)=QMA(poly) [55] by using the so-called *product test* protocol that determines whether a multipartite pure state is a product state when two copies of it are given. Also, variants of QMA(2), such as BellQMA and LOCCQMA with restricted verifiers that perform only nonadaptive or adaptive local measurements respectively, were defined in [2] and studied in [21, 22].

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

Our contributions. In this chapter we provide efficient algorithms for Problem 6.1.1 in either time or space for several Qs of interest. Our idea is to enumerate via epsilon-nets more "cleverly" with the help of certain structures of Q.

Now we briefly describe our strategy of obtaining space-efficient algorithms. When the total number of points to enumerate is not large, one can represent, and hence enumerate each point in polynomial space. If the additional computation for each point can also be done in polynomial space, one immediately gets a polynomial-space implementation for the whole algorithm by composing those two components naturally. We make use of the relation NC(poly)=PSPACE [20] to obtain space-efficient implementation for the

additional computation, which in our case basically includes the following two parts. The first part assures that the enumeration procedure functions correctly because these epsilon-nets of interest are not standard. This part turns out to be a simple application of our equilibrium value method to compute a min-max form. The second part only contains fundamental matrix operations, which usually admit efficient parallel algorithms [108]. As a result, both parts of the additional computation admit efficient parallel algorithms, and therefore can be implemented in polynomial space.

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

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

Theorem 6.1.2 (Informal. See Section 6.3)**.** *Given any Hermitian Q (of dimension d) and its decomposition $Q = \sum_{i=1}^{M} Q_i^1 \otimes \cdots \otimes Q_i^k$, $OptSep(Q)$ can be approximated with additive error δ in quasi-polynomial time*[1] *in d and $1/\delta$ if kM is $O(poly\text{-}log(d))$.*

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

[1] Quasi-polynomial time is upper bounded by $2^{O((\log n)^c)}$ for some fixed c, where n is the input size.

Corollary 6.1.3 (Informal. See Section 6.3)**.** *If $kM/\delta \in O(poly(n))$, the quantity $OptSep(Q)$ can be approximated with additive error δ in* PSPACE*.*

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

Corollary 6.1.4. QMA(2)$[\mathrm{poly}(n), O(\log(n))] \subseteq$ PSPACE*.*

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

We also study Problem 6.1.1 when Q is a local Hamiltonian over k parties. Recall that a promise version of this problem in the one party case, namely the *local-Hamiltonian problem*, is QMA-complete [73]. Our definition extends the original local Hamiltonian problem to its k-partite version, which, however, is no longer necessarily QMA(k)-complete. Indeed, our result supports this fact in the algorithmic aspect. An independent work of Chailloux and Sattath [25], which complements our result, shows that the 2-partite local Hamiltonian problem defined above lies in QMA.

Corollary 6.1.5 (Informal. See Section 6.5)**.** *Given some local Hamiltonian Q over k parties, $OptSep(Q)$ can be approximated with additive error δ in quasi-polynomial time in $d, 1/\delta$; the k-partite local Hamiltonian problem is in* PSPACE*.*

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

Theorem 6.1.6. *For $Q \geq 0$, OptSep(Q) can be approximated with additive error δ in time* $\exp(O(\log(d) + \delta^{-2}\|Q\|_F^2 \ln(\|Q\|_F/\delta)))$.

A similar running time $\exp(O(\log^2(d)\delta^{-2}\|Q\|_{\mathrm{F}}^2))$ was obtained in [22] using some advanced results (i.e., the semidefinite programming for finding symmetric extension [37] and an improved quantum de Finetti-type bound) in quantum information theory. In contrast, our algorithm only uses fundamental operations of matrices and epsilon-nets. To approximate with precision δ, it suffices to consider the eigenspace of Q of eigenvalues greater than δ whose dimension is bounded by $\|Q\|_{\mathrm{F}}^2/\delta^2$. Nevertheless, naively enumerating density operators over that subspace does not work since one cannot detect the separability of those density operators. We circumvent this difficulty by making nontrivial use of the Schmidt decomposition of bipartite pure states.

We note, however, that other results in [22] do not follow from our algorithm, and our method cannot be seen as a replacement of the kernel technique therein. Furthermore, our method does not extend to the k-partite case, as there is no Schmidt decomposition in that case.

The rest of this chapter is organized as follows. The necessary background knowledge on the epsilon-nets is introduced in Section 6.2. Our main algorithm based on the decomposability of Q is illustrated in Section 6.3, which is followed by the simulation of variants of QMA(2) in Section 6.4 and the local Hamiltonian case in Section 6.5. We conclude with the demonstration of an algorithm with running time exponential in $\|Q\|_{\mathrm{F}}$ for Problem 6.1.1 in Section 6.6.

6.2 Epsilon Net

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

Definition 6.2.1 (ϵ-net)**.** Let (X, d) [2] be any metric space and let $\epsilon > 0$. A subset $\mathcal{N}_\epsilon$ is called an ϵ-net of X if for each $x \in X$, there exists $y \in \mathcal{N}_\epsilon$ with $d(x, y) \leq \epsilon$.

Now we turn to the particular ϵ-net in this chapter. Let $\mathcal{H}$ be any Hilbert space of dimension d and $\mathcal{Q} = \mathcal{Q}(M, w) = (Q_1, Q_2, \cdots Q_M)$ be a sequence of operators on $\mathcal{H}$ with $\|Q_i\|_\infty \leq w$, for all i. Define the $\mathcal{Q}$-space, denoted by $\mathrm{SP}(\mathcal{Q})$, as

$$\mathrm{SP}(\mathcal{Q}) = \{(\langle Q_1, \rho\rangle, \langle Q_2, \rho\rangle, \cdots, \langle Q_M, \rho\rangle) : \rho \in \mathrm{D}(\mathcal{H})\} \subseteq \mathbb{C}^M.$$

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

Selection process

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

$$\mathrm{dis}(\vec{p}) = \min_{\vec{q} \in \mathrm{SP}(\mathcal{Q})} \|\vec{p} - \vec{q}\|_1.$$

The distance $\mathrm{dis}(\vec{p})$ can be efficiently computed in time by casting the problem as a semidefinite program, e.g., the following one:

$$\text{(6.2.1)} \qquad \text{min:} \quad \sum_{i=1}^{M} t_i,$$

$$\text{(6.2.2)} \qquad \text{subject to:} \quad \begin{pmatrix} t_i & \vec{p}_i - \vec{q}(\rho)_i \\ \vec{p}_i^* - \vec{q}(\rho)_i^* & t_i \end{pmatrix} \geq 0, \quad \forall i = 1, \cdots, M.$$

$$\rho \in \mathrm{D}(\mathcal{H}).$$

(6.2.3)

[2] We will abuse the notation later where the metric d is replaced by the norm from which the metric is induced.

The correctness of the SDP comes from that the positive semidefinite constraint in Eq. (6.2.2) implies $t_i^2 \geq \|\vec{p}_i - \vec{q}(\rho)_i\|^2$ for each i=1,...,M. For any fixed ρ, the minimization over t_is will give $\|\vec{p} - \vec{q}(\rho)\|_1$. And then the minimization over ρ gives the desired answer.

Although semidefinite programs admit polynomial time solutions (when Q has a concise description, then this corresponds to exponential time solutions), it is generally unknown whether these polynomial time solutions can also be made space-efficient (i.e., in poly-logarithmic space). In our case where Q has a concise description, space-efficient solution corresponds to a PSPACE upper bound. Thus we need to develop our own space-efficient algorithm for this problem. Due to the duality of the ℓ_1 norm, one has

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

where $\vec{q}(\rho) = (\langle Q_1, \rho \rangle, \langle Q_2, \rho \rangle, \cdots, \langle Q_M, \rho \rangle) \in \mathbb{C}^M$. By rephrasing $\mathrm{dis}(\vec{p})$ in the above form, one shows the quantity $\mathrm{dis}(\vec{p})$ is actually an equilibrium value. This follows from the well-known extensions of von' Neumann's Min-Max Theorem [107, 39]. One can easily verify that the density operator set $\mathrm{D}(\mathcal{H})$ and the unit ball of $\mathbb{C}^M$ under ℓ_∞ norm are convex and compact and the objective function is a bilinear form over the two sets.

$$\min_{\rho \in \mathrm{D}(\mathcal{H})} \max_{\vec{z} \in \mathbf{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \mathrm{Re}\, \langle \vec{p} - \vec{q}(\rho), \vec{z} \rangle = \max_{\vec{z} \in \mathbf{B}(\mathbb{C}^M, \|\cdot\|_\infty)} \min_{\rho \in \mathrm{D}(\mathcal{H})} \mathrm{Re}\, \langle \vec{p} - \vec{q}(\rho), \vec{z} \rangle . \tag{6.2.4}$$

Fortunately, there is an efficient algorithm in both time and space (in terms of $d, M, w, 1/\epsilon$) to approximate $\mathrm{dis}(\vec{p})$ with additive error ϵ. This is a simple application of our equilibrium value method in Chapter 3.

Lemma 6.2.2. *Given any point $\vec{p} \in$ Raw-(M, w) and $\epsilon > 0$, there is an algorithm that approximates $dis(\vec{p})$ with additive error ϵ in $poly(d, M, w, 1/\epsilon)$ time. Furthermore, if d is considered as the input size and $M, w, 1/\epsilon \in O(poly\text{-}log(d))$, this algorithm is also efficient in parallel, namely, it is inside* NC.

Construction of the ϵ-net

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

- Construct the ϵ-net of the set Raw-(M,w) with the metric induced from the ℓ_1 norm. Denote such an ϵ-net by $\mathcal{R}_\epsilon$.
- For each point $\vec{p} \in \mathcal{R}_\epsilon$, determine $\mathrm{dis}(\vec{p})$ and select it to $\mathcal{N}_\epsilon$ if $\mathrm{dis}(\vec{p}) \leq \epsilon$. We claim $\mathcal{N}_\epsilon$ is the ϵ-net of $(\mathrm{SP}(\mathcal{Q}), \ell_1)$.

The construction for the first step is rather routine. Creating an ϵ'-net T'_ϵ over a bounded complex region $\{z \in \mathbb{C} : \|z\| \leq w\}$ is simple: we can place a 2D grid over the complex plane to cover the disk $\|z\| \leq w$. Simple argument shows $|T'_\epsilon| \in O(\frac{w^2}{\epsilon'^2})$. Then $\mathcal{R}_\epsilon$ can be obtained by cross-producting T'_ϵ for M times. To ensure the closeness in the ℓ_1 norm, we will choose $\epsilon' = \epsilon/M$.

Theorem 6.2.3. *The $\mathcal{N}_\epsilon$ constructed above is indeed an ϵ-net of $(\mathrm{SP}(\mathcal{Q}), \ell_1)$ with cardinality at most $O((\frac{w^2M^2}{\epsilon^2})^M)$. For any point $\vec{n} \in \mathcal{N}_\epsilon$, we have $dis(\vec{n}) \leq \epsilon$.*

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

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

Remarks. If one chooses $\mathcal{Q}$ to be $\mathcal{Q}(d^2,1) = \{|i\rangle\langle j| : i,j = 1,\cdots,d\}$, one can generate the ϵ-net of the density operator set with ℓ_1 norm in the method described above. It is akin to generating an ϵ-net for every entry of the density operator. At the other extreme, one can

also efficiently generate the ϵ-net of a small size $\mathrm{SP}(\mathcal{Q})$ even when the space dimension d is relatively large.

6.3 The Main Algorithm

Without loss of generality, we assume $\mathcal{A}_1, \mathcal{A}_2$ are identical, and of dimension d in Problem 6.1.1. Moreover, our algorithm will deal with the set of product states rather than separable states. Namely, we consider the following problem.

$$\text{(6.3.1)} \qquad \begin{aligned} \text{max:} \quad & \langle Q, \rho \rangle , \\ \text{subject to:} \quad & \rho = \rho_1 \otimes \rho_2, \rho_1 \in \mathrm{D}(\mathcal{A}_1), \rho_2 \in \mathrm{D}(\mathcal{A}_2). \end{aligned}$$

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

Problem 6.3.1 (k-partite version)**.** *Given any Hermitian matrix Q over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ ($k \geq 2$), compute the optimum value $\mathit{OptSep}(Q)$ with additive error δ.*

$$\text{(6.3.2)} \qquad \begin{aligned} \textit{max:} \quad & \langle Q, \rho \rangle , \\ \textit{subject to:} \quad & \rho = \rho_1 \otimes \cdots \otimes \rho_k, \forall i, \rho_i \in \mathrm{D}(\mathcal{A}_i). \end{aligned}$$

Before describing the algorithm, we need some terminology about the *decomposability* of a multi-partite operator. Any Hermitian operator Q is called *M-decomposable* if there exists $(Q_1^t, Q_2^t, \cdots, Q_M^t) \in \mathrm{L}(\mathcal{A}_t)^M$ for each t such that

$$Q = \sum_{i=1}^{M} Q_i^1 \otimes Q_i^2 \otimes \cdots \otimes Q_i^{k-1} \otimes Q_i^k.$$

To facilitate the use of ϵ-net, we adopt a slight variation of the decomposability above. Let $\vec{w} \in \mathbb{R}_+^k$ denote the widths of operators over each $\mathcal{A}_i$. Any Q is called $(M, \vec{w})$-

1. Let $\mathcal{Q}_t(M, w_t) = (Q_1^t, Q_2^t, \cdots, Q_M^t)$ for t=1,..., k-1. Let $W = \Pi_{i=1}^k w_i$. Generate the ϵ_t-net (by Theorem 6.2.3) of $(\mathrm{SP}(\mathcal{Q}_t), \ell_1)$ for each t=1,..., k-1 with $\epsilon_t = w_t\delta/(k-1)W$ and denote such a set by $\mathcal{N}_{\epsilon_t}^t$. Also let OPT store the optimum value.
2. For each point $\vec{q} = (\vec{q}^1, \vec{q}^2, \cdots \vec{q}^{k-1}) \in \mathcal{N}_{\epsilon_1}^1 \times \mathcal{N}_{\epsilon_2}^2 \times \cdots \times \mathcal{N}_{\epsilon_{k-1}}^{k-1}$, let Q^k be

$$Q^k = \sum_{i=1}^{M} q_i^1 q_i^2 \cdots q_i^{k-1} Q_i^k,$$

   and calculate $\tilde{Q}^k = \frac{1}{2}(Q^k + Q^{k*})$. Then compute the maximum eigenvalue of $\tilde{Q}^k$, denoted by $\lambda_{\max}(\vec{q})$. Update OPT as follows: $\mathrm{OPT} = \max\{\mathrm{OPT}, \lambda_{\max}(\vec{q})\}$.
3. Return OPT.

Figure 6.1: The main algorithm with precision δ.

decomposable if Q is M-decomposable and the widths of those operators in the decomposition are bounded in the sense that $\max_i \left\| Q_i^t \right\|_\infty \leq w_t$ for each t. It is noteworthy to mention that the decomposability defined above is related to the concept tensor rank. However, given the representation Q as input, it is hard in general to compute the tensor rank of Q or its corresponding decomposition. Therefore, for any $(M, \vec{w})$-decomposable Q we assume its corresponding decomposition is also a part of the input to our algorithm.

Theorem 6.3.2. *Let Q be some $(M, \vec{w})$-decomposable Hermitian over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ (each $\mathcal{A}_i$ is of dimension d) and $\delta > 0$. Also let $(Q_1^t, Q_2^t, \cdots, Q_M^t)$,t=1,..., k be the operators in the corresponding decomposition of Q. The algorithm shown in Fig. 6.1 approximates $\mathrm{OptSep}(Q)$ of Problem 6.3.1 with additive error δ in $O((\frac{(k-1)^2W^2M^2}{\delta^2})^{(k-1)M}) \times \mathrm{poly}(d, M, k, W, 1/\delta)$ time where $W = \Pi_{i=1}^k w_i$.*

Proof. By substituting the identity $Q = \sum_{i=1}^M Q_i^1 \otimes Q_i^2 \otimes \cdots \otimes Q_i^{k-1}$, the optimization problem becomes,

$$\text{max:} \quad \left\langle \sum_{i=1}^{M} p_i^1 p_i^2 \cdots p_i^{k-1} Q_i^k, \rho_k \right\rangle$$

$$\text{subject to:} \quad \forall t \in \{1, \cdots, k-1\}, \vec{p}_t \in \mathrm{SP}(\mathcal{Q}_t(M, w_t)), \text{ and } \rho_k \in \mathrm{D}(\mathcal{A}_k).$$

Thus, solving the optimization problem amounts to first enumerating $\vec{p}_t \in \mathrm{SP}(\mathcal{Q}_t(M, w_1))$ for each t, and then solving the optimization problem over $\mathrm{D}(\mathcal{A}_k)$.

Consider any point $\vec{p} = (\vec{p}^1, \vec{p}^2, \cdots, \vec{p}^{k-1}) \in \mathrm{SP}(Q_i)^{k-1}$ where $\mathrm{SP}(Q_i)^{k-1}$ denotes $\mathrm{SP}(\mathcal{Q}_1) \times \cdots \times \mathrm{SP}(\mathcal{Q}_{k-1})$. Due to Theorem 6.2.3, there is at least one point $\vec{q} = (\vec{q}^1, \vec{q}^2, \cdots \vec{q}^{k-1}) \in \{\mathcal{N}^i_{\epsilon_i}\}^{k-1}$ where $\{\mathcal{N}^i_{\epsilon_i}\}^{k-1}$ denotes $\mathcal{N}^1_{\epsilon_1} \times \mathcal{N}^2_{\epsilon_2} \times \cdots \times \mathcal{N}^{k-1}_{\epsilon_{k-1}}$ such that $\| \vec{q}^t - \vec{p}^t \|_1 \leq \epsilon_t$ for t=1,..,k-1. The choice of $\tilde{Q}^k$ is to symmetrize Q^k. With $\tilde{Q}^k$ being Hermitian, it is clear that $\lambda_{\max}(\vec{q}) = \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \langle \tilde{Q}^k, \rho_k \rangle$. Now let's analyze how much error will be induced in this process.

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

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

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

$$\begin{aligned} \left\| P^k - Q^k \right\|_\infty &= \left\| \sum_{i=1}^M (p_i^1 p_i^2 \cdots p_i^{k-1} - q_i^1 q_i^2 \cdots q_i^{k-1}) Q_i^k \right\|_\infty \\ &= \left\| \sum_{i=1}^M \sum_{t=1}^{k-1} (q_i^1 \cdots q_i^{t-1} p_i^t p_i^{t+1} \cdots p_i^{k-1} - q_i^1 \cdots q_i^{t-1} q_i^t p_i^{t+1} \cdots p_i^{k-1}) Q_i^k \right\|_\infty , \end{aligned}$$

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

$$\sum_{i=1}^M |p_i^1 - q_i^1| |p_i^2 \cdots p_i^{k-1}| \left\| Q_i^k \right\|_\infty , , \cdots , \sum_{i=1}^M |q_i^1 \cdots q_i^{k-2}| |p_i^{k-1} - q_i^{k-1}| \left\| Q_i^k \right\|_\infty .$$

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

$$\left\| \tilde{P}^k - \tilde{Q}^k \right\|_\infty \leq \epsilon_1 W/w_1 + \epsilon_2 W/w_2 + \cdots + \epsilon_{k-1} W/w_{k-1} = \underbrace{\frac{\delta}{k-1} + \cdots + \frac{\delta}{k-1}}_{\text{k-1 terms}} = \delta.$$

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

$$\max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{P}^k, \rho_k \right\rangle = \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{Q}^k, \rho_k \right\rangle + \left\langle \tilde{P}^k - \tilde{Q}^k, \rho_k \right\rangle .$$

By Hölder Inequalities we have $|\left\langle \tilde{P}^k - \tilde{Q}^k, \rho_k \right\rangle| \leq \left\| \tilde{P}^k - \tilde{Q}^k \right\|_\infty \|\rho_k\|_{\mathrm{Tr}} \leq \delta$,

$$\lambda_{\max}(\vec{q}) - \delta \leq \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{P}^k(\vec{p}), \rho_k \right\rangle \leq \lambda_{\max}(\vec{q}) + \delta.$$

We now optimize $\vec{p}$ over $\mathrm{SP}(\mathcal{Q}_i)^{k-1}$ and the corresponding $\vec{q}$ will run over the ϵ-net $\{\mathcal{N}_{\epsilon_i}^i\}^{k-1}$. As every point $\vec{q} \in \{\mathcal{N}_{\epsilon_i}^i\}^{k-1}$ is also close to $\mathrm{SP}(\mathcal{Q}_i)^{k-1}$ in the sense that $\mathrm{dis}(\vec{q}^t) \leq \epsilon_t$ for each t, we have

$$\max_{\vec{q} \in \{\mathcal{N}_{\epsilon_i}^i\}^{k-1}} \lambda_{\max}(\vec{q}) - \delta \leq \max_{\vec{p} \in \mathrm{SP}(\mathcal{Q}_i)^{k-1}} \max_{\rho_k \in \mathrm{D}(\mathcal{A}_k)} \left\langle \tilde{P}^k(\vec{p}), \rho_k \right\rangle \leq \max_{\vec{q} \in \{\mathcal{N}_{\epsilon_i}^i\}^{k-1}} \lambda_{\max}(\vec{q}) + \delta.$$

Finally, it is not hard to see that $\mathrm{OPT} = \max_{\vec{q} \in \{\mathcal{N}_{\epsilon_i}^i\}^{k-1}} \lambda_{\max}(\vec{q})$ and therefore

$$\mathrm{OPT} - \delta \leq \mathrm{OptSep}(Q) \leq \mathrm{OPT} + \delta.$$

Now let us analyze the efficiency of this algorithm. The total number of points in the ϵ-net $\{\mathcal{N}_{\epsilon_i}^i\}^{k-1}$ is upper bounded by $O((\frac{(k-1)^2W^2M^2}{\delta^2})^{(k-1)M})$. The generation of each point $\vec{q}$ will cost time polynomial in $d, M, W, 1/\delta$ (See Lemma 6.2.2.). Afterwards, one needs to calculate $\tilde{Q}^k$ and its maximum eigenvalue for each point, which can be done in time polynomial in d, k, M. Thus, the total running time is bounded by $O((\frac{(k-1)^2W^2M^2}{\delta^2})^{(k-1)M}) \times \mathrm{poly}(d, M, k, W, 1/\delta)$. □

Remarks. All operations in the algorithm described in Fig. 6.1 can be implemented efficiently in parallel in some situation. This is because fundamental operations of matrices can be done in NC and the calculation of $\mathrm{dis}(\vec{p})$ can be done in NC (See Lemma 6.2.2) when $M, W, k, 1/\delta$ are in nice forms of d.

Corollary 6.3.3. *Let n be the input size such that* $d = \exp(\mathrm{poly(n)})$, *if* $W/\delta \in O(\mathrm{poly}(n))$, $kM \in O(\mathrm{poly(n)})$, *then OptSep*$(Q)$ *can be approximated with additive error* δ *in* PSPACE.

Proof. Given Q and its decomposition, consider the following algorithm

1. Enumerate each point $\vec{p} = (\vec{p}_1, \cdots, \vec{p}_{k-1})$ in the raw set $\mathcal{R}_{\epsilon_1}^1 \times \cdots \times \mathcal{R}_{\epsilon_{k-1}}^{k-1}$.

2. Compute dis$(\vec{p}_t)$ for each t=1,...,k-1. If $\vec{p}$ is a valid point in the epsilon-net, then we continue with the rest part in Step 2 of the algorithm in Fig. 6.1.

3. Compare the values obtained by each point $\vec{p}$ and keep the optimum one.

Given the condition $W/\delta \in O(\text{poly}(n)), kM \in O(\text{poly(n)})$, the first part of the algorithm can be done in polynomial space. This is because in this case each point in the raw set can be represented in polynomial space and therefore enumerated in polynomial space. The second part is more difficult. Computing dis$(\vec{p}_t)$ for each t=1,...,k-1 can be done in NC(poly(n)) as shown in Lemma 6.2.2. Step 2 in the main algorithm only contains fundamental operations of matrices and the spectral decomposition. Thus, it also admits a parallel algorithm in NC(poly(n)). One can easily compose the two circuits and get a polynomial space implementation by the relation NC(poly)=PSPACE [20]. The third part can obviously be done in polynomial space. Thus, by composing these three polynomial-space implementable parts, one proves that the whole algorithm can be done in PSPACE.

□

6.4 Simulation of several variants of QMA(2)

This section illustrates the use of the algorithm shown in Section 6.3 to simulate some variants of the complexity class QMA(2). The idea is to show for those variants, the corresponding POVM matrices of acceptance are $(M, \vec{w})$-decomposable with small Ms. Recall the definition of the complexity class QMA(2).

Definition 6.4.1. A language $\mathcal{L}$ is in QMA$(2)_{m,c,s}$ if there exists a polynomial-time generated family of quantum verification circuits $Q = \{Q_n | n \in \mathbb{N}\}$ such that for any input x of size n, the circuit Q_n implements a two-outcome measurement $\{Q_x^{\text{acc}}, \mathbb{1} - Q_x^{\text{acc}}\}$. Furthermore,

- Completeness: If $x \in \mathcal{L}$, there exist $|\psi_1\rangle \in \mathcal{A}_1, |\psi_2\rangle \in \mathcal{A}_2$, each of m qubits,

$$\langle Q_x^{\text{acc}}, |\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|\rangle \geq c.$$

- Soundness: If $x \notin \mathcal{L}$, then for any states $|\psi_1\rangle \in \mathcal{A}_1, |\psi_2\rangle \in \mathcal{A}_2$,

$$\langle Q_x^{\text{acc}}, |\psi_1\rangle\langle\psi_1| \otimes |\psi_2\rangle\langle\psi_2|\rangle \leq s.$$

We call QMA(2)=QMA$(2)_{\text{poly}(n),2/3,1/3}$. It is easy to see that simulating the complexity class QMA(2) amounts to distinguishing between the two promises of the maximum acceptance probability (i.e. OptSep(Q_x^{acc})).

The first example is the variant with logarithmic size proofs (QMA$(2)_{O(\log(n)),2/3,1/3}$). It is not hard to find out the corresponding POVMs of acceptance (i.e. Q_x^{acc}) need to be (poly(n),$\vec{w}$)-decomposable where $\vec{w} = (1,1)$ since $\mathcal{A}_1, \mathcal{A}_2$ are only of polynomial dimension. Thus, it follows directly from Corollary 6.3.3 that OptSep(Q_x^{acc}) can be approximated in polynomial space. Namely,

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

The next example is slightly less trivial. Before moving on, we need some terminology about the quantum verification circuits Q. Assume the input x is fixed from now on. Let $\mathcal{A}_1, \mathcal{A}_2$ be the Hilbert space of size $d_{\mathcal{A}}$ for the two proofs and let $\mathcal{V}$ be the ancillary space of size $d_{\mathcal{V}}$. Then the quantum verification process will be carried out on the space $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \mathcal{V}$ with some initial state $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \left|\vec{0}\right\rangle$ where $|\psi_1\rangle, |\psi_2\rangle$ are provided by the provers. The verification process is also efficient in the sense that the whole circuit only consists of polynomial many elementary gates. Without loss of generality, we can fix one universal gate set for the verification circuits. Particularly, we choose the universal gate set to be single qubit gates plus the CNOT gates [91]. One can also choose other universal gate sets without any change of the main result.

We categorize all elementary gates in the verification circuits into two types. A gate is of *type-I* if it only affects the qubits within the same space (i.e, $\mathcal{A}_1, \mathcal{A}_2,$ or, $\mathcal{V}$). Otherwise, this gate is of *type-II*. It is easy to see single qubit gates are always type-I gates. The only type-II gates are CNOT gates whose control qubit and target qubit sit in different spaces. Let $p, r : \mathbb{N} \to \mathbb{N}$ be polynomial-bounded functions. A polynomial-time generated family of quantum verification circuits Q is called $Q[p, r]$ if each Q_n only contains $p(n)$ type-I elementary gates and $r(n)$ type-II elementary gates.

Definition 6.4.2. A language $\mathcal{L}$ is in $\mathrm{QMA}(2)_{m,c,s}[p, r]$ if $\mathcal{L}$ is in $\mathrm{QMA}(2)_{m,c,s}$ with some $Q[p, r]$ verification circuit family.

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

Lemma 6.4.3. *For any family of verification circuits $Q[p, r]$, the POVM Q_x^{acc} is $(4^{r(n)}, (1, 1))$-decomposable for any $n \in \mathbb{N}$ and input x. Moreover, this decomposition can be calculated in parallel with $O(t(n)4^{r(n)}) \times poly(n)$ time.*

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

$$Q_x^{\mathrm{acc}} = \mathrm{Tr}_{\mathcal{V}} \left(\mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \left|\vec{0}\right\rangle\left\langle\vec{0}\right| \left(U^* \mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \mathbb{1}_{\bar{\mathcal{V}}} \otimes |1\rangle\langle 1| \, U\right) \mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \left|\vec{0}\right\rangle\left\langle\vec{0}\right| \right).$$

Let $P_{t+1} = \mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \mathbb{1}_{\bar{\mathcal{V}}} \otimes |1\rangle\langle 1|$ and $P_\tau = U_\tau^* P_{\tau+1} U_\tau$ for τ=t,t-1,...,1. It is easy to see $P_1 = U^*(\mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \mathbb{1}_{\bar{\mathcal{V}}} \otimes |1\rangle\langle 1|)U$. Also it is straightforward to verify that P_{t+1} is 1-decomposable. Now let us observe how the decomposability of P_τ changes with τ.

For each τ, the unitary U_τ either corresponds to a type-I or type-II elementary gate. In the former case, applying U_τ won't change the decomposability. Thus, P_τ is M-

decomposable if $P_{\tau+1}$ is. In the latter case, applying U_τ will potentially change the decomposability in the following way. For any such CNOT gate one has $U_\tau = |0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes X$ where X is the Pauli matrix for the flip. And one can show

$$\begin{aligned} P_\tau &= (|0\rangle\langle 0| \otimes \mathbb{1}) P_{\tau+1} (|0\rangle\langle 0| \otimes \mathbb{1}) + (|0\rangle\langle 0| \otimes \mathbb{1}) P_{\tau+1} (|1\rangle\langle 1| \otimes X) \\ &+ (|1\rangle\langle 1| \otimes X) P_{\tau+1} (|0\rangle\langle 0| \otimes \mathbb{1}) + (|1\rangle\langle 1| \otimes X) P_{\tau+1} (|1\rangle\langle 1| \otimes X). \end{aligned}$$

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

Finally, it is not hard to verify that multiplications with $\mathbb{1}_{\mathcal{A}_1\mathcal{A}_2} \otimes \left|\vec{0}\right\rangle\left\langle\vec{0}\right|$ and partial trace over $\mathcal{V}$ won't change the decomposability of P_1. Namely, we have Q_x^{acc} is $(4^{r(n)}, (1,1))$-decomposable. The above proof can also be considered as the process to compute the decomposition of Q_x^{acc}. Each multiplication of matrices can be done in NC(poly(n)). And the total number of multiplications is upper bounded by $O(t(n)4^{r(n)})$. Therefore, the total parallel running time is upper bounded by $O(t(n)4^{r(n)}) \times$ poly(n). □

We will show that when the number of type-II gates is relatively small, one can simulate this complexity model efficiently by the algorithm in Fig. 6.1.

Corollary 6.4.4. QMA(2)$[poly(n), O(\log(n))] \subseteq$ PSPACE.

Proof. This is a simple consequence of Lemma 6.4.3 and Corollary 6.3.3. For any fixed x of length n, one can first compute the decomposition of Q_x^{acc} in parallel with $O(t(n)4^{r(n)}) \times$ poly(n) time, which is parallel polynomial time in n when $r(n) = O(\log(n))$ and $t(n) \in$ poly(n). Hence the first step can be done in polynomial space by NC(poly)=PSPACE [20].

Then one can invoke the parallel algorithm in Corollary 6.3.3 to approximate $\mathrm{OptSep}(Q_x^{\mathrm{acc}})$ to sufficient precision δ such that one can distinguish between the two promises. Precisely in this case, we choose those parameters as follows,

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

Thus the whole algorithm can be done in polynomial space. □

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

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

6.5 Quasi-polynomial algorithms for local Hamiltonian cases

In this section, we illustrate that if Q appears in the objective function that is a local Hamiltonian then the optimal value $\mathrm{OptSep}(Q)$ can be efficiently computed by our main algorithm. Consider any k-partite space $\mathcal{A}_1 \otimes \mathcal{A}_2 \otimes \cdots \otimes \mathcal{A}_k$ where each $\mathcal{A}_i$ contains n qubits (i.e., of dimension 2^n).

Definition 6.5.1. Any Hermitian Q over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ is a l-local Hamiltonian if Q is expressible as $Q = \sum_{i=1}^{r} H_i$ where each term is a Hermitian operator acting on at most l qubits among k parties.

Hamiltonians are widely studied in physics since they usually characterize the energy of a physical system. Local Hamiltonians are of particular interest since they refer to the energy of many interesting models in low-dimension systems. Our algorithm can be

considered as a way to find the minimum energy of some physical system achieved by separable states.

Local Hamiltonians are also appealing to computational complexity theorists since the discovery of the promise 5-local Hamiltonian problem [73] which turns out to be QMA-complete. Precisely, it refers to the following promise problem when $k = 1, l = 5$.

Problem 6.5.2 (k-partite l-local Hamiltonian problem)**.** *Take the expression $Q = \sum_{i=1}^{r} H_i$ for any l-local Hamiltonian over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ as input*[3]*, where $\|H_i\|_\infty \leq 1$ for each i. Let OptSep(Q) denote the minimum value of $\langle Q, \rho \rangle$ achieved for some $\rho \in \mathrm{SepD}(\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k)$. The goal is to tell between the following two promises: either OptSep$(Q) \geq a$ or OptSep$(Q) \leq b$ for some $a > b$ with an inverse polynomial gap.*

When $k = 1$, the promise problem defined above is exactly the original l-local Hamiltonian problem. Subsequent results demonstrate that it remains QMA-complete even when $l = 3, 2$ [3, 59, 92]. Our definition of the promise problem naturally extends to the k-partite case. We refer to Chapter 14 in [73] for technical details. It is not hard to see that k-partite l-local Hamiltonian problem belongs to QMA(k) by applying similar techniques in the original proof. However, it does not remain QMA(k)-complete. Indeed, Chailloux and Sattath [25] proved that the k-partite local Hamiltonian problem defined above lies in QMA for constant k.

Lemma 6.5.3. *Any l-local Hamiltonian Q over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ such that $Q = \sum_{i=1}^{r} H_i$ and $\|H_i\|_\infty \leq w$ is $(O((4nk)^l), w)$-decomposable.*

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

[3] It is noteworthy to mention that the input size of local Hamiltonian problems can be only poly-logarithm in the dimension of the space where Q sits in.

decomposable. □

Corollary 6.5.4. *Take the expression $Q = \sum_{i=1}^{r} H_i$ of any l-local Hamiltonian over $\mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k$ (each $\mathcal{A}_i$ is of dimension $d = 2^n$) such that $\|H_i\|_\infty \leq w$ for each i as input. Assuming $k, l = O(1)$, the quantity OptSep(Q) can be approximated to precision δ in quasi-polynomial time in $d, w, 1/\delta$.*

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

Proof. The proof of the first part follows directly from Lemma 6.5.3 and Theorem 6.3.2. Recall the proof of Lemma 6.5.3 also provides a way to compute the decomposition of Q given the expression $Q = \sum_{i=1}^{r} H_i$ as input. It is easy to verify that $O(r4^l)$ time (upper bounded by $O((4k \log d)^l)$) is sufficient to complete this computation. After that, one may directly invoke the algorithm in Fig. 6.1 and make use of Theorem 6.3.2. Now we substitute the following identities into our main algorithm. Note $k, l = O(1)$ and we have $M = O(\log^{O(1)} d), W = w^{O(1)}$. One immediately gets the total running time bounded by

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

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

For the second part when n is considered as the input size, it is easy to see that the computation of the decomposition of Q according to Lemma 6.5.3 can be done in NC(poly), henceforth in polynomial space. (Note $M = O(\text{poly(n)})$.) Then by composing with the polynomial-space algorithm implied by Corollary 6.3.3, one proves that the whole algorithm can be implemented in polynomial space. □

Remarks. It is a direct consequence of Corollary 6.5.4 that Problem 6.5.2 is inside PSPACE.

1. Compute the spectral decomposition of $Q = \sum_t \lambda_t |\Psi_t\rangle\langle\Psi_t|$. Choose $\epsilon = \delta/2$ and $\Gamma_\epsilon = \{t : \lambda_t \geq \epsilon\}$. Also let OPT store the optimum value.
2. Generate the ε-net of the unit ball of $\mathbb{C}^{|\Gamma_\epsilon|}$ under the Euclidean norm with $\varepsilon = \frac{\delta}{4\|Q\|_F}$. Denote such a set by $\mathcal{N}_\varepsilon$. Then for each point $\alpha \in \mathcal{N}_\varepsilon$,
   (a) Compute $|\phi_\alpha\rangle = \sum_{t\in\Gamma_\epsilon} \alpha_t^* \sqrt{\lambda_t} |\Psi_t\rangle$ and compute the Schmidt decomposition of $|\phi_\alpha\rangle$, i.e.,
   $$|\phi_\alpha\rangle = \sum_i \mu_i |u_i\rangle |v_i\rangle ,$$
   where $\mu_1 \geq \mu_2 \geq \cdots$ and $\{u_i\}, \{v_i\}$ are orthogonal bases.
   (b) Update OPT as follows: OPT=max$\{$OPT, $\mu_1\}$.
3. Return OPT.

Figure 6.2: The algorithm runs in time exponential in $\|Q\|_F/\delta$.

6.6 Exponential running time algorithm in $\|Q\|_F$

In this section we demonstrate another application of the simple idea "enumeration" by epsilon-net to Problem 6.1.1. As a result, we obtained an algorithm with running time exponential in $\|Q\|_F$ (or $\|Q\|_{\text{LOCC}}$ [109][4]) for computing OptSep(Q) with additive error δ. A similar running time $\exp(O(\log^2(d)\delta^{-2}\|Q\|_F^2))$ was obtained in [22] using some known results in quantum information theory.

Theorem 6.6.1. *Given any positive semidefinite Q over $\mathcal{A}_1 \otimes \mathcal{A}_2$ (of dimension $d \times d$) and $\delta > 0$, the algorithm in Fig. 6.2 approximates OptSep(Q) with additive error δ with running time* $\exp(O(\log(d) + \delta^{-2}\|Q\|_F^2 \ln(\|Q\|_F/\delta)))$.

Proof. We first prove the correctness of the algorithm. The analysis will mainly be divided into two parts. Let $S_\epsilon = \text{span}\{|\Psi_t\rangle \, | t \in \Gamma_\epsilon\}$. The first part shows it suffices to only consider vectors inside the subspace S_ϵ for approximating OptSep(Q) with additive error δ. The second one demonstrates that our algorithm in Fig. 6.2 approximates the optimal value obtained by only considering vectors in S_ϵ. Precisely, since $\{|\Psi_i\rangle\}$ forms a basis,

[4]This follows easily from the fact $\|Q\|_F = O(\|Q\|_{\text{LOCC}})$ [109] where $\|Q\|_{\text{LOCC}}$ stands for the LOCC norm of the operator Q.

one has $|u\rangle|v\rangle = \sum_{t\in\Gamma_\epsilon}\beta_t|\Psi_t\rangle + \sum_{t\notin\Gamma_\epsilon}\beta_t|\Psi_t\rangle$ where β is a unit vector in $\mathbb{C}^{d^2}$. Then we have

$$\langle Q, |u\rangle\langle u| \otimes |v\rangle\langle v|\rangle = \underbrace{\sum_{t\in\Gamma_\epsilon}\lambda_t|\beta_t|^2}_{(I)} + \underbrace{\sum_{t\notin\Gamma_\epsilon}\lambda_t|\beta_t|^2}_{(II)},$$

where term (II) is obviously bounded by $\delta/2$ (i.e., $\sum_{t\notin\Gamma_\epsilon}\lambda_t|\beta_t|^2 \le \delta/2$). For term (I), it is equivalent to OptSep$(\tilde{Q})$ where $\tilde{Q} = \sum_{t\in\Gamma_\epsilon}\lambda_t|\Psi_t\rangle\langle\Psi_t|$. Namely, small eigenvalues are truncated in $\tilde{Q}$. Now observe the following identity.

$$\begin{aligned}
\max_{|u\rangle|v\rangle}\langle\tilde{Q}, |u\rangle\langle u|\otimes|v\rangle\langle v|\rangle &= \max_{|u\rangle|v\rangle}\sum_{t\in\Gamma_\epsilon}\lambda_t|\langle u|\langle v|\Psi_t\rangle|^2 = \max_{|u\rangle|v\rangle}\|\gamma^{u,v}\|^2 \\
&= \max_{|u\rangle|v\rangle}\max_{\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)}|\sum_{t\in\Gamma_\epsilon}\alpha_t^*\sqrt{\lambda_t}\langle u|\langle v|\Psi_t\rangle|^2 \\
&= \max_{|u\rangle|v\rangle}\max_{\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)}|\langle u|\langle v|\phi_\alpha\rangle|^2 \\
&= \max_{\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)}\max_{|u\rangle|v\rangle}|\langle u|\langle v|\phi_\alpha\rangle|^2,
\end{aligned}$$

where $\gamma^{u,v}\in\mathbb{C}^{|\Gamma_\epsilon|}$ and $\gamma_t^{u,v} = \sqrt{\lambda_t}\langle u|\langle v|\Psi_t\rangle$ for each $t\in\Gamma_\epsilon$. The second line comes from the duality of the Euclidean norm (i.e., $\|y\| = \max_{\|z\|\le 1}|\langle z|y\rangle|$). The third line comes by exchanging positions of the two maximizations. We then make use of the following well-known fact.

Fact 6.6.2. For any bipartite vector $|\psi\rangle$ with the Schmidt decomposition

$$|\psi\rangle = \sum_i \mu_i|u_i\rangle|v_i\rangle,$$

where $\mu_1 \ge \mu_2 \ge \cdots$ and $\{u_i\}, \{v_i\}$ are orthogonal bases . Then $\max_{|u\rangle|v\rangle}|\langle u|\langle v|\psi\rangle| = \mu_1$ and the maximum value is obtained by choosing $|u\rangle|v\rangle$ to be $|u_1\rangle|v_1\rangle$.

It is not hard to see that our algorithm computes exactly the term on the third line except that we replace the unit ball by its ϵ-net. However, this won't incur too much extra error. For any $\alpha\in\mathbf{B}(\mathbb{C}^{|\Gamma_\epsilon|},\|\cdot\|)$, there exists $\tilde{\alpha}\in\mathcal{N}_\epsilon$, such that $\|\alpha-\tilde{\alpha}\|\le\varepsilon$. Thus, the extra

error incurred is $|| \langle u| \langle v|\phi_\alpha\rangle |^2 - | \langle u| \langle v|\phi_{\tilde{\alpha}}\rangle |^2|$ and can be bounded by

$$\begin{aligned}(\| |\phi_\alpha\rangle \| + \| |\phi_{\tilde{\alpha}}\rangle \|)| \langle u| \langle v|\psi_\alpha - \psi_{\tilde{\alpha}}\rangle | &\leq 2 \max_{\|\beta_1\|\leq 1} \|\phi_{\beta_1}\| \max_{\beta_2=\alpha-\tilde{\alpha},\|\beta_2\|\leq\varepsilon} \|\phi_{\beta_2}\| \\ &= 2\sqrt{\|Q\|_{\mathrm{F}}} \times \varepsilon\sqrt{\|Q\|_{\mathrm{F}}} \leq \delta/2,\end{aligned}$$

where $\max_{\|\beta\|\leq\epsilon'} \|\phi_\beta\| \leq \epsilon'\sqrt{\|Q\|_{\mathrm{F}}}$ for any $\epsilon' > 0$ can be verified directly and therefore the total additive error is bounded by $\delta/2 + \delta/2 = \delta$.

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

CHAPTER 7

Non-Identity Check of Quantum Circuits

This chapter is based on a joint work with Zhengfeng Ji [66].

In this chapter we study one QMA-complete problem, called Non-Identity Check, which asks whether a given quantum circuit is far away from identity or not. It is well-known that similar problems for classical circuits admit efficient randomized solutions. Thus, it is striking to show that the Non-Identity Check problem for quantum circuits is QMA-complete [65]. We study this problem further to see whether this QMA-hardness remains for small and structured quantum circuits. In this chapter we conclude that even for poly-logarithmic depth quantum circuits, the Non-Identity Check problem remains QMA-complete. This suggests that even short-depth quantum circuits deviate significantly from their classical counterparts and might imply the hardness of simulating these circuits.

7.1 Introduction

Quantum circuits are the natural quantum analogues of classical circuits and serve as an important model [115] to analyze the power of quantum computation. In this chapter we study quantum circuits in the complexity perspective. We restrict our attention to the following definition of quantum circuits and refer curious readers to [91] for general cases.

We consider pure state quantum circuits. In this model, the whole quantum circuit

can be represented as a large unitary U. The initial pure state $|\psi\rangle$, which the quantum circuit U applies on, and the measurements, which apply on the final state $U|\psi\rangle$, are however not interesting for our purpose, and are thus not discussed here. Let U be such a quantum circuit acting on n qubits that consists of a sequence of quantum elementary gates $U_1, U_2, \cdots, U_T$ such that,

$$U = U_T U_{T-1} \cdots U_2 U_1,$$

in which each elementary gate U_i is a unitary applied on a constant-size (often two or three) subset of n qubits and T is the number of elementary gates in this circuit. It follows easily that the classical descriptions of gate $U_i, i = 1 \cdots T$ and their order provide a classical description of the quantum circuit U.

Each elementary gate builds a connection among a constant number of qubits. When applying these elementary gates sequentially, it forms paths from input qubits to output qubits. The *depth* of a quantum circuit U is the maximum number of gates encountered on any path from an input qubit to an output qubit in the circuit. Circuit depths can also be treated as the parallel running time, or the number of time units needed to apply a circuit when operations may be parallelized in any way that respects the topology of the circuit.

Much of the difficulty in implementing quantum computation is the decoherence effect of the qubits which happens in a very short time. In this sense, short-depth quantum circuits seem to be more resilient to the decoherence effect. Thus, analyzing the power of short depth quantum circuits is of significant interest. A few examples about the power of logarithmic depth quantum circuits have been proposed in the past few years [30, 90]. Also, a systematic procedure has also been discovered [23] to parallelize a class of quantum circuits to logarithmic depths.

The *Non-Identity Check* problem is to decide whether a quantum circuit is far away from identity, given a classical description of the circuit. More generally, one can ask

whether two quantum circuits U and V are equivalent or not. It is easy to see that the equivalence problem can be reduced to the identity check problem of $UV^\dagger$. Classically, similar problems [19, 98, 117] that determine whether two given classical circuits are equivalent turn out to admit efficient randomized solutions, i.e., in BPP. In contrast, we know that the quantum Non-Identity Check problem is QMA-complete [65].

We have elaborated on the history of QMA-complete problems in Chapter 1. The main result in this chapter is that Non-Identity Check for short quantum circuits remains QMA-complete. Formally we have,

Theorem 7.1.1. *Non-Identity Check of constant depth quantum circuits on n qubits is* QMA*-complete if the encoding of the circuit describes each gate to at least $\Omega(\log n)$ bit of precision.*

When restricting to circuits that consist of gates from a finite universal gate set, we have the following corollary, which follows directly from *Solovay-Kitaev Theorem* [32].

Corollary 7.1.2. *Non-Identity Check is* QMA*-complete for $O(\log^\delta(n))$-depth quantum circuits of an arbitrary universal gate set on n qubits where $\delta \approx 3$.*

If input circuits make use of efficient universal gate sets as shown in Ref. [56], then we have the following stronger result:

Corollary 7.1.3. *There exists a universal gate set such that Non-Identity Check is* QMA*-complete for logarithmic-depth quantum circuits using this particular universal gate set.*

To prove Theorem 7.1.1, we start with the 1-D local Hamilton problem (QMA-complete), and reduce it to the Non-Identity Check problem. The reminder of this chapter is organized as follows. In Section 7.2, we elaborate on our main technique tool, the *phase and numerical range* of an operator. We prove our main result in Section 7.3 and conclude in Section 7.4.

7.2 Phase and Numerical Range

The *numerical range* of an operator A is the subset of the complex plane $\{\langle\psi| A |\psi\rangle\}$ and is known to be convex. In particular, for normal matrices the numerical range is simply the convex hull of all eigenvalues. For any Hermitian matrix H, $\lambda_{\max}(H)$ and $\lambda_{\min}(H)$ are the largest and smallest eigenvalue of H. Denote the eigenvalue range of H by $\lambda(H) = \lambda_{\max}(H) - \lambda_{\min}(H)$.

The eigenvalues of a unitary matrix U lie on the unit circle of the complex plane. The distribution of the eigenvalues is important to characterize the closeness of U to identity I. See for example the illustration made in Figure 7.1 where the eigenvalues of U are marked on the unit circle as small hollow circles. Use $\alpha_{\max}(U)$ and $\alpha_{\min}(U)$ to denote the maximal and minimal value of the arguments of eigenvalues of U taken in the interval $(-\pi, \pi]$. They correspond to the arguments of point A and C in Figure 7.1. Let $\tilde{\alpha}(U)$ be the length of the shortest arc that contains all eigenvalues of U (which corresponds to arc $\overset{\frown}{AC}$ in the figure). It was known that U is perfectly distinguishable from I if and only if $\tilde{\alpha}(U) \geq \pi$ [94]. Define a new quantity called *phase range* as

$$\alpha(U) = \min\{\pi, \tilde{\alpha}(U)\}, \tag{7.2.1}$$

and extend it to be defined on two unitary operations U and V as

$$\alpha(U, V) = \alpha(U^\dagger V). \tag{7.2.2}$$

The diamond norm [75] serves as a good way of measuring distances between quantum operations. For any superoperator Φ mapping operators acting on Hilbert space $\mathcal{H}_1$ to operators acting on Hilbert space $\mathcal{H}_2$, the diamond norm $\|\Phi\|_\diamond$ is

$$\|\Phi\|_\diamond = \max_\rho \|\Phi \otimes I_{\mathcal{H}_1}(\rho)\|_{\mathrm{Tr}}, \tag{7.2.3}$$

where the maximum takes over density matrices ρ.

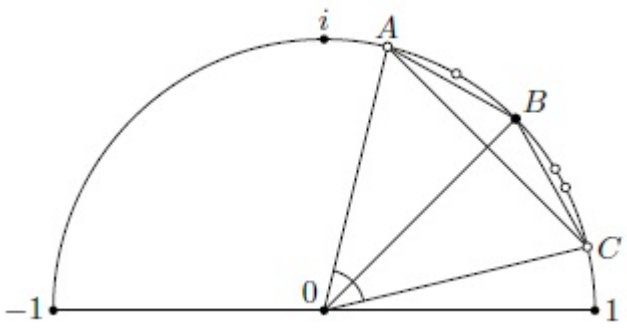

Figure 7.1: Comparison of different distances

Let $\mathcal{U}$ be the quantum operation corresponding to unitary U,

$$\mathcal{U}(\rho) = U\rho U^\dagger.$$

It was known that [111]

$$\|\mathcal{U} - \mathcal{I}\|_\diamond = 2\sqrt{1 - \nu^2(U)},$$

where $\mathcal{I}$ is the identity operation and $\nu(U)$ is the minimum distance of the zero point to the numerical range of U. As U is normal, its numerical range is the convex hull of all of its eigenvalues and the diamond norm is exactly the length of segment AC in Figure 7.1. Therefore, we have

$$\|\mathcal{U} - \mathcal{I}\|_\diamond = 2\sin\frac{\alpha(U)}{2}.$$

Another way to measure the closeness of U and I is the following quantity [65]:

$$\min_\varphi \|U - e^{i\varphi} I\|. \tag{7.2.4}$$

We can also visualize the idea of the definition in Figure 7.1. The minimum in Eq. (7.2.4) will be achieved when φ is the argument of point B in the middle of the arc connection A and C, and the minimum value is the length of segment AB. Its relation with phase range α when $\alpha(U) < \pi$ is

$$\min_\varphi \|U - e^{i\varphi} I\| = 2\sin\frac{\alpha(U)}{4}.$$

When $\alpha(U) = \pi$, they are not related but we will always have

$$\min_{\varphi} \|U - e^{i\varphi} I\| \geq 2 \sin \frac{\alpha(U)}{4}. \tag{7.2.5}$$

The main technical tools in our proof are $\alpha, \alpha_{\max}, \alpha_{\min}$. In particular, we find the following properties about $\alpha_{\max}, \alpha_{\min}$ extremely useful. Note that the proof of the first two lemmas can be found in the Appendix of Ref. [28].

Lemma 7.2.1. *For unitaries U_1 and U_2 such that*

$$\alpha_{\max}(U_1) + \alpha_{\max}(U_2) < \pi,$$

$$\alpha_{\min}(U_1) + \alpha_{\min}(U_2) > -\pi,$$

we have

$$\alpha_{\max}(U_1 U_2) \leq \alpha_{\max}(U_1) + \alpha_{\max}(U_2),$$

$$\alpha_{\min}(U_1 U_2) \geq \alpha_{\min}(U_1) + \alpha_{\min}(U_2).$$

Lemma 7.2.2. *For Hermitian matrices H, K and $-\pi < H + K < \pi$,*

$$\begin{aligned} \alpha_{\max}(e^{iH} e^{iK}) &\leq \alpha_{\max}(e^{i(H+K)}), \\ \alpha_{\min}(e^{iH} e^{iK}) &\geq \alpha_{\min}(e^{i(H+K)}). \end{aligned} \tag{7.2.6}$$

Lemma 7.2.3. $\alpha(U_1, U_2) \leq \alpha(U_1) + \alpha(U_2)$.

Proof. If either $\alpha(U_1)$ or $\alpha(U_2)$ equals π, the above equation obviously holds. Now if both $\alpha(U_1)$ and $\alpha(U_2)$ are less than π, we can choose phases φ_1 and φ_2 such that

$$U_1^{\dagger} = e^{i\varphi_1} V_1, U_2 = e^{i\varphi_2} V_2,$$

and V_1 and V_2 have eigenvalues of arguments in $(-\pi/2, \pi/2)$. The condition in Lemma 7.2.1 holds for V_1 and V_2 and it follows that $\alpha(V_1 V_2) \leq \alpha(V_1) + \alpha(V_2)$ which finishes the proof by noticing that $\alpha(U)$ is invariant under the change of a global phase in U. □

It's interesting to note that Lemma 7.2.3 implies that $\alpha(U_1, U_2)$ is a distance measure on the space of $U(d)/U(1)$. Specifically,

$$\alpha(U_1, U_3) = \alpha(U_1^\dagger U_2 U_2^\dagger U_3) \leq \alpha(U_1^\dagger U_2) + \alpha(U_2^\dagger U_3) = \alpha(U_1, U_2) + \alpha(U_2, U_3).$$

Lemma 7.2.4. *For unitaries U and V, $|\alpha(U) - \alpha(V)| \leq \pi\|U - V\|$.*

Proof. Since α is a distance measure,

$$|\alpha(U) - \alpha(V)| = |\alpha(U, I) - \alpha(V, I)| \leq \alpha(U^\dagger V).$$

Using Eq. (7.2.5) and $\sin(x) \geq 2x/\pi$ for $x \in [0, \pi/2]$, we have

$$\|U - V\| \geq \min_\varphi \|U - e^{i\varphi} V\| \geq 2 \sin \frac{\alpha(U^\dagger V)}{4} \geq \frac{1}{\pi}\alpha(U^\dagger V) \geq \frac{1}{\pi}|\alpha(U) - \alpha(V)|.$$

□

Lemma 7.2.5. *For Hermitian matrices H, K and $0 \leq H, K, H + K \leq \pi$, $0 < t < 1$,*

$$|\alpha(e^{iHt} e^{iKt}) - \alpha(e^{i(H+K)t})| \leq ct^2, \tag{7.2.7}$$

where c is a constant independent of H, K and t.

Proof. Using the expansion of the matrix exponential function and the condition $0 \leq H, K, H + K \leq \pi$, $0 < t < 1$, it's easy to show that there exists some constant c_1 such that

$$\|e^{iHt} e^{iKt} - e^{i(H+K)t}\| \leq c_1 t^2.$$

The inequality follows immediately from Lemma 7.2.4. □

7.3 Hardness of Non-Identity Check for Short Circuits

We will make use of the 1-D local Hamiltonian problem as our start point. Consider a Hamiltonian H of an n-particle system with constant local dimensions. H is called k-local

if it is the sum $\sum_i H_i$ where each H_i acts non-trivially only on k particles. In some cases, there is also an underlying layout of the particles in the problem, for example the 1-D chain or the 2-D lattice, such that each local term H_i acts only on neighbouring particles corresponding to the layout. We will call them 1-D and 2-D local Hamiltonian problems respectively. For any 1-D Hamiltonian $H = \sum H_i$, the particles are arranged on a line, and each local term H_i acts non-trivially only on two neighbouring particles. General local Hamiltonian problems can be formalized as follows.

Definition 7.3.1 (The Local Hamiltonian Problem)**.** Given a k-local Hamiltonian $H = \sum_{i=1}^{r} H_i$ of n particles and two real numbers a, b, where H_i has a bounded norm and $b - a \geq 1/\text{poly}(n)$, r is polynomial in n and k is $O(1)$. It is promised that the lowest eigenvalue of H is either smaller than a or larger than b. Output "Yes" in the first case and "No" otherwise.

The problem was first shown to be QMA-complete for 5-local Hamiltonians [73, 5]. Recent improvements show that Hamiltonians with much simpler structures – 3-local, 2-local, 2-D, and even 1-D cases – are all complete for QMA [69, 59, 92, 3]. Non-Identity Check problem was first considered in Ref. [65]. It can be stated as,

Definition 7.3.2 (Non-Identity Check)**.** Given a classical description of a quantum circuit U on n qubits and two real numbers a, b with $b - a \geq 1/\text{poly}(n)$. It is promised that

$$\min_{\varphi} \ \|U - e^{i\varphi} I\|$$

is either larger than b or smaller than a. Output "Yes" in the first case and "No" in the second.

In the definition, the quantity $\min_{\varphi} \|U - e^{i\varphi} I\|$ is used to evaluate the closeness of U to identity. We can also use phase ranges $\alpha(U)$ or diamond norms instead. And it's

easy to see that, all the three definitions mentioned above can be used in defining the Non-Identity Check problem without changing its complexity. The point is that they are quantities related to each other by monotonic trigonometric functions. Moreover, any inverse polynomial gap in one of them implies a similar gap in others. In the following, we use phase ranges to define and analyze the Non-Identity Check problem. It is interesting to note at this point that the hardness of Non-Identity Check implies the hardness of the estimation of the diamond norm of the difference of two unitary quantum circuits to inverse polynomial precision.

Proof of Theorem 7.1.1. Now we sketch the proof of the QMA-hardness of the Non-Identity Check problem by reducing the 1-D local Hamiltonian problem to it.

Suppose we are given an instance of the 1-D local Hamiltonian problem which has input $H = \sum_{i=1}^{r} H_i$ and real numbers a, b with at least an inverse polynomial gap. H is a Hamiltonian of an n particle system with local dimension d and each term H_i is an operator on two neighboring particles which can be described by a d^2 by d^2 Hermitian matrix. It is a "Yes" instance if there exists some density matrix ρ such that $\langle H, \rho \rangle \leq a$ and a "No" instance if $\langle H, \rho \rangle \geq b$ for all ρ. This problem is known to be QMA-complete for $d \geq 12$. For simplicity, one can always rescale the problem and assume that H_i's are positive semidefinite and $\|H_i\| \leq 1$.

Note that the 1-D property of the problem allows us to write H as $H_{\text{odd}} + H_{\text{even}}$ where H_{odd} and H_{even} each consists of local terms acting on disjoint particles. This is illustrated in Figure 7.2. H_{odd} is the sum of $H_1, H_3, H_5, \ldots$ where H_1 acts on particle 1 and 2, H_3 acts on particle 3 and 4, etc. H_{even} consists of $H_2, H_4, \ldots$ where H_2 acts on particle 2 and 3, H_4 acts on particle 4 and 5, etc.

The first step in the reduction is to modify the Hamiltonian such that it will have eigenvalue r, where r is the number of local terms. To that end, we add an additional dimension,

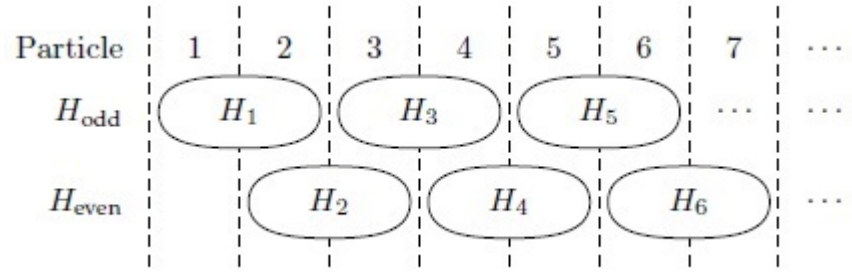

Figure 7.2: A 1-D local Hamiltonian

labeled by $|d\rangle$, to each particle, and consider the Hamiltonian with local terms

$$\tilde{H}_i = |d\rangle\langle d| \otimes |d\rangle\langle d| + H_i.$$

It should be understood that H_i acts trivially when underlying particles are in state $|d\rangle$. Let $\tilde{H}$ be the sum $\sum_i \tilde{H}_i$. It's obvious that $|d\rangle^{\otimes n}$ is an eigenvector of $\tilde{H}$ with eigenvalue r, while the smallest eigenvalue of $\tilde{H}$ equals that of H. The 1-D Hamiltonian problem of H is now reduced to deciding if the eigenvalue range of $\tilde{H}$, denoted by $\lambda(\tilde{H})$, is at least $r-a$ or no more than $r-b$.

Before further reducing the problem, we normalize $\tilde{H}$ by dividing $2r/\pi$ for technical convenience. Denote the normalized Hamiltonian again by H and its local terms by H_i for simplicity. After the normalization, we have $\|H\| \leq \pi/2$. Let l and s be $(r-a)\pi/2r$ and $(r-b)\pi/2r$ respectively. It's a "Yes" instance if $\lambda(H) \geq l$ and a "No" instance if $\lambda(H) \leq s$. Note that l and s are separated by an inverse polynomial gap.

We can construct a Non-Identity Check instance as follows. The circuit is simply

$$U_H = e^{iH_{\text{even}}t} e^{iH_{\text{odd}}t}, \tag{7.3.1}$$

with two carefully chosen threshold numbers a, b that have an inverse polynomial gap. As the local terms $H_1, H_3, H_5, \ldots$ in H_{odd} are on disjoint particles, $e^{iH_{\text{odd}}t}$ equals the tensor product of $e^{iH_1t}, e^{iH_3t}, e^{iH_5t}, \ldots$ and can be implemented in parallel. Similar properties hold for H_{even}. Therefore U_H is indeed a constant depth circuit.

Since $\lambda(H)$ is promised either larger than l or smaller than s, one can verify that the promises for the above Non-Identity problem also hold. If $\lambda(H)$ is larger than l, it follows from Lemma 7.2.5 that $\alpha(U_H)$ is at least

$$\alpha(e^{iHt}) - ct^2 = \lambda(Ht) - ct^2 \geq lt - ct^2 = b.$$

If $\lambda(H)$ is smaller than s, Lemma 7.2.2 implies that $\alpha(U_H)$ is at most

$$\alpha(e^{iHt}) \leq st = a.$$

It's also easy to check that the eigenvalue range problem of H is a "Yes" (or "No") instance if and only if the Non-Identity Check problem of U_H is a "Yes" (or "No") instance. □

It's worth noting that the main idea in the proof is highly related to quantum simulation using Trotter expansion

$$e^{A+B} = \lim_{n\to\infty} (e^{A/n} e^{B/n})^n.$$

Fortunately however, it is enough to simulate the first round of $e^{A/n}e^{B/n}$ and leave the amplification procedure to the verifier.

The circuit we constructed above contains quantum gates such as $e^{iH_1 t}$ which need $\Omega(\log n)$ bits to specify. In order to translate the result to the case in which only a finite universal set of quantum gates are allowed, we need to expand each gate in the circuit using Solovay-Kitaev theorem. This will give us the result in Corollary 7.1.2. The main problem here is to analyze how the imperfections in each gate will affect the phase range α of the circuit. Suppose we want to use unitary gates U_1 and U_2 but the actual implementations

are unitary gates V_1 and V_2 with $V_1 = U_1 + E_1$ and $V_2 = U_2 + E_2$, then

$$\begin{aligned}\|V_1V_2 - U_1U_2\| &= \|(U_1 + E_1)(U_2 + E_2) - U_1U_2\| \\ &= \|U_1E_2 + E_1(U_2 + E_2)\| \\ &= \|U_1E_2 + E_1V_2\| \\ &\leq \|E_1\| + \|E_2\|,\end{aligned}$$

and similarly,

$$\|V_1 \otimes V_2 - U_1 \otimes U_2\| \leq \|E_1\| + \|E_2\|.$$

These two facts and Lemma 7.2.4 imply that for any circuit C and its imperfect implementation C', we have,

$$|\alpha(C) - \alpha(C')| \leq \pi\|C - C'\| \leq \pi \sum_i \|E_i\|,$$

where E_i's are errors in all gates of C'. Thus, the total error in α of the circuit is at most π times the summation of norms of all errors in each gate. It can be made inverse polynomially small and much smaller than the gap between threshold parameter a and b. This validates the claim in Corollary 7.1.2.

It's proved in [56] that there exists some universal gate set such that only a $O(\log(1/\epsilon))$ number of gates are required to achieve an error bound of ϵ. A similar argument as above gives the proof of Corollary 7.1.3.

7.4 Conclusion

In this chapter, we conclude that Non-Identity Check for constant depth quantum circuits is QMA-complete given $\Omega(\log n)$ bits of precision to each gate. However, the depth may vary when using different universal gate sets. Employing different versions of Solovay-Kitaev theorem, we are able to prove the hardness for circuits of poly-logarithmic or even logarithmic depths.

It is interesting to compare our result with the problem of distinguishing mixed state quantum circuits in terms of the diamond norm [4]. Although the main difference is simply whether some output are discarded or not, the problem of distinguishing mixed state quantum computation seems to be much harder. In fact, it was shown to be QIP-complete [97]. Rosgen [96] further proved that logarithmic depth mixed state quantum circuits are as hard to distinguish as polynomial depth ones and thus distinguishing logarithmic depth mixed state quantum circuits remains QIP-complete.

We leave the complexity of Non-Identity Check for constant depth quantum circuits with gates from a finite universal gate set as an interesting open problem.

CHAPTER 8

Conclusions and Future Directions

In this chapter, we summarize our results about quantum interactive proof systems. Moreover, we also briefly survey the author's contribution to other topics in quantum computation during his PhD period. We conclude this chapter with a few on-going projects about quantum interactive proof systems and our attack plans.

8.1 Conclusions

In this dissertation, we make several important contributions to the study of quantum interactive proof systems. Our results help determine the complexity of a few variants of quantum interactive proof systems. Since quantum models usually contain their classical counterparts as special cases, and thus inherit their classical lower bounds, our main contributions are to show their upper bounds, namely to provide a way to simulate these quantum models in desired complexity classes. To that end, we formulate a framework, called the *equilibrium value method*, to provide space-efficient solutions to a class of optimization problems that arise naturally in these quantum models. Applying this framework to specific models, we obtain PSPACE upper bounds for single-prover quantum interactive proof systems, short quantum refereed games and a variant of multiple-prover quantum Merlin-Arthur games. We elaborate more as follows.

- In Chapter 3, we formulate our equilibrium value method. We illustrate our approach

of reformulating optimization problems as zero-sum games and point out a few technical difficulties in doing so. In particular, we highlight two difficulties: the first one is that one can only hope to approximate equilibrium values of zero-sum games rather than to solve them exactly, and the second one is that we need an extra rounding theorem to convert approximate solutions to exact solutions without incurring too much error. We also highlight a specific form of equilibrium values of zero-sum games that can be directly solved by the matrix multiplicative weight update method. To illustrate our approach, we go through one example about simulating SDPs that come from QIP(2). We conclude this chapter with comparisons with Arora-Kale's use of the matrix multiplicative weight update method and how our solutions can be made space-efficient.

- In Chapter 4, we demonstrate then how our equilibrium value method can directly lead to the result that PSPACE contains QIP, and thus QIP=PSPACE. In fact, we provide two approaches to this result. The first one exploits more about the property of quantum interactive proof systems. It begins with one QIP-complete problem, called *close images*, and then translates it into an equilibrium value problem, which admits a direct solution from the matrix multiplicative weight update method. The second approach is to apply our equilibrium value method to solve SDPs from QMAM, which resembles Jain et al.'s approach, however, provides a much cleaner solution.
- In Chapter 5, we demonstrate how our equilibrium value method extends to much broader situations and provides PSPACE upper bounds for short quantum refereed games. This is the situation where direct applications of the matrix multiplicative weight update method fail and two main difficulties in applying our equilibrium value method come into play. We find a unified solution to these two difficulties by using appropriate penalties and a recursive rounding method. Our upper bounds also imply

a crucial difference between public and private randomness in refereed games.

- In Chapter 6, we switch to quantum Merlin-Arthur games with two-provers (QMA(2)). We provide a PSPACE upper bound to a non-trivial variant of QMA(2) that is up to date the most general one known in PSPACE. This result follows from a clever enumeration idea with the help of the problem structure and our equilibrium value method. We also provide an alterative and conceptually simpler algorithm for a related problem than previous results.
- In Chapter 7, we contribute to QMA-complete problems, where we demonstrate that the "Non-Identity Check" problem remains QMA-complete on circuits of poly-logarithmic depths, improving on polynomial depths from previous results. Our result follows from an application of the QMA-completeness of 1-d local Hamiltonian problems and also from a careful analysis of phase and number ranges of operators.

Contributions to Other Quantum Computation Topics

Besides working on quantum interactive proof systems, the author has also been actively working on several other interesting topics in quantum computation and made the following contributions.

- **Quantum Communications**. Proving quantum analogues of classical information theorems is usually a task fraught with difficulty, even for fundamental problems. Compression of quantum mixed states is one such example, where no clean characterization but a conjectured Holevo bound exists for the attainable compression rate.

  In an on-going project with Yaoyun Shi, we identify a class of ensembles of quantum mixed states whose compression rate, under a reasonable conjecture, is arbitrarily far away from the Holevo bound, contrary to the widely held belief. The compression rate of our examples is lower bounded by the sign rigidity of a slightly corrupted

Hadamard matrix; we conjecture that the high sign rigidity of uncorrupted Hadamard matrices [43] is preserved under this particular corruption.

It is also fascinating to study the limit of the quantum advantage in communication complexity problems, especially in problems that are both interesting on their own and require new lower bounding techniques. The generalized "Hidden Matching Problem" to arbitrary group size is one such example: this original problem serves as the first exponential separation between quantum and classical one-way communication complexity [10] and is also intimately related to the quantum argument for the exponential lower bound of 2-query Locally Decodable Codes (LDCs) [71]. In [103], with Yaoyun Shi and Wei Yu, we illustrate that this kind of quantum-classical separation is unique for group size two, and rule out the possibility of extending the quantum argument of super-polynomial lower bounds for LDCs to more queries. Our proofs are new applications of the matrix hypercontractive inequalities [14].

- **Sum of squares and representation of quantum convex sets**. This project tries to build the connection between the sum of squares relaxation of the general positive polynomials, a pure mathematical field related to Hilbert 17th problem, and the representation of interesting convex sets in quantum information. Note that we are only interested in those convex sets that cannot be represented by a finite number of hyperplanes, i.e., not polyhedral. Our first result systematically investigates such a connection about the set of separable states. Previously, one such connection appeared implicitly in [11]. Moreover, we prove that a version of the quantum de Finetti theorem follows from the universal denominator argument in the approach of [95] to Hilbert 17th problem. We also recover several properties about small dimension or low rank PPT states.

- **Quantum Correlations**. The set of quantum correlations generated by shared entan-

glement between distant parties is an important but mysterious object. It is closely connected to *quantum non-locality*, an intrinsic feature of quantum mechanics. However, the only known clean characterization of quantum correlations is for XOR games through Tsirelson's bound [29]. Fundamental questions, such as whether there is a bound on the amount of entanglement to approximate any given quantum correlation, remain open.

In an on-going work with Zhengfeng Ji, we partially answer the above question by giving an explicit upper bound on the amount of entanglement for approximating any correlation in a relaxed model called *compatible games*. In this new model, the tensor product structure among the Hilbert spaces of different parties is not enforced, but rather replaced by a weak commutation condition. Interesting instances such as XOR games and unique games can be characterized in this model, although it is not clear how it connects to general entangled non-local games.

We are also interested in the computational hardness of approximating the value of non-local games, in which we focus on the difference between non-local games with classical correlations and those with quantum correlations. A prominent example is that approximating the value of unique games (a problem conjectured to be NP-hard) becomes polynomial-time solvable in the presence of shared entanglement [70]. In an on-going work with Kai-Min Chung and Fang Song, we confirm that the NP-hardness of approximating the value of projection games remains with shared entanglement.

- **Device-independent Quantum Cryptography**. Device-independent quantum cryptography is a recent active research topic. It studies the scenario to perform quantum cryptographic tasks (such as quantum key distributions and so on) without trusting quantum devices, which could be potentially manufactured by a malicious adversary. In an on-going project with Kai-Min Chung and Yaoyun Shi, we provide a device-

independent protocol to extract uniform randomness from any arbitrary weak random source through deterministic operations, which is totally impossible in the classical setting without the help of untrusted quantum devices.

8.2 Future Directions

8.2.1 Continued Effort on Quantum Refereed Games

In Chapter 5, we fully characterize the power of QRG(2). However, the power of QRG(k) (as well as RG(k)) remains unclear for $k \geq 3$. In particular, we would like to investigate the $k \geq 3$ case in the following two aspects.

Relation between EXP **and** $\mathrm{QRG(k)}$

It is well-known that quantum interactive proofs admit three-turn protocols [74] (i.e., $\mathrm{QIP} = \mathrm{QIP(3)}$), while an interaction of a polynomial number of turns is necessary in the classical case. A natural question is to ask whether such a phenomenon is also true in the context of refereed games. Following the protocols in [40], the classical refereed games seem to require a polynomial number of turns in order to simulate EXP. It is unclear whether a short (i.e., with a small number of turns) quantum refereed game protocol can simulate an arbitrary polynomial-turn protocol analogously to [74]. It is also unclear whether one can directly make use of the public-coin version of the refereed games to simulate EXP. On the other side, one can also try to show that $\mathrm{QRG(k)}$ for constant $k \geq 3$ is strictly contained in EXP.

Attack Plan: one possible approach is to compress the classical refereed game protocols that simulate EXP [40] into a small number of turns by using quantum protocols. Another possibility is to make use of the public-coin protocols in the way similar to Watrous's work, which is, however, excluded by our partial result [53]. One can also try to apply other techniques, such as an ingenious improvement of the scheme that simulates QIP by

$\mathrm{QIP}(3)$ [74], to our case.

On the other side, one can also try to design algorithms that simulate quantum refereed games of a small number of turns more efficiently (i.e., better than exponential time), and thus exclude the possibility of QRG=QRG(k) for sub-polynomial ks. The main difficulty is to find a good representation of the alternative interaction with two provers for multiple turns. Competing-prover quantum games of multiple turns admit a natural representation by *quantum strategies* [52], while working with such representations is a task fraught with difficulty [61]. Alternatively, we use Kitaev's *transcript* representation [72] to characterize the multi-turn interaction with one prover and then the multi-turn interaction with another prover [53] separately. However, it is unclear whether such a representation can be extended to the interaction of three or more turns. Even if such an extension exists, it is conceivable that a more sophisticated rounding method, other than the Bures metric one [53], will be necessary.

Relation between $\mathrm{QRG}(\mathrm{k})$ and other quantum complexity classes

It is also natural to ask about the relative expressive power within quantum complexity classes. For instance, $\mathrm{QIP}(2)$ and $\mathrm{QRG}(1)$ are both known to be contained in PSPACE, but the relation between these two are not known.

Attack Plan: we would like to investigate whether the equilibrium value formulation of QIP(2), proposed in the alternative proof of $\mathrm{QIP}(2) \subseteq$ PSPACE in Section 3.3, can be carried out by a one-turn quantum refereed game protocol or its reasonable variant.

8.2.2 Continued Effort on QMA(2)

We already discussed the upper bound of a variant of the computational class QMA(2). This project is a continued effort in that direction. In particular, we believe that a proof of a better upper bound of QMA(2) might need to utilize an improved technique in the

following three directions, separately or jointly. On the other hand, a better understanding about any following topic is also interesting on its own right.

Manipulation of the protocol

A good QMA(2) protocol not only provides a special Q to Problem 6.1.1 that might have an efficient solution, but also provides a large gap between completeness and soundness. Partial results also suggest this possibility. It is known [2] that a *strong amplification* scheme of QMA(2) protocols, if exists, would imply the collapse of QMA(2) to QMA. In [55], Harrow *et al.* show any QMA(2) protocol is equivalent to a canonical protocol where the only operation applied to the separable proofs at the same time is the SWAP-test. As a result, they obtain a *week amplification* scheme to get better bounds on soundness and completeness.

Algorithms with large allowed errors

The NP-hardness of Problem 6.1.1 does not exclude the possibility of the existence of efficient algorithms for large additive errors. In fact, it is a common situation that certain combinatorial optimization problems are NP-hard to solve *exactly* but can be efficiently *approximated* with large additive errors. An ambitious target is to develop a generic algorithm to solve Problem 6.1.1 in this scenario. However, one can take a more humble approach by developing an efficient algorithm based on special protocols or finding a polynomial time approximation scheme (PTAS) rather than an efficient algorithm. Such a PTAS does exist when Q is restricted to certain structure. Furthermore, it comes to our attention that similar problems are actively studied in the field of operations research [24] under the name of "Bi-Quadratic Optimization over Unit Spheres". The latter problem falls into the category of homogenous polynomial optimization with quadratic constraints, which is an interesting and actively studied instance of the so-called polynomial optimiza-

tion. The connection between polynomial optimization and theoretical computer science has been observed in many places, especially in the recent study of the hardness of approximation and the unique game conjecture.

Good relaxation and rounding schemes for separable states

As a potential quantum approach of obtaining good algorithms, we want to highlight the possible contribution of a better understanding of separable states to a good relaxation and rounding scheme for solving Problem 6.1.1. There are a few well-known approaches to relax the separability requirement, such as the Positive Partial Transpose (PPT) condition. Unfortunately, almost for every such relaxation, we know the existence of a bad case where a very entangled state (measured in terms of the trace distance from the set of separable states) lies in the feasible set after the relaxation. However, this is not necessarily an issue for the purpose of solving Problem 6.1.1, especially when large errors are allowed. It is possible to develop an approach that is insufficient for the purpose of telling separable states from entangled ones but yet powerful enough for the purpose of distinguishing between the two promises of the optimization problem.

Summary: We would like to make use of the protocol in [55] to obtain a good completeness and soundness separation as our start point. We want to continue the effort of finding a PTAS, especially via the randomized rounding method following the framework of Barvinok's. In particular, we hope the simplified protocol in [55] that only contains SWAP-test applied on both proofs is helpful for this effort. Also, we hope a better understanding of the set of separable states is instrumental to the design of a good relaxation and rounding scheme. We plan to explore further on the connection between criterions of the separability of quantum states and Hilbert 17th problem.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] S. Aaronson. Quantum computing, postselection, and probabilistic polynomial time. Proceedings of the Royal Society A, pages 461(2063):3473–3482, 2005.

[2] S. Aaronson, S. Beigi, A. Drucker, B. Fefferman, and P. Shor. The power of unentanglement. *Theory of Computing*, 5:1–42, 2009.

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

[4] D. Aharonov, A. Kitaev, and N. Nisan. Quantum circuits with mixed states. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing*, pages 20–30, 1998. arXiv:quant-ph/9806029v1.

[5] D. Aharonov and T. Naveh. Quantum NP – a survey. Available as arXiv.org e-Print quant-ph/0210077, 2002.

[6] 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.

[7] 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.

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

[9] 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.

[10] Z. Bar-Yossef, T. S. Jayram, and I. Kerenidis. Exponential separation of quantum and classical one-way communication complexity. *SIAM Journal on Computing*, pages 38(1):366–384, 2008.

[11] B. Barak, F. Brandao, A. Harrow, J. Kelner, D. Steurer, and Y. Zhou. Hypercontractivity, sum-of-squares proofs, and their applications. In *In Proceedings of 44th ACM Symposium on eory of Computing (STOC)*, 2012.

[12] S. Beigi. NP vs $\mathrm{QMA}_{\log(2)}$. *Quantum Information and Computation*, 54(1&2):0141–0151, 2010.

[13] S. Beigi and P. W. Shor. On the complexity of computing zero-error and holevo capacity of quantum channels. arXiv:0709.2090, 2007.

[14] A. Ben-Aroya, O. Regev, and R. de Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and LDC's. Proceedings of IEEE FOCS, 2008.

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

[16] P. Benioff. Quantum mechanical hamiltonian models of turing machines that dissipate no energy. *Physical Review Letters*, page 48:1581, 1982.

[17] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[18] H. Blier and A. Tapp. All languages in np have very short quantum proofs. In *Proceedings of the ICQNM*, pages 34–37, 2009.

[19] M. Blum, A. K. Chandra, and M. N. Wegman. Equivalence of free boolean graphs can be decided probabilistically in polynomial time. *Information Processing Letters*, pages 10(2):80–82, 1980.

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

[21] F. G. S. L. Brandão. *Entanglement Theory and the Quantum Simulation of Many-Body Physics*. PhD thesis, Imperial College, 2008.

[22] F. G. S. L. Brandão, M. Christandl, and J. Yard. A quasipolynomial-time algorithm for the quantum separability problem. In *Proceedings of the 43rd annual ACM Symposium on Theory of Computation (STOC'11)*, page 343, 2011.

[23] A. Broadbent and E. Kashefi. Parallelizing quantum circuits. arXiv:0704.1736, 2007.

[24] J. Nie C. Ling, L. Qi and Y. Ye. Bi-quadratic optimization over unit spheres and semidefinite programming relaxations. *SIAM Journal on Optimization*, 20(3):1286–1310, 2009.

[25] A. Chailloux and O. Sattath. The complexity of the separable hamiltonian problem. arXiv:1111.5247, 2011.

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

[27] A. Chiesa and M. Forbes. Improved soundness for QMA with multiple provers. arXiv:1108.2098, 2011.

[28] A. M. Childs, J. Preskill, and J. Renes. Quantum information and precision measurement. *Journal of Modern Optics*, 47:155–176, 2000.

[29] B. S. (Tsirelson) Cirel'son. Quantum generalizations of Bell's inequality. *Letters in Mathematical Physics*, 4(2):93–100, 1980.

[30] R. Cleve and J. Watrous. Fast parallel circuits for the quantum fourier transform. In *Proc. 41st ACM Symposium on the Theory of Computing*, pages 526–536, 2000.

[31] S. Cook. The complexity of theorem proving procedures. *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, 1971.

[32] C. M. Dawson and M. A. Nielsen. The solovay-kitaev algorithm. arXiv:quant-ph/0505030, 2005.

[33] R. de Wolf. Lower bounds on matrix rigidity via a quantum argument. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, pages 62–71, 2006.

[34] A. Defant and K. Floret. *Tensor norms and operator ideals*. North Holland, 1992.

[35] D. Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London*, A400:97–117, 1985.

[36] D. Deutsch and R. Jozsa. Rapid solutions of problems by quantum computation. *Proceedings of the Royal Society of London*, A439:553–558, 1992.

[37] A. C. Doherty, P. A. Parrilo, and F. M. Spedalieri. A complete family of separability criteria. *Phys. Rev. A*, 69:022308, 2004.

[38] S. Nakagawa F. L. Gall and H. Nishimura. On QMA protocols with two short quantum proofs. arXiv:1108.4306, 2011.

[39] K. Fan. Minimax theorems. *Proceedings of the National Academy of Sciences*, 39:42–47, 1953.

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

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

[42] R. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1983.

[43] J. Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *J. Comput. Syst. Sci.*, 65:612–625, 2002.

[44] L. Fortnow, R. Impagliazzo, V. Kabanets, and C. Umans. On the complexity of succinct zero-sum games. *Computational Complexity*, pages 17(3):353–376, 2008.

[45] C. Fuchs and J. van de Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.

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

[47] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(1):691–729, 1991.

[48] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In S. Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 73–90. JAI Press, 1989.

[49] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer–Verlag, 1988.

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

[51] G. Gutoski and J. Watrous. Quantum interactive proofs with competing provers. In *Proceedings of the 22nd Symposium on Theoretical Aspects of Computer Science (STACS'05)*, volume 3404 of *Lecture Notes in Computer Science*, pages 605–616. Springer, 2005. arXiv:cs/0412102v1 [cs.CC].

[52] G. Gutoski and J. Watrous. Toward a general theory of quantum games. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC 2007)*, pages 565–574, 2007. arXiv:quant-ph/0611234v2.

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

[54] S. Hallgren, A. Kolla, P. Sen, and S. Zhang. Making classical honest verifier zero knowledge protocols secure against quantum attacks. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, pages 592–603, 2008.

[55] A. Harrow and A. Montanaro. An efficient test for product states, with applications to quantum merlin-arthur games. In *Proceedings of IEEE 51st Annual Symposium on Foundations of Computer Science (FOCS'10)*, page 633, 2010.

[56] A. Harrow, B. Recht, and I. L. Chuang. Efficient discrete approximations of quantum gates. *J. Math. Phys.*, page 43:4445, 2002.

[57] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Rev. Mod. Phys.*, 81:865, 2009.

[58] L. M. Ioannou. Computational complexity of the quantum separability problem. *Quantum Information and Computation*, 7:335, 2007.

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

[60] R. Jain, Z. Ji, S. Upadhyay, and J. Watrous. QIP=PSPACE. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 573–582, 2010. arXiv:0907.4737v2 [quant-ph].

[61] R. Jain, S. Upadhyay, and J. Watrous. Two-message quantum interactive proofs are in PSPACE. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 534–543, 2009. arXiv:0905.1300v1 [quant-ph].

[62] R. Jain and J. Watrous. Parallel approximation of non-interactive zero-sum quantum games. In *Proceedings of the 24th IEEE Conference on Computational Complexity (CCC 2009)*, pages 243–253, 2009. arXiv:0808.2775v1 [quant-ph].

[63] R. Jain and P. Yao. A parallel approximation algorithm for positive semidefinite programming. In *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS 2011)*, pages 463–471, 2011. arXiv:1104.2502v1 [cs.CC].

[64] R. Jain and P. Yao. A parallel approximation algorithm for mixed packing and covering semidefinite programs. arXiv:1201.6090v1 [cs.DS], 2012.

[65] D. Janzing, P. Wocjan, and T. Beth. Non-identity check is QMA-complete. *International Journal of Quantum Information*, pages 3(3):463–473, 2005.

[66] Z. Ji and X. Wu. Non-identity check remains qma-complete for short circuits. In *the Asian Conference on Quantum Information Science*, 2009.

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

[68] J. Kempe, H. Kobayashi, K. Matsumoto, and T. Vidick. Using entanglement in quantum multi-prover interactive proofs. *Computational Complexity*, 18(2):273–307, 2009.

[69] J. Kempe and O. Regev. 3-local hamiltonian is QMA-complete. *Quantum Information and Computation*, pages 3(3):258–264, 2003.

[70] J. Kempe, O. Regev, and B. Toner. Unique games with entangled provers are easy. In *Proceedings of the 49th Annual Symposium on Foundations of Computer Science*, pages 457–466, 2008. arXiv:0710.0655v2 [quant-ph].

[71] I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, 2003.

[72] A. Kitaev. Quantum coin-flipping. Presentation at the 6th Workshop on *Quantum Information Processing* (QIP 2003), 2002.

[73] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002.

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

[75] A. Y. Kitaev. Quantum computations: algorithms and error correction. Russian Mathematical Surveys, pages 52(6):1191–1249, 1997.

[76] H. Kobayashi. General properties of quantum zero-knowledge proofs. In *Proceedings of the Fifth IACR Theory of Cryptography Conference*, pages 107–124, 2008.

[77] H. Kobayashi and K. Matsumoto. Quantum multi-prover interactive proof systems with limited prior entanglement. *Journal of Computer and System Sciences*, 66(3):429–450, 2003. arXiv:cs/0102013v5 [cs.CC].

[78] H. Kobayashi, K. Matsumoto, and T. Yamakami. Quantum certificate verification: Single versus multiple quantum certificates. quant-ph/0110006, 2001.

[79] H. Kobayashi, K. Matsumoto, and T. Yamakami. Quantum merlin-arthur proof systems: Are multiple merlins more helpful to arthur? Lecture Notes in Computer Science, 2906:189–198, 2003.

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

[81] D. Koller, N. Megiddo, and B. von Stengel. Fast algorithms for finding randomized strategies in game trees. In *Proceedings of the 26th ACM Symposium on Theory of Computing (STOC 1994)*, pages 750–759, 1994.

[82] L. Levin. Universal search problems. Problems of Information Transmission, pages 9(3):265–266, 1973.

[83] Y.-K. Liu. Consistency of local density matrices is QMA-complete. In *Proc. RANDOM 2006*, pages 438–449, 2006.

[84] Y.-K. Liu, M. Christandl, and F. Verstraete. N-representability is QMA-complete. *Physical Review Letters*, page 98, 2007.

[85] M. Luby and N. Nisan. A parallel approximation algorithm for positive linear programming. In *Proceedings of the 25th ACM Symposium on Theory of Computing (STOC 1993)*, pages 448–457, 1993.

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

[87] C. Marriott and J. Watrous. Quantum Arthur-Merlin games. In *Proceedings of the 19th Annual Conference on Computational Complexity*, pages 275–285, 2004.

[88] C. Marriott and J. Watrous. Quantum Arthur-Merlin games. *Computational Complexity*, 14(2):122–152, 2005. arXiv:cs/0506068v1 [cs.CC].

[89] N. Megiddo. A note on approximate linear programming. *Information Processing Letters*, page 42(1):53, 1992.

[90] C. Moore and M. Nilsson. Parallel quantum computation and quantum codes. Available as arXiv.org e-Print quant-ph/9808027, 1998.

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

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

[93] R. Peng and K. Tangwongsan. Faster and simpler width-independent parallel algorithms for positive semidefinite programming. arXiv:1201.5135v1 [cs.DS], 2012.

[94] Y. Feng R. Duan and M. Ying. Entanglement is not necessary for perfect discrimination between unitary operations. *Physical Review Letters*, page 98(10):100503, 2007.

[95] B. Reznick. Uniform denominators in hilbert's seventeenth problem. *Mathematische Zeitschrift*, pages 220: 75–98.

[96] B. Rosgen. Distinguishing short quantum computations. In *Proceedings of the 25th STACS*, pages 597–608, 2008.

[97] B. Rosgen and J. Watrous. On the hardness of distinguishing mixed-state quantum computations. In *Proceedings of the 20th Annual Conference on Computational Complexity*, pages 344–354, 2005. arXiv:cs/0407056v1 [cs.CC].

[98] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, pages (4):701–717, 1980.

[99] M. Serna. Approximating linear programming is log-space complete for P. *Information Processing Letters*, pages 37(4):233–236, 1991.

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

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

[102] Y. Shi and X. Wu. Epsilon-net method for optimizations over separable states. In *the 15th Workshop on Quantum Information Processing (QIP 2012)*, 2011.

[103] Y. Shi, X. Wu, and W. Yu. Limits of quantum one-way communication by matrix hypercontractive inequalities. manuscript, 2012.

[104] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[105] M. Mosca T.-C. Wei and A. Nayak. Interacting boson problems are QMA-hard. arXiv:0905.3413, 2009.

[106] L. Trevisan and F. Xhafa. The parallel complexity of positive linear programming. Parallel Processing Letters, pages 8(4):527–533, 1998.

[107] J. von Neumann. Zur theorie der gesellschaftspiele. *Mathematische Annalen*, 100(1):295–320, 1928. In German.

[108] J. von zur Gathen. Parallel linear algebra. *In J. Reif, editor, Synthesis of Parallel Algorithms*, 1993.

[109] S. Wehner W. Matthews and A. Winter. Distinguishability of quantum states under restricted families of measurements with an application to quantum data hiding. *Comm. Math. Phys.*, page 291, 2009.

[110] J. Watrous. PSPACE has constant-round quantum interactive proof systems. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 112–119, 1999.

[111] J. Watrous. *Lecture Notes on Theory of Quantum Information*. 2008.

[112] J. Watrous. Zero-knowledge against quantum attacks. SIAM Journal on Computing, 39(1):25–58, 2009.

[113] X. Wu. Equilibrium value method for the proof of QIP=PSPACE. arXiv:1004.0264v2 [quant-ph], 2010.

[114] X. Wu. Parallized solutions to semidefinite programmings in quantum complexity theory. arXiv:1009.2211v3 [quant-ph], 2010.

[115] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science*, pages 352–361, 1993.

[116] N. Young. Sequential and parallel algorithms for mixed packing and covering. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 538–546, 2001.

[117] R. Zippel. Probabilistic algorithms for sparse polynomials. In *International Symposium on Symbolic and Algebraic Computation, Lecture Notes in Computer Science*, pages 216–226, 1979.