Quantum Complexity Theory
Summer 2019, University of Paderborn

Sevag Gharibian

Lecture 8: Quantum Interactive Proofs (QIP), semidefinite programs, and multiplicative weights

"There is one thing stronger than all the armies in the world, and that is an idea whose time has come."
— Victor Hugo

Contents

Introduction. In this course, we have considered the power of quantum verification systems in the presence of various types of communication: No communication (BQP), one-way classical communication (QCMA), and one-way quantum communication (QMA). A natural extension of these studies is to ask: *What happens when the communication is interactive, i.e. back and forth between the prover and verifier?* This is roughly the class QIP, and the high-level aim of this lecture is to study the result that QIP = PSPACE.

Along the way, we will encounter two elegant ideas which have found surprisingly fruitful uses in quantum information (as the opening quote of this section suggests, it is these ideas whose "time has come"). The first is the notion of *semidefinite programs (SDP)*, which are a fairly broad class of optimization problems typically solvable in polynomial-time, and which have become rather ubiquitous in quantum information. The second is the *multiplicative weights* (MW) method, which in some sense is the oldest trick in the book in terms of how humans handle unpredictable situations (at least according to the present instructor). Remarkably, both these tools come together to prove that polynomially many messages of interaction with a quantum verifier yields precisely PSPACE.

Organization. We begin in Section 1 with the multiplicative weights algorithm; while the context we present it in is unrelated to QIP, our discussion will nevertheless hopefully give you an intuition as to how the method arises. Section 2 then introduces quantum interactive proofs, semidefinite programs, and how the former can be phrased in terms of the latter. Section 3 finally gives the proof combining all the previous ingredients to show that QIP = PSPACE. We stress that this lecture is full of deep and fundamental ideas and tools; the MW method, SDPs, and interactive proofs alone can each fill many lectures many times over.

1 The Multiplicative Weights algorithm

In a nutshell, the Multiplicative Weights (MW) algorithm captures the age-old idea that *"if it worked once, it's likely to work again"*. Typically, the algorithm is introduced in the context of the stock market and stock advisors, but let us focus here on a more troublesome bunch: Professors. Suppose you are a first-year student entering a Computer Science program at your university, and at a loss as to which of many advanced courses to choose. Confused, you assemble an A-team of professors, labelled 1 through n, to guide you through this process. In your first semester, you ask: "Professors, professors, heed my call, which is the fairest course of them all?" Each professor gives you an answer; but since you're new to the school, you have no idea whose advice to take. So in round 1, you pick some professor $i_1 \in [n]$ uniformly at random, and follow his/her advice. Once the semester is over, you reflect: *Was Professor i's advice good or bad?* If the advice was good, then once semester 2 starts and you repeat this process, you will naturally be more likely to follow Professor i's advice again. The question is: *How should you update your probability of following Professor i's advice at the end of each semester?* It turns out the right answer is "multiplicatively", and this is precisely where MW gets its name.

Formal setup and algorithm. Imagine we have a set E of n experts, and T rounds of some process for which we wish to take the experts' advice into account. In each round $t \in [T]$, we have a probability distribution p^t over E, such that we pick and follow (only) E_i's advice in round t with probability p_i^t. We imagine that the environment now assigns a "cost" to each expert i's choice in round t, denoted $-1 \leq c_i^t \leq 1$. Absolutely no assumptions are made on how these costs are set by the environment; they may even be set adversarially.

The MW algorithm is an *online* algorithm, meaning it tries to make the best choices at each step *without* access to future environmental data (i.e. costs c_i^t for future rounds t). Ideally, the goal would be to pick the "best expert" from the beginning, i.e. the one attaining minimum cost

$$\min_{i\in[n]} \sum_{t=1}^{T} c_i^t,$$

and follow this expert in each round. This is naturally impossible, but remarkably we can get "close" to OPT, *even if the environment acts adversarially.* To formalize this, let $c^t \in [-1,1]^n$ and $p^t \in [0,1]^n$ denote the vectors encoding the cost and probability for all experts in a round t. (Thus, the odds of picking expert i in round t are $p_i^t \in [0,1]$, and the cost of doing so is $c_i^t \in [-1,1]$.) The *expected* cost of using distribution p^t in round t is hence

$$C_t := \sum_{i=1}^{n} p_i^t c_i^t = \left\langle c^t, p^t \right\rangle,$$

for $\langle a, b\rangle$ the inner product of a and b. The expected cost over *all* rounds is thus $C := \sum_{t=1}^{T} C_t = \sum_{t=1}^{T} \langle c^t, p^t\rangle$. Before stating the algorithm, let us state what it gives us.

Theorem 1. *Fix $0 < \epsilon \leq 1/2$. Then, for any expert E_i, after T rounds the MW algorithm obtains expected cost*

$$C \leq \sum_{t=1}^{T} c_i^t + \left[\epsilon \sum_{t=1}^{T} \left|c_i^t\right| + \frac{\ln n}{\epsilon}\right].$$

Remarks. We stress that the right hand side of the bound above holds for *any expert E_i*; thus, the MW algorithm is "close" to the best expert in terms of performance. The term in square brackets is the *additive error* term; it depends on the magnitude of the costs $|c_i^t|$ incurred by E_i.

The algorithm itself is now stated as Algorithm 1 (yes, it really is this simple).

Exercise. Why is there a division involved in Step 2(a) of Algorithm 1?

Proof of Theorem 1. As done in amortized analysis, it turns out to be useful to study a "potential function" as a "reference point": $\Phi^t := \sum_i w_i^t$. By giving upper and lower bounds on Φ, we will obtain the claim.

Algorithm 1 The Multiplicative Weights algorithm.

```
1. Fix $0 < \epsilon \leq 1/2$, and give each expert $E_i$ "weight" $w_i^t = 1$.
2. For $t = 1, \ldots, T$:
    (a) Pick expert $E_i$ with probability $w_i^t / (\sum_t w_i^t)$.
    (b) Obtain costs $c^t$ for round $t$ from the environment.
    (c) Update weight of expert $E_i$ as $w_i^{t+1} := \begin{cases} w_i^t(1-\epsilon)^{c_i^t} & \text{if } c_i^t \geq 0 \\ w_i^t(1+\epsilon)^{-c_i^t} & \text{if } c_i^t < 0 \end{cases}$.
```

Upper bound. By definition,

$$\Phi^{t+1} = \sum_i w_i^{t+1} \leq w_i^t(1 - \epsilon c_i^t) = \Phi^t(1 - \epsilon \langle c^t, p^t \rangle) \leq \Phi^t e^{-\epsilon \langle c^t, p^t \rangle},$$

where the first inequality follows from the next exercise.

Exercise. Prove that $(1 \mp \epsilon)^{\pm x} \leq (1 - \epsilon x)$ when $x \in [-1, 1]$ and $x \in [-1, 0]$, respectively. Use these facts to obtain the first inequality above.

Exercise. Use the fact that $p_i^t = w_i^t / \Phi^t$ to show the second equality above.

Exercise. Prove that $e^{-x} \geq 1 - x$ for $x \in [-1, 1]$. Use this to prove that last inequality above. (Hint: You need to use an assumption we made on the costs c_i^t.)

Since by definition $\Phi^1 := n$, it follows that after T steps, $\Phi^{T+1} \leq n e^{-\epsilon \sum_{t=1}^{T} \langle c^t, p^t \rangle}$, our desired upper bound.

Lower bound. Since the weights $w_i^t \geq 0$, we have that for any E_i, $\Phi^{T+1} \geq w_i^{t+1}$. Combining the upper bound above with the following exercise now yields the claim.

Exercise. Prove that for $0 \leq \epsilon \leq 1/2$, $\ln(1/(1 - \epsilon)) \leq \epsilon + \epsilon^2$ and $\ln(1 + \epsilon) \geq \epsilon - \epsilon^2$.

Exercise. Use the exercise above to prove that $\ln(w_i^{T+1}) \geq -\epsilon \sum_i c_i^t - \epsilon^2 \sum_i |c_i^t|$. Combine this with our upper bound to obtain the claim. ☐

This completes our discussion of the basic MW method. For a gentle introduction to its extension to the matrix setting, the reader is referred to the thesis of Kale, available at `http://www.satyenkale.com/papers/thesis.pdf`. In the interest of time, we shall instead jump right into QIP and semidefinite programs.

2 QIP and semidefinite programs

Ultimately, our goal is to use a matrix variant of the MW algorithm to show that QIP = PSPACE. To do so, we first require a definition of *quantum interactive proofs*, and subsequently an approach for embedding their maximum acceptance probability into a *semidefinite program*. These are given in Sections 2.1 and 2.2, respectively.

2.1 Quantum interactive proofs

Roughly, classical interactive proofs are the natural extension of NP to the setting in which the prover (which remains computationally unbounded) can now exchange polynomially many rounds of communication with the verifier (which remains polynomially bounded). Quantumly, the idea is analogous: The quantum prover (which remains computationally unbounded, with the exception of obeying the laws of quantum mechanics) now exchanges polynomially many rounds of quantum communiction with the verifier (which is still computationally bounded). Slightly more formally, we imagine the prover and verifier send a common quantum "message register" M back and forth, and each take turns applying a local unitary circuit on M and their private workspace.

Formal definition. To make this intuition formal, we must generalize our definition of a QMA verifier to an m-round quantum verifier (and analogously, an m-round quantum prover).

Definition 2. *(m-round quantum verifier) An m-round quantum verifier is a P-uniform circuit family $Q = \{Q_{n,1}, \dots Q_{n,m}\}$, acting on three registers: An input register A containing $x \in \{0,1\}^n$, a message register M consisting of $p(n)$ qubits, and an ancilla or "private" register V consisting of $q(n)$ qubits, for some polynomials $p, q : \mathbb{N} \mapsto \mathbb{N}$. We imagine the verifier acts in "rounds", applying circuit $Q_{n,i}$ in round $i \in [m]$. Before round 1, the message and private registers are initialized to all zeroes.*

Definition 3. *(m-round quantum prover) An m-round quantum prover is defined identically to an m-round verifier, except the circuit family $Q = \{Q_{n,1}, \dots Q_{n,m}\}$ need not be P-uniform (indeed, the circuits can in be superpolynomial in size). To help distinguish between verifier and prover, for the latter we label the private register as P for "prover" (versus V for "verifier").*

Exercise. Why does dropping the P-uniformity condition in Definition 3 capture the notion of a prover which is limited only by the laws of quantum mechanics?

For brevity, we henceforth drop the input size n when referring to circuits $Q_{n,i}$. We can now model an interactive protocol between an m-round verifier and m-round prover in the natural way: Letting $\{Q_i\}$ and $\{R_i\}$ denote their respective circuits, the interaction is given by (for simplicity, we omit the A register, as we may assume without loss of generality that its contents remain fixed to the input x for both parties):

$$(Q_m)_{V,M}(R_m)_{M,P} \cdots (Q_1)_{V,M}(R_1)_{M,P}|0\cdots 0\rangle_V|0\cdots 0\rangle_M|0\cdots 0\rangle_P.$$

Just as in QMA, the verifier now measures a designated output qubit of register V, say V_1, and accepts if and only if the output is 1.

Exercise. Suppose we wish the interactive protocol to instead begin with a message from the verifier to the prover; how can we model that in the setup above?

Exercise. Why can we assume without loss of generality that the verifier acts last?

We may now formally define the class QIP.

Definition 4. *(Quantum Interactive Proof Systems (QIP)) A promise problem $\mathbb{A} = (A_{\text{yes}}, A_{\text{no}}, A_{\text{inv}})$ is in QIP if there exists a polynomial $m : \mathbb{N} \mapsto \mathbb{N}$ and m-round quantum verifier satisfying the following for any input $x \in \{0,1\}^n$:*

- *(Completeness/YES case) If $x \in A_{\text{yes}}$, there exists an m-round quantum prover causing the verifier to accept with probability at least $2/3$.*
- *(Soundness/NO case) If $x \in A_{\text{no}}$, then for all m-round quantum provers, the verifier accepts with probability at most $1/3$.*
- *(Invalid case) If $x \in A_{\text{inv}}$, the verifier may accept or reject arbitrarily.*

Magic: Only a constant number of rounds are needed. Remarkably, and in strong contrast to classical interactive proofs, it turns out quantumly that the use of polynomially many rounds of communication in QIP is overkill — it suffices to use just 2 rounds:

$$(Q_2)_{V,M}(R_2)_{M,P}(Q_1)_{V,M}(R_1)_{M,P}|0\cdots 0\rangle_V|0\cdots 0\rangle_M|0\cdots 0\rangle_P.$$

In fact, we may even assume the only message from the verifier to the prover (modelled by Q_1) is an unbiased coin flip (i.e. not conditioned on the action of R_1). This is known as a "Quantum Arthur-Merlin" game, and in the context of QIP, we may assume the accompanying completeness and soundness parameters are 1 and $1/2+\epsilon$ for any fixed $\epsilon > 0$, respectively. For the remainder of this lecture, we shall henceforth use this Quantum Arthur-Merlin characterization of QIP.

Exercise. Since Q_1 above is just an unbiased coin flip independent of R_1, at first glance it may seem one can simply remove R_1 altogether. Why might this be a bad idea?

2.2 Semidefinite programming

Linear programming and its generalization, semidefinite programming, are both misnomers: Neither of them has anything to do "programming" in the usual computer science sense. Both actually refer to a fairly broad class of optimization problems with a wide variety of applications. Generally, a linear program (LP) attempts to maximize a given linear function $f : \mathbb{R}^n \mapsto \mathbb{R}$, subject to a set of given linear inequality constraints. The main syntactic difference in generalizing from an LP to a semidefinite program (SDP) is that now the variables are not vectors in $\mathbb{R}^n$, but Hermitian *matrices* in $\mathcal{L}(\mathbb{C}^n)$, and hence the correct notion of inequality is the generalized inequality $\succeq$ denoting the positive semidefinite ordering (i.e. $A \succeq B$ if and only if $A - B$ is positive semidefinite).

Exercise. Is $I \succeq 2I$? How about $I \succeq X$ for Pauli X? $I \succeq 2X$? If $A \succeq B$ and $C \succeq D$, is $A + C \succeq B + D$?

Exercise. If $A \succeq 0$ and $B \succeq 0$, is it always true that $AB \succeq 0$? (Hint: A necessary condition for AB to be positive semidefinite is for it to be Hermitian.)

Standard form for SDPs. We begin by stating the standard form for SDPs we shall use. To do so, we need three things: (1) A *cost matrix* $C \in \text{Herm}(\mathcal{X})$, (2) a *constraint matrix* $D \in \text{Herm}(\mathcal{Y})$, and (3) a linear *constraint map* $\Psi : \text{Herm}(\mathcal{X}) \mapsto \text{Herm}(\mathcal{Y})$. Here, $\mathcal{X}$ and $\mathcal{Y}$ are fixed complex vector spaces. The (primal) semidefinite program is then given by:

Primal problem (P)		Dual problem (D)	
supremum:	$\text{Tr}(CX)$	infimum:	$\text{Tr}(DY)$
subject to:	$\Psi(X) \preceq D$,	subject to:	$\Psi^*(Y) \succeq C$,
	$X \succeq 0$,		$Y \succeq 0$.

This may look cryptic at first sight, so let us first break down some facts about the primal problem, P:

- The *variable* being optimized over is $X \in \text{Herm}(\mathcal{X})$. The *feasible region* is the set of all "valid assignments", i.e. all X satisfying the given constraints, $\Psi(X) \preceq D$ and $X \succeq 0$.
- The *objective function* being maximized, $\text{Tr}(CX)$, is linear in X. In analogy with Section 1, we may use inner product notation $\langle C, X\rangle := \text{Tr}(C^\dagger X)$ (recall C is Hermitian in our setting).

  Exercise. Prove that the objective function is indeed linear in X.

Exercise. The function $\langle A, B\rangle := \mathrm{Tr}(A^\dagger B)$ is sometimes called the Hilbert-Schmidt inner product for matrices. Why does this name make sense? (Hint: How is this function really just an example of the usual vector inner product?)

- The map Ψ must be *Hermiticity preserving*, i.e. map Hermitian operators to Hermitian operators.

  Exercise. Why is the requirement that Ψ be Hermiticity preserving necessary?

- Once an SDP is in the standard form P above, one can formulate the corresponding *dual* problem, D, which is also an SDP. We shall say more about duality shortly, but for now let us define the *adjoint* map $\Psi^* : \mathrm{Herm}(\mathcal{Y}) \mapsto \mathrm{Herm}(\mathcal{X})$; it is the unique linear map satisfying $\langle A, \Psi(B)\rangle = \langle \Psi^*(A), B\rangle$ for all $A \in \mathcal{L}(\mathcal{Y}), B \in \mathcal{L}(\mathcal{X})$.

  Exercise. Let $\Psi : \mathbb{C}^n \mapsto \mathbb{C}$ be the trace function. What is Ψ^*?

The topic of semidefinite programming, and its generalization to convex optimization, fills up entire textbooks. Here, we shall aim to convey the basic intuition and facts needed for studying QIP via some examples and key statements.

Examples and optimal solutions. As with most things in life, you have actually been using SDPs without knowing it. The simplest example of an SDP which you know is the computation of the largest eigenvalue of a Hermitian matrix $C \in \mathrm{Herm}(\mathbb{C}^n)$, which can be written:

Primal problem (P)		Dual problem (D)	
supremum:	$\mathrm{Tr}(CX)$	infimum:	y
subject to:	$\mathrm{Tr}(X) \leq 1$	subject to:	$y \cdot I \succeq C,$
	$X \succeq 0,$		

Exercise. What space does dual variable y live in?

Exercise. Technically, our standard definition of the dual problem required $y \geq 0$. How can we rewrite D above to be in standard form? (Hint: Any $a \in \mathbb{R}$ can be written $a = b - c$ for some $b, c \geq 0$.)

Two comments: (1) It is not necessary to always put an SDP into standard form; this is more for convenience, as it makes computation of the dual SDP easier. (Certain numerical SDP solvers may also request it.) Generally speaking, an SDP is any optimization (min or max) of a linear function, subject to linear inequality constraints and non-negativity constraints (with respect to $\succeq$). (2) The use of *supremum* in P above is unnecessary; in this case, the optimal value is attained and is precisely $\lambda_{\max}(C)$. More generally, however, this is not true, as the following example demonstrates.

$$\begin{aligned} \text{infimum:} \quad & x \\ \text{subject to:} \quad & \begin{pmatrix} x & 1 \\ 1 & y \end{pmatrix} \succeq 0 \\ & x, y \in \mathbb{R} \end{aligned}$$

Exercise. Prove that $x = 0$ is not in the feasible region. Conclude that 0 as an objective function value is not attainable. (Hint: Use the Determinant Test, which states for a 2×2 matrix that $A \succeq 0$ if and only if $A_{11}, A_{22} \geq 0$ and $\det(A) \geq 0$.)

Exercise. Prove that for *any* $x > 0$, there exists a choice of y so that (x, y) is a feasible solution to the dual problem. Conclude that the use of infimum is necessary in this example.

Duality theory. You may be wondering why we've been dragging around the dual problem for each primal problem we've stated. Let p and d denote the optimal values for a primal and corresponding dual SDP, respectively. These values satisfy an amazing property known as *weak duality*:

$$p \leq d.$$

This immediately gives a powerful use for SDPs. Suppose you have some maximization problem Π which is difficult to solve analytically (for example, this might encode the optimal cheating probability for a cryptographic protocol). If you can formulate Π (or a relaxation of it) as an SDP, then it is "easy" to give an upper bound on Π — any feasible solution to the dual SDP will, by weak duality, yield *some* upper bound on Π. Note, crucially, that this does *not* require solving an SDP; one can often make a clever guess as to what a good dual solution should be.

Of course, the natural question is whether this upper bound can be made tight, i.e. is it true that $p = d$? This is called *strong duality.* In general, strong duality unfortunately does not hold. However, a simple sufficient condition for strong duality is Slater's constraint qualification, which states that if (say for the primal) there is a *strictly feasible* solution X (i.e. $\Phi(X) \prec D$ and $X \succ 0$), then strong duality holds.

Exercise. Consider the non-standard dual program below. Show that it does not satisfy strong duality.

$$\begin{aligned} \text{infimum:} \quad & x \\ \text{subject to:} \quad & \begin{pmatrix} 0 & x & 0 \\ x & y & 0 \\ 0 & 0 & x+1 \end{pmatrix} \succeq 0 \\ & x, y \in \mathbb{R} \end{aligned}$$

Runtime. It is a common fallacy that "SDPs can be solved in polynomial time". While the spirit of the statement is true, in order to actually attain a poly-time solution, two constraints must be met: The feasible region must be contained in a ball of radius R, and must contain a ball of radius r. The runtime of the Ellipsoid Algorithm is then polynomial in the input size (i.e. encodings of C, Ψ, D; we assume this encoding size scales at least as the dimension of the space C acts on), $\log R$, $\log(1/r)$, and $\log(1/\epsilon)$, where ϵ is the additive error in the optimal objective function value we are willing to tolerate. In practice, these conditions are typically met. Note also that while the runtime of the Ellipsoid Algorithm is often cited in theoretical algorithmic results relying on SDPs, in practice more stable and modern methods such as Interior Point Methods are deployed. In this lecture, we will see an alternative method for solving certain SDPs; the matrix multiplicative weights method.

2.3 Quantum interactive proofs as SDPs

Having introduced quantum interactive proofs and SDPs, we can now formulate the former as an example of the latter. As stated in Section 2.1, we shall assume a 3-message Quantum Arthur-Merlin protocol, which suffices to capture QIP. The setup is as follows:

1. The prover (Merlin) sends the verifier (Arthur) a density operator σ in register M_1 (for "message 1").
2. Arthur has a pair of measurements $\{P_0, I - P_0\}$ and $\{P_1, I - P_1\}$ in mind, for $0 \preceq P_0, P_1 \preceq I$ and acting on joint space $M_1 \otimes M_2$. Arthur chooses a uniformly random bit $b \in \{0, 1\}$, and sends it to Merlin.
3. Merlin sends quantum register M_2 (for "message 2") to Arthur.
4. Arthur performs measurement $\{P_b, I - P_b\}$ on the message registers $M_1 \otimes M_2$, and accepts if and only if the outcome is P_b.

Formally, one can model the acceptance POVM for Arthur's random measurement via operator

$$Q := |0\rangle\langle 0|_C \otimes (P_0)_{M_1,M_2} + |1\rangle\langle 1|_C \otimes (P_1)_{M_1,M_2}.$$

To see why, note that conditioned on bit flip b, Merlin prepares joint state $\rho_b \in \mathcal{L}(M_1 \otimes M_2)$. In other words, the density operator prepared by Merlin is

$$X := \frac{1}{2}|0\rangle\langle 0|_C \otimes (\rho_0)_{M_1,M_2} + \frac{1}{2}|1\rangle\langle 1|_C \otimes (\rho_1)_{M_1,M_2}. \tag{1}$$

Exercise. Show that the probability of Arthur accepting is $\mathrm{Tr}(QX) = \frac{1}{2}(\mathrm{Tr}(P_0\rho_0) + \mathrm{Tr}(P_1\rho_1))$.

Of course, ρ_0 and ρ_1 cannot be *arbitrary* — in step 1 of the protocol, Merlin committed some state σ on space M_1, which remains untouched throughout the remainder of the protocol. Thus, it must be that both possible end states ρ_0 and ρ_1 agrees on this "commitment space" M_1, i.e.

$$\mathrm{Tr}_{M_2}(\rho_0) = \mathrm{Tr}_{M_2}(\rho_1) = \sigma.$$

Exercise. Consider any pair of purifications $|\psi_0\rangle_{AB}, |\psi_1\rangle_{AB}$ of some density operator σ_B. Prove that there exists a unitary U_A such that $(U_A \otimes I_B)|\psi_0\rangle\langle\psi_0|_{AB}(U_A^\dagger \otimes I_B) = |\psi_1\rangle\langle\psi_1|_{AB}$. Conclude that the restriction to fixing σ on the commitment space M_1 above is without loss of generality (i.e. for any pair of pure states $|\psi_0\rangle$ and $|\psi_1\rangle$ Merlin wants to prepare on $M_1 \otimes M_2$, he may do so (possibly inefficiently), as long as $|\psi_0\rangle$ and $|\psi_1\rangle$ agree on their reduced state σ on M_1).

With these observations in hand, we can state the primal and dual SDPs capturing our interactive protocol, where for convenience we have moved the factor of 1/2 from X to Q:

Primal problem (P)

$$\begin{aligned} \text{maximize:} \quad & \mathrm{Tr}(QX) \\ \text{subject to:} \quad & \mathrm{Tr}_{M_2}(X) \preceq I_C \otimes \sigma_{M_1} \\ & \mathrm{Tr}(\sigma) = 1 \\ & \sigma_{M_1} \succeq 0 \\ & X_{C,M_1,M_2} \succeq 0 \end{aligned}$$

Dual problem (D)

$$\begin{aligned} \text{infimum:} \quad & \|\,\mathrm{Tr}_C(Y)\,\|_\infty \\ \text{subject to:} \quad & Y_{C,M_1} \otimes I_{M_2} \succeq Q, \\ & Y_{C,M_1} \succeq 0. \end{aligned}$$

Again, let us break down the primal SDP:

- The constraints $\mathrm{Tr}(\sigma) = 1$ and $\sigma \succeq 0$ ensure σ is a density operator.
- Ideally, we wish to force X to be of the form in Equation (1) (but without the 1/2), which is block diagonal with respect to C:
$$X = \begin{pmatrix} \rho_0 & 0 \\ 0 & \rho_1 \end{pmatrix}.$$
In principle, we could explicitly enforce this by having separate variables for ρ_0 and ρ_1; however, since Q is also block diagonal with respect to C, the off-diagonal blocks of X do not alter the value of the objective function. Thus, without loss of generality the optimal X sets the off-diagonal blocks to 0 (this technically requires the following exercise).

  Exercise. Prove that if $\begin{pmatrix} A & B \\ B^\dagger & C \end{pmatrix} \succeq 0$, then $\begin{pmatrix} A & 0 \\ 0 & C \end{pmatrix} \succeq 0$. Why is this exercise needed in assuming the off-diagonal blocks of X are 0?

Exercise. Write Q in block form with respect to register C, and confirm it is block-diagonal.

- We must enforce that ρ_0 and ρ_1 have reduced state σ on M_1. To see why the SDP captures this, we require the following exercise.

  Exercise. Prove in Equation (1) that tracing out M_1 yields $I_C \otimes \sigma_{M_1}$, assuming ρ_0 and ρ_1 have reduced state σ on M_1 (and omitting the 1/2 factor).

  This almost explains the last primal SDP constraint, $\mathrm{Tr}_{M_2}(X) \preceq I_C \otimes \sigma_{M_1}$ — here we have an inequality without loss of generality (as opposed to an equality), since any feasible solution to the inequality can be "boosted" to make the inequality tight, while only increasing the objective function value.

  Exercise. Prove the claim above: Suppose $\mathrm{Tr}_{M_2}(X) \preceq I_C \otimes \sigma_{M_1}$ but $\mathrm{Tr}_{M_2}(X) \neq I_C \otimes \sigma_{M_1}$. Give a new operator X' such that $\mathrm{Tr}_{M_2}(X) = I_C \otimes \sigma_{M_1}$ and $\mathrm{Tr}(QX') \geq \mathrm{Tr}(QX)$. (Hint: By assumption, $I_C \otimes \sigma_{M_1} - \mathrm{Tr}_{M_2}(X) \succeq 0$.)

- Finally, we have quietly replaced our use of supremum with maximum.

  Exercise. Prove that Slater's constraint qualification holds, implying strong duality holds for P.

Some final massaging. Just as applying a unitary change of basis in our analysis of the Quantum Cook-Levin theorem helped with its analysis, here it turns out to be useful to also perform an appropriate change of variables. The final SDP we obtain is

<u>Primal problem (P)</u>

$$\begin{array}{rl} \text{maximize:} & \mathrm{Tr}(X) \\ \text{subject to:} & \Phi(X) \preceq I_C \otimes \sigma_{M_1} \\ & \mathrm{Tr}(\sigma) = 1 \\ & \sigma_{M_1} \succeq 0 \\ & X_{C,M_1,M_2} \succeq 0 \end{array}$$

<u>Dual problem (D)</u>

$$\begin{array}{rl} \text{infimum:} & \| \mathrm{Tr}_C(Y) \|_\infty \\ \text{subject to:} & \Phi^*(Y_{C,M_1}) \succeq I_{C,M_1,M_2} \\ & Y_{C,M_1} \succeq 0, \end{array}$$

where we define $\Phi : \mathcal{L}(C \otimes M_1 \otimes M_2) \mapsto \mathcal{L}(C \otimes M_1)$ as

$$\Phi(X) := \mathrm{Tr}_{M_2}(Q^{-1/2} X Q^{-1/2}), \tag{2}$$

with adjoint map $\Phi^* : \mathcal{L}(C \otimes M_1) \mapsto \mathcal{L}(C \otimes M_1 \otimes M_2)$ as $\Phi^*(Y) = Q^{-1/2}(Y \otimes I_{M_2})Q^{-1/2}$. Henceforth, P and D will always refer to this primal and dual problem, respectively.

3 QIP = PSPACE

We are now in a position to show that QIP = PSPACE. One direction of this equality is "trivial", in that QIP $\supseteq$ IP, for IP the classical analogue of QIP, which is known to equal PSPACE. Thus, the non-trivial direction is the containment QIP $\subseteq$ PSPACE.

A bit of interpretation and context. On the one hand, the statement QIP = PSPACE = IP is somewhat disappointing, in that as far as (single-prover) interactive proofs are concerned, quantum resources add no power. On the other hand, it may be interpreted as saying that interactive proofs are themselves so powerful that additional resources such as quantum computation add nothing new to the picture. It is worth noting, however, that remarkably, this state of affairs is only the case for *single-prover* interactive proofs. If we move to multiple prover interactive proofs (i.e. MIP, with multiple provers who may not communicate with each other once the protocol starts), it is known that classically MIP = NEXP, whereas quantumly MIP* $\supseteq$ NEEXP (you read that right; that's non-determenistic *doubly* exponential time)! Here, MIP* is MIP but where the provers are allowed to share entanglement before the protocol starts.

We begin by stating the algorithm which allows us to put QIP in PSPACE in Section 3.1. Correctness is shown in Section 3.2.

3.1 The algorithm

Before stating the algorithm, we sketch why it will imply containment of QIP in PSPACE.

Connection to PSPACE. In Section 2.3, we showed how to exactly capture the acceptance probability of a 3-message Quantum Arthur-Merlin game (and hence QIP) via an SDP P. In principle, one can then try to apply the Ellipsoid method to solve P, which would require time polynomial in the dimension of the matrices involved, such as Q. Unfortunately, Q has dimension *exponential* in the number of qubits, n, and so the best the Ellipsoid method could give us is containment in EXP.

Exercise. Why is the dimension of Q exponential in n?

We hence need an alternate approach for solving P. To begin, what we certainly *can* do in PSPACE is produce explicit descriptions of the matrix Q.

Exercise. Convince yourself that Q can be expressed exactly using a circuit of size exponential in n and depth polynomial in n. This class is called NC(poly), and formally requires the circuits be generated by a polynomial-space TM. It is known that NC(poly) = PSPACE.

Next, given a matrix M of dimension $d \times d$, it turns out one can also compute common matrix operations on M using circuits of size poly(d) and depth polylog(d); this includes, for example, matrix powers (M^k), matrix exponentials (e^M), and spectral decompositions. The corresponding complexity class is called NC, and formally requires all circuits to be generated by a log-space TM. (Intuitively, one can think of NC as the log-depth analogue of P.) This means that after computing the explicit matrix representation for Q in PSPACE above, we can then do things like take the spectral decomposition of Q in PSPACE as well. In other words, if we could just come up with an algorithm for solving SDP P which relies solely on the application of common matrix operations to Q, then we could put the entire thing into PSPACE. This is precisely what the matrix analogue of the multiplicative weights method allows us to do.

Statement of the algorithm. The idea for solving SDP P via the MW method is analogous to Section 1. Let us restate the SDP and its dual for convenience first.

Primal problem (P)

$$\begin{aligned} \text{maximize:} \quad & \mathrm{Tr}(X) \\ \text{subject to:} \quad & \Phi(X) \preceq I_C \otimes \sigma_{M_1} \\ & \mathrm{Tr}(\sigma) = 1 \\ & \sigma_{M_1} \succeq 0 \\ & X_{C,M_1,M_2} \succeq 0 \end{aligned}$$

Dual problem (D)

$$\begin{aligned} \text{infimum:} \quad & \|\,\mathrm{Tr}_C(Y)\,\|_\infty \\ \text{subject to:} \quad & \Phi^*(Y_{C,M_1}) \succeq I_{C,M_1,M_2} \\ & Y_{C,M_1} \succeq 0, \end{aligned}$$

for $\Phi(X) := \mathrm{Tr}_{M_2}(Q^{-1/2}XQ^{-1/2})$, $\Phi^*(Y) = Q^{-1/2}(Y \otimes I_{M_2})Q^{-1/2}$, and

$$Q := \frac{1}{2}\left(|0\rangle\langle 0|_C \otimes (P_0)_{M_1,M_2} + |1\rangle\langle 1|_C \otimes (P_1)_{M_1,M_2}\right).$$

There are two primal variables in play: X and σ. We have no idea what they should be set to, so just as in Section 1, we start with a random guess by setting both to the maximally mixed state. In each iteration, we check "how badly" the constraint $I_C \otimes \sigma_{M_1} - \mathrm{Tr}_{M_2}(X) \succeq 0$ is violated, and update our guesses for X and σ accordingly. For clarity, the actual implementation, given below, differs somewhat from this, but this is the basic spirit. The variables roughly ρ and ζ play the roles of X and σ, respectively (the actual choices of X and σ for P will be slight modifications of the final values of ρ and ζ).

Algorithm 2 Multiplicative Weights for QIP

```
1. For brevity, define $N = \dim(C \otimes M_1 \otimes M_2)$ and $M = \dim(M_1)$.
2. Set parameters $\gamma = 4/3$, $\epsilon = 1/64$, $\delta = \epsilon/(2\left\| Q^{-1} \right\|_\infty)$, $T = \lceil 4\log N/(\epsilon^3\delta)\rceil$.
3. Set initial states $\rho_0 = W_0/N$ for $W_0 = I_{C\otimes M_1\otimes M_2}$ and $\zeta_0 = Z_0/M$ for $Z_0 = I_C$.
4. For $t = 0,\dots,T$:
   (a) (Check constraint violation) Let $\Pi_t$ project onto the space spanned by the eigenvectors of $\Phi(\rho_t) - \gamma I_C \otimes \zeta_t$ with non-negative eigenvalues. Set $\beta_t = \mathrm{Tr}(\Phi(\rho_t)\Pi_t)$.
   (b) If $\beta_t \leq \epsilon$, accept.
   (c) (Update current solution) Set
       $\rho_{t+1} = W_{t+1}/\mathrm{Tr}(W_{t+1})$   for   $W_{t+1} = e^{-\epsilon\delta\sum_{j=0}^{t}\Phi^*\left(\frac{1}{\beta_j}\Pi_j\right)}$
       $\zeta_{t+1} = Z_{t+1}/\mathrm{Tr}(Z_{t+1})$   for   $Z_{t+1} = e^{\epsilon\delta\sum_{j=0}^{t}\mathrm{Tr}_C\left(\frac{1}{\beta_j}\Pi_j\right)}$.
5. Reject.
```

It is worth stressing that Algorithm 2 does *not* directly output a solution (X,σ) to SDP P. Instead, given its output (ρ,ζ), what we can do is the following: If Algorithm 2 accepts, then we can extract a "good" feasible solution (X,σ) for P from (ρ,ζ) which convinces us we are in a YES case. Conversely, if Algorithm 2 rejects, we can construct a "good" feasible *dual* solution Y for D which convinces us we are in a NO case (recall that any dual feasible solution upper bounds the value of the primal SDP from Section 2.2).

3.2 Correctness

We now show correctness of Algorithm 2. Let $\mathbb{A} = (A_{\text{yes}}, A_{\text{no}}, A_{\text{inv}})$ denote a QIP promise problem, which recall has a 3-message Quantum Arthur-Merlin game with verifier V. For any input $x \in \{0,1\}^*$, we may assume that if $x \in A_{\text{yes}}$, V accepts with certainty, and if $x \in A_{\text{no}}$, V accepts with probability at most $1/2+\epsilon$ for (say) $\epsilon = 1/64$. Thus, if $x \in A_{\text{yes}}$, the primal SDP P and (by strong duality) dual SDP D in Section 2.3 achieve value 1, and if $x \in A_{\text{no}}$, they achieve at most $1/2 + 1/64$.

Theorem 5. *If Algorithm 2 accepts, then P has optimal value strictly larger than $5/8$. If Algorithm 2 rejects, then P has optimal value strictly smaller than $7/8$.*

Exercise. Why does Theorem 5 show that Algorithm 2 correctly decides our QIP instance x?

We break the proof down into lemmas for the YES and NO cases.

Lemma 6. *(YES case) If Algorithm 2 accepts, then P has optimal value strictly larger than $5/8$.*

Proof. Let ρ, ζ, Π, and β denote the final values to $\rho_t, \zeta_t, \Pi_t\beta_t$ set by Algorithm 2. We claim that

$$X = \frac{1}{\gamma + 2\beta}\rho \qquad\qquad \sigma = \frac{1}{\gamma + 2\beta}\left[\gamma\zeta + 2\mathrm{Tr}_C(\Pi\Phi(\rho)\Pi)\right]$$

is a feasible solution to P with value strictly larger than $5/8$.

Exercise. Prove that the objective function value, $\mathrm{Tr}(X)$, is indeed strictly larger than $5/8$.

Exercise. Show that σ is a density operator.

It thus remains to show that the constraint $\Phi(X) \preceq I_C \otimes \sigma_{M_1}$ is satisfied. For this, rearrange

$$\Phi(X) - \gamma I_C \otimes \zeta \preceq \Pi(\Phi(X) - \gamma I_C \otimes \zeta)\Pi \preceq \Pi\Phi(X)\Pi \preceq 2I_C \otimes \mathrm{Tr}_C(\Pi\Phi(X)\Pi),$$

where the first and second inequalities hold by the definition of Π, and the third by the fact that $M_{AB} \preceq 2I_A \otimes \mathrm{Tr}_A(M)$ for any $M_{AB} \succeq 0$ and $A = \mathbb{C}^2$. □

Lemma 7. *(NO case) If Algorithm 2 rejects, then P has optimal value strictly smaller than $7/8$.*

Proof. We claim that a dual feasible solution to D is

$$Y = \frac{1+2\epsilon}{T}\sum_{t=0}^{T-1}\frac{1}{\beta_t}\Pi_t,$$

and that Y attains dual objective function value $\|\,\mathrm{Tr}_C(Y)\,\|_\infty < 7/8$; the lemma then holds by weak duality. As the proofs of both claims use a similar approach, we show only feasibility here.

Y **is dual feasible.** The proof approach is reminiscent of that of Theorem 1 (the classical MW algorithm); it turns out the correct "potential function" to use now, however, is $\mathrm{Tr}(W_T)$. We show upper and lower bounds on $\mathrm{Tr}(W_T)$ to establish that $\Phi^*(Y_{C,M_1}) \succeq I_{C,M_1,M_2}$. Equivalently, we show $\lambda_{\min}(\Phi^*(Y_{C,M_1})) \geq 1$.

Exercise. Show that $Y \succeq 0$.

Lower bound on $\mathrm{Tr}(W_T)$. We begin with the lower bound, which is simpler, and follows from the same principle as in the classical lower bound of Theorem 1 — the sum of non-negative numbers $\sum_i w_i$ is at least as large as any one number w_i.

Exercise. Prove that

$$\mathrm{Tr}(W_T) = \mathrm{Tr}\left(e^{-\epsilon\delta\sum_{j=0}^{T-1}\Phi^*\left(\frac{1}{\beta_j}\Pi_j\right)}\right) \geq e^{-\epsilon\delta\lambda_{\min}\left(\Phi^*\left(\sum_{j=0}^{T-1}\frac{1}{\beta_j}\Pi_j\right)\right)}. \tag{3}$$

Upper bound on $\mathrm{Tr}(W_T)$. We would like to upper bound

$$\mathrm{Tr}(W_T) = \mathrm{Tr}\left(e^{-\epsilon\delta\sum_{j=0}^{T-1}\Phi^*\left(\frac{1}{\beta_j}\Pi_j\right)}\right)$$

using an inductive approach similar to the upper bound in the proof of Theorem 1. The problem is that the exponents $\Phi^*(\Pi_j/\beta_j)$ do not pairwise commute; thus we cannot simply factor out the last term $\exp(-\epsilon\delta\Phi^*(\frac{1}{\beta_{T-1}}\Pi_{T-1}))$ and apply the induction hypothesis. Luckily, since we are interested in the *trace* of W_T, we are saved by the Golden-Thompson inequality, which states that

$$\mathrm{Tr}\left(e^{A+B}\right) \leq \mathrm{Tr}\left(e^A e^B\right) \text{ for Hermitian } A, B.$$

Exercise. For any $t \in 1, \ldots, T$, use the Golden-Thompson inequality to show that

$$\mathrm{Tr}(W_t) \leq \mathrm{Tr}(W_{t-1} e^{-\epsilon\delta\Phi^*(\Pi_{t-1}/\beta_{t-1})}). \tag{4}$$

Now that we've isolated the last term in the exponential, we wish to bring its argument $\delta\Phi^*(\Pi_{t-1}/\beta_{t-1})$ down out of the exponent. For this, we use the fact that for any Hermitian M satisfying $0 \preceq M \preceq I$, and every real $r > 0$,

$$e^{rM} \preceq I + re^r M \quad \text{and} \quad e^{-rM} \preceq I - re^{-r} M.$$

From this, we immediately have that

$$e^{-\epsilon[\delta\Phi^*(\Pi_{t-1}/\beta_{t-1})]} \preceq I - \epsilon\delta e^{-\epsilon}\Phi^*(\Pi_{t-1}/\beta_{t-1}), \tag{5}$$

assuming the result of the following exercise.

Exercise. Prove that $\| \delta\Phi^*(\Pi_{t-1}/\beta_{t-1}) \|_\infty \leq 1$. (Hint: Use submultiplicativity of the spectral norm to show that $\| \Phi^*(\Pi_{t-1}) \|_\infty \leq \left\| Q^{-1} \right\|_\infty$.)

Exercise. Use Equation (5) and the fact that $W_{t-1} \succeq 0$ to conclude that

$$\mathrm{Tr}(W_t) \leq \mathrm{Tr}(W_{t-1})(1 - \epsilon\delta e^{-\epsilon}\mathrm{Tr}\left[\rho_{t-1}\Phi^*(\Pi_{t-1}/\beta_{t-1})\right]) = \mathrm{Tr}(W_{t-1})\left(1 - \epsilon\delta e^{-\epsilon}\frac{\mathrm{Tr}\left[\Phi(\rho_{t-1})(\Pi_{t-1})\right]}{\beta_{t-1}}\right).$$

Combining this with the fact that by definition, $\beta_{t-1} = \mathrm{Tr}\left[\Phi(\rho_{t-1})(\Pi_{t-1})\right]$, and that for all real r, $1+r \leq e^r$, we conclude

$$\mathrm{Tr}(W_t) \leq \mathrm{Tr}(W_{t-1})e^{-\epsilon\delta e^{-\epsilon}}.$$

Exercise. Use the fact that $\mathrm{Tr}(W_0) = N$ to obtain our final upper bound,

$$\mathrm{Tr}(W_T) \leq e^{-T\epsilon\delta e^{-\epsilon} + \log N}. \tag{6}$$

Combining upper and lower bounds. Combining Equations (3) and (6), we have that

$$\lambda_{\min}\left(\Phi^*\left(\sum_{j=0}^{T-1}\frac{1}{\beta_j}\Pi_j\right)\right) \geq Te^{-\epsilon} - \frac{\log N}{\epsilon\delta}.$$

Exercise. Recalling that $Y = \frac{1+2\epsilon}{T}\sum_{t=0}^{T-1}\frac{1}{\beta_t}\Pi_t$, use the fact that $e^{-\epsilon} - \epsilon^2/4 > 1 - \epsilon$ to complete the proof of dual feasibility by concluding that $\lambda_{\min}(\Phi^*(Y)) > 1$. (Note the argument to Φ^* is now Y.) □

In conclusion, we have covered a rather long journey for this set of lectures notes, spanning the MW method, SDPs, interactive proofs, and finally the proof that QIP = PSPACE. Each of these is a fundamental tool or result in its own right; that they fit together to tell an elegant story is nothing short of remarkable.