A search for quantum coin-flipping protocols using optimization techniques

Ashwin Nayak[*]   Jamie Sikora[†]   Levent Tunçel[‡]

February 28, 2014

Abstract

Coin-flipping is a cryptographic task in which two physically separated, mistrustful parties wish to generate a fair coin-flip by communicating with each other. Chailloux and Kerenidis (2009) designed quantum protocols that guarantee coin-flips with near optimal bias away from uniform, even when one party deviates arbitrarily from the protocol. The probability of any outcome in these protocols is provably at most $\frac{1}{\sqrt{2}} + \delta$ for any given $\delta > 0$. However, no explicit description of these protocols is known, and the number of rounds in the protocols tends to infinity as δ goes to 0. In fact, the smallest bias achieved by known explicit protocols is $1/4$ (Ambainis, 2001).

We take a *computational optimization* approach, based mostly on convex optimization, to the search for simple and explicit quantum strong coin-flipping protocols. We present a search algorithm to identify protocols with low bias within a natural class, protocols based on *bit-commitment* (Nayak and Shor, 2003). To make this search computationally feasible, we further restrict to commitment states *à la* Mochon (2005). An analysis of the resulting protocols via semidefinite programs (SDPs) unveils a simple structure. For example, we show that the SDPs reduce to second-order cone programs. We devise novel cheating strategies in the protocol by restricting the semidefinite programs and use the strategies to prune the search.

The techniques we develop enable a computational search for protocols given by a mesh over the corresponding parameter space. The protocols have up to six rounds of communication, with messages of varying dimension and include the best known explicit protocol (with bias $1/4$). We conduct two kinds of search: one for protocols

---
[*]Department of Combinatorics and Optimization, and Institute for Quantum Computing, University of Waterloo. Address: 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada.
Email: `ashwin.nayak@uwaterloo.ca`.

[†]Laboratoire d'Informatique Algorithmique: Fondements et Applications, Université Paris Diderot. Address: 5 rue Thomas-Mann 75205 Paris cedex 13, France.
Email: `jamie.sikora@liafa.univ-paris-diderot.fr`.

[‡]Department of Combinatorics and Optimization, University of Waterloo. Address: 200 University Ave. W., Waterloo, ON, N2L 3G1, Canada.
Email: `ltuncel@uwaterloo.ca`.

with bias below 0.2499, and one for protocols in the neighbourhood of protocols with bias $1/4$. Neither of these searches yields better bias. Based on the mathematical ideas behind the search algorithm, we prove a lower bound of 0.2487 on the bias of a class of four-round protocols.

Contents

1 Introduction

Some fundamental problems in the area of Quantum Cryptography allow formulations in the language of convex optimization in the space of hermitian matrices over the complex numbers, in particular, in the language of semidefinite optimization. These formulations enable us to take a computational optimization approach towards solutions of some of these problems. In the rest of this section, we describe *quantum coin-flipping* and introduce our approach.

1.1 Quantum coin-flipping

Coin-flipping is a classic cryptographic task introduced by Blum [Blu81]. In this task, two remotely situated parties, Alice and Bob, would like to agree on a uniformly random bit by communicating with each other. The complication is that neither party trusts the other. If Alice were to toss a coin and send the outcome to Bob, Bob would have no means to verify whether this was a uniformly random outcome. In particular, if Alice wishes to cheat, she could send the outcome of her choice without any possibility of being caught cheating. We are interested in a communication protocol that is *designed to protect* an honest party from being cheated.

More precisely, a "strong coin-flipping protocol" with bias ϵ is a two-party communication protocol in the style of Yao [Yao79, Yao93]. In the protocol, the two players, Alice and Bob, start with no inputs and compute a value $c_A, c_B \in \{0, 1\}$, respectively, or declare that the other player is cheating. If both players are honest, i.e., follow the protocol, then they agree on the outcome of the protocol ($c_A = c_B$), and the coin toss is fair ($\Pr(c_A = c_B = b) = 1/2$, for any $b \in \{0, 1\}$). Moreover, if one of the players deviates arbitrarily from the protocol in his or her local computation, i.e., is "dishonest" (and the other party is honest), then the probability of either outcome 0 or 1 is at most $1/2 + \epsilon$. Other variants of coin-flipping have also been studied in the literature. However, in the rest of the article, by "coin-flipping" (without any modifiers) we mean *strong* coin flipping.

3

A straightforward game-theoretic argument proves that if the two parties in a coin-flipping protocol communicate classically and are computationally unbounded, at least one party can cheat perfectly (with bias $1/2$). In other words, there is at least one party, say Bob, and at least one outcome $b \in \{0, 1\}$ such that Bob can ensure outcome b with probability 1 by choosing his messages in the protocol appropriately. Consequently, classical coin-flipping protocols with bias $\epsilon < 1/2$ are only possible under complexity-theoretic assumptions, and when Alice and Bob have limited computational resources.

Quantum communication offers the possibility of "unconditionally secure" cryptography, wherein the security of a protocol rests solely on the validity of quantum mechanics as a faithful description of nature. The first few proposals for quantum information processing, namely the Wiesner quantum money scheme [Wie83] and the Bennett-Brassard quantum key expansion protocol [BB84] were motivated by precisely this idea. These schemes were indeed eventually shown to be unconditionally secure in principle [May01, LC99, PS00, MVW12]. In light of these results, several researchers have studied the possibility of *quantum* coin-flipping protocols, as a step towards studying more general secure multi-party computations.

Lo and Chau [LC97] and Mayers [May97] were the first to consider quantum protocols for coin-flipping without any computational assumptions. They proved that no protocol with a finite number of rounds could achieve 0 bias. Nonetheless, Aharonov, Ta-Shma, Vazirani, and Yao [ATVY00] designed a simple, three-round quantum protocol that achieved bias $\approx 0.4143 < 1/2$. This is impossible classically, even with an unbounded number of rounds. Ambainis [Amb01] designed a protocol with bias $1/4$ *à la* Aharonov *et al.*, and proved that it is optimal within a class (see also Refs. [SR01, KN04] for a simpler version of the protocol and a complete proof of security). Shortly thereafter, Kitaev [Kit02] proved that any strong coin-flipping protocol with a finite number of rounds of communication has bias at least $(\sqrt{2} - 1)/2 \approx 0.207$ (see Ref. [GW07] for an alternative proof). Kitaev's seminal work uses semidefinite optimization in a central way. This argument extends to protocols with an unbounded number of rounds. This remained the state of the art for several years, with inconclusive evidence in either direction as to whether $1/4 = 0.25$ or $(\sqrt{2} - 1)/2$ is optimal. In 2009, Chailloux and Kerenidis [CK09] settled this question through an elegant protocol scheme that has bias at most $(\sqrt{2} - 1)/2 + \delta$ for any $\delta > 0$ of our choice (building on [Moc07], see below). We refer to this as the CK protocol.

The CK protocol uses breakthrough work by Mochon [Moc07], which itself builds upon the "point game" framework proposed by Kitaev. Mochon shows there are *weak* coin-flipping protocols with arbitrarily small bias. (This work has appeared only in the form of an unpublished manuscript, but has been verified by experts on the topic; see e.g. [ACG$^+$13].) A weak coin-flipping protocol is a variant of coin-flipping in which each party favours a distinct outcome, say Alice favours 0 and Bob favours 1. The requirement when they are honest is the same as before. We say it has bias ϵ if the following condition holds. When Alice is dishonest and Bob honest, we only require that Bob's outcome

is 0 (Alice's favoured outcome) with probability at most $1/2 + \epsilon$. A similar condition to protect Alice holds, when she is honest and Bob is dishonest. The weaker requirement of security against a dishonest player allows us to circumvent the Kitaev lower bound. While Mochon's work pins down the optimal bias for weak coin-flipping, it does this in a non-constructive fashion: we only know of the *existence* of protocols with arbitrarily small bias, not of its *explicit description*. Moreover, the number of rounds tends to infinity as the bias decreases to 0. As a consequence, the CK protocol for strong coin-flipping is also existential, and the number of rounds tends to infinity as the bias decreases to $(\sqrt{2} - 1)/2$. It is perhaps very surprising that no progress on finding better explicit protocols has been made in over a decade.

1.2 Search for explicit protocols

This work is driven by the quest to find *explicit* and *simple* strong coin-flipping protocols with bias smaller than $1/4$. There are two main challenges in this quest. First, there seems to be little insight into the structure (if any) that protocols with small bias have; knowledge of such structure might help narrow our search for an optimal protocol. Second, the analysis of protocols, even those of a restricted form, with more than three rounds of communication is technically quite difficult. As the first step in deriving the $(\sqrt{2} - 1)/2$ lower bound, Kitaev [Kit02] proved that the optimal cheating probability of any dishonest party in a protocol with an explicit description is characterized by a semidefinite program (SDP). While this does not entirely address the second challenge, it reduces the analysis of a protocol to that of a well-studied optimization problem. In fact this formulation as an SDP enabled Mochon to analyze an important class of weak coin-flipping protocols [Moc05], and later discover the optimal weak coin flipping protocol [Moc07]. SDPs resulting from strong coin-flipping protocols, however, do not appear to be amenable to similar analysis.

We take a *computational optimization* approach to the search for explicit strong coin-flipping protocols. We focus on a class of protocols studied by Nayak and Shor [NS03] that are based on "bit commitment". This is a natural class of protocols that generalizes those due to Aharonov *et al.* and Ambainis, and provides a rich test bed for our search. (See Section 3.3 for a description of such protocols.) Early proposals of multi-round protocols in this class were all shown to have bias at least $1/4$, without eliminating the possibility of smaller bias (see, e.g., Ref. [NS03]). A characterization of the smallest bias achievable in this class would be significant progress on the problem: it would either lead to simple, explicit protocols with bias smaller than $1/4$, or we would learn that protocols with smaller bias take some other, yet to be discovered form.

Chailloux and Kerenidis [CK11] have studied a version of quantum bit-commitment that may have implications for coin-flipping. They proved that in any quantum bit-commitment protocol with computationally unbounded players, at least one party can cheat with bias at least ≈ 0.239. Since the protocols we study involve two interleaved

5

commitments to independently chosen bits, this lower bound does not apply to the class. Chailloux and Kerenidis also give a protocol scheme for bit-commitment that guarantees bias arbitrarily close to 0.239. The protocol scheme is non-constructive as it uses the Mochon weak coin-flipping protocol. It is possible that any explicit protocols we discover for coin-flipping could also lead to explicit bit-commitment with bias smaller than $1/4$.

We present an algorithm for finding protocols with low bias. Each bit-commitment based coin-flipping protocol is specified by a 4-tuple of quantum states. At a high level, the algorithm iterates through a suitably fine mesh of such 4-tuples, and computes the bias of the resulting protocols. The size of the mesh scales faster than $1/\nu^{\kappa D}$, where ν is a precision parameter, κ is a universal constant, and D is the dimension of the states. The dimension itself scales as 2^n, where n is the number of quantum bits involved. In order to minimize the doubly exponential size of the set of 4-tuples we examine, we further restrict our attention to states of the form introduced by Mochon for weak coin-flipping [Moc05]. The additional advantage of this kind of state is that the SDPs in the analysis of the protocols simplify drastically. In fact, all but a few constraints reduce to linear equalities so that the SDPs may be solved more efficiently.

Next, we employ two techniques to prune the search space of 4-tuples. First, we use a sequence of strategies for dishonest players whose bias is given by a closed form expression determined by the four states. The idea is that if the bias for any of these strategies is higher than $1/4$ for any 4-tuple of states, we may safely rule it out as a candidate optimal protocol. This also has the advantage of avoiding a call to the SDP solver, the computationally most intensive step in the search algorithm. The second technique is to invoke symmetries in the search space as well as in the problem to identify protocols with the same bias. The idea here is to compute the bias for as few members of an equivalence class of protocols as possible.

These techniques enable a computational search for protocols with up to six rounds of communication, with messages of varying dimension. The Ambainis protocol with bias $1/4$ has three rounds, and it is entirely possible that a strong coin-flipping protocol with a small number of rounds be optimal. Thus, the search non-trivially extends our understanding of this cryptographic primitive. We elaborate on this next.

1.3 The results

We performed two types of search. The first was an optimistic search that sought protocols within the mesh with bias at most $1/4$ minus a small constant. We chose the constant to be 0.001. The rationale here was that if the mesh contains protocols with bias close to the lower bound of ≈ 0.207, we would find protocols that have bias closer to 0.25 (but smaller than it) relatively quickly. We searched for four-round protocols in which each message is of dimension ranging from 2 to 9, each with varying fineness for the mesh. We found that our heuristics, i.e., the filtering by fixed cheating strategies, performed so well that they eliminated every protocol: all of the protocols given by the mesh were found to have bias

larger than 0.2499 without the need to solve any SDP. Inspired by the search algorithm, we give an analytical proof that four-round qubit protocols have bias at least 0.2487.

The initial search for four-round protocols helped us fine-tune the filter by a careful selection of the order in which the cheating strategies were tried. The idea was to eliminate most protocols with the least amount of computation. This made it feasible for us to search for protocols in finer meshes, with messages of higher dimension, and with a larger number of rounds. In particular, we were able to check six-round protocols with messages of dimension 2 and 3. Our heuristics again performed very well, eliminating almost every protocol before any SDP needed to be solved. Even during this search, not a single protocol with bias less than 0.2499 was found. We also performed a search over meshes shifted by a randomly chosen parameter. This was to avoid potential anomalies caused by any special properties of the mesh we used. No protocols with bias less than 0.2499 were found in this search either.

The second kind of search focused on protocols with bias close to 0.25. We first identified protocols in the mesh with the least bias. Not surprisingly, these protocols all had computationally verified bias $1/4$. We zoned in on the neighbourhood of these protocols. The idea here was to see if there are perturbations to the 4-tuple that lead to a decrease in bias. This search revealed 2 different equivalence classes of protocols for the four-round version and 6 for the six-round version. Four of these eight protocols are equivalent to optimal three-round protocols (within this class). However, the four remaining six-round protocols bear no resemblance to any known protocol with bias $1/4$. A search in the neighbourhoods of all these protocols revealed no protocols with bias less than $1/4$ (details in Section 8).

It may not immediately be evident that the above searches involved a computational examination of extremely large sets of protocols and that the techniques described above in Section 1.2, were crucial in enabling this search. The symmetry arguments pruned the searches drastically, and in some cases only 1 in every $1,000,000$ protocols needed to be checked. In most cases, the cheating strategies (developed in Section 5) filtered out the rest of the protocols entirely. To give an example of the efficiency of our search, we were able to check 2.74×10^{16} protocols in a matter of days. Without the symmetry arguments and the use of cheating strategies as a filter, this same search would have taken well over 69 million years, even using the very simplified forms of the SDPs. Further refinement of these ideas may make a more thorough search of protocols with four or more rounds feasible.

The search algorithm, if implemented with exact feasibility guarantees, has the potential to give us *computer aided proofs* that certain classes of protocols in the family *do not achieve optimal bias*. Suppose we use a mesh such that given any 4-tuple S of states, there is a 4-tuple S' in the mesh such that the pairwise fidelity between corresponding distributions is at least $1 - \delta$. Further suppose the numerical approximation to the bias for S'

has additive error τ due to the filter or SDP solver, and finite precision arithmetic[1]. If the algorithm reports that there are no tuples in the mesh with bias at most ϵ^*, then it holds that there are no 4-tuples, even outside the mesh, with bias at most $\epsilon^* - \sqrt{8\delta} - \tau$. The fineness of the mesh we are able to support currently is not sufficient for such proofs. A refinement of the search algorithm along the lines described above, however, would yield lower bounds for new classes of bit-commitment based protocols.

Finally, based on our computational findings, we make the following conjecture:

Conjecture 1.1 *Any strong coin-flipping protocol based on bit-commitment as defined formally in Section 3.3 has bias at least* $1/4$.

This conjecture, if true, would imply that we need to investigate new kinds of protocols to find ones with bias less than $1/4$. Regardless of the truth of the above conjecture, we hope that the new techniques developed for analyzing protocols via modern optimization methods and for simplifying semidefinite optimization problems with special structure will be helpful in future work in the areas of quantum computing and semidefinite programming.

Organization of the paper. We begin with an introduction to the ideas contained in this paper in Section 2. Section 2.3 introduces quantum computing background and Section 2.4 introduces semidefinite programming and related optimization classes. Section 3.1 defines strong coin-flipping protocols and the measure of their security (namely, their bias). We define the notion of protocols based on bit-commitment in Section 3.3. We model optimal cheating strategies for such protocols using semidefinite programming in Section 4. Sections 5 and 6 exploit the structure of the semidefinite programs in order to design the search algorithm presented in Section 7. We conclude with computational results in Section 8 and some final remarks in Section 9.

The background material on quantum computation and optimization is aimed at making this work accessible to researchers in both communities. Readers conversant with either topic need only skim the corresponding sections to familiarize themselves with the notation used. Proofs of most results are deferred to the appendix.

2 Background and notation

In this section, we establish the notation and the necessary background for this paper.

---

[1]Note that in our experiments feasibility is guaranteed only up to a tolerance, so as a result we do not have an independently verifiable upper bound on the additive error in terms of the objective value. Indeed, efficiently obtaining an exact feasible solution to SDPs, in general, is still an open problem at the time of this writing.

2.1 Linear algebra

For a finite set A, we denote by $\mathbb{R}^A$, $\mathbb{R}^A_+$, Prob^A, and $\mathbb{C}^A$ the set of real vectors, nonnegative real vectors, probability vectors, and complex vectors, respectively, each indexed by A. We use $\mathbb{R}^n$, $\mathbb{R}^n_+$, Prob^n, and $\mathbb{C}^n$ for the special case when $A = \{1, \ldots, n\}$. For $x \in A$, the vectors e_x denote the standard basis vectors of $\mathbb{R}^A$. The vector $e_A \in \mathbb{R}^A$ denotes the all 1 vector $\sum_{x \in A} e_x$.

We denote by $\mathbb{S}^A$ and $\mathbb{S}^A_+$ the set of Hermitian matrices and positive semidefinite matrices, respectively, each over the reals with columns and rows indexed by A.

It is convenient to define $\sqrt{x}$ to be the element-wise square root of a nonnegative vector x. The element-wise square root of a probability vector yields a unit vector (in the Euclidean norm). This operation maps a probability vector to a quantum state, see Subsection 2.3.

For vectors x and y, the notation $x \geq y$ denotes that $x - y$ has nonnegative entries, $x > y$ denotes that $x - y$ has positive entries, and for matrices X and Y, the notation $X \succeq Y$ denotes that $X - Y$ is positive semidefinite, and $X \succ Y$ denotes $X - Y$ is positive definite when the underlying spaces are clear from context. When we say that a matrix is positive semidefinite or positive definite, it is assumed to be Hermitian which implies that $\mathbb{S}^A_+ \subset \mathbb{S}^A$.

The Kronecker product of an $n \times n$ matrix X and another matrix Y, denoted $X \otimes Y$, is defined as

$$
X \otimes Y := \begin{bmatrix}
X_{1,1}\,Y & X_{1,2}\,Y & \cdots & X_{1,n}\,Y \\
X_{2,1}\,Y & X_{2,2}\,Y & \cdots & X_{2,n}\,Y \\
\vdots & \vdots & \ddots & \vdots \\
X_{n,1}\,Y & X_{n,2}\,Y & \cdots & X_{n,n}\,Y
\end{bmatrix}.
$$

Note that $X \otimes Y \in \mathbb{S}^{A \times B}_+$ when $X \in \mathbb{S}^A_+$ and $Y \in \mathbb{S}^B_+$ and $\mathrm{Tr}(X \otimes Y) = \mathrm{Tr}(X) \cdot \mathrm{Tr}(Y)$ when X and Y are square.

The *Schatten 1-norm*, or *nuclear norm*, of a matrix X is defined as

$$
\|X\|_* := \mathrm{Tr}(\sqrt{X^*X}),
$$

where X^* is the adjoint of X and $\sqrt{X}$ denotes the square root of a positive semidefinite matrix X, i.e., the positive semidefinite matrix Y such that $Y^2 = X$. Note that the 1-norm of a matrix is the sum of its singular values. The 1-norm of a vector $p \in \mathbb{C}^A$ is denoted as

$$
\|x\|_1 := \sum_{x \in A} |p_x|.
$$

We use the notation $\bar{a}$ to denote the complement of a bit a with respect to 0 and 1 and $a \oplus b$ to denote the XOR of the bits a and b. We use $\mathbb{Z}^n_2$ to denote the set of n-bit binary strings.

For a vector $p \in \mathbb{R}^A$, we denote by $\mathrm{Diag}(p) \in \mathbb{S}^A$ the diagonal matrix with p on the diagonal. For a matrix $X \in \mathbb{S}^A$, we denote by $\mathrm{diag}(X) \in \mathbb{R}^A$ the vector on the diagonal of X.

For a vector $x \in \mathbb{C}^A$, we denote by $\mathrm{supp}(x)$ the set of indices of A where x is nonzero. We denote by x^{-1} the element-wise inverse of x (mapping the 0 entries to 0).

For a matrix X, we denote by $\mathrm{Null}(X)$ the nullspace of X, by $\det(X)$ the determinant of X, and by $\lambda_{\max}(X)$ the largest eigenvalue of X. We denote by $\langle X, Y \rangle$ the standard inner product $\mathrm{Tr}(X^*Y)$ of matrices X, Y of the same dimension.

2.2 Convex analysis

A *convex combination* of finitely many vectors $x_1, \ldots, x_n$ is any vector of the form $\sum_{i=1}^n \lambda_i x_i$, when $\lambda_1, \ldots, \lambda_n \in [0, 1]$ satisfy $\sum_{i=1}^n \lambda_i = 1$. The *convex hull* of a set C is the set of convex combinations of elements of C, denoted $\mathrm{conv}(C)$. A set C is *convex* if $C = \mathrm{conv}(C)$.

A *convex function* $f : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is one that satisfies

$$f(\lambda x + (1 - \lambda)y) \le \lambda f(x) + (1 - \lambda)f(y), \text{ for all } x, y \in \mathbb{R}^n, \lambda \in [0, 1].$$

A convex function is *strictly convex* if

$$f(\lambda x + (1 - \lambda)y) < \lambda f(x) + (1 - \lambda)f(y), \text{ for all } x \ne y, \ x, y \in \mathbb{R}^n, \lambda \in (0, 1).$$

We say that a convex function is proper if $f(x) < +\infty$ for some $x \in \mathbb{R}^n$. The *epigraph* of a function f is the set

$$\mathrm{epi}(f) := \{(x, t) : f(x) \le t\}$$

which are the points above the graph of the function f. A function is convex if and only if its epigraph is a convex set.

A function $f : \mathbb{R}^n \to \mathbb{R} \cup \{-\infty\}$ is *(strictly) concave* if $-f$ is (strictly) convex, and proper when $f(x) > -\infty$ for some $x \in \mathbb{R}^n$. The *hypograph* of a function f is the set

$$\mathrm{hypo}(f) := \{(x, t) : f(x) \ge t\}$$

which are the points below the graph of the function f. A function is concave if and only if its hypograph is a convex set.

Let $f_1, \ldots, f_n : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ be proper, convex functions. We denote the *convex hull* of the functions $\{f_1, \ldots, f_n\}$ by $\mathrm{conv}\{f_1, \ldots, f_n\}$ which is the greatest convex function f such that $f(x) \le f_1(x), \ldots, f_n(x)$ for every $x \in \mathbb{R}^m$. The convex hull can be written in terms of the epigraphs

$$\mathrm{conv}\{f_1, \ldots, f_n\}(x) := \inf \{t : (x, t) \in \mathrm{conv}(\cup_{i=1}^n \mathrm{epi}(f_i))\}.$$

We denote the *concave hull* of $\{f_1, \ldots, f_n\}$ by $\mathrm{conc}\{f_1, \ldots, f_n\}$ which can be written as

$$\mathrm{conc}\{f_1, \ldots, f_n\} := -\mathrm{conv}\{-f_1, \ldots, -f_n\}$$

when $f_1, \ldots, f_n : \mathbb{R}^m \to \mathbb{R} \cup \{-\infty\}$ are proper, concave functions. The concave hull is the least concave function f such that $f(x) \geq f_1(x), \ldots, f_n(x)$ for every $x \in \mathbb{R}^m$ and can be written as

$$\text{conc}\{f_1, \ldots, f_n\}(x) := \sup \{t : (x,t) \in \text{conv}(\cup_{i=1}^n \text{hypo}(f_i))\}.$$

A *convex optimization problem* or *convex program* is one of the form

$$\inf_{x \in C} f(x),$$

where f is a convex function and C is a convex set. Alternatively, one could maximize a concave function over a convex set.

2.3 Quantum information

In this subsection, we give a brief introduction to quantum information. For a more thorough treatment of the subject, we refer the reader to [NC00].

Quantum states

Quantum states are a description of the state of a physical system, such as the spin of an electron. In the simplest case, such a state is a *unit vector* in a finite-dimensional Hilbert space (which is a complex Euclidean space). For example, the following vectors are quantum states in $\mathbb{C}^2$

$$e_0 := \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \ e_1 := \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \ e_+ := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \ e_- := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ -1 \end{bmatrix}.$$

The first two are standard basis vectors and can be thought of as the logical states of a standard computer. In general, a qubit can be written as

$$\psi := \alpha_0 \, e_0 + \alpha_1 \, e_1,$$

where $\alpha_0, \alpha_1 \in \mathbb{C}$ satisfy $|\alpha_0|^2 + |\alpha_1|^2 = 1$. This condition ensures that ψ has norm equal to 1. Up to factor of modulus 1, the set of pairs (α_0, α_1) defining a two-dimensional quantum state is in one-to-one correspondence with the unit sphere in $\mathbb{R}^3$.

Systems with a two dimensional state space are called *quantum bits* or *qubits*. The state space of a sequence of n qubits is given by the n-fold tensor product $(\mathbb{C}^2)^{\otimes n} \cong \mathbb{C}^{2^n}$. Higher dimensional systems, say, of dimension $d \leq 2^n$, may be viewed as being composed of a sequence of n qubits via a canonical isometry $\mathbb{C}^d \to \mathbb{C}^{2^n}$.

Notice that $e_+ = \frac{1}{\sqrt{2}} e_0 + \frac{1}{\sqrt{2}} e_1$ and $e_- = \frac{1}{\sqrt{2}} e_0 - \frac{1}{\sqrt{2}} e_1$. These states are said to be in a *superposition* of the states e_0 and e_1 and exhibit properties of being in both states at the same time. This is in part what gives quantum computers the power to efficiently tackle hard problems such as factoring [Sho97].

In general, a system may be in a random superposition according to some probability distribution. Suppose a quantum system is in such a state drawn from the ensemble of states $(\psi_0, \psi_1, \ldots, \psi_n)$ with probabilities $(p_0, p_1, \ldots, p_n)$, respectively. This quantum state may be described more succinctly as a *density matrix*, defined as

$$\sum_{i=0}^{n} p_i \, \psi_i \psi_i^*.$$

Notice that this matrix is positive semidefinite and has unit trace. Moreover, any positive semidefinite matrix with unit trace can be written in the above form using its spectral decomposition.

Two different probability distributions over superpositions may have the same density matrix. For example, density matrices do not record "phase information", i.e., the density matrix of state ψ is the same as that of $-\psi$. However, two ensembles with the same density matrix behave identically under all allowed physical operations. Therefore, there is no loss in working with density matrices, and we identify an ensemble with its density matrix.

A quantum superposition given by the vector ψ corresponds to the rank 1 density matrix $\psi\psi^*$ and we call it a *pure state*. States with a density matrix of rank 2 or more are said to be *mixed*.

Quantum operations

The most basic quantum operation is specified by a unitary transformation. Suppose U is a unitary operator acting on $\mathbb{C}^A$ and $\psi \in \mathbb{C}^A$ is a quantum state. If we apply U to ψ then the resulting quantum state is $U\psi \in \mathbb{C}^A$. Note this is a well-defined quantum state since unitary operators preserve Euclidean norm.

Suppose we are given a state drawn from the ensemble $(\psi_0, \psi_1, \ldots, \psi_n)$ with probabilities $(p_0, p_1, \ldots, p_n)$. Then if we apply a unitary matrix U to the state, the resulting state is given by the ensemble $(U\psi_0, U\psi_1, \ldots, U\psi_n)$ with the same probabilities. The new density matrix is thus

$$\sum_{i=0}^{n} p_i \, U\psi_i\psi_i^*U^* = U \left(\sum_{i=0}^{n} p_i \, \psi_i\psi_i^* \right) U^*,$$

where U^* is the adjoint of U. Thus, if we apply the unitary U to a state (with density matrix) ρ, then the resulting quantum state is $U\rho U^*$. Note that this matrix is still positive semidefinite with unit trace.

We assume that parties capable of quantum information processing have access to qubits initialized to a fixed quantum state, say e_0, can apply arbitrary unitary operations, and can physically transport ("send") qubits without disturbing their state. We use the phrase "prepare a quantum state $\psi \in \mathbb{C}^A$" to mean that we start with sufficiently many

qubits (say n such that $\mathbb{C}^A \subseteq \mathbb{C}^{2^n}$) in state $e_0^{\otimes n}$ and apply any unitary transformation that maps $e_0^{\otimes n}$ to ψ.

Quantum measurement

Measurement is a means of extracting classical information from a quantum state. A *quantum measurement* on space $\mathbb{C}^A$ is a sequence of positive semidefinite operators $(\Pi_1, \ldots, \Pi_n)$, with $\Pi_i \in \mathbb{S}_+^A$ for each $i \in \{1, \ldots, n\}$, satisfying $\sum_{i=1}^n \Pi_i = \mathrm{I}$. This sequence of operators is also called a *positive operator valued measure* or a POVM in the literature. If we have some qubits in state ρ and we apply the measurement $(\Pi_1, \ldots, \Pi_n)$ (or "observe the qubits according to the measurement"), we obtain *outcome "i"* with probability $\langle \Pi_i, \rho \rangle$, and the state of the qubits becomes $\Pi_i \rho \Pi_i / \langle \Pi_i, \rho \rangle$. The definitions of density matrices and measurements establish $(\langle \Pi_i, \rho \rangle)$ as a well-defined probability distribution over the indices. The alteration of state resulting from a measurement is referred to as a *collapse*. Due to this restricted kind of access, in general only a limited amount of classical information may be extracted from a given quantum state.

For example, if we apply the measurement $\{\Pi_0 := e_0 e_0^*, \Pi_1 := e_1 e_1^*\}$ to the state $e_+ e_+^*$, we obtain the outcomes:

$$\left\{ \begin{array}{llll} \text{``0''} & \text{with probability} & \langle \Pi_0, e_+ e_+^* \rangle = 1/2, \\ \text{``1''} & \text{with probability} & \langle \Pi_1, e_+ e_+^* \rangle = 1/2. \end{array} \right.$$

Multiple quantum systems

For convenience, we refer to a quantum system with state space $\mathbb{C}^A$ by the index set A. Suppose we have two quantum systems A_1, A_2 that are independently in pure states $\psi_1 \in \mathbb{C}^{A_1}$ and $\psi_2 \in \mathbb{C}^{A_2}$. Their combined state is $\psi_1 \otimes \psi_2 \in \mathbb{C}^{A_1} \otimes \mathbb{C}^{A_2} \cong \mathbb{C}^{A_1 \times A_2}$ where $\otimes$ denotes the Kronecker (or tensor) product. Note that the Kronecker product has the property that $\|x \otimes y\|_2 = \|x\|_2 \|y\|_2$ so unit norm is preserved. It is not always possible to decompose a vector in $\mathbb{C}^{A_1} \otimes \mathbb{C}^{A_2}$ as a Kronecker product of vectors in $\mathbb{C}^{A_1}$ and $\mathbb{C}^{A_2}$; a state with this property is said to be *entangled*. For example, the state $\Phi^+ = [1/\sqrt{2}, 0, 0, 1/\sqrt{2}]^\mathrm{T}$ is entangled; it cannot be expressed as $\psi_1 \otimes \psi_2$ for any choice of $\psi_1, \psi_2 \in \mathbb{C}^2$.

These concepts extend to mixed states as well. If two disjoint quantum systems are independently in states $\rho_1 \in \mathbb{S}_+^{A_1}$ and $\rho_2 \in \mathbb{S}_+^{A_2}$, then the joint state of the combined system is the density matrix $\rho_1 \otimes \rho_2 \in \mathbb{S}_+^{A_1 \times A_2}$. We make use of the properties that Kronecker products preserve positive semidefiniteness and that $\mathrm{Tr}(A \otimes B) = \mathrm{Tr}(A)\,\mathrm{Tr}(B)$. It is not always possible to write a density matrix $\rho \in \mathbb{S}_+^{A_1 \times A_2}$ as $\rho_1 \otimes \rho_2$ where $\rho_1 \in \mathbb{S}_+^{A_1}$ and $\rho_2 \in \mathbb{S}_+^{A_2}$, or more generally, as a convex combination of such Kronecker products. In the latter case, the state is said to be *entangled*, and otherwise, it is said to be *unentangled*.

We typically consider systems consisting of two-dimensional particles (qubits), but it is sometimes convenient to work with higher dimensional particles. Since higher dimen-

sional spaces may be viewed as subspaces of suitable tensor powers of $\mathbb{C}^2$, we continue to describe such systems in terms of qubits.

Partial trace

The *partial trace over* A_1 is the unique linear transformation $\mathrm{Tr}_{A_1} : \mathbb{S}^{A_1 \times A_2} \to \mathbb{S}^{A_2}$, which satisfies

$$\mathrm{Tr}_{A_1}(\rho_1 \otimes \rho_2) = \mathrm{Tr}(\rho_1) \cdot \rho_2,$$

for all $\rho_1 \in \mathbb{S}^{A_1}$ and $\rho_2 \in \mathbb{S}^{A_2}$. More explicitly, given any matrix $X \in \mathbb{S}_+^{A_1 \times A_2}$ we define Tr_{A_1} as

$$\mathrm{Tr}_{A_1}(X) := \sum_{x_1 \in A_1} \left(e_{x_1}^* \otimes \mathrm{I}_{A_2} \right) X \left(e_{x_1} \otimes \mathrm{I}_{A_2} \right),$$

where $\{e_{x_1} : x_1 \in A_1\}$ is the standard basis for $\mathbb{C}^{A_1}$. In fact, the definition is independent of the choice of basis, so long as it is orthonormal. Note that the partial trace is positive, i.e., $\mathrm{Tr}_{A_1}(X) \in \mathbb{S}_+^{A_2}$ when $X \in \mathbb{S}_+^{A_1 \times A_2}$, and also trace-preserving. (In fact, it is a *completely positive* operation.) This ensures that the image of any density matrix under this operation, called its *reduced state*, is a well-defined density matrix.

Consider the scenario where two parties, Alice and Bob, hold parts of a quantum system which are jointly in some state ρ, i.e., they "share" a quantum state ρ over the space $\mathbb{C}^A \otimes \mathbb{C}^B$. Then the partial trace of ρ over one space characterizes the quantum state over the remaining space (if we are interested only in operations on the latter space). For example, $\mathrm{Tr}_A(\rho)$ is the density matrix representing Bob's half of the state and $\mathrm{Tr}_B(\rho)$ represents Alice's half. Note that ρ may not equal $\mathrm{Tr}_B(\rho) \otimes \mathrm{Tr}_A(\rho)$ in general.

Suppose we are given the density matrix $\rho \in \mathbb{S}_+^A$. We call the pure state $\psi \in \mathbb{C}^A \otimes \mathbb{C}^B$ a *purification* of ρ if $\mathrm{Tr}_B(\psi\psi^*) = \rho$. A purification always exists if $|B| \geq |A|$, and is in general not unique. An important property of purifications of the same state is that they are related to each other by a unitary operator: if $\mathrm{Tr}_B(\psi\psi^*) = \mathrm{Tr}_B(\phi\phi^*)$, then there exists a unitary U acting on $\mathbb{C}^B$ alone such that $\psi = (\mathrm{I}_A \otimes U)\phi$.

The partial trace operation is the quantum analogue of calculating marginal probability distributions. Consider the linear operator $\mathrm{Tr}_A : \mathbb{R}^{A \times B} \to \mathbb{R}^B$ defined by

$$[\mathrm{Tr}_A(v)]_y = \sum_{x \in A} v_{x,y} \ ,$$

for $y \in B$. This is called the partial trace over A. Note that $\mathrm{Tr}_A(p)$ gives the marginal distribution over B of the probability distribution $p \in \mathrm{Prob}^{A \times B}$. One may view probability distributions as diagonal positive semidefinite matrices with unit trace. Then, taking the partial trace (as defined for quantum states) corresponds exactly to the computation of marginal distributions.

Distance measures

Notions of distance between quantum states and probability distributions are very important in quantum cryptography. Here, we discuss two measures used in this paper and how they are related.

We define the *fidelity* of two nonnegative vectors $p, q \in \mathbb{R}_+^A$ as

$$\mathrm{F}(p, q) := \left(\sum_{x \in A} \sqrt{p_x} \sqrt{q_x} \right)^2$$

and the fidelity of two positive semidefinite matrices ρ_1 and ρ_2 as

$$\mathrm{F}(\rho_1, \rho_2) := \left\| \sqrt{\rho_1} \sqrt{\rho_2} \right\|_*^2 .$$

Notice, $\mathrm{F}(\rho_1, \rho_2) \geq 0$ with equality if and only if $\langle \rho_1, \rho_2 \rangle = 0$ and, if ρ_1 and ρ_2 are quantum states, $\mathrm{F}(\rho_1, \rho_2) \leq 1$ with equality if and only if $\rho_1 = \rho_2$. An analogous statement can be made for fidelity over probability vectors.

Fidelity has several useful properties, which we later use in this paper. We have occasion to consider fidelity only of probability distributions, and state the properties in terms of these. However, the following properties hold for quantum states as well. Fidelity is symmetric, positively homogeneous in both arguments, i.e., $\lambda \mathrm{F}(p, q) = \mathrm{F}(\lambda p, q) = \mathrm{F}(p, \lambda q)$ for all $\lambda > 0$, and is concave, i.e., $\mathrm{F}\left(\sum_{i=1}^n \lambda_i \, p_i, q \right) \geq \sum_{i=1}^n \lambda_i \, \mathrm{F}(p_i, q)$, for all $\lambda \in \mathrm{Prob}^n$.

Another distance measure is the *trace distance*. We define the trace distance between two probability vectors p and q, denoted $\Delta(p, q)$, as

$$\Delta(p, q) := \frac{1}{2} \left\| p - q \right\|_1 .$$

We similarly define the trace distance between two quantum states ρ_1 and ρ_2 as

$$\Delta(\rho_1, \rho_2) := \frac{1}{2} \left\| \rho_1 - \rho_2 \right\|_* .$$

Notice $\Delta(\rho_1, \rho_2) \geq 0$ with equality if and only if $\rho_1 = \rho_2$, and $\Delta(\rho_1, \rho_2) \leq 1$ with equality if and only if $\langle \rho_1, \rho_2 \rangle = 0$. An analogous statement can be made for the trace distance between probability vectors.

It is straightforward to show that for any Π with $0 \preceq \Pi \preceq \mathrm{I}$,

$$\mathrm{Tr}(\Pi(\rho_1 - \rho_2)) \quad \leq \quad \Delta(\rho_1, \rho_2) \ . \tag{1}$$

We now discuss two important notions in quantum cryptography. The first is how easily two states can be distinguished from each other. For example, if Alice gives to Bob

one of two states ρ_1 or ρ_2 chosen uniformly at random, then Bob can measure to learn whether he has been given ρ_1 or ρ_2 with maximum probability

$$\frac{1}{2} + \frac{1}{4}\left\|\rho_1 - \rho_2\right\|_* = \frac{1}{2} + \frac{1}{2}\Delta(\rho_1, \rho_2).$$

The second notion is *quantum steering*. Suppose Alice has given to Bob the A_1 part (i.e., the subsystem A_1 of qubits) of $\phi \in \mathbb{C}^{A_1 \times A_2}$. Now suppose she wants to modify and send the A_2 part in a way so as to convince Bob that a different state was sent, say $\psi \in \mathbb{C}^{A_1 \times A_2}$. Her most general strategy is to apply a quantum operation on A_2 (i.e., a sequence of unitary operations and measurements) before sending it to Bob. If Bob measures according to the POVM $(\psi\psi^*, \mathrm{I} - \psi\psi^*)$, Alice can convince him that the state is ψ with maximum probability

$$\mathrm{F}(\mathrm{Tr}_{A_2}(\psi\psi^*), \mathrm{Tr}_{A_2}(\phi\phi^*)) \ .$$

Trace distance and fidelity are closely related. The Fuchs-van de Graaf inequalities [FvdG99] illustrate this relationship:

Proposition 2.1 *For any finite dimensional quantum states* $\rho_1, \rho_2 \in \mathbb{S}_+^D$, *we have*

$$1 - \sqrt{\mathrm{F}(\rho_1, \rho_2)} \quad \leq \quad \Delta(\rho_1, \rho_2) \quad \leq \quad \sqrt{1 - \mathrm{F}(\rho_1, \rho_2)} \ .$$

2.4   Semidefinite programming

A natural model of optimization when studying quantum information is semidefinite programming. A *semidefinite program*, abbreviated as SDP, is an optimization problem of the form

$$
\begin{array}{rrcl}
\text{(P)} & \sup & \langle C, X \rangle & \\
& \text{subject to} \quad \mathcal{A}(X) & = & b, \\
& X & \in & \mathbb{S}_+^n,
\end{array}
$$

where $\mathcal{A} : \mathbb{S}^n \to \mathbb{R}^m$ is linear, $C \in \mathbb{S}^n$, and $b \in \mathbb{R}^m$. The SDPs that arise in quantum computation involve optimization over complex matrices. However, they may be transformed to the above standard form in a straightforward manner, by observing that Hermitian matrices form a real subspace of the vector space of $n \times n$ complex matrices. We give direct arguments as to why we may restrict ourselves to SDPs over real matrices when they arise in this article.

Similar to linear programs, every SDP has a dual. We can write the dual of (P) as

$$
\begin{array}{rrcl}
\text{(D)} & \inf & \langle b, y \rangle & \\
& \text{subject to} \quad \mathcal{A}^*(y) - S & = & C, \\
& S & \in & \mathbb{S}_+^n,
\end{array}
$$

where $\mathcal{A}^*$ is the adjoint of $\mathcal{A}$. We refer to (P) as the primal problem and to (D) as its dual. We say X is *feasible* for (P) if it satisfies the constraints $\mathcal{A}(X) = b$ and $X \in \mathbb{S}^n_+$, and (y, S) is feasible for (D) if $\mathcal{A}^*(y) - S = C, S \in \mathbb{S}^n_+$. The usefulness of defining the dual in the above manner is apparent in the following lemmas.

Lemma 2.2 (Weak duality) *For every X feasible for* (P) *and* (y, S) *feasible for* (D) *we have*

$$\langle C, X \rangle \leq \langle b, y \rangle .$$

Using weak duality, we can prove bounds on the optimal objective value of (P) and (D), i.e., the objective function value of any primal feasible solution yields a lower bound on (D) and the objective function value of any dual feasible solution yields an upper bound on (P).

Under mild conditions, we have that the optimal objective values of (P) and (D) coincide.

Lemma 2.3 (Strong duality) *If the objective function of* (P) *is bounded from above on the set of feasible solutions of* (P) *and there exists a strictly feasible solution, i.e., there exists $\bar{X} \succ 0$ such that $\mathcal{A}(\bar{X}) = b$, then* (D) *has an optimal solution and the optimal objective values of* (P) *and* (D) *coincide.*

A strictly feasible solution as in the above lemma is also called a *Slater point*.

Semidefinite programming has a powerful and rich duality theory and the interested reader is referred to [WSV00, TW12] and the references therein.

2.4.1 Second-order cone programming

The *second-order cone* (or *Lorentz cone*) in $\mathbb{R}^n$, $n \geq 2$, is defined as

$$\mathrm{SOC}^n := \{(x, t) \in \mathbb{R}^n : t \geq \|x\|_2\} .$$

A *second-order cone program*, denoted SOCP, is an optimization problem of the form

$$
\begin{aligned}
\text{(P)} \qquad \sup \quad & \langle c, x \rangle \\
\text{subject to} \quad Ax \;=\; & b, \\
x \;\in\; & \mathrm{SOC}^{n_1} \oplus \cdots \oplus \mathrm{SOC}^{n_k},
\end{aligned}
$$

where A is an $m \times (\sum_{i=1}^k n_k)$ matrix, $b \in \mathbb{R}^m$, and k is finite. We say that a feasible solution $\bar{x}$ is strictly feasible if $\bar{x}$ is in the interior of $\mathrm{SOC}^{n_1} \oplus \cdots \oplus \mathrm{SOC}^{n_k}$.

An SOCP also has a dual which can be written as

$$
\begin{aligned}
\text{(D)} \qquad \inf \quad & \langle b, y \rangle \\
\text{subject to} \quad A^{\mathrm{T}} y - s \;=\; & c, \\
s \;\in\; & \mathrm{SOC}^{n_1} \oplus \cdots \oplus \mathrm{SOC}^{n_k}.
\end{aligned}
$$

Note that weak duality and strong duality also hold for SOCPs for the properly modified definition of a strictly feasible solution.

A related cone, called the *rotated second-order cone*, is defined as

$$\text{RSOC}^n := \left\{ (a, b, x) \in \mathbb{R}^n : a, b \geq 0, \; 2ab \geq \|x\|_2^2 \right\}.$$

Optimizing over the rotated second-order cone is also called second-order cone programming because $(x, t) \in \text{SOC}^n$ if and only if $(t/2, t, x) \in \text{RSOC}^{n+1}$ and $(a, b, x) \in \text{RSOC}^n$ if and only if $(x, a, b, a + b) \in \text{SOC}^{n+1}$ and $a, b \geq 0$. In fact, both second-order cone constraints can be cast as positive semidefinite constraints:

$$t \geq \|x\|_2 \iff \begin{bmatrix} t & x^{\mathrm{T}} \\ x & t\,\mathrm{I} \end{bmatrix} \succeq 0 \quad \text{and} \quad a, b \geq 0, \; 2ab \geq \|x\|_2^2 \iff \begin{bmatrix} 2a & x^{\mathrm{T}} \\ x & b\,\mathrm{I} \end{bmatrix} \succeq 0.$$

There are some notable differences between semidefinite programs and second-order cone programs. One is that the algorithms for solving second-order cone programs can be more efficient and robust than those for solving semidefinite programs. We refer the interested reader to [Stu99, Stu02, Mit03, AG03] and the references therein.

3 Coin-flipping protocols

3.1 Strong coin-flipping

A strong coin-flipping protocol is a two-party quantum *communication protocol* in the style of Yao [Yao93]. We concentrate on a class of communication protocols relevant to coin-flipping. Informally, in such protocols, two parties Alice and Bob hold some number of qubits; the qubits with each party are initialized to a fixed pure state. The initial joint state is therefore unentangled across Alice and Bob. The two parties then "play" in turns. Suppose it is Alice's turn to play. Alice applies a unitary transformation on her qubits and then sends one or more qubits to Bob. Sending qubits does not change the overall superposition, but rather changes the ownership of the qubits. This allows Bob to apply his next unitary transformation on the newly received qubits. At the end of the protocol, each player makes a measurement of their qubits and announces the outcome as the result of the protocol.

Formally, the players Alice and Bob, hold some number of qubits, which initially factor into a tensor product $\mathbb{C}^{A_0} \otimes \mathbb{C}^{B_0}$ of Hilbert spaces. The qubits corresponding to $\mathbb{C}^{A_0}$ are in Alice's possession, and those in $\mathbb{C}^{B_0}$ are in Bob's possession. When the protocol starts, the qubits in $\mathbb{C}^{A_0}$ are initialized to some superposition $\psi_{A,0}$ and those in $\mathbb{C}^{B_0}$ to $\psi_{B,0}$, both of which specified by the protocol. The communication consists of $t \geq 1$ alternations of message exchange ("rounds"), in which the two players "play". Either party may play first. The protocol specifies a factorization of the joint state space just before each round, corresponding to the ownership of the qubits. In the ith round, $i \geq 1$, suppose it is Alice's

turn to play. Suppose the factorization of the state space just before the ith round is $\mathbb{C}^{A_{i-1}} \otimes \mathbb{C}^{B_{i-1}}$. Alice applies a unitary operator $U_{\mathrm{A},i}$ to the qubits in $\mathbb{C}^{A_{i-1}}$. Then, Alice sends some of her qubits to Bob. Formally, the space $\mathbb{C}^{A_{i-1}}$ is expressed as $\mathbb{C}^{A_i} \otimes \mathbb{C}^{M_i}$, where $\mathbb{C}^{A_i}$ is Alice's state space after the ith message is sent and $\mathbb{C}^{M_i}$ is the state space for the ith message. Consequently, Bob's state space after receiving the ith message is $\mathbb{C}^{B_i} = \mathbb{C}^{M_i} \otimes \mathbb{C}^{B_{i-1}}$. In the next round, Bob may thus apply a unitary operation to the qubits previously in Alice's control.

At the end of the t rounds of play, Alice and Bob observe the qubits in their possession according to some measurement. The outcomes of these measurements represent their outputs. We emphasize that there are no measurements until all rounds of communication are completed. A protocol with intermediate measurements may be transformed into this form by appealing to standard techniques [BV97].

Definition 3.1 (Strong coin-flipping) *A* strong coin-flipping protocol *is a two-party communication protocol as described above, in which the measurements of Alice and Bob are given by three-outcome POVMs* $(\Pi_{\mathrm{A},0}, \Pi_{\mathrm{A},1}, \Pi_{\mathrm{A,abort}})$ *and* $(\Pi_{\mathrm{B},0}, \Pi_{\mathrm{B},1}, \Pi_{\mathrm{B,abort}})$, *respectively. When both parties follow the protocol, they do not abort, i.e., only get outcomes in* $\{0,1\}$. *Further, each party outputs the same bit* $c \in \{0,1\}$ *and each binary outcome occurs with probability* $1/2$.

We are interested in the probabilities of the different outcomes in a coin-flipping protocol, when either party "cheats". Suppose Alice and Bob have agreed upon a protocol, i.e., a set of rules for the state initialization, communication, quantum operations, and measurements. What if Alice or Bob do not follow protocol? Suppose Alice is dishonest and would like to force an outcome of 0. She may use a different number of qubits for her private operations, so that her space $\mathbb{C}^{A'_i}$ may be much larger than $\mathbb{C}^{A_i}$. She may create any initial state she wants. During the communication, the only restriction is that she send a state of the correct dimension, e.g., if the protocol requires a message with 3 qubits in the first message, then Alice sends 3 qubits. Between messages, she may apply any quantum operation she wants on the qubits in her possession. At the end of the protocol, she may use a different measurement of her choice. For example, she may simply output "0" as this is her desired outcome (which corresponds to a trivial measurement). The rules that Alice chooses to follow instead of the protocol constitute a *cheating strategy*.

We would like to quantify the extent to which a cheating player can convince an honest one of a desired outcome, so we focus on runs of the protocol in which at most one party is dishonest. We analyze in this paper the maximum probability with which Alice (or Bob) can force a desired outcome in terms of the "bias", i.e., the advantage over $1/2$ that a cheating party can achieve.

Definition 3.2 (Bias) *For a given strong coin-flipping protocol, for each* $c \in \{0,1\}$, *define*

- $P^*_{\mathrm{A},c} := \sup \{\Pr[\textit{honest Bob outputs } c \textit{ when Alice may cheat}]\}$,

- $P^*_{\mathrm{B},c} := \sup \{\Pr[\textit{honest Alice outputs } c \textit{ when Bob may cheat}]\}$,

19

where the suprema are taken over all cheating strategies of the dishonest player. The bias ϵ of the protocol is defined as

$$\epsilon := \max\{P_{A,0}^*, P_{A,1}^*, P_{B,0}^*, P_{B,1}^*\} - 1/2 \ .$$

3.2 An example protocol

Here we describe a construction of strong coin-flipping protocols based on quantum *bit-commitment* [ATVY00, Amb01, SR01, KN04] that consists of three messages. First, Alice chooses a uniformly random bit a, creates a state of the form

$$\psi_a \in \mathbb{C}^A \otimes \mathbb{C}^{A'}$$

and sends A to Bob, i.e., the first message consists of qubits corresponding to the space $\mathbb{C}^A$. (For ease of exposition, we use this language throughout, i.e., refer to qubits by the labels of the corresponding spaces.) This first message is the *commit stage* since she potentially gives some information about the bit a, for which she may be held accountable later. Then Bob chooses a uniformly random bit b and sends it to Alice. Alice then sends a and A' to Bob. Alice's last message is the *reveal stage*. Bob checks to see if the qubits he received are in state ψ_a (we give more details about this step below). If Bob is convinced that the state is correct, they both output 0 when $a = b$, or 1 if $a \neq b$, i.e., they output the XOR of a and b.

This description can be cast in the form of a quantum protocol as presented in Section 3.1: we can encode 0 as basis state e_0 and 1 as e_1, we can simulated the generation of a uniformly random bit by preparing a uniform superposition over the two basis states, and we can "send" qubits by permuting their order (a unitary operation) so that they are part of the message subsystem. In fact, we can encode an entirely classical protocol using a quantum one in this manner. A more general protocol of this kind is described formally in Section 3.3.

We present a protocol from [KN04] which follows the above framework.

Definition 3.3 (Coin-flipping protocol example)
Let $A := \{0, 1, 2\}$, $A' := A$, and let $\mathbb{C}^A$ and $\mathbb{C}^{A'}$ be spaces for Alice's two messages.

- *Alice chooses $a \in \{0, 1\}$ uniformly at random and creates the state*

$$\psi_a = \frac{1}{\sqrt{2}} e_a \otimes e_a + \frac{1}{\sqrt{2}} e_2 \otimes e_2 \in \mathbb{C}^A \otimes \mathbb{C}^{A'},$$

  where $\{e_0, e_1, e_2\}$ are standard basis vectors. Alice sends the A part of ψ_a to Bob.

- *Bob chooses $b \in \{0, 1\}$ uniformly at random and sends it to Alice.*

- *Alice reveals a to Bob and sends the rest of ψ_a, i.e., she sends A'.*

- *Bob checks to see if the state sent by Alice is ψ_a, i.e., he checks to see if Alice has tampered with the state during the protocol. The measurement on $\mathbb{C}^A \otimes \mathbb{C}^{A'}$ corresponding to this check is*

$$(\Pi_{\text{accept}} := \psi_a \psi_a^*, \quad \Pi_{\text{abort}} := I - \Pi_{\text{accept}}).$$

   If the measurement outcome is "abort" then Bob aborts the protocol.

- *Each player outputs the XOR of the two bits, i.e., Alice outputs $a \oplus b'$, where b' is the bit she received in the second round, and if he does not abort, Bob outputs $a' \oplus b$, where a' is the bit received by him in the third round.*

In the honest case, Bob does not abort since $\langle \Pi_{\text{abort}}, \psi_a \psi_a^* \rangle = 0$. Furthermore, Alice and Bob get the same outcome which is uniformly random. Therefore, this is a well-defined coin-flipping protocol. We now sketch a proof that this protocol has bias $\epsilon = 1/4$.

Bob cheating: We consider the case when Bob cheats towards 0; the analysis of cheating towards 1 is similar. If Bob wishes to maximize the probability of outcome 0, he has to maximize the probability that the bit b he sends equals a. In other words, he may only cheat by measuring Alice's first message to try to learn a, then choose b suitably to force the desired outcome. Define $\rho_a := \text{Tr}_{A'}(\psi_a \psi_a^*)$. This is the reduced state of the A-qubits Bob has after the first message. Recall Bob can learn the value of a with probability

$$\frac{1}{2} + \frac{1}{2}\Delta(\rho_0, \rho_1) = 3/4 \ ,$$

and this bound can be achieved. This strategy is independent of the outcome Bob desires, thus $P_{\text{B},0}^* = P_{\text{B},1}^* = 3/4$.

Alice cheating: Alice's most general cheating strategy is to send a state in the first message such that she can decide the value of a after receiving b, and yet pass Bob's cheat detection step with maximum probability. For example, if Alice wants outcome 0 then she returns $a = b$ and if she wants outcome 1, she returns $a = \bar{b}$. Alice always gets the desired outcome as long as Bob does not detect her cheating. As a primer for more complicated protocols, we show an SDP formulation for a cheating Alice based on the above cheating strategy description. There are three important quantum states to consider here. The first is Alice's first message, which we denote as $\sigma \in \mathbb{S}_+^A$. The other two states are the states Bob has at the end of the protocol depending on whether $b = 0$ or $b = 1$, we denote them by $\sigma_b \in \mathbb{S}_+^{A \otimes A'}$. Note that $\text{Tr}_{A'}(\sigma_0) = \text{Tr}_{A'}(\sigma_1) = \sigma$ since they are consistent with the first message σ—Alice does not know b when σ is sent. However, they could be different on A' because Alice may apply some quantum operation depending upon b before sending the A' qubits. Then Alice can cheat with probability given by the optimal

objective value of the following SDP:

$$
\begin{aligned}
\sup \quad & \tfrac{1}{2}\langle \psi_0\psi_0^*, \sigma_0\rangle \;+\; \tfrac{1}{2}\langle \psi_1\psi_1^*, \sigma_1\rangle \\
\text{subject to} \quad \mathrm{Tr}_{A'}(\sigma_b) &= \sigma, && \text{for all } b \in \{0,1\}, \\
\mathrm{Tr}(\sigma) &= 1, \\
\sigma &\in \mathbb{S}_+^A, \\
\sigma_b &\in \mathbb{S}_+^{A\otimes A'}, && \text{for all } b \in \{0,1\},
\end{aligned}
$$

recalling that the partial trace is trace-preserving, any unit trace, positive semidefinite matrix represents a valid quantum state, and that two purifications of the same density matrix are related to each other by a unitary transformation on the part that is traced out.

A few words about the above optimization problem are in order here. First, the restriction to real positive semidefinite matrices does not change the optimum: the real part of any feasible set of complex matrices $\sigma, \sigma_0, \sigma_1$ is also feasible, and has the same objective function value. Second, using straightforward transformations, we may verify that the problem is an SDP of the form defined in Section 2.4.

It has been shown [SR01, Amb01, NS03] that the optimal objective value of this problem is

$$
\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\rho_0, \rho_1)} = 3/4
$$

given by the optimal solution $(\sigma_0, \sigma_1, \sigma) = (\psi\psi^*, \psi\psi^*, \mathrm{Tr}_{A'}(\psi\psi^*))$, where

$$
\psi = \sqrt{\frac{1}{6}}\, e_0 \otimes e_0 + \sqrt{\frac{1}{6}}\, e_1 \otimes e_1 + \sqrt{\frac{2}{3}}\, e_2 \otimes e_2 \ .
$$

Therefore, the bias of this protocol is $\max\{P_{A,0}^*, P_{A,1}^*, P_{B,0}^*, P_{B,1}^*\} - 1/2 = 3/4 - 1/2 = 1/4$. Using Proposition 2.1, it was shown in [Amb01] that for any ρ_0 and ρ_1, we have

$$
\max\left\{\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\rho_0, \rho_1)}, \frac{1}{2} + \frac{1}{2}\Delta(\rho_0, \rho_1)\right\} - 1/2 \quad \geq \quad 1/4 \ .
$$

Thus, we cannot improve the bias by simply changing the starting states in this type of protocol, suggesting a substantial change of the form of the protocol is necessary to find a smaller bias.

3.3 A family of protocols

We now consider a family of protocols which generalizes the above idea. Alice and Bob each flip a coin and commit to their respective bits by exchanging quantum states. Then they reveal their bits and send the remaining part of the commitment state. Each party checks the received state against the one they expect, and abort the protocol if they detect an inconsistency. They output the XOR of the two bits otherwise. We see that this is uniformly random, when a and b are uniformly random.

22

The difficulty in designing a good protocol is in deciding how Alice and Bob commit to their bits. If Alice or Bob leaks too much information early, then the other party has more freedom to cheat. Thus, we try to maintain a balance between the two parties so as to minimize the bias they can achieve by cheating.

Consider the following Cartesian product of finite sets $A = A_1 \times \cdots \times A_n$. These are used for Alice's first n messages to Bob. Suppose we are given two probability distributions $\alpha_0, \alpha_1 \in \text{Prob}^A$. Define the following two quantum states

$$\psi_a = \sum_{x \in A} \sqrt{\alpha_{a,x}}\, e_x \otimes e_x \in \mathbb{C}^A \otimes \mathbb{C}^{A'} \quad \text{for} \quad a \in \{0,1\},$$

where $A' = A$. The reason we define the state over $\mathbb{C}^A$ and a copy is because in the protocol, Alice sends states in $\mathbb{C}^A$ while retaining copies in $\mathbb{C}^{A'}$ for herself. We may simulate Alice's choice of uniformly random a and the corresponding messages by preparing the initial state

$$\psi := \sum_{a \in \{0,1\}} \frac{1}{\sqrt{2}}\, e_a \otimes e_a \otimes \psi_a \in \mathbb{C}^{A_0} \otimes \mathbb{C}^{A_0'} \otimes \mathbb{C}^A \otimes \mathbb{C}^{A'},$$

where $A_0 = A_0' = \{0,1\}$ are used for two copies of Alice's bit a, one for Bob and a copy for herself.

We now describe the setting for Bob's messages. Consider the following Cartesian product of finite sets $B = B_1 \times \cdots \times B_n$ used for Bob's first n messages to Alice. Suppose we are given two probability distributions $\beta_0, \beta_1 \in \text{Prob}^B$. Define the following two quantum states

$$\phi_b = \sum_{y \in B} \sqrt{\beta_{b,y}}\, e_y \otimes e_y \in \mathbb{C}^B \otimes \mathbb{C}^{B'} \quad \text{for} \quad b \in \{0,1\},$$

where $B' = B$. Bob's choice of uniformly random b, and the corresponding messages may be simulated by preparing the initial state

$$\phi := \sum_{b \in \{0,1\}} \frac{1}{\sqrt{2}}\, e_b \otimes e_b \otimes \phi_b \in \mathbb{C}^{B_0} \otimes \mathbb{C}^{B_0'} \otimes \mathbb{C}^B \otimes \mathbb{C}^{B'},$$

where $B_0 = B_0' = \{0,1\}$ are used for two copies of Bob's bit b, one for Alice and a copy for himself.

We now describe the communication and cheat detection in the protocol.

Definition 3.4 (Coin-flipping protocol based on bit-commitment) *A coin-flipping protocol based on bit-commitment is specified by a 4-tuple of probability distributions $(\alpha_0, \alpha_1, \beta_0, \beta_1)$ that define states ψ, ϕ as above.*

- *Alice prepares the state ψ and Bob prepares the state ϕ as defined above.*

- *For i from 1 to n: Alice sends $\mathbb{C}^{A_i}$ to Bob who replies with $\mathbb{C}^{B_i}$.*

- *Alice fully reveals her bit by sending $\mathbb{C}^{A'_0}$. She also sends $\mathbb{C}^{A'}$ which Bob uses later to check if she was honest. Bob then reveals his bit by sending $\mathbb{C}^{B'_0}$. He also sends $\mathbb{C}^{B'}$ which Alice uses later to check if he was honest.*

- *Alice observes the qubits in her possession according to the measurement $(\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\text{abort}})$ defined on the space $\mathbb{S}_+^{A_0 \times B'_0 \times B \times B'}$, where*

$$\Pi_{A,0} := \sum_{b \in \{0,1\}} e_b e_b^* \otimes e_b e_b^* \otimes \phi_b \phi_b^*, \quad \Pi_{A,1} := \sum_{b \in \{0,1\}} e_{\bar{b}} e_b^* \otimes e_b e_b^* \otimes \phi_b \phi_b^*,$$

and $\Pi_{A,\text{abort}} := I - \Pi_{A,0} - \Pi_{A,1}$.

- *Bob observes the qubits in his possession according to the measurement $(\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\text{abort}})$ defined on the space $\mathbb{S}_+^{B_0 \times A'_0 \times A \times A'}$, where*

$$\Pi_{B,0} := \sum_{a \in \{0,1\}} e_a e_a^* \otimes e_a e_a^* \otimes \psi_a \psi_a^*, \quad \Pi_{B,1} := \sum_{a \in \{0,1\}} e_{\bar{a}} e_a^* \otimes e_a e_a^* \otimes \psi_a \psi_a^*,$$

and $\Pi_{B,\text{abort}} := I - \Pi_{B,0} - \Pi_{B,1}$. (These last two steps can be interchanged.)

Note that the measurements check two things. First, they check whether the outcome is 0 or 1. The first two terms determine this, i.e., whether $a = b$ or if $a \neq b$. Second, they check whether the other party was honest. For example, if Alice's measurement projects onto a subspace where $b = 0$ and Bob's messages are not in state ϕ_0, then Alice knows Bob has cheated and aborts. A six-round protocol is depicted in Figure 1, above.

We could also consider the case where Alice and Bob choose a and b with different probability distributions, i.e., we could change the $1/\sqrt{2}$ in the definitions of ψ and ϕ to other values depending on a or b. This causes the honest outcome probabilities to not be uniformly random and this no longer falls into our definition of a coin-flipping protocol. However, sometimes such "unbalanced" coin-flipping protocols are useful, see [CK09]. We note that our optimization techniques in Section 4 are robust enough to handle the analysis of such modifications.

Notice that our protocol is parameterized by the four probability distributions α_0, α_1, β_0, and β_1. It seems to be a very difficult problem to solve for the choice of these parameters that gives us the least bias. Indeed, we do not even have an upper bound on the dimension of these parameters in an optimal protocol. However, we can solve for the bias of a protocol once these parameters are fixed using the optimization techniques in Section 4. Once we have a means for computing the bias given some choice of fixed parameters, we then turn our attention to solving for the best choice of parameters. We use the heuristics in Sections 5 and 6 to design an algorithm in Section 7 to search for these.

Alice prepares $\psi \in \mathbb{C}^{A_0 \times A_0' \times A_1 \times A_2 \times A_1' \times A_2'}$ Bob prepares $\phi \in \mathbb{C}^{B_0 \times B_0' \times B_1 \times B_2 \times B_1' \times B_2'}$

Alice sends $\mathbb{C}^{A_1}$ $(x_1 \in A_1)$

Bob sends $\mathbb{C}^{B_1}$ $(y_1 \in B_1)$

Alice sends $\mathbb{C}^{A_2}$ $(x_2 \in A_2)$

Bob sends $\mathbb{C}^{B_2}$ $(y_2 \in B_2)$

Alice sends $\mathbb{C}^{A_0' \times A_1' \times A_2'}$ $(a \in \{0,1\}$ and a copy of $x_1, x_2)$

Bob sends $\mathbb{C}^{B_0' \times B_1' \times B_2'}$ $(b \in \{0,1\}$ and a copy of $y_1, y_2)$

Alice checks if Bob cheated   Bob checks if Alice cheated

Alice and Bob output $a \oplus b$ if no cheating is detected
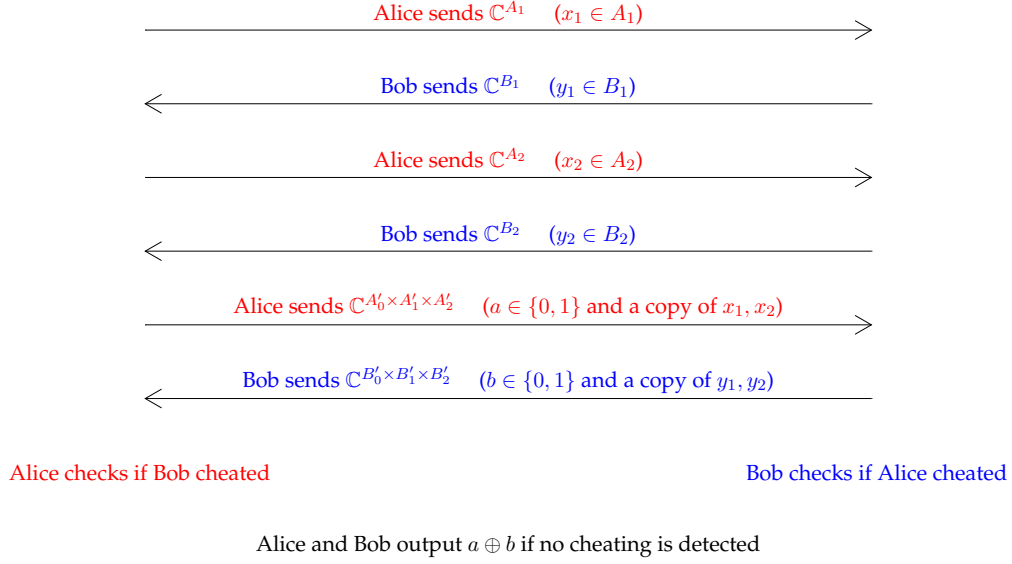
Figure 1: Six-round coin-flipping protocol based on bit-commitment. Alice's actions are in red and Bob's actions are in blue.

4 Cheating strategies as optimization problems

In this section, we show that the optimal cheating strategy of a player in a coin-flipping protocol is characterized by highly structured semidefinite programs.

4.1 Characterization by semidefinite programs

We start by formulating strategies for cheating Bob and cheating Alice as semidefinite optimization problems as proposed by Kitaev [Kit02]. The extent to which Bob can cheat is captured by the following lemma.

Lemma 4.1 *The maximum probability with which cheating Bob can force honest Alice to accept* $c \in \{0, 1\}$ *is given by the optimal objective value of the following SDP:*

$$
\begin{aligned}
\sup \quad & \langle \rho_F, \Pi_{A,c} \rangle \\
\text{subject to} \quad \mathrm{Tr}_{B_1}(\rho_1) &= \mathrm{Tr}_{A_1}(\psi\psi^*), \\
\mathrm{Tr}_{B_j}(\rho_j) &= \mathrm{Tr}_{A_j}(\rho_{j-1}), \quad \text{for all } j \in \{2, \dots, n\}, \\
\mathrm{Tr}_{B' \times B_0'}(\rho_F) &= \mathrm{Tr}_{A' \times A_0'}(\rho_n), \\
\rho_j &\in \mathbb{S}_+^{A_0 \times A_0' \times B_1 \times \cdots \times B_j \times A_{j+1} \times \cdots \times A_n \times A'}, \quad \text{for all } j \in \{1, \dots, n\}, \\
\rho_F &\in \mathbb{S}_+^{A_0 \times B_0' \times B \times B'}.
\end{aligned}
$$

Furthermore, an optimal cheating strategy for Bob may be derived from an optimal feasible solution of this SDP.

We depict Bob cheating, and the context of the SDP variables, in a six-round protocol in Figure 2, on the next page.
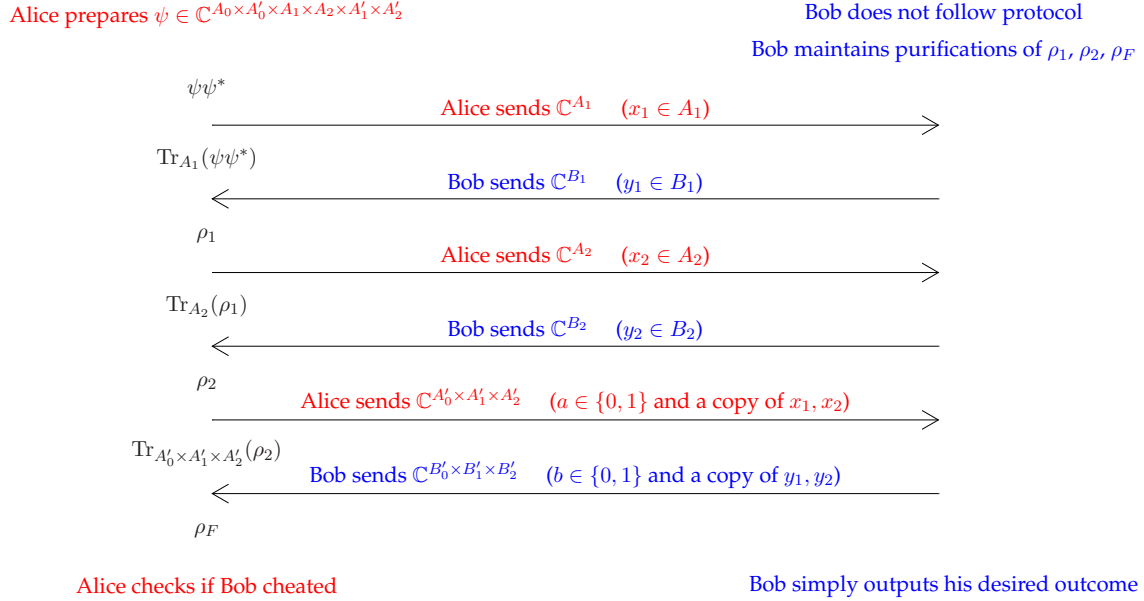
Figure 2: Bob cheating in a six-round protocol.

We call the SDP Lemma 4.1 Bob's cheating SDP. In a similar fashion, we can formulate Alice's cheating SDP.

Lemma 4.2 *The maximum probability with which cheating Alice can force honest Bob to accept* $c \in \{0, 1\}$ *is given by the optimal objective value of the following SDP:*

$$
\begin{array}{rl}
\sup & \left\langle \sigma_F, \Pi_{\mathrm{B},c} \otimes \mathrm{I}_{B'_0 \times B'} \right\rangle \\
\text{subject to} & \mathrm{Tr}_{A_1}(\sigma_1) = \phi \phi^*, \\
& \mathrm{Tr}_{A_2}(\sigma_2) = \mathrm{Tr}_{B_1}(\sigma_1), \\
& \qquad\qquad \vdots \\
& \mathrm{Tr}_{A_n}(\sigma_n) = \mathrm{Tr}_{B_{n-1}}(\sigma_{n-1}), \\
& \mathrm{Tr}_{A' \otimes A'_0}(\sigma_F) = \mathrm{Tr}_{B_n}(\sigma_n), \\
& \sigma_j \in \mathbb{S}_+^{B_0 \times B'_0 \times A_1 \times \cdots \times A_j \times B_j \times \cdots \times B_n \times B'}, \\
& \qquad \text{for all } j \in \{1, \ldots, n\}, \\
& \sigma_F \in \mathbb{S}_+^{B_0 \times B'_0 \times A'_0 \times A \times A' \times B'}.
\end{array}
$$

Furthermore, we may derive an optimal cheating strategy for Alice from an optimal feasible solution to this SDP.

For completeness, we present proofs of these lemmas in Appendix A.

We depict Alice cheating, and the context of her SDP variables, in a six-round protocol in Figure 3.
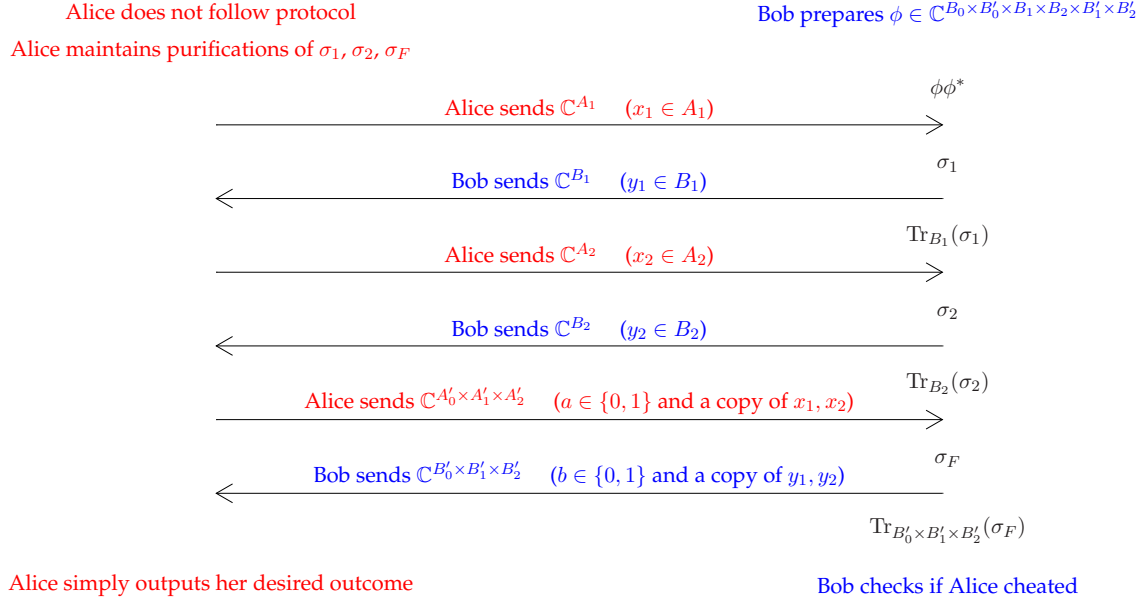
Figure 3: Alice cheating in a six-round protocol.

Analyzing and solving these problems computationally gets increasingly difficult and time consuming as n increases, since the dimension of the variables increases exponen-

tially in n. In the analysis of the bias, we make use of the following results which simplify the underlying optimization problems without changing their optimal objective values.

Definition 4.3 *We define* Bob's cheating polytope, *denoted as $\mathcal{P}_B$, as the set of all vectors $(p_1, p_2, \ldots, p_n)$ such that*

$$
\begin{aligned}
\mathrm{Tr}_{B_1}(p_1) &= e_{A_1}, \\
\mathrm{Tr}_{B_2}(p_2) &= p_1 \otimes e_{A_2}, \\
&\vdots \\
\mathrm{Tr}_{B_n}(p_n) &= p_{n-1} \otimes e_{A_n}, \\
p_j &\in \mathbb{R}_+^{A_1 \times B_1 \times \cdots \times A_j \times B_j}, \text{ for all } j \in \{1, \ldots, n\},
\end{aligned}
$$

where e_{A_j} denotes the vector of all ones on the corresponding space $\mathbb{C}^{A_j}$.

We can now define a simpler "reduced" problem that captures Bob's optimal cheating probability.

Theorem 4.4 (Bob's Reduced Problem) *The maximum probability with which cheating Bob can force honest Alice to accept outcome $c \in \{0, 1\}$ is given by the optimal objective function value of the following convex optimization problem*

$$
P_{B,c}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \mathrm{F}\left((\alpha_a \otimes \mathrm{I}_B)^{\mathrm{T}} p_n, \, \beta_{a \oplus c}\right) : (p_1, \ldots, p_n) \in \mathcal{P}_B \right\},
$$

where the arguments of the fidelity functions are probability distributions over B.

The connection between the fidelity function and semidefinite programming is detailed in the next subsection. A proof of the above theorem is presented in Appendix B.

We can also define Alice's cheating polytope.

Definition 4.5 *We define* Alice's cheating polytope, *denoted as $\mathcal{P}_A$, as the set of all vectors $(s_1, s_2, \ldots, s_n, s)$ satisfying*

$$
\begin{aligned}
\mathrm{Tr}_{A_1}(s_1) &= 1, \\
\mathrm{Tr}_{A_2}(s_2) &= s_1 \otimes e_{B_1}, \\
&\vdots \\
\mathrm{Tr}_{A_n}(s_n) &= s_{n-1} \otimes e_{B_{n-1}}, \\
\mathrm{Tr}_{A_0'}(s) &= s_n \otimes e_{B_n}, \\
s_1 &\in \mathbb{R}_+^{A_1}, \\
s_j &\in \mathbb{R}_+^{A_1 \times B_1 \times \cdots \times B_{j-1} \times A_j}, \text{ for all } j \in \{2, \ldots, n\}, \\
s &\in \mathbb{R}_+^{A_0' \times A \times B},
\end{aligned}
$$

where e_{B_j} denotes the vector of all ones on the corresponding space $\mathbb{C}^{B_j}$.

28

Now we can define Alice's reduced problem.

Theorem 4.6 (Alice's Reduced Problem) *The maximum probability with which cheating Alice can force honest Bob to accept outcome $c \in \{0, 1\}$ is given by the optimal objective function value of the following convex optimization problem*

$$P_{\mathrm{A},c}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \sum_{y \in B} \beta_{a \oplus c, y} \; \mathrm{F}(s^{(a,y)}, \alpha_a) \; : \; (s_1, \ldots, s_n, s) \in \mathcal{P}_{\mathrm{A}} \right\},$$

where $s^{(a,y)} \in \mathbb{R}_+^A$ is the restriction of s with the indices (a, y) fixed, i.e., $[s^{(a,y)}]_x := s_{a,x,y}$.

We postpone a proof of the above theorem until Appendix B.

We note here that we can get similar SDPs and reductions if Alice chooses a with a non-uniform probability distribution and similarly for Bob. It only changes the multiplicative factor $1/2$ in the reduced problems to something that depends on a (or b) and the proofs are nearly identical to those in the appendix.

We point out that the reduced problems are also semidefinite programs. The containment of the variables in a polytope is captured by linear constraints, so it suffices to express the objective function as a linear functional of an appropriately defined positive semidefinite matrix variable.

Lemma 4.7 *For any $p, q \in \mathbb{R}_+^A$, we have*

$$\mathrm{F}(p, q) = \max \left\{ \left\langle X, \sqrt{p}\sqrt{p}^{\mathrm{T}} \right\rangle : \mathrm{diag}(X) = q, \; X \in \mathbb{S}_+^A \right\}.$$

Proof: Notice that $\bar{X} := \sqrt{q}\sqrt{q}^{\mathrm{T}}$ is a feasible solution to the SDP with objective function value $\mathrm{F}(p, q)$. All that remains to show is that it is an optimal solution. If $p = 0$, then we are done, so assume $p \neq 0$. The dual can be written as

$$\inf \{ \langle y, q \rangle : \mathrm{Diag}(y) \succeq \sqrt{p}\sqrt{p}^{\mathrm{T}}, y \in \mathbb{R}^A \}.$$

Define y, as a function of $\varepsilon > 0$, entry-wise for each $x \in A$ as

$$y_x(\varepsilon) := \begin{cases} (\sqrt{\mathrm{F}(p,q)} + \varepsilon) \frac{\sqrt{p_x}}{\sqrt{q_x}} & \text{if} \quad p_x, q_x > 0, \\ \frac{(\sqrt{\mathrm{F}(p,q)} + \varepsilon)\|p\|_1}{\varepsilon} & \text{if} \quad q_x = 0, \\ \varepsilon & \text{if} \quad p_x = 0, q_x > 0. \end{cases}$$

We can check that $\langle y(\varepsilon), q \rangle \to \mathrm{F}(p, q)$ as $\varepsilon \to 0$, so it suffices to show that $y(\varepsilon)$ is dual feasible for all $\varepsilon > 0$. For any $y > 0$,

$$\begin{aligned} \mathrm{Diag}(y) \succeq \sqrt{p}\sqrt{p}^{\mathrm{T}} &\iff \mathrm{I}_A \succeq \mathrm{Diag}(y)^{-1/2}\sqrt{p}\sqrt{p}^{\mathrm{T}}\mathrm{Diag}(y)^{-1/2} \\ &\iff 1 \geq \sqrt{p}^{\mathrm{T}}\mathrm{Diag}(y)^{-1}\sqrt{p} \\ &\iff 1 \geq \sum_{x \in A} \frac{p_x}{y_x}, \end{aligned}$$

29

noting $\mathrm{Diag}(y)^{-1/2}\sqrt{p}\sqrt{p}^{\mathrm{T}}\mathrm{Diag}(y)^{-1/2}$ is rank 1 so the largest eigenvalue is equal to its trace. From this, we can check that $y(\varepsilon)$ is feasible for all $\varepsilon > 0$. $\quad\square$

The optimization problem in Lemma 4.7 remains an SDP if we replace q with a variable constrained to be in a polytope. Therefore, the reduced problems in Theorems 4.4 and 4.6 can be modelled as semidefinite programs.

4.2 SOCP formulations for the reduced problems

In this section, we show that the reduced SDPs can be modelled using a simpler class of optimization problems, second-order cone programs. We elaborate on this below and explain the significance to solving these problems computationally.

We start by first explaining how to model fidelity as an SOCP. Suppose we are given the problem

$$\max\left\{\sqrt{\mathrm{F}(p,q)} : q \in \mathbb{R}^n_+ \cap S\right\} = \max\left\{\sum_{i=1}^{n} \sqrt{p_i}\, t_i : t_i^2 \leq q_i,\ \forall i \in \{1,\ldots,n\},\ q \in \mathbb{R}^n_+ \cap S\right\},$$

where $p \in \mathbb{R}^n_+$ and $S \subseteq \mathbb{R}^n$. We can replace $t_i^2 \leq q_i$ with the equivalent constraint $(1/2, q_i, t_i) \in \mathrm{RSOC}^3$, for all $i \in \{1,\ldots,n\}$. Therefore, we can maximize the fidelity using n rotated second-order cone constraints.

For the same reason, we can use second-order cone programming to solve a problem of the form

$$\max\left\{\sum_{j=1}^{m} a_j \sqrt{\mathrm{F}(p_j,q_j)} : (q_1,\ldots,q_m) \in \mathbb{R}^{mn}_+ \cap S'\right\},$$

where $a \in \mathbb{R}^m_+$ and $S' \subseteq \mathbb{R}^{mn}$. However, this does not apply directly to the reduced problems since we need to optimize over a linear combination of fidelities and $f(x) = x^2$ is not a concave function. For example, Alice's reduced problem is of the form

$$\max\left\{\sum_{j=1}^{m} a_j\, \mathrm{F}(p_j,q_j) : (q_1,\ldots,q_m) \in \mathbb{R}^{mn}_+ \cap S'\right\}.$$

The root of this problem arises from the fact that the fidelity function, which is concave, is a composition of a concave function with a convex function, thus we cannot break it into these two steps. Even though the above analysis does not work to capture the reduced problems as SOCPs, it does have a desirable property that it only uses $O(n)$ second-order cone constraints and perhaps this formulation will be useful for future applications.

We now explain how to model the reduced problems as SOCPs directly.

Lemma 4.8 *For $p, q \in \mathbb{R}_+^n$, we have*

$$\mathrm{F}(p, q) = \max \left\{ \frac{1}{\sqrt{2}} \sum_{i,j=1}^n \sqrt{p_i p_j}\, t_{i,j} : (q_i, q_j, t_{i,j}) \in \mathrm{RSOC}^3, \text{ for all } i, j \in \{1, \ldots, n\} \right\}.$$

Proof: For every $i, j \in \{1, , \ldots, n\}$, we have $(q_i, q_j, t_{i,j}) \in \mathrm{RSOC}^3$ if and only if $q_i, q_j \geq 0$, and $2 q_i q_j \geq t_{i,j}^2$. Thus, $t_{i,j} = \sqrt{2 q_i q_j}$ is optimal with objective function value $\mathrm{F}(p, q)$. $\qquad\square$

This lemma provides an SOCP representation for the hypograph of the fidelity function. Recall that the hypograph of a concave function is a convex set. Also, the dimension of the hypograph of $\mathrm{F}(\cdot, q) : \mathbb{R}_+^n \to \mathbb{R}$ is equal to n (assuming $q > 0$). Since the hypograph is $O(n)$-dimensional and convex, there exists a *self-concordant barrier function* for the set with complexity parameter $O(n)$, shown by Nesterov and Nemirovski [NN94]. This allows the derivation of interior-point methods for the underlying convex optimization problem which use $O(\sqrt{n} \log(1/\varepsilon))$ iterations, where ε is an accuracy parameter. The above lemma uses $\Omega(n^2)$ second-order cone constraints and the usual treatment of these "cone constraints" with optimal self-concordant barrier functions lead to interior-point methods with an iteration complexity bound of $O(n \log(1/\varepsilon))$. It is conceivable that there exist better convex representations of the hypograph of the fidelity function than the one we provided in Lemma 4.8.

We can further simplify the reduced problems using fewer SOC constraints than derived above. We first consider the dual formulation of the reduced problems, so as to avoid the hypograph of the fidelity function.

Using Lemma 4.7, we write Alice's reduced problem for forcing outcome 0 as an SDP. The dual of this SDP is

$$
\begin{array}{rlll}
\inf & z_1 & & \\
\text{subject to} & z_1 \cdot e_{A_1} & \geq & \mathrm{Tr}_{B_1}(z_2), \\
& z_2 \otimes e_{A_2} & \geq & \mathrm{Tr}_{B_2}(z_3), \\
& \vdots & & \\
& z_n \otimes e_{A_n} & \geq & \mathrm{Tr}_{B_n}(z_{n+1}), \\
& \mathrm{Diag}(z_{n+1}^{(y)}) & \succeq & \tfrac{1}{2} \beta_{a,y} \sqrt{\alpha_a} \sqrt{\alpha_a}^{\mathrm{T}}, & \text{for all } a \in \{0,1\}, y \in B, \\
& z_1 & \in & \mathbb{R}, & \\
& z_i & \in & \mathbb{R}^{A_1 \times B_1 \times \cdots \times A_{i-1} \times B_{i-1}}, & \text{for all } i \in \{2, \ldots, n+1\}, \\
\text{where} & z_{n+1,x}^{(y)} & = & z_{n+1, x_1 y_1 x_2 y_2 \cdots, x_n y_n}, & \text{for all } x \in A, y \in B.
\end{array}
$$

The only nonlinear constraint in the above problem is of the form

$$\mathrm{Diag}(z) \succeq \sqrt{q} \sqrt{q}^{\mathrm{T}},$$

for some fixed $q \geq 0$. From the proof of Lemma 4.7, we see that for z which is positive in every coordinate

$$\mathrm{Diag}(z) \succeq \sqrt{q} \sqrt{q}^{\mathrm{T}} \iff \langle z^{-1}, q \rangle \leq 1.$$

So, it suffices to characterize inverses using SOCP constraints which can be done by considering
$$(z_i, r_i, \sqrt{2}) \in \text{RSOC} \iff r_i \geq z_i^{-1}.$$

With this observation, we can write the dual of Alice and Bob's reduced problems using $O(n)$ RSOC constraints for each fidelity function in the objective function as opposed to $\Omega(n^2)$ constraints as above.

4.3   Numerical performance of SDP formulation vs. SOCP formulation

Since the search algorithm designed in this paper examines the optimal cheating probabilities of many protocols (more than 10^{16}) we are concerned with the efficiency of solving the reduced problems. In this subsection, we discuss the efficiency of this computation. Our computational platform is an SGI XE C1103 with 2x 3.2 GHz 4-core Intel X5672 x86 CPUs processor, and 10 GB memory, running Linux. The reduced problems were solved using SeDuMi 1.3, a program for solving semidefinite programs and rotated second-order cone programs in Matlab (Version 7.12.0.635) [Stu99, Stu02].

Table 1 (on the next page) compares the computation of Alice's reduced problem in a four-round protocol for forcing an outcome of 0 with 5-dimensional messages. The top part of the table presents the average running time, the maximum running time, and the worst gap (the maximum of the extra time needed to solve the problem compared to the other formulation). The bottom part of the table presents the average number of iterations, the average feasratio, the average timing (the time spent in preprocessing, iterations, and postprocessing, respectively), and the average cpusec.

Table 1 suggests that solving the rotated second-order cone programs are comparable to solving the semidefinite programs. However, before testing the other three cheating probabilities, we test the performance of the two formulations from Table 1 in a setting that appears more frequently in the search. In particular, the searches detailed in Section 8 deal with many protocols with very sparse parameters. We retest the values in Table 1 when we force the first entry of α_0, the second entry of α_1, the third entry of β_0, and the fourth entry of β_1 to all be 0. The results are shown in Table 2.

As we can see, the second-order cone programming formulation stumbles when the data does not have full support. Since we search over many vectors without full support, we use the semidefinite programming formulation to solve the reduced problems and for the analysis throughout the rest of this paper.

Table 1: Comparison of solving the SOCP and SDP formulations of Alice forcing outcome 0 with 5-dimensional messages in four-rounds (averaged over $1,000$ randomly selected protocols).

INFO parameters	SOCP	SDP
Average running time (s)	0.1551	0.1529
Max running time (s)	0.7491	0.2394
Worst gap (s)	$+0.5098$	$+0.0927$
Average iteration	14.4420	12.2940
Average feasratio	0.9990	1.0000
Average timing	$[0.0270, 0.1267, 0.0010]^{\mathrm{T}}$	$[0.0024, 0.1494, 0.0009]^{\mathrm{T}}$
Average cpusec	0.9283	0.6588

Table 2: Comparison of solving the SOCP and SDP formulations of Alice forcing outcome 0 with 5-dimensional messages in four-rounds (averaged over $1,000$ randomly selected protocols with forced 0 entries).

INFO parameters	SOCP	SDP
Average running time (s)	0.4104	0.1507
Max running time (s)	0.7812	0.2084
Worst gap (s)	$+0.6323$	$+0$
Average iterations	32.7370	12.2530
Average feasratio	0.5172	1.0000
Average timing	$[0.0279, 0.3814, 0.0010]^{\mathrm{T}}$	$[0.0023, 0.1473, 0.0009]^{\mathrm{T}}$
Average cpusec	2.4953	0.5605

5 Protocol filter

In this section, we describe ways to bound the optimal cheating probabilities from below by finding feasible solutions to Alice and Bob's reduced cheating problems. In the search for parameters that lead to the lowest bias, our algorithm tests many protocols. The idea is to devise simple tests to check whether a protocol is a good candidate for being optimal. For example, suppose we can quickly compute the success probability of a certain cheating strategy for Bob. If this strategy succeeds with too high a probability for a given set of parameters, then we can rule out these parameters as being good choices. This saves the time it would have taken to solve the SDPs (or SOCPs).

We illustrate this idea using the Kitaev lower bound below.

Theorem 5.1 ([Kit02, GW07]) *For any coin-flipping protocol, we have*

$$P_{A,0}^* P_{B,0}^* \geq \frac{1}{2} \quad and \quad P_{A,1}^* P_{B,1}^* \geq \frac{1}{2}.$$

Suppose that we find that $P_{A,0}^* \approx 1/2$, that is, the protocol is very secure against dishonest Alice cheating towards 0. Then, from the Kitaev bound, we infer that $P_{B,0} \approx 1$ and the protocol is highly insecure against cheating Bob. Therefore, we can avoid solving for $P_{B,0}^*$.

The remainder of this section is divided according to the party that is dishonest. We discuss cheating strategies for the two parties for the special cases of 4-round and 6-round protocols.

Cheating Alice

We now present a theorem which captures some of Alice's cheating strategies.

Theorem 5.2 *For a protocol parameterized by $\alpha_0, \alpha_1 \in \mathrm{Prob}^A$ and $\beta_0, \beta_1 \in \mathrm{Prob}^B$, we can bound Alice's optimal cheating probability as follows:*

$$P_{A,0}^* \geq \frac{1}{2} \sum_{y \in B} \mathrm{conc} \left\{ \beta_{a,y} \mathrm{F}(\cdot, \alpha_a) : a \in \{0,1\} \right\} (v) \tag{2}$$

$$\geq \frac{1}{2} \lambda_{\max} \left(\eta \sqrt{\alpha_0} \sqrt{\alpha_0}^{\mathrm{T}} + \tau \sqrt{\alpha_1} \sqrt{\alpha_1}^{\mathrm{T}} \right) \tag{3}$$

$$\geq \left(\frac{1}{2} + \frac{1}{2} \sqrt{\mathrm{F}(\alpha_0, \alpha_1)} \right) \left(\frac{1}{2} + \frac{1}{2} \Delta(\beta_0, \beta_1) \right), \tag{4}$$

where

$$\eta := \sum_{\substack{y \in B: \\ \beta_{0,y} \geq \beta_{1,y}}} \beta_{0,y} \quad and \quad \tau := \sum_{\substack{y \in B: \\ \beta_{0,y} < \beta_{1,y}}} \beta_{1,y} ,$$

and $\sqrt{v}$ is the normalized principal eigenvector of $\eta \sqrt{\alpha_0} \sqrt{\alpha_0}^{\mathrm{T}} + \tau \sqrt{\alpha_1} \sqrt{\alpha_1}^{\mathrm{T}}$.

Furthermore, in a six-round protocol, we have

$$P_{A,0}^* \geq \frac{1}{2}\lambda_{\max}\left(\eta'\sqrt{\mathrm{Tr}_{A_2}(\alpha_0)}\sqrt{\mathrm{Tr}_{A_2}(\alpha_0)}^{\mathrm{T}} + \tau'\sqrt{\mathrm{Tr}_{A_2}(\alpha_1)}\sqrt{\mathrm{Tr}_{A_2}(\alpha_1)}^{\mathrm{T}}\right) \quad (5)$$

$$\geq \left(\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\mathrm{Tr}_{A_2}(\alpha_0), \mathrm{Tr}_{A_2}(\alpha_1))}\right)\left(\frac{1}{2} + \frac{1}{2}\Delta(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1))\right), \quad (6)$$

where

$$\eta' := \sum_{\substack{y_1 \in B_1: \\ [\mathrm{Tr}_{B_2}(\beta_0)]_{y_1} \geq [\mathrm{Tr}_{B_2}(\beta_1)]_{y_1}}} [\mathrm{Tr}_{B_2}(\beta_0)]_{y_1} \quad \text{and} \quad \tau' := \sum_{\substack{y_1 \in B_1: \\ [\mathrm{Tr}_{B_2}(\beta_0)]_{y_1} < [\mathrm{Tr}_{B_2}(\beta_1)]_{y_1}}} [\mathrm{Tr}_{B_2}(\beta_1)]_{y_1} \ .$$

We have analogous bounds for $P_{A,1}^*$, which are obtained by interchanging β_0 and β_1 in the above expressions.

We call (2) Alice's *improved eigenstrategy*, (3) her *eigenstrategy*, and (4) her *three-round strategy*. For six-round protocols, we call (5) Alice's *eigenstrategy* and (6) her *measuring strategy*.

Note that only the improved eigenstrategy is affected by switching β_0 and β_1 (as long as we are willing to accept a slight modification to how we break ties in the definitions of η, η', τ, and τ').

We now briefly describe the strategies that yield the corresponding cheating probabilities in Theorem 5.2. Her three-round strategy is to prepare the qubits AA' in the state $\psi' = (\psi_0 + \psi_1)/\|\psi_0 + \psi_1\|$ instead of ψ_0 or ψ_1, send the first n messages accordingly, then measure the qubits received from Bob to try to learn b, and reply with a bit a using the measurement outcome (along with the rest of the state ψ'), to bias the coin towards her desired output. Her eigenstrategy is the same as her three-round strategy, except that the first message is further optimized. The improved eigenstrategy has the same first message as in her eigenstrategy, but the last message is further optimized.

For a six-round protocol, Alice's measuring strategy is to prepare the qubits AA' in the following state $\psi' = (\psi_0' + \psi_1')/\|\psi_0' + \psi_1'\|$ where ψ_0' and ψ_1' are purifications of $\mathrm{Tr}_{A_2,A'}(\psi_0\psi_0^*)$ and $\mathrm{Tr}_{A_2,A'}(\psi_1\psi_1^*)$, respectively. She measures Bob's first message to try to learn b, then depending on the outcome, she applies a (fidelity achieving) unitary before sending the rest of her messages. Her six-round eigenstrategy is similar to her measuring strategy, except her first message is optimized in a way described in the proof.

We prove Theorem 5.2 in the appendix.

Cheating Bob

We turn to strategies for a dishonest Bob.

35

Theorem 5.3 *For a protocol parameterized by $\alpha_0, \alpha_1 \in \text{Prob}^A$ and $\beta_0, \beta_1 \in \text{Prob}^B$, we can bound Bob's optimal cheating probability as follows:*

$$P^*_{B,0} \geq \frac{1}{2} + \frac{1}{2}\sqrt{F(\beta_0, \beta_1)}, \tag{7}$$

and

$$P^*_{B,0} \geq \frac{1}{2} + \frac{1}{2}\Delta(\text{Tr}_{A_2 \times \cdots \times A_n}(\alpha_0), \text{Tr}_{A_2 \times \cdots \times A_n}(\alpha_1)). \tag{8}$$

In a four-round protocol, we have

$$P^*_{B,0} \geq \frac{1}{2} \sum_{a \in \{0,1\}} F\left(\sum_{x \in A} \alpha_{a,x} v_x, \beta_a\right) \tag{9}$$

$$\geq \frac{1}{2} \sum_{x \in A} \lambda_{\max}\left(\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a}\sqrt{\beta_a}^T\right) \tag{10}$$

$$\geq \max\left\{\frac{1}{2} + \frac{1}{2}\Delta(\alpha_0, \alpha_1), \; \frac{1}{2} + \frac{1}{2}\sqrt{F(\beta_0, \beta_1)}\right\} ,$$

where $\sqrt{v_x}$ is the normalized principal eigenvector of $\sum_{a \in \{0,1\}} \alpha_{a,x}\sqrt{\beta_a}\sqrt{\beta_a}^T$.
In a six-round protocol, we have

$$P^*_{B,0} \geq \frac{1}{2} \sum_{a \in A'_0} F\left(\sum_{x \in A} \alpha_{a,x} \tilde{p}_2^{(x)}, \beta_a\right) \tag{11}$$

$$\geq \frac{1}{2} \lambda_{\max}\left(\kappa \sqrt{\text{Tr}_{B_2}(\beta_0)}\sqrt{\text{Tr}_{B_2}(\beta_0)}^T + \zeta \sqrt{\text{Tr}_{B_2}(\beta_1)}\sqrt{\text{Tr}_{B_2}(\beta_1)}^T\right) \tag{12}$$

$$\geq \left(\frac{1}{2} + \frac{1}{2}\sqrt{F(\text{Tr}_{B_2}(\beta_0), \text{Tr}_{B_2}(\beta_1))}\right)\left(\frac{1}{2} + \frac{1}{2}\Delta(\alpha_0, \alpha_1)\right), \tag{13}$$

where

$$[\tilde{p}_2^{(x)}]_{y_1, y_2} := \begin{cases} c_{y_1} \dfrac{\beta_{g(x), y_1, y_2}}{[\text{Tr}_{B_2}(\beta_{g(x)})]_{y_1}} & \text{if } [\text{Tr}_{B_2}(\beta_{g(x)})]_{y_1} > 0 , \\[4mm] c_{y_1} \dfrac{1}{|B_2|} & \text{if } [\text{Tr}_{B_2}(\beta_{g(x)})]_{y_1} = 0 , \end{cases}$$

$$\kappa = \sum_{\substack{x \in A: \\ \alpha_{0,x} \geq \alpha_{1,x}}} \alpha_{0,x} , \qquad \zeta = \sum_{\substack{x \in A: \\ \alpha_{0,x} < \alpha_{1,x}}} \alpha_{1,x} , \qquad g(x) = \arg\max_a \{\alpha_{a,x}\} ,$$

and $\sqrt{c}$ is the normalized principal eigenvector of

$$\frac{1}{2} \lambda_{\max}\left(\kappa \sqrt{\text{Tr}_{B_2}(\beta_0)}\sqrt{\text{Tr}_{B_2}(\beta_0)}^T + \zeta \sqrt{\text{Tr}_{B_2}(\beta_1)}\sqrt{\text{Tr}_{B_2}(\beta_1)}^T\right) .$$

Furthermore, if $|A_i| = |B_i|$ for all $i \in \{1, \ldots, n\}$, then

$$P_{B,0}^* \geq \frac{1}{2} \sum_{a \in \{0,1\}} \mathrm{F}(\alpha_a, \beta_a) \ . \tag{14}$$

We get analogous lower bounds for $P_{B,1}^*$ by switching the roles of β_0 and β_1 in the above expressions.

We prove Theorem 5.3 in the appendix. We call (7) Bob's *ignoring strategy* and (8) his *measuring strategy*. For four-round protocols, we call (9) Bob's *eigenstrategy* and (10) his *eigenstrategy lower bound*. For six-round protocols, we call (11) Bob's *six-round eigenstrategy*, (12) his *eigenstrategy lower bound*, and (13) his *three-round strategy*. We call (14) Bob's *returning strategy*.

Note that the only strategies that are affected by switching β_0 and β_1 are the eigenstrategy and the returning strategy.

We now briefly describe the strategies that yield the corresponding cheating probabilities in Theorem 5.3. Bob's ignoring strategy is to prepare the qubits BB' in the state $\phi' = (\phi_0 + \phi_1)/\|\phi_0 + \phi_1\|$ instead of ϕ_0 or ϕ_1, send the first n messages accordingly, then send a value for b that favours his desired outcome (along with the rest of ϕ'). His measuring strategy is to measure Alice's first message, choose b according to his best guess for a and run the protocol with ϕ_b. His returning strategy is to send Alice's messages right back to her. For the four-round eigenstrategy, Bob's commitment state is a principal eigenvector depending on Alice's first message.

For a six-round protocol, Bob's three-round strategy is to prepare the qubits BB' in the following state $\phi' = (\phi_0' + \phi_1')/\|\phi_0' + \phi_1'\|$ where ϕ_0' and ϕ_1' are purifications of $\mathrm{Tr}_{B_2, B'}(\phi_0 \phi_0^*)$ and $\mathrm{Tr}_{B_2, B'}(\phi_1 \phi_1^*)$, respectively. He measures Alice's second message to try to learn a, then depending on the outcome, he applies a (fidelity achieving) unitary before sending the rest of his messages. His six-round eigenstrategy is similar to his three-round strategy except that the first message is optimized in a way described in the proof.

6 Protocol symmetry

In this section, we discuss equivalence between protocols due to symmetry in the states used in them. Namely, we identify transformations on states under which the bias remains unchanged. This allows us to prune the search space of parameters needed to specify a protocol in the family under scrutiny. As a result, we significantly reduce the time required for our searches.

6.1 Index symmetry

We show that if we permute the elements of A_i or B_i, for any $i \in \{1, \ldots, n\}$, then this does not change the bias of the protocol. We first show that cheating Bob is unaffected.

Cheating Bob

Bob's reduced problems are to maximize $\frac{1}{2}\sum_{a\in A_0'}\mathrm{F}\big((\alpha_a\otimes\mathrm{I}_B)^{\mathrm{T}}p_n,\,\beta_a\big)$, for forcing outcome 0, and $\frac{1}{2}\sum_{a\in A_0'}\mathrm{F}\big((\alpha_a\otimes\mathrm{I}_B)^{\mathrm{T}}p_n,\,\beta_{\bar{a}}\big)$, for forcing outcome 1, over the polytope $\mathcal{P}_\mathrm{B}$ defined as the set of all vectors $(p_1,p_2,\dots,p_n)$ that satisfy

$$
\begin{aligned}
\mathrm{Tr}_{B_1}(p_1) &= e_{A_1}, \\
\mathrm{Tr}_{B_2}(p_2) &= p_1\otimes e_{A_2}, \\
&\;\;\vdots \\
\mathrm{Tr}_{B_n}(p_n) &= p_{n-1}\otimes e_{A_n}, \\
p_j &\in \mathbb{R}_+^{A_1\times B_1\times\cdots\times A_j\times B_j}, \text{ for all } j\in\{1,\dots,n\}\ .
\end{aligned}
$$

Suppose we are given a new protocol where the elements of A_i have been permuted, for some $i\in\{1,\dots,n\}$ (and therefore the entries of α_a for both $a\in\{0,1\}$). We can write the entries of $(\alpha_a\otimes\mathrm{I}_B)^{\mathrm{T}}p_n$ as

$$
[(\alpha_a\otimes\mathrm{I}_B)^{\mathrm{T}}p_n]_y = \sum_{x\in A}\alpha_{a,x}[p_n]_{x,y},
$$

for each $y\in B$. For any feasible solution for the original protocol, we construct a feasible solution by permuting the elements of p_j corresponding to A_i. This gives us a bijection, and the feasible solution so constructed has the same objective function value as the original one. Thus, dishonest Bob cannot cheat more or less than in the original protocol.

Now suppose we are given a new protocol where the elements of B_i have been permuted for some $i\in\{1,\dots,n\}$. We can write

$$
\mathrm{F}\big((\alpha_a\otimes\mathrm{I}_B)^{\mathrm{T}}p_n,\,\beta_a\big) = \left(\sqrt{\overline{(\alpha_a\otimes\mathrm{I}_B)^{\mathrm{T}}p_n}}^{\,\mathrm{T}}\sqrt{\beta_a}\right)^2.
$$

If we permute the entries in p_n corresponding to B_i (and likewise for every variable in the polytope) we get the same objective function value.

Similar arguments hold for $P_{\mathrm{B},1}^*$. In both cases, Bob's cheating probabilities are unaffected.

Cheating Alice

To show that the bias of the protocol remains unchanged, we still need to check that cheating Alice is unaffected by a permutation of the elements of A_i or B_i. Alice's reduced problem is to maximize $\frac{1}{2}\sum_{a\in A_0'}\sum_{y\in B}\beta_{a,y}\ \mathrm{F}(s^{(a,y)},\alpha_a)$ for forcing outcome 0, and $\frac{1}{2}\sum_{a\in A_0'}\sum_{y\in B}\beta_{\bar{a},y}\ \mathrm{F}(s^{(a,y)},\alpha_a)$ for forcing outcome 1, over the set of all vectors

$(s_1, s_2, \ldots, s_n, s)$ that satisfy

$$
\begin{aligned}
\mathrm{Tr}_{A_1}(s_1) &= 1 , \\
\mathrm{Tr}_{A_2}(s_2) &= s_1 \otimes e_{B_1} , \\
&\vdots \\
\mathrm{Tr}_{A_n}(s_n) &= s_{n-1} \otimes e_{B_{n-1}} , \\
\mathrm{Tr}_{A_0'}(s) &= s_n \otimes e_{B_n} , \\
s_j &\in \mathbb{R}_+^{A_1 \times B_1 \times \cdots \times B_{j-1} \times A_j} , \quad \text{for all } j \in \{1, \ldots, n\} , \\
s &\in \mathbb{R}_+^{A_0' \times A \times B} .
\end{aligned}
$$

By examining the above problem, we see that the same arguments that apply to cheating Bob also apply to cheating Alice. We can simply permute any feasible solution to account for any permutation in A_i or B_i.

Note that these arguments only hold for "local" permutations, i.e., we cannot in general permute the indices in $A_i \times A_{i'}$ without affecting the bias.

6.2 Symmetry between probability distributions

We now identify a different kind of symmetry in the protocols. Recall the four objective functions

$$
P_{\mathrm{B},0}^* = \frac{1}{2} \sum_{a \in A_0'} \mathrm{F}\big((\alpha_a \otimes \mathrm{I}_B)^{\mathrm{T}} p_n, \beta_a\big) \quad \text{and} \quad P_{\mathrm{B},1}^* = \frac{1}{2} \sum_{a \in A_0'} \mathrm{F}\big((\alpha_a \otimes \mathrm{I}_B)^{\mathrm{T}} p_n, \beta_{\bar{a}}\big)
$$

for Bob and

$$
P_{\mathrm{A},0}^* = \frac{1}{2} \sum_{y \in B} \sum_{a \in \{0,1\}} \beta_{a,y} \, \mathrm{F}(s^{(a,y)}, \alpha_a) \quad \text{and} \quad P_{\mathrm{A},1}^* = \frac{1}{2} \sum_{y \in B} \sum_{a \in \{0,1\}} \beta_{\bar{a},y} \, \mathrm{F}(s^{(a,y)}, \alpha_a)
$$

for Alice.

We argue that the four quantities above are not affected if we switch β_0 and β_1 and simultaneously switch α_0 and α_1. This is immediate for cheating Bob, but requires explanation for cheating Alice. The only constraints involving $s^{(a,y)}$ can be written as

$$
\sum_{a \in A_0'} s^{(a,y)} = s_n^{(y_1, \ldots, y_{n-1})},
$$

for all $y = (y_1, \ldots, y_{n-1}, y_n) \in B$. Since this constraint is symmetric about a, the result follows.

It is also evident that switching β_0 and β_1 switches $P_{\mathrm{A},0}^*$ and $P_{\mathrm{A},1}^*$ and it also switches $P_{\mathrm{B},0}^*$ and $P_{\mathrm{B},1}^*$. With these symmetries, we can effectively switch the roles of α_0 and α_1 and the roles of β_0 and β_1 independently and the bias is unaffected.

6.3 The use of symmetry in the search algorithm

Since we are able to switch the roles of α_0 and α_1, we assume α_0 has the largest entry out of α_0 and α_1 and similarly that β_0 has the largest entry out of β_0 and β_1.

In four-round protocols, since we can permute the elements of $A = A_1$, we also assume α_0 has entries that are non-decreasing. This allows us to upper bound all the entries of α_0 and α_1 by the last entry in α_0. We do this simultaneously for β_0 and β_1.

In the six-round version, we need to be careful when applying the index symmetry, we cannot permute all of the entries in α_0. The index symmetry only applies to local permutations so we only partially order them. We order A_2 such that the entries $\alpha_{0,\tilde{x}_1 x_2}$ do not decrease for *one particular index* $\tilde{x}_1 \in A_1$. It is convenient to choose the index corresponding to the largest entry. Then we order the last block of entries in α_0 such that they do not decrease. Note that the last entry in α_0 is now the largest among all the entries in α_0 and α_1. We do this simultaneously for β_0 and β_1. Note that the search algorithm does not stop all symmetry; for example if α_0 and α_1 both have an entry of largest magnitude, we do not compare the second largest entries. But, as will be shown in the computational tests, we have a dramatic reduction in the number of protocols to be tested using the symmetry in the way described above.

7 Search algorithm

In this section, we develop an algorithm for finding coin-flipping protocols with small bias within our parametrized family.

To search for protocols, we first fix a dimension d for the parameters

$$\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathrm{Prob}^d.$$

We then create a finite mesh over these parameters by creating a mesh over the entries in the probability vectors $\alpha_0, \alpha_1, \beta_0,$ and β_1. We do so by increments of a precision parameter $\nu \in (0,1)$. For example, we range over the values

$$\{0, \nu, 2\nu, \ldots, 1 - \nu, 1\}$$

for $[\alpha_0]_0$, the first entry of α_0. For the second entry of α_0, we range over

$$\{0, \nu, 2\nu, \ldots, 1 - [\alpha_0]_0\}$$

and so forth. Note that we only consider $\nu = 1/N$ for some positive integer N so that we use the endpoints of the intervals.

This choice in creating the mesh makes it very easy to exploit the symmetry discussed in Section 6. We show computationally (in Section 8) that this symmetry helps by dramatically reducing the number of protocols to be tested. This is important since there are $\binom{d+N-1}{N}^4$ protocols to test (before applying symmetry considerations).

Each point in this mesh is a set of candidate parameters for an optimal protocol. As described in Section 5, the protocol filter can be used to expedite the process of checking whether the protocol has high bias or is a good candidate for an optimal protocol. There are two things to be considered at this point which we now address.

First, we have to determine the order in which the cheating strategies in the protocol filter are applied. It is roughly the case that the computationally cheaper tasks give a looser lower bound to the optimal cheating probabilities. Therefore, we start with these easily computable probabilities, i.e. the probabilities involving norms and fidelities, then check the more computationally expensive tasks such as largest eigenvalues and calculating principal eigenvectors. We lastly solve the semidefinite programs. Another heuristic that we use is alternating between Alice and Bob's strategies. Many protocols with high bias seem to prefer either cheating Alice or cheating Bob. Having cheating strategies for both Alice and Bob early in the filter removes the possibility of checking many of Bob's strategies when it is clearly insecure concerning cheating Alice and vice versa. Starting with these heuristics, we then ran preliminary tests to see which order seemed to perform the best. The order (as well as the running times for the filter strategies) is shown in Tables 3 and 4 for the four-round version and Tables 13 and 14 for the six-round version.

Second, we need to determine a threshold for what constitutes a "high bias." If a filter strategy has success probability 0.9, do we eliminate this candidate protocol? The lower the threshold, the more quickly the filter eliminates protocols. However, if the threshold is too low, we may be too ambitious and not find any protocols. To determine a good threshold, consider the following protocol parameters

$$\alpha_0 = \frac{1}{2}\left[1, 0, 1\right]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{2}\left[0, 1, 1\right]^{\mathrm{T}}, \quad \beta_0 = \left[1, 0\right]^{\mathrm{T}}, \quad \beta_1 = \left[0, 1\right]^{\mathrm{T}}.$$

This is the four-round version of the optimal three-round protocol in Subsection 3.2. Numerically solving for the cheating probabilities for this protocol shows that

$$P_{\mathrm{A},0}^* = P_{\mathrm{A},1}^* = P_{\mathrm{B},0}^* = P_{\mathrm{B},1}^* = 3/4.$$

Thus, there exists a protocol with the same bias as the best-known explicit coin-flipping protocol constructions. This suggests that we use a threshold around 0.75. Preliminary tests show that using a threshold of 0.75 or larger is much slower than a value of 0.7499. This is because using the larger threshold allows protocols with optimal cheating probabilities (or filter cheating probabilities) of 0.75 to slip through the filter and these protocols are no better than the one mentioned above (and many are just higher dimensional embeddings of it). Therefore, we use a threshold of 0.7499. (Tests using a threshold of slightly larger than 0.75 are considered in Subsection 8.6.)

Using these ideas, we now state the search algorithm.

> Search algorithm for finding the best protocol parameters
>
> Fix a dimension d and mesh precision ν.
> For each protocol in the mesh (modulo the symmetry):
> • Use the Protocol Filter to eliminate (some) protocols with bias above 0.2499.
> • Calculate the optimal cheating probabilities by solving the SDPs.
>   If any are larger than 0.7499, move on to the next protocol.
>   Else, output the protocol parameters with bias $\epsilon < 1/4$.

We test the algorithm on the cases of four and six-round protocols and for certain dimensions and precisions for the mesh. These are presented in detail next.

8   Numerical results

Computational Platform.   We ran our programs on Matlab, Version 7.12.0.635, on an SGI XE C1103 with 2x 3.2 GHz 4-core Intel X5672 x86 CPUs processor, and 10 GB memory, running Linux.

We solved the semidefinite programs using SeDuMi 1.3, a program for solving semidefinite programs in Matlab [Stu99, Stu02].

Sample programs can be found at the following link:

`http://www.math.uwaterloo.ca/~anayak/coin-search/`

8.1   Four-round search

We list the filter cheating strategies in Tables 3 and 4 which also give an estimate of how long it takes the program to compute the success probability for each strategy based on the average over 1000 random instances (i.e. four randomly chosen probability vectors α_0, α_1, β_0, and β_1.)

Notice the two strategies with codes F1 and F2 are special because they only involve two of the four probability distributions. Preliminary tests show that first generating β_0 and β_1 and checking with F1 is much faster than first generating α_0 and α_1 and checking with F2, even though F2 is much faster to compute.

We can similarly justify the placement of $P^*_{A,0}$ before $P^*_{B,0}$ or $P^*_{B,1}$. The strategies F8 and F9 perform very well and the cheating probabilities are empirically very close to $P^*_{B,0}$ and $P^*_{B,1}$. Thus, if a protocol gets through the F8 and F9 filter strategies, then it is likely that $P^*_{B,0}$ and $P^*_{B,1}$ are also less than 0.7499. This is why we place $P^*_{A,0}$ first (although it will be shown that the order of solving the SDPs does not matter much).

Table 3: Average running times for filter strategies for a 4-round protocol when $d = 5$ over random protocol states (1 of 2).

Success Probability	Comp. Time (s)	Code
$\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\beta_0, \beta_1)}$	0.000034429	F1
$\frac{1}{2} + \frac{1}{2}\Delta(\alpha_0, \alpha_1)$	0.000004640	F2
$\left(\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\alpha_0, \alpha_1)}\right)\left(\frac{1}{2} + \frac{1}{2}\Delta(\beta_0, \beta_1)\right)$	0.000025980	F3
$\frac{1}{2}\sum_{a\in\{0,1\}}\mathrm{F}(\alpha_a, \beta_a)$	0.000023767	F4
$\frac{1}{2}\sum_{a\in\{0,1\}}\mathrm{F}(\alpha_a, \beta_{\bar{a}})$	0.000018019	F5
$\frac{1}{2}\lambda_{\max}\left(\left(\displaystyle\sum_{y:\beta_{0,y}\geq\beta_{1,y}}\beta_{0,y}\right)\sqrt{\alpha_0}\sqrt{\alpha_0}^{\mathrm{T}} + \left(\displaystyle\sum_{y:\beta_{0,y}<\beta_{1,y}}\beta_{1,y}\right)\sqrt{\alpha_1}\sqrt{\alpha_1}^{\mathrm{T}}\right)$ ($\sqrt{v}$ is the principal eigenvector)	0.000036613	F6
$\frac{1}{2}\sum_{x\in A}\lambda_{\max}\left(\sum_{a\in\{0,1\}}\alpha_{a,x}\sqrt{\beta_a}\sqrt{\beta_a}^{\mathrm{T}}\right)$ ($\sqrt{v_x}$ is the principal eigenvector for each $x \in A$)	0.000073010	F7
$\frac{1}{2}\sum_{a\in\{0,1\}}\mathrm{F}\left(\sum_{x\in A}\alpha_{a,x}(v_x), \beta_a\right)$	0.000697611	F8
$\frac{1}{2}\sum_{a\in\{0,1\}}\mathrm{F}\left(\sum_{x\in A}\alpha_{a,x}(v_x), \beta_{\bar{a}}\right)$	0.000532954	F9

43

Table 4: Average running times for filter strategies for a 4-round protocol when $d = 5$ over random protocol states (2 of 2).

Success Probability	Comp. Time (s)	Code
$\sum_{y \in B} \text{conc}\left\{ \frac{1}{2}\beta_{0,y}\text{F}(\cdot, \alpha_0), \frac{1}{2}\beta_{1,y}\text{F}(\cdot, \alpha_1) \right\}(v)$	0.122971205	F10
$\sum_{y \in B} \text{conc}\left\{ \frac{1}{2}\beta_{1,y}\text{F}(\cdot, \alpha_0), \frac{1}{2}\beta_{0,y}\text{F}(\cdot, \alpha_1) \right\}(v)$	0.123375678	F11
$P_{A,0}^*$	0.149814373	SDPA0
$\dfrac{1}{2P_{A,0}^*}$	0.000000947	F12
$P_{B,0}^*$	0.070846378	SDPB0
$P_{A,1}^*$	0.149176117	SDPA1
$\dfrac{1}{2P_{A,1}^*}$	0.000000760	F13
$P_{B,1}^*$	0.070479449	SDPB1

Recall from Subsection 4.3 that we solve for $P_{B,0}^*$, $P_{B,1}^*$, $P_{A,0}^*$, and $P_{A,1}$ using the semidefinite programming formulations of the reduced problems.

We then give tables detailing how well the filter performs for four-round protocols, by counting the number of protocols that are *not* determined to have bias greater than 0.2499 by each prefix of cheating strategies. We test four-round protocols with message dimension $d \in \{2, \ldots, 9\}$ and precision ν ranging up to $1/2000$ (depending on d).

Table 5: The number of protocols that get past each strategy in the filter for $d = 2$.

$d = 2$	$\nu = 1/500$	$\nu = 1/1000$	$\nu = 1/1250$	$\nu = 1/1500$	$\nu = 1/2000$
Protocols	6.30 e+10	1.00 e+12	2.44 e+12	5.07 e+12	1.60 e+13
Symmetry	3,969,126,001	63,001,502,001	153,566,799,376	318,097,128,001	1,004,006,004,001
F1	96,706,535	1,499,479,974	3,636,609,280	7,506,289,309	23,607,143,560
F2	72,336,875	1,123,112,000	2,724,552,320	5,624,716,125	17,693,560,000
F3	5	27	50	67	124
F4	0	0	0	0	0
F5	0	0	0	0	0
F6	0	0	0	0	0
F7	0	0	0	0	0
F8	0	0	0	0	0

Table 6: The number of protocols that get past each strategy in the filter for $d = 3$.

$d = 3$	$\nu = 1/5$	$\nu = 1/10$	$\nu = 1/20$	$\nu = 1/30$	$\nu = 1/50$
Protocols	1.94 e+05	1.89 e+07	2.84 e+09	6.05 e+10	3.09 e+12
Symmetry	4,356	272,484	29,430,625	55,436,7025	25,475,990,544
F1	1,254	37,584	2,175,425	30,985,220	1,020,080,292
F2	665	19,656	1,300,042	19,366,256	662,158,728
F3	49	470	22,282	225,098	4,414,994
F4	29	261	11,667	110,931	2,028,518
F5	28	258	11,495	109,515	2,009,141
F6	28	241	10,405	96,464	1,765,114
F7	0	3	54	148	1,158
F8	0	0	0	0	0

Table 7: The number of protocols that get past each strategy in the filter for $d = 4$.

$d = 4$	$\nu = 1/10$	$\nu = 1/12$	$\nu = 1/16$	$\nu = 1/20$	$\nu = 1/24$	$\nu = 1/30$
Protocols	6.69 e+09	4.28 e+10	8.81 e+11	9.83 e+12	7.31 e+13	8.86 e+14
Symmetry	13,498,276	74,166,544	1,154,640,400	10,334,552,281	69,927,455,844	736,486,643,344
F1	2,432,188	12,616,580	146,114,000	934,856,164	5,916,006,936	49,798,933,264
F2	1,036,030	5,616,810	71,246,700	489,282,376	3,170,626,956	27,760,130,976
F3	66,623	302,547	3,185,895	19,670,642	101,703,667	738,284,522
F4	46,734	209,747	2,061,868	12,000,187	59,503,895	406,963,112
F5	46,531	208,961	2,054,891	11,962,104	59,353,374	406,099,637
F6	42,591	198,192	1,886,782	11,004,125	54,702,075	367,847,304
F7	329	756	3,439	17,144	55,929	190,699
F8	0	0	0	0	0	0

Table 8: The number of protocols that get past each strategy in the filter for $d = 5$.

$d = 5$	$\nu = 1/5$	$\nu = 1/8$	$\nu = 1/10$	$\nu = 1/12$
Protocols	2.52 e+08	6.00 e+10	1.00 e+12	1.09 e+13
Symmetry	240,100	29,539,225	284,529,424	2,485,919,881
F1	105,840	9,467,770	66,257,504	567,544,997
F2	37,584	2,687,906	22,774,544	203,983,360
F3	8,561	241,420	2,440,765	17,794,655
F4	7,423	201,569	1,937,298	13,682,059
F5	7,417	200,965	1,933,833	13,665,087
F6	7,417	189,144	1,790,144	13,117,165
F7	0	1,415	10,790	43,459
F8	0	0	0	0

Table 9: The number of protocols that get past each strategy in the filter for $d = 6$.

$d = 6$	$\nu = 1/7$	$\nu = 1/8$	$\nu = 1/9$	$\nu = 1/10$	$\nu = 1/11$	$\nu = 1/12$
Protocols	3.93 e+11	2.74 e+12	1.60 e+13	8.13 e+13	3.64 e+14	1.46 e+15
Symmetry	53,144,100	265,950,864	1,021,825,156	3,534,302,500	12,577,398,201	46,107,255,076
F1	25,070,310	107,583,876	387,459,886	1,034,786,700	3,605,814,648	13,370,558,568
F2	7,276,924	23,294,007	123,246,328	287,251,218	1,330,224,696	3,841,063,848
F3	1,744,038	2,811,374	25,114,451	42,503,208	258,455,916	468,218,324
F4	1,551,522	2,526,900	21,682,087	36,628,517	214,823,642	390,846,158
F5	1,550,617	2,524,052	21,666,437	36,594,682	214,698,072	390,649,931
F6	1,451,038	2,419,474	20,598,749	34,117,986	203,605,433	377,899,946
F7	9,169	13,976	57,720	174,118	526,077	1,153,864
F8	0	0	0	0	0	0

Table 10: The number of protocols that get past each strategy in the filter for $d = 7$.

$d = 7$	$\nu = 1/5$	$\nu = 1/6$	$\nu = 1/7$	$\nu = 1/8$	$\nu = 1/9$	$\nu = 1/10$
Protocols	4.55 e+10	7.28 e+11	8.67 e+12	8.13 e+13	6.27 e+14	4.11 e+15
Symmetry	3,709,476	46,963,609	289,374,121	1,730,643,201	7,402,021,225	30,490,398,225
F1	2,270,754	26,952,849	161,111,181	841,297,023	3,456,456,125	10,915,707,495
F2	495,180	3,154,266	36,330,756	136,788,372	851,509,125	2,419,940,743
F3	149,806	369,434	10,277,699	20,469,535	216,148,269	449,464,967
F4	142,255	351,290	9,583,747	19,200,670	197,250,330	409,366,494
F5	142,241	351,219	9,582,215	19,194,692	197,214,454	409,185,885
F6	142,241	351,219	9,034,728	18,734,072	187,977,589	383,402,064
F7	0	0	60,155	91,787	512,171	1,804,382
F8	0	0	0	0	0	0

Table 11: The number of protocols that get past each strategy in the filter for $d = 8$.

$d = 8$	$\nu = 1/4$	$\nu = 1/5$	$\nu = 1/6$	$\nu = 1/7$	$\nu = 1/8$	$\nu = 1/9$
Protocols	1.18 e+10	3.93 e+11	8.67 e+12	1.38 e+14	1.71 e+15	1.71 e+16
Symmetry	1,572,516	11,532,816	179,345,664	1,293,697,024	9,018,161,296	42,352,405,209
F1	1,054,614	7,797,216	115,131,024	814,855,040	5,050,850,268	23,061,817,617
F2	60,552	1,356,936	9,766,192	142,862,430	606,597,735	4,417,668,742
F3	0	431,956	1,254,420	44,457,239	106,851,420	1,276,499,496
F4	0	417,759	1,213,728	42,541,702	102,719,851	1,204,238,273
F5	0	417,741	1,213,629	42,539,430	102,710,139	1,204,173,244
F6	0	417,741	1,213,629	40,425,272	101,061,706	1,151,097,965
F7	0	0	0	277,225	452,792	3,194,346
F8	0	0	0	0	0	0

Table 12: The number of protocols that get past each strategy in the filter for $d = 9$.

$d = 9$	$\nu = 1/3$	$\nu = 1/4$	$\nu = 1/5$	$\nu = 1/6$	$\nu = 1/7$	$\nu = 1/8$
Protocols	7.41 e+08	6.00 e+10	2.74 e+12	8.13 e+13	1.71 e+15	2.74 e+16
Symmetry	164,025	3,744,225	32,069,569	594,433,161	4,957,145,649	39,808,629,441
F1	131,625	2,666,430	23,348,549	414,160,047	3,423,681,189	24,851,338,155
F2	14,300	115,752	3,273,662	26,075,045	470,028,582	2,216,082,560
F3	2,700	0	1,065,271	3,484,092	153,932,946	432,754,976
F4	2,639	0	1,041,339	3,405,532	149,523,487	421,903,500
F5	2,639	0	1,041,317	3,405,403	149,520,361	421,889,260
F6	2,639	0	1,041,317	3,405,403	142,916,565	416,869,327
F7	0	0	0	0	1,053,222	1,809,800
F8	0	0	0	0	0	0

Observations on the four-round search

We were able to search larger spaces than feasible with the SDP formulations alone. For example, suppose we took the 2.74×10^{16} protocols from the $d = 9$, $\nu = 1/8$ search and checked to see if any of these had bias less than 0.7499 by solving only the reduced SDPs. Since each SDP takes at least 0.08 seconds to solve, this search would take at least 69 million years to finish. By applying the techniques in this paper, we were able to run this search in a matter of days.

We see that symmetry helped dramatically reduce the number of protocols that needed to be tested. In the largest search, we were able to cut the 2.74×10^{16} protocols down to 3.98×10^{10}. F1 and F2 perform very well, together cutting down the number of protocols by a factor of about 10. An interesting observation is that F2 performs much better than F1, and is also 10 times faster to compute. It may seem better to put F2 before F1 in the tests, however, we place F1 first since it is beneficial to have the more expensive strategy being computed first. This way, it only needs to be computed for every choice of β_0 and β_1. If we were to calculate F2 first, we would have to calculate F1 on $(\alpha_0, \alpha_0, \beta_0, \beta_1)$ for every α_0, α_1 that F2 did not filter out.

Being the first strategy to rely on all four probability distributions, F3 performs very well by reducing the number of protocols by another factor of 10. F4, F5, and F6 do not perform well (F5 being the same as F4 but with β_0 swapped with β_1); they cut down the number of protocols by a very small number. F7 and F8 perform so well that no SDPs were needed to be solved.

These numbers suggest a conjecture along the lines of

$$\min_{\alpha_0, \alpha_1, \beta_0, \beta_1 \in \mathrm{Prob}^9} \max \left\{ \mathrm{F1}, \ldots, \mathrm{F8} \right\} \geq 0.7499.$$

However, we shall soon see computational evidence in Subsection 8.6 showing that this may not be true if we replace 0.7499 with 0.75 and conduct zoning-in searches with much higher precision.

8.2   Six-round search

We list the filter cheating strategies in the tables on the next two pages and give an estimate for how long it takes to compute the success probability for each strategy by taking the average over 1000 random instances. We then give tables of how well the filter performs for six-round protocols with $d \in \{2, 3\}$ and ν as small as $1/15$ for $d = 2$ and $1/4$ for $d = 3$. The measure of performance of the filter that we use is as before. For each prefix of cheating strategies in the filter, we count the number of protocols in the mesh that are *not* determined to have bias greater than 0.2499 by that prefix.

Again, we choose which strategy to put first, G1 or G2. Preliminary tests show that placing G1 first results in a much faster search, similar to the four-round case. Even

Table 13: Average running times for filter strategies in a six-round protocol for $d = 3$ over random protocol states (1 of 2).

Success Probability	Comp. Time (s)	Code
$\frac{1}{2} + \frac{1}{2}\sqrt{F(\beta_0, \beta_1)}$	0.000036128	G1
$\frac{1}{2} + \frac{1}{2}\Delta(\text{Tr}_{A_2}(\alpha_0), \text{Tr}_{A_2}(\alpha_1))$	0.000005552	G2
$\frac{1}{2}\lambda_{\max}\left(\kappa\sqrt{\text{Tr}_{B_2}(\beta_0)}\sqrt{\text{Tr}_{B_2}(\beta_0)}^{\text{T}} + \zeta\sqrt{\text{Tr}_{B_2}(\beta_1)}\sqrt{\text{Tr}_{B_2}(\beta_1)}^{\text{T}}\right)$ where $\kappa := \sum_{x:\alpha_{0,x} \geq \alpha_{1,x}} \alpha_{0,x}$ and $\zeta := \sum_{x:\alpha_{0,x} < \alpha_{1,x}} \alpha_{1,x}$	0.000015667	G3
$\left(\frac{1}{2} + \frac{1}{2}\sqrt{F(\text{Tr}_{A_2}(\alpha_0), \text{Tr}_{A_2}(\alpha_1))}\right)\left(\frac{1}{2} + \frac{1}{2}\Delta(\text{Tr}_{B_2}(\beta_0), \text{Tr}_{B_2}(\beta_1))\right)$	0.000028408	G4
$\frac{1}{2}\lambda_{\max}\left(\left(\sum_{y:\beta_{0,y} \geq \beta_{1,y}} \beta_{0,y}\right)\sqrt{\alpha_0}\sqrt{\alpha_0}^{\text{T}} + \left(\sum_{y:\beta_{0,y} < \beta_{1,y}} \beta_{1,y}\right)\sqrt{\alpha_1}\sqrt{\alpha_1}^{\text{T}}\right)$	0.000052325	G5
$\frac{1}{2}\lambda_{\max}\left(\eta'\sqrt{\text{Tr}_{A_2}(\alpha_0)}\sqrt{\text{Tr}_{A_2}(\alpha_0)}^{\text{T}} + \tau'\sqrt{\text{Tr}_{A_2}(\alpha_1)}\sqrt{\text{Tr}_{A_2}(\alpha_1)}^{\text{T}}\right)$ where $\eta' := \sum_{y_1 \in B_1 : [\text{Tr}_{B_2}(\beta_0)]_{y_1} \geq [\text{Tr}_{B_2}(\beta_1)]_{y_1}} [\text{Tr}_{B_2}(\beta_0)]_{y_1}$, and $\tau' := \sum_{y_1 \in B_1 : [\text{Tr}_{B_2}(\beta_0)]_{y_1} < [\text{Tr}_{B_2}(\beta_1)]_{y_1}} [\text{Tr}_{B_2}(\beta_1)]_{y_1}$	0.000044243	G6

Table 14: Average running times for filter strategies in a six-round protocol for $d = 3$ over random protocol states (2 of 2).

Success Probability	Comp. Time (s)	Code
$\frac{1}{2}\sum_{a\in A_0} F\left(\sum_{x\in A}\alpha_{a,x}\tilde{p}_2^{(x)}, \beta_a\right)$	0.000879119	G7
$\frac{1}{2}\sum_{a\in A_0} F\left(\sum_{x\in A}\alpha_{a,x}\tilde{p}_2^{(x)}, \beta_{\bar{a}}\right)$ where $\tilde{p}_2^{(x)}$ is as defined in Theorem 5.3	0.000797106	G8
$\sum_{y\in B}\text{conc}\left\{\frac{1}{2}\beta_{0,y}F(\cdot,\alpha_0), \frac{1}{2}\beta_{1,y}F(\cdot,\alpha_1)\right\}(v)$	0.256377981	G9
$\sum_{y\in B}\text{conc}\left\{\frac{1}{2}\beta_{1,y}F(\cdot,\alpha_0), \frac{1}{2}\beta_{0,y}F(\cdot,\alpha_1)\right\}(v)$ ($\sqrt{v}$ is the principal eigenvector of the matrix in G5)	0.249946219	G10
$P^*_{\mathrm{B},0}$	0.164744870	SDPB0
$\dfrac{1}{2P^*_{\mathrm{B},0}}$	0.000000996	G11
$P^*_{\mathrm{A},0}$	0.276034548	SDPA0
$P^*_{\mathrm{B},1}$	0.162818974	SDPB1
$\dfrac{1}{2P^*_{\mathrm{B},1}}$	0.000001075	G12
$P^*_{\mathrm{A},1}$	0.271631913	SDPA1

though G5 takes longer to compute than G6, tests show that it is better to have G5 first. We calculate $P^*_{B,0}$ before $P^*_{A,0}$ since G9 and G10 are close approximations of $P^*_{A,0}$ and $P^*_{A,1}$, respectively. It will be evident that the order of solving the SDPs does not matter much.

We note here a few omissions as compared to the four-round tests. First, we have removed the two returning strategies, F4 and F5. These did not perform well in the four-round tests and preliminary tests show that they did not perform well in the six-round search either. Also, we do not have all the lower bounds for the eigenstrategies. Preliminary tests show that the lower bounds omitted take just as long or longer to compute than the corresponding upper bound, thus we just use the upper bound in the filter. Also, the marginal probabilities take approximately 5.49×10^{-6} seconds to compute which is negligible compared to the other times. Thus, we need not be concerned whether the strategies rely on the full probability distributions or marginal distributions.

Table 15: The number of protocols that get past each strategy in the filter for $d = 2$.

$d = 2$	$\nu = 1/3$	$\nu = 1/4$	$\nu = 1/5$	$\nu = 1/6$	$\nu = 1/7$	$\nu = 1/8$
Protocols	160,000	1,500,625	9,834,496	49,787,136	207,360,000	7.41 e+08
Symmetry	6,400	59,049	280,900	1,517,824	5,683,456	19,713,600
G1	3,200	20,412	82,680	389,312	1,397,024	4,115,880
G2	2,320	12,516	67,548	272,392	1,112,228	3,057,246
G3	1,725	9,627	52,424	223,034	899,450	2,526,712
G4	714	4,206	27,965	105,050	430,454	1,240,106
G5	210	684	7,743	20,373	112,435	228,274
G6	210	684	7,743	20,373	110,401	228,274
G7	30	48	1,285	1,856	10,979	17,831
G8	0	0	466	164	3,427	4,620
G9	0	0	466	164	3,419	4,512
G10	0	0	466	164	3,369	4,512
SDPB0	0	0	6	0	26	20
G11	0	0	6	0	26	20
SDPA0	0	0	0	0	0	0

Table 16: The number of protocols that get past each strategy in the filter for $d = 2$.

$d = 2$	$\nu = 1/9$	$\nu = 1/10$	$\nu = 1/11$	$\nu = 1/12$	$\nu = 1/13$	$\nu = 1/14$	$\nu = 1/15$
Protocols	2.34 e+09	6.69 e+09	1.75 e+10	4.28 e+10	9.83 e+10	2.13 e+11	4.43 e+11
Symmetry	58,247,424	155,276,521	401,080,729	973,502,401	2,052,180,601	4,632,163,600	9,372,176,100
G1	11,020,608	23,862,815	60,761,918	140,154,892	240,820,116	555,641,840	1,048,452,300
G2	8,944,136	18,717,210	50,337,094	110,274,108	204,522,468	444,537,964	877,684,860
G3	7,335,617	15,503,308	41,447,668	93,222,286	167,717,637	380,238,435	739,653,758
G4	3,477,093	8,534,326	20,503,550	45,888,192	91,991,055	185,971,770	350,105,435
G5	696,601	1,367,115	3,435,390	6,577,917	12,425,039	23,210,979	43,785,997
G6	688,613	1,367,115	3,435,390	6,577,917	12,258,117	23,097,713	43,188,099
G7	57,598	87,303	232,382	355,057	678,384	1,051,339	1,977,185
G8	17,512	18,105	64,273	86,272	177,297	230,146	479,088
G9	16,005	15,689	50,847	74,114	143,172	195,858	411,864
G10	15,875	15,124	49,819	71,439	137,232	185,696	386,741
SDPB0	68	58	152	126	492	346	594
G11	68	58	152	126	492	346	594
SDPA0	0	0	0	0	0	0	0

Table 17: The number of protocols that get past each strategy in the filter for for $d = 3$.

$d = 3$	$\nu = 1/2$	$\nu = 1/3$	$\nu = 1/4$
Protocols	$4,100,625$	$741,200,625$	$60,037,250,625$
Symmetry	$68,121$	$6,395,841$	$279,324,369$
G1	$42,282$	$5,222,385$	$180,500,400$
G2	$8,748$	$3,324,650$	$86,151,600$
G3	$5,643$	$1,958,070$	$58,038,667$
G4	161	$714,393$	$30,773,918$
G5	0	$464,538$	$15,310,116$
G6	0	$464,538$	$15,310,116$
G7	0	$310,518$	$6,557,007$
G8	0	$284,418$	$5,447,015$
G9	0	$284,418$	$5,393,911$
G10	0	$284,418$	$5,393,911$
SDPB0	0	$2,655$	$24,012$
G11	0	$2,655$	$24,012$
SDPA0	0	0	0

Observations on the six-round search

We first note that the filter does not work as effectively as in the four-round case. The six-round search for $d = 3$ ran for about a month. In comparison, all the four-round searches ran in the matter of days.

The symmetry arguments cut down the number of protocols we need to examine significantly, by a factor of roughly 100. Note that in the four-round case it was a factor of $1,000,000$ (for the $d = 9$ case). This can be explained by the weaker index symmetry in the six-round version.

Cheating strategy G1 cut the number of protocols down by a factor of 10 with G2 performing less well than the corresponding strategy in the four-round tests. G5 also performed well, but after this, G6 was not much help. G7 and G8 cut down the number of protocols by a factor of 10 each in the $d = 2$ case, but not as much in the $d = 3$ case. The next notable strategy was G10, being G9 with β_0 and β_1 swapped, which performed very poorly. It seems that the swapped strategies do not help much in the filters, that is, there is not much discrepancy between cheating towards 0 or 1. SDPB0 almost filtered out the rest of the protocols, relying on SDPA0 to stop the rest. The implicit strategy from Kitaev's bound, G11, did not perform well after SDPB0 (note that it relies on SDPB0 so it is computed afterwards). Again, we notice that no protocols with bias less than 0.2499 were found.

We notice that G9 and G10, the improved eigenstrategies for Alice, hardly filter out any protocols, if any at all, in the low-precision tests. In these strategies, we compute a value on the concave hull

$$\mathrm{conc}\left\{\frac{1}{2}\beta_{0,y}\,\mathrm{F}(\cdot,\alpha_0), \frac{1}{2}\beta_{1,y}\,\mathrm{F}(\cdot,\alpha_1)\right\},$$

for every value of y. In the eigenstrategy, we approximate the concave hull with the one of the two that has the larger constant. When we choose these constants according to a coarse mesh, e.g., $\nu = 1/3$ or $\nu = 1/4$, the one with the larger constant is a very good approximation of the concave hull. It appears that, we need finer precisions to bring out the power of this strategy in the filter.

In all our searches, we did not find any protocols with bias less than 0.2499, and it seems that $1/4$ might be the least bias achievable by the class of protocols we study. To further test this conjecture, in the next two subsections we present two other kinds of search.

8.3 Random offset

We would like to test more protocols, and also avoid anomalies that may have arisen in the previous tests due to the structure of the mesh we use and also any special relation the protocol states may have with each other due to low precision. The six-round searches

Table 18: The percentage of protocols that get stopped by each strategy in the worst case over 100 random instances of offset parameter δ.

$d = 2$	$\nu = 1/3$	$\nu = 1/4$	$\nu = 1/5$	$\nu = 1/6$
G1	71.87%	82.35%	84.06%	86.63%
G2	17.18%	29.80%	15.80%	24.15%
G3	8.17%	10.73%	13.46%	12.12%
G4	51.45%	49.68%	53.99%	48.44%
G5	70.00%	83.29%	78.02%	82.86%
G6	0%	0%	0%	0%
G7	75.00%	92.43%	87.32%	94.35%
G8	100%	100%	49.10%	100%
G9			0%	
G10			0%	
SDPB0			100%	

take a long time, which restricts the precision ν we can use. The resulting mesh is also highly structured. We would like to test protocol parameters that do not necessarily have such regular entries. With this end in mind, we offset all of the values in the search by some random additive term $\delta > 0$. For example, say the entries of α_0, α_1, β_0, and β_1 have been selected from the set $\{0, \nu, 2\nu, \ldots, 1 - \nu, 1\}$. With an offset parameter $\delta \in (0, \nu/2)$, we use the range

$$\{\delta, \delta + \nu, \delta + 2\nu, \ldots, \delta + 1 - \nu\}.$$

Note that this destroys index symmetry. The simplest way to see this is to consider the 2-dimensional probability distributions created in this way. They are

$$\left\{ \begin{bmatrix} \delta \\ 1 - \delta \end{bmatrix}, \begin{bmatrix} \delta + \nu \\ 1 - \delta - \nu \end{bmatrix}, \begin{bmatrix} \delta + 2\nu \\ 1 - \delta - 2\nu \end{bmatrix}, \ldots, \begin{bmatrix} \delta + 1 - \nu \\ \nu - \delta \end{bmatrix} \right\}.$$

We see that the set of first entries is not the same as the set of second entries when $\delta > 0$. We choose the last entry in each vector to be such that the entries add to 1. Since we generate all four of the probability distributions in the same manner, we can still apply the symmetry arguments to suppose α_0 has the largest entry out of both α_0 and α_1 and similarly for β_0 and β_1.

Table 18 (above) shows how well each strategy in the filter performs in the worst case and Table 19 (on the next page) shows the average case over 100 random choices of offset parameter $\delta \in [0, 1/100]$.

Table 19: The percentage of protocols that get stopped by each strategy in the average case over 100 random instances of offset parameter δ.

$d = 2$	$\nu = 1/3$	$\nu = 1/4$	$\nu = 1/5$	$\nu = 1/6$
G1	85.75%	87.30%	89.42%	90.47%
G2	17.18%	29.80%	15.80%	24.15%
G3	10.85%	13.15%	14.53%	12.35%
G4	62.49%	52.53%	55.34%	53.03%
G5	70.00%	87.11%	93.46%	93.29%
G6	0%	0%	0%	0%
G7	98.70%	99.01%	96.58%	98.77%

Observations on the random offset tests

We notice that G6 performs very poorly on these tests. We need finer precision to see the effects of G6 in the filter. Also, G1 performs generally better as the filter precision increases. We see from the previous tables that it should stay at roughly 90%. We see that G5 and G7 perform very well. G7 sometimes filters out the rest (why the average case table only displays up to G7). G8 performs well most of the time, except in the $\nu = 1/5$ case in the worst case table. Few protocols made it past the entire filter, and only SDPB0 needed to be solved of the four SDPs. No protocols with bias at most 0.2499 were found.

8.4 Computer aided bounds on bias

The search algorithm has the potential to give us computer aided *proofs* that certain coin-flipping protocols have bias within a small interval. In this section, we describe the kind of bound we can deduce under the assumption that the software provides us an independently verifiable upper bound on the additive error in terms of the objective value.

We begin by showing that any state $\xi \in \mathbb{R}^D$ of the form used in the protocols is suitably close to a state given by the mesh used in the search algorithm. For an integer $N \geq 1$, let $\mathbb{M}_N = \{j/N : j \in \mathbb{Z}, 0 \leq j \leq N\}$.

Lemma 8.1 *Let $N \geq 1$ be an integer. Consider the state $\xi = \sum_{i=1}^{D} \sqrt{\gamma_i}\, e_i$ in $\mathbb{R}^D$, where $\gamma \in \mathrm{Prob}^D$. Then there is a probability distribution $\gamma' \in \mathrm{Prob}^D \cap \mathbb{M}_N^D$ such that the corresponding state $\xi' = \sum_{i=1}^{D} \sqrt{\gamma_i'}\, e_i$ satisfies $\xi^* \xi' \geq 1 - D/2N$.*

Proof: Let $\tilde{\gamma}_i = \lfloor \gamma_i N \rfloor /N$ for $i \in \{1, 2, \ldots, D\}$. Note that $\sum_{i=1}^{D} \tilde{\gamma}_i \leq 1$, and that $1 - \sum_{i=1}^{D} \tilde{\gamma}_i = \sum_{i=1}^{D} \gamma_i - \sum_{i=1}^{D} \tilde{\gamma}_i = j/N$, for some $j \in \{0, 1, 2, \ldots, D\}$. We may obtain γ' by

adding $1/N$ to j coordinates of $\tilde{\gamma}$. For concreteness, let $\gamma'_i = \tilde{\gamma}_i + 1/N$ for $i \in \{1, 2, \ldots, j\}$ and $\gamma'_i = \tilde{\gamma}_i$ for $i \in \{j+1, \ldots, D\}$. We therefore have $\|\gamma - \gamma'\|_1 \leq D/N$, and

$$\xi^* \xi' \quad = \quad \mathrm{F}(\gamma, \gamma')^{1/2} \quad \geq \quad 1 - \frac{D}{2N} \;,$$

by Proposition 2.1. $\quad \square$

The above lemma helps us show that any protocol in the family we consider is approximated by one given by the mesh.

Lemma 8.2 *Consider a bit-commitment based coin-flipping protocol $\mathcal{A}$ with bias ϵ of the form defined in Section 3.3. Let $\mathcal{A}$ be specified by the 4-tuple $(\alpha_0, \alpha_1, \beta_0, \beta_1)$, where $\alpha_i, \beta_i \in \mathrm{Prob}^D$. There is a protocol $\mathcal{A}'$ with bias ϵ' of the same form, defined by a 4-tuple $(\alpha'_0, \alpha'_1, \beta'_0, \beta'_1)$, where $\alpha'_i, \beta'_i \in \mathrm{Prob}^D \cap \mathbb{M}_N^D$, such that $|\epsilon - \epsilon'| \leq 2\sqrt{D/N}$.*

Proof: The statement of the lemma is vacuous if $1 - D/2N < 0$, we therefore assume $1 - D/2N \geq 0$. We show that $\epsilon' \leq \epsilon + 2\sqrt{D/N}$. The other inequality $\epsilon \leq \epsilon' + 2\sqrt{D/N}$ follows similarly.

Without loss in generality, assume that bias ϵ' is achieved when Bob cheats towards 0 in protocol $\mathcal{A}'$. Recall

$$\psi \quad = \quad \frac{1}{\sqrt{2}} \left(e_0 \otimes e_0 \otimes \psi_0 + e_1 \otimes e_1 \otimes \psi_1 \right) \;, \qquad \text{and}$$

$$\Pi_{\mathrm{A},0} \quad = \quad \sum_{b \in \{0,1\}} e_b e_b^* \otimes e_b e_b^* \otimes \phi_b \phi_b^* \;.$$

Let the probability distributions $\alpha'_0, \alpha'_1, \beta'_0, \beta'_1$ and states $\psi'_0, \psi'_1, \phi'_0, \phi'_1$ corresponding to the distributions $\alpha_0, \alpha_1, \beta_0, \beta_1$, respectively, be the ones guaranteed by Lemma 8.1. Let

$$\psi' \quad = \quad \frac{1}{\sqrt{2}} \left(e_0 \otimes e_0 \otimes \psi'_0 + e_1 \otimes e_1 \otimes \psi'_1 \right) \;, \qquad \text{and}$$

$$\Pi'_{\mathrm{A},0} \quad = \quad \sum_{b \in \{0,1\}} e_b e_b^* \otimes e_b e_b^* \otimes \phi'_b (\phi'_b)^* \;.$$

We have $\psi^* \psi' \geq 1 - \frac{D}{2N}$, by Lemma 8.1, and

$$\begin{aligned}
\left\| \psi'(\psi')^* - \psi \psi^* \right\|_* \quad &\leq \quad 2 \left(1 - (\psi^* \psi')^2 \right)^{1/2} \\
&\leq \quad 2\sqrt{D/N} \;,
\end{aligned}$$

by Proposition 2.1. Further,

$$\begin{aligned}
\left\| \Pi'_{\mathrm{A},0} - \Pi_{\mathrm{A},0} \right\|_{\mathrm{op}} \quad &\leq \quad \max \left\{ \left\| \phi'_0(\phi'_0)^* - \phi_0 \phi_0^* \right\|_{\mathrm{op}}, \left\| \phi'_1(\phi'_1)^* - \phi_1 \phi_1^* \right\|_{\mathrm{op}} \right\} \\
&\leq \quad \sqrt{D/N} \;,
\end{aligned}$$

60

using the identity $\|vv^* - uu^*\|_{\mathrm{op}} = \left(1 - (v^*u)^2\right)^{1/2}$ for normalized real vectors v and u. Here, $\|X\|_{\mathrm{op}}$ denotes the operator norm of X, namely the largest singular value of the matrix X.

For this analysis, we assume that the protocol $\mathcal{A}'$ is presented in the form described in Section 3.1, and the two parties start with joint initial state $e_0^{\otimes 4n}$, apply $U_1, U_2, \ldots, U_{2n}$ alternately, and finally measure their parts of the system to obtain the output.

Consider Bob's cheating strategy towards 0 (which we assumed achieves bias ϵ'). As in the proof of Lemma 4.1, it follows that there are spaces $\mathcal{H}_i$ and corresponding unitary operations U_i' on them for even $i \leq 2n$ that characterize his cheating strategy. When Alice measures $\zeta' = (U_{2n}' U_{2n-1} U_{2n-2}' \cdots U_1) e_0^{\otimes 4n}$, she obtains outcome 0 with probability $\left\|\Pi_{\mathrm{A},0}' \zeta'\right\|_2^2 = \frac{1}{2} + \epsilon'$. (In the expression for the final state ζ', we assume that the unitary operations extend to the combined state space by tensoring with identity over the other part.)

We consider the same cheating strategy for Bob in the protocol $\mathcal{A}$, in which Alice starts with the commitment state ψ, and performs the measurement $\{\Pi_{\mathrm{A},0}, \Pi_{\mathrm{A},1}, \Pi_{\mathrm{A,abort}}\}$. This corresponds to a different initial unitary transformation for Alice instead of U_1. Let ζ be the corresponding final joint state. Note that ψ is mapped to ζ using the same unitary transformation that maps ψ' to ζ' since Bob is using the same cheating strategy. The probability of outcome 0 is $\|\Pi_{\mathrm{A},0}\zeta\|_2^2 \leq \frac{1}{2} + \epsilon$, as the protocol $\mathcal{A}$ has bias ϵ. We may bound the difference in probabilities as follows.

$$
\begin{aligned}
\epsilon' - \epsilon &\leq \mathrm{Tr}\left(\Pi_{\mathrm{A},0}' \zeta'(\zeta')^*\right) - \mathrm{Tr}\left(\Pi_{\mathrm{A},0}\zeta\zeta^*\right) \\
&= \mathrm{Tr}\left((\Pi_{\mathrm{A},0}' - \Pi_{\mathrm{A},0})\zeta'(\zeta')^*\right) + \mathrm{Tr}\left(\Pi_{\mathrm{A},0}(\zeta'(\zeta')^* - \zeta\zeta^*)\right) \\
&\leq \left\|\Pi_{\mathrm{A},0}' - \Pi_{\mathrm{A},0}\right\|_{\mathrm{op}} + \frac{1}{2}\left\|\zeta\zeta^* - \zeta'(\zeta')^*\right\|_* \qquad \text{By Eq. (1)} \\
&= \left\|\Pi_{\mathrm{A},0}' - \Pi_{\mathrm{A},0}\right\|_{\mathrm{op}} + \frac{1}{2}\left\|\psi\psi^* - \psi'(\psi')^*\right\|_* \\
&\leq 2\sqrt{D/N} \ ,
\end{aligned}
$$

as claimed. $\square$

We may infer bounds on classes of protocols using the search algorithm and the lemma above. Suppose the computational approximation to the bias obtained by the algorithm has net additive error τ due to the protocol filter and SDP solver and the finite precision arithmetic used in the computations. If the algorithm reports that there are no protocols with bias at most ϵ^* given by a mesh with precision parameter N, then it holds that there are no 4-tuples, even outside the mesh, with bias at most $\epsilon^* - 2\sqrt{D/N} - \tau$. Here D is the dimension of Alice's (or Bob's) first n messages (i.e., commitment states used, or equivalently, the size of the support of an element of the 4-tuple).

A quick calculation with $\epsilon^* = 0.2499$ shows that mesh fineness parameter $N \geq 2184 \times d$ for four-round protocols and $N \geq 2184 \times d^2$ for six-round protocols with message dimen-

sion d, would be sufficient for us to conclude that such protocols *do not achieve* optimal bias ≈ 0.2071. We would then obtain *computer aided* lower bounds for new classes of bit-commitment based protocols. Thus, a refinement of the search algorithm that allows finer meshes for messages of larger dimension and over more rounds would be well worth pursuing.

8.5 New bounds for four-round qubit protocols

We can derive analytical bounds on the bias of four-round protocols using the strengthened Fuchs-van de Graaf inequality for qubit states, below:

Proposition 8.3 ([SR01]) *For any quantum states $\rho_1, \rho_2 \in \mathbb{S}^2_+$, i.e., qubits, we have*

$$1 \quad \leq \quad \Delta(\rho_1, \rho_2) + \mathrm{F}(\rho_1, \rho_2) \ .$$

Recall from Section 5 that Bob can cheat in a four-round protocol with probability bounded below by

$$P^*_{B,0} \quad \geq \quad \frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\beta_0, \beta_1)} \tag{15}$$

and

$$P^*_{B,0} \quad \geq \quad \frac{1}{2} + \frac{1}{2}\Delta(\alpha_0, \alpha_1) \tag{16}$$

and Alice can cheat with probability bounded below by

$$P^*_{A,0} \quad \geq \quad \left(\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\alpha_0, \alpha_1)}\right)\left(\frac{1}{2} + \frac{1}{2}\Delta(\beta_0, \beta_1)\right) \ . \tag{17}$$

If $\beta_0, \beta_1 \in \mathrm{Prob}^2$, then by (15) and Proposition 8.3, we have $\Delta(\beta_0, \beta_1) \geq 4P^*_{B,0}(1 - P^*_{B,0})$ and if $\alpha_0, \alpha_1 \in \mathrm{Prob}^2$, then from (16) and Proposition 8.3, we have $\mathrm{F}(\alpha_0, \alpha_1) \geq 2 - 2P^*_{B,0}$. Combining these two bounds with (17), we get

$$4P^*_{A,0} \quad \geq \quad \left(1 + \sqrt{2 - 2P^*_{B,0}}\right)\left(1 + 4P^*_{B,0}(1 - P^*_{B,0})\right)$$

implying $\max\{P^*_{A,0}, P^*_{B,0}\} \geq 0.7487 > 1/\sqrt{2} \approx 0.7071$. In fact, using the Fuchs-van de Graaf inequalities from Proposition 2.1, we can get bounds when they are not both two-dimensional. If β_0, β_1 are two-dimensional and α_0, α_1 are not, we get a lesser bound of $\max\{P^*_{A,0}, P^*_{B,0}\} \geq 0.7140 > 1/\sqrt{2}$. On the other hand, if α_0, α_1 are two-dimensional and β_0, β_1 are not, then we get $\max\{P^*_{A,0}, P^*_{B,0}\} \geq 0.7040 \not> 1/\sqrt{2}$, so we do not rule out the possibility of optimal protocols with these parameters.

Note that tests where α_0, α_1 are two-dimensional are subsumed in the higher-dimensional tests we performed. However, future experiments could include computationally testing the case where Alice's first message is two-dimensional and Bob's first message has dimension 10 or greater.

8.6 Zoning-in on near-optimal protocols

The computational tests that we performed so far suggest that there are no protocols with cheating probabilities less than 0.7499, that is, slightly smaller than the best known constructions. The tests also show that the number of protocols grows very large as the mesh precision increases. This poses the question of whether there are protocols that have optimal cheating probabilities just slightly less than $3/4$ when one considers increased mesh precisions. In this subsection, we focus on searching for such protocols.

There are a few obstacles to deal with in such a search. The first is that increasing the precision of the mesh drastically increases the number of protocols to be tested. To deal with this, we restrict the set of parameters to be tested by only considering protocols which are close to optimal, i.e., near-optimal protocols. In other words, we "zone in" on some promising protocols to see if there is any hope of improving the bias by perturbing some of the entries. To do this, we fix a near-optimal protocol and create a mesh over a small ball around the entries in each probability vector. We would like a dramatic increase in precision, so we use a ball of radius 2ν (unless stated otherwise), yielding up to 5 increments tested around each entry. This gives us the advantage of having a constant number of protocols to check, independent of the mesh precision. However, this comes at the cost that we lose symmetry, since we do not wish to permute the entries nor the probability distributions defining the protocol.

Another challenge is to find the near-optimal protocols. The approach we take is to keep track of the best protocol found, updating the filter threshold accordingly. There are two issues with this approach. One is that increasing the threshold decreases the efficiency of the filter, so we are not able to search over the same mesh precisions given earlier in this section. The second is that there is an abundance of protocols with cheating probabilities exactly equal to $3/4$. As was done in Section 7, we can embed an optimal three-round protocol with optimal cheating probabilities $3/4$ into a four-round or six-round protocol. One way to do this is to set $\alpha_0 = \alpha_1$ (i.e. Alice's first n messages contain *no* information) or by setting $\beta_0 \perp \beta_1$ (i.e. Bob's first message reveals b, making the rest of his messages meaningless). So we already know many protocols with cheating probabilities equal to $3/4$, but can we find others? We now discuss the structure of near-optimal protocols in the case of four-round and six-round protocols, and how we zone in on them.

Four-round version

For the four-round search, we fix a message dimension $d = 5$ and use precision parameters $\nu \in \{1/7, 1/8, 1/9, 1/10, 1/11\}$. This search yields a minimum (computer verified) bias of $\epsilon = 0.2647$ when we rule out protocols with $\alpha_0 = \alpha_1$ or $\beta_0 \perp \beta_1$. In other words, we have that all of the protocols tested had one of the following three properties:

- $\alpha_0 = \alpha_1$,

- $\langle \beta_0, \beta_1 \rangle = 0$,

- $\max \left\{ P^*_{A,0}, P^*_{A,1}, P^*_{B,0}, P^*_{B,1} \right\} \geq 0.7647$.

This suggests that near-optimal four-round protocols behave similarly to optimal three-round protocols. We now zone in on two protocols, one representing each of the first two conditions above. The first protocol is

$$\alpha_0 = \frac{1}{2} [0,0,0,1,1]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{2} [0,0,1,0,1]^{\mathrm{T}}, \quad \beta_0 = [0,0,0,0,1]^{\mathrm{T}}, \quad \beta_1 = [0,0,0,1,0]^{\mathrm{T}}$$

which satisfies $\beta_0 \perp \beta_1 = 0$ and has all four (computationally verified) cheating probabilities equal to $3/4$. The second protocol is

$$\alpha_0 = [0,0,0,0,1]^{\mathrm{T}}, \quad \alpha_1 = [0,0,0,0,1]^{\mathrm{T}}, \quad \beta_0 = \frac{1}{2} [0,0,0,1,1]^{\mathrm{T}}, \quad \beta_1 = \frac{1}{2} [0,0,1,0,1]^{\mathrm{T}}$$

which satisfies $\alpha_0 = \alpha_1$ and has all four (computationally verified) cheating probabilities equal to $3/4$. Tables 20 and 21 display the zoning-in searches for these two protocols with threshold exactly $3/4$. Note we use mesh precisions up to 10^{-16} which, by Lemma 8.2, can guarantee us a change in bias up to 4×10^{-8}. A (computationally verified) change in bias of this magnitude could be argued to be an actual decrease in bias and not an error due to finite precision arithmetic.

Table 20: The number of four-round protocols that get past each strategy when zoning-in on the first near-optimal protocol (showing F1 and only the other strategies that helped to weed out protocols).

$d = 5$	$\nu = 1/10^{10}$	$\nu = 1/10^{11}$	$\nu = 1/10^{12}$	$\nu = 1/10^{13}$	$\nu = 1/10^{14}$	$\nu = 1/10^{15}$	$\nu = 1/10^{16}$
F1	119,574,225	119,574,225	119,574,225	119,574,225	119,574,225	119,574,225	119,574,225
F2	20,253,807	20,253,807	20,411,271	21,067,371	20,253,807	20,253,807	6,337,926
F3	493,557	493,557	493,557	581,503	493,557	498,504	33,279
F6	493,557	493,557	493,557	576,819	493,557	480,276	13,695
F7	981	981	981	1,245	981	855	0
F8	0	0	0	0	0	29	0
F10	0	0	0	0	0	0	0

Table 21: The number of four-round protocols that get past each strategy when zoning-in on the second near-optimal protocol (showing F1 and only the other strategies that helped to weed out protocols).

$d = 5$	$\nu = 1/10^{10}$	$\nu = 1/10^{11}$	$\nu = 1/10^{12}$	$\nu = 1/10^{13}$	$\nu = 1/10^{14}$	$\nu = 1/10^{15}$	$\nu = 1/10^{16}$
F1	9,277,254	9,277,254	9,277,254	9,277,254	9,277,254	8,516,178	4,953,555
F3	907,608	907,608	913,496	912,864	907,608	828,952	1,030,152
F6	693,576	693,576	695,016	713,424	693,576	412,392	8,624
F7	45,376	45,376	45,376	55,064	43,264	3,136	5,056
F8	0	0	0	0	0	68	3,140
F9	0	0	0	0	0	8	2,072
F10	0	0	0	0	0	0	0

Note that not all filter strategies are useful in the zoning-in tests. For example, if the strategy F1 $\approx 1/2 < 3/4$ for the protocol we are zoning-in on, then it never filters out any protocols with the precisions considered. Considering this, and by examining the tables, we see that most strategies filter out many protocols, or none at all. Also from the tables, we see that no protocols get through the entire filter. Notice that we needed to use more strategies than were needed in previous tables, namely F9 and F10. In the previous searches, F8 was the last filter strategy needed, thus demonstrating some protocols which F8 fails to filter out (noting a larger threshold was used here than in the previous tests). It is worth noting the efficiency of the four-round filter. The algorithm did not need to solve for any optimal cheating values in any of the four-round zoning-in tests.

These tables suggest that perturbing the entries of the parameters defining these two near-optimal protocols does not yield better bias.

Six-round version

For the six-round search, we fix a message dimension $d = 2$ and use precision parameters $\nu \in \{1/7, 1/8, 1/9, 1/10, 1/11, 1/12\}$. For $\nu > 1/12$, the test results were similar to the four-round version, that all of the protocols tested had one of the following three properties:

- $\alpha_0 = \alpha_1$,

- $\langle \beta_0, \beta_1 \rangle = 0$,

- $\max \left\{ P_{A,0}^*, P_{A,1}^*, P_{B,0}^*, P_{B,1}^* \right\} \geq 0.7521$.

We choose the following two near-optimal protocols to represent the first two conditions:

$$\alpha_0 = \frac{1}{2} [0, 0, 1, 1]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{2} [0, 1, 0, 1]^{\mathrm{T}}, \quad \beta_0 = [0, 0, 0, 1]^{\mathrm{T}}, \quad \beta_1 = [0, 0, 1, 0]^{\mathrm{T}},$$

which satisfies $\beta_0 \perp \beta_1 = 0$, and

$$\alpha_0 = [0, 0, 0, 1]^{\mathrm{T}}, \quad \alpha_1 = [0, 0, 0, 1]^{\mathrm{T}}, \quad \beta_0 = \frac{1}{2} [0, 0, 1, 1]^{\mathrm{T}}, \quad \beta_1 = \frac{1}{2} [0, 1, 0, 1]^{\mathrm{T}},$$

which satisfies $\alpha_0 = \alpha_1$. Both of these protocols have all four (computationally verified) cheating probabilities equal to $3/4$.

However, when $\nu = 1/12$, we found several protocols with a (computationally found) bias of 0.25. We therefore searched for all protocols with bias 0.2501 or less. We discovered the following 4 protocols, no two of which are equivalent to each other with respect to symmetry. Note that these protocols bear no resemblance to any bias $1/4$ protocols previously discovered. These protocols are below:

$$\alpha_0 = \frac{1}{3}\,[0,1,1,1]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{3}\,[1,1,0,1]^{\mathrm{T}}, \quad \beta_0 = \frac{1}{12}\,[0,3,0,9]^{\mathrm{T}}, \quad \beta_1 = \frac{1}{12}\,[0,3,9,0]^{\mathrm{T}}$$

and

$$\alpha_0 = \frac{1}{3}\,[0,1,1,1]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{3}\,[1,1,0,1]^{\mathrm{T}}, \quad \beta_0 = \frac{1}{12}\,[1,2,0,9]^{\mathrm{T}}, \quad \beta_1 = \frac{1}{12}\,[1,2,9,0]^{\mathrm{T}}$$

and

$$\alpha_0 = \frac{1}{3}\,[0,1,1,1]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{3}\,[1,1,1,0]^{\mathrm{T}}, \quad \beta_0 = \frac{1}{12}\,[0,3,0,9]^{\mathrm{T}}, \quad \beta_1 = \frac{1}{12}\,[0,3,9,0]^{\mathrm{T}}$$

and

$$\alpha_0 = \frac{1}{3}\,[0,1,1,1]^{\mathrm{T}}, \quad \alpha_1 = \frac{1}{3}\,[1,1,1,0]^{\mathrm{T}}, \quad \beta_0 = \frac{1}{12}\,[1,2,0,9]^{\mathrm{T}}, \quad \beta_1 = \frac{1}{12}\,[1,2,9,0]^{\mathrm{T}}.$$

Note that these four protocols have the property that all the filter strategies for them have cheating probabilities strictly less than $3/4$. Since many of these strategies are derived from optimal three-round strategies, this property makes them especially interesting. (Other six-round protocols were found. However, these were equivalent to the ones above, under the equivalence relation described in Section 6.)

We now zone in on these six protocols as indicated in the following tables. Note that we decrease the radius of the balls to ν for the third, fourth, fifth, and sixth protocol (compared to 2ν for the other protocols). This is for two reasons. One is that most the entries are bounded away from 0 or 1, making the intersection of the ball and valid probability vectors large. Second, the filter has to work harder in this case since many of the filter cheating probabilities are bounded away from $3/4$ and thus more computationally expensive cheating probabilities need to be computed.

Preliminary tests show that when zoning-in on some of these 6 protocols, the default SDP solver precision is not enough to determine whether the bias is strictly less than $3/4$, or whether it is numerical round-off. To provide a further test, we add an extra step for those protocols that get through the filter and SDPs, we increase the SDP solver accuracy (set pars.eps = 0 in SeDuMi) and let the solver run until no more progress is being made. The row "Better Accuracy" shows how many protocols get through this added step. Furthermore, we use the maximum of the primal and dual values when calculating the optimal cheating values since we are not guaranteed exact feasibility of both primal and dual solutions in these computational experiments.

Table 22: The number of six-round protocols that get past each strategy when zoning-in on the first near-optimal protocol (showing G1 and only the other strategies that helped to weed out protocols).

$d = 2$	$\eta = 1/10^{10}$	$\eta = 1/10^{11}$	$\eta = 1/10^{12}$	$\eta = 1/10^{13}$	$\eta = 1/10^{14}$	$\eta = 1/10^{15}$	$\eta = 1/10^{16}$
G1	1,476,225	1,476,225	1,476,225	1,476,225	1,476,225	1,476,225	1,476,225
G2	874,800	874,800	874,800	879,174	874,800	874,800	601,425
G3	533,439	533,439	533,655	538,326	533,439	448,065	149,040
G5	20,434	20,434	20,434	21,250	20,434	14,494	359
G7	656	656	668	685	579	455	0
G8	70	70	70	76	42	21	0
G9	0	0	0	0	0	0	0

Table 23: The number of six-round protocols that get past each strategy when zoning-in on the second near-optimal protocol (showing G1 and only the other strategies that helped to weed out protocols).

$d = 2$	$\eta = 1/10^{10}$	$\eta = 1/10^{11}$	$\eta = 1/10^{12}$	$\eta = 1/10^{13}$	$\eta = 1/10^{14}$	$\eta = 1/10^{15}$	$\eta = 1/10^{16}$
G1	93,312	93,312	93,312	93,312	93,312	86,022	40,824
G4	38,061	38,061	38,061	38,061	38,061	28,125	4,995
G5	2,664	2,664	2,664	2,716	2,664	1,418	0
G6	2,376	2,376	2,376	2,420	2,376	1,174	0
G9	1,270	0	0	0	0	0	0
G10	774	0	0	0	0	0	0
SDPA0	538	0	0	0	0	0	0
SDPA1	474	0	0	0	0	0	0
Better Accuracy	0	0	0	0	0	0	0

Table 24: The number of six-round protocols that get past each strategy when zoning-in on the third, fourth, fifth and sixth near-optimal protocols (showing G1 and only the other strategies that helped to weed out protocols).

$d = 2$	$\eta = 1/10^{10}$	$\eta = 1/10^{11}$	$\eta = 1/10^{12}$	$\eta = 1/10^{13}$	$\eta = 1/10^{14}$	$\eta = 1/10^{15}$	$\eta = 1/10^{16}$
G1	34,992	34,992	34,992	34,992	34,992	34,992	34,992
SDPB0	9,720	9,720	9,720	9,720	9,720	9,720	27,215
SDPA0	0	0	0	0	0	0	0

$d = 2$	$\eta = 1/10^{10}$	$\eta = 1/10^{11}$	$\eta = 1/10^{12}$	$\eta = 1/10^{13}$	$\eta = 1/10^{14}$	$\eta = 1/10^{15}$	$\eta = 1/10^{16}$
G1	99,144	99,144	93,312	99,144	99,144	99,144	99,144
SDPB0	0	0	0	0	0	0	0

$d = 2$	$\eta = 1/10^{10}$	$\eta = 1/10^{11}$	$\eta = 1/10^{12}$	$\eta = 1/10^{13}$	$\eta = 1/10^{14}$	$\eta = 1/10^{15}$	$\eta = 1/10^{16}$
G1	34,992	34,992	34,992	34,992	34,992	34,992	34,992
SDPB0	9,720	9,720	9,720	9,720	9,720	9,720	27,215
SDPA0	0	0	0	0	0	0	0

$d = 2$	$\eta = 1/10^{10}$	$\eta = 1/10^{11}$	$\eta = 1/10^{12}$	$\eta = 1/10^{13}$	$\eta = 1/10^{14}$	$\eta = 1/10^{15}$	$\eta = 1/10^{16}$
G1	99,144	99,144	93,312	99,144	99,144	99,144	99,144
SDPB0	0	0	0	0	0	0	0

We see in Tables 22, 23, and 24 that zoning-in on the six protocols yields no protocols with bias less than $1/4$. The zoning-in tests for the second near-optimal protocol are the only ones where we needed the added step of increasing the SDP solver accuracy. We see that this added step removed the remaining protocols.

We remark on the limitations of using such fine mesh precisions. For example, when zoning-in on the fourth and sixth protocol, only two strategies were used, G1 and SDPB0. These are both strategies for Bob which suggests that there are some numerical precision issues. We expect that some perturbations would decrease Bob's cheating probability, for example when α_0 and α_1 become "closer" and β_0 and β_1 remain the same. However, the precisions used in these searches do not find any such perturbations.

From the outcome of the zoning-in tests, along with the computational evidence from all the other tests we conducted, we conjecture that any strong coin-flipping protocol based on bit-commitment as defined formally in Section 3.3 has bias at least $1/4$ (Conjecture 1.1 in Section 1.3).

9   Conclusions

We introduced a parameterized family of quantum coin-flipping protocols based on bit-commitment, and formulated the cheating probabilities of Alice and Bob as simple semidefinite programs. Using these semidefinite programming formulations, we designed an algorithm to search for parameters yielding a protocol with small bias. We exploited symmetry and developed cheating strategies to create a protocol filter so that a wider array of protocols can be searched. For example, without the heuristics used in this paper, it would have taken over 69 million years to search the same 3×10^{16} protocols that we tested.

Using the search algorithm, we searched four and six-round protocols from a mesh over the parameter space, with messages of varying dimension and with varying fineness for the mesh. After the initial systematic searches, no protocols having all four cheating probabilities less than 0.7499 were found. We then performed a search over a randomly translated mesh to avoid any anomalies that may have occurred while testing structured parameter sets. These tests also did not find any protocols with cheating probabilities less than 0.7499. Our final tests zoned-in on protocols with maximum cheating probability $3/4$ to test whether there are protocols with cheating probabilities between 0.7499 and 0.75. A computational search to find such protocols yielded 8 equivalence classes of protocols representing all the protocols with cheating probabilities equal to $3/4$. Four of these protocols bear no resemblance to previously known protocols with bias $1/4$. Zoning-in on these protocols showed that we cannot improve the bias by perturbing the parameters defining the protocols. Improvements to the algorithm may yield computer aided proofs of bounds on the bias of new sets of protocols.

An obvious open problem is to resolve the conjecture that all the protocols in the family we study have bias at least $1/4$. It seems the smallest bias does not decrease when

the number of messages increases from four rounds to six. We conjecture the smallest bias does not decrease even if more messages are added. One way to show this is to find closed-form expressions of the optimal objective values of the SDP formulations. This would be of great theoretical significance since very few highly interactive protocols (such as those examined in this paper) have been characterized by closed-form expressions for their bias or even by a description of optimal cheating strategies.

Acknowlegdements

References

[ABDR04] Andris Ambainis, Harry Buhrman, Yevgeniy Dodis, and Hein Röhrig. Multiparty quantum coin flipping. In *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, pages 250–259. IEEE Computer Society, 2004.

[ACG+13] Dorit Aharonov, André Chailloux, Maor Ganz, Iordanis Kerenidis, and Loïck Magnin. A simpler proof of existence of quantum weak coin flipping with arbitrarily small bias. Manuscript, 2013.

[AG03] Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Mathematical Programming*, 95:3–51, 2003.

[Amb01] Andris Ambainis. A new protocol and lower bounds for quantum coin flipping. In *Proceedings of 33rd Annual ACM Symposium on the Theory of Computing*, pages 134 – 142. ACM, 2001.

[Amb02] Andris Ambainis. Lower bound for a class of weak quantum coin flipping protocols. Available as arXiv.org e-Print quant-ph/0204063, 2002.

[ATVY00] Dorit Aharonov, Amnon Ta-Shma, Umesh Vazirani, and Andrew Chi-Chih Yao. Quantum bit escrow. In *Proceedings of 32nd Annual ACM Symposium on the Theory of Computing*, pages 705–714. ACM, 2000.

[BB84]   Charles Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179. IEEE Computer Society, 1984.

[Blu81]   Manuel Blum. Coin flipping by telephone. In Allen Gersho, editor, *Advances in Cryptology: A Report on CRYPTO 81, CRYPTO 81, IEEE Workshop on Communications Security, Santa Barbara, California, USA, August 24-26, 1981*, pages 11–15. U. C. Santa Barbara, Dept. of Elec. and Computer Eng., ECE Report No. 82-04, 1982, 1981.

[BV97]   Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997.

[CK09]   André Chailloux and Iordanis Kerenidis. Optimal quantum strong coin flipping. In *Proceedings of 50th IEEE Symposium on Foundations of Computer Science*, pages 527–533. IEEE Computer Society, 2009.

[CK11]   André Chailloux and Iordanis Kerenidis. Optimal bounds for quantum bit commitment. In *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 354–362. IEEE Computer Society Press, October 2011.

[FvdG99]   Christopher A. Fuchs and Jeroen van de Graaf. Cryptographic distinguishability measures for quantum mechanical states. *IEEE Transactions on Information Theory*, 45:1216–1227, 1999.

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

[Kit02]   Alexei Kitaev. Quantum coin-flipping. Unpublished result. Talk in the 6th Annual workshop on Quantum Information Processing, QIP 2003, Berkeley, CA, USA, December 2002, 2002.

[KN04]   Iordanis Kerenidis and Ashwin Nayak. Weak coin flipping with small bias. *Information Processing Letters*, 89(3):131–135, 2004.

[LC97]   Hoi-Kwong Lo and Hoi Fung Chau. Is quantum bit commitment really possible? *Physical Review Letters*, 78(17):3410–3413, 1997.

[LC98]   Hoi-Kwong Lo and Hoi Fung Chau. Why quantum bit commitment and ideal quantum coin tossing are impossible. *Physica D: Nonlinear Phenomena*, 120(1–2):177–187, September 1998. Proceedings of the Fourth Workshop on Physics and Consumption.

[LC99]   Hoi-Kwong Lo and Hoi Fung Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *Science*, 283:2050–2056, 1999.

[May97]   Dominic Mayers. Unconditionally secure quantum bit commitment is impossible. *Physical Review Letters*, 78(17):3414–3417, 1997.

[May01]   Dominic Mayers. Unconditional security in quantum cryptography. *Journal of the ACM*, 48(3):351–406, 2001.

[Mit03]   Hans D. Mittelmann. An independent benchmarking of SDP and SOCP solvers. Computational semidefinite and second order cone programming: the state of the art. *Mathematical Programming*, 95(2):407–430, 2003.

[Moc04]   Carlos Mochon. Quantum weak coin-flipping with bias of 0.192. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 2–11. IEEE Computer Society, 2004.

[Moc05]   Carlos Mochon. A large family of quantum weak coin-flipping protocols. *Physical Review A*, 72(2):022341, 2005.

[Moc07]   Carlos Mochon. Quantum weak coin flipping with arbitrarily small bias. Available as arXiv.org e-Print quant-ph/0711.4114, 2007.

[MVW12]   Abel Molina, Thomas Vidick, and John Watrous. Optimal counterfeiting attacks and generalizations for Wiesner's quantum money. In *Proceedings of the 7th Conference on Theory of Quantum Computation, Communication, and Cryptography*, pages 45–64, 2012.

[NC00]   Michael Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, New York, NY, USA, 2000.

[NN94]   Yurii Nesterov and Arkadii Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[NS03]   Ashwin Nayak and Peter W. Shor. On bit-commitment based quantum coin flipping. *Physical Review A*, 67(1):012304, 2003.

[PS00]   John Preskill and Peter W. Shor. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85(2):441–444, 2000.

[Sho97]   Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, October 1997.

[SR01]   Robert W. Spekkens and Terence Rudolph. Degrees of concealment and bind-
           ingness in quantum bit commitment protocols. *Physical Review A*, 65:012310,
           2001.

[Stu99]   Jos F. Sturm. Using SeDuMi 1.02, a MATLAB toolbox for optimization over
           symmetric cones. *Optimization Methods and Software*, 11:625–653, 1999.

[Stu02]   Jos F. Sturm. Implementation of interior point methods for mixed semidefi-
           nite and second order cone optimization problems. *Optimization Methods and
           Software*, 17(6):1105–1154, 2002.

[TW12]   Levent Tunçel and Henry Wolkowicz. Strong duality and minimal representa-
           tions for cone optimization. *Computational Optimization and Applications*, pages
           1–30, 2012.

[Wie83]   Stephen Wiesner. Conjugate coding. *SIGACT News*, 15(1):78–88, January 1983.

[WSV00]   Henry Wolkowicz, Romesh Saigal, and Lieven Vandenberghe, editors. *Hand-
           book of Semidefinite Programming*. Kluwer Academic Publishers, 2000.

[Yao79]   Andrew Chi-Chih Yao. Some complexity questions related to distributive
           computing. In *Proceedings of the Eleventh Annual ACM Symposium on Theory
           of Computing*, STOC '79, pages 209–213, New York, NY, USA, 1979. ACM.

[Yao93]   Andrew Chi-Chih Yao. Quantum circuit complexity. In *Proceedings of the 34th
           Annual IEEE Symposium on Foundations of Computer Science*, pages 352–361, Los
           Alamitos, CA, USA, 1993. IEEE Computer Society Press.

A   SDP characterization of cheating strategies

In this section, we present proofs for Lemmas 4.1 and 4.2, originally due to Kitaev.

Proof of Lemma 4.1:   The matrix constraints in the SDP may readily be rewritten as linear
constraints on the variables ρ_j, so the optimization problem is an SDP. The variables are
the density matrices of qubits under Alice's control after each of Bob's messages. The
partial trace is trace-preserving, so any feasible solution satisfies

$$\mathrm{Tr}(\rho_F) = \mathrm{Tr}(\rho_n) = \cdots = \mathrm{Tr}(\rho_1) = \mathrm{Tr}(\psi\psi^*) = 1.$$

Since $\rho_1, \ldots, \rho_n, \rho_F$ are constrained to be positive semidefinite, they are quantum states.

Bob sends the B_1 qubits to Alice replacing the A_1 part already sent to him. Being the
density matrix Alice has after Bob's first message, ρ_1 satisfies

$$\mathrm{Tr}_{B_1}(\rho_1) = \mathrm{Tr}_{A_1}(\psi\psi^*),$$

74

since the state of the qubits other than those in A_1, B_1 remains unchanged. Similarly, we have the constraint

$$\mathrm{Tr}_{B_j}(\rho_j) = \mathrm{Tr}_{A_j}(\rho_{j-1}), \quad \text{for} \quad j \in \{2, \ldots, n\},$$

for each ρ_j after Bob's j'th message. Also ρ_F, the state Alice has at the end of the protocol, satisfies

$$\mathrm{Tr}_{B' \times B_0'}(\rho_F) = \mathrm{Tr}_{A' \times A_0'}(\rho_n).$$

She then measures ρ_F and accepts c with probability $\langle \rho_F, \Pi_{A,c} \rangle$.

These constraints are necessary conditions on the states under Alice's control. We may further restrict the states to be real matrices: the real parts of any complex feasible solution also form a feasible solution with the same objective function value.

We now show that every feasible solution to the above problem yields a valid cheating strategy for Bob with success probability equal to the objective function value of the feasible solution. He can find such a strategy by maintaining a purification of each density matrix in the feasible solution. For example, suppose the protocol starts in the state $\psi \otimes \phi'$, where $\phi' \in \mathbb{C}^K := \mathbb{C}^{B_0} \otimes \mathbb{C}^{B_0'} \otimes \mathbb{C}^B \otimes \mathbb{C}^{B'} \otimes \mathbb{C}^{K'}$ where $\mathbb{C}^{K'}$ is extra space Bob uses to cheat. Consider $\tau \in \mathbb{C}^{A_0} \otimes \mathbb{C}^{A_0'} \otimes \mathbb{C}^A \otimes \mathbb{C}^{A'} \otimes \mathbb{C}^K$ a purification of ρ_1 and $\eta := \psi \otimes \phi'$ a purification of $\psi\psi^*$. Since $\mathrm{Tr}_{B_1}(\rho_1) = \mathrm{Tr}_{A_1}(\psi\psi^*)$,

$$\mathrm{Tr}_{A_1 \times K}(\tau\tau^*) = \mathrm{Tr}_{B_1}(\rho_1) = \mathrm{Tr}_{A_1}(\psi\psi^*) = \mathrm{Tr}_{A_1 \times K}(\eta\eta^*).$$

Thus, there exists a unitary U which acts on $\mathbb{C}^{A_1} \otimes \mathbb{C}^K$ which maps τ to η. If Bob applies this unitary after Alice's first message and sends the B_1 qubits back then he creates ρ_1 under Alice's control. The same argument can be applied to the remaining constraints.

The states corresponding to honest Bob yield a feasible solution. Attainment of an optimal solution then follows from continuity of the objective function and from the compactness of the feasible region. An optimal solution yields an optimal cheating strategy. $\square$

The characterization of Alice's cheating strategies is almost the same as that for cheating Bob; we only sketch the parts that are different.

Proof of Lemma 4.2: There are two key differences from the proof of Lemma 4.1. One is that Alice sends the first message and Bob sends the last, explaining the slightly different constraints. Secondly, Bob measures only the $\mathbb{C}^{B_0} \otimes \mathbb{C}^{A_0'} \otimes \mathbb{C}^A \otimes \mathbb{C}^{A'}$ part of his state after Alice's last message, i.e., he measures $\mathrm{Tr}_{B_0' \times B'}(\sigma_F)$. Note that the adjoint of the partial trace can be written as $\mathrm{Tr}_{B_0' \times B'}^*(Y) = Y \otimes I_{B_0' \times B'}$. Therefore we have

$$\left\langle \mathrm{Tr}_{B_0' \times B'}(\sigma_F), \Pi_{B,c} \right\rangle = \left\langle \sigma_F, \Pi_{B,c} \otimes I_{B_0' \times B'} \right\rangle,$$

which explains the objective function. $\square$

B Derivations of the reduced cheating strategies

In this appendix, we show the derivation of Alice's reduced cheating strategy (the derivation of Bob's is very similar and the arguments are the same). We show that if we are given an optimal solution to Alice's cheating SDP, then we can assume it has a special form while retaining the same objective function value. Then we show this special form for an optimal solution can be written in the way desired.

Technical lemmas

We now discuss some of the tools used in the proofs in the rest of the appendix.

Lemma B.1 *Suppose A is a finite set. Suppose $p = \sum_{x \in A} p_x\, e_x \otimes e_x \in \mathrm{Prob}^{A \times A}$ and $\sigma \in \mathbb{S}_+^A$ is a density matrix. Then we have*

$$\max_{\rho \in \mathbb{S}_+^{A \times A}} \left\{ \left\langle \sqrt{p}\sqrt{p}^\mathrm{T}, \rho \right\rangle : \mathrm{Tr}_A(\rho) = \sigma \right\} \leq \max_{\rho \in \mathbb{S}_+^{A \times A}} \left\{ \left\langle \sqrt{p}\sqrt{p}^\mathrm{T}, \rho \right\rangle : \mathrm{Tr}_A(\rho) = \mathrm{Diag}(\sigma) \right\},$$

where Diag restricts to the diagonal of a square matrix. Moreover, an optimal solution to the problem on the right is $\overline{\rho} := \sqrt{q}\sqrt{q}^\mathrm{T}$, where $q = \sum_{x \in A}[\sigma]_{x,x}\, e_x \otimes e_x \in \mathrm{Prob}^{A \times A}$, yielding an objective function value of $\mathrm{F}(p, q)$.

Proof: Consider $\overline{\rho}$ as defined in the statement of the lemma. Since $\mathrm{Tr}_A(\overline{\rho}) = \mathrm{Diag}(\sigma)$, it suffices to show that for any density matrix $\rho \in \mathbb{S}_+^{A \times A}$ satisfying either $\mathrm{Tr}_A(\rho) = \sigma$ or $\mathrm{Tr}_A(\rho) = \mathrm{Diag}(\sigma)$, we have $\left\langle \sqrt{p}\sqrt{p}^\mathrm{T}, \rho \right\rangle \leq \left\langle \sqrt{p}\sqrt{p}^\mathrm{T}, \overline{\rho} \right\rangle = \mathrm{F}(p, q)$.

Expanding the first inner product, and using the Cauchy-Schwartz inequality, we get

$$\left\langle \sqrt{p}\sqrt{p}^\mathrm{T}, \rho \right\rangle = \sum_{x,y \in A} \sqrt{p_x p_y}(e_x \otimes e_x)^\mathrm{T} \rho\,(e_y \otimes e_y) \leq \sum_{x,y \in A} \sqrt{p_x p_y}\,\|\sqrt{\rho}\,(e_x \otimes e_x)\| \cdot \|\sqrt{\rho}\,(e_y \otimes e_y)\|.$$

We can simplify this by noting

$$\|\sqrt{\rho}\,(e_x \otimes e_x)\|^2 = (e_x \otimes e_x)^\mathrm{T} \rho\,(e_x \otimes e_x) \leq \sum_{z \in A}(e_z \otimes e_x)^\mathrm{T} \rho\,(e_z \otimes e_x) = e_x^\mathrm{T}\mathrm{Tr}_A(\rho)e_x = [\sigma]_{x,x}$$

implying $\left\langle \sqrt{p}\sqrt{p}^\mathrm{T}, \rho \right\rangle \leq \sum_{x,y \in A}\sqrt{p_x p_y}\,([\sigma]_{x,x}[\sigma]_{y,y})^{\frac{1}{2}} = \left(\sum_{x \in A}\sqrt{p_x[\sigma]_{x,x}}\right)^2 = \mathrm{F}(p, q)$, as desired. $\square$

Definition B.2 *We define the* partial Diag operator *over the subspace $\mathbb{C}^A$, denoted Diag_A, as the operator that projects density matrices over $\mathbb{C}^B \otimes \mathbb{C}^A$ onto the diagonal only on the subspace $\mathbb{C}^A$:*

$$\mathrm{Diag}_A(\rho) \quad = \quad \sum_{x \in A}(\mathrm{I}_B \otimes e_x^\mathrm{T})\,\rho\,(\mathrm{I}_B \otimes e_x) \otimes e_x e_x^\mathrm{T}.$$

We may write Diag_A as the superoperator $\mathbb{I} \otimes \mathrm{Diag}_A$, where $\mathbb{I}$ is the identity superoperator acting on the rest of the space. Similarly, we may write the partial trace over A as the superoperator $\mathrm{Tr}_A := \mathbb{I} \otimes \mathrm{Tr}(\cdot)$ where $\mathrm{Tr}(\cdot)$ acts only on $\mathbb{C}^A$. Using this perspective, we see that the partial trace and the partial Diag operators commute when they act on different subspaces. Also, $\mathrm{Tr}_A \circ \mathrm{Diag}_A = \mathrm{Tr}_A$ since the trace only depends on the diagonal elements.

We also make use of the following lemma.

Lemma B.3 *Consider a matrix $\rho \in \mathbb{S}_+^{A \times B}$. If $\mathrm{Tr}_A(\rho) = \psi\psi^*$ for some vector $\psi \in \mathbb{C}^B$, then ρ can be written as $\rho = \tilde{\rho} \otimes \psi\psi^*$, for some $\tilde{\rho} \in \mathbb{S}_+^A$.*

This is easily proven using the fact that the half-line emanating through a rank one positive semidefinite matrix forms an extreme ray of the cone of positive semidefinite matrices, or more directly by expressing ρ using an orthogonal basis for $\mathbb{C}^B$ that includes ψ.

Derivations of Alice's reduced cheating strategy

Assume $(\sigma_1, \sigma_2, \ldots, \sigma_n, \sigma_F)$ is optimal for Alice's cheating SDP. We now define

$$(\sigma_1', \sigma_2', \ldots, \sigma_n', \sigma_F') = (\sigma_1, \mathrm{Diag}_{B_1'}(\sigma_2), \ldots, \mathrm{Diag}_{B_1' \times \cdots \times B_{n-1}'}(\sigma_n), \mathrm{Diag}_{B' \times A_0'}(\sigma_F))$$

and show it is also optimal. All we need to show is feasibility since the objective function value is preserved because $\Pi_{\mathrm{B},c} \otimes I_{B_0' \times B'}$ is diagonal in the space $\mathbb{S}_+^{B' \times A_0'}$.

The first constraint is satisfied since $\sigma_1' = \sigma_1$ is part of a feasible solution. From Lemma B.3, we can write $\sigma_1' = \phi\phi^* \otimes \tilde{\sigma}_1$ for some $\tilde{\sigma}_1 \in \mathbb{S}_+^{A_1}$. We can write

$$\mathrm{Tr}_{B_1}(\sigma_1') = \sum_{y_1 \in B_1'} e_{y_1} e_{y_1}^* \otimes \phi_{y_1} \phi_{y_1}^* \otimes \tilde{\sigma}_1,$$

where $\phi_{y_1, \ldots, y_j} := \sum_{b \in B_0} \sum_{y_{j+1} \in B_{j+1}'} \cdots \sum_{y_n \in B_n'} \frac{1}{\sqrt{2}} \sqrt{\beta_{b,y}}\, e_b \otimes e_b \otimes e_{y_{j+1}} \otimes e_{y_{j+1}} \otimes \cdots \otimes e_{y_n} \otimes e_{y_n}$,

which is in $\mathbb{C}^{B_0 \times B_0' \times B_{j+1} \times B_{j+1}' \times \cdots \times B_n \times B_n'}$. Therefore, $\mathrm{Tr}_{B_1}(\sigma_1')$ is diagonal in B_1' and

$$\mathrm{Tr}_{B_1}(\sigma_1') = \mathrm{Diag}_{B_1'}(\mathrm{Tr}_{B_1}(\sigma_1')) = \mathrm{Diag}_{B_1'}(\mathrm{Tr}_{B_1}(\sigma_1)) = \mathrm{Diag}_{B_1'}(\mathrm{Tr}_{A_2}(\sigma_2)) = \mathrm{Tr}_{A_2}(\sigma_2'). \quad (18)$$

Therefore, the second constraint is satisfied. Since σ_2' is diagonal in B_1' we can write it as

$$\sigma_2' = \sum_{y_1 \in B_1'} e_{y_1} e_{y_1}^* \otimes \sigma_{2,y_1}, \quad \text{for some} \quad \sigma_{2,y_1} \in \mathbb{S}_+^{B_0 \times B_0' \times A_1 \times A_2 \times B_2 \times \cdots \times B_n \times B_2' \times \cdots \times B_n'}.$$

By feasibility, $\mathrm{Tr}_{A_2}(\sigma_2') = \sum_{y_1 \in B_1'} e_{y_1} e_{y_1}^* \otimes \mathrm{Tr}_{A_2}(\sigma_{2,y_1}) = \mathrm{Tr}_{B_1}(\sigma_1') = \sum_{y_1 \in B_1'} e_{y_1} e_{y_1}^* \otimes \phi_{y_1} \phi_{y_1}^* \otimes \tilde{\sigma}_1$,

therefore $\sigma_2' = \sum_{y_1 \in B_1'} e_{y_1} e_{y_1}^* \otimes \phi_{y_1} \phi_{y_1}^* \otimes \tilde{\sigma}_{2,y_1}$, where $\tilde{\sigma}_{2,y_1} \in \mathbb{S}_+^{B_0 \times B_0' \times A_1 \times A_2}$ satisfies

77

$\text{Tr}_{A_2}(\tilde{\sigma}_{2,y_1}) = \tilde{\sigma}_1$ for all $y_1 \in B_1'$. Using similar arguments, we may show that the rest of the first n constraints are satisfied. For every $j \in \{3, \ldots, n\}$, we have

$$\sigma_j' = \sum_{y_1 \in B_1'} \cdots \sum_{y_{j-1} \in B_{j-1}'} e_{y_1} e_{y_1}^* \otimes \cdots \otimes e_{y_{j-1}} e_{y_{j-1}}^* \otimes \phi_{y_1,\ldots,y_{j-1}} \phi_{y_1,\ldots,y_{j-1}}^* \otimes \tilde{\sigma}_{j,y_1,\ldots,y_{j-1}},$$

where $\tilde{\sigma}_{j,y_1,\ldots,y_{j-1}} \in \mathbb{S}_+^{B_0 \times B_0' \times A_1 \times \cdots \times A_j}$ satisfies $\text{Tr}_{A_j}(\tilde{\sigma}_{j,y_1,\ldots,y_{j-1}}) = \tilde{\sigma}_{j-1,y_1,\ldots,y_{j-2}}$ for all $y_1 \in B_1', \ldots, y_{j-1} \in B_{n-1}'$. Note that $\text{Tr}_{B_n}(\sigma_n') = \sum_{y \in B'} e_y e_y^* \otimes \phi_y \phi_y^* \otimes \tilde{\sigma}_{n,y_1,\ldots,y_{n-1}}$ which is helpful in proving feasibility of the last constraint. For the last constraint, we can use a similar reduction as in Equation (18) to show $\text{Tr}_{A' \times A_0'}(\sigma_F') = \text{Tr}_{B_n}(\sigma_n')$ proving $(\sigma_1', \ldots, \sigma_n', \sigma_F')$ is feasible. We now use this feasible solution to simplify the problem.

We can clean up σ_F' by noting that it is diagonal in $\mathbb{C}^{B'}$ and $\mathbb{C}^{A_0'}$ and write it as

$$\sigma_F' = \sum_{a \in A_0'} \sum_{y \in B'} e_a e_a^* \otimes e_y e_y^* \otimes \sigma_{F,a,y}, \quad \text{for some} \quad \sigma_{F,a,y} \in \mathbb{S}_+^{B_0 \times B_0' \times A \times A'}.$$

Thus, $\text{Tr}_{A' \times A_0'}(\sigma_F') = \sum_{a \in A_0'} \sum_{y \in B'} e_y e_y^* \otimes \text{Tr}_{A'}(\sigma_{F,a,y}) = \sum_{y \in B'} e_y e_y^* \otimes \left(\sum_{a \in A_0'} \text{Tr}_{A'}(\sigma_{F,a,y}) \right)$. Similarly, by feasibility, we have $\text{Tr}_{A' \times A_0'}(\sigma_F') = \text{Tr}_{B_n}(\sigma_n') = \sum_{y \in B'} e_y e_y^* \otimes \phi_y \phi_y^* \otimes \sigma_{n,y_1,\ldots,y_{n-1}}$. Thus,

$$\sigma_F' = \sum_{a \in A_0'} \sum_{y \in B'} e_a e_a^* \otimes e_y e_y^* \otimes \phi_y \phi_y^* \otimes \tilde{\sigma}_{F,a,y},$$

by writing $\sigma_{F,a,y} = \phi_y \phi_y^* \otimes \tilde{\sigma}_{F,a,y}$ where $\tilde{\sigma}_{F,a,y} \in \mathbb{S}_+^{A \times A'}$ satisfies $\sum_{a \in A_0'} \text{Tr}_{A'}(\tilde{\sigma}_{F,a,y}) = \sigma_{n,y_1,\ldots,y_{n-1}}$

for all $a \in A_0'$ and $y \in B'$.

The objective function becomes $\left\langle \sigma_F', \Pi_{B,0} \otimes I_{B_0' \times B'} \right\rangle = \frac{1}{2} \sum_{a \in A_0'} \sum_{y \in B'} \beta_{a,y} \langle \tilde{\sigma}_{F,a,y}, \psi_a \psi_a^* \rangle$.

At this point, we note that $\left\langle \sigma_F', \Pi_{B,1} \otimes I_{B_0' \times B'} \right\rangle = \frac{1}{2} \sum_{a \in A_0'} \sum_{y \in B'} \beta_{\bar{a},y} \langle \tilde{\sigma}_{F,a,y}, \psi_a \psi_a^* \rangle$, proving that evaluating Alice's success probability of cheating towards 0 or 1 with this strategy is a matter of switching Bob's two probability distributions.

Carrying on with $P_{A,0}^*$, we get the following SDP

$$\sup \quad \frac{1}{2} \sum_{a \in A_0', \, y \in B'} \beta_{a,y} \, \langle \tilde{\sigma}_{F,a,y}, \psi_a \psi_a^* \rangle$$

$$\begin{aligned}
\text{subject to} \qquad \operatorname{Tr}_{A_1}(\tilde{\sigma}_1) &= 1, \\
\operatorname{Tr}_{A_j}(\tilde{\sigma}_{j,y_1,\dots,y_{j-1}}) &= \tilde{\sigma}_{j-1,y_1,\dots,y_{j-2}}, \quad \forall j \in \{2,\dots,n\}, \\
& \qquad \qquad \forall y_1 \in B_1', \\
& \qquad \qquad \vdots \\
& \qquad \qquad \forall y_{j-1} \in B_{j-1}', \\
\sum_{a \in A_0'} \operatorname{Tr}_{A'}(\tilde{\sigma}_{F,a,y}) &= \tilde{\sigma}_{n,y_1,\dots,y_{n-1}}, \quad \forall y \in B', \\
\tilde{\sigma}_{j,y_1,\dots,y_{j-1}} &\in \mathbb{S}_+^{A_1 \times \cdots \times A_j}, \quad \forall j \in \{1,\dots,n\}, \\
& \qquad \qquad \forall y_1 \in B_1', \\
& \qquad \qquad \vdots \\
& \qquad \qquad \forall y_{j-1} \in B_{j-1}', \\
\tilde{\sigma}_{F,a,y} &\in \mathbb{S}_+^{A' \times A}, \qquad \forall a \in A_0', y \in B'.
\end{aligned}$$

By Lemma B.1, the following restrictions can only improve the objective function value:

$$\begin{aligned}
s_1 &:= \operatorname{diag}(\tilde{\sigma}_1), \\
s_2^{(y_1)} &:= \operatorname{diag}(\tilde{\sigma}_{2,y_1}), \qquad \forall y_1 \in B_1', \\
& \qquad \vdots \\
s_n^{(y_1,\dots,y_{n-1})} &:= \operatorname{diag}(\tilde{\sigma}_{n,y_1,\dots,y_{n-1}}), \qquad \forall y_1 \in B_1', \dots, y_{n-1} \in B_{n-1}', \\
s^{(a,y)} &:= \operatorname{diag}(\operatorname{Tr}_{A'}(\tilde{\sigma}_{F,a,y})), \qquad \forall a \in A_0', \, y \in B', \\
\operatorname{Tr}_{A'}(\tilde{\sigma}_{F,a,y}) &= \operatorname{Diag}(s^{(a,y)}), \qquad \forall a \in A_0', \, y \in B',
\end{aligned}$$

where the superscripts are the restrictions of the vectors as before. With these new variables, and using Lemma B.1, we can write the new objective function as

$$\frac{1}{2} \sum_{a \in \{0,1\}} \sum_{y \in B} \beta_{a,y} \, \mathrm{F}(s^{(a,y)}, \alpha_a),$$

where $(s_1, \dots, s_n, s) \in \mathcal{P}_A$. Any feasible solution to the reduced SDP also gives us a feasible solution to the original SDP, so their optimal values are equal. $\square$

This proof shows that the reduced cheating problem does not eliminate all of the optimal solutions of the corresponding SDP. We can also show that the reduced problems capture optimal solutions to the corresponding SDPs by examining the dual SDPs. However, the primal SDPs are more important for the purposes of this paper and this proof is more illustrative.

C Developing the strategies in the filter

In this appendix, we prove Theorems 5.2 and 5.3 by developing the cheating strategies used in the filter for Alice and Bob.

Cheating Alice

Recall Alice's optimization problem

$$P_{A,0}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \sum_{y \in B} \beta_{a,y} \, \mathrm{F}(s^{(a,y)}, \alpha_a) \; : \; (s_1, \ldots, s_n, s) \in \mathcal{P}_A \right\}.$$

To get a feasible solution, suppose Alice guesses b before she reveals a in the following way. If Bob reveals $y \in B$, then Alice guesses $b = 0$ if $\beta_{0,y} \geq \beta_{1,y}$ and $b = 1$ if $\beta_{0,y} < \beta_{1,y}$. Let Alice's guess be denoted by $f(y)$, so

$$f(y) = \arg \max_a \{\beta_{a,y}\} \in \{0, 1\},$$

and we set $f(y) = 0$ in the case of a tie. We have chosen a way to satisfy the last constraint in Alice's cheating polytope, but we can choose how Alice sends her first n messages $s_1, \ldots, s_n$. We make one more restriction, we set $s_n = d \otimes e_{B_1 \times \cdots \times B_{n-1}}$ and optimize over $d \in \mathrm{Prob}^A$. We can easily satisfy the rest of the constraints given any d by choosing each variable as the corresponding marginal probability distribution.

Under these restrictions, we have that Alice's reduced problem can be written as

$$\max_{d \in \mathrm{Prob}^A} \left\{ \frac{1}{2} \sum_{y \in B} \beta_{f(y),y} \mathrm{F}(d, \alpha_{f(y)}) \right\} = \max_{d \in \mathrm{Prob}^A} \{\eta \, \mathrm{F}(d, \alpha_0) + \tau \, \mathrm{F}(d, \alpha_1)\}.$$

We can simplify this using the following lemma.

Lemma C.1 *For nonnegative vectors $\{z_1, \ldots, z_n\} \subset \mathbb{R}_+^n$, we have that*

$$\max \left\{ \sum_{i=1}^n \mathrm{F}(p, z_i) : p \in \mathrm{Prob}^n \right\} = \lambda_{\max} \left(\sum_{i=1}^n \sqrt{z_i} \sqrt{z_i}^{\mathrm{T}} \right).$$

Furthermore, an optimal solution is the entry-wise square of the normalized principal eigenvector.

Proof: Since $\sum_{i=1}^n \mathrm{F}(p, z_i) = \sum_{i=1}^n \left\langle \sqrt{p} \sqrt{p}^{\mathrm{T}}, \sqrt{z_i} \sqrt{z_i}^{\mathrm{T}} \right\rangle = \sqrt{p}^{\mathrm{T}} \left(\sum_{i=1}^n \sqrt{z_i} \sqrt{z_i}^{\mathrm{T}} \right) \sqrt{p}$, where $\sqrt{\cdot}$ is the entry-wise square root, the maximization problem reduces to

$$\max \left\{ \sqrt{p}^{\mathrm{T}} \left(\sum_{i=1}^n \sqrt{z_i} \sqrt{z_i}^{\mathrm{T}} \right) \sqrt{p} : p \in \mathrm{Prob}^n \right\}.$$

Let $\hat{x} \in \mathbb{R}^m$ be the restriction of a vector x onto $\cup_{i=1}^n \operatorname{supp}(z_i)$. Then the optimal objective value of the above optimization problem is equal to that of

$$\max\left\{ \sqrt{\hat{p}}^{\mathrm{T}} \left(\sum_{i=1}^n \sqrt{\hat{z}_i}\sqrt{\hat{z}_i}^{\mathrm{T}} \right) \sqrt{\hat{p}} : \hat{p} \in \operatorname{Prob}^{\cup_{i=1}^n \operatorname{supp}(z_i)} \right\}.$$

If the nonnegativity constraint were not present, the optimum value would be attained by setting $\sqrt{\hat{p}}$ to be the normalized principal eigenvector of the matrix $\sum_{i=1}^n \sqrt{\hat{z}_i}\sqrt{\hat{z}_i}^{\mathrm{T}}$. Because $\sum_{i=1}^n \sqrt{\hat{z}_i}\sqrt{\hat{z}_i}^{\mathrm{T}}$ has positive entries, we know the principal eigenvector is also positive by the Perron-Frobenius Theorem. Since this does not violate the nonnegativity constraint in the problem, $\hat{p}$, where $\sqrt{\hat{p}}$ is the normalized principal eigenvector, is an optimal solution yielding an optimal objective value of $\lambda_{\max}\left(\sum_{i=1}^n \sqrt{\hat{z}_i}\sqrt{\hat{z}_i}^{\mathrm{T}} \right)$. Notice that $\sum_{i=1}^n \sqrt{\hat{z}_i}\sqrt{\hat{z}_i}^{\mathrm{T}}$ is the matrix obtained by removing the zero rows and columns from $\sum_{i=1}^n \sqrt{z_i}\sqrt{z_i}^{\mathrm{T}}$ and thus has the same largest eigenvalue. $\quad\square$

Using this lemma, Alice can cheat with probability

$$\frac{1}{2}\lambda_{\max}\left(\eta\sqrt{\alpha_0}\sqrt{\alpha_0}^{\mathrm{T}} + \tau\sqrt{\alpha_1}\sqrt{\alpha_1}^{\mathrm{T}} \right),$$

which we call Alice's *eigenstrategy*.

We can find a lower bound on this value using the following two lemmas.

Lemma C.2 *For β_0, β_1, η, and τ defined above, we have $\eta + \tau = 1 + \Delta(\beta_0, \beta_1)$.*

Proof: Notice that we can write $\sum_{y \in B} \max_{a \in \{0,1\}} \{\beta_{a,y}\} + \sum_{y \in B} \min_{a \in \{0,1\}} \{\beta_{a,y}\} = 2$ and we can also write $\sum_{y \in B} \max_{a \in \{0,1\}} \{\beta_{a,y}\} - \sum_{y \in B} \min_{a \in \{0,1\}} \{\beta_{a,y}\} = 2\Delta(\beta_0, \beta_1)$. With this, we can conclude that $\eta + \tau = \sum_{y \in B} \max_{a \in \{0,1\}} \{\beta_{a,y}\} = 1 + \Delta(\beta_0, \beta_1)$, as desired. $\quad\square$

The above lemma can be restated as $\sum_{y \in B} \max_{a \in \{0,1\}} \{\beta_{a,y}\} = 1 + \Delta(\beta_0, \beta_1)$ for any probability distributions β_0 and β_1. This is helpful when looking at Bob's cheating strategies as well.

Lemma C.3 *For $\eta, \tau \in \mathbb{R}$ and $p, q \in \operatorname{Prob}^n$, we have*

$$\lambda_{\max}\left(\eta\sqrt{p}\sqrt{p}^{\mathrm{T}} + \tau\sqrt{q}\sqrt{q}^{\mathrm{T}} \right) = \frac{1}{2}\left(\eta + \tau + \sqrt{(\eta - \tau)^2 + 4\eta\tau\,\mathrm{F}(p,q)} \right).$$

Proof: Since we can write $\mathrm{F}(p,q) = \left(\sqrt{p}^{\mathrm{T}}\sqrt{q}\right)^2$, we can apply a unitary to both $\sqrt{p}$ and $\sqrt{q}$ and both sides of the equality we want to prove are unaffected. Choose a unitary U such that

$$U\sqrt{p} = [1,0,0,\ldots,0]^{\mathrm{T}} \quad \text{and} \quad U\sqrt{q} = [\sin\theta, \cos\theta, 0, \ldots, 0]^{\mathrm{T}},$$

for some $\theta \in [0, 2\pi)$. Then we can write $\mathrm{F}(p,q) = \sin^2\theta$. Let $\lambda_{\max}$ be the largest eigenvalue of $\eta\sqrt{p}\sqrt{p}^{\mathrm{T}} + \tau\sqrt{q}\sqrt{q}^{\mathrm{T}}$, or equivalently, of $\eta U\sqrt{p}\sqrt{p}^{\mathrm{T}}U^* + \tau U\sqrt{q}\sqrt{q}^{\mathrm{T}}U^*$, and let λ_2 be the second largest eigenvalue. Then

$$\lambda_{\max} + \lambda_2 = \mathrm{Tr}(\eta\sqrt{p}\sqrt{p}^{\mathrm{T}} + \tau\sqrt{q}\sqrt{q}^{\mathrm{T}}) = \eta + \tau$$

and, by taking the determinant of the only nonzero block, we get

$$\lambda_{\max} \cdot \lambda_2 = \eta\tau\cos^2\theta = \eta\tau(1 - \mathrm{F}(p,q))$$

implying $\lambda_{\max} = \frac{1}{2}\left(\eta + \tau + \sqrt{(\eta - \tau)^2 + 4\eta\tau\mathrm{F}(p,q)}\right)$, as desired. $\square$

Note that Lemma C.3 shows that switching the roles of η and τ does not affect the largest eigenvalue.

Using the above two lemmas, we have

$$
\begin{aligned}
\frac{1}{2}&\lambda_{\max}\left(\eta\sqrt{\alpha_0}\sqrt{\alpha_0}^{\mathrm{T}} + \tau\sqrt{\alpha_1}\sqrt{\alpha_1}^{\mathrm{T}}\right) \\
&= \frac{1}{4}\left(\eta + \tau + \sqrt{(\eta - \tau)^2 + 4\eta\tau\,\mathrm{F}(\alpha_0, \alpha_1)}\right) \\
&\geq \frac{1}{4}\left(\eta + \tau + \sqrt{(\eta - \tau)^2\,\mathrm{F}(\alpha_0, \alpha_1) + 4\eta\tau\,\mathrm{F}(\alpha_0, \alpha_1)}\right) \\
&= \frac{1}{4}\left(\left(1 + \sqrt{\mathrm{F}(\alpha_0, \alpha_1)}\right)(\eta + \tau)\right) \\
&= \left(\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\alpha_0, \alpha_1)}\right)\left(\frac{1}{2} + \frac{1}{2}\Delta(\beta_0, \beta_1)\right).
\end{aligned}
$$

This lower bound has a natural interpretation. This is the strategy where Alice ignores all of Bob's messages until $\mathbb{C}^{B_n}$ is sent. Then she measures it to learn b with probability $\frac{1}{2} + \frac{1}{2}\Delta(\beta_0, \beta_1)$. Conditioned on having the correct value for b, she tries to get past Bob's cheat detection and can do so with probability $\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\alpha_0, \alpha_1)}$. We call this Alice's *three-round strategy* since it combines optimal strategies for the three-round protocol example in Subsection 3.2. It makes sense that this is a lower bound on the success probability of Alice's eigenstrategy since her eigenstrategy is optimized from the same restrictions that apply to her three-round strategy.

We can also examine how Alice can choose her last message optimally supposing she has already sent her first n messages in a particular way. I.e., suppose $s_n := c \otimes e_{B_1 \times \cdots \times B_{n-1}}$ for some $c \in \mathrm{Prob}^A$ (as in the eigenstrategy). From this we can find $s_1, \ldots, s_{n-1}$ satisfying

the first $n-1$ constraints of her cheating polytope by taking the corresponding marginal distributions of c. We want to optimize over s satisfying $\mathrm{Tr}_{A_0'}(s) = s_n \otimes e_{B_n} = c \otimes e_B$. In this case, this constraint can be written as $\sum_{a \in \{0,1\}} s^{(a,y)} = c$, for each $y \in B$, where again, $s^{(a,y)}$ is the restriction of s with a and y fixed. Now we get the following optimization problem

$$
\begin{array}{rrcl}
\max & \frac{1}{2} \sum_{a \in \{0,1\}} \sum_{y \in B} \beta_{a,y}\, \mathrm{F}(s^{(a,y)}, \alpha_a) & & \\
\text{subject to} & \sum_{a \in \{0,1\}} s^{(a,y)} & = & c, \quad \text{for all } y \in B, \\
& s^{(a,y)} & \geq & 0,
\end{array}
$$

where c is now constant. If we rewrite this as

$$
\begin{array}{rrcl}
\max & \frac{1}{2} \sum_{y \in B} \sum_{a \in \{0,1\}} \mathrm{F}(s^{(a,y)}, \beta_{a,y}\alpha_a) & & \\
\text{subject to} & \sum_{a \in \{0,1\}} s^{(a,y)} & = & c, \quad \text{for all } y \in B, \\
& s^{(a,y)} & \geq & 0,
\end{array}
$$

we have a separable problem over $y \in B$. That is, for each fixed $\tilde{y} \in B$, Alice needs to solve the optimization problem

$$
G_{\tilde{y}}(c) := \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \mathrm{F}(s^{(a,\tilde{y})}, \beta_{a,\tilde{y}}\alpha_a) : \sum_{a \in \{0,1\}} s^{(a,\tilde{y})} = c,\ s^{(a,\tilde{y})} \geq 0,\ \forall a \in \{0,1\} \right\}.
$$

This optimization problem has a special structure.

Definition C.4 *The* infimal convolution *of the convex functions $f_1, f_2, \ldots, f_n$, where $f_1, \ldots, f_n : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$, is*

$$
(f_1 \Box f_2 \Box \cdots \Box f_n)(d) := \inf_{x_1, \ldots, x_n \in \mathbb{R}^m} \left\{ \sum_{i=1}^n f_i(x_i) : \sum_{i=1}^n x_i = d \right\}.
$$

We do not need to worry about the nonnegativity constraints on the variables since we can define our convex function $-\mathrm{F}(p,q) = +\infty$ if p or q is not nonnegative. Note for every $p \in \mathbb{R}_+^m$, that $-\mathrm{F}(p, \cdot)$ is a *proper, convex function*, i.e., it is convex and $-\mathrm{F}(p,q) < +\infty$ for some $q \in \mathbb{R}_+^m$ and $-\mathrm{F}(p,q) > -\infty$ for every $q \in \mathbb{R}_+^m$. Proper, convex functions have many useful properties as detailed in this section. Using these properties and the fact that $-\mathrm{F}(p, \cdot)$ is positively homogeneous, we show a way to express $G_{\tilde{y}}$.

Recall that for proper, convex functions $f_1, \ldots, f_n : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$, the convex hull of $\{f_1, \ldots, f_n\}$ is the greatest convex function f such that $f(x) \leq f_1(x), \ldots, f_n(x)$ for every $x \in \mathbb{R}^m$. To write down explicitly what the convex hull is, we use the following theorem.

Theorem C.5 ([Roc70, page 37]) *Let $f_1, \ldots, f_n : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ be proper, convex functions. Then we have*

$$\text{conv} \{f_1, \ldots, f_n\}(d) = \inf \left\{ \sum_{i=1}^{n} \lambda_i f_i(x_i) : \sum_{i=1}^{n} \lambda_i x_i = d \right\}.$$

For a positively homogeneous function f, we have $\lambda f\left(\lambda^{-1} x\right) = f(x)$, for $\lambda > 0$. Therefore, we have the following corollary.

Corollary C.6 *Let $f_1, \ldots, f_n : \mathbb{R}^m \to \mathbb{R} \cup \{\infty\}$ be positively homogeneous, proper, convex functions. Then we have*

$$\text{conv} \{f_1, \ldots, f_n\} = f_1 \square f_2 \square \cdots \square f_n.$$

Therefore, we can write Alice's cheating probability using concave hulls as shown below

$$
\begin{aligned}
G_{\tilde{y}}(c) &= \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \text{F}(s^{(a,\tilde{y})}, \beta_{a,\tilde{y}} \alpha_a) : \sum_{a \in \{0,1\}} s^{(a,\tilde{y})} = c, \; s^{(a,\tilde{y})} \geq 0, \; \forall a \in \{0,1\} \right\} \\
&= -\min \left\{ -\frac{1}{2} \sum_{a \in \{0,1\}} \text{F}(s^{(a,\tilde{y})}, \beta_{a,\tilde{y}} \alpha_a) : \sum_{a \in \{0,1\}} s^{(a,\tilde{y})} = c, \; s^{(a,\tilde{y})} \geq 0, \; \forall a \in \{0,1\} \right\} \\
&= -\left(-\frac{1}{2} \text{F}(\cdot, \beta_{0,\tilde{y}} \alpha_0) \right) \square \left(-\frac{1}{2} \text{F}(\cdot, \beta_{1,\tilde{y}} \alpha_1) \right)(c) \\
&= -\text{conv} \left\{ \frac{-1}{2} \beta_{0,\tilde{y}} \text{F}(\cdot, \alpha_0), \frac{-1}{2} \beta_{1,\tilde{y}} \text{F}(\cdot, \alpha_1) \right\}(c) \\
&= \text{conc} \left\{ \frac{1}{2} \beta_{0,\tilde{y}} \text{F}(\cdot, \alpha_0), \frac{1}{2} \beta_{1,\tilde{y}} \text{F}(\cdot, \alpha_1) \right\}(c).
\end{aligned}
$$

Thus, for each $c \in \text{Prob}^A$, we can write Alice's cheating probability as

$$\sum_{y \in B} \text{conc} \left\{ \frac{1}{2} \beta_{0,y} \text{F}(\cdot, \alpha_0), \frac{1}{2} \beta_{1,y} \text{F}(\cdot, \alpha_1) \right\}(c).$$

Note this way of optimizing the last message works for any strategy. For a general strategy, we would have a different c for every $y_1, \ldots, y_{n-1}$.

Thus, we have Alice's *improved eigenstrategy* which is when Alice chooses her first n messages according to her eigenstrategy, yet reveals a optimally.

Cheating Alice in six-round protocols

In six-round protocols, Alice's goal is to maximize the objective function

$$\frac{1}{2} \sum_{a \in \{0,1\}} \sum_{y_1 \in B_1} \sum_{y_2 \in B_2} \beta_{a,y_1 y_2} \mathrm{F}(s^{(a,y_1 y_2)}, \alpha_a)$$

over (s_1, s_2, s) satisfying:

$$\begin{aligned}
\mathrm{Tr}_{A_1}(s_1) &= 1, \\
\mathrm{Tr}_{A_2}(s_2) &= s_1 \otimes e_{B_1}, \\
\mathrm{Tr}_{A_0'}(s) &= s_2 \otimes e_{B_2},
\end{aligned}$$

$$\begin{aligned}
s_1 &\in \mathbb{R}_+^{A_1}, \\
s_2 &\in \mathbb{R}_+^{A_1 \times B_1 \times A_2}, \\
s &\in \mathbb{R}_+^{A_1 \times A_2 \times B_1 \times B_2 \times A_0'}.
\end{aligned}$$

We suppose that Alice chooses her commitment a based on the most likely choice of b after seeing y_1 from Bob's first message. Let $f'(y_1) = \arg\max_{a \in A_0'} \{[\mathrm{Tr}_{B_2}(\beta_a)]_{y_1}\}$ and 0 in the case of a tie. The last constraint can be written as $\sum_{a \in A_0'} s^{(a,y_1 y_2)} = s_2^{(y_1)}$, for all $y_1 \in B_1$, where $s_2^{(y_1)}$ is the projection of s_2 with the index y_1 fixed. We set $s^{(a,y_1,y_2)} = s_2^{(y_1)}$, if $a = f'(y_1)$, and 0 otherwise. Now we set $s_2^{(y_1)} = s_2^0$, if $f'(y_1) = 0$, and $s_2^{(y_1)} = s_2^1$, if $f'(y_1) = 1$, where we optimize $s_2^0, s_2^1 \in \mathbb{R}_+^{A_1 \times A_2}$. The new objective function can be written as

$$\begin{aligned}
\frac{1}{2} \sum_{a \in A_0'} \sum_{y_1 \in B_1, y_2 \in B_2} \beta_{a,y_1 y_2} \mathrm{F}(s^{(a,y_1 y_2)}, \alpha_a) &= \frac{1}{2} \sum_{y_1 \in B_1} \left[\sum_{y_2 \in B_2} \beta_{f'(y_1),y_1 y_2} \right] \mathrm{F}(s_2^{f'(y_1)}, \alpha_{f'(y_1)}) \\
&= \frac{1}{2} \eta' \mathrm{F}(s_2^0, \alpha_0) + \frac{1}{2} \tau' \mathrm{F}(s_2^1, \alpha_1).
\end{aligned}$$

Since the only constraints remaining are $\mathrm{Tr}_{A_2}(s_2^0) = s_1 = \mathrm{Tr}_{A_2}(s_2^1)$, we now optimize over each choice of s_2^0 and s_2^1 separately using the following lemma.

Lemma C.7 *For $\alpha \in \mathbb{R}_+^{A_1 \times A_2}$ and $c \in \mathbb{R}_+^{A_1}$, we have*

$$\max \{\mathrm{F}(p, \alpha) : \mathrm{Tr}_{A_2}(p) = c, \, p \geq 0\} \geq \mathrm{F}(c, \mathrm{Tr}_{A_2}(\alpha)).$$

The inequality can be shown to hold with equality by Uhlmann's theorem. However, we prove the inequality by exhibiting a feasible solution which is also useful for the analysis of cheating Bob.

Proof: For each $x_1 \in A_1$, $x_2 \in A_2$, define p_{x_1,x_2} as

$$
p_{x_1,x_2} := \begin{cases} c_{x_1} \frac{\alpha_{x_1,x_2}}{[\mathrm{Tr}_{A_2}(\alpha)]_{x_1}} & \text{if } [\mathrm{Tr}_{A_2}(\alpha)]_{x_1} > 0, \\[2ex] c_{x_1} \frac{1}{|A_2|} & \text{if } [\mathrm{Tr}_{A_2}(\alpha)]_{x_1} = 0. \end{cases}
$$

Then we have $p \geq 0$ is feasible since $[\mathrm{Tr}_{A_2}(p)]_{x_1} = c_{x_1}$ and it has objective function value $\mathrm{F}(p,\alpha) = \mathrm{F}(c, \mathrm{Tr}_{A_2}(\alpha))$, as desired. $\quad\square$

Using the lemma, we can write the problem as

$$
\max_{c \in \mathrm{Prob}^{A_1}} \eta' \, \mathrm{F}(c, \mathrm{Tr}_{A_2}(\alpha_0)) + \tau' \, \mathrm{F}(c, \mathrm{Tr}_{A_2}(\alpha_1))
$$

which has optimal value $\frac{1}{2}\lambda_{\max}\left(\eta' \sqrt{\mathrm{Tr}_{A_2}(\alpha_0)} \sqrt{\mathrm{Tr}_{A_2}(\alpha_0)}^{\mathrm{T}} + \tau' \sqrt{\mathrm{Tr}_{A_2}(\alpha_1)} \sqrt{\mathrm{Tr}_{A_2}(\alpha_1)}^{\mathrm{T}} \right)$ and is lower bounded by $\left(\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\mathrm{Tr}_{A_2}(\alpha_0), \mathrm{Tr}_{A_2}(\alpha_1))} \right) \left(\frac{1}{2} + \frac{1}{2}\Delta(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1)) \right)$. Again, this last quantity has context. This is the strategy where Alice measures the first message to learn b early and then tries to change the value of a. She can learn b with probability $\frac{1}{2} + \frac{1}{2}\Delta(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1))$. She can successfully change the value of a with probability $\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\mathrm{Tr}_{A_2}(\alpha_0), \mathrm{Tr}_{A_2}(\alpha_1))}$. Thus, she can cheat with probability at least

$$
\left(\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\mathrm{Tr}_{A_2}(\alpha_0), \mathrm{Tr}_{A_2}(\alpha_1))} \right) \left(\frac{1}{2} + \frac{1}{2}\Delta(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1)) \right).
$$

Cheating Bob

Bob's returning strategy is to send Alice's messages right back to her (if the dimensions agree). This way, the state that Alice checks at the end of the protocol is her own state. This is a good strategy when Alice and Bob share the same starting states, i.e., for a protocol with parameters $\alpha_0 = \beta_0$ and $\alpha_1 = \beta_1$. To calculate the cheating probability of this strategy, for any choice of parameters, it is easier to use the original cheating SDP as opposed to the reduced cheating SDP. This cheating strategy corresponds to the feasible solution

$$
\bar{\rho}_1 = \bar{\rho}_2 = \cdots = \bar{\rho}_n = \bar{\rho}_F = \psi\psi^*
$$

which has success probability given by the objective function value

$$
\langle \bar{\rho}_F, \Pi_{A,0} \rangle = \langle \psi\psi^*, \Pi_{A,0} \rangle = \frac{1}{2} \sum_{a \in \{0,1\}} \mathrm{F}(\alpha_a, \beta_a).
$$

This is clearly optimal when $\alpha_0 = \beta_0$ and $\alpha_1 = \beta_1$.

Recall Bob's reduced problem below

$$P_{B,0}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} F\left((\alpha_a \otimes I_B)^T p_n, \beta_a \right) : (p_1, \ldots, p_n) \in \mathcal{P}_B \right\}.$$

There is a strategy for Bob that works for any n and is very important in the search algorithm. This is the strategy where Bob ignores all of Alice's messages and tries to choose b after learning a from Alice. By ignoring Alice's messages, he effectively sets $p_n = e_A \otimes d$, for some $d \in \text{Prob}^B$, which we optimize. Under this restriction, he can cheat with probability

$$\max_{d \in \text{Prob}^B} \frac{1}{2} \sum_{a \in \{0,1\}} F\left((\alpha_a \otimes I_B)^T (e_A \otimes d), \beta_a \right) \;=\; \max_{d \in \text{Prob}^B} \frac{1}{2} \sum_{a \in \{0,1\}} F(d, \beta_a)$$

$$= \frac{1}{2} \lambda_{\max} \left(\sqrt{\beta_0} \sqrt{\beta_0}^T + \sqrt{\beta_1} \sqrt{\beta_1}^T \right)$$

$$= \frac{1}{2} + \frac{1}{2} \sqrt{F(\beta_0, \beta_1)}$$

using Lemma C.1 and Lemma C.3. Note this is similar to the three-round case (discussed in Subsection 3.2). The reason this strategy is important is that it is easy to compute, only depends on half of the parameters, and is effective in pruning sub-optimal protocols. We call this Bob's *ignoring strategy*.

Another strategy for Bob is to measure Alice's first message, choose b accordingly, then play honestly. This is called Bob's *measuring strategy* and succeeds with probability

$$\frac{1}{2} + \frac{1}{2} \Delta(\text{Tr}_{A_2 \times \cdots \times A_n}(\alpha_0), \text{Tr}_{A_2 \times \cdots \times A_n}(\alpha_1)),$$

when $n \geq 2$.

Cheating Bob in four-round protocols

There are cheating strategies that apply to four-round protocols, that do not extend to a larger number of rounds. For example, Bob has all of Alice's $\mathbb{C}^A$ space before he sends any messages. We show that Bob can use this to his advantage. One example is Bob's measuring strategy, which leads to a cheating probability of

$$\frac{1}{2} + \frac{1}{2} \Delta(\alpha_0, \alpha_1) .$$

Similar to cheating Alice, we can develop an eigenstrategy for Bob. For the special case of four-round protocols, notice that Bob's cheating polytope contains only the constraints

$\mathrm{Tr}_B(p) = e_A$ and $p \in \mathbb{R}_+^{A \times B}$. This can be rewritten as $p_x \in \mathrm{Prob}^B$ for all $x \in A$. Also, $\mathrm{F}\left((\alpha_a \otimes \mathrm{I}_B)^{\mathrm{T}} p_n, \beta_a\right)$ can be written as $\mathrm{F}\left(\sum_{x \in A} \alpha_{a,x} p_n^{(x)}, \beta_a\right)$, where $p_n^{(x)}$ is the projection of p_n with x fixed. Thus, we can simplify Bob's reduced problem as

$$P_{\mathrm{B},0}^* = \max \left\{ \frac{1}{2} \sum_{a \in \{0,1\}} \mathrm{F}\left(\sum_{x \in A} \alpha_{a,x} p_n^{(x)}, \beta_a\right) : p_n^{(x)} \in \mathrm{Prob}^B, \text{ for all } x \in A \right\}.$$

Since fidelity is concave, we have that $\mathrm{F}\left(\sum_{x \in A} \alpha_{a,x} p_n^{(x)}, \beta_a\right) \geq \sum_{x \in A} \alpha_{a,x} \mathrm{F}(p_n^{(x)}, \beta_a)$. Therefore Bob's optimal cheating probability is bounded below by

$$\max \left\{ \frac{1}{2} \sum_{x \in A} \sum_{a \in \{0,1\}} \alpha_{a,x} \mathrm{F}(p_n^{(x)}, \beta_a) : p_n^{(x)} \in \mathrm{Prob}^B, \text{ for all } x \in A \right\}$$

which separates over $x \in A$. That is, we choose each $p_n^{(x)} \in \mathrm{Prob}^B$ separately to maximize

$$\sum_{a \in \{0,1\}} \alpha_{a,x} \mathrm{F}(p_n^{(x)}, \beta_a), \text{ which has optimal objective value } \lambda_{\max}\left(\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right)$$

using Lemma C.1. Thus, we know that

$$P_{\mathrm{B},0}^* \geq \frac{1}{2} \sum_{x \in A} \lambda_{\max}\left(\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right).$$

Since we use the concavity of the objective function, the bound we get may not be tight. Notice that solving the smaller separated problems yields a solution which is feasible for the original problem. Therefore, we can substitute this into the original objective function to get a better lower bound on Bob's optimal cheating probability. We call this Bob's *eigenstrategy*.

Since eigenvalues are expensive to compute, we can bound this quantity by

$$\begin{aligned}
\frac{1}{2} \sum_{x \in A} \lambda_{\max}\left(\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right) &\geq \min_{\beta_0, \beta_1 \in \mathrm{Prob}^B} \frac{1}{2} \sum_{x \in A} \lambda_{\max}\left(\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right) \\
&= \frac{1}{2} \sum_{x \in A} \max_{a \in \{0,1\}} \{\alpha_{a,x}\} \\
&= \frac{1}{2} + \frac{1}{2} \Delta(\alpha_0, \alpha_1) ,
\end{aligned}$$

where the last equality follows from Lemma C.2.

88

Since $\lambda_{\max}(X + Y) \leq \lambda_{\max}(X) + \lambda_{\max}(Y)$ for all matrices X and Y, we have that

$$\frac{1}{2} \sum_{x \in A} \lambda_{\max}\left(\sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right) \geq \frac{1}{2}\lambda_{\max}\left(\sum_{x \in A} \sum_{a \in \{0,1\}} \alpha_{a,x} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right)$$

$$= \frac{1}{2}\lambda_{\max}\left(\sum_{a \in \{0,1\}} \sqrt{\beta_a} \sqrt{\beta_a}^{\mathrm{T}}\right)$$

$$= \frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\beta_0, \beta_1)} \ .$$

Therefore, Bob's eigenstrategy performs better than both his measuring strategy and ignoring strategy.

Cheating Bob in six-round protocols

In six-round protocols, Bob's goal is to maximize the objective function

$$\frac{1}{2} \sum_{a \in \{0,1\}} \mathrm{F}((\alpha_a \otimes \mathrm{I}_{B_1 \times B_2})^{\mathrm{T}} p_2, \beta_a)$$

over (p_1, p_2) satisfying:

$$\begin{aligned}
\mathrm{Tr}_{B_1}(p_1) &= e_{A_1}, \\
\mathrm{Tr}_{B_2}(p_2) &= p_1 \otimes e_{A_2}, \\
p_1 &\in \mathbb{R}_+^{A_1 \times B_1}, \\
p_2 &\in \mathbb{R}_+^{A_1 \times B_1 \times A_2 \times B_2}.
\end{aligned}$$

Like in four-round protocols, we can lower bound the objective function as

$$\frac{1}{2} \sum_{a \in A_0'} \mathrm{F}\left(\sum_{x \in A} \alpha_{a,x} p_2^{(x)}, \beta_a\right) \geq \frac{1}{2} \sum_{x \in A} \sum_{a \in A_0'} \mathrm{F}(p_2^{(x)}, \alpha_{a,x} \beta_a)$$

and focus our attention on optimizing the function $\sum_{a \in A_0'} \mathrm{F}(p_2^{(x)}, \alpha_{a,x} \beta_a)$. We use the following lemma.

Lemma C.8 *For $\beta_0, \beta_1 \in \mathbb{R}_+^{B_1 \times B_2}$ and $c \in \mathbb{R}_+^{B_1}$, we have*

$$\max\left\{\sum_{a \in \{0,1\}} \mathrm{F}(p, \beta_a) : \mathrm{Tr}_{B_2}(p) = c, \ p \geq 0\right\} \geq \mathrm{F}(c, \mathrm{Tr}_{B_2}(\beta_{\tilde{a}})),$$

for any $\tilde{a} \in \{0, 1\}$.

Proof: Fix any $\tilde{a}$ and choose $p \in \arg\max \{\mathrm{F}(p, \beta_{\tilde{a}}) : \mathrm{Tr}_{B_2}(p) = c, \, p \geq 0\}$. Since the fidelity is nonnegative, the result follows by Lemma C.7. $\square$

By setting $p_1 = c \otimes e_{A_1}$, we have the constraint $\mathrm{Tr}_{B_2}(p^{(x)}) = c$ for all $x \in A$. We now apply Lemma C.8 to get

$$\max_{p_2^{(x)}} \left\{ \sum_{a \in A_0'} \mathrm{F}(p_2^{(x)}, \alpha_{a,x}\beta_a) \right\} \geq \alpha_{g(x),x} \, \mathrm{F}(c, \mathrm{Tr}_{B_2}(\beta_{g(x)})),$$

where $g(x) := \arg\max_{a \in A_0'} \{\alpha_{a,x}\}$, and 0 in the case of a tie.

Substituting this into the relaxed objective function above, we have

$$\max_{c \in \mathrm{Prob}^{B_1}} \frac{\kappa}{2} \mathrm{F}(c, \mathrm{Tr}_{B_2}(\beta_0)) + \frac{\zeta}{2} \mathrm{F}(c, \mathrm{Tr}_{B_2}(\beta_1))$$

$$= \frac{1}{2} \lambda_{\max} \left(\kappa \sqrt{\mathrm{Tr}_{B_2}(\beta_0)} \sqrt{\mathrm{Tr}_{B_2}(\beta_0)}^{\mathrm{T}} + \zeta \sqrt{\mathrm{Tr}_{B_2}(\beta_1)} \sqrt{\mathrm{Tr}_{B_2}(\beta_1)}^{\mathrm{T}} \right) \quad (19)$$

$$\geq \left(\frac{1}{2} + \frac{1}{2} \Delta(\alpha_0, \alpha_1) \right) \left(\frac{1}{2} + \frac{1}{2} \sqrt{\mathrm{F}(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1))} \right). \quad (20)$$

The quantity (20) corresponds to the strategy where Bob measures Alice's second message to try to learn a early, then tries to change the value of b. He can learn a after Alice's second message with probability $\frac{1}{2} + \frac{1}{2}\Delta(\alpha_0, \alpha_1)$. He can change the value of b with probability $\frac{1}{2} + \frac{1}{2}\sqrt{\mathrm{F}(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1))}$. Thus, he can cheat with probability at least

$$\left(\frac{1}{2} + \frac{1}{2} \sqrt{\mathrm{F}(\mathrm{Tr}_{B_2}(\beta_0), \mathrm{Tr}_{B_2}(\beta_1))} \right) \left(\frac{1}{2} + \frac{1}{2} \Delta(\alpha_0, \alpha_1) \right).$$

We call this Bob's *three-round strategy*.

Although we used many bounds in developing the quantity (12), such as concavity and the lower bound in Lemma C.8, we can recover some of the losses by generating its corresponding feasible solution and computing its objective function value for the original objective function. For example, we can calculate c as the entry-wise square of the normalized principal eigenvector of

$$\frac{1}{2} \lambda_{\max} \left(\kappa \sqrt{\mathrm{Tr}_{B_2}(\beta_0)} \sqrt{\mathrm{Tr}_{B_2}(\beta_0)}^{\mathrm{T}} + \zeta \sqrt{\mathrm{Tr}_{B_2}(\beta_1)} \sqrt{\mathrm{Tr}_{B_2}(\beta_1)}^{\mathrm{T}} \right),$$

then calculate $p_2^{(x)}$ for each value of x from the construction of the feasible solution in the proof of Lemma C.7. We call this Bob's *eigenstrategy*.