Quantum Coin Flipping

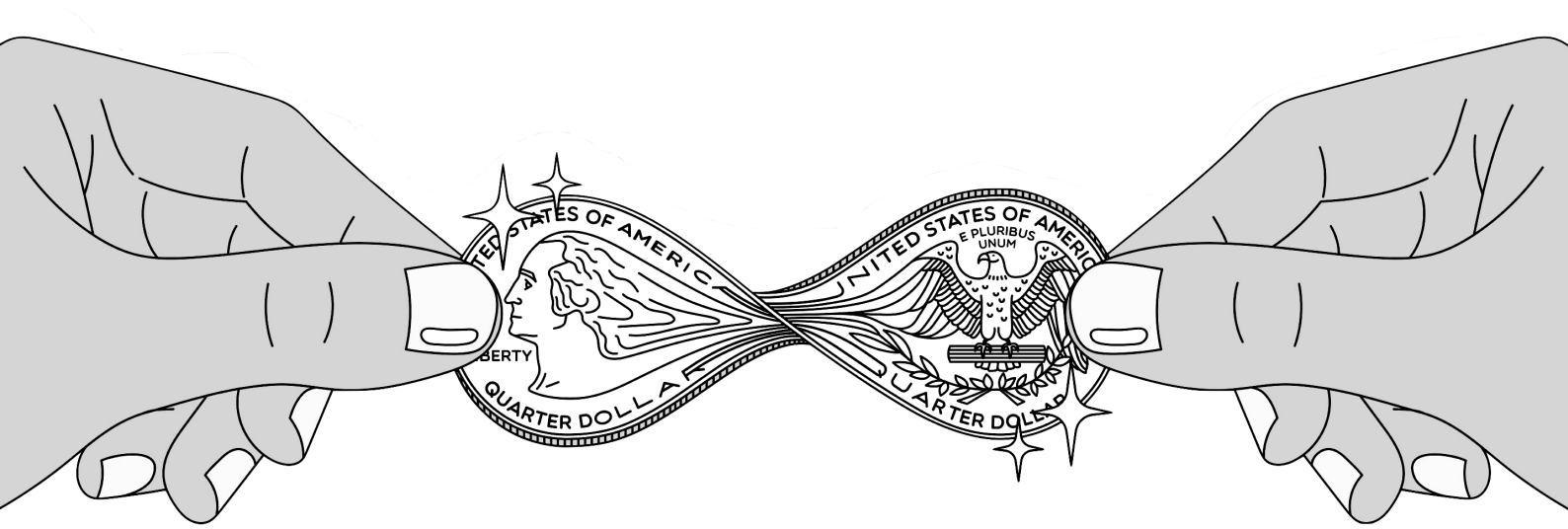
and

Circuit Design Problems

in

Distributed Quantum Computing

Roy van Houte



Master Thesis at Delft University of Technology

The image on the cover of this thesis has been retrieved and adapted from the article *New Quantum Paradox Clarifies Where Our Views of Reality Go Wrong* on November 15, 2019 from Quanta Magazine, see: https://www.quantamagazine.org/frauchiger-renner-paradox-clarifies-where-our-views-of-reality-go-wrong-20181203/

Quantum Coin Flipping

and

Circuit Design Problems

in

Distributed Quantum Computing

January 6, 2020

by

R. van Houte

to obtain the degree of Master of Science
at the Delft University of Technology,

Student:	Roy van Houte	
Student number:	4296931	
Master's program:	Applied Mathematics, optimization	
Faculty:	Electrical Engineering, Mathematics and Computer Science	
Thesis defence date:	January 16, 2020	
Supervisors:	Prof. dr. K.I. Aardal,	TU Delft, Responsible Professor.
	Dr. D. de Laat,	TU Delft, Daily supervisor,
	Prof. dr. S.O. Fehr	CWI, External expert,
	T. Attema MSc,	TNO, Daily supervisor,
	Dr. J.W. Bosman,	TNO, Daily supervisor,
Research Institute:	Delft University of Technology in collaboration with TNO The Hague	
Contact:	`royvanhoute@me.com`	

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.

Abstract

Quantum coin flipping is a cryptographic primitive in which two or more parties that do not trust each other want establish a fair coin flip. These parties are not physically near each other and use quantum communication channels to interact. A quality of protocols is measured by the best possible cheating strategy, which is the solution of a complex semidefinite optimization problem. In this master thesis we show new explicit bounds in multiparty quantum coin flipping, we investigate how to explicitly formulate these problem in a standard form, we show that a fair coin flip results in the lowest possible bias and we determine more measures of the quality of a protocol. Furthermore, this master thesis presents a rigorous and detailed mathematical description of semidefinite optimization, quantum information theory and quantum coin flipping.

This thesis also includes an article written together with J. Mulderij, T. Attema, I. Chiscop and F. Phillipson on distributed quantum computing. In this article, we pose new questions and formulate integer linear programs that solve to find optimal assignment of qubits to computers for a given network of quantum computers and quantum algorithm.

Preface

"It is easier to square the circle than to get round a mathematician."

- Augustus De Morgan

This master thesis on quantum coin flipping is the result of nine months doing research on what is fair and how to prevent cheating in protocols to establish randomness. With this thesis I complete the master Applied Mathematics with a specialisation in optimization at the Delft University of Technology. I conducted this thesis in an internship at the department of Cyber Security and Robustness at TNO, located at Anna van Buerenplein in The Hague.

I started at TNO in September of 2018 on an internship project on quantum applications. In this internship I got to know TNO very well and and it even resulted in an article on quantum phase estimation. In April 2019 I started my graduation project at TNO on parameter optimization of practical implementations of quantum key distribution systems. During the process of gathering literature and information I got interested in the elegant combination of quantum information theory and semidefinite optimization. After some thorough research I proposed to focus the thesis on this subject, in particular in the application of quantum coin flipping.

Nine months later and this thesis is the result. Of course, I could not have done this alone. I want to thank Thomas Attema, for his excellent supervision, confidence and time to discuss yet another mathematical problem[1]; David de Laat, for sharing his knowledge on quantum information theory and semidefinite optimization[2], motivating me and inspiration; Karen Aardal, for her good advises and supervision of this thesis; Joost Bosman, for his eagerness to learn new subjects, enthusiasm and good questions; Serge Fehr, for taking the time and interest in this thesis; special thanks to Sander Gribling, for his interest, help with filling in the details of this thesis and inviting me to get to know Centrum Wiskunde & Informatica (CWI) in Amsterdam.

I would like to thank Jesse Mulderij and Nicholas Meinhardt[3], for collaborating, pondering about quantum related topics, attempting heavy sports like bouldering and cheese fondue, and always making time for a good talk.

Furthermore, I would like to thank my parents and family, not for understanding this thesis, but for taking the risk of trusting that these equations actually mean something[4]. I also want to thank my friends from Delft, Zeeland, and elsewhere, for paying interest in my graduation project, but also provide me with sufficient distraction in the form of games, sports, drinks, parties, and so on.

---

[1] Quick problem: Let x, y be positive integers. Prove that if $(x^2 + y^2)/(xy + 1)$ is integer, it is a perfect square.

[2] I really appreciated the comments on mathematical notation, (American) English, and LaTeX typesetting.

[3] Who will keep an eye on *the guys*, now that we both graduated?

[4] To be honest, I do not know what quantum mechanics actually means.

I am thankful for my time at the department of Cyber Security and Robustness of TNO and getting to know colleagues and interns[5], I enjoyed going to TNO every day. I learned a lot about working in a research organisation. I cherish the conversations and discussions and I am glad to be able to attend not one, but two Christmas parties.

I also want to thank the treasury department of Rabobank and in particular Frank Mulder, for giving me the opportunity to get to know this financial institution. I learned a lot during these months and I enjoyed being part of this close team.

Last but not least, I would like to thank Alice and Bob, your relation might not always be puppies and sunshine, but you have been a great inspiration.

Of course, a preface is always complete when it includes a poem, especially when it is related to the subject.

A Psychological Tip

Whenever you're called on to make up your mind,
And you're hampered by not having any,
The best way to solve the dilemma, you'll find,
Is simply by spinning a penny.

No—not so that chance shall decide the affair
While you're passively standing there moping;
But the moment the penny is up in the air,
You suddenly know what you're hoping.

- Piet Hein

I hope you enjoy reading this thesis.

- Roy van Houte
The Hague, January 6, 2020

[5]Special thanks of course go to Stijn Pletinckx, for his immeasurable endeavor to be acknowledged in theses.

Contents

Notation

The following notation is used throughout this thesis.

Notation	Description
$[n]$	The set of integers $\{1, \ldots, n\}$
$\mathbf{N}$	The set of all positive integers $\{1, 2, \ldots\}$
$\mathbf{Z}$	The ring of integers
$\mathbf{Q}$	The field of rational numbers
$\mathbf{R}$	The field of real numbers
$\mathbf{C}$	The field of complex numbers
$\mathcal{X}, \mathcal{Y}, \ldots$	Complex Euclidean spaces
$L(\mathcal{X}, \mathcal{Y})$	The linear operators from $\mathcal{X} \to \mathcal{Y}$
$L(\mathcal{X})$	The linear operator from $\mathcal{X} \to \mathcal{X}$
$T(\mathcal{X}, \mathcal{Y})$	Superoperators, linear operators from $L(\mathcal{X}) \to L(\mathcal{Y})$
$D(\mathcal{X})$	The set of density matrices on $\mathcal{X}$
$\mathrm{Herm}(\mathcal{X})$	The set of Hermitian operators on $\mathcal{X}$
$\mathrm{Sym}(\mathcal{X})$	The set of symmetric matrices on $\mathcal{X}$
$I_{\mathcal{X}}$	Identity operator on the space $\mathcal{X}$
$J_{\mathcal{X}}$	The all-ones matrix on $\mathcal{X}$
J	The Choi-Jamiołkowski map
$[\cdot, \cdot]$	The commutator of linear operators
$\rho, \sigma, \ldots$	Density operators
$\succeq, \succ$	Positive (semi)definite, Loewner partial order
$\cdot^{\top}$	Transpose of a vector or operator
$\cdot^{\dagger}$	Hermitian transpose of a vector or adjoint of an operator
$\cdot^{*}$	Optimal value or solution to an optimization problem
$\otimes$	Tensor product of vector spaces, a tensor or the Kronecker product
$\mathcal{X}^n$	The n-fold Cartesian product $\underbrace{\mathcal{X} \times \cdots \times \mathcal{X}}_{n}$, direct sum $\underbrace{\mathcal{X} \oplus \cdots \oplus \mathcal{X}}_{n}$
vec	The vectorize function $L(\mathcal{Y}, \mathcal{X}) \to \mathcal{X} \otimes \mathcal{Y}$
$\oplus$	Direct sum of vector spaces, XOR-operation on $\{0, 1\}$
Tr	Trace of an operator
$\mathrm{Tr}_{\mathcal{X}}$	Partial trace over $\mathcal{X}$
Re	The real part of a complex number or matrix
Im	The imaginary part of a complex number or matrix
$\lvert\cdot\rangle$	*ket*, notation of a quantum mechanical state (vector)
$\langle\cdot\rvert$	*bra*, notation of the conjugate of a state
$\langle\cdot, \cdot\rangle, \langle\cdot\rvert\cdot\rangle$	Inner product
$\lVert\cdot\rVert$	The norm in a Euclidean space
h	Planck's constant, $h = 6.62607015 \cdot 10^{-34}$ Js
$\hbar$	Planck's reduced constant, $\hbar = h/2\pi$.

Chapter 1

Introduction

"A mathematician is a blind man in a dark room looking for a black cat which isn't there."

- Attributed to Charles Darwin

WRITING a collaborative paper is often a holistic phenomenon: working together results in more than everyone individually. However, when it comes to, for example, the order in which authors appear in the article, it may be subjective what is considered to be fair. Flipping a fair coin will settle the debate [1][1]. But how to flip a coin when two parties are not physically near each other? And how to make sure that a cheater does not influence the outcome too much? Fortunately, quantum communication provides the answer. A rich and elegant subject on the intersection of quantum physics, analysis and optimization. Altogether a subject, which leads to a lot collaboration by itself.

1.1 Quantum Coin Flipping

In this thesis we consider the problem of creating shared randomness between two or more parties that do not trust each other. Besides the situation described in the introduction of this chapter, another applications can be found in the context of mental poker [2]. In this setting, players wish to play a game of poker and do not trust each other. It is therefore essential that a deck of cards is shuffled in a random way and that a potential cheater is limited in his or her ability to fix a prefered outcome. Randomness in its most primitive form is a coin flip. Combining coin flips gives random strings of bits or, equivalently, random numbers. If we have two parties, Alice and Bob, at different locations, a simple protocol would be to let one of the parties generate a private coin flip and share it with the other party. However, in this case the generator has full control over the protocol and can force any outcome if he or she decides to cheat.

Therefore, Alice and Bob use a protocol that requires actions on both sides and communication over a channel. They want to meet the following conditions:

1. If Alice and Bob are both honest, then the outcome should be a realization of a Bernoulli random variable with equal probability for both outcomes.

2. If one of the players is dishonest, then the deviation from a fair coin as a result of cheating should be limited. For any cheating strategy, the probability that the honest player will find

[1] In this article, the order of the first and last pair of authors is determined by a coin toss

a chosen outcome is in some interval $[1/2 - \varepsilon, 1/2 + \varepsilon]$. The smallest value ε that applies to every cheating strategy is called the *bias* of a protocol.

If for such a protocol $\varepsilon = 0$, then the dishonest player cannot influence the coin at all, in which case a protocol is called *perfect*. If $\varepsilon = 1/2$, then he or she can completely determine the outcome of the coin flip and a protocol is called *completely broken*. The main goal of studying coin flipping protocols is to determine protocols that have a bias as small a possible.

In a coin flipping protocol, both players have their own space of (quantum) information and a shared space with which they can both interact and communicate. A quantum coin flipping protocol consists of a number of alternating operations on their private and shared information ending a measurement which determines the outcome of the coin. Players might have a preference for a specific outcome. For example when authors want to determine the order of their names in an article. Consequently, if a cheater is present we know that he or she will only prefer one certain outcome. In this case we consider the *weak* bias. If honest players do not have a preference and the preference of a cheater is not predetermined, we look at the *strong* bias. A protocol in itself is not weak or strong. But a specific protocol will often be designed to result in a good weak or strong bias. We will mostly discuss protocols with a good strong bias.

Protocols that use classical communications and do not make any assumptions on the computational power of both players, do not exist [3, 4], neither weak nor strong. However, if we consider protocols that use classical communication with computational assumptions, then there do exist protocols to create a realization of a fair coin flip in which a cheater can not influence the outcome. For example Blum's coin flipping protocol [5], described in Section 5.1 provides a procedure to generate fair randomness. This method is based on the assumption that it is computationally hard to factor integers. These are the same assumptions that are made in for example RSA encryption. It is currently unknown whether there exists a polynomial time algorithm to factor integers. Furthermore, it is unknown whether integer factorization is NP-complete.

These open questions on the problem of factoring integers do, however, not apply in quantum computing. Storing information and performing operations on quantum mechanical systems is different from ordinary computers and allows for effects that cannot be replicated efficiently by classical computers. An example of a quantum algorithm is *Shor's factoring algorithm* [6], that factors integers in time $O((\log n)^2(\log\log n)(\log\log\log n))$, where n is the integer. Complexity assumptions related to other mathematical problems than factoring, might also defeat the security on a quantum computer. This means we have to be careful with using complexity assumptions and we want to discard them altogether if possible.

Nevertheless, cryptography and security in ICT applications remains a requirement and we therefore have to look for solutions. Two main directions to look are:

1. Designing *post quantum* coin flipping protocols. These protocols can be executed on classical computers using classical information channels and rely on complexity assumptions that are also valid for quantum computers, i.e., mathematical problems that are most probably also hard for quantum computers. Lattice based schemes or code based schemes offer solutions in, e.g., key distribution and commitment schemes. These schemes can be used make secure coin flipping protocols;

2. A second way of establishing a quantum safe coin flipping protocol is to use quantum information and communication by itself. In this case the security is fundamentally provided by the physical laws of quantum mechanics. This means the states of the systems and operations are quantum mechanical instead of classical.

If we investigate the second option in which we use quantum information and communication as a fundamental basis for protocols, we first have to mathematically formalize the problem. This

means we have to specify what the laws of quantum mechanics are, which manipulations in the systems are possible and how to optimize over these possibilities.

A cheater will perform different operations on the system than specified by the protocol. The sequence of these alternate operations is called *a cheating strategy*. An equivalent way of describing a cheating strategy is in terms of density operators of the system after the alternative operators are applied. A density operator is a positive semidefinite operator with unit trace. We want to find a cheating strategy that results in the highest probability of measuring a chosen outcome of the honest party. The objective and constraints are given by linear superoperators and the optimization problem to find the optimal cheating strategy therefore becomes a complex semidefinite program. The highest probability of successfully cheating leads to the bias of a protocol. Hence, the bias of a quantum coin flipping protocol given by a semidefinite program.

Two honest players wish to find a coin flip, that results in the probability of heads and tails with equal probability. We call this *balanced* coin flipping. If two honest players decide to do a coin flip that has not the same probability for both outcomes, we will refer to this as *imbalanced* coin flipping. Depending on the application, there might be a trade-off between the optimal cheating probability and whether or not the coin flip is balanced. For example, it might be preferable to choose an imbalanced quantum coin flipping protocol with a 55% chance of finding heads and 45% chance of finding tails and a strong bias of 10%, rather than a balanced protocol with a strong bias of 25%. It is therefore interesting to know what the relation is between the probability of both outcomes when both players are honest and the minimal bias of corresponding quantum coin flipping protocols.

Using duality and constructing feasible solutions of these semidefinite programs it is possible to derive a lower bound that holds for any kind of quantum coin flipping protocol. Furthermore, the mathematical framework of two player quantum coin flipping can be extended in a natural way to a multiplayer setting. In this setting we assume that a known fraction of the players may behave dishonest, but we do not know which exact players.

A coin flip can result in three possible outcomes: heads, tails or abort. Depending on the situation, a cheater might have a preference for these outcomes. Suppose that a cheater above all wants to maximize the probability of the outcome 1. This leads to the primary optimization to determine optimal cheating strategy. Within this set of possible optimal cheating strategies, a cheater may want to minimize his or her probability of aborting the protocol. This optimization program is a modification of the primary optimization problem and is applied to several explicit protocols. In a similar way, if we weigh the outcomes of a protocol, we formulate an optimization program that determines the optimal expected win and a corresponding optimal cheating strategy.

In this thesis we answer the following main question.

> **Main question:** *How can semidefinite optimization be used to solve problems and questions in quantum coin flipping?*

This main question is answered by considering the following subquestions.

The subquestions of this thesis are:

1. *How can one find the optimal cheating strategy and cheating probability of a quantum coin flipping protocol?*
2. *Is balanced coin flipping the optimal way of achieving protocols with the lowest possible bias?*
3. *How can one explicitly setup the semidefinite programming problems to solve?*
4. *How can one determine the optimal probability of a secondary preference in quantum coin flipping and what are its values for explicit protocols?*
5. *How can we use new bounds on quantum coin flipping to determine better explicit bounds on multiparty quantum coin flipping?*

1.2 Mathematical Problem Statement

The bias of a given protocol is the maximum deviation from a fair coin regarding all players, all outcomes and all possible cheating strategies. The goal is to find a protocol with a bias as low as possible, because in that case honest players are secured against cheaters as much as possible.

The problem can be broken down into three stages. First, we fix a quantum coin flipping protocol $\mathcal{P}$, an honest player and an outcome the cheater wants the honest player to output. Suppose that Alice is honest and Bob wants to enforce outcome 0. For a given cheating strategy[2] X, we denote the resulting probability of Alice measuring the outcome 0 by $P_{A,0}(X,\mathcal{P})$. The optimal cheating probability is

$$P^*_{A,0}(\mathcal{P}) = \sup\{P_{A,0}(X,\mathcal{P}) : X \text{ is a cheating strategy against Alice in protocol } \mathcal{P}\}. \tag{1.2.1}$$

The other three situations, in which Bob cheats and forces outcome 1 on Alice and Alice cheats to force outcomes 0 or 1 on Bob, have similar definitions denoted by $P_{A,1}(X,\mathcal{P})$, $P_{B,0}(X,\mathcal{P})$ and $P_{B,1}(X,\mathcal{P})$ respectively.

This optimization problem is a complex semidefinite programming problem. To quantify the bias of a protocol we have to take all four possible situations of cheaters and outcomes into account. The worst possible deviation from a fair coin is called the strong bias,

$$\varepsilon(\mathcal{P}) = \max\{P^*_{A,0}(\mathcal{P}), P^*_{A,1}(\mathcal{P})\, P^*_{B,0}(\mathcal{P}), P^*_{B,1}(\mathcal{P})\} - \frac{1}{2}. \tag{1.2.2}$$

In the field of quantum coin flipping, a protocol that has a low bias is considered to be a good protocol, since cheaters have relatively little influence on the distribution of the coin. Ideally we want to find a best protocol. The best bias and corresponding protocol is given by

$$\varepsilon^* = \inf\{\varepsilon(\mathcal{P}) : \mathcal{P} \text{ is a quantum coin flipping protocol}\}. \tag{1.2.3}$$

Kitaev in 2002 proved that $\varepsilon^* \geq 1/\sqrt{2} - 1/2 \approx 0.20711\ldots$. This proof is based on formulating the semidefinite programming approach of determining $P^*_{A,0}(\mathcal{P})$. Combining the dual semidefinite programs for both parties cheating and forcing a certain outcome leads to lower bounds. In 2009 Chailloux and Kerenidis proved that this bound is in fact tight. These results hold for two party quantum coin flipping. In 2004 an assymptotic bound was found for the multiplayer situation.

[2] In Section 5.4 we will discuss what kind of mathematical object a cheating strategy X is.

Theorem 1.2.1 (Ambainis, Burhman, Dodis and Röhrig, 2004, [7]) *For group of k players of whom g are honest the bias of quantum coin flipping is*

$$\frac{1}{2} - \Theta\left(\frac{g}{k}\right). \tag{1.2.4}$$

It is particularly hard to determine explicit bounds for given values of g and k, so our goal will be to determine upper and lower bounds on the multiplayer bias.

Lower bounds on the multiplayer bias can be determined by extending Kitaev's method for two players to many players.

To determine upper bounds on the bias, we build protocols for k players based on rounds of two player quantum coin flipping protocols. To get the asymptotic result of Theorem 1.2.1 it won't be sufficient to use coin flipping protocols that are suitable for weak and strong coin flipping. A third variation of coin flipping is introduced, called *coin flipping with penalty*, in which a cheater that has been detected loses a specified amount of money. By making the amount of money depend on the round in the overall protocol, the cheaters are more careful and this in the end leads to the asymptotically optimal bias of Theorem 1.2.1.

1.3 Quantum Coin Flipping Protocols

The first classical coin flipping protocol was introduced in 1981 in the article "Coin flipping by telephone: A protocol for solving impossible problems" by Manuel Blum [5]. In the now famous article by Bennett and Brassard in 1984, that presented the first quantum key distribution protocol [8], the authors also gave a quantum coin flipping protocol. However, this protocol has bias $\varepsilon = 1/2$ and hence is completely broken.

An overview of explicit protocols and families of protocols are shown in Table 1.1.

Table 1.1: Known quantum coin flipping protocols with their type and bias. Remarks are indicated by (1): completely broken, (2): practically implementable, (3): family of protocols, (4): optimal, $(*)$ this (long) expression can be found in the corresponding reference.

Protocol or authors	Type	Bias	Decimal form	year	Remark	Ref.
Bennett & Brassard	Strong	$1/2$	0.5	1984	1	[8]
Aharonov, Ta-Shma, Vazirani & Yao	Strong	$\sqrt{2}/4$	$0.35355\ldots$	2000		[9, 10]
Spekkens & Rudolph	Strong	$(\sqrt{5}-1)/4$	$0.30902\ldots$	2001		[10]
Spekkens & Rudolph	Strong	$1/4$	0.25	2001		[11]
Ambainis	Strong	$1/4$	0.25	2004		[12]
Mochon	Weak	$1/6$	$0.16667\ldots$	2005		[13]
Mochon	Weak	$*$	$0.192\ldots$	2004	3	[14]
Berlín, Brassard, Bussières & Godbout	Strong	$2/5$	0.4	2009	2,3	[15, 16]
Chailloux & Kerenidis	Strong	$(\sqrt{2}-1)/2$	$0.20711\ldots$	2009	3,4	[17]
Mochon	Weak	0	0	2007	3,4	[18]
Singh Arora, Roland & Weis	Weak	$1/10$	0.1	2019	3	[19]

Very little is known about the relation between the optimal strong or weak bias and the number of rounds or dimensions of the private and message spaces of a protocol. One result is a weak

lower bound on the number of

Theorem 1.3.1 (A. Ambainis, [12]) *Suppose we have a quantum coin flipping protocol with a weak bias $\varepsilon > 0$. Then the number of rounds of this protocol is at least $\Omega(\log\log(1/\varepsilon))$.*

This theorem suggests that as ε approaches zero, the number of rounds goes to infinity.

Related subjects to quantum coin flipping are for example oblivious transfer protocols, in which a sender sends information to a receiver, without the receiver knowing whether or what information has been sent [20]. This applications do not necessarily share the same properties, but the security of these protocols can be modelled in a similar way as quantum coin flipping using semidefinite optimization. A coin flip can be considered as a realization of a Bernoulli random variable. A die roll with n sides is a realization of a uniform random variable on the set $\{1, \ldots, n\}$, and is therefore a natural generalization of quantum coin flipping. Quantum die rolling is discussed by Sikora in 2017 [21], and provides extensions to the bounds for coin flipping. Similar to multiple parties for quantum coin flipping, one can also consider multiple players in quantum die rolling.

1.4 Contributions

The contributions of this thesis to quantum coin flipping are relatively diverse and build upon several existing problems. The contributions in this thesis are:

1. A comprehensive description of semidefinite optimisation, quantum information theory and quantum coin flipping.

2. We introduce known results of two player quantum coin flipping in order to establish bounds in multiparty quantum coin flipping. This leads to new explicit bounds that are better than the bounds on which the proof of Theorem 1.2.1 is based.

3. We present variations on problems in quantum coin flipping. We show how we can find the best cheating strategy is we have an ordered list of preferences of outcomes and how to optimize the expected win based on weights of the different outcomes.

4. We explicitly model a quantum coin flipping protocol as a quantum circuit. From this quantum circuit it is easy to determine the standard form of a protocol which is used in the optimization program. Based on the standard form, the optimal cheating strategies, probabilities and biases of three protocol are solved numerically.

5. We show that balanced coin flipping is preferable over imbalanced coin flipping. This result has not been discussed in literature and strengthens the setting of balanced coin flipping on which many theory is based.

6. We present a detailed description of parts that are skipped by literature such as a rigorous proof of strong duality, assumptions that lead to symmetry in the pair of programs and reductions of multiparty quantum coin flipping. We also give a detailed description on why real and complex semidefinite programming are equivalent and how we can transform complex to real semidefinite programs.

7. The joined work on distributed algorithms in networks of quantum computers. This is a separate topic of this thesis that does not have a lot of overlap with quantum coin flipping and is therefore discussed in Part I.

Because of the differences between the subjects quantum coin flipping and quantum network optimization, the thesis is split in two parts. In Part I we look a quantum coin flipping and in

Part II we look at modelling quantum circuit design problems in networks of quantum computers.

In Part I we have the following structure. First, in Chapter 2, we develop the necessary mathematical preliminaries. In Chapter 3 we look at the theory of semidefinite programming applied to real and complex operators. Next, in Chapter 4 we present the necessary ideas from quantum mechanics and quantum information theory and consider some interesting applications of semidefinite programming in quantum information theory. In Chapter 5.4 we combine semidefinite optimization and quantum information theory to describe the problem of quantum coin flipping and derive bounds. Finally we end with conclusions and recommendations for further research in Chapter 6.

Part II consists of an introduction on the subject of Distributed quantum computing and quantum networks and this is followed by the article.

Part I

Quantum Coin Flipping

Chapter 2

Mathematical Preliminaries

"We (Halmos and Kaplansky) share a philosophy about linear algebra: we think basis-free, we write basis-free, but when the chips are down we close the office door and compute with matrices like fury."

- Irving Kaplansky

To describe quantum systems, we need to define states, unitary operations, measurements and combinations of systems in terms of Euclidean spaces, operators and tensor products. Semidefinite programming requires knowledge of positive semidefinite operators and inner products on the space of operators. Both can be done in the language of linear algebra, as we will see later in Section 4. In this chapter we introduce the necessary mathematical tools to describe the problem rigorously.

Throughout this thesis we will make use of Hilbert spaces. In this chapter, we assume these spaces are finite dimensional (over $\mathbf{R}$ and $\mathbf{C}$) as the applications are based on finite dimensional spaces. This means that every space can be regarded as the vector space $\mathbf{R}^n$ or $\mathbf{C}^n$ for some positive integer n, and thus in particular as a complex Euclidean space. When dealing with multiple vector spaces it is sometimes more convenient to refer to them in an abstract way as $\mathcal{A}, \mathcal{B}, \mathcal{X}, \mathcal{Y}, \ldots$ instead of the more explicit $\mathbf{C}^n$.

2.1 Complex Euclidean Spaces and Tensor Products

The first mathematical tool we describe is the *tensor product* of vector spaces [22]. Informally, the tensor product allows us to describe a vector space that consists of *'vectors of vectors'* from different spaces that share the same scalar field. We will first give the general definition, followed by some examples and properties.

Definition 2.1.1 (Tensor product of vector spaces) *Let $\mathcal{X}, \mathcal{Y}$ be finite-dimensional vector spaces over the field of complex numbers $\mathbf{C}$. Then the tensor product $\mathcal{X} \otimes \mathcal{Y}$ is the vector space over $\mathbf{C}$ generated by the elements $x \otimes y$, $x \in \mathcal{X}, y \in \mathcal{Y}$, called elementary tensors, for which it holds that:*

1. *For all $x_1, x_2 \in \mathcal{X}$ and $y \in \mathcal{Y}$,*
$$(x_1 + x_2) \otimes y = x_1 \otimes y + x_2 \otimes y. \tag{2.1.1}$$
2. *For all $x \in \mathcal{X}$ and $y_1, y_2 \in \mathcal{Y}$,*
$$x \otimes (y_1 + y_2) = x \otimes y_1 + x \otimes y_2. \tag{2.1.2}$$
3. *For all $\alpha \in \mathbf{C}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$,*
$$\alpha(x \otimes y) = (\alpha x) \otimes y = x \otimes (\alpha y). \tag{2.1.3}$$

Notice that the set $\{x \otimes y : x \in \mathcal{X}, y \in \mathcal{Y}\} \subseteq \mathcal{X} \otimes \mathcal{Y}$ is in general *not* a vector space.

Since both $\mathcal{X}$ and $\mathcal{Y}$ are finite dimensional, every tensor in $\mathcal{X} \otimes \mathcal{Y}$ can be written as

$$\sum_{i \in I} x_i \otimes y_i, \tag{2.1.4}$$

for some finite index set I and vectors $x_i \in \mathcal{X}, i \in I$, and $y_i \in \mathcal{Y}, i \in I$.

We will first consider an intuitive description on finite dimensional spaces. Suppose $\mathcal{X} = \mathbf{C}^n$ and $\mathcal{Y} = \mathbf{C}^m$, then $\mathcal{X} \otimes \mathcal{Y}$ is isomorphic to $\mathbf{C}^{nm}$. This isomorphism is given by the map (defined on elementary tensors):

$$\begin{aligned} \psi\colon\ \mathbf{C}^n \otimes \mathbf{C}^m &\xrightarrow{\sim} \mathbf{C}^{nm} \\ (a_1, \ldots, a_n) \otimes (b_1, \ldots, b_m) &\mapsto (a_1 b_1, \ldots, a_1 b_m, \ldots, a_n b_1, \ldots, a_n b_m). \end{aligned} \tag{2.1.5}$$

In particular,

$$\dim(\mathcal{X} \otimes \mathcal{Y}) = \dim(\mathcal{X}) \cdot \dim(\mathcal{Y}). \tag{2.1.6}$$

If $\{e_1, \ldots, e_n\}$ is a basis of $\mathcal{X}$ and $\{f_1, \ldots, f_m\}$ is a basis of $\mathcal{Y}$, then

$$\{e_i \otimes f_j : i \in [n], j \in [m]\}, \tag{2.1.7}$$

is a basis of $\mathcal{X} \otimes \mathcal{Y}$ and every element can be written as

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \alpha_{ij} e_i \otimes f_j, \tag{2.1.8}$$

for number $\alpha_{ij} \in \mathbf{C}, i \in [n], j \in [m]$.

Example 2.1.1 *Suppose we have two vectors $(2, -3), (-1, 4) \in \mathbf{C}^2$, then $\mathbf{C}^2 \otimes \mathbf{C}^2$ can be associated with $\mathbf{C}^4$ by the map 2.1.5. Therefore*

$$\psi((2, -3) \otimes (-1, 4)) = (-2, 8, 3, -12). \tag{2.1.9}$$

On the other hand, suppose we have the vector $(0, 1, 2, -3) \in \mathbf{C}^4$, then it is easy to see that there is no pair of vectors $x, y \in \mathbf{C}^2$ such that $\psi(x \otimes y) = (0, 1, 2, -3)$. Otherwise, the product of the first component of x and y has to be zero, therefore one of these first components has to be zero and necessarily $x \otimes y$ has to have at least another component zero, which is not the case.

However, we can write

$$(0, 1, 2, -3) = \psi((1, 0) \otimes (0, 1) + 2(0, 1) \otimes (1, 0) - 3(0, 1) \otimes (0, 1)), \tag{2.1.10}$$

as a linear combination of simple tensors in terms of the standard basis.

Also the inner product of two vector spaces can be extended to an inner product of the tensor space. More specifically, if $\mathcal{X}, \mathcal{Y}$ are two complex Euclidean spaces with inner products $\langle \cdot, \cdot \rangle_{\mathcal{X}}$ and $\langle \cdot, \cdot \rangle_{\mathcal{Y}}$ respectively, then $\mathcal{X} \otimes \mathcal{Y}$ is a complex Euclidean space with inner product $\langle \cdot, \cdot \rangle$ defined on simple tensors and extended linearly on the extended linearly on the first argument en conjugate linear on the second argument to $\mathcal{X} \otimes \mathcal{Y}$ by

$$\langle x_1 \otimes y_1, x_2 \otimes y_2 \rangle = \langle x_1, x_2 \rangle_1 \langle y_1, y_2 \rangle_2, \quad \text{for all } x_1, x_2 \in \mathcal{X}, y_1, y_2 \in \mathcal{Y}. \tag{2.1.11}$$

If $\mathcal{X}$ and $\mathcal{Y}$ are complex vector spaces, then $L(\mathcal{X}, \mathcal{Y})$ denotes the space of all linear maps $\mathcal{X} \to \mathcal{Y}$. The space $L(\mathcal{X}, \mathcal{Y})$ is a complex vector space itself. If $\mathcal{X} = \mathcal{Y}$, then $L(\mathcal{X}, \mathcal{X})$ is also denoted simply by $L(\mathcal{X})$.

The identity map on $\mathcal{X}$ is denoted by $I_{\mathcal{X}}$. If explicitly $\mathcal{X} = \mathbf{C}^n$, then we also write I_n. Similarly, for $\mathcal{X} = \mathbf{C}^n$ the *elementary matrices* are represented by the set of matrices E_{ij} for $i, j \in [n]$.

$$(E_{ij})_{k\ell} = \begin{cases} 1 & \text{if } k = i \text{ and } j = \ell \\ 0 & \text{otherwise} \end{cases}, \quad \text{for } k, \ell \in [n], \tag{2.1.12}$$

The set of elementary matrices forms a basis of $L(\mathcal{X})$ and thus every operator $X \in L(\mathcal{X})$ can be written as the matrix

$$X = \sum_{i=1}^{n} \sum_{j=1}^{n} \langle E_{ij}, X \rangle E_{ij}. \tag{2.1.13}$$

A real operator $X \in L(\mathbf{R}^n)$ is called *symmetric* if $X = X^\top$ and the set of symmetric operators is denoted by $\mathrm{Sym}(\mathbf{R}^n)$ which forms a real vector space. A complex operator $X \in L(C^n)$ is called *Hermitian* if $X = X^\dagger$ and the set of all Hermitian operators is denoted by $\mathrm{Herm}(\mathbf{C}^n)$. The set $\mathrm{Herm}(\mathbf{C}^n)$ is a real vector space but not a complex vector space. This can easily be seen from the fact that if $X \in \mathrm{Herm}(\mathbf{C}^n)$, then $iX = -iX^\dagger$.

An important map in $L(\mathcal{X}, \mathbf{C})$ where $\mathcal{X} = L(\mathbf{C}^n)$ for some positive integer n, is the *trace* of an operator. Suppose an operator $X \in L(\mathbf{C}^n)$ is represented by a matrix

$$X = \begin{pmatrix} X_{11} & \dots & X_{1n} \\ \vdots & \ddots & \vdots \\ X_{n1} & \dots & X_{nn} \end{pmatrix}, \tag{2.1.14}$$

then the trace is the linear map, defined by

$$\mathrm{Tr}(X) = \sum_{i=1}^{n} X_{ii}. \tag{2.1.15}$$

An important property of the trace is that $\mathrm{Tr}(XY) = \mathrm{Tr}(YX)$ for all $X, Y \in L(\mathcal{X})$. In particular, if U is a unitary operator in $L(\mathcal{X})$, then

$$\mathrm{Tr}(UXU^\dagger) = \mathrm{Tr}(XU^\dagger U) = \mathrm{Tr}(X) \tag{2.1.16}$$

and thus the trace map is independent of the chosen basis.

Linear maps between spaces can also be extended to linear maps between the tensor product of their spaces in a natural way. From a pair of operators $A \in L(\mathcal{X}, \mathcal{X}')$ and $B \in L(\mathcal{Y}, \mathcal{Y}')$ we define an operator in $L(\mathcal{X} \otimes \mathcal{Y}, \mathcal{X}' \otimes \mathcal{Y}')$ on simple tensors by

$$\begin{aligned} A \otimes B \colon \mathcal{X} \otimes \mathcal{Y} &\to \mathcal{X}' \otimes \mathcal{Y}' \\ x \otimes y &\mapsto (Ax) \otimes (By). \end{aligned} \tag{2.1.17}$$

Explicitly we can define the map given a basis of a finite dimensional vector space

$$\begin{gathered} \Psi \colon L(\mathbf{C}^n, \mathbf{C}^m) \times L(\mathbf{C}^p, \mathbf{C}^q) \xrightarrow{\sim} L(\mathbf{C}^{np}, \mathbf{C}^{mq}) \\ (A, B) = \left(\begin{pmatrix} a_{11} & \dots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{n1} & \dots & a_{nm} \end{pmatrix}, \begin{pmatrix} b_{11} & \dots & b_{1q} \\ \vdots & \ddots & \vdots \\ b_{p1} & \dots & b_{pq} \end{pmatrix} \right) \mapsto \begin{pmatrix} a_{11}B & \dots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{n1}B & \dots & a_{nm}B \end{pmatrix} \end{gathered} \tag{2.1.18}$$

$$= \begin{pmatrix} a_{11}b_{11} & \dots & a_{11}b_{1q} & \dots & a_{1m}b_{11} & \dots & a_{1m}b_{1q} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & \dots & a_{11}b_{pq} & \dots & a_{1m}b_{p1} & \dots & a_{1m}b_{pq} \\ \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}b_{11} & \dots & a_{n1}b_{1q} & \dots & a_{nm}b_{11} & \dots & a_{nm}b_{1q} \\ \vdots & \ddots & \vdots & & \vdots & \ddots & \vdots \\ a_{n1}b_{p1} & \dots & a_{n1}b_{pq} & \dots & a_{nm}b_{p1} & \dots & a_{nm}b_{pq} \end{pmatrix}. \tag{2.1.19}$$

This map is called the *Kronecker product,* named after Leopold Kronecker (1823-1891), and is used in many fields of science. For $m = q = 1$ this map reduces to the map from Equation 2.1.5 by regarding $n \times 1$ and $p \times 1$-matrices as vectors.

Example 2.1.2 *Consider the matrices* $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, B = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \in L(\mathbf{C}^2)$*, then*

$$\Psi(A \otimes B) = \begin{pmatrix} 1\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 2\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \\ 3\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} & 4\begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix} \in L(\mathbf{C}^4). \tag{2.1.20}$$

If $v = (1, 2)$ *and* $w = (3, 4)$ *then* $\psi(v \otimes w) = (3, 4, 6, 8)$ *so*

$$\Psi(A \otimes B)\psi(v \otimes w) = \begin{pmatrix} 5 & 6 & 10 & 12 \\ 7 & 8 & 14 & 16 \\ 15 & 18 & 20 & 24 \\ 21 & 24 & 28 & 32 \end{pmatrix} \begin{pmatrix} 3 \\ 4 \\ 6 \\ 8 \end{pmatrix} = \begin{pmatrix} 195 \\ 265 \\ 429 \\ 583 \end{pmatrix}, \tag{2.1.21}$$

and,

$$\psi((Av) \otimes (Bw)) = \psi\left(\begin{pmatrix} 5 \\ 11 \end{pmatrix} \otimes \begin{pmatrix} 39 \\ 53 \end{pmatrix}\right) = \begin{pmatrix} 195 \\ 265 \\ 429 \\ 583 \end{pmatrix}, \tag{2.1.22}$$

hence both Equations 2.1.21 and 2.1.22 yield the same result as expected.

So far, we described the relation and properties of the tensor product of vector spaces. We will now zoom in on a particular linear operator, the *partial trace*. For a vector space $\mathcal{X}$ over a field $\mathbf{C}$, the trace operator is defined as a functional, i.e. $\mathrm{Tr}\colon L(\mathcal{X}) \to \mathbf{C}$, so similarly we could define the trace on a tensor product $\mathrm{Tr}\colon L(\mathcal{X} \otimes \mathcal{Y}) \to \mathbf{C}$ for some vector spaces $\mathcal{X}, \mathcal{Y}$ over $\mathbf{C}$. The partial trace is a generalization of this linear operator, that maps to $L(\mathcal{X})$ or $L(\mathcal{Y})$ instead of the field $\mathbf{C}$. As we will see in Section 4 this operation corresponds to describing quantum mechanical subsystems.

Definition 2.1.2 (Partial trace operation) *Let* $\mathcal{X}, \mathcal{Y}$ *be vector spaces over* $\mathbf{C}$*. Then the partial trace over* $\mathcal{Y}$ *is the unique linear operator*

$$\mathrm{Tr}_{\mathcal{Y}}\colon L(\mathcal{X} \otimes \mathcal{Y}) \to L(\mathcal{X}), \tag{2.1.23}$$

defined by

$$\mathrm{Tr}_{\mathcal{Y}}(X \otimes Y) = X \,\mathrm{Tr}(Y), \tag{2.1.24}$$

for all $X \in L(\mathcal{X})$*,* $Y \in L(\mathcal{Y})$*. Similarly we can define the partial trace over* $\mathcal{X}$ *as the unique operator defined by*

$$\begin{aligned} \mathrm{Tr}_{\mathcal{X}}\colon L(\mathcal{X} \otimes \mathcal{Y}) &\to L(\mathcal{Y}) \\ X \otimes Y &\mapsto \mathrm{Tr}(X)Y, \end{aligned} \tag{2.1.25}$$

for all $X \in L(\mathcal{X})$*,* $Y \in L(\mathcal{Y})$*.*

In short, one could write $\mathrm{Tr}_{\mathcal{Y}} = I_{\mathcal{X}} \otimes \mathrm{Tr}$ and $\mathrm{Tr}_{\mathcal{X}} = \mathrm{Tr} \otimes I_{\mathcal{Y}}$.

We can use the association $\mathbf{C}^n \otimes \mathbf{C}^m \cong \mathbf{C}^{nm}$ from Equation 2.1.18 and the standard basis to explicitly express the partial trace operator. For example if $\mathcal{X} = \mathbf{C}^2$ and $\mathcal{Y} = \mathbf{C}^3$, then

$$\begin{aligned}\mathrm{Tr}_{\mathcal{Y}} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{pmatrix} &= \begin{pmatrix} \mathrm{Tr}\begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 13 & 14 & 15 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 4 & 5 & 6 \\ 10 & 11 & 12 \\ 16 & 17 & 18 \end{pmatrix} \\ \mathrm{Tr}\begin{pmatrix} 19 & 20 & 21 \\ 25 & 26 & 27 \\ 31 & 32 & 33 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 22 & 23 & 24 \\ 28 & 29 & 30 \\ 34 & 35 & 36 \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} 24 & 33 \\ 78 & 87 \end{pmatrix} \in L(\mathbf{C}^2).\end{aligned} \tag{2.1.26}$$

Similarly if we take the trace over $\mathcal{X} = \mathbf{C}^2$ we get

$$\begin{aligned}\mathrm{Tr}_{\mathcal{X}} \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 7 & 8 & 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 \\ 25 & 26 & 27 & 28 & 29 & 30 \\ 31 & 32 & 33 & 34 & 35 & 36 \end{pmatrix} &= \begin{pmatrix} \mathrm{Tr}\begin{pmatrix} 1 & 4 \\ 19 & 22 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 2 & 5 \\ 20 & 23 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 3 & 6 \\ 21 & 24 \end{pmatrix} \\ \mathrm{Tr}\begin{pmatrix} 7 & 10 \\ 25 & 28 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 8 & 11 \\ 26 & 29 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 9 & 12 \\ 27 & 30 \end{pmatrix} \\ \mathrm{Tr}\begin{pmatrix} 13 & 16 \\ 31 & 34 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 14 & 17 \\ 32 & 35 \end{pmatrix} & \mathrm{Tr}\begin{pmatrix} 15 & 18 \\ 33 & 36 \end{pmatrix} \end{pmatrix} \\ &= \begin{pmatrix} 23 & 25 & 27 \\ 35 & 37 & 39 \\ 47 & 49 & 51 \end{pmatrix} \in L(\mathbf{C}^3).\end{aligned} \tag{2.1.27}$$

In general, if we have vector spaces $\mathcal{X}_1, \ldots, \mathcal{X}_n$, then the partial trace over $\mathcal{X}_k$ for some $1 \leq k \leq n$ of an linear operator on the tensor product $\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n$ is defined on a basis of the linear operators

$$\begin{aligned}\mathrm{Tr}_{\mathcal{X}_k} \colon L(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n) &\to L(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_{k-1} \otimes \mathcal{X}_{k+1} \otimes \cdots \otimes \mathcal{X}_n) \\ E_{i_1,j_1} \otimes \cdots \otimes E_{i_n,j_n} &\mapsto \mathrm{Tr}(E_{i_k,j_k}) E_{i_1,j_1} \otimes \cdots \otimes E_{i_{k-1},j_{k-1}} \otimes E_{i_{k+1},j_{k+1}} \otimes \cdots \otimes E_{i_n,j_n} \\ &= \begin{cases} E_{i_1,j_1} \otimes \cdots \otimes E_{i_{k-1},j_{k-1}} \otimes E_{i_{k+1},j_{k+1}} \otimes \cdots \otimes E_{i_n,j_n} & \text{if } i_k = j_k \\ 0 & \text{otherwise} \end{cases},\end{aligned}$$

We have seen that there is an isomorphism between $\mathbf{C}^n \otimes \mathbf{C}^m$ and $\mathbf{C}^{nm}$. Furthermore, note that the vector space of $n \times m$-matrices, $L(\mathbf{C}^n, \mathbf{C}^m)$ has the same dimension as $\mathbf{C}^n \otimes \mathbf{C}^m$, and therefore there is an isomorphism between these spaces. The following linear map given by its action on the basis represents such an isomorphism

$$\begin{aligned}\mathrm{vec} \colon L(\mathcal{Y}, \mathcal{X}) &\xrightarrow{\sim} \mathcal{X} \otimes \mathcal{Y} \\ E_{i,j} &\mapsto e_i \otimes e_j.\end{aligned} \tag{2.1.28}$$

This linear map, called the *operator-vector correspondence*, has the following important properties.

First of all, with regard to a general systems of equations, we have the relation

$$(A \otimes C)\,\mathrm{vec}(B) = \mathrm{vec}(ABC^\top). \tag{2.1.29}$$

The partial trace over $\mathcal{X}$ or $\mathcal{Y}$ of an outer product of elements in the tensor product $\mathcal{X} \otimes \mathcal{Y}$ relates to the ordinary matrix product by the equations

$$\mathrm{Tr}_{\mathcal{Y}}(\mathrm{vec}(A)\,\mathrm{vec}(B)^\dagger) = AB^\dagger, \tag{2.1.30}$$

and

$$\mathrm{Tr}_{\mathcal{X}}(\mathrm{vec}(A)\,\mathrm{vec}(B)^\dagger) = A^\top \overline{B}. \tag{2.1.31}$$

All of these properties can be proven by considering a basis of the linear maps. We will prove identity 2.1.30 and identity 2.1.31 will follow similarly by taking $\mathrm{Tr}_{\mathcal{Y}}$ in the last step.

Let $A = (a_{ij})_{i\in[n],j\in[m]}$ and $B = (b_{ij})_{i\in[n],j\in[m]}$ be complex matrices. Then

$$\begin{aligned}
\mathrm{vec}(A)\,\mathrm{vec}(B)^\dagger &= \left(\sum_{i\in[n]}\sum_{j\in[m]} a_{ij} e_i \otimes e_j\right)\left(\sum_{k\in[n]}\sum_{\ell\in[m]} b_{k\ell} e_k \otimes e_\ell\right)^\dagger \\
&= \sum_{i,k\in[n]}\sum_{j,\ell\in[m]} a_{ij}\overline{b_{k\ell}}(e_i \otimes e_j)(e_k \otimes e_\ell)^\top \\
&= \sum_{i,k\in[n]}\sum_{j,\ell\in[m]} a_{ij}\overline{b_{k\ell}}(e_i e_k^\top) \otimes (e_j e_\ell^\top).
\end{aligned} \tag{2.1.32}$$

If we now take the partial trace of the expression with regard to the space $\mathcal{Y}$ we get

$$\begin{aligned}
\mathrm{Tr}_{\mathcal{Y}}(\mathrm{vec}(A)\,\mathrm{vec}(B)^\dagger) &= \sum_{i,k\in[n]}\sum_{j,\ell\in[m]} a_{ij}\overline{b_{k\ell}}(e_i e_k^\top)\underbrace{\mathrm{Tr}(e_j e_\ell^\top)}_{\delta_{j\ell}} \\
&= \sum_{i,k\in[n]}\left(\sum_{j\in[m]} a_{ij}\overline{b_{kj}}\right)(e_i e_k^\top) \\
&= AB^\dagger.
\end{aligned} \tag{2.1.33}$$

Similarly, if we take the partial trace with regard to the space $\mathcal{X}$ we eliminate the matrix $e_i e_k^\top$ and find $A^\top\overline{B}$ as the result.

In semidefinite programming we consider pairs of programs called *primal* and *dual* semidefinite optimisation programs. If the primal program is defined by an operator in $L(\mathcal{X},\mathcal{Y})$, then the dual program is determined by the *adjoint operator* in $L(\mathcal{Y},\mathcal{X})$. Formally the adjoint is defined as follows:

Definition 2.1.3 *Let $\mathcal{X},\mathcal{Y}$ be complex Euclidean spaces and $\Phi \in L(\mathcal{X},\mathcal{Y})$ a linear operator. The adjoint of Φ, denoted by $\Phi^\dagger \in L(\mathcal{Y},\mathcal{X})$, is the unique operator such that for every $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ holds*

$$\langle \Phi(X), Y\rangle = \langle X, \Phi^\dagger(Y)\rangle. \tag{2.1.34}$$

The existence and uniqueness of such an operator is not trivial. In the case of finite dimensional vector spaces the adjoint of an operator is the transpose (real) or Hermitian transpose (complex) of its matrix. The operation of taking the adjoint is conjugate linear on the space of operators as can be easily seen from the definition.

An explicit example of an operator from which we will need the adjoint is the partial trace map. This operator appears in the semidefinite program that encodes the cheating strategy of quantum coin flipping.

Lemma 2.1.1 *Let $\mathcal{X},\mathcal{Y}$ be complex Euclidean spaces. The map*

$$\begin{aligned}
\Omega_{\mathcal{Y}}\colon L(\mathcal{X}) &\to L(\mathcal{X}\otimes\mathcal{Y}) \\
X &\mapsto X \otimes I_{\mathcal{Y}},
\end{aligned} \tag{2.1.35}$$

is the adjoint operator of the partial trace $\mathrm{Tr}_{\mathcal{Y}} \in L(\mathcal{X}\otimes\mathcal{Y},\mathcal{X})$.

Proof: Let $A \in L(\mathcal{X}), B \in L(\mathcal{X}\otimes\mathcal{Y})$, then we can write $B = \sum_{i\in I} X_i \otimes Y_i$ for $X_i \in L(\mathcal{X}), Y_i \in L(\mathcal{Y})$ for some index set I (we could for example take a basis of $L(\mathcal{X})$. Then

$$\begin{aligned}\langle \Omega_{\mathcal{Y}}(A), B\rangle &= \left\langle A \otimes I_{\mathcal{Y}}, \sum_{i\in I} X_i \otimes Y_i \right\rangle = \sum_{i\in I} \langle A \otimes I_{\mathcal{Y}}, X_i \otimes Y_i\rangle = \sum_{i\in I} \langle A, X_i\rangle_{\mathcal{X}} \langle I_{\mathcal{Y}}, Y_i\rangle_{\mathcal{Y}} \\ &= \sum_{i\in I} \langle A, X_i\rangle_{\mathcal{X}} \operatorname{Tr}(Y_i) = \sum_{i\in I} \langle A, X_i \operatorname{Tr}(Y_i)\rangle = \sum_{i\in I} \langle A, \operatorname{Tr}_{\mathcal{Y}}(X_i \otimes Y_i)\rangle \\ &= \left\langle A, \operatorname{Tr}_{\mathcal{Y}} \left(\sum_{i\in I} X_i \otimes Y_i \right) \right\rangle = \langle A, \operatorname{Tr}_{\mathcal{Y}}(B)\rangle.\end{aligned} \tag{2.1.36}$$

Hence the adjoint of the map $\Omega_{\mathcal{Y}}$ is the partial trace over $\mathcal{Y}$. □

As a special case we can consider the 'full' trace map $\operatorname{Tr} : L(\mathcal{X}) \to \mathbf{C}$ (if we associate $L(\mathbf{C})$ with $\mathbf{C}$) and its adjoint

$$\begin{aligned}\Omega_{\mathcal{X}} \colon \mathbf{C} &\to L(\mathcal{X}) \\ z &\mapsto z I_{\mathcal{X}}.\end{aligned} \tag{2.1.37}$$

Another map that is useful is the map that changes the basis of matrix. Let $U \in L(\mathcal{X})$ be a unitary matrix and define $\Phi \in T(\mathcal{X})$ by $\Phi(X) = UXU^\dagger$. Then the adjoint of Φ is $\Phi^\dagger(X) = U^\dagger X U$, which is clear from the fact that

$$\langle UXU^\dagger, Y\rangle = \langle UX, YU\rangle = \langle X, U^\dagger Y U\rangle, \tag{2.1.38}$$

for every $X, Y \in L(\mathcal{X})$.

2.2 Positive Semidefinite Operators on Euclidean Spaces

The main mathematical object of this thesis will be positive semidefinite operators on real and complex Euclidean spaces. These operators will be the decision variables of semidefinite optimisation programs and will be the representative mathematical object for quantum states. We will first of all introduce real positive semidefinite operators and later show that complex semidefinite operators can be related to real operators and share the same properties. We will consider properties of positive semidefinite operators

Definition 2.2.1 *A symmetric operator $X \in \operatorname{Sym}(\mathbf{R}^n)$ is positive semidefinite, denoted by $X \succeq 0$, if for all $v \in \mathbf{R}^n$ we have $v^\top X v \geq 0$.*

Similarly, X is positive definite, denoted by $X \succ 0$, if for all $v \in \mathbf{R}^n \setminus \{0\}$ we have $v^\top X v > 0$.

The sum of two positive semidefinite matrices is again positive semidefinite. Scalar multiples of positive semidefinite matrices are also again positive semidefinite. These properties are immediately clear from the definition and lead to the fact that the set of all positive semidefinite matrices form a cone. However the product of positive semidefinite matrices is not always positive semidefinite. In particular the product is not always symmetric. This is shown by the following example, let

$$X = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}, \quad Y = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}, \tag{2.2.1}$$

then $XY = \begin{pmatrix} 6 & -2 \\ 3 & -1 \end{pmatrix}$. The eigenvalues of X are $0, 5$ and the eigenvalues of Y are $(3 \pm \sqrt{5})/2$.

Definition 2.2.1 can be applied easily in many circumstances to show that an operator is positive semidefinite. However, there are a number of equivalent statements that may be more practical in some cases.

Lemma 2.2.1 (Equivalent statements for positive semidefinite operators) *Let X be a symmetric matrix on $\mathbf{R}^n$, the following statements are equivalent*

1. *X is positive semidefinite.*
2. *All eigenvalues of X are non-negative.*
3. *X has a Cholesky decomposition; i.e., there exists an $n \times k$-matrix L such that $X = LL^\top$.*
4. *X is a Gram matrix, i.e. there exist vectors $v_1, \ldots, v_n \in \mathbf{R}^k$ such that $X_{ij} = v_i^\top v_j$.*
5. *All principal minors of X are non-negative, i.e. for all $I \subseteq [n]$ we have*

$$\det X_{I,I} \geq 0. \tag{2.2.2}$$

Proof: We will prove these statements in a the following way

$$\begin{aligned} 1. \implies 2. \implies 3. \implies 4. \implies 1. \\ 3 \implies 5. \implies 1. \end{aligned} \tag{2.2.3}$$

$1. \implies 2.$ Suppose X is positive semidefinite and let v be an eigenvector corresponding to an eigenvalue λ of X, then

$$0 \leq v^\top X v = x^\top (\lambda v) = \lambda x^\top x = \lambda \|x\|^2, \tag{2.2.4}$$

hence $\lambda \geq 0$.

$2. \implies 3.$ Suppose X has a spectral decomposition $X = \sum_{i=1}^n \lambda_i v_i v_i^\top$, with $\lambda_i \geq 0$ for all $i \in [n]$, if we let $L = (\sqrt{\lambda_1} v_1 \cdots \sqrt{\lambda_n} v_n)$, we have $X = LL^\top$.

$3. \implies 4.$ Suppose we have a Cholesky decomposition of X, i.e. $X = LL^\top$. If we let v_i be the i-th row of L, then clearly $X_{ij} = v_i^\top v_j$.

$4. \implies 1.$ Suppose X is the Gram matrix of $v_1, \ldots, v_n \in \mathbf{R}^k$. Let $y \in \mathbf{R}^n$, then

$$y^\top X y = \sum_{i=1}^n \sum_{j=1}^n y_i X_{i,j} y_j = \sum_{i=1}^n \sum_{i=1}^n y_i v_i^\top v_j y_j = \left\| \sum_{i=1}^n y_i v_i \right\|^2 \geq 0, \tag{2.2.5}$$

and thus X is positive semidefinite.

$3. \implies 5.$ Suppose X has a Cholesky decomposition $X = LL^\top$. Let $I \subseteq [n]$, then $X_{I,I}$ has a Cholesky decomposition of $L_I L_I^\top$, where L_I is created by removing all rows not in I from L. Since $X_{I,I}$ has a Cholesky decomposition we conclude that X is positive semidefinite and thus $\det X_{I,I} \geq 0$.

$5. \implies 1.$ Suppose every principal minor of X is non-negative. We will prove the statement by induction. Clearly the statement holds for every 1×1-matrix, which is just a number. Suppose the statement holds for all $(n-1) \times (n-1)$-matrices.

Suppose X has an eigenvalue $\lambda < 0$ and all other eigenvalues are positive, then $\det X < 0$ and we have a contradiction. So there must be another eigenvalue μ of X for which $\mu \leq 0$. Let x and y be their corresponding orthonormal eigenvectors. We pick a number $r \in \mathbf{R}$ such that the vector $z = x + ry$ has a coordinate equal to zero, say $z_i = 0$ for some index $i \in [n]$. Consider the $(n-1) \times (n-1)$-principal submatrix $Y = X_{[n]\setminus\{i\},[n]\setminus\{i\}}$ of X created by removing both column and row i. Similarly $w = z_{[n]\setminus\{i\}}$ is created by removing the i-th index. We now have

$$w^\top Y w = z^\top A z = x^\top \lambda x + (ry)^\top \mu (ry) = \lambda \|x\|^2 + r^2 \mu \|y\|^2 = \lambda + r^2 \mu < 0, \tag{2.2.6}$$

which contradicts the induction hypothesis. We thus conclude X has to be positive semidefinite. □

The fifth statement of Lemma 5 can be sophisticated in the case of positive definite matrices and is referred to as *Sylvester's criterion*. This criterion does not require to check every principal minor by only the *leading* principal minors.

Theorem 2.2.2 (Sylvester's criterion, J.J. SYLVESTER) *Let X be a symmetric matrix, then X is positive definite if and only if all leading principal minors of X are positive, that is, for all $k \in [n]$*

$$\Delta_k := \det X_{[k],[k]} > 0. \tag{2.2.7}$$

It would be natural to think this theorem extends to positive *semidefinite* matrices too, by replacing positive by non-negative. This, however, is not true. As an example we can consider the matrix

$$X = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & x \end{pmatrix}, \tag{2.2.8}$$

for some $x \in \mathbf{R}$. Then clearly $\Delta_1 = \Delta_2 = \Delta_3 = 0$, independent of x. If x is negative, then clearly $e_3^\top X e_3 = x < 0$ and we thus have to conclude X is *not* positive semidefinite. Algorithmically, this makes a big differences. Sylvester's criterion requires us to calculate n determinants, whereas 2.2.1.5 requires to make $2^n - 1$ calculations of determinants.

The second statement of Lemma 2.2.1 relates the eigenvalues to positive semidefiniteness. The following theorem by SEMYON A. GERSHGORIN (1901-1933) gives a set that is easy to determine, in which the eigenvalues of a real or complex operator must lie. This theorem shows that an operator is positive semidefinite if all the elements in this set have positive real part. This implies that the corresponding operator is positive semidefinite without explicitly calculating the eigenvalues.

Theorem 2.2.3 (Gershgorin circle theorem, S.A. GERSHGORIN, 1931) *Let X be an $n \times n$-matrix over $\mathbf{C}$ (not necessarily symmetric). For $i \in [n]$ define the radius $r_i = \sum_{j=1, j\neq i}^{n} |X_{ij}|$ and the closed discs with radius r_i and center X_{ii} by $B(X_{ii}, r_i) = \{z \in \mathbf{C} : |z - X_{ii}| \leq r_i\}$, called the Gershgorin discs of X.*

Then every eigenvalue of X lies within one of the Gershgorin discs.

Proof: Let λ be an eigenvalue of X and v an eigenvector. We can always choose an eigenvector from the eigenspace with at least one coordinate equal to 1 and all other components having a modulus $|v_i| \leq 1$. If we now look at the eigenvalue equation $Xv = \lambda v$, then for the i-th component

$$\sum_{j=1}^{n} X_{ij} v_j = \lambda v_i = \lambda, \tag{2.2.9}$$

which is equal to

$$X_{ii} + \sum_{\substack{j=1 \\ j\neq i}}^{n} X_{ij} v_j = \lambda. \tag{2.2.10}$$

We can now apply the triangle inequality and see that

$$|\lambda - X_{ii}| = \left| \sum_{j=1, j\neq i}^{n} X_{ij} v_j \right| \leq \sum_{j=1, j\neq i}^{n} |X_{ij}||v_j| \leq \sum_{j=1, j\neq i}^{n} |X_{ij}|, \tag{2.2.11}$$

and thus the result follows. □

An immediate consequence of Gershgorin's circle theorem is a sufficient condition of for an operator to be positive semidefinite.

Corollary 2.2.4 *Let X be a symmetric matrix that is diagonally dominant, i.e. for all $i \in [n]$ we have*

$$X_{ii} \geq \sum_{\substack{j=1 \\ j \neq i}} |X_{ij}|. \tag{2.2.12}$$

Then X is positive semidefinite.

Proof: If X is symmetric, then all eigenvalues are real. Let λ be an eigenvalue of X. Because X is diagonally dominant, we know from Gershgorin's circle theorem (Theorem 2.2.3) that there exists an $i \in [n]$ such that $\lambda \in [X_{ii} - r_i, X_{ii} + r_i]$. Because $X_{ii} \geq r$ we have $\lambda \geq 0$. Hence all eigenvalues are positive and thus X is positive semidefinite. □

The converse of Corollary 2.2.4 is in general not true. An easy example is given by the 3×3 all-ones matrix

$$J_3 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \in L(\mathbf{R}^3). \tag{2.2.13}$$

Clearly J_3 is *not* diagonally dominant. However, the three eigenvalues of J_3 are 3,0,0 and therefore J_3 is positive semidefinite by Lemma 2.2.1.

Another characterisation of positive semidefinite matrices is given by the following lemma, due to Lipót Fejér (1880-1959).

Lemma 2.2.5 *Let X be a symmetric matrix, we have the following equivalence*

1. *X is positive semidefinite;*
2. *For all $Y \succeq 0$ we have $\langle X, Y \rangle \geq 0$.*

Proof: 1. $\implies$ 2. Suppose $X \succeq 0$ and its spectral decomposition is

$$X = \sum_{i=1}^{n} \lambda_i v_i v_i^\top, \tag{2.2.14}$$

where $\lambda_i \geq 0$ for any $i \in [n]$ and let $Y \succeq 0$, then

$$\langle X, Y \rangle = \left\langle \sum_{i=1}^{n} \lambda_i v_i v_i^\top, Y \right\rangle = \sum_{i=1}^{n} \lambda_i \langle v_i v_i^\top, Y \rangle = \sum_{i=1}^{n} \lambda_i v_i^\top Y v_i \geq 0, \tag{2.2.15}$$

since $v_i^\top Y v_i \geq 0$ for any $i \in [n]$.

2. $\implies$ 1. Suppose for all $Y \succeq 0$ we have $\langle X, Y \rangle \geq 0$. In particular for any $y \in \mathbf{R}^n$ the matrix $Y = yy^\top$ is positive semidefinite, therefore

$$\langle X, Y \rangle = \langle X, yy^\top \rangle = y^\top X y \geq 0, \tag{2.2.16}$$

and hence X is positive semidefinite. □

Operators on Euclidean spaces can be extended to operators on the direct sum and tensor product of these spaces. These operators preserve the property of being positive semidefinite, which follows relatively straightforward from the definitions. This property is essential when describing quantum systems and interactions. First we will consider the direct sum of operators.

Lemma 2.2.6 *Let $X \in \mathrm{Sym}(\mathbf{R}^n)$ and $Y \in \mathrm{Sym}(\mathbf{R}^m)$ be symmetric matrices and then the matrix $X \oplus Y \in \mathrm{Sym}(\mathbf{R}^n) \oplus \mathrm{Sym}(\mathbf{R}^m)$ is positive semidefinite if and only if both X and Y are positive semidefinite.*

Proof: " $\Longrightarrow$ " Suppose $X \oplus Y$ is positive semidefinite, let $v \in \mathbf{R}^n$, then

$$(v \oplus 0)^\top (X \oplus Y)(v \oplus 0) = v^\top X v \geq 0. \tag{2.2.17}$$

Similarly, for $u \in \mathbf{R}^m$ we have

$$(0 \oplus u)^\top (X \oplus Y)(0 \oplus u) = u^\top Y u \geq 0, \tag{2.2.18}$$

so $X \succeq 0$ and $Y \succeq 0$.

" $\Longleftarrow$ " Suppose $X, Y \succeq 0$. Let $z \in \mathbf{R}^n \oplus \mathbf{R}^m$, then $z = u \oplus v$ for some $u \in \mathbf{R}^n, v \in \mathbf{R}^m$, and thus

$$z^\top (X \oplus Y) z = u^\top X u + v^\top Y v \geq 0, \tag{2.2.19}$$

and thus $X \oplus Y$ is positive semidefinite. □

In general a matrix of the for $X_1 \oplus X_2 \oplus \cdots \oplus X_m$ is positive semidefinite if all X_i are positive semidefinite for $i \in [m]$.

For the tensor product of linear operators we use the second statement of Lemma 2.2.1. The spectrum of the tensor product is given by the set of pairwise products of the spectra of both operator individually. This is formally stated in the following lemma.

Lemma 2.2.7 *Let $X \in L(\mathbf{R}^n), Y \in L(\mathbf{R}^m)$ be diagonizable operators, with eigenvalues $\lambda_i, i \in [n]$, $\mu_j, j \in [m]$ and eigenvectors $u_i, i \in [n]$, $v_j, j \in [m]$ respectively. The operator $X \otimes Y \in L(\mathbf{R}^n \otimes \mathbf{R}^m)$ has eigenvalues $\lambda_i \mu_j$ and corresponding eigenvectors $u_i \otimes v_j$ for $i \in [n]$ and $j \in [m]$.*

Proof: We can write the spectral decomposition as

$$X = \sum_{i=1}^{n} \lambda_i u_i u_i^\top, \quad Y = \sum_{j=1}^{m} \mu_j v_j v_j^\top, \tag{2.2.20}$$

then

$$X \otimes Y = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_i \mu_j (u_i u_i^\top) \otimes (v_j v_j^\top) = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda_i \mu_j (u_i \otimes v_j)(u_i \otimes v_j)^\top, \tag{2.2.21}$$

from which the statement directly follows. □

It is now easy to see why the tensor product preserves positive semidefiniteness.

Lemma 2.2.8 *Let $X \in L(\mathbf{R}^n)$, $Y \in L(\mathbf{R}^m)$. If X and Y are positive semidefinite, then $X \otimes Y$ is positive semidefinite.*

Proof: Clear. Since the spectrum of $X \otimes Y$ is $\{\lambda_i \mu_j : i \in [n], j \in [m]\}$ and contains only non-negative products. □

Similar to the direct sum, if we have a number of positive semidefinite operators $X_1 \in L(\mathcal{X}_1), \ldots, X_n \in L(\mathcal{X}_n)$ then the operator $X_1 \otimes \cdots \otimes X_n \in L(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n)$ is positive semidefinite as well. This is a direct consequence of applying Lemma 2.2.8 repeatedly.

Chapter 3

Semidefinite Programming

"The price of metaphor is eternal vigilance."

- Norbert Wiener

Semidefinite programming is a relatively new and very powerful optimization formalism. It allows for a flexible description or relaxations of many problems in for example combinatorics and graph theory. Solving a semidefinite program can be done in polynomial running time and is therefore efficient. In this section we look at the descriptions of a general real and complex semidefinite programs, duality theory, examples and methods of solving. We provide the necessary mathematical concepts to understand these subjects.

Useful sources that have been used here are the book on convex optimization by Boyd and Vandenberge [23], lecture notes from the Mastermath course on semidefinite optimization in 2018 [24], and Handbook on semidefinite, conic and polynomial optimization by Anjos and Laserre [25]. Much of the notation is used from Watrous' book on quantum information theory [26].

3.1 Linear Programming

We start with the widely known general *linear program* (LP), which is given by

$$p^* = \sup\{c^\top x : Ax = b, x \in \mathbf{R}^n, x \geq 0\}. \tag{3.1.1}$$

In this program $c \in \mathbf{R}^n$ and $b \in \mathbf{R}^m$ are vectors and $A \in L(\mathbf{R}^n, \mathbf{R}^m)$ is a real $m \times n$-matrix. The decision variable x is a vector in $\mathbf{R}^n$. This optimization program is characterized by the cone of vectors for which every component is positive, denoted by $x \geq 0$. The dual of Program 3.1.1 is given by

$$d^* = \inf\{b^\top y : A^\top y \geq c, y \in \mathbf{R}^m\}. \tag{3.1.2}$$

Many theoretical and practical problems can be formulated as linear programs. Moreover, if we restrict the vectors, x to be integer, i.e., all components are integer, then the problem becomes an *integer linear program* (ILP). Many difficult combinatorial optimization problems can be formulated as an integer linear program such as the travelling salesman problem and chromatic number of a graph.

Linear problems can be solved quickly, that is, there exist an algorithm that solves the problem with a running time that is polynomial in the input size. This can established by the ellipsoid method or an interior point method. However in practice the simplex method works well, contrary to the fact that it has a worst case exponential running time[1].

[1] this result actually depends on the pivoting rules used in the simplex method. It is unknown whether there exist pivoting rules that do lead to polynomial worst case performance.

Contrary to linear programs, integer linear programs are in general NP-hard and therefore cannot be solved quickly unless P $=$ NP.

A natural option would be to look for an approximation of the exact problems with the advantage of solving the problem quickly. Natural relaxations of ILPs to LPs are therefore a good option. However, some problems yield an arbitrary large ratio between the optimal and relaxed solutions, this makes some of the problems less suitable to relax to LPs.

The best option would be to have a programming formulation more sophisticated than linear programming whilst retaining the polynomial running time. One of the options is *semidefinite programming*.

Informally semidefinite programming is the result of three actions: 1) replacing vectors by matrices, 2) regarding a new inner product, now on matrices, and 3) posing a new partial ordering on matrices.

We will define the necessary mathematical structures and the define the general semidefinite program. To begin, we have to define a matrix inner product. We pose the 'inner product' and show that it actually satisfies all requirements of an inner product. The inner product is called the *trace inner product* or *Frobenius inner product* after Ferdinand G. Frobenius (1849 – 1917). We define this inner product for linear operators on a complex Euclidean space. By restricting to the field of real numbers we end up with the same properties. The trace inner product is defined by

$$\begin{aligned} \langle \cdot, \cdot \rangle \colon L(\mathcal{X}) \times L(\mathcal{X}) &\to \mathbf{C} \\ (X, Y) &\mapsto \operatorname{Tr}(X^\dagger Y) = \sum_{i=1}^{n} \sum_{j=1}^{n} \overline{X_{ij}} Y_{ij}. \end{aligned} \tag{3.1.3}$$

This inner product is linear in the second argument, conjugate linear in the first argument and positive definite.

If we want to optimise over complex matrices we do require the inner product to map to $\mathbf{R}$, because the whole space $\mathbf{C}$ cannot be ordered. This can be done by restricting to Hermitian matrices. To show this, suppose that X, Y are Hermitian matrices. By using the fact that $X = X^\dagger$, $Y = Y^\dagger$ and the cyclic property of the trace, it follows that

$$\langle X, Y \rangle = \operatorname{Tr}(X^\dagger Y) = \operatorname{Tr}((X^\dagger Y)^\dagger)) = \overline{\langle X, Y \rangle}. \tag{3.1.4}$$

Since the spaces $L(\mathcal{X})$ and $\mathcal{X} \otimes \mathcal{X}$ correspond to each other using the map vec, we can identify that this forms an isometry, i.e.,

$$\langle X, Y \rangle = \langle \operatorname{vec}(X), \operatorname{vec}(Y) \rangle. \tag{3.1.5}$$

An inner product generates an induced norm, which in this case unsurprisingly is called the *Frobenius norm*, denoted by $\| \cdot \|_F$ or even just $\| \cdot \|$. Therefore

$$\|X\|_F = \sqrt{\langle X, X \rangle} = \left(\sum_{i=1}^{n} \sum_{j=1}^{n} |X_{ij}|^2 \right)^{1/2}. \tag{3.1.6}$$

Finally, we need to introduce a partial order. The set $(\mathbf{R}, \geq)$ forms a totally ordered field, however, the complex numbers can not be totally ordered. Since complex numbers can be represented by operators in $L(\mathbf{R}^2)$, as we will see in Section 3.8, we can not have a total ordering on $L(\mathbf{R}^n)$ for $n \geq 2$ or $L(\mathbf{C}^n)$ for $n \geq 1$.

The partial ordering we will use is called the *Loewner order*, after CHARLES LOEWNER (1893-1968) and is given by the relation $X \succeq Y$ if and only if $X - Y$ is positive semidefinite. A partial order is a relation that is reflexive, antisymmetric and transitive, so for all $X, Y, Z \in L(\mathbf{C}^n)$ we must have:

1. (Reflexivity): $X \succeq X$. This is clearly the case since $X - X = 0 \succeq 0$.
2. (Antisymmetry) Suppose $X \succeq Y$ and $Y \succeq X$. Let $Z = X - Y$, then $Z \succeq 0$ and $Z \preceq 0$. Let $x \in \mathbf{C}^n$, then
$$0 \leq x^\dagger Z x = -x^\dagger(-Z)x \leq 0, \tag{3.1.7}$$
so we conclude that $Z = 0$ and hence $X = Y$.
3. (Transitivity) Suppose $X \succeq Y$ and $Y \succeq Z$, then
$$X - Z = (X - Y) + (Y - Z) \succeq 0, \tag{3.1.8}$$
since the sum of two positive semidefinite matrices is again positive semidefinite. So $X \succeq Z$.

Based on the Loewner partial order we can define inequalities of the form

$$\alpha_1 A_1 + \alpha_2 A_2 + \cdots + \alpha_m A_m \succeq B, \tag{3.1.9}$$

for matrices $A_1, \ldots, A_m, B \in L(\mathbf{C}^n)$ and numbers $\alpha_1, \ldots, \alpha_m \in \mathbf{C}$. Such an inequality is called a *linear matrix inequality* (LMI).

3.2 Semidefinite Programming

Based on the inner product and the Loewner order, we will define a general semidefinite program. We will first do this for real Euclidean spaces and later extend it to complex spaces.

Let $A_1, \ldots, A_m, C \in \mathrm{Sym}(\mathbf{R}^n)$ be real symmetric $n \times n$-matrices and $b_1, \ldots, b_m \in \mathbf{R}$ be real numbers. From this we define a general semidefinite program has the form

$$P^* = \sup\{\langle C, X\rangle : \langle A_i, X\rangle = b_i, i \in \{1, \ldots, m\}, X \succeq 0\}. \tag{3.2.1}$$

The objective of this program is the map $X \mapsto \langle C, X\rangle$ and the constraints are the set of equalities $\langle A_1, X\rangle = b_1, \ldots, \langle A_m, X\rangle = b_m$. The *feasible region* of this program is the set

$$\{X \in L(\mathbf{R}^n) : \langle A_1, X\rangle = b_1, \ldots, \langle A_m, X\rangle = b_m, X \succeq 0\}, \tag{3.2.2}$$

and elements are called *feasible*. If the feasible region of a semidefinite program is empty, then the semidefinite program is called *infeasible* and $P^* = -\infty$.

The dual of a semidefinite program as given in 3.2.1 is given by the program

$$D^* = \inf\left\{\sum_{i=1}^m b_i y_i : \sum_{i=1}^m y_i A_i - C \succeq 0, y \in \mathbf{R}^m\right\}. \tag{3.2.3}$$

Let $C \in \mathrm{Sym}(\mathbf{R}^n)$. We will come back to the importance of the dual program in 3.4.

As an example, we consider the following semidefinite program:

$$\sup\{\langle C, X\rangle : \mathrm{Tr}\, X = 1, X \in \mathrm{Sym}(\mathbf{R}^n), X \succeq 0\}. \tag{3.2.4}$$

Note that the constraint $\mathrm{Tr}(X) = 1$ is equivalent to $\langle I, X\rangle = 1$. The dual semidefinitie program is

$$\inf\{y : yI_n \succeq C, y \in \mathbf{R}\}. \tag{3.2.5}$$

In this case the primal and dual optimal values are attainable and equal. In particular the optimal value is the largest eigenvalue of the matrix C, $P^* = D^* = \lambda_{\max}(C)$.

The optimal value of a semidefinite program is not necessarily attained. Consider for example the program

$$P^* = \inf\left\{\langle E_{11}, X\rangle : \langle E_{12}, X\rangle = 1, \langle E_{21}, X\rangle = 1, X \in L(\mathbf{R}^2), X \succeq 0\right\}. \tag{3.2.6}$$

If $X = \begin{pmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{pmatrix}$, then the first pair of constraints translate to $X_{12} = X_{21} = 1$. The optimal value of this program is $P^* = 0$. To see why, note that for $n \geq 1$ the sequences $X_{11} = 1/n$, $X_{22} = n$ are feasible solutions and thus the infimum of X_{11} is $\lim_{n\to\infty} 1/n = 0$. Yet the optimum cannot be attained, otherwise

$$0 \leq \det X = 0 \cdot X_{22} - 1 = -1 < 0. \tag{3.2.7}$$

The class of semidefinite programs includes linear programs. This can be seen by encoding the linear program 3.1.1 into a semidefinite program by letting $C = \mathrm{diag}(c_1, \ldots, c_n) \in \mathrm{Sym}(\mathbf{C}^n)$ and $A_i = \mathrm{diag}(A_{i1}, \ldots, A_{in}) \in \mathrm{Sym}(\mathbf{R}^n)$. If $X \succeq 0$, then $X_{ii} \geq 0$ for $i \in [n]$. As a result, we associate the diagonal of X with the vector x we have all the requirements. The off-diagonal elements do not play a role anymore.

A more abstract, but equivalent, description is in terms of *super operators*. A super operator is a linear operator on a space of linear operators, i.e., $\Phi : L(\mathcal{X}) \to L(\mathcal{Y})$. The set of linear super operators is denoted by $T(\mathcal{X}, \mathcal{Y})$ or simply $T(\mathcal{X})$ if $\mathcal{X} = \mathcal{Y}$. A graphical representation of the space of super operators is given in Figure 3.1.

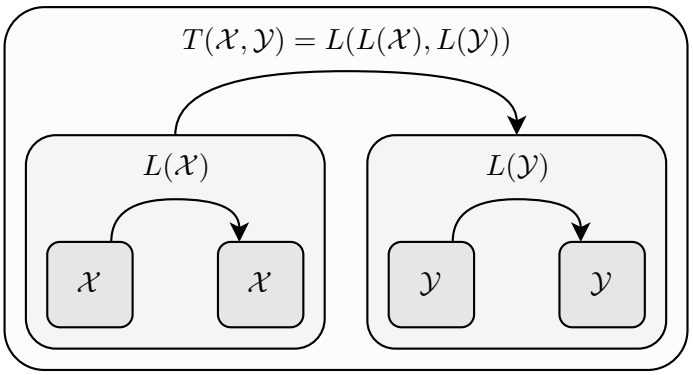


Figure 3.1: The operators in $T(\mathcal{X}, \mathcal{Y})$ map operators that act on the space $\mathcal{X}$ to $\mathcal{Y}$.

The space $T(\mathcal{X}, \mathcal{Y})$ is also a complex Euclidean space and has dimension

$$\dim T(\mathcal{X}, \mathcal{Y}) = \dim L(\mathcal{X}) \cdot \dim L(\mathcal{Y}) = (\dim \mathcal{X})^2 (\dim \mathcal{Y})^2. \tag{3.2.8}$$

Using the map vec, we have the association

$$T(\mathcal{X}, \mathcal{Y}) \cong L(\mathcal{X} \otimes \mathcal{X}, \mathcal{Y} \otimes \mathcal{Y}). \tag{3.2.9}$$

If a super operator maps Hermitian operators to Hermitian operators, i.e., $\Phi(X) \in \mathrm{Herm}(\mathcal{Y})$ for every $X \in \mathrm{Herm}(\mathcal{X})$, then Φ is called *Hermitian preserving*. The introduction of super operators allows us to describe an alternative standard form of a semidefinite program. This form

allows for more efficient description of the problems that arise in quantum information theory, but can always be converted to other standard forms too.

The alternative form can also be applied to real semidefinite programming in which Hermitian matrices reduce to symmetric matrices.

Definition 3.2.1 (Alternative definition of a semidefinite program) *Let $\mathcal{X}, \mathcal{Y}$ be complex Euclidean spaces. A complex semidefinite program is defined by a triple (Φ, C, B), where $C \in \mathrm{Herm}(\mathcal{X})$, $B \in \mathrm{Herm}(\mathcal{Y})$ and $\Phi \in T(\mathcal{X}, \mathcal{Y})$ a Hermitian preserving map, as the program*

$$\sup\{\langle C, X\rangle : \Phi(X) = B, X \in \mathrm{Herm}(\mathcal{X}), X \succeq 0\}. \tag{3.2.10}$$

The dual of this program is

$$\inf\{\langle B, Y\rangle : \Phi^\dagger(Y) \succeq C, Y \in \mathrm{Herm}(\mathcal{Y})\}, \tag{3.2.11}$$

where $\Phi^\dagger$ is the adjoint operator of Φ.

We will first show that this alternative Definition agrees with Definition 3.2.1 by showing that both forms can be transformed into the other.

Suppose we have a semidefinite program in the form of 3.2.1 with a list of constraints $\langle A_i, X\rangle = b_i$ where $A_i \in \mathrm{Herm}(\mathbf{C}^n)$ and $b_i \in \mathbf{R}$ for $i \in [m]$. We can then define $\Phi \in T(\mathbf{C}^n, \mathbf{C}^m)$ as

$$\Phi(X) = \sum_{i=1}^{n} \langle A_i, X\rangle E_{i,i} = \begin{pmatrix} \langle A_1, X\rangle & 0 & \dots & 0 \\ 0 & \langle A_2, X\rangle & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \langle A_m, X\rangle \end{pmatrix} \in \mathrm{Herm}(\mathbf{C}^m). \tag{3.2.12}$$

and

$$B = \sum_{i=1}^{n} b_i E_{i,i} = \begin{pmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & b_m \end{pmatrix} \in \mathrm{Herm}(\mathbf{C}^m). \tag{3.2.13}$$

then the optimization program $\sup\{\langle C, X\rangle : \Phi(X) = B, X \in \mathrm{Herm}(\mathcal{X}), X \succeq 0\}$ in the form of Definition 3.2.1 is equivalent to 3.2.1.

On the other hand, suppose we have a semidefinite program of the form of 3.2.1 and let for $k, \ell \in [m]$:

$$Z_{k\ell} = \begin{cases} E_{kk} & \text{if } k = \ell, \\ \dfrac{1}{\sqrt{2}}(E_{k\ell} + E_{\ell k}) & \text{if } k > \ell, \\ \dfrac{i}{\sqrt{2}}(E_{k\ell} - E_{\ell k}) & \text{if } k < \ell, \end{cases} \tag{3.2.14}$$

then the set $\{Z_{k\ell} : k, \ell \in [m]\}$ is a real orthonormal basis[2] for $\mathrm{Herm}(\mathbf{C}^m)$. Consequently the equation $\Phi(X) = B$ is uniquely determined by the set of equations

$$\langle Z_{k\ell}, \Phi(X)\rangle = \langle Z_{k\ell}, B\rangle, \quad k, \ell \in [m], \tag{3.2.15}$$

[2] Consequently $\dim_{\mathbf{R}}(\mathrm{Herm}(\mathbf{C}^n)) = n^2$.

We can use the adjoint of Φ such that $\langle Z_{k\ell}, \Phi(X)\rangle = \langle \Phi^\dagger(Z_{k\ell}), X\rangle$ and by defining $A_{k\ell} = \Phi^\dagger(Z_{k\ell})$ and $b_{k\ell} = \langle Z_{k\ell}, B\rangle$. We can rephrase the semidefinite program as

$$\begin{aligned} &\sup \langle C, X\rangle \\ &\text{s. t.} \langle A_{k\ell}, X\rangle = b_{k\ell}, \quad k, \ell \in [m], \\ &\quad X \succeq 0, \end{aligned} \tag{3.2.16}$$

which is the standard form of the definition in Equation 3.2.1.

To show that the programs in Definition 3.2.1 are in the same form, we show that every Hermitian matrix can be written as a difference of two positive semidefinite matrices and every matrix inequality can be written into a matrix equality by introducing a slack positive semidefinite matrix.

First, let $Y \in \mathrm{Herm}(\mathcal{Y})$ with spectral decomposition

$$Y = \sum_{i=1}^{m} \lambda_i y_i y_i^\dagger. \tag{3.2.17}$$

Now let

$$P = \sum_{i=1}^{n} \max\{0, \lambda_i\} y_i y_i^\dagger, \quad N = \sum_{i=1}^{n} \max\{0, -\lambda_i\} y_i y_i^\dagger. \tag{3.2.18}$$

Then clearly P, N are positive semidefinite, because all of their eigenvalues are non-negative by construction and furthermore $Y = P - N$. Also note that $PN = 0$. For any Hermitian operator this decomposition is unique and is called the *Jordan-Hahn decomposition*. This means that we can regard every Hermitian operator on $\mathcal{Y}$ as a positive semidefinite operator on the space $\mathcal{Y} \oplus \mathcal{Y}$.

Secondly, we will show that inequality constraints are equivalent to equality constraints by introducing slack operators. Let $X, Y \in \mathrm{Herm}\,\mathcal{X}$. The statement $X \succeq Y$ is equivalent to $X - Y \succeq 0$. This constraints is equivalent to the equality $Z = X - Y$, for some $Z \succeq 0$.

We conclude that both the primal and dual semidefinite programs in Definition 3.2.1 are of the same form because they can be transformed into each other.

3.3 Quadratic Programming Relaxations

Linear programs can be solved efficiently. On the other hand, quadratic programs can not be solved efficiently with currently known algorithms[3]. This major drawback of quadratic programs naturally asks whether we can approximate these programs. In this section we consider a semidefinite relaxation.

A general quadratic program is defined as

$$\inf\{x^\top C x + c^\top x : x^\top A_i x + a_i^\top x = b_i, i \in [m], x \in \mathbf{R}^n\}. \tag{3.3.1}$$

where $A_1, \ldots, A_m, C \in \mathrm{Sym}(\mathbf{R}^n)$ is are symmetric real matrices, $a_1, \ldots, a_m \in \mathbf{R}^n$ vectors and $b_1, \ldots, b_m \in \mathbf{R}$ are real numbers. We can incorporate the decision variable $x \in \mathbf{R}^n$ in the matrix

$$Y = \begin{pmatrix} 1 & x^\top \\ x & xx^\top \end{pmatrix} = \begin{pmatrix} 1 \\ x \end{pmatrix} \begin{pmatrix} 1 \\ x \end{pmatrix}^\top \in L(\mathbf{R}^{n+1}). \tag{3.3.2}$$

[3] Whether efficient algorithms exists is still open

Note that this matrix is symmetric and in particular positive semidefinite. Furthermore $\langle E_{11}, Y\rangle = Y_{11} = 1$. We can rewrite the program as

$$\inf\{\langle C, xx^\top\rangle + c^\top x : \langle A_i, xx^\top\rangle + a_i^\top x = b_i, i \in [m], x \in \mathbf{R}^n\}. \tag{3.3.3}$$

By substituting $xx^\top \in \mathrm{Sym}(\mathbf{R}^n)$ with a positive semidefinite matrix $X \in \mathrm{Sym}(\mathbf{R}^n)$ we relax the problem and find the semidefinite program

$$\inf\left\{\langle C, X\rangle + c^\top x : \langle A_i, X\rangle + a_i^\top x = b_i, i \in [m], \begin{pmatrix} 1 & x^\top \\ x & X \end{pmatrix} \succeq 0, X \in \mathrm{Sym}(\mathbf{R}^n), x \in \mathbf{R}^n\right\}. \tag{3.3.4}$$

Note that the optimization program 3.3.4 is not in standard form, however by adding and transforming the constraint and the objective we find that it is indeed a semidefinite program. If we let the matrices

$$S = \begin{pmatrix} 0 & \frac{1}{2}c^\top \\ \frac{1}{2}c & C \end{pmatrix}, T_i = \begin{pmatrix} 0 & \frac{1}{2}a_i^\top \\ \frac{1}{2}a_i & A \end{pmatrix} \in \mathrm{Sym}(\mathbf{R}^{n+1}), \quad \text{for all } i \in [m], \tag{3.3.5}$$

then the semidefinite program

$$\inf\{\langle S, Y\rangle : \langle T_i, Y\rangle = b_i, i \in [m], \langle E_{11}, Y\rangle = 1, Y \in \mathrm{Sym}(\mathbf{R}^{n+1}), Y \succeq 0\}. \tag{3.3.6}$$

is equivalent to Program 3.3.4.

From this perspective one can view quadratic programming as semidefinite programming with a rank constraint, since adding $\mathrm{rank}(Y) = 1$ and $\langle E_{11}, Y\rangle = 1$ automatically results in a feasible solution of the form in Equation 3.3.2.

3.4 Duality Theory of Semidefinite Programming

The primal and dual programs presented in Equations 3.2.1 and 3.2.3 are defined in general in different spaces. However, the objective values of feasible solutions are related. Every feasible dual solution is an upper bound for the primal optimal solution and every primal feasible solution is a lower bound of the dual optimal solution. This property is called *weak duality*.

More formally, suppose X is a primal feasible solution and y is a dual feasible solution, then

$$\begin{aligned} b^\top y - \langle C, X\rangle &= \sum_{i=1}^m b_i y_i - \langle C, X\rangle = \sum_{i=1}^m \langle A_i, X\rangle y_i - \langle C, X\rangle \\ &= \left\langle \sum_{i=1}^m y_i A_i - C, X \right\rangle \geq 0, \end{aligned} \tag{3.4.1}$$

where in the last inequality we used 2.2.5 and as a result we get $b^\top y \geq \langle C, X\rangle$, so in particular is we consider the supremum and infimum, we have $P^* \leq D^*$. More general, for every primal feasible solution X and every dual feasible solution y, we have

$$\langle C, X\rangle \leq P^* \leq D^* \leq b^\top y. \tag{3.4.2}$$

This means that every dual feasible solution forms an upper bound for the primal problem and vice versa. The *duality gap* of an optimization problem is the non-negative number $D^* - P^*$.

3.4.1 Strong Duality

A particularly interesting situation is when the duality gap is zero, i.e., the primal and dual optimal values are the same.

In many applications of semidefinite programming this is the case and we say that *strong duality* holds. A sufficient condition for strong duality is given by *Slater's condition*, named after Morton Slater. This condition connects existence of interior points of the feasible region of the primal or dual program to strong duality.

Theorem 3.4.1 (Strong duality, Slater's theorem) *Consider the primal and dual standard semidefinite program as presented in equation 3.2.1 and 3.2.3. Then the following statements hold:*

1. *If the primal semidefinite program is bounded from above and there exists a feasible solution $0 \prec X \in \mathrm{Sym}(\mathbf{R}^n)$, then strong duality holds, i.e., $P^* = D^*$. Moreover, there exists a dual feasible solution $y \in \mathbf{R}^m$ such that $\sum_{i=1}^m b_i y_i = D^*$.*
2. *If the dual semidefinite program is bounded from below and there exists an $y \in \mathbf{R}^m$ such that $\sum_{i=1}^m y_i A_i - C \succ 0$, then strong duality holds, i.e., $P^* = D^*$. Moreover, there exists a primal feasible solution $X \in \mathrm{Sym}(\mathbf{R}^n)$ such that $\langle C, X \rangle = P^*$.*

These statements also apply to the alternative Definition 3.2.1, in which we consider a positive definite primal feasible solution $X \in \mathrm{Herm}(\mathcal{X})$ for which $\Phi(X) = B$. Similarly we can consider a dual feasible solution $Y \in \mathrm{Herm}(\mathcal{Y})$ for which $\Phi^\dagger(Y) \succ C$.

3.5 Applications of Semidefinite Programming to Graph Theory

In this Section we will discuss three major applications of semidefinite programming to graph theory: the Lovász ϑ-number for approximating the stability number and chromatic number, the maximum edge biclique problem and the Max-Cut relaxation to a semidefinite program. These applications have lead to the popularity of semidefinite programming and demonstrate its power.

3.5.1 The Lovász ϑ-number

Two important numbers in graph theory are the stability number and the chromatic number of a graph. The stability number $\alpha(G)$ is the largest size of a subset of vertices such that no two vertices in this subset are connected. The chromatic number $\chi(G)$ is the smallest number of colours required to colour every vertex in G such that every pair of vertices that is connected by an edge has a different colour. For these numbers we have the relation

$$\alpha(G) \leq \chi(\overline{G}), \tag{3.5.1}$$

where $\overline{G}$ is the complement graph of G defined by the same vertex set and the complement of the edge set.

Calculating the stability or chromatic number of a graph is an NP-complete problem. This means that it is expected to be impossible to determine efficiently (unless P = NP). An approximation of the stability number of a graph is given by the *Lovász ϑ-number*. This number is the optimal value of a semidefinite program.

Definition 3.5.1 *Let $G = (V, E)$ be a graph, then the Lovász number or Lovász theta function of a graph is defined as*

$$\vartheta(G) = \sup\{\langle J, X \rangle : \mathrm{Tr}(X) = 1, X_{i,j} = 0, \{i, j\} \in E, X \succeq 0\}. \tag{3.5.2}$$

The Lovász ϑ-number is has the following major property.

Theorem 3.5.1 (Lovász sandwich theorem) *For any graph G we have*

$$\alpha(G) \leq \vartheta(G) \leq \chi(\overline{G}). \tag{3.5.3}$$

In particular, for perfect graphs we have $\alpha(G) = \chi(\overline{G})$, so $\vartheta(G) = \alpha(G) = \chi(\overline{G})$, and thus semidefinite optimization gives an exact method for determining the stability number.

Example 3.5.1 *Consider the Petersen graph, shown in Figure 3.2.*

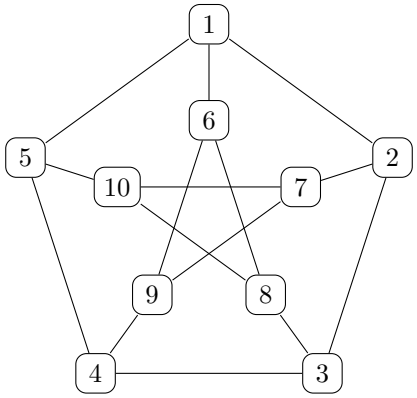


Figure 3.2: The Petersen graph has 10 nodes and 15 edges. Every node has degree three.

The Lovász theta number of this graph is $\vartheta(G) = 4$ and a stable set of G is $\{2, 5, 8, 9\}$. Since $\alpha(G) \leq \vartheta(G) = 4$, this proves this stable set is optimal. Furthermore $\vartheta(\overline{G}) = 5/2$. Coloring G with three colors with different colors for the sets $\{1, 3, 9, 10\}$, $\{2, 4, 6\}$ and $\{5, 7, 8\}$ is valid and since $\chi(G) \geq \vartheta(\overline{G}))$ also optimal.

3.5.2 Semidefinite Programming Relaxations of the Maximum Edge Biclique Problem

Another application of semidefinite programming is found in finding relaxations of the maximum edge biclique problem [27]. In this problem we consider a bipartite graph $G = (V, E)$, i.e., a graph whose vertex set can be partitioned in $V = V_1 \sqcup V_2$ such that there are no edges within V_1 and within V_2.

A biclique is a complete bipartite subgraph where (u_1, u_2) is an edge for every $u_1 \in U_1 \subseteq V_1$ and $u_2 \in U_2 \subseteq V_2$. A complete bipartite graph with a vertex set partition of sizes n and m is denoted by $K_{n,m}$. The maximum edge biclique problem asks to find a biclique that has the maximum number of *edges* in a bipartite graph. This number is denoted by $\kappa(G)$ and is defined by

$$\kappa(G) = \max\{|E(K_{n,m})| : K_{n,m} \subseteq G\}. \tag{3.5.4}$$

Determining $\kappa(G)$ is NP-complete, since the maximum edge biclique problem is a reduction of the clique problem [28]. This is contrary to the fact the maximum *vertex* biclique problem can be solved in polynomial time by using the matching algorithm.

An integer linear program (with quadratic constraint formulation) is given by

$$\begin{aligned}\kappa(G) = \max & \sum_{e\in E} x_e \\ \text{s.t. } & x_{u_1v_1}x_{u_2v_2} = \begin{cases} 0 & \text{if } (u_1,v_2)\notin E \text{ or } (u_2,v_1)\notin E, \\ x_{u_1v_2}x_{u_1v_2} & \text{otherwise,}\end{cases} \\ & \text{for all } (u_1,v_2)\cap(u_2,v_2)=\varnothing, \\ & x_{uv}\in\{0,1\} \text{ for all } (u,v)\in E.\end{aligned} \tag{3.5.5}$$

For any feasible solution $x \in \{0,1\}^E$ of this program, we can define the matrix

$$X = \frac{1}{\sum_{e\in E} x_e} xx^\top \in \mathrm{Sym}(\mathbf{R}^E), \tag{3.5.6}$$

this matrix is positive semidefinite and has rank one. Another property of X is that it's unit trace

$$\langle I, X\rangle = \mathrm{Tr}(X) = \frac{1}{\sum_{e\in E} x_e} x^\top x = \frac{1}{\sum_{e\in E} x_e}\sum_{e\in E} x_e^2 = 1, \tag{3.5.7}$$

since $x_e^2 = x_e$ for all $e \in E$.

We can rewrite the integer linear program in terms of this matrix as follows:

$$\begin{aligned}\kappa(G) = \max & \langle J, X\rangle \\ \text{s.t. } & X_{u_1v_1,u_2v_2} = \begin{cases} 0 & \text{if } (u_1,v_2)\notin E \text{ or } (u_2,v_1)\notin E, \\ X_{u_1v_2,u_1v_2} & \text{otherwise,}\end{cases} \\ & \text{for all } (u_1,v_2)\cap(u_2,v_2)=\varnothing, \\ & \langle I, X\rangle = 1, \\ & X \succeq 0, \\ & X = \frac{1}{\sum_{e\in E} x_e} xx^\top, x_{uv}\in\{0,1\} \text{ for all } (u,v)\in E.\end{aligned} \tag{3.5.8}$$

If we remove the last constraint we get a semidefinite program. The optimal value of this program is denoted by $\sigma(G)$ and we thus have the inequality $\kappa(G) \leq \sigma(G)$. If we also remove the constraint first constraint we find another relaxation that is equal to the lovász ϑ-number of the graph $\Gamma(G)$. $\Gamma(G)$ is defined by the vertices $V(\Gamma(G)) = E$ and $((u_1,v_1),(u_2,v_2)) \in E(\Gamma(G))$ if and only if $G(\{u_1,v_1,u_2,v_2\})$ is a complete bipartite subgraph of G, i.e., isomorphic to $K_{1,2}$ or $K_{2,2} \cong C_4$. This means we have the inequalities[4]

$$\kappa(G) \leq \sigma(G) \leq \vartheta(\Gamma(G)). \tag{3.5.9}$$

The relative sizes between the pair $\kappa(G)$ and $\sigma(G)$ and the pair $\sigma(G)$ and $\vartheta(\Gamma(G))$ can become arbitrarily large, which shows that this approximation might be very bad. Nevertheless, for the three graph numbers κ, σ and ϑ, the bipartite graph product results in the product in of these numbers. This also leads to the tensor product of semidefinite programs, which also has applications in quantum information theory.

3.5.3 The Max-Cut Problem and Semidefinite Relaxation

The third application of semidefinite programming we will discuss is the Max-Cut problem.

[4] Pasechnik calls this type of inequalities *biparite sandwiches*, as a reference to Theorem 3.5.1.

Given a graph $G = (V, E)$ and we want to split the set of vertices in two disjoint subsets such that the number of edges between these parts is as high as possible. More formally let $U \subseteq V$ and define $\delta(U)$ be the set of all edges with one endpoint in U and one in $V \backslash U$, called the *cut*. Thus our optimization problem is

$$\text{Max-Cut}(G) := \max\{|\delta(U)| : U \subseteq V\}. \tag{3.5.10}$$

The feasible region of max-cut is the complete power set of V and has size $2^{|V|}$. Contrary to the min-cut problem, which can be solved in polynomial time, the Max-Cut is an NP-complete problem, in particular it is the last of Karp's 21 NP-complete problems [29]. We can rewrite the problem as an (integer) quadratic program by introducing for a given $U \subseteq V$ the variables $x \in \{-1, 1\}^n$ defined as

$$x_v = \begin{cases} 1 & \text{if } v \in U, \\ -1 & \text{if } v \in V \backslash U. \end{cases} \tag{3.5.11}$$

For $\{u, v\} \in E$ we can look at

$$\frac{1 - x_u x_v}{2} = \begin{cases} 0 & \text{if } u, v \text{ are on the same side,} \\ 1 & \text{if } u, v \text{ are on different sides.} \end{cases}. \tag{3.5.12}$$

We can therefore reformulate the program as

$$\text{Max-Cut}(G) = \max\left\{ \sum_{\{u,v\} \in E} \frac{1 - x_u x_v}{2} : x \in \{-1, 1\}^n \right\}. \tag{3.5.13}$$

If we let $X = xx^\top$, then X has the defining properties

1. $X_{vv} = (x_v)^2 = 1$, for all $v \in V$.
2. X is positive semidefinite.
3. X has rank 1.

we can therefore write 3.5.13 as a semidefinite program with rank constraints. We can relax the problem by deleting the rank constraint and we therefore obtain the semidefinite program

$$\text{SDP}(G) = \max\left\{ \sum_{\{u,v\} \in E} \frac{1 - X_{uv}}{2} : X_{uu} = 1, u \in V, X \succeq 0 \right\}. \tag{3.5.14}$$

If A_G is the adjacency matrix of a graph G, i.e., A_{ij} is 1 if ij is an edge of G and 0 otherwise, and D_G the *degree matrix*, $D_G = \text{diag}(\deg(v_1), \ldots, \deg(v_n))$, then $L_G = D_G - A_G$ is the *Laplacian matrix* of G. We can reformulate 3.5.14 as

$$\text{SDP}(G) = \max\left\{ \frac{1}{4}\langle L_G, X\rangle : X \succeq 0, X_{ii} = 1 \forall i \in [n] \right\}. \tag{3.5.15}$$

An important result is that for any graph G we have the inequality

$$\text{Max-Cut}(G) \leq \text{SDP}(G) \leq 1.1383 \cdot \text{Max-Cut}(G). \tag{3.5.16}$$

Solving $\text{SDP}(G)$ returns a matrix $X \succeq 0$ and not explicitly a description of how to partition the graph in two pieces. To do this we apply the *hyperplane rounding procedure*. This randomized procedure results in vector $x \in \{-1, 1\}^{|V|}$ with corresponding value less than or equal to $1.1383 \cdot$ Max-Cut(G) with high probability. The procedure works as follows. Let X be the Gram matrix of the vectors $\xi_u \in \mathbf{R}, u \in V$, then we pick a random vector r of norm 1 and calculate $x_u = \text{sign}(r^\top \xi_u)$ for all $u \in V$. The sets $\{u \in V : x_u = -1\}$ and $\{u \in V : x_u = 1\}$ form a partition of V. If its cut has a value $\leq 1.1383 \cdot$ Max-Cut(G), we return this cut and otherwise pick another vector r and repeat the procedure.

Example 3.5.2 *If we use the Petersen Graph of Figure 3.2 again. Solving the semidefinite relaxation, we find* $\mathrm{SDP}(G) \leq 25/2$. *If we partition the graph in* $\{1, 3, 4, 6, 7, 10\}$ *and* $\{2, 5, 8, 9\}$, *we find a cut of value* 12. *Since a maximum cut has a value less than or equal to the semidefinite programming relaxtion, this cut is optimal.*

3.6 Grothendieck's Constant

The inequality of the Max-Cut semidefinite program can be extended to a more general setting. In the Max-Cut setting the matrix had to be the Laplace matrix of a graph, but clearly not every matrix is the Laplacian of a graph. If we consider the set of all square $n \times n$ complex matrices $L(\mathbf{C}^n)$, then the corresponding quadratic problem and semidefinite relaxing still differ by a constant factor. The following theorem more formally states this result.

Theorem 3.6.1 (Grothendieck's constant, A. Grothendieck, 1953) *Let* $A \in L(\mathbf{C}^n)$ *and*

$$\left| \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} x_i y_i \right| \leq 1, \tag{3.6.1}$$

for all $x_i, y_i \in \mathbf{R}$ *with* $|x_i|, |y_i| \leq 1$ *for all* $i \in [n]$. *Then there exists a number* $K(\mathbf{C}, n)$ *dependent only on* n *such that*

$$\left| \sum_{i=1}^{n} \sum_{j=1}^{n} A_{ij} \langle X_i, Y_j \rangle \right| \leq K(\mathbf{C}, n), \tag{3.6.2}$$

for all vectors X_i, Y_i *in a complex Hilbert space for which* $\|X_i\|, \|Y_i\| \leq 1$ *for all* $i \in [n]$.

The smallest such constant is called *Grothendieck's constant* after Alexander Grothendieck (1928-2014) and denoted by $K_G(\mathbf{C}, n)$. In particular

$$K_G(\mathbf{C}) = \sup\{K(\mathbf{C}, n) : n \in \mathbf{N}\}. \tag{3.6.3}$$

The exact value of $K_G(\mathbf{C})$ is unknown, but it is known that

$$K_G(\mathbf{C}) < \frac{\pi}{2 \log(1 + \sqrt{2})} \approx 1.7822\ldots \tag{3.6.4}$$

If we restrict ourselves to real numbers and real positive semidefinite operators, similar results exist.

3.7 Solving Semidefinite Programs

Semidefinite programs can be solved in polynomial time. This was first proven using the the ellipsoid method [23]. This method starts with an ellipsoid[5] that contains the feasible region of the semidefinite program. Using a polynomial time oracle based on the objective we can create a new ellipsoid that contains an optimal solution. This ellipsoid has a smaller volume. By repeating this process the volume of the ellipsoid decreases to zero sufficiently fast and the optimum can be approximated. This method requires some numerical stabilization in practice and has a relatively slow running time. The ellipsoid method is powerful theoretical tool, but is not used in practice.

Interior point methods are in practice faster than the ellipsoid method and can for some semidefinite programs informally be described as a combination between Newton's method with constraints [30]. These methods are used in the solvers for convex optimization programs in for

[5] An ellipsoid is a non-singular linear transformation of the unit ball.

example the CVX package, that we will use in this thesis to solve problems.

The previous methods for solving convex optimization problems and in particular semidefinite optimization problems is based on the classical computing. Using quantum computing and objects such as quantum oracles we can get an additional speed up [31, 32]. Currently, there are no quantum computers available with sufficient qubits and fidelity to be able to practically solve optimization problems. As development progresses there might be a moment when it is preferable to use a quantum computer for convex optimization problems.

3.8 Semidefinite Programming over Complex Operators

So far we considered the various definitions, properties and applications of semidefinite optimization over real matrices. However, the application of quantum coin flipping is formulated in terms of complex matrices. In this section we will show that we can switch between real and complex semidefinite optimization by considering complex numbers as a special class of 2×2-matrices over the real numbers and in general the size of the matrices involved increases by a factor of two.

We begin by associating the field of complex numbers with the following field of matrices

$$\Gamma = \left\{ \begin{pmatrix} a & -b \\ b & a \end{pmatrix} : a, b \in \mathbf{R} \right\} \subseteq L(\mathbf{R}^2), \tag{3.8.1}$$

We define the following field isomorphism:

$$\begin{aligned} \psi\colon \mathbf{C} &\xrightarrow{\sim} \Gamma \\ z = a + bi &\mapsto \begin{pmatrix} a & -b \\ b & a \end{pmatrix} = \begin{pmatrix} \mathrm{Re}(z) & -\mathrm{Im}(z) \\ \mathrm{Im}(z) & \mathrm{Re}(z) \end{pmatrix}. \end{aligned} \tag{3.8.2}$$

It can easily be checked, by considering the sum of two elements $z = a + bi, w = c + di \in \mathbf{C}$, that

$$\begin{aligned} \psi(z + w) = \psi((a + c) + (b + d)i) &= \begin{pmatrix} a + c & -(b + d) \\ b + d & a + c \end{pmatrix} \\ &= \begin{pmatrix} a & -b \\ b & a \end{pmatrix} + \begin{pmatrix} c & -d \\ d & c \end{pmatrix} = \psi(z) + \psi(w), \end{aligned} \tag{3.8.3}$$

and

$$\begin{aligned} \psi(zw) = \psi((ac - bd) + (ad + bc)i) &= \begin{pmatrix} ac - bd & -ad - bc \\ ad + bc & ac - bd \end{pmatrix} \\ &= \begin{pmatrix} a & -b \\ b & a \end{pmatrix} \begin{pmatrix} c & -d \\ d & c \end{pmatrix} = \psi(z)\psi(w). \end{aligned} \tag{3.8.4}$$

As a result we for example have $\psi(z^{-1}) = \psi(z)^{-1}$ for all $z \neq 0$, so

$$\psi(z^{-1}) = \psi\left(\frac{1}{a^2 + b^2} a - bi \right) = \frac{1}{a^2 + b^2} \begin{pmatrix} a & b \\ -b & a \end{pmatrix}. \tag{3.8.5}$$

and indeed we see

$$\psi(z^{-1})\psi(z) = \frac{1}{a^2 + b^2} \begin{pmatrix} a & b \\ -b & a \end{pmatrix} \begin{pmatrix} a & -b \\ b & a \end{pmatrix} = \frac{1}{a^2 + b^2} \begin{pmatrix} a^2 + b^2 & 0 \\ 0 & a^2 + b^2 \end{pmatrix} = I. \tag{3.8.6}$$

Note that this map does *not* preserve the inner product. This is a property that we will require for the to associate complex an real matrices and we therefore introduce a factor $\frac{1}{\sqrt{2}}$. For our purposes we only use Hermitian matrices.

For a matrix $X \in \mathrm{Herm}(\mathbf{C}^n)$, let

$$\mathrm{Re}(X) = \frac{X + \overline{X}}{2}, \quad \mathrm{Im}(X) = \frac{X - \overline{X}}{2i}. \tag{3.8.7}$$

It is easy to check that both matrices are real and in particular $\mathrm{Re}(X)$ is symmetric and $\mathrm{Im}(X)$ is skew-symmetric. We now consider the map

$$\begin{aligned} \Psi\colon \mathrm{Herm}(\mathbf{C}^n) &\to \mathrm{Sym}(\mathbf{R}^{2n}) \\ X &\mapsto \frac{1}{\sqrt{2}} \begin{pmatrix} \mathrm{Re}(X) & -\mathrm{Im}(X) \\ \mathrm{Im}(X) & \mathrm{Re}(X) \end{pmatrix}, \end{aligned} \tag{3.8.8}$$

then we have the following important property.

Proposition 3.8.1 *For every $X, Y \in \mathrm{Herm}(\mathbf{C}^n)$, we have*

$$\langle X, Y \rangle = \langle \Psi(X), \Psi(Y) \rangle. \tag{3.8.9}$$

Proof: We can always write $X = \mathrm{Re}(X) + i\,\mathrm{Im}(X)$ and $Y = \mathrm{Re}(Y) + i\,\mathrm{Im}(Y)$. Since X and Y are Hermitian, $\mathrm{Re}(X)$ and $\mathrm{Re}(Y)$ are symmetric and $\mathrm{Im}(X)$ and $\mathrm{Im}(Y)$ are skew symmetric. Another fact we use is that every pair of symmetric and skew-symmetric matrices is orthogonal to each other. The inner product between the matrices is.

$$\begin{aligned} \langle X, Y \rangle &= \langle \mathrm{Re}(X) + i\,\mathrm{Im}(X), \mathrm{Re}(Y) + i\,\mathrm{Im}(Y) \rangle \\ &= \langle \mathrm{Re}(X), \mathrm{Re}(Y) \rangle + i\langle \mathrm{Im}(X), \mathrm{Re}(Y) \rangle - i\langle \mathrm{Re}(X), \mathrm{Im}(Y) \rangle + \langle \mathrm{Im}(X), \mathrm{Im}(Y) \rangle \\ &= \langle \mathrm{Re}(X), \mathrm{Re}(Y) \rangle + \langle \mathrm{Im}(X), \mathrm{Im}(Y) \rangle. \end{aligned} \tag{3.8.10}$$

On the right hand side we find the inner product

$$\begin{aligned} \langle \Psi(X), \Psi(Y) \rangle &= \frac{1}{2} \mathrm{Tr} \left(\begin{pmatrix} \mathrm{Re}(X) & -\mathrm{Im}(X) \\ \mathrm{Im}(X) & \mathrm{Re}(X) \end{pmatrix} \begin{pmatrix} \mathrm{Re}(Y) & -\mathrm{Im}(Y) \\ \mathrm{Im}(Y) & \mathrm{Re}(Y) \end{pmatrix} \right) \\ &= \frac{1}{2} \mathrm{Tr} \begin{pmatrix} \mathrm{Re}(X)\,\mathrm{Re}(Y) - \mathrm{Im}(X)\,\mathrm{Im}(Y) & -\mathrm{Re}(X)\,\mathrm{Im}(Y) - \mathrm{Im}(X)\,\mathrm{Re}(Y) \\ \mathrm{Re}(X)\,\mathrm{Im}(Y) + \mathrm{Im}(X)\,\mathrm{Re}(Y) & -\mathrm{Im}(X)\,\mathrm{Im}(Y) + \mathrm{Re}(X)\,\mathrm{Re}(Y) \end{pmatrix} \\ &= \mathrm{Tr}(\mathrm{Re}(X)\,\mathrm{Re}(Y) - \mathrm{Im}(X)\,\mathrm{Im}(Y)) \\ &= \mathrm{Tr}(\mathrm{Re}(X)\,\mathrm{Re}(Y)^\top) + \mathrm{Tr}(\mathrm{Im}(X)\,\mathrm{Im}(Y)^\top) \\ &= \langle \mathrm{Re}(X), \mathrm{Re}(Y) \rangle + \langle \mathrm{Im}(X), \mathrm{Im}(Y) \rangle. \end{aligned} \tag{3.8.11}$$

We see that both expressions indeed do coincide. □

If we have a general semidefinite program over the complex number of the form

$$(P)\colon \quad \sup\{\langle C, Z \rangle : \langle A_i, Z \rangle = b_i, i \in I, Z \in \mathrm{Herm}(\mathbf{C}^n), Z \succeq 0\}, \tag{3.8.12}$$

then we would like to write is as a double size *real* semidefinite program by applying the map Ψ

$$(P')\colon \quad \sup\{\langle \Psi(C), X \rangle : \langle \Psi(A_i), X \rangle = b_i, i \in I, X \in \mathrm{Sym}(\mathbf{R}^{2n}), X \succeq 0\}. \tag{3.8.13}$$

Clearly, if Z is feasible for (P), then $\Psi(Z)$ is feasible for (P') and has the same objective value due to the fact that Ψ preserves the inner product. On the other hand, if X is feasible for (P'), it does not necessarily have the right *form*. At first, it may seem as if the optimal value (P') may therefore be bigger than (P') and we can only enforce this form by adding more constraints.

The following lemma does however show that adding constraints is not necessary and by applying a transformation to any feasible solution we end up with a feasible solution that does have the right form. In abstract terms, this transformation is given by *integrating over the symmetries of* $\mathrm{im}\,\Psi$.

Lemma 3.8.2 *If X is a feasible solution to (P'), then there exists a feasible solution Y with the same objective value such that $Y \in \mathrm{im}(\Psi)$. Hence solving (P') also solves (P).*

Proof: Let X be a feasible solution of (P'), and consider the following maps $\sigma, \tau \colon L(\mathbf{R}^{2n}) \to L(\mathbf{R}^{2n})$:

$$\sigma : \begin{pmatrix} A & B \\ C & D \end{pmatrix} \mapsto \begin{pmatrix} D & B \\ C & A \end{pmatrix}, \quad \tau : \begin{pmatrix} A & B \\ C & D \end{pmatrix} \mapsto \begin{pmatrix} A & -C \\ -B & D \end{pmatrix}, \tag{3.8.14}$$

where $A, B, C, D \in L(\mathbf{R}^n)$. Note that $G = \{I, \sigma, \tau, \sigma\tau\}$ forms a group acting on real $2n \times 2n$-matrices that is isomorphic to Klein's four group V_4. Now consider the operator

$$Y = \frac{1}{|G|} \sum_{g \in G} g(X), \tag{3.8.15}$$

then

$$\begin{aligned} Y &= \frac{1}{4} \left(\begin{pmatrix} A & B \\ C & D \end{pmatrix} + \begin{pmatrix} D & B \\ C & A \end{pmatrix} + \begin{pmatrix} A & -C \\ -B & D \end{pmatrix} + \begin{pmatrix} D & -C \\ -B & A \end{pmatrix} \right) \\ &= \frac{1}{2} \begin{pmatrix} A+D & B-C \\ C-B & A+D \end{pmatrix} = \Psi \left(\frac{A+D}{\sqrt{2}} + i \frac{C-B}{\sqrt{2}} \right), \end{aligned} \tag{3.8.16}$$

thus Y is in the image of Ψ. We now have to show that the Y is feasible for (P') and has the same objective value.

First, note that for every $X, Y \in L(\mathbf{R}^{2n})$ and $g \in G$

$$\langle X, Y \rangle = \langle g(X), g(Y) \rangle, \tag{3.8.17}$$

since g permutes the position of the matrix elements both matrices and thus the sum of the product of all elements is preserved. Thus, for every $i \in I$

$$\begin{aligned} \langle \Psi(A_i), Y \rangle &= \frac{1}{4} \sum_{g \in G} \langle \Psi(A_i), g(X) \rangle = \frac{1}{4} \sum_{g \in G} \langle g(\Psi(A_i)), g(g(X)) \rangle \\ &= \frac{1}{4} \sum_{g \in G} \langle \Psi(A_i), X \rangle = \langle \Psi(A_i), X \rangle = b_i. \end{aligned} \tag{3.8.18}$$

Similarly

$$\begin{aligned} \langle \Psi(C), Y \rangle &= \frac{1}{4} \sum_{g \in G} \langle \Psi(C), g(X) \rangle = \frac{1}{4} \sum_{g \in G} \langle g(\Psi(C)), g(g(X)) \rangle \\ &= \frac{1}{4} \sum_{g \in G} \langle \Psi(C), X \rangle = \langle \Psi(C), X \rangle. \end{aligned} \tag{3.8.19}$$

As claimed. □

The transformation of the complex semidefinite program to the real semidefinite program also conserves the dual program. With that we mean that applying the transformation and then considering the dual is the same as first considering the dual and then apply the map transformation. This allows for a more flexible choice of solving either the real primal or real dual semidefinite program.

To show this fact is true, we first need the following lemma.

Lemma 3.8.3 *Let $X \in \mathrm{Herm}(\mathbf{C}^n)$, then X is positive semidefinite if and only if $\Psi(X)$ is positive semidefinite.*

Proof: We will use the association $\mathbf{C}^n \cong \mathbf{R}^n \oplus \mathbf{R}^n$. Let $z \in \mathbf{C}^n$ arbitrary and $a = \mathrm{Re}(z), b = \mathrm{Im}(z)$ and define $x = a \oplus b \in \mathbf{R}^{2n}$. We show that

$$z^\dagger X z = \sqrt{2} x^\top \Psi(X) x. \tag{3.8.20}$$

We work out the left hand side. Note that we can write $z = a + ib$ and $X = \mathrm{Re}(X) + i\,\mathrm{Im}(X)$

$$\begin{aligned} z^\dagger X z &= (a^\top - ib^\top)(\mathrm{Re}(X) + i\,\mathrm{Im}(X))(a + ib) \\ &= (a^\top - ib^\top)(\mathrm{Re}(X)a + i\,\mathrm{Re}(X)b + i\,\mathrm{Im}(X)a - \mathrm{Im}(X)b) \\ &= a^\top \mathrm{Re}(X)a - a^\top \mathrm{Im}(X)b + b^\top \mathrm{Re}(X)b + b^\top \mathrm{Im}(X)a \\ &+ i(a^\top \mathrm{Re}(X)b + a^\top \mathrm{Im}(X)a - b^\top \mathrm{Re}(X)a - b^\top \mathrm{Im}(X)b). \end{aligned} \tag{3.8.21}$$

Note that the sum $a^\top \mathrm{Re}(X)b + a^\top \mathrm{Im}(X)a - b^\top \mathrm{Re}(X)a - b^\top \mathrm{Im}(X)b$ consists of only real numbers and thus the sum is real. Since $z^\dagger X z$ is real, it's imaginary part is zero, so

$$z^\dagger X z = a^\top \mathrm{Re}(X)a - a^\top \mathrm{Im}(X)b + b^\top \mathrm{Re}(X)b + b^\top \mathrm{Im}(X)a. \tag{3.8.22}$$

If we evaluate the right hand side we get

$$\begin{aligned} \sqrt{2} x^\top \Psi(X) x &= (a^\top \oplus b^\top) \begin{pmatrix} \mathrm{Re}(X) & -\mathrm{Im}(X) \\ \mathrm{Im}(X) & \mathrm{Re}(X) \end{pmatrix} (a \oplus b) \\ &= (a^\top \oplus b^\top) \begin{pmatrix} \mathrm{Re}(X)a - \mathrm{Im}(X)b \\ \mathrm{Im}(X)a + \mathrm{Re}(X)b \end{pmatrix} \\ &= a^\top \mathrm{Re}(X)a - a^\top \mathrm{Im}(X)b + b^\top \mathrm{Im}(X)a + b^\top \mathrm{Re}(X)b. \end{aligned} \tag{3.8.23}$$

so indeed $z^\dagger X z = \sqrt{2} x^\top \Psi(X) x$ and thus X is positive semidefinite if and only if $\Psi(X)$ is. □

As a consequence, we can apply the map Ψ to the dual complex semidefinite problem too

$$(D): \quad \inf \left\{ \sum_{j=1}^m b_j y_j : \sum_{j=1}^m y_j A_j - C \succeq 0, y_1, \ldots, y_m \in \mathbf{R} \right\}, \tag{3.8.24}$$

which give the program

$$(D'): \quad \inf \left\{ \sum_{j=1}^m b_j y_m : \sum_{j=1}^m y_j \Psi(A_j) - \Psi(C) \succeq 0, y_1 \ldots, y_m \in \mathbf{R} \right\}, \tag{3.8.25}$$

because of the linearity of Ψ the set of numbers $y_1, \ldots, y_m \in \mathbf{R}$ is feasible for (D) if and only if $\sum_{j=1} y_j A_j - C \succeq 0$ and thus if and only if

$$\sum_{j=1} y_j \Psi(A_j) - \Psi(C) = \Psi\left(\sum_{j=1} y_j A_j - C \right) \succeq 0, \tag{3.8.26}$$

which means that the solution $y_1, \ldots, y_m$ is also feasible for (D').

This gives us a '*commutative diagram*' of the form

$$\begin{array}{ccc} (P) & \xleftrightarrow{\text{dual}} & (D) \\ \Big\downarrow{\scriptstyle \Psi} & & \Big\downarrow{\scriptstyle \Psi} \\ (P') & \xleftrightarrow{\text{dual}} & (D') \end{array}$$

Chapter 4

Quantum Mechanics and Quantum Information Theory

> "Quantum mechanics: *a bunch of positive semidefinite things that interact with other positive semidefinite things in some kind of linear way.*"
>
> - Jamie Sikora

Quantum mechanics is the collection of theories in physics that describe interactions at the very small scale. These interactions are fundamentally different from what we experience on the macroscopic scale and lead to phenomena that have no analogous effect in classical mechanics. In this section we consider the postulates of quantum mechanics in terms of linear algebra and functional analysis. This toolbox of mathematical objects gives us the possibility of describing a number of optimization problems in quantum information theory, including quantum coin flipping.

This chapter is mainly based on the book of Nielsen and Chuang [33], which is one of the standard works in this field, the book on quantum mechanics by Griffiths [34] and the recently published book on quantum information theory by Watrous [26].

4.1 The Postulates of Quantum Mechanics

In this section we describe the postulates of quantum mechanics. Contrary to classical mechanics, the description and phenomena of quantum mechanics are hard if not impossible to intuitively grasp. Consequently there are phenomena which do not have a classical analogue. It is therefore a natural to resort to a mathematical description that allows for a rigorous treatment and analysis. The four postulates describe what mathematical structure a quantum mechanical system has, how it evolves over time, how to extract information from the system by measuring and how to describe systems consisting of multiple information carriers.

The first postulate of quantum mechanics describes the mathematical structure of the state of a system at a specific moment in time. All of the other postulates build upon this postulate so it is natural that this postulate is considered first.

Postulate I: Quantum mechanical systems are described by Euclidean spaces over the field of complex numbers. The state of a quantum mechanical object is fully described by a unit vector in this space.

Postulate I describes the state space as a finite dimensional space. In general this does not have to be the case, but is sufficient for applications in quantum information theory. This means that

in general any state space is isomorphic to $\mathbf{C}^n$ for some $n \geq 0$. If the Euclidean space of a system is (isomorphic to) $\mathbf{C}^2$, we call such a object an *qubit*. This is an analogy to the two state classical bits, whose states are described by the finite field $\{0, 1\}$.

The notation of quantum mechanics differs slightly from linear algebra. Vectors (states) are written in *ket* notation: $|\psi\rangle \in \mathcal{X}$. If $\mathcal{X} = \mathbf{C}^2$ then the standard basis of this space is written as $|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Consequently, every qubit can be written as

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \tag{4.1.1}$$

for some complex numbers $\alpha, \beta \in \mathbf{C}$ satisfying $|\alpha|^2 + |\beta|^2 = 1$.

Elements from the dual vector space $\mathcal{X}^\dagger$ of $\mathcal{X}$ are written in *bra* notation: $\langle\psi| \in \mathcal{X}^\dagger$. A bra vector is the Hermitian transpose of a ket vector. If we apply a bra to a ket we get an inner product in this space is, which is a *bracket* and is denoted by $\langle\psi_1|\psi_2\rangle$.

For a given quantum mechanical system, we want to alter the state of let it evolve over time. Such an evolution has to satisfy some properties: it has to be linear, it should map unit vectors to unit vectors in the same space and the evolution has to be reversible. Mathematically, these requirements are fulfilled exactly by unitary operations on this space. The second postulate of quantum mechanics states that all unitary operators are valid transformations of the system.

> **Postulate II:** Evolution of a quantum state is described by unitary transformations on the state space.

A special kind of evolution is time evolution. Time evolution is also given by a unitary operation but is physically described by the interactions with its environment. The *Hamiltonian* of a system can be regarded as the 'energy' of the system and the evolution is given by a partial differntial equation in Postulate II'.

> **Postulate II':** Time evolution of a quantum mechanical systems is described by *Schrödinger's equation*:
>
> $$i\hbar\frac{\partial\,|\psi(x,t)\rangle}{\partial t} = H\,|\psi(x,t)\rangle\,, \tag{4.1.2}$$
>
> where H is the Hamiltonian of the system and $\hbar$ is the reduced Planck constant.

The corresponding time transformation is the unitary operator

$$U_t = \exp(-itH/\hbar), \tag{4.1.3}$$

where $\exp$ is the exponential operator on operators defined by

$$\begin{aligned} \exp\colon L(\mathcal{X}) &\to L(\mathcal{X}) \\ X &\mapsto \sum_{i=1}^{\infty} \frac{1}{n!}X^n = I + X + \frac{1}{2}X^2 + \frac{1}{6}X^3 + \cdots \end{aligned} \tag{4.1.4}$$

For example, the Hamiltonian of an electron in a hydrogen atom is given by

$$H = \frac{p^2}{2m} + V(x) = -\frac{\hbar^2}{2m}\nabla^2 - \frac{e^2}{4\pi\varepsilon_0\|x\|}, \tag{4.1.5}$$

where $p = -i\hbar\partial_t$ is the impulse operator, m the mass of the electron and $V(x)$ the Coulomb potential for any $x \in \mathbf{R}^3$. Moreover, $\nabla^2 = \partial_x^2 + \partial_y^2 + \partial_z^2$ the Laplacian operator, e the charge

of a proton and ε_0 the electric permittivity of the vacuum. Note that the solution to this system is a function of the spatial coordinates and time to $\mathbf{C}$, this means the state space is an infinite dimensional Hilbert space.

For finite dimensional systems, we can describe evolutions of the system by unitary matrices. Suppose we have a one qubit system with standard basis $\{|0\rangle, |1\rangle\}$, then the *Hadamard transform* is the operation

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \in L(\mathbf{C}^2). \tag{4.1.6}$$

This gate can be used to create systems in a superposition (with respect to the standard basis).

Another set of useful operations are the Pauli matrices, defined by

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \in L(\mathbf{C}^2). \tag{4.1.7}$$

The Hadamard and Pauli-matrices act on single qubits. A gate that acts on two qubits is for example the CNOT-gate, which stand for controlled-not-gate, defined by

$$\mathrm{CNOT} = |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes X = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{4.1.8}$$

If we look at its action on the standard basis $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ we see that if the first qubit is in state $|0\rangle$, then the state of the second qubit is not changed. However if the first qubit is in state $|1\rangle$, we apply an X-gate on the second qubit. Because it is sufficient (by linearity) to define gates on a basis, we have fully characterized the CNOT-gate.

Three CNOT gates can be combined to form a SWAP gate as represented in the following circuit.

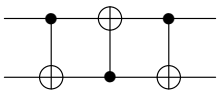


Figure 4.1: Implementation of a SWAP-gate by three alternating CNOT gates.

To see why this works, one can easily apply the three gates to a state in the standard basis. The matrix representation of the SWAP-gate on two qubits is

$$\mathrm{SWAP}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{4.1.9}$$

Contrary to evolution, measurements are not always reversible. Measuring the system may *change* the system. The outcome of a measurement is not always predetermined and is the outcome of a random process. After measuring the system adopts a state that corresponds to this outcome in such a way that immediately measuring again yields the same result. This phenomena of irreversible changing the state of the system is called *the collapse of the wave function*.

Example 4.1.1 *Suppose we have a qubit in the state*

$$\frac{1}{2}|0\rangle + \frac{1}{2}\sqrt{3}|1\rangle, \tag{4.1.10}$$

then measuring in the basis $\{|0\rangle, |1\rangle\}$ *yield the outcome* 0 *with probability* $(1/2)^2 = 1/4$ *and the outcome* 1 *with probability* $(\sqrt{3}/2)^2 = 3/4$*. If the outcome was 1, then the current state of the system would have collapsed and now is* $|1\rangle$*.*

Mathematically, the formalism of measuring is given by Postulate III.

Postulate III: A measurement is described by a set of operators $\{M_a : a \in \Sigma\}$ on the state space of the system for some set of measurement outcomes Σ, satisfying the *completeness relation*

$$\sum_{a\in\Sigma} M_a^\dagger M_a = I. \tag{4.1.11}$$

on the state space. The probability of measuring $a \in \Sigma$ is

$$p(a) = \langle\psi| M_a^\dagger M_a |\psi\rangle, \tag{4.1.12}$$

and the state after the measuring outcome a with $p(a) \neq 0$ is

$$|\psi_a\rangle = \frac{1}{\sqrt{p(a)}} M_a |\psi\rangle. \tag{4.1.13}$$

The completeness axiom is a direct result of the fact that the probability of finding any outcome is 1

$$\sum_{a\in\Sigma} p(a) = \sum_{a\in\Sigma} \langle\psi| M_a^\dagger M_a |\psi\rangle = \langle\psi| \sum_{a\in\Sigma} M_a^\dagger M_a |\psi\rangle = \langle\psi| I |\psi\rangle = 1. \tag{4.1.14}$$

because $|\psi\rangle$ is a unit vector in its state space, the state after measuring is a quantum state because the norm of the state is one

$$\langle\psi_a|\psi_a\rangle = \frac{1}{p(a)} \langle\psi_a| M_a^\dagger M_a |\psi\rangle = \frac{1}{p(a)} p(a) = 1. \tag{4.1.15}$$

If we are not interested in the state after the measurement, then the only important operators are $\{M_a^\dagger M_a : a \in \Sigma\}$. Note that the operators $M_a^\dagger M_a$ are positive semidefinite because of their Cholesky decomposition as shown in Lemma 2.2.1. In particular we can define a set of positive semidefinite operators $\{\Pi_a : a \in \Sigma\}$ with the property

$$\sum_{a\in\Sigma} \Pi_a = I, \tag{4.1.16}$$

to be the measurement operators. The probability of the measurement is

$$p(a) = \langle\psi| \Pi_a |\psi\rangle. \tag{4.1.17}$$

Finally, we have to explain how to describe combinations of multiple quantum systems and interactions. This is where the tensor product of Euclidean spaces an all its properties come into play.

Postulate IV: Multiple quantum mechanical systems can be described as one by the tensor product of their Euclidean spaces. The possible states are unit vectors within this tensor product space.

If $\mathcal{X}_1, \ldots, \mathcal{X}_n$ are the state spaces of the individual systems, then the combined system has a state space

$$\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n, \tag{4.1.18}$$

and any state is described by a linear combination of simple tensors

$$|\psi_1\rangle \otimes \cdots \otimes |\psi_n\rangle\,, \tag{4.1.19}$$

for $|\psi_1\rangle \in \mathcal{X}_1, \ldots, |\psi_n\rangle \in \mathcal{X}_n$.

A simple tensor of quantum states $|\psi_1\rangle \otimes |\psi_2\rangle$ is also often written like $|\psi_1\rangle\,|\psi_2\rangle$ or simply $|\psi_1\psi_2\rangle$.

In particular the state space of an n-qubit system is interesting from the perspective of quantum computing and quantum information theory, this is described by the Euclidean space

$$\mathcal{X} = \underbrace{\mathbf{C}^2 \otimes \cdots \otimes \mathbf{C}^2}_{n} \cong \mathbf{C}^{2^n}\,. \tag{4.1.20}$$

and a basis is given by $\{|b\rangle : b \in \{0,1\}^n\}$.

In general the state space of the combined system is isomorphic to $\mathbf{C}^m$, where $m = \dim(\mathcal{X}_1) \cdot \ldots \cdot \dim(\mathcal{X}_n)$.

Unitary operations allow for a lot of flexibility in transforming the state space. The second postulate states that we can also transform the combined system with unitary operations on $\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n$. However, operations that are common in classical computing do not always extend in a similar way to quantum computing, because of the restriction that evolution has to be unitary. One of these operations is copying information. The impossibility of copying an arbitrary quantum state is given by the *no cloning theorem*.

Theorem 4.1.1 (No cloning theorem) *Let $\mathcal{X}$ be a complex Euclidean space, then does not exist a unitary operation U on $\mathcal{X} \otimes \mathcal{X}$ with the property*

$$U\colon\ |\psi\rangle\,|0\rangle \mapsto |\psi\rangle\,|\psi\rangle\,, \tag{4.1.21}$$

for every $|\psi\rangle \in \mathcal{X}$.

Proof: Suppose such a unitary operation U does exist and let $|0\rangle\,, |1\rangle$ be two basis states of $\mathcal{X}$. U is unitary so by linearity we find

$$U(|0\rangle + |1\rangle)\,|0\rangle = U\,|0\rangle\,|0\rangle + U\,|1\rangle\,|0\rangle = |0\rangle\,|0\rangle + |1\rangle\,|1\rangle\,. \tag{4.1.22}$$

However, we can also apply U directly:

$$U(|0\rangle + |1\rangle)\,|0\rangle = (|0\rangle + |1\rangle)(|0\rangle + |1\rangle) = |00\rangle + |01\rangle + |10\rangle + |11\rangle\,. \tag{4.1.23}$$

It is clear from Eq. 4.1.22 and 4.1.23 that U is in fact not linear, so such a U does indeed not exist. □

Summarized we have the following correspondences between the physical phenomena and the mathematical descriptions

	Physical phenomena		Mathematical description
I	States	$\leftrightarrow$	Unit vectors in a complex Euclidean space,
II	Evolution	$\leftrightarrow$	Unitary operators,
II'	Time evolution	$\leftrightarrow$	Schrödinger's equation,
III	Measurements	$\leftrightarrow$	Positive semidefinite operators,
IV	Multiple systems	$\leftrightarrow$	Tensor products.

In the following sections and Chapters we will build structures on this basis and exploit the mathematical properties to reason about its properties. In particular the density matrix formalism will allow us to extend our toolbox by describing ensembles and ways to ignore parts of the system.

V	Ensembles	$\leftrightarrow$	Unit trace, positive semidefinite operators;
VI	Ignoring systems	$\leftrightarrow$	Partial trace.

We will first consider one of the consequences that do not have a classical analogue.

4.2 Entanglement of Quantum Mechanical States

An entangled state is a state that does not have an analogue in classical mechanics. One qubit cannot be fully described without referring to another qubit. We present this concept by considering the following two qubit state (in $\mathbf{C}^2 \otimes \mathbf{C}^2$):

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle). \tag{4.2.1}$$

This state has norm one and is therefore a valid quantum mechanical state of the system. Suppose that this system can be written like $|\psi\rangle = |\psi_1\rangle\,|\psi_2\rangle$ for some $|\psi_1\rangle\,, |\psi_2\rangle \in \mathbf{C}^2$. Then by the postulates of quantum mechanics, doing any measurements or operations on the individual qubits does not affect the other. Assume $|\psi_i\rangle = \alpha_i\,|0\rangle + \beta_i\,|1\rangle$ for some $\alpha_i, \beta_i \in \mathbf{C}$, $i \in \{1, 2\}$ normalized. Then

$$\begin{aligned} |\psi\rangle &= (\alpha_1\,|0\rangle + \beta_1\,|0\rangle)(\alpha_2\,|0\rangle + \beta_2\,|1\rangle) \\ &= \alpha_1\alpha_2\,|0\rangle\,|0\rangle + \alpha_1\beta_2\,|0\rangle\,|1\rangle + \beta_1\alpha_2\,|1\rangle\,|0\rangle + \beta_1\beta_2\,|1\rangle\,|1\rangle\,. \end{aligned} \tag{4.2.2}$$

In Equation 4.2.1 we see that the states $|0\rangle\,|1\rangle$ and $|1\rangle\,|0\rangle$ do not appear. If $|0\rangle\,|1\rangle$ does not appear, then $\alpha_1\beta_2 = 0$, hence $\alpha_1 = 0$ or $\beta_2 = 0$. If $\alpha_1 = 0$, then $|0\rangle\,|0\rangle$ has amplitude 0, but this is not the case. However, if $\beta_2 = 0$, then $|1\rangle\,|1\rangle$ has amplitude 0, which is also not the case. We conclude that $|\psi\rangle$ is *not* the tensor product of two single qubit states. As a consequence, if we measure the first qubits in the standard basis we have an equal probability of $1/2$ for finding the outcome 0 or 1 and the system will collapse to the state $|0\rangle\,|0\rangle$ of $|1\rangle\,|1\rangle$ respectively. This means that the state of the second system was also affected whilst only performing a physical measurement of the first qubit.

This has rather big implications. Suppose we bring both qubits in the shared state $|\psi\rangle$ and we physically separate them. Because we cannot determine the outcome before, there is not actual transfer of information, but the collapse of the wave function is instantaneously, no matter how big the distance between the qubits is. This phenomena in quantum mechanics has be experimentally confirmed [35].

4.3 The Density Operator Formalism

The state vector of a quantum mechanical systems gives all the possible information there is to know about one specific state of the system. However, sometimes it is possible that randomness

is present in a higher level in the system. For example when there is probability distribution of sending a specific state. In this case the we cannot describe the state of the system solely by a ket-vector in the same space, but we need an *ensemble*. An ensemble is a collection of pairs $(p_i, |\psi_i\rangle)$ for $i \in I$ with I some index set, where p_i is the probability of having state $|\psi_i\rangle \in \mathcal{X}$.

To describe an ensemble we use the *density operator* or *density matrix* formalism of quantum mechanics. This operator describes all the information of the system.

Definition 4.3.1 (Density operator) *Suppose we have an ensemble with states $|\psi_i\rangle \in \mathcal{X}$ and probability p_i for all $i \in I$, where I is some index set. Then the density matrix of this ensemble is*

$$\rho = \sum_{i \in I} p_i \, |\psi_i\rangle \, \langle\psi_i| \in L(\mathcal{X}). \tag{4.3.1}$$

The immediate thing to notice is that Definition 4.3.1 is an eigenvalue decomposition of the density operator ρ, if the states in $\{|\psi_i\rangle : i \in I\}$ are orthogonal. If there is a state $|\psi\rangle \in \mathcal{X}$ such that $\rho = |\psi\rangle \langle\psi|$ we call ρ a *pure state* and a *mixed state* otherwise. Algebraically, pure states are the density operators that have rank 1.

Density matrices have two important properties that follow easily form the definition:

1. The matrix ρ has trace 1.

   Proof: By using the linearity of the trace we get

   $$\begin{aligned} \operatorname{Tr} \rho &= \operatorname{Tr} \left(\sum_{i \in I} p_i \, |\psi_i\rangle \, \langle\psi_i| \right) = \sum_{i \in I} p_i \operatorname{Tr} |\psi_i\rangle \, \langle\psi_i| \\ &= \sum_{i \in I} p_i \langle\psi_i|\psi_i\rangle = \sum_{i \in I} p_i = 1. \end{aligned} \tag{4.3.2}$$

2. The matrix ρ is positive semidefinite.

   Proof: Let $|\varphi\rangle$ be a state from the system, then:

   $$\begin{aligned} \langle\varphi| \, \rho \, |\varphi\rangle &= \langle\varphi| \left(\sum_{i \in I} p_i \, |\psi_i\rangle \, \langle\psi_i| \right) |\varphi\rangle = \sum_{i \in I} p_i \langle\varphi|\psi_i\rangle\langle\psi_i|\varphi\rangle \\ &= \sum_{i \in I} p_i |\langle\varphi|\psi_i\rangle|^2 \geq 0. \end{aligned} \tag{4.3.3}$$

Conversely, if these two properties are met, then the operator is also a density matrix. If an operator $X \in L(\mathbf{C}^n)$ has a spectral decomposition $X = \sum_{i=1}^{n} \lambda_i \, |\varphi_i\rangle \, \langle\varphi_i|$, then X is the density matrix of the ensemble $\{(\lambda_i, |\varphi_i\rangle) : i \in I\}$. This allows us to define the set of all density matrices of a quantum mechanical system with Euclidean space $\mathcal{X}$ by $D(\mathcal{X}) = \{X \in L(\mathcal{X}) : X \succeq 0, \operatorname{Tr} X = 1\}$. The geometric name of this object is the *complex spectraplex*, it is the semidefinite analogue of the n-simplex

$$\{(x_1, \ldots, x_{n+1}) \in \mathbf{R}^{n+1} : x_1 + \ldots + x_{n+1} = 1, x_1, \ldots, x_{n+1} \geq 0\}. \tag{4.3.4}$$

The set $D(\mathcal{X})$ is *not* a vector space. Scaling a density matrix also scales the trace and thus the result is in general not of unit trace anymore. The set $D(\mathcal{X})$ is a convex set, which is a simple consequence of the fact that the cone of semidefinite matrices is convex and the trace map is linear. The extreme points of $D(\mathcal{X})$ are the density matrices that are pure states.

A visual description of pure and mixed states can be constructed by the *Bloch sphere*. To start, the set of density operators on $\mathcal{X} = \mathbf{C}^2$. Every density matrix $\rho \in D(\mathbf{C}^2)$ can be written as

$$\rho = \frac{1}{2}(I_2 + a_1 X + a_2 Y + a_3 Z), \quad a_1, a_2, a_3 \in \mathbf{R}\,. \tag{4.3.5}$$

The vector $a = (a_1, a_2, a_3) \in \mathbf{R}^3$ is called the *Bloch vector* of ρ. The eigenvalues of ρ are $\frac{1}{2}(1+\|a\|)$ and $\frac{1}{2}(1-\|a\|)$. Since $\rho \succeq 0$, we have $\|a\| \leq 1$. Pure states correspond to Bloch vectors with norm 1 and can be parametrized by the states

$$\begin{aligned} |\psi\rangle &= \cos(\theta/2)\,|0\rangle + \exp(i\varphi)\sin(\theta/2)\,|1\rangle \\ &= \cos(\theta/2)\,|0\rangle + (\cos(\varphi) + i\sin(\varphi))\sin(\theta/2)\,|1\rangle\,, \end{aligned} \tag{4.3.6}$$

for some $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi)$. The Bloch-vector of this state is given by

$$\begin{aligned} a = (a_1, a_2, a_3) &= (\langle\psi|\,X\,|\psi\rangle\,, \langle\psi|\,Y\,|\psi\rangle\,, \langle\psi|\,Z\,|\psi\rangle) \\ &= (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta) \in \mathbf{R}^3\,. \end{aligned} \tag{4.3.7}$$

A visual representation is shown in Figure 4.2.

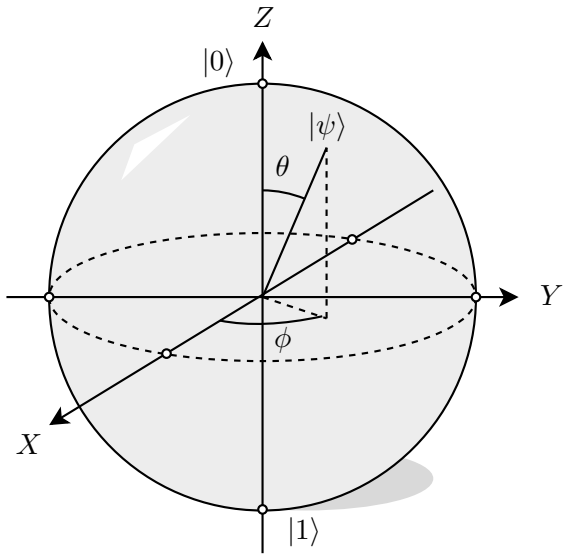


Figure 4.2: The Bloch sphere in (real) three dimensional space. The poles in the Z-direction correspond (Up to multiplication with complex number on the unit circle in $\mathbf{C}$.) to the states $|0\rangle$ and $|1\rangle$, in the X-direction to $(|0\rangle + |1\rangle)/\sqrt{2}$, $(|0\rangle + |1\rangle)/\sqrt{2}$ and in the Z-direction $(|0\rangle + i\,|1\rangle)/\sqrt{2}$, $(i\,|0\rangle + |1\rangle)/\sqrt{2}$. Mixed states exist within the *Bloch ball*.

All the postulates of quantum mechanics can be rephrased in terms of the density operator. At first, it might seem that the density operators formalism is more general than state vector formalism. This is true if we consider the same Euclidean space. However, if we consider a bigger Euclidean space, then the formalisms do coincide, which is represented by the existence of purifications we will encounter later (Theorem 4.3.3).

For evolution of the system we apply a unitary operation U on both sides of the matrix, that is

$$\rho \mapsto U\rho U^\dagger. \tag{4.3.8}$$

This is justified because if we look at the Postulate II and apply the map U to every state individually, we get $|\psi_i\rangle \mapsto U|\psi_i\rangle$ we get

$$\begin{aligned}\rho \mapsto \sum_{i\in I} p_i (U|\psi_i\rangle)(U|\psi_i\rangle)^\dagger &= \sum_{i\in I} p_i U|\psi_i\rangle\langle\psi_i| U^\dagger \\ &= U\left(\sum_{i\in I} p_i |\psi_i\rangle\langle\psi_i|\right) U^\dagger = U\rho U^\dagger.\end{aligned} \tag{4.3.9}$$

Similarly, we can rephrase Schrödinger's equation in terms of density matrices.

Proposition 4.3.1 (Von Neumann equation) *The time evolution of the density matrix is given by the von Neumann equation*

$$i\hbar\frac{\partial\rho}{\partial t} = [H,\rho] = H\rho - \rho H, \tag{4.3.10}$$

where H is the Hamiltonian of the system.

Proof: We work out both sides of the commutator $[H,\rho] = H\rho - \rho H$. We do use the Schrödinger equation of Postulate II' for *ordinary* ket-vectors, so

$$\begin{aligned}H\rho = H\sum_{i\in I} p_i |\psi_i\rangle\langle\psi_i| = \sum_{i\in I} p_i H|\psi_i\rangle\langle\psi_i| = \sum_{i\in I} p_i\left(i\hbar\frac{\partial|\psi_i\rangle}{\partial t}\right)\langle\psi_i| \\ = i\hbar\sum_{i\in I} p_i \frac{\partial|\psi_i\rangle}{\partial t}\langle\psi_i|,\end{aligned} \tag{4.3.11}$$

and for the term ρH we can use the fact that H is Hermitian to obtain

$$\begin{aligned}\rho H = \sum_{i\in I} p_i |\psi_i\rangle\langle\psi_i| H = \sum_{i\in I} p_i |\psi_i\rangle (H|\psi_i\rangle)^\dagger = \sum_{i\in I} p_i |\psi_i\rangle\left(i\hbar\frac{\partial|\psi_i\rangle}{\partial t}\right)^\dagger \\ = -i\hbar\sum_{i\in I} p_i |\psi_i\rangle\frac{\partial\langle\psi_i|}{\partial t}.\end{aligned} \tag{4.3.12}$$

In the last step we use the total derivative to to obtain the required expressions

$$\begin{aligned}[H,\rho] = i\hbar\sum_{i\in I} p_i\left(\frac{\partial|\psi_i\rangle}{\partial t}\langle\psi_i| + |\psi_i\rangle\frac{\partial\langle\psi_i|}{\partial t}\right) = i\hbar\sum_{i\in I} p_i\frac{\partial(|\psi_i\rangle\langle\psi_i|)}{\partial t} \\ = i\hbar\frac{\partial}{\partial t}\sum_{i\in I} p_i|\psi_i\rangle\langle\psi_i| = i\hbar\frac{\partial\rho}{\partial t}.\end{aligned} \tag{4.3.13}$$

□

Remark 4.3.1 *Since time evolution is just a special case of the more general evolution of a state, we could also apply the unitary operation $U = \exp(-itH/\hbar)$ to get the solution*

$$\rho \mapsto \exp(-itH/\hbar)\rho\exp(itH/\hbar). \tag{4.3.14}$$

Postulate III can also be translated in density matrix formalism by applying the measurement operators to the ket-states individually. Suppose we have a measurement defined by a set $\{M_a : a\in\Sigma\}$ as in postulate III, then we can extend the measurement to density matrices by

$$p(a) = \sum_{i\in I} p_i \Pr(a|i) = \sum_{i\in I} p_i \operatorname{Tr}(M_a^\dagger M_a |\psi_i\rangle\langle\psi_i|) = \operatorname{Tr}(M_a^\dagger M_a\rho). \tag{4.3.15}$$

The state after measuring $a \in \Sigma$ is

$$\rho_a = \frac{1}{\mathrm{Tr}(M_a^\dagger M_a \rho)} M_a \rho M_a^\dagger \in D(\mathcal{X}). \tag{4.3.16}$$

If we perform a measurement on a system and do not keep track of the results of the measurement, we can incorporate the probability of the measurement result into the description of the state of the system after measuring. That is

$$\rho' = \sum_{a \in \Sigma} p(a) \rho_a = \sum_{a \in \Sigma} M_a \rho M_a^\dagger. \tag{4.3.17}$$

If we regard measuring in this sense as the operation $\rho \mapsto \sum_{a \in \Sigma} M_a \rho M_a^\dagger$, then measuring is in general not linear, but is does map density matrices to density matrices, as we will see later, this is an example of a *quantum channel*.

For a measurement that is described by a set of positive semidefinite operators $\{\Pi_a \succeq 0 : a \in \Sigma\}$ for which $\sum_{a \in \Sigma} \Pi_a = I_{\mathcal{X}}$, the probability of measuring outcome $a \in \Sigma$ is

$$p(a) = \mathrm{Tr}(\Pi_a \rho). \tag{4.3.18}$$

Finally, if we have multiple quantum systems that are described by Euclidean spaces $\mathcal{X}_1, \ldots, \mathcal{X}_n$, then in terms of density matrices the whole system can be described by elements from the tensor space

$$D(\mathcal{X}_1) \otimes \cdots \otimes D(\mathcal{X}_n) = D(\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n). \tag{4.3.19}$$

Simple tensors in this set are then described by

$$\rho_1 \otimes \cdots \otimes \rho_n, \tag{4.3.20}$$

for $\rho_1 \in D(\mathcal{X}_1), \ldots, \rho_n \in D(\mathcal{X}_n)$ and every tensor is a convex combination of these simple tensors.

If $\mathcal{X}_1 = \cdots = \mathcal{X}_n = \mathbf{C}^2$, then a density matrix that describes all qubits in the space $\mathcal{X}_1 \otimes \cdots \otimes \mathcal{X}_n$ is given by a $2^n \times 2^n$-matrix. As n grows the size becomes very big really quickly. For quantum simulations, this is one of the major bottlenecks.

Besides the ability to describe ensembles of quantum states with density matrices, there is another major motivation to regard quantum systems in this formalism: the ability to ignore systems. Suppose we describe a system on a Euclidean space $\mathcal{X} \otimes \mathcal{Y}$ by a density matrix $\rho \in D(\mathcal{X} \otimes \mathcal{Y})$. If we want to describe the system if we only look at the space $\mathcal{X}$, then mathematically the state is described by $\mathrm{Tr}_{\mathcal{Y}}(\rho) \in D(\mathcal{X})$.

The other way around can be incredibly useful. Regarding a given system in a sufficiently bigger space (by taking the tensor product) will allow us to describe every density operator as a pure state and every *quantum channel* as a unitary operation. The latter method is often referred to by the following quote.

> "Going to the church of larger Hilbert space."
>
> - John A. Smolin

This means that is we allow a more flexible view by considering higher dimensional spaces, we end up with simpler and more degenerate descriptions of the same objects but from a different perspective.

The first description we will encounter is purification of density matrices. Let $\mathcal{X}$ be a complex

Euclidean space and $\rho \in D(\mathcal{X})$ be a density matrix. A purification is a state $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ for some complex Euclidean space $\mathcal{Y}$ such that the full description of $|\psi\rangle$ on $\mathcal{X}$ alone is exactly the density matrix ρ, i.e., $\rho = \mathrm{Tr}_{\mathcal{Y}} |\psi\rangle \langle\psi|$.

It is a first sight not obvious that that a purification always exists. The following theorem states that for a sufficiently large space $\mathcal{Y}$, a purification indeed does exist.

Theorem 4.3.2 *(Existence of purifications) Let $\mathcal{X}$ and $\mathcal{Y}$ be two complex Euclidean spaces and ρ a density operator on $\mathcal{X}$. Then there exists a purification $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ of ρ if and only if* $\dim(\mathcal{Y}) \geq \mathrm{rank}(\rho)$.

Proof: " $\Longrightarrow$ " Suppose a purification exists and let $X \in L(\mathcal{Y}, \mathcal{X})$ such that $|\psi\rangle = \mathrm{vec}(X)$, then by Equation 2.1.30

$$\rho = \mathrm{Tr}_{\mathcal{Y}}(\mathrm{vec}(X)\,\mathrm{vec}(X)^*) = XX^*. \tag{4.3.21}$$

So $\mathrm{rank}(\rho) = \mathrm{rank}(X)$ and thus $\dim \mathcal{Y} \geq \mathrm{rank}(\rho)$.

" $\Longleftarrow$ " Suppose $\dim(\mathcal{Y}) \geq \mathrm{rank}(\rho) =: r$, a spectral decomposition of ρ is

$$\rho = \sum_{i=1}^{r} p_i |\varphi_i\rangle \langle\varphi_i| . \tag{4.3.22}$$

Let $|\xi_1\rangle, \ldots, |\xi_r\rangle \in \mathcal{Y}$ be an orthonormal set of vectors. These exist because r does not exceed the dimension of $\mathcal{Y}$. By letting

$$X = \sum_{i=1}^{r} \sqrt{p_i} |\varphi_i\rangle \langle\xi_i| , \tag{4.3.23}$$

we have $\rho = XX^*$. If we now let $|\psi\rangle = \mathrm{vec}(X)$ we get the required relation. □

Note that $\dim(\mathcal{X}) \geq \mathrm{rank}(\rho)$ for all operators, so in particular density operators. This means that if we let $\mathcal{Y} = \mathcal{X}$ we are always certain to find a purification of a density operator.

This theorem does not state anything about uniqueness and uniqueness does not apply in general. However, there is a connection between different purifications of the same density matrix on the same space $\mathcal{X}$.

The following theorem, called the *purification theorem*, informally states that two purifications in the same space $\mathcal{X} \otimes \mathcal{Y}$ of a given density operator on $\mathcal{X}$ agree on $\mathcal{X}$ by taking the partial trace and can be transformed into each other using a unitary operation on the space $\mathcal{Y}$. It is exactly this property of purifications that makes quantum bit commitment impossible, as we will see later.

Theorem 4.3.3 (Purification theorem) *Let $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{X} \otimes \mathcal{Y}$ be purifications of a density operator $\rho \in D(\mathcal{X})$, i.e.,*

$$\rho = \mathrm{Tr}_{\mathcal{Y}}(|\psi_1\rangle \langle\psi_1|) = \mathrm{Tr}_{\mathcal{Y}}(|\psi_2\rangle \langle\psi_2|), \tag{4.3.24}$$

then there exists a unitary operation U acting on $\mathcal{Y}$ such that $|\psi_2\rangle = (I_{\mathcal{X}} \otimes U) |\psi_1\rangle$.

Proof: Suppose we have a pair of purifications $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{X} \otimes \mathcal{Y}$ of a density matrix $\rho \in D(\mathcal{X})$. Let $A, B \in L(\mathcal{Y}, \mathcal{X})$ such that $\mathrm{vec}(A) = |\psi_1\rangle$ and $\mathrm{vec}(B) = |\psi_2\rangle$, we thus have the identities $\rho = AA^\dagger = BB^\dagger$ by Equation 2.1.30. This means that $r := \mathrm{rank}(\rho) = \mathrm{rank}(A) = \mathrm{rank}(B)$. Let $|\varphi_1\rangle, \ldots, |\varphi_r\rangle \in \mathcal{X}$ be a sequence of orthonormal eigenvectors of ρ corresponding to the eigenvalues $p_1, \ldots, p_r$, i.e., the spectral decomposition of ρ is

$$\rho = \sum_{i=1}^{r} p_i |\varphi_i\rangle \langle\varphi_i| . \tag{4.3.25}$$

We also have singular value decompositions of A and B given by

$$A = \sum_{i=1}^{r} \sqrt{p_i} \left|\varphi_i\right\rangle \left\langle\alpha_i\right|, \quad \text{and } B = \sum_{i=1}^{r} \sqrt{p_i} \left|\varphi_i\right\rangle \left\langle\beta_i\right|. \tag{4.3.26}$$

for vectors $\left|\alpha_1\right\rangle, \ldots, \left|\alpha_r\right\rangle, \left|\beta_1\right\rangle, \ldots, \left|\beta_r\right\rangle \in \mathcal{Y}$. Now we create a unitary matrix V such that V maps the vectors $\left|\beta_i\right\rangle$ to $\left|\alpha_i\right\rangle$ for all $i \in [r]$. It follows that

$$AV = \sum_{i=1}^{r} \sqrt{p_i} \left|\varphi_i\right\rangle \left\langle\alpha_i\right| V = \sum_{i=1}^{r} \sqrt{p_i} \left|\varphi_i\right\rangle \left(V^{-1} \left|\alpha_i\right\rangle\right)^\dagger = \sum_{i=1}^{r} \sqrt{p_i} \left|\varphi_i\right\rangle \left\langle\beta_i\right| = B. \tag{4.3.27}$$

Finally, we let $U = V^\top$ and calculate

$$\begin{aligned} (I_{\mathcal{X}} \otimes U) \left|\psi_1\right\rangle &= (I_{\mathcal{X}} \otimes U) \operatorname{vec}(A) = \operatorname{vec}(I_{\mathcal{X}} A U^\top) \\ &= \operatorname{vec}(AV) = \operatorname{vec}(B) = \left|\psi_2\right\rangle. \end{aligned} \tag{4.3.28}$$

Hence, $\left|\psi\right\rangle$ and $\left|\varphi\right\rangle$ are related by a unitary transformation on $\mathcal{Y}$. □

We have seen that density matrices and quantum states coincide in a larger dimensional space complex Euclidean space. A similar observation holds for unitary evolution and measurements and any combination of them. Both can be regarded as unitary operations on a large space. We can regard these interactions with the system as a *quantum channel*. A quantum channel is a map $\Phi \in T(\mathcal{X}, \mathcal{Y})$ of which we require that it maps density matrices to density matrices. In particular:

1. If $\operatorname{Tr}(\rho) = 1$, for some $\rho \in D(\mathcal{X})$, then $\operatorname{Tr}(\Phi(\rho)) = 1$. This property is called *trace preserving*.
2. If $\rho \succeq 0$, then $\Phi(\rho) \succeq 0$. In this case the map Φ is called *positive*.

However, properties 1 and 2 alone are not enough. A problem might occur when we add another system with space $\mathcal{Z}$. We can extend the of the channel to a channel in $T(\mathcal{X} \otimes \mathcal{Z}, \mathcal{Y} \otimes \mathcal{Z})$ in a canonical way by considering $\Phi \otimes I_{L(\mathcal{Z})}$. However, if Φ is positive, then $\Phi \otimes I_{L(\mathcal{Z})}$ is *not* necessarily always positive. We need to be more strict, since the result of applying this channel in $\mathcal{Y} \otimes \mathcal{Z}$ should still be positive semidefinite. We call an operator $\Phi \in T(\mathcal{X}, \mathcal{Y})$ *completely positive* if for every Euclidean space $\mathcal{Z}$ the operator $\Phi \otimes I_{L(\mathcal{Z})} \in T(\mathcal{X} \otimes \mathcal{Z}, \mathcal{Y} \otimes \mathcal{Z})$ is positive.

A quantum channel is now defined as an operator that is both trace-preserving and completely positive.

Quantum channels in $T(\mathcal{X}, \mathcal{Y})$ may be quite abstract to describe. By changing the space again, quantum channels get a description in terms of linear maps that we are familiar with. Suppose we associate $\mathcal{X}$ with $\mathbf{C}^n$. The Choi-Jamiołkowski isomorphism is given by

$$\begin{aligned} &J \colon T(\mathcal{X}, \mathcal{Y}) \xrightarrow{\sim} L(\mathcal{Y} \otimes \mathcal{X}) \\ &\qquad \Phi \mapsto (\Phi \otimes I_{L(\mathcal{X})})(\operatorname{vec}(I_{\mathcal{X}}) \operatorname{vec}(I_{\mathcal{X}})^\dagger) = \sum_{i=1}^{n} \sum_{j=1}^{n} \Phi(E_{ij}) \otimes E_{ij}. \end{aligned} \tag{4.3.29}$$

The inverse of this map is given by

$$F \mapsto (X \mapsto \operatorname{Tr}_{\mathcal{X}}(F(I_{\mathcal{Y}} \otimes X^\top)). \tag{4.3.30}$$

This isomorphism has two important properties

1. The map Φ is trace-preserving if and only if $\operatorname{Tr}_{\mathcal{Y}}(J(\Phi)) = I_{\mathcal{X}}$.

2. The map $\Phi \in T(\mathcal{X}, \mathcal{Y})$ is completely positive if and only if $J(\Phi)$ is a positive semidefinite operator.

This means we have an association between quantum channels from spaces $\mathcal{X}$ to $\mathcal{Y}$ and operators in $L(\mathcal{Y} \otimes \mathcal{X})$ that are positive semidefinite and the partial trace over $\mathcal{Y}$ is the identity on $\mathcal{X}$.

Finally, we discuss that every quantum channel can be described unitary operations in larger spaces. This perspective is given by Stinesprings representation theorem. By adding a sufficiently large space $\mathcal{Z}$, which is just theoretical and we can not access, all quantum channels become unitary operations. After applying this operations on the larger space, we ignore this extra space and only take the actual space the channels maps to into account.

Theorem 4.3.4 (Stinespring's representation of quantum channels, [36]) *Let $\mathcal{X}$ and $\mathcal{Y}$ be Euclidean spaces. Then $\Phi \in T(\mathcal{X}, \mathcal{Y})$ is a quantum channel if and only if there exists an isometry U from $\mathcal{X}$ to $\mathcal{Y} \otimes \mathcal{Z}$ for some Euclidean space $\mathcal{Z}$ such that*

$$\Phi(\rho) = \operatorname{Tr}_{\mathcal{Z}}(U \rho U^{\dagger}), \tag{4.3.31}$$

for every $\rho \in D(\mathcal{X})$.

This theorem describes an arbitrary quantum channel in terms of isometries. If $\dim \mathcal{X} \neq \dim(\mathcal{Y} \otimes \mathcal{Z})$ then we can always extend both spaces $\mathcal{X}$ and $\mathcal{Y} \otimes \mathcal{Z}$ to spaces of dimension $\operatorname{lcm}(\dim \mathcal{X}, \dim(\mathcal{Y} \otimes \mathcal{Z}))$ and canonically to extend the channel to obtain unitary representations.

4.4 Application of Semidefinite Programming: Optimal Measurements

This application can be found in the book *The Theory of Quantum Information* by John Watrous [26] and the article by Eldar, Megretski and Verghese [37].

Suppose we have a two player system connected by a quantum communication channel. Alice sends states from a finite set according to a distribution to Bob. Without any other means of communication Bob, has to measure these states and determine in which state Alice had sent a qubit. Bob knows which ensemble Alice uses, but not which particular state she will send. Bob's task is to find measurements that result in the highest probability of detecting a state correctly.

More formally, suppose Alice sends repeatedly one of the states $\rho_1, \ldots, \rho_m \in D(\mathcal{X})$ with probability $p_1, \ldots, p_m$ respectively, for some integer $m \geq 1$. This information is public and in particular Bob knows the states and the distribution.

Bob has to determine a collection of measurement operators $\Pi_1, \ldots, \Pi_m \succeq 0$ on $\mathcal{X}$ associated with the states $\rho_1, \ldots, \rho_m$ with the property of being positive semidefinite and

$$\sum_{i=1}^{m} \Pi_i = I_{\mathcal{X}}, \tag{4.4.1}$$

such that the probability of finding the correct state is as high as possible. This probability is given by

$$\sum_{i=1}^{m} p_i \operatorname{Tr}(\Pi_i \rho_i). \tag{4.4.2}$$

Therefore, we can state this problem as the following optimization program:

$$\sup \left\{ \sum_{i=1}^{m} p_i \operatorname{Tr}(\Pi_i \rho_i) : \sum_{i=1}^{m} \Pi_i = I_{\mathcal{X}}, \Pi_1, \ldots, \Pi_m \succeq 0 \right\}. \tag{4.4.3}$$

We will show that this optimization problem can be phrased as a semidefinite program. Let $\mathcal{Y} = \mathbf{C}^m$ and define

$$C = \sum_{i=1}^{m} E_{ii} \otimes p_i \rho_i = p_1 \rho_1 \oplus \cdots \oplus p_m \rho_m \in \mathrm{Herm}(\mathcal{Y} \otimes \mathcal{X}). \tag{4.4.4}$$

The matrix C therefore contains all information about the system. The decision variables $\Pi_1, \ldots, \Pi_m$ can be summarized in one variable

$$X = \sum_{i=1}^{m} E_{ii} \otimes \Pi_i = \Pi_1 \oplus \cdots \oplus \Pi_m \in \mathrm{Herm}(\mathcal{Y} \otimes \mathcal{X}). \tag{4.4.5}$$

The partial trace of X over $\mathcal{Y}$ is equal to the sum $\sum_{i=1}^{m} \Pi_i$. Thus, the primal optimization problem can be written as

$$\begin{aligned} &\sup \langle C, X \rangle \\ &\text{s.t. } \mathrm{Tr}_{\mathcal{Y}}(X) = I_{\mathcal{X}}, \\ &\quad X \succeq 0. \end{aligned} \tag{4.4.6}$$

The dual is given by

$$\begin{aligned} &\inf \ \mathrm{Tr}(Y) \\ &\text{s.t.} I_{\mathcal{Y}} \otimes Y \succeq C, \\ &\quad Y \in \mathrm{Herm}(\mathcal{X}). \end{aligned} \tag{4.4.7}$$

As an example we take the BB84 quantum key distribution protocol [8]. We have the four states

$$\begin{aligned} \rho_1 = |0\rangle\langle 0| = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \rho_2 |1\rangle\langle 1| = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \\ \rho_3 = |+\rangle\langle +| = \frac{1}{2}\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \quad \rho_4 = |-\rangle\langle -| = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \end{aligned} \tag{4.4.8}$$

where

$$|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle). \tag{4.4.9}$$

Furthermore, we have uniform probability of sending any of these states, i.e., $p_1 = p_2 = p_3 = p_4 = 1/4$. Then the matrix C is given by the 8×8-matrix

$$C = \begin{pmatrix} 1/4 & 0 & & & & & & \\ 0 & 0 & & & & & \varnothing & \\ & & 0 & 0 & & & & \\ & & 0 & 1/4 & & & & \\ & & & & 1/8 & 1/8 & & \\ & & & & 1/8 & 1/8 & & \\ & \varnothing & & & & & 1/8 & -1/8 \\ & & & & & & -1/8 & 1/8 \end{pmatrix}. \tag{4.4.10}$$

When solving the problem in SDPT3 4.0 [38], we find that the optimal solution is given by $\Pi_1 = \frac{1}{2}\rho_1, \ldots, \Pi_4 = \frac{1}{2}\rho_4$ with optimal value $1/2$. Thus measuring in these operators yields a probability of 50% of correct detection. A dual optimal solution can easily be found by letting $Y = \frac{1}{4} I_{\mathcal{X}} \in \mathrm{Herm}(\mathcal{X})$. This yields the objective value of $1/2$ and all constraint satisfy

$$\frac{1}{4} I_{\mathcal{X}} \succeq p_i \rho_i, \quad \text{for all } i \in \{1, \ldots, 4\}. \tag{4.4.11}$$

This confirms the optimality of both solution by weak duality.

The BB84 problem is easily extended to the situation of the six-state protocol [39]. In this case Alice sends one of the states $\rho_1, \ldots, \rho_6$ where

$$\rho_5 = \frac{1}{2}\begin{pmatrix} 1 & -i \\ i & 1 \end{pmatrix}, \quad \rho_6 = \frac{1}{2}\begin{pmatrix} 1 & i \\ -i & 1 \end{pmatrix}, \tag{4.4.12}$$

with probability $1/6$. This results in the optimal measurements $\Pi_1 = \frac{1}{3}\rho_1, \ldots, \Pi_6 = \frac{1}{3}\rho_6$ with corresponding probability 1/3 of correct detection. Similarly in this situation, a dual optimal solution is given by $Y = 1/6 I_{\mathcal{X}} \in \mathrm{Herm}(\mathcal{X})$, with objective value $1/3$ and all constraints

$$\frac{1}{6} I_{\mathcal{X}} \succeq p_i \rho_i, \quad \text{for all } i \in \{1, \ldots, 6\}, \tag{4.4.13}$$

are satisfied.

4.5 Quantum Bit Commitment

In this section we discuss quantum bit commitment. This is both an application of quantum information theory and a step towards quantum coin flipping. Similarly to quantum coin flipping, in quantum bit commitment, two parties do not trust each other. We will model the way parties can cheat using semidefinite programming and give bounds on the probability of cheating effectively using duality theory.

Suppose Alice wants to commit information to Bob. She will send Bob encrypted information, but only wants to reveal the actual information when she chooses to do so. Bob agrees on this condition, but does not want Alice to change the bit in the meantime. We want the protocol to have two important properties:

1. The protocol has to be *binding*, meaning that once Alice committed the bit to Bob, she can not change the bit anymore.

2. Furthermore, the protocol has to be *hiding*, that is, it has to be impossible for Bob to know the bit until Alice allows him to know.

If we make no complexity assumptions, then bit commitment using classical bits of information is impossible [40]. Since quantum information theory offers a more general way of interacting with information, we may hope that *quantum bit commitment* is possible. Nevertheless it is not. A simple argument is given by using the purification theorem 4.3.3.

Suppose there exists a quantum protocol for bit commitment. Let $\mathcal{A}, \mathcal{B}$ be two complex Euclidean spaces. If Alice's bit is in $\{0, 1\}$ she prepares the system in the state $|\psi_0\rangle \in \mathcal{A} \otimes \mathcal{B}$ or $|\psi_1\rangle \in \mathcal{A} \otimes \mathcal{B}$ respectively. She now sends the information in the space $\mathcal{B}$ to Bob. If she wants the commitment to be hiding, she wants the system that Bob can interact with to be the same in both cases, i.e.,

$$\mathrm{Tr}_{\mathcal{A}} |\psi_0\rangle \langle\psi_0| = \mathrm{Tr}_{\mathcal{A}} |\psi_1\rangle \langle\psi_1| . \tag{4.5.1}$$

However, by the purification theorem 4.3.3, this means there exists a unitary operation $U \in L(\mathcal{A})$ such that

$$(U \otimes I_{\mathcal{B}}) |\psi_0\rangle = |\psi_1\rangle , \tag{4.5.2}$$

thus Alice can change the bit by only interacting with her system $\mathcal{A}$, the bit is therefore not binding.

A quantum bit commitment protocol in which it is impossible to cheat does not exist. We will now determine how big the probability is of cheating successfully if either of the parties cheat. To do so, we first look at the structure of a general quantum bit commitment scheme:

1. Alice has a bit $b \in \{0,1\}$ and produces the corresponding state $|\psi_b\rangle \in \mathcal{A} \otimes \mathcal{B}$. She sends the part of state in the space $\mathcal{B}$ to Bob. This is the *commitment phase*.
2. Alice sends the rest of the state $|\psi_b\rangle$ in the space $\mathcal{A}$ to Bob together with the bit b. This is the *reveal phase*.

   Bob can now check the correctness of the state by measuring the complete state he received with the operators $\{|\psi_b\rangle\langle\psi_b|, I_{\mathcal{A}\otimes\mathcal{B}} - |\psi_b\rangle\langle\psi_b|\}$.

We will now determine optimization programs for both parties that maximize their probability of cheating successfully.

If Bob wants to cheat, he wants to know the bit b after the commitment phase but before the reveal phase, he want to find a pair of operators Π_0, Π_1 on $\mathcal{B}$ that maximizes the probability of correct detection. This reduces the problem to the optimal measurement problem of Section 4.4. With equal probability, Bob possesses the states $\mathrm{Tr}_{\mathcal{A}}(|\psi_0\rangle\langle\psi_0|)$ and $\mathrm{Tr}_{\mathcal{A}}(|\psi_1\rangle\langle\psi_1|)$, the probability of correct detection is therefore

$$\frac{1}{2}\mathrm{Tr}(\Pi_0 \mathrm{Tr}_{\mathcal{A}}(|\psi_0\rangle\langle\psi_0|)) + \frac{1}{2}\mathrm{Tr}(\Pi_1 \mathrm{Tr}_{\mathcal{A}}(|\psi_1\rangle\langle\psi_1|)). \tag{4.5.3}$$

Of course the $\{\Pi_0, \Pi_1\}$ has to be a measurement so we need the constraints $\Pi_0 + \Pi_1 = I_{\mathcal{B}}$ and $\Pi_0, \Pi_1 \succeq 0$. The semidefinite program optimizing the probability of correct detection is

$$P_A^* = \sup\left\{\frac{1}{2}\langle\Pi_0, \mathrm{Tr}_{\mathcal{A}}(|\psi_0\rangle\langle\psi_0|)\rangle + \frac{1}{2}\langle\Pi_1, \mathrm{Tr}_{\mathcal{A}}(|\psi_1\rangle\langle\psi_1|)\rangle : \Pi_0 + \Pi_1 = I_{\mathcal{B}}, \Pi_0, \Pi_1 \succeq 0\right\} \tag{4.5.4}$$

and the dual is

$$D_A^* = \inf\left\{\mathrm{Tr}(X) : X \succeq \frac{1}{2}\mathrm{Tr}_{\mathcal{A}}(|\psi_0\rangle\langle\psi_0|), X \succeq \frac{1}{2}\mathrm{Tr}_{\mathcal{A}}(|\psi_1\rangle\langle\psi_1|), X \in \mathrm{Herm}(\mathcal{B})\right\}. \tag{4.5.5}$$

On the other hand, if Alice wants to cheat, she has to take several steps into account. She has to send a state in the commitment phase but wants to change the state later to a chosen bit $b \in \{0,1\}$ such that she passes Bob's correctness test.

We denote the state she prepares in the first part and sends to Bob by $\rho \in D(\mathcal{B})$. After the commitment phase she decides on a bit $b \in \{0,1\}$ and prepares one of the corresponding states $\rho_0, \rho_1 \in D(\mathcal{A}\otimes\mathcal{B})$. Of course she can not alter the state that Bob owns anymore, so we have the pair of constraints $\mathrm{Tr}_{\mathcal{A}}(\rho_0) = \mathrm{Tr}_{\mathcal{A}}(\rho_1) = \rho$. The probability that Bob agrees with the measurement is

$$\frac{1}{2}\langle|\psi_0\rangle\langle\psi_0|, \rho_0\rangle + \frac{1}{2}\langle|\psi_1\rangle\langle\psi_1|, \rho_1\rangle. \tag{4.5.6}$$

The cheating SDP for Alice is thus given by

$$\begin{aligned} P_B^* = \sup\Big\{\frac{1}{2}\langle|\psi_0\rangle\langle\psi_0|, \rho_0\rangle + \frac{1}{2}\langle|\psi_1\rangle\langle\psi_1|, \rho_1\rangle \; &: \mathrm{Tr}_{\mathcal{A}}(\rho_0) = \rho, \mathrm{Tr}_{\mathcal{A}}(\rho_1) = \rho, \\ &\mathrm{Tr}(\rho) = 1, \rho, \rho_0, \rho_1 \succeq 0\Big\} \end{aligned} \tag{4.5.7}$$

and the dual

$$\begin{aligned} D_B^* = \inf\Big\{t \; : \; &tI_{\mathcal{B}} \succeq Z_0 + Z_1, I_{\mathcal{A}} \otimes Z_0 \succeq \frac{1}{2}|\psi_0\rangle\langle\psi_0|, \\ &I_{\mathcal{A}} \otimes Z_1 \succeq \frac{1}{2}|\psi_1\rangle\langle\psi_1|, Z_0, Z_1 \in \mathrm{Herm}(\mathcal{B}), t \in \mathbf{R}\Big\}. \end{aligned} \tag{4.5.8}$$

Strong duality applies in both cases. Let X be dual optimal for Program 4.5.5 and (t, Z_0, Z_1) be dual optimal for Program 4.5.8. Then by taking the product and using all dual constraints we find

$$\begin{aligned} P_B^* P_A^* &= D_B^* D_A^* = t \operatorname{Tr}(X) = \langle tI_{\mathcal{B}}, X \rangle \\ &\geq \langle Z_0 + Z_1, X \rangle = \langle Z_0, X \rangle + \langle Z_1, X \rangle \\ &\geq \frac{1}{2}\langle Z_0, \operatorname{Tr}_{\mathcal{A}}(|\psi_0\rangle\langle\psi_0|)\rangle + \frac{1}{2}\langle Z_1, \operatorname{Tr}_{\mathcal{A}}(|\psi_1\rangle\langle\psi_1|)\rangle \\ &= \frac{1}{2}\langle I_{\mathcal{A}} \otimes Z_0, |\psi_0\rangle\langle\psi_0|\rangle + \frac{1}{2}\langle I_{\mathcal{A}} \otimes Z_1, |\psi_1\rangle\langle\psi_1|\rangle \\ &\geq \frac{1}{4}\langle |\psi_0\rangle\langle\psi_0|, |\psi_0\rangle\langle\psi_0|\rangle + \frac{1}{4}\langle |\psi_1\rangle\langle\psi_1|, |\psi_1\rangle\langle\psi_1|\rangle = \frac{1}{2}. \end{aligned} \tag{4.5.9}$$

From $P_B^* P_A^* \geq 1/2$ we conclude that $\max\{P_A, P_B\} \geq 1/\sqrt{2}$. This strengthens the result that quantum bit commitment is impossible by showing that any protocol allows for one of the players to cheat with at least 70% success.

We will revisit the application of quantum bit commitment in Chapter 5 and discuss its application to quantum coin flipping. Furthermore, a slight variation on the lower bound on the cheating probability shown in this section also applies in the more general setting of quantum coin flipping.

4.6 More Applications of Semidefinite Programming in Quantum Information Theory

In this section we discuss a two other applications of the combination of quantum information theory and semidefinite programming. Other references of these applications and more can be found in Lecture notes by John Watrous [41].

Some of these applications are calculating the maximum output fidelity of a quantum channel, Tsirelson's inequality as a generalizations of Bell's inequality [42], hedging bets with correlated quantum strategies [43], quantum XOR games [44], determining the quantum min- and max-entropy, quantum query complexity [45] and a quantum graph variant of the Lovász ϑ-function [46, 47].

4.6.1 Calculating the Fidelity of Two Density Operators

The fidelity of a pair of quantum states is a measure of the amount in which states are the same. The fidelity is expressed as a number in the interval $[0, 1]$, with the value to 1 if and only two states are the same.

Let ρ, σ be two density operators on $\mathcal{X}$. The Fidelity between ρ and σ is

$$F(\rho, \sigma) = \operatorname{Tr}(\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}). \tag{4.6.1}$$

In this definition we use the functional calculus for a positive semidefinite operator $X = \sum_{i \in I} \lambda_i x_i x_i^\dagger \in \operatorname{Herm}(\mathcal{X})$ defined by

$$\sqrt{X} = \sum_{i \in I} \sqrt{\lambda_i} x_i x_i^\dagger. \tag{4.6.2}$$

Definition 4.6.1 is practical to use in numerical situations, but might be unpractical in theoretical applications, such as in proving bounds. There are other equivalent characterizations to this definition that might be more useful in these situations. One of them is Uhlmann's theorem, which described the fidelity in terms of the maximum inner product of purifications of both density operators.

Theorem 4.6.1 (Uhlmann's theorem) *Let $\rho, \sigma \in D(\mathcal{X})$ be density operators and $|\psi\rangle \in \mathcal{X} \otimes \mathcal{Y}$ be a purification of ρ for some complex Euclidean space $\mathcal{Y}$, i.e., $\rho = \mathrm{Tr}_{\mathcal{Y}} |\psi\rangle \langle\psi|$. Then the fidelity between ρ and σ is*

$$F(\rho, \sigma) = \sup\{|\langle\psi|\varphi\rangle| : |\varphi\rangle \in \mathcal{X} \otimes \mathcal{Y} \text{ and } |\varphi\rangle \in \mathcal{X} \otimes \mathcal{Y} \text{ is a purification of } \sigma\}. \tag{4.6.3}$$

From Uhlmann's theorem it is for example clear that the fidelity is a number in $[0, 1]$ and $F(\rho, \sigma) = F(\sigma, \rho)$ for all $\rho, \sigma \in D(\mathcal{X})$. Another characterization of the fidelity function is given by Alberti's theorem.

Theorem 4.6.2 (Alberti's theorem) *Let $\rho, \sigma \in D(\mathcal{X})$ be two density operators. The fidelity between ρ and σ is*

$$F(\rho, \sigma) = \sqrt{\inf\{\langle\rho, X\rangle\langle\sigma, X^{-1}\rangle : X \in \mathrm{Herm}(\mathcal{X}), X \succ 0\}}. \tag{4.6.4}$$

Uhlmann's theorem and Alberti's theorem are equivalent expressions for the fidelity, but it is not immediatly clear how they are connected. It is relatively easy to derive Uhlmann's theorem from Definition 4.6.1, but to prove Alberti's theorem we need some detailed topological arguments. There is a more elegant way to prove the equivalence of Uhlmann's and Alberti's theorems by using semidefinite optimization. Consider the following semidefinite optimization program

$$\sup\left\{\frac{1}{2}\mathrm{Tr}(X) + \frac{1}{2}\mathrm{Tr}(X^*) : \begin{pmatrix} \rho & X \\ X^\dagger & \sigma \end{pmatrix} \succeq 0, X \in L(\mathcal{X})\right\}, \tag{4.6.5}$$

It is relatively easy to show that this program is equivalent to Uhlmann's theorem. The dual semidefinite program is

$$\inf\left\{\frac{1}{2}\langle\rho, Y_0\rangle + \frac{1}{2}\langle\sigma, Y_1\rangle : \begin{pmatrix} Y_0 & -I \\ -I & Y_1 \end{pmatrix} \succeq 0, Y_0, Y_1 \in \mathrm{Herm}(\mathcal{X})\right\}. \tag{4.6.6}$$

By using Slater's condition 3.4.1, we can show that strong duality holds and thus the optimal value of Program 4.6.6 is $F(\rho, \sigma)$. With some relatively easy manipulations, it is possible to show that 4.6.6 is equivalent to Alberti's theorem. In this case the topological arguments that were required to give a direct proof are handled in Slater's condition.

4.6.2 Optimal Quantum Cloning

The application in this section uses semidefinite optimization to find the optimal probability of successfully counterfeiting money that is characterized by quantum information [48]. In theory, it is possible to embed qubits into banknotes and store quantum information as a way of identifying a legal banknote. If the probability of counterfeiting money is low, then the system will be considered secure. We will sketch the problem and formulate the corresponding semidefinite program.

Consider an ensemble of quantum states $|\psi_1\rangle, \ldots, |\psi_n\rangle \in \mathcal{X}$ with probability $p_1, \ldots, p_n$ respectively, for some integer $n \geq 1$. Each note contains one of these states and is also labelled by a unique number.

If an owner wants to check the validity of their note, the bank will use the unique number to determine which of the states $|\psi_i\rangle$ for some $i \in \{1, \ldots, n\}$, is embedded in the note. This can be done by using a private hash function. The bank then measures the embedded state in the set of operators $\{|\psi_i\rangle \langle\psi_i|, I_{\mathcal{X}} - |\psi_i\rangle \langle\psi_i|\}$ and declare whether the note is legal or counterfeited respectively.

If we want to make copies of these states that pass the validity test with the highest probability. The problem is that although the ensemble of notes is known, we do not know which of the states

corresponds to the unique number. Simply measuring the state arbitrarily will collapse the state.

To counterfeit an arbitrary note as good as possible, we are looking for a quantum channel $\Phi : D(\mathcal{X}) \to D(\mathcal{Y}_1 \otimes \mathcal{Y}_2)$ where $\mathcal{Y}_1$ and $\mathcal{Y}_2$ are both isomorphic to $\mathcal{X}$. This means the quantum information of one note is being transformed into quantum information of two notes in the same space. An ideal channel that copies the note does not exist in general, which we proved in the no cloning theorem (Theorem 4.1.1). Our objective is to find the best possible channel. That is, we want a quantum channel $\Phi \in T(\mathcal{X}, \mathcal{Y}_1 \otimes \mathcal{Y}_2)$, that optimizes the probability of passing the test

$$\sum_{i=1}^{n} p_i \langle \psi_i \otimes \psi_i | \Phi(|\psi_i\rangle \langle \psi_i|) |\psi_i \otimes \psi_i\rangle . \tag{4.6.7}$$

By the Choi-Jamiołkowski map, we can describe any channel $\Phi \in T(\mathcal{X}, \mathcal{Y}_1 \otimes \mathcal{Y}_2)$ uniquely by a positive semidefinite operator $J(\Phi)$ on $\mathcal{Y}_1 \otimes \mathcal{Y}_2 \otimes \mathcal{X}$ for which the partial trace over $\mathcal{Y}_1 \otimes \mathcal{Y}_2$ is the identity on $\mathcal{X}$. With this representation we can rewrite the objective as well and find the following semidefinite program:

$$\sup \left\{ \sum_{i=1}^{n} p_i \left\langle \psi_i \otimes \psi_i \otimes \overline{\psi_i} \right| J \left| \psi_i \otimes \psi_i \otimes \overline{\psi_i} \right\rangle : \mathrm{Tr}_{\mathcal{Y}_1 \otimes \mathcal{Y}_2} J = I_{\mathcal{X}}, J \succeq 0 \right\} . \tag{4.6.8}$$

To find the corresponding quantum channel we can apply the inverse of the Choi-Jamiołkowski isomorphism from Equation 4.3.30. The dual of this program is

$$\inf \left\{ \mathrm{Tr}(Y) : I_{\mathcal{Y}_1 \otimes \mathcal{Y}_2} \otimes Y \succeq \sum_{k=1}^{n} p_k \left| \psi_k \otimes \psi_k \otimes \overline{\psi_k} \right\rangle \left\langle \psi_k \otimes \psi_k \otimes \overline{\psi_k} \right| , Y \in \mathrm{Herm}(\mathcal{X}) \right\} . \tag{4.6.9}$$

Analysing this pair of semidefinite programs results in bounds and optimal counterfeiting channels that prove unconditional security. It is unlikely that Wiesner's quantum money will find a real world implementation. Decoherence will affact the quantum information really quicly and make the validity test unreliable. However, it might be that this method of embedding quantum information as a way of authenticating can be applied in a more general setting.

Chapter 5

Quantum Coin Flipping

"Humor is the ability to see three sides to one coin."

- Ned Rorem

This chapter focuses on formalizing and describing classical and quantum coin flipping. We consider different protocols and discuss new and known results. An important measure of the quality of a protocol is the bias, which is the largest possible deviation from a fair coin. The major goal of creating a quantum coin flipping protocol is to make the bias as low as possible. However, some fundamental limitations in classical and quantum coin flipping prevent arbitrary small biases from existing. We will encounter these limitations, describe explicit protocols and introduce new extensions on these protocols.

5.1 Coin Flipping using Classical Communication

We will start with a classical coin flipping protocol, that introduced the subject in the field of cryptography. This protocol is described in the paper 'Coin Flipping by Telephone: a Protocol for Solving Impossible Problems' by computer scientist Manuel Blum in 1983 [5]. This protocol can be viewed as the coin flipping variant of the well known RSA-protocol for asymmetric key distribution [49] and is based on an equivalent mathematical problem to factoring that provides complexity.

Protocol 5.1.1 (Blum's coin flipping protocol, [5])

1. *Alice picks two different prime number $p, q \equiv 3 \pmod 4$ and sends the product $N = pq$ to Bob. She keeps the factors p and q private.*
2. *Bob randomly picks a number $x \in \{0, \ldots, N-1\}$ and sends $s = x^2 \mod N$ to Alice.*
3. *Alice computes the unique four square roots of s in $\mathbf{Z}/N\mathbf{Z}$, namely $\{-x, x, -y, y\}$ for some $y \in \mathbf{Z}/N\mathbf{Z}$ where $y \neq x$. She can use modular exponentiation and the Chinese remainder theorem to do this efficiently.*
4. *Alice randomly picks one of the four roots and sends it to Bob.*
5. *The winner is determined by the following rules:*
   (a) *If Alice's choice is $\pm x$, then Alice wins, Bob has to announce to Alice that she won;*
   (b) *If Alice's choice is $\pm y$, then Bob wins, Bob announces this to Alice and proves he won by announcing the value x he chose.*

The important part is that Bob can easily calculate the number s, but it's hard to calculate all square roots and in particular find $\pm y$ without knowledge of the factors of N. However, Alice knows the factors of N and can therefore calculate the square root of s efficiently. However, she does know which of the roots Bob picked, because they are all valid.

We will show how Alice can calculate the four roots of an element $s \in \mathbf{Z}/N\mathbf{Z}$ efficiently. Alice will calculate the square roots in the fields $\mathbf{F}_p$ and $\mathbf{F}_q$ first, this can be done easily by modular exponentiation. In $\mathbf{F}_p$ the two different roots of s are given by $\pm s^{\frac{p+1}{4}} \pmod p$. To see why we square the element.

$$\left(\pm s^{\frac{p+1}{4}}\right)^2 \pmod p = s^{\frac{p+1}{2}} \pmod p = s^{\frac{p-1}{2}} \cdot s \pmod p = s \pmod p. \tag{5.1.1}$$

where we have used Euler's criterion that tells us that $s^{\frac{p-1}{2}} = 1 \pmod p$ for all squares $s \in \mathbf{F}_p$. Furthermore $\frac{p+1}{4}$ is an integer since $p \equiv 3 \pmod 4$, therefore we can apply modular exponentiation. The same arguments hold to determine the roots in the field $\mathbf{F}_q$ for the square roots $\pm s^{\frac{q+1}{4}} \pmod q$.

We can now make four pairs of square roots in $\mathbf{F}_p \times \mathbf{F}_q$. By the Chinese remainder theorem we can now find for the four square roots of s in the ring $\mathbf{Z}/N\mathbf{Z}$.

This elegant separation of the information that Alice knows and Bob does not, and vice versa is the reason this protocol works. Alice has the ability to calculate all square roots of s but does not know which of these Bob took to create s in the first place. On the other hand, Bob does know x and s but in order for him to always win, he has to announce y to Alice. If Bob knows the factorisation of N he can compute the square roots (he knows the same information as Alice), but the converse is also true: if Bob know the four square roots he can factor N. This result is shown in the following lemma.

Lemma 5.1.1 *If we know four different square roots of some $s \in \mathbf{Z}/N\mathbf{Z}$, where $N = pq$ and $p, q \equiv 3 \pmod 4$ are two different unknown prime numbers, then we can calculate p, q efficiently from the four square roots.*

Proof: Suppose we have four different square roots $-x, x, -y, y \in \mathbf{Z}/N\mathbf{Z}$ of some element $s \in \mathbf{Z}/N\mathbf{Z}$. We will show that $z = \gcd(x - y, N)$ is one of the prime factors of N. First note that $r|N$, therefore $r \in \{1, p, q, N\}$. We have to exclude two cases

1. Suppose $r = N$, then $N|(x - y)$ and thus $x = y \pmod N$, but we assumed x and y are different square roots, so we have a contradiction.
2. Suppose $r = 1$. Since $0 = x^2 - y^2 = (x - y)(x + y)$ we have $N|(x - y)(x + y)$, and since $x - y$ and N are coprime, we must have that $N|(x + y)$, but then $x = -y \pmod N$. Which is again a contradiction.

Therefore r is a prime factor of N and is obtained by just calculating the greatest common divisor, which can be done efficiently by the Euclidean algorithm. The other prime factor can of course be obtained by simply calculating N/r. □

Since factoring is in the class NP, and no efficient algorithm (yet?) exists, we deduce that taking square roots in the ring $\mathbf{Z}/N\mathbf{Z}$ is at least as hard.

Before we proceed, we give an example of how this algorithms works with explicit numbers.

Example 5.1.1 *Let $p = 19$ and $q = 31$, then $N = 589$, so we work in the ring $\mathbf{Z}/589\mathbf{Z}$. This ring is isomorphic to $\mathbf{Z}/19\mathbf{Z} \times \mathbf{Z}/31\mathbf{Z}$ by the Chinese remainder theorem. Bob now picks an arbitrary element from this ring, say $x = 201 \in \mathbf{Z}/N\mathbf{Z}$, and calculates $s = x^2 \pmod N = 349 \pmod{589}$ and sends this number to Alice. Alice will calculate all square roots of s in the fields $\mathbf{F}_{19}$ and $\mathbf{F}_{31}$:*

$$\begin{aligned} \pm s^{\frac{p+1}{4}} \pmod p &= \pm 7^5 \pmod{19} = \pm 11 \pmod{19} = 8, 11 \pmod{19}, \\ \pm s^{\frac{q+1}{4}} \pmod q &= \pm 8^8 \pmod{31} = \pm 16 \pmod{31} = 15, 16 \pmod{31}. \end{aligned} \tag{5.1.2}$$

This gives four possible square roots and Alice now calculates the corresponding element in $\mathbf{Z}/589\,\mathbf{Z}$ *by the Chinese remainder theorem*

$$\begin{aligned}(8 \pmod{19}, 15 \pmod{31}) &\mapsto 46 \pmod{589},\\ (8 \pmod{19}, 16 \pmod{31}) &\mapsto 388 \pmod{589},\\ (11 \pmod{19}, 15 \pmod{31}) &\mapsto 201 \pmod{589},\\ (11 \pmod{19}, 16 \pmod{31}) &\mapsto 543 \pmod{589}.\end{aligned} \tag{5.1.3}$$

A simple check confirms that the square of all these numbers is indeed $349 \pmod{589}$*. We can now recognize* $x = 201$*,* $-x = 388$ *and* $y = 46$*,* $-y = 543$*. From* s *itself it is impossible to know which of the four Bob picked. Alice chooses and announces her choice. If her choice is* $\pm x = 201, 388$ *then Bob has to announce that Alice won. If Alice took* $\pm y = 46, 543$*, then Bob won, and he has to prove this to Alice by announcing* x*.*

Of course in this example it is very easy to brute force the factorization of N *and determine all square roots of* s*. As the size of* p *and* q *factoring becomes practically impossible with the current methods. The lack of knowledge Alice has does not depend on the size of the prime factors.*

5.2 Complexity Assumptions and Shor's Algorithm

Complexity assumptions such as in Blum's coin flipping protocol make this protocol secure. Both parties have different pieces of information and calculating all the information requires a lot of computing power.

However, the complexity assumption of the discrete logarithm problem, which is the foundation of for example factoring, is not valid anymore in the context of quantum computing. In 1994 Peter Shor published an algorithm [6] on a quantum computer that is capable of factoring a composite number N in $O((\log N)^3)$ operations [33]. *Shor's algortihm,* as it is most commonly referred to, is based on *order finding*. In order finding, we consider two positive integers x, N that do not share a common factor and the goal is to find the smallest r such that $x^r = 1 \pmod{N}$. This problem is a specific instance of the phase estimation algorithm which provides a general algorithm for the discrete logarithm problem in abelian groups.

A natural question would be if we could perform a coin flipping protocol with classical communication that is not completely broken, without having to resort to any complexity assumption at all. The answer is no. This can be proven using game theory. The optimization we will employ in the following sections applies to quantum coin flipping. By restricting this model such that it represents the most general setting in classical coin flipping, we find a game theoretic description. This game has a number of properties: it is executed by two players, it is a zero-sum game, both players have all information about the state of the game, there are only two possible outcome and no turns that introduce chance into the game. The famous book *Theory of games and economic behavior* by John von Neumann and Oskar Morgenstern in 1944 [4], shows that such game has a strategy such that a player can win with certainty. Translating this back to coin flipping, we find that any classical coin flipping protocol is completely broken. More details on this proof and references can be found in [3].

5.3 Coin Flipping Based on Classical or Quantum Bit Commitment

One of the two main solutions to the vulnerability of security of classical coin flipping using discrete log based problems is by considering a protocol that uses classical communications, but is also secure against using quantum computers. Complexity classes in quantum computing are related to classical computing and in some cases overlap. In particular can quantum computers

simulate classical computers by just considering computational states $\{|0\rangle, |1\rangle\}$ of every qubit and operations that map computational states to computational states[1]. This means that if we have a classical protocol that is based on a mathematical problem that is hard to solve for a quantum computer (and therefore also hard for a classical computer), then the protocol is secure by complexity assumptions.

Two of these mathematical problems are finding the shortest vector of a lattice and code based schemes. These problems serve as the building blocks for cryptographic primitives. For example, there do exist post-quantum bit-commitment schemes [50]. We will use these schemes to build a post-quantum protocol for coin flipping for two parties.

Protocol 5.3.1 (Coin flipping protocol based on bit commitment, [51]) *This protocol is based on a post-quantum bit-commitment protocol that we regard as a black-box.*

1. *Alice and Bob both independently flip a fair coin $a, b \in \{0, 1\}$ respectively.*
2. *Alice commits a to Bob, and similarly Bob commits b to Alice using the black-box post-quantum bit commitment protocol.*
3. *If each party received the commitment of the other party, they can reveal their bit, so both Alice and Bob now the information of a, b. They can check whether the information they received corresponds to the commitment.*
4. *If the checks agree, the result of the coin flip is $a + b \mod 2$. Both parties announce the result. If the results are different the protocol is aborted.*

A graphical representation of the steps in Protocol 5.3.1 is shown in Figure 5.1. This show that both parties first commit and after receival reveal their information.

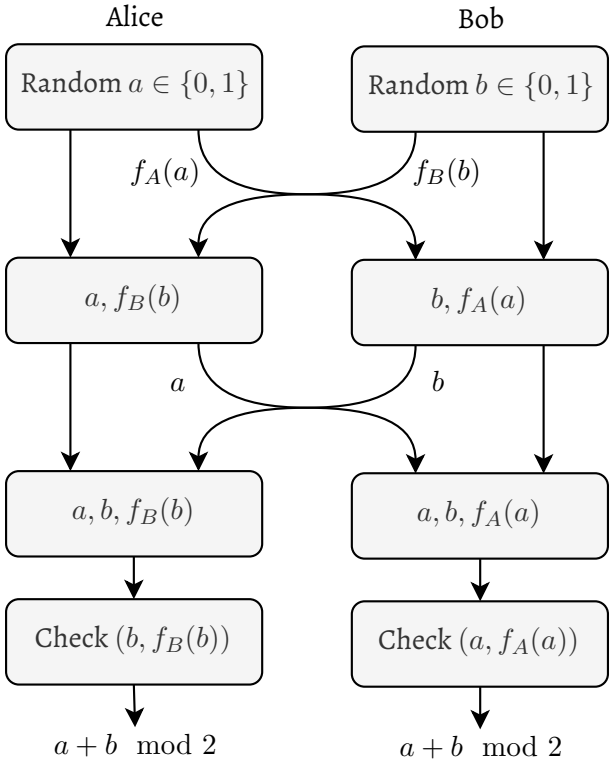


Figure 5.1: A quantum coin flipping protocol based on bit commitment. Both Alice and Bob commit their bit and reveal after receival. The XOR (addition modulo 2) is the result of the process.

[1]The no-cloning theorem (Theorem 4.1.1) does, for example, not hold if we just consider computational states. Measuring in the computational basis doe no longer change the state.

To show why this protocol is secure. We have to look for it's vulnerabilities. Suppose that Alice is honest and Bob is dishonest. Because of the symmetry in this problem the situation in which Alice is dishonest and Bob is honest is the same. One strategy for him, is to change the distribution of his coin flip. We denote this by some random variable B, which is 0 with probability p and 1 with probability $1-p$, for some $p \in [0,1]$. Alice her private coin flip A is fair has thus has equal probability for both outcomes. Since the bit-commitment is secure, the resulting coin flip is a realisation of the random variable $X = A + B \mod 2$. The range of X is again $\{0,1\}$ and thus the distribution is determined by

$$\begin{aligned} \Pr(X=0) &= \Pr((A=0 \text{ and } B=0) \text{ or } (A=1 \text{ and } B=1)) \\ &= \Pr(A=0)\Pr(B=0) + \Pr(A=1)\Pr(B=1) \\ &= \frac{1}{2}p + \frac{1}{2}(1-p) = \frac{1}{2} \end{aligned} \tag{5.3.1}$$

The commitment and revealing phase are secure and therefore Bob can't cheat in this phase.

If we want to make the protocol applicable for multiparty, we have to beware that dishonest parties may work together to perturb or even fix the outcome. As it turns out, one honest player is sufficient to be protected against cheating.

Protocol 5.3.2 (Post-quantum multiparty coin flipping protocol based on bit commitment) *This protocol uses a post-quantum bit commitment scheme as a black-box. This protocol is based on n players.*

1. *All players independently flip a fair coin $a_i \in \{0,1\}$ for $i \in \{1,\dots,n\}$.*
2. *Every players $i \in \{1,\dots,n\}$ commits a_i to all other players publicly.*
3. *If each party received the commitment of the other party, then every party reveals his of her bit. The full information $a_1,\dots,a_n$ is now known to every player and players can check whether the bit and commitment they received agree with each other.*
4. *If no players aborts the protocol, the result of the coin flip is $a_n + \cdots + a_k \mod 2$ and all parties announce the result.*

When at least one players is honest, disturbing the distribution or even when some of the coin flips are dependent by dishonest players doesn't influence the outcome. This is shown in Lemma 5.3.1 and is a direct generalization of the two party argument.

Lemma 5.3.1 *Let $A_1,\dots,A_n$ be coin flips with probability $p_1,\dots,p_n$ respectively. If there is at least one $i \in \{1,\dots,n\}$ such that $p_i = 1/2$ and independent of $A_1,\dots,A_{i-1},A_{i+1},\dots,A_n$, then $X = A_1 + \ldots + A_n \mod 2$ is a fair coin flip.*

Proof: Without loss of generality, we assume $p_1 = 1/2$ and A_1 is independent from $A_2,\dots,A_n$. Note that $B = A_2 + \ldots + A_n \mod 2$ is a coin flip with parameter p for some $p \in [0,1]$. Note that $X = A_1 \oplus B$ and thus

$$\begin{aligned} \Pr(X=0) &= \Pr((A_1=0 \text{ and } B=0) \text{ or } (A_1=1 \text{ and } B=1)) \\ &= \Pr(A_1=0)\Pr(B=0) + \Pr(A_1=1)\Pr(B=1) \\ &= \frac{1}{2}p + \frac{1}{2}(1-p) = \frac{1}{2}, \end{aligned} \tag{5.3.2}$$

so X is fair coin flip. □

In this case the post-quantum bit commitment scheme is secure and therefore a dishonest player cannot influence this step. The result is a fair coin if all parties announce the same outcome.

5.4 Quantum Coin Flipping

In the previous sections, we looked at coin flipping protocols in which players communicated through classical channels. This lead to perfect protocols if make complexity assumptions and an impossibility result if we do not make complexity assumptions. In this section we consider protocols in which we communicate and interact with information through classical channels. We will not make complexity assumptions and instead use the properties of quantum information itself to secure the protocols.

To quantify this, we will determine how much a cheater can make the protocol deviate from a fair coin. We will first formalize the bias, quantum coin flipping protocols and present how to cheat. First, let $P^*_{A,0}$ and $P^*_{A,1}$ be the maximum probability of Alice outputting a 0 resp. 1 when Bob is cheating and Alice is honest. Similarly, $P^*_{B,0}$ and $P^*_{B,1}$ is the maximum probability of Bob outputting a 0 resp. 1 when Alice is cheating and Bob is honest. Here the maximum is taken over all *cheating strategies*. Informally, a cheating strategy is a number of steps taken by a cheater in the protocol to achieve a certain predefined goal. In describing quantum coin flipping in terms of semidefinite optimization we will mathematically formalize the concept op cheating strategies as well.

A good protocol will limit the possibility of a cheater perturbing the outcome of the protocol by a lot. It is a priori not always clear whether Alice or Bob will be a cheater and furthermore, if a cheater is present, which outcome he or she will try to establish. The worst-case scenario is therefore considered when we look at all four possible variations, determined by the probabilities $P^*_{A,0}, P^*_{A,1}, P^*_{B,0}$ and $P^*_{B,1}$. This leads to the definition of the *strong bias* of a coin flipping protocol.

If we do know that players have a preference, e.g., Alice prefers the outcome 0 and Bob the outcome 1, it is useless to consider the situations in which they would cheat to find their opposite outcomes $P^*_{A,0}$ and $P^*_{B,1}$. In this case we take these preferences into account and determine the *weak bias*.

Definition 5.4.1 *The strong bias of a coin flipping protocol is defined as the number*

$$\varepsilon = \max\{P^*_{A,0}, P^*_{A,1}, P^*_{B,0}, P^*_{B,1}\} - \frac{1}{2}. \tag{5.4.1}$$

Similarly, the weak bias of a coin flipping protocol in which Alice prefers the outcome 0 and Bob the outcome 1, the bias is given by

$$\varepsilon_{WCF} = \max\{P^*_{A,1}, P^*_{B,0}\} - \frac{1}{2}. \tag{5.4.2}$$

This definition can be interpreted such that for a given protocol and cheating strategy, the resulting coin will be perturbed to a unbiased coin with probability in the interval $[1/2-\varepsilon, 1/2+\varepsilon]$. Note that any quantum coin flipping protocol has a weak bias less than or equal to the strong bias.

We will first consider and compare some quantum coin flipping protocols. The first protocol by Andris Ambainis consists of three rounds of communications and is based on qutrits: states in the space $\mathcal{X} \cong \mathbf{C}^3$.

Protocol 5.4.1 (Ambainis' coin flipping protocol, [12]) *This protocol uses two qutrits, first in the possession of Alice. The basis states are denoted by* $\{|0\rangle, |1\rangle, |2\rangle\}$.

1. *Alice picks a random element* $i \in \{0, 1\}$ *and creates the two-qutrit state*

$$|\psi_i\rangle = \frac{1}{\sqrt{2}}(|i\rangle\,|i\rangle + |2\rangle\,|2\rangle) \in \mathcal{X} \otimes \mathcal{X}. \tag{5.4.3}$$

Alice sends the first qutrit to Bob.

2. *Bob picks a random bit $j \in \{0,1\}$ and sends it to Alice by classical means.*
3. *Alice reveals i to Bob and she sends him send the second qutrit. Bob now has the whole state $|\psi_i\rangle$*
4. *Bob will now measure the state he received form Alice in the operators*

$$\{\Pi_0 = |\psi_0\rangle\langle\psi_0|, \Pi_1 = |\psi_1\rangle\langle\psi_1|, \Pi_\varnothing = I_{\mathcal{X}\otimes\mathcal{X}} - \Pi_0 - \Pi_1\} \tag{5.4.4}$$

   If the outcome of his measurement is $\varnothing$, then Bob aborts the protocol. Otherwise he can check it the bit that Alice send is correct.
5. *If Bob does not abort the protocol, then both Alice and Bob will output $i + j \mod 2$.*

It is clear that this protocol is not completely symmetric. Alice her task is to create the two-qutrit, whilst Bob does not interact with the state other then measuring it. On the other hand, Bob has the possibility to detect when Alice cheats, whilst Alice does not have this possibility. Nevertheless

Lemma 5.4.1 *The quantum coin flipping procedure presented in protocol 5.4.1 has optimal cheating probabilities $P^*_{A,0} = P^*_{A,1} = P^*_{B,0} = P^*_{B,1} = 3/4$ and consequently the bias is $\varepsilon = 1/4$.*

When we formalize the protocol into a standard form in which we can determine the optimal cheating strategy with semidefinite programming, we will also confirm the optimal values in Lemma 5.4.1.

The following quantum coin flipping protocol is inspired on BB84 the quantum key distribution protocol. Alice will send a (rotated) BB84 states and Bob will measure them in one of the two basis arbitrary basis. The outcome is determined by a coin flip of Bob and the encoded bit of Alice.

Protocol 5.4.2 (Berlín et al.'s quantum coin flipping, [15]) *Let α be a number in $[0, \pi/4]$.*

1. *Alice picks a two random bits $a, x \in \{0,1\}$ independently and prepares*

$$|\psi_{a,x}\rangle = \begin{cases} \cos\alpha|0\rangle + \sin\alpha|1\rangle & \text{for } a = 0, x = 0, \\ \cos\alpha|0\rangle - \sin\alpha|1\rangle & \text{for } a = 1, x = 0, \\ \sin\alpha|0\rangle - \cos\alpha|1\rangle & \text{for } a = 0, x = 1, \\ \sin\alpha|0\rangle + \cos\alpha|1\rangle & \text{for } a = 1, x = 1, \end{cases} \tag{5.4.5}$$

   and sends this state to Bob.
2. *Bob picks a random $b \in \{0,1\}$ and measures the qubit he received in the basis $\{|\psi_{b,0}\rangle, |\psi_{b,1}\rangle\}$. He also picks a random bit $y \in \{0,1\}$ and sends this bit to Alice.*
3. *After Alice received Bob's bit, she sends him both her basis a and bit x.*
4. *If the basis agree, i.e., $a = b$, and the state Bob measured is not $|\psi_{a,x}\rangle$, he aborts the protocol.*
5. *If Bob does not abort the protocol both parties output the coin $x + y \mod 2$.*

Similar to Ambainis' protocol 5.4.1, in Protocol 5.4.2 Alice does not have the possibility to abort whilst Bob does. Contrary to Ambainis' protocol, this protocol by Berlín et al. can be implemented using existing quantum key distribution infrastructure, which has been done in 2011 [16]. However, the cheating probabilities of the Berlín et al. protocol are worse than those of Ambainis' protocol.

Lemma 5.4.2 *The cheating probabilities of Protocol 5.4.2 are*

$$P^*_{A,0} = P^*_{A,1} = \cos^2 \alpha, \quad P^*_{B,0} = P^*_{B,1} = \frac{3 + \sin(2\alpha)}{4}. \tag{5.4.6}$$

The protocol is balanced if and only if $\alpha = \arctan(1/3) \approx 0.32175... \approx 18.43°$*, then* $\varepsilon = 9/10$*.*

A plot of the optimal cheating probabilities of Protocol 5.4.2 is shown in Figure 5.2.

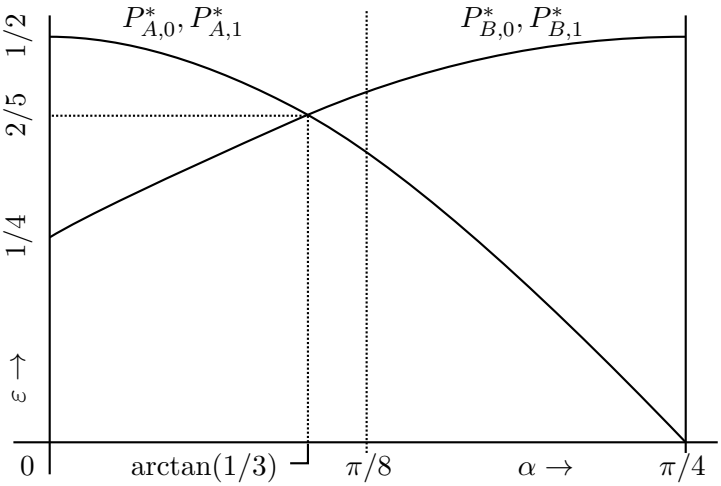


Figure 5.2: In this plot, the cheating probabilities (corrected for the bias by the term $1/2$) are shown as a function of the input parameter $\alpha \in [0, \pi/4]$. The bias is the maximum of both curves and has a minimum at $\alpha \approx 0.32175\ldots$.

The two protocols we discussed share a common structure: they manipulate quantum and classical information in an alternating way, send information back and forth and measure a final state that fixes the outcome of the protocol. The following definition describes this in a rigorous way and allows us to analyse quantum coin flipping protocols.

Definition 5.4.2 (Quantum coin flipping protocol, two parties) *A quantum coin flipping protocol with two parties is defined by the following collection of structures and rules:*

1. *A triple of complex Euclidean spaces* $\mathcal{A}, \mathcal{M}$ *and* $\mathcal{B}$*, respectively the spaces in which the quantum information of Alice, the message space and Bob exist.*
2. *A positive integer* N*, denoting the number rounds in the protocol.*
3. *The initial state of the system:*

$$|\psi_0\rangle = |0\rangle_{\mathcal{A}} \otimes |0\rangle_{\mathcal{M}} \otimes |0\rangle_{\mathcal{B}} \in \mathcal{A} \otimes \mathcal{M} \otimes \mathcal{B}. \tag{5.4.7}$$

4. *Two* N*-tuples of unitary operations:*

$$U_{A,1}, \ldots, U_{A,N} \quad \text{and} \quad U_{B,1}, \ldots, U_{B,N}, \tag{5.4.8}$$

   where $U_{A,i}$ *acts on* $\mathcal{A} \otimes \mathcal{M}$ *and* $U_{B,i}$ *acts on* $\mathcal{M} \otimes \mathcal{B}$ *for every* $i \in \{1, \ldots, N\}$*.*

5. *A measurement* $\{\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\varnothing}\}$ *on the space* $\mathcal{A}$*, representing the outcomes 0, 1 or abort respectively.*
6. *A measurement* $\{\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\varnothing}\}$ *on the space* $\mathcal{B}$*.*
7. *Bob prepares the state* $|0\rangle \in \mathcal{M}$ *and sends it to Alice before the first round. Alice and Bob apply their unitary operations in an alternating way on* $\mathcal{A} \otimes \mathcal{M}$ *and* $\mathcal{M} \otimes \mathcal{B}$*. Before both parties measure the quantum state, Bob sends the state in* $\mathcal{M}$ *to Alice.*

These structures satisfy

$$(\Pi_{A,0} \otimes I_{\mathcal{M}} \otimes \Pi_{B,1}) |\psi_N\rangle = (\Pi_{A,1} \otimes I_{\mathcal{M}} \otimes \Pi_{B,0}) |\psi_N\rangle = 0, \tag{5.4.9}$$

where $|\psi_N\rangle$ is the state of the system just before measuring

$$|\psi_N\rangle = (I_{\mathcal{A}} \otimes U_{B,N})(U_{A,N} \otimes I_{\mathcal{B}}) \cdots (U_{A,1} \otimes I_{\mathcal{B}}) |\psi_N\rangle . \tag{5.4.10}$$

The whole protocol can be described as a sequence of alternating operations done by Alice and Bob and sending messages to each other and in the end measure the result and determine the output head, tails or abort the protocol. This is shown graphically in Figure 5.3.

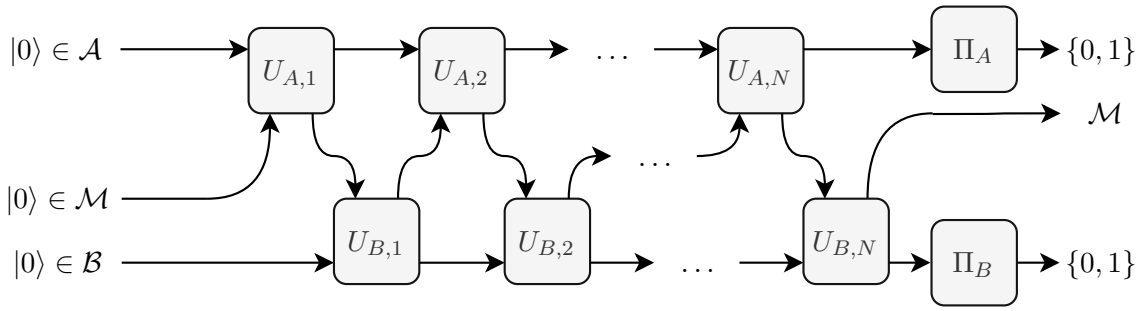


Figure 5.3: The general two player quantum coin flipping protocol consisting of N rounds. The entire process consists of applying unitary operations in an alternating way on $\mathcal{A} \otimes \mathcal{M}$ and $\mathcal{M} \otimes \mathcal{B}$ and finally measuring the state in $\mathcal{A}$ and $\mathcal{B}$.

The fact that we do have to take interactions other than unitary operations (such as measurements or the introduction of classical randomness) in Item 4 of Definition 5.4.2 into account is a direct consequence of Stinespring's representation theorem 4.3.4. In particular we can regard any possible action on the system as a quantum channel and than apply Stinespring's representation theorem to obtain a description solely by unitary operations. This approach is thoroughly described in Watrous' lecture notes on quantum coin flipping [26]. In general, this does of result in larger spaces $\mathcal{A}$, $\mathcal{M}$ and $\mathcal{B}$.

Item 7 does not affect the generality of a quantum coin flipping protocol, since adding trivial operations $I_{\mathcal{A}\otimes\mathcal{M}}$ and $I_{\mathcal{M}\otimes\mathcal{B}}$ yields the same result. However, it does lead to more symmetry in the semidefinite program for determining the optimal cheating strategy, as we will see later.

Furthermore, if we want the coin flip to be balanced, we require

$$\langle\psi_n| (\Pi_{A,0} \otimes I_{\mathcal{M}} \otimes \Pi_{B,0}) |\psi_n\rangle = \langle\psi_n| (\Pi_{A,1} \otimes I_{\mathcal{M}} \otimes \Pi_{B,1}) |\psi_n\rangle = 1/2. \tag{5.4.11}$$

Consequently, the probability of aborting the protocol if both players are honest, is 0.

Remark 5.4.1 *Definition 5.4.2 does not state whether a protocol is strong or weak since this is only relevant in determining the bias of a protocol. A protocol may however be suitable for a strong or weak situation depending on whether it yields a good quality by considering its bias.*

Alice and Bob do not share any entanglement between their spaces when they start the protocol. If they would share entanglement, say the EPR state $(|00\rangle + |11\rangle)/\sqrt{2}$, they would not have to apply any operation at all and could immediately measure in the standard basis to get a protocol with bias 0.

We are now ready to state one of the most important results in quantum coin flipping, i.e., Kitaev's lower bound based on semidefinite optimization.

Theorem 5.4.3 (A.Y. Kitaev, 2002, [52], unpublished) *Every strong quantum coin flipping protocol has bias* $\varepsilon \geq 1/\sqrt{2} - 1/2$.

Before we can prove this result, we have to determine what cheating means with respect to Definition 5.4.2. Suppose that Bob is a cheater and wants to enforce outcome 1 on Alice. He can not interact with the space $\mathcal{A}$ immediately, but he can alter the preparation of the message space and his own unitary operations $U_{B,1}, \ldots, U_{B,N}$. Suppose $\rho_{A,j} \in D(\mathcal{A} \otimes \mathcal{M})$ is the state right after Alice applied the operation $U_{A,j}$ for some $j \in \{1, \ldots, N\}$ and $\sigma_{A,j} \in D(\mathcal{A} \otimes \mathcal{M})$ be the state after Bob applied his replacement channel for the operation $U_{B,j}$. Then Bob could not have changed the state on Alice her space, so we have the constraint

$$\operatorname{Tr}_{\mathcal{M}} \rho_{A,j} = \operatorname{Tr}_{\mathcal{M}} \sigma_{A,j}. \tag{5.4.12}$$

If the state of the system before Bob's operation is pure, then the state after his operation on $\mathcal{M} \otimes \mathcal{B}$ is also pure, for sufficiently large $\mathcal{B}$. This means that Bob effectively applied an operation $I \otimes U$ on the whole system for some unitary U operation on $\mathcal{M} \otimes \mathcal{B}$. We can always pick a purification of the system that satisfies Equation 5.4.12.

From Bob's perspective as a cheater, we can alter the system in any possible way by preparing states $\rho_{A,0}, \ldots, \rho_{A,N} \in D(\mathcal{A} \otimes \mathcal{M})$ that take into account that we cannot access Alice her space and Alice performs operations on the system, specified by the protocol. This is schematically represented in Figure 5.4.

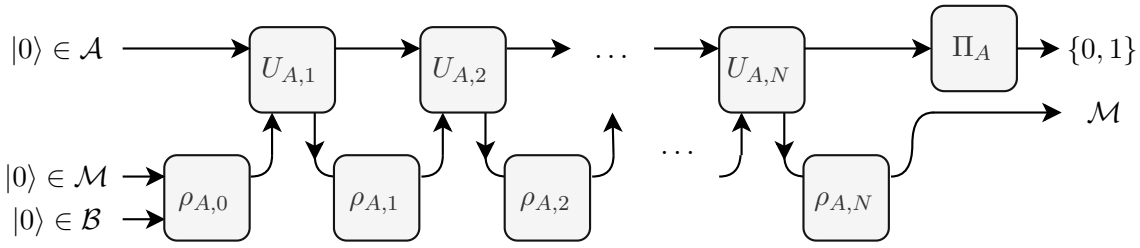


Figure 5.4: This diagram depicts the situation of Bob cheating. Bob replaces his unitary operations $U_{B,1}, \ldots, U_{B,N}$ with the preparation of states $\rho_{A,0}, \ldots, \rho_{A,N}$ while Alice performs the same operations as in Figure 5.3. Bob has knowledge on the state of the system at any time, but he is still unable to perform operations on the system $\mathcal{A}$ directly.

Now that we have specified any possible way a cheater can interact with the system, we want to optimize over all these possible ways such that Alice will measure a chosen outcome with maximum probability. This is shown in Proposition 5.4.4.

Proposition 5.4.4 (Cheating strategy by semidefinite programming) *Consider a quantum coin flipping protocol as in Definition 5.4.2 and suppose that Alice is honest. The optimal cheating strategy for Bob such that Alice measures the outcome 1 with probability as high as possible, is given by the optimal solution of the following semidefinite program over the density matrices $\rho_{A,0}, \ldots, \rho_{A,N} \in D(\mathcal{A} \otimes \mathcal{M})$:*

$$\begin{aligned} P^*_{A,1} = \sup\ & \mathrm{Tr}((\Pi_{A,1} \otimes I_{\mathcal{M}})\rho_{A,N}) \\ \text{s.t. } & \mathrm{Tr}_{\mathcal{M}}(\rho_{A,0}) = |0\rangle\langle 0|_{\mathcal{A}} \\ & \mathrm{Tr}_{\mathcal{M}}(\rho_{A,1}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,1}\rho_{A,0}U^\dagger_{A,1}) \\ & \mathrm{Tr}_{\mathcal{M}}(\rho_{A,2}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,2}\rho_{A,1}U^\dagger_{A,2}) \\ & \vdots \\ & \mathrm{Tr}_{\mathcal{M}}(\rho_{A,N}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,N}\rho_{A,N-1}U^\dagger_{A,N}) \\ & \rho_{A,0}, \ldots, \rho_{A,N} \in \mathrm{Herm}(\mathcal{A} \otimes \mathcal{M}) \\ & \rho_{A,0}, \ldots, \rho_{A,N} \succeq 0. \end{aligned} \tag{5.4.13}$$

Its dual is given by the semidefinite program over the operators $Z_{A,0}, \ldots, Z_{A,N} \in \mathrm{Herm}(\mathcal{A})$

$$\begin{aligned} D^*_{A,1} = \inf\ & \langle 0| Z_{A,0} |0\rangle \\ \text{s.t. } & Z_{A,0} \otimes I_{\mathcal{M}} \succeq U^\dagger_{A,1}(Z_{A,1} \otimes I_{\mathcal{M}})U_{A,1} \\ & Z_{A,1} \otimes I_{\mathcal{M}} \succeq U^\dagger_{A,2}(Z_{A,2} \otimes I_{\mathcal{M}})U_{A,2} \\ & \vdots \\ & Z_{A,N-1} \otimes I_{\mathcal{M}} \succeq U^\dagger_{A,N}(Z_{A,N} \otimes I_{\mathcal{M}})U_{A,N} \\ & Z_{A,N} \succeq \Pi_{A,1} \\ & Z_{A,0}, \ldots, Z_{A,N} \in \mathrm{Herm}(\mathcal{A}). \end{aligned} \tag{5.4.14}$$

Remark 5.4.2 *Some references pose the constraint $Z_{A,N} = \Pi_{A,1}$ in the dual semidefinite program, e.g. [7] and [53]. If we apply the primal and dual pair of Definition 3.2.1 we find the dual constraint $Z_{A,N} \succeq \Pi_{A,1}$ as suggested in Proposition 5.4.4 and this constraint can for example be found in [41]. However, the constraint $Z_{A,N} = \Pi_{A,1}$ will lead to the same optimal solution but yields a different feasible region. Kitaev's proof for the lower bound uses all of the dual constrains, including this last constraint. The statement and proof remains virtually the same and only acquires an extra inequality instead of an equality.*

Any feasible cheating strategy of the primal semidefinite program in Proposition 5.4.4 is represented by $N+1$ positive semidefinite matrices. It is a priori not clear that these operators have unit trace and therefore represent quantum states. This can be shown by considering for $j = 0$:

$$\mathrm{Tr}(\rho_{A,0}) = \mathrm{Tr}(\mathrm{Tr}_{\mathcal{M}}(\rho_{A,0})) = \mathrm{Tr}(|0\rangle\langle 0|_{\mathcal{A}}) = 1, \tag{5.4.15}$$

because $|0\rangle\langle 0|_{\mathcal{A}}$ is a valid quantum state and therefore has unit trace. Suppose for $j \in \{0, \ldots, N-1\}$ that $\mathrm{Tr}(\rho_{A,j}) = 1$, then

$$\begin{aligned} \mathrm{Tr}(\rho_{A,j+1}) &= \mathrm{Tr}(\mathrm{Tr}_{\mathcal{M}}(\rho_{A,j+1})) \\ &= \mathrm{Tr}(\mathrm{Tr}_{\mathcal{M}}(U_{A,j+1}\rho_{A,j}U_{A,j+1})) \\ &= \mathrm{Tr}(U_{A,j+1}\rho_{A,j}U_{A,j+1}) \\ &= \mathrm{Tr}(\rho_{A,j}) = 1. \end{aligned} \tag{5.4.16}$$

Thus by induction

$$\mathrm{Tr}(\rho_{A,0}) = \mathrm{Tr}(\rho_{A,1}) = \ldots = \mathrm{Tr}(\rho_{A,N}) = 1, \tag{5.4.17}$$

and therefore the operators $\rho_{A,0}, \ldots, \rho_{A,N}$ are indeed quantum states in the set $D(\mathcal{A} \otimes \mathcal{M})$.

We analyze why Proposition 5.4.4 is correct. First, suppose Alice is honest and Bob is dishonest and wants to enforce the outcome 1. This means Bob wants to maximize the probability after the final round of communication for Alice to measure the outcome 1. The state in the final round is $\rho_{A,N}$ and Alice measures only her own qubits, so she applies $\Pi_{A,1}$ to her system and nothing happens to the message qubits, i.e., the identity $I_{\mathcal{M}}$ is applied. The probability of measuring the outcome 1 is now given by taking the trace of $(\Pi_{A,1} \otimes I_{\mathcal{M}})\rho_{A,N}$ according to the measurements postulate of quantum mechanics.

The constraints are modeled according to Definition 5.4.2. Since Alice is honest, she starts with all of her private qubits in state $|0\rangle$. This is represented by the first constraint

$$\mathrm{Tr}_{\mathcal{M}}(\rho_{A,0}) = |0\rangle \langle 0|_{\mathcal{A}} . \tag{5.4.18}$$

Furthermore, Bob wants to alter the quantum states to his advantage but only has access to the message space. If we do not consider the message space, Alice applies the unitary operations according to the protocol, therefore

$$\mathrm{Tr}_{\mathcal{M}}(\rho_{A,j+1}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,j}\rho_{A,j}U_{A,j}^{\dagger}), \quad j = 0, \ldots, N-1. \tag{5.4.19}$$

This objective and these constraints forms the semidefinite program for finding the optimal cheating strategy.

If both players are fair, then the probability of aborting the protocol is 0. However, if a cheater is present it may be possible that an honest player detects a cheater and aborts the protocol. Since $\rho_{A,N}$ determines the state of the system from the perspective of Alice just before measuring, the probability of aborting the protocol is

$$\mathrm{Tr}((\Pi_{A,\varnothing} \otimes I_{\mathcal{M}})\rho_{A,N}). \tag{5.4.20}$$

We will now show that the the dual of the primal semidefinite program in Proposition 5.4.4 is correct. We first transform it to it's standard form

$$\sup\{\langle A, X\rangle : \Phi(X) = B, X \in \mathrm{Herm}((\mathcal{A} \otimes \mathcal{M})^{N+1}), X \succeq 0\}, \tag{5.4.21}$$

where $A \in \mathrm{Herm}((\mathcal{A} \otimes \mathcal{M})^{N+1})$, $B \in \mathrm{Herm}(\mathcal{A}^{N+1})$ and $\Phi \in T((\mathcal{A} \otimes \mathcal{M})^{N+1}, \mathcal{A}^{N+1})$ a Hermitian preserving superoperator.

We can do this by letting our decision variable be the direct sum of all density matrices of the intermediate states, i.e.,

$$\begin{aligned} X &= \rho_{A,0} \oplus \rho_{A,1} \oplus \cdots \oplus \rho_{A,N} \\ &= \begin{pmatrix} \rho_{A,0} & 0 & \ldots & 0 \\ 0 & \rho_{A,1} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \rho_{A,N} \end{pmatrix} \in \mathrm{Herm}((\mathcal{A} \otimes \mathcal{M})^{N+1}). \end{aligned} \tag{5.4.22}$$

By Lemma 2.2.6 this matrix is also positive semidefinite. We can define the constraint linear form as

$$\Phi(X) = \Phi_0(X) \oplus \Phi_1(X) \oplus \cdots \oplus \Phi_N(X) \tag{5.4.23}$$

where

$$\Phi_j(X) = \begin{cases} \mathrm{Tr}_{\mathcal{M}}(\rho_{A,0}) & \text{for } j = 0, \\ \mathrm{Tr}_{\mathcal{M}}(\rho_{A,j}) - \mathrm{Tr}_{\mathcal{M}}(U_{A,j}\rho_{A,j-1}U_{A,j}^{\dagger}) & \text{for } j = 1, \ldots, N, \end{cases} \tag{5.4.24}$$

and

$$\begin{aligned} B &= (|0\rangle\langle 0|_{\mathcal{A}}) \oplus 0 \oplus \cdots \oplus 0 \\ &= \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 \end{pmatrix} \in \mathrm{Herm}(\mathcal{A}^N). \end{aligned} \tag{5.4.25}$$

Furthermore, the objective is given by

$$\begin{aligned} A &= 0 \oplus \cdots \oplus 0 \oplus (\Pi_{A,1} \otimes I_{\mathcal{M}}) \\ &= \begin{pmatrix} 0 & \dots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & 0 \\ 0 & \dots & 0 & \Pi_{A,1} \otimes I_{\mathcal{M}} \end{pmatrix} \in \mathrm{Herm}((\mathcal{A} \otimes \mathcal{M})^N), \end{aligned} \tag{5.4.26}$$

then the optimization problem is of the form $\sup\{\langle A, X\rangle : \Phi(X) = B, X \succeq 0\}$. The dual is given by $\inf\{\langle B, Y\rangle : \Phi^*(Y) \succeq A, Y \in \mathrm{Herm}(\mathcal{A}^{N+1})\}$.

The dual variables can be written as

$$\begin{aligned} Y &= Z_{A,0} \oplus Z_{A,1} \oplus \cdots \oplus Z_{A,N} \\ &= \begin{pmatrix} Z_{A,0} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & Z_{A,N} \end{pmatrix} \in \mathrm{Herm}(\mathcal{A}^{N+1}). \end{aligned} \tag{5.4.27}$$

This means the dual objective becomes

$$\begin{aligned} \langle B, Y\rangle &= \langle |0\rangle\langle 0|_{\mathcal{A}}, Z_{A,0}\rangle + \langle 0, Z_{A,1}\rangle + \cdots + \langle 0, Z_{A,N}\rangle \\ &= \langle |0\rangle\langle 0|_{\mathcal{A}}, Z_{A,0}\rangle = \langle 0|Z_{A,0}|0\rangle_{\mathcal{A}} \end{aligned} \tag{5.4.28}$$

and the adjoint of Φ is given by

$$\Phi^\dagger(Y) = \Phi_0^\dagger(Y) \oplus \cdots \oplus \Phi_N^\dagger(Y), \tag{5.4.29}$$

where

$$\Phi_j^\dagger(Y) = \begin{cases} Z_{A,j} \otimes I_{\mathcal{M}} - U_{A,j+1}^\dagger (Z_{A,j+1} \otimes I_{\mathcal{M}}) U_{A,j+1} & \text{if } j = 0, \dots, N-1, \\ Z_{A,N} \otimes I_{\mathcal{M}} & \text{if } j = N. \end{cases} \tag{5.4.30}$$

To prove this operator is indeed the adjoint we show

$$\begin{aligned}
\langle \Phi(X), Y\rangle &= \langle \Phi_0(X)\oplus\cdots\oplus\Phi_N(X), Z_{A,0}\oplus\cdots\oplus Z_{A,N}\rangle = \sum_{j=0}^{N}\langle \Phi_j(X), Z_{A,j}\rangle \\
&= \langle \mathrm{Tr}_{\mathcal{M}}(\rho_{A,0}), Z_{A,0}\rangle + \sum_{j=1}^{N}\langle \mathrm{Tr}_{\mathcal{M}}(\rho_{A,j}) - \mathrm{Tr}_{\mathcal{M}}(U_{A,j}\rho_{A,j-1}U_{A,j}^{\dagger}), Z_{A,j}\rangle \\
&= \langle \rho_{A,0}, Z_{A,0}\otimes I_{\mathcal{M}}\rangle + \sum_{j=1}^{N}\Big(\langle \rho_{A,j}, Z_{A,j}\otimes I_{\mathcal{M}}\rangle - \langle U_{A,j}\rho_{A,j-1}U_{A,j}^{\dagger}, Z_{A,j}\otimes I_{\mathcal{M}}\rangle\Big) \\
&= \langle \rho_{A,0}, Z_{A,0}\otimes I_{\mathcal{M}}\rangle + \sum_{j=1}^{N}\Big(\langle \rho_{A,j}, Z_{A,j}\otimes I_{\mathcal{M}}\rangle - \langle \rho_{A,j-1}, U_{A,j}^{\dagger}(Z_{A,j}\otimes I_{\mathcal{M}})U_{A,j}\rangle\Big) \\
&= \langle \rho_{A,0}, Z_{A,0}\otimes I_{\mathcal{M}} - U_{A,1}^{\dagger}(Z_{A,1}\otimes I_{\mathcal{M}})U_{A,1}\rangle + \cdots \\
&\cdots + \langle \rho_{A,N-1}, Z_{A,N-1}\otimes I_{\mathcal{M}} - U_{A,N}^{\dagger}(Z_{A,N}\otimes I_{\mathcal{M}})U_{A,N}\rangle + \langle \rho_{A,N}, Z_{A,N}\otimes I_{\mathcal{M}}\rangle \\
&= \sum_{j=0}^{N}\langle X, \Phi_j^{\dagger}(Y)\rangle = \langle X, \Phi^{\dagger}(Y)\rangle.
\end{aligned} \tag{5.4.31}$$

We can now write $\Phi^{\dagger}(Y)$ into a list of $N+1$ constraints that provides a clearer formulation of the dual.

We will now consider the situation where Bob is honest and Alice is dishonest. We can look at Figure 5.3 and replace every interaction of Alice in the system with the preparation of a state $\rho_{B,j}$ for $j\in\{0,\ldots,N\}$. This is shown in Figure 5.5.

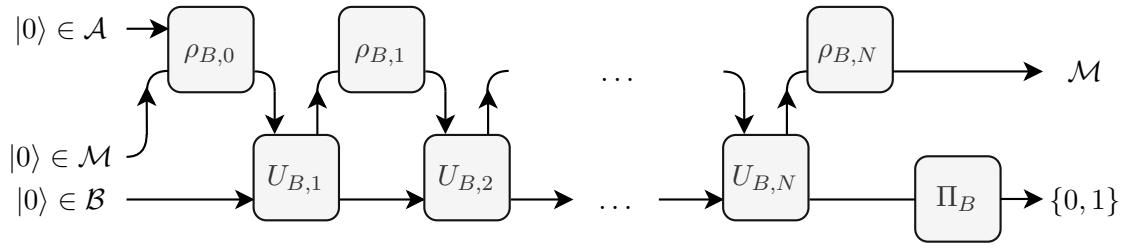


Figure 5.5: This diagram depicts the situation of Alice cheating and is thereby similar to Figure 5.4. Note that Alice can perform a last interaction with the system after she received the system $\mathcal{M}$ from Bob in round N. Hence the semidefinite programs are of the same form.

If Bob is honest and Alice wants to cheat (say she wants to force outcome 1), then the primal SDP to find her optimal cheating strategy is analogously given by

$$\begin{aligned}
P_{B,1}^{*} = \sup\ & \mathrm{Tr}((I_{\mathcal{M}}\otimes\Pi_{B,1})\rho_{B,N}) \\
\text{s.t. }& \mathrm{Tr}_{\mathcal{M}}(\rho_{B,0}) = |0\rangle\langle 0|_{\mathcal{B}} \\
& \mathrm{Tr}_{\mathcal{M}}(\rho_{B,1}) = \mathrm{Tr}_{\mathcal{M}}(U_{B,1}\rho_{B,0}U_{B,1}^{\dagger}) \\
& \mathrm{Tr}_{\mathcal{M}}(\rho_{B,2}) = \mathrm{Tr}_{\mathcal{M}}(U_{B,2}\rho_{B,1}U_{B,2}^{\dagger}) \\
& \vdots \\
& \mathrm{Tr}_{\mathcal{M}}(\rho_{B,N}) = \mathrm{Tr}_{\mathcal{M}}(U_{B,N}\rho_{B,N-1}U_{B,N}^{\dagger}) \\
& \rho_{B,0},\ldots,\rho_{B,N}\in\mathrm{Herm}(\mathcal{M}\otimes\mathcal{B}) \\
& \rho_{B,0},\ldots,\rho_{B,N}\succeq 0.
\end{aligned} \tag{5.4.32}$$

Again, the dual of this SDP is given by the semidefinite program over the operators $Z_{B,0}, \ldots, Z_{B,N} \in \mathrm{Herm}(\mathcal{B})$

$$\begin{aligned} D^*_{B,1} = \inf\ & \langle 0| Z_{B,0} |0\rangle \\ \text{s. t. } & I_{\mathcal{M}} \otimes Z_{B,0} \succeq U^\dagger_{B,1}(I_{\mathcal{M}} \otimes Z_{B,1})U_{B,1} \\ & I_{\mathcal{M}} \otimes Z_{B,1} \succeq U^\dagger_{A,2}(I_{\mathcal{M}} \otimes Z_{B,2})U_{B,2} \\ & \vdots \\ & I_{\mathcal{M}} \otimes Z_{B,N-1} \succeq U^\dagger_{B,N}(I_{\mathcal{M}} \otimes Z_{B,N})U_{B,N} \\ & Z_{B,N} \succeq \Pi_{B,1} \\ & Z_{B,0}, \ldots, Z_{B,N} \in \mathrm{Herm}(\mathcal{B}). \end{aligned} \tag{5.4.33}$$

If we have a given quantum coin flipping protocol in the form of Definition 5.4.2, we determine the optimal cheating probabilities and thus the bias by solving four semidefinite programs. An important property that is essential in Kitaev's proof, is strong duality of the semidefinite program.

Lemma 5.4.5 *Strong duality holds for the semidefinite program and its dual given in Proposition 5.4.4, i.e., $P^*_{A,1} = D^*_{A,1}$ and there exist a primal optimal solution that attains the optimum. Analogously, strong duality holds for the semidefinite primal and dual pair in Program 5.4.32 and 5.4.33: $P^*_{B,1} = D^*_{B,1}$ and there exists a primal optimal solution that at taints the optimum.*

Proof: Clearly the feasible region is non-empty, because playing honest is a strategy. We prove that the feasible region is bounded. Let $X = \rho_{A,0} \oplus \cdots \oplus \rho_{A,N} \succeq 0$ be a feasible solution for the primal problem. Then

$$\|X\|^2 = \sum_{j=0}^{N} \|\rho_{A,j}\|^2. \tag{5.4.34}$$

For arbitrary $j \in [N]$, let $\lambda_1, \ldots, \lambda_r \geq 0$ be the eigenvalues of $\rho_{A,j}$, then

$$\|\rho_{A,j}\|^2 = \mathrm{Tr}(\rho^2_{A,j}) = \lambda_1^2 + \cdots + \lambda_r^2 \leq (\lambda_1 + \cdots + \lambda_r)^2 = (\mathrm{Tr}(\rho_{A,j}))^2 = 1. \tag{5.4.35}$$

Hence for every feasible solution X we have $\|X\| \leq \sqrt{N+1} < \infty$, and thus the feasible region is bounded.

We will now determine a strictly feasible dual solution. Let $Z_{A,j} = (2 + N - j)I_{\mathcal{A}} \in \mathrm{Herm}(\mathcal{A})$ for every $j \in \{0, \ldots, N\}$, then

$$Z_{A,N} = 2I_{\mathcal{A}} \succ \Pi_{A,1}. \tag{5.4.36}$$

and

$$\begin{aligned} Z_{A,j} \otimes I_{\mathcal{M}} &= (2 + N - j)I_{\mathcal{A}\otimes\mathcal{M}} \succ (2 + N - (j+1))I_{\mathcal{A}\otimes\mathcal{M}} \\ &= U^\dagger_{A,j+1}((2 + N - (j+1))I_{\mathcal{A}} \otimes I_{\mathcal{M}})U_{A,j+1} = U^\dagger_{A,j+1}(Z_{A,j+1} \otimes I_{\mathcal{M}})U_{A,j+1}. \end{aligned} \tag{5.4.37}$$

Thus $Y = Z_{A,0} \otimes \cdots \otimes Z_{A,N}$ is a solution for which $\Phi^\dagger(Y) \succ A$. Slater's condition is met and thus strong duality holds. The same argument applies when Alice is a cheater and Bob is honest. □

We are now ready to prove Kitaev's lower bound for strong quantum coin flipping.

Proof of Theorem 5.4.3 Consider a dual feasible solution of the semidefinite program and denote this in both cases where Alice and Bob are honest by $Z_{A,0}, \ldots, Z_{A,N}$ and $Z_{B,0}, \ldots, Z_{B,N}$ respectively, such that $P^*_{A,1} + \delta = \langle 0| Z_{A,0} |0\rangle$ and $P_{B,1} + \delta = \langle 0| Z_{B,0} |0\rangle$ for some arbitrary $\delta > 0$.

We define a family of quantum states and numbers. For $j \in \{0, \ldots, N\}$ denote the following (ket-) states

$$|\psi_j\rangle = (I_{\mathcal{A}} \otimes U_{B,j})(U_{A,j} \otimes I_{\mathcal{B}}) \cdots (I_{\mathcal{A}} \otimes U_{B,1})(U_{A,1} \otimes I_{\mathcal{B}}) |0\rangle \in \mathcal{H}, \tag{5.4.38}$$

that is, $|\psi_j\rangle$ is the state of the system after round j if both players are honest. Based on the optimal dual solutions we also define the numbers

$$F_j = \langle\psi_j| (Z_{A,j} \otimes I_{\mathcal{M}} \otimes Z_{B,j}) |\psi_j\rangle \in \mathbf{R}\,. \tag{5.4.39}$$

We will now prove the following equations that immediate lead to the full proof

$$(P^*_{A,1} + \delta)(P^*_{B,1} + \delta) = F_0, \tag{5.4.40}$$

$$F_j \geq F_{j+1} \quad \text{for } j = 0, 1, \ldots, N-1, \tag{5.4.41}$$

$$F_N \geq \frac{1}{2}. \tag{5.4.42}$$

In Figure 5.6 the inequalities are shown and which parts of the four optimization programs lead to the inequalities.

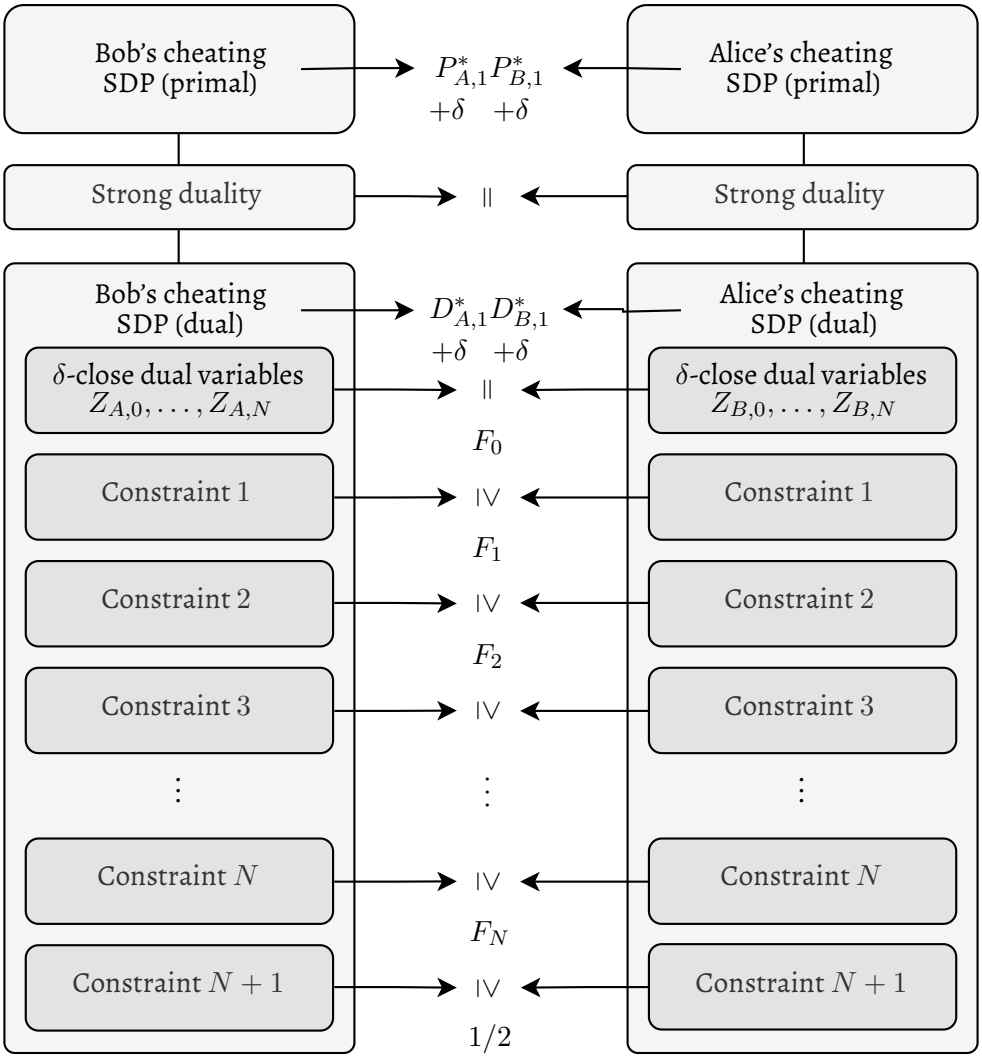


Figure 5.6: Schematical outline of Kitaev's proof. The proof uses strong duality, dual optimal variables and all dual constraints to derive a chain of inequalities that lead to the result.

To prove Equation 5.4.40 we have use strong duality from Lemma 5.4.5 We can now calculate the product

$$\begin{aligned}(P^*_{B,1}+\delta)(P^*_{A,1}+\delta) &= \langle 0|_{\mathcal{A}} Z_{A,0} |0\rangle_{\mathcal{A}} \langle 0|_{\mathcal{B}} Z_{B,0} |0\rangle_{\mathcal{B}} \\ &= \langle 0|_{\mathcal{A}} Z_{A,0} |0\rangle_{\mathcal{A}} \langle 0|_{\mathcal{M}} I_{\mathcal{M}} |0\rangle_{\mathcal{M}} \langle 0|_{\mathcal{B}} Z_{B,0} |0\rangle_{\mathcal{B}} \\ &= \langle 0|_{\mathcal{A}\otimes\mathcal{M}\otimes\mathcal{B}} (Z_{A,0}\otimes I_{\mathcal{M}}\otimes Z_{B,0}) |0\rangle_{\mathcal{A}\otimes\mathcal{M}\otimes\mathcal{B}} = F_0.\end{aligned} \tag{5.4.43}$$

Next, to prove Equation 5.4.41 we use the dual constrains of the semidefinite program pair. From the dual SDP where Alice is honest and Bob is dishonest we have

$$Z_{A,j}\otimes I_{\mathcal{M}} \succeq U^\dagger_{A,j+1}(Z_{A,j+1}\otimes I_{\mathcal{M}})U_{A,j+1}, \quad \text{for } j=0,\ldots,N-1. \tag{5.4.44}$$

Similarly, if Alice is dishonest and Bob is honest we have the constraint

$$I_{\mathcal{M}}\otimes Z_{B,j} \succeq U^\dagger_{B,j+1}(I_{\mathcal{M}}\otimes Z_{B,j+1})U_{B,j+1}, \quad \text{for } j=0,\ldots,N-1. \tag{5.4.45}$$

We will now apply this

$$\begin{aligned}F_j &= \langle\psi_j| Z_{A,j}\otimes I_{\mathcal{M}}\otimes Z_{B,j} |\psi_j\rangle \\ &\geq \langle\psi_j| U^\dagger_{A,j+1}(Z_{A,j+1}\otimes I_{\mathcal{M}})U_{A,j+1}\otimes Z_{B,j} |\psi_j\rangle \\ &= \langle\psi_j| (U^\dagger_{A,j+1}\otimes I_{\mathcal{B}})(Z_{A,j+1}\otimes I_{\mathcal{M}}\otimes Z_{B,j})(U_{A,j+1}\otimes I_{\mathcal{B}}) |\psi_j\rangle \\ &\geq \langle\psi_j| (U^\dagger_{A,j+1}\otimes I_{\mathcal{B}})\cdots \\ &\qquad \cdots(Z_{A,j+1}\otimes (U^\dagger_{B,j+1}(I_{\mathcal{M}}\otimes Z_{B,j+1})U_{B,j+1}))\cdots \\ &\qquad \cdots(U_{A,j+1}\otimes I_{\mathcal{B}}) |\psi_j\rangle \\ &= \langle\psi_j| (U^\dagger_{A,j+1}\otimes I_{\mathcal{B}})(I_{\mathcal{A}}\otimes U^\dagger_{B,j+1})\cdots \\ &\qquad \cdots(Z_{A,j+1}\otimes I_{\mathcal{M}}\otimes Z_{B,j+1})\cdots \\ &\qquad \cdots(I_{\mathcal{A}}\otimes U_{B,j+1})(U_{A,j+1}\otimes I_{\mathcal{B}}) |\psi_j\rangle \\ &= \langle\psi_{j+1}| Z_{A,j+1}\otimes I_{\mathcal{M}}\otimes Z_{B,j+1} |\psi_{j+1}\rangle = F_{j+1}.\end{aligned} \tag{5.4.46}$$

Finally we prove Equation 5.4.42. Note that for any state $|\varphi\rangle \in \mathcal{A}\otimes\mathcal{M}\otimes\mathcal{B}$ we have

$$\langle\varphi| (Z_{A,N}\otimes I_{\mathcal{M}}\otimes Z_{B,N}) |\varphi\rangle \geq \langle\varphi| (\Pi_{A,1}\otimes I_{\mathcal{M}}\otimes\Pi_{B,1}) |\varphi\rangle \tag{5.4.47}$$

In particular if we take $|\varphi\rangle = |\psi_N\rangle$, then

$$F_N \geq \langle\psi_N| (\Pi_{A,1}\otimes I_{\mathcal{M}}\otimes I_{\mathcal{B}})(I_{\mathcal{A}}\otimes I_{\mathcal{M}}\otimes\Pi_{B,1}) |\psi_N\rangle = \frac{1}{2}. \tag{5.4.48}$$

This concludes the proof that $(P^*_{A,1}+\delta)(P^*_{B,1}+\delta) \geq 1/2$ and thus $\max\{P^*_{A,1}, P^*_{B,1}\}+\delta \geq 1/\sqrt{2}$, from which it immediately follows that $\max\{P^*_{A,1}, P^*_{B,1}\} \geq 1/\sqrt{2}$, since $\delta > 0$ was arbitrary. Thus the bias of any two player strong quantum coin flipping protocol is

$$\varepsilon \geq \max\{P^*_{A,1}, P^*_{B,1}\} - \frac{1}{2} \geq \frac{1}{\sqrt{2}} - \frac{1}{2} \approx 0.20711\ldots \tag{5.4.49}$$

The remarkable fact of this theorem and lower bound is how the proof uses two pairs of optimization programs and combines their dual constraints to eliminate the dependence of the unitary operations and measurements and therefore applies to any protocol.

From this proof it is clear that in the same way $\max\{P^*_{A,0}, P^*_{B,0}\} \geq 1/\sqrt{2}$ and this leads to the same conclusion for strong quantum coin flipping. This proof does however *not* apply to the setting of weak coin flipping, simply because applying the proof to $P^*_{A,0}P^*_{B,1}$ yields the trivial lower bound of 0 in the final step.

5.5 Coin Flipping beyond Kitaev's Proof: Optimal Coin Flipping

Kitaev's proof of a lower bound on the strong bias of a coin flipping protocol is based on the pair of inequalities $\max\{P^*_{A,0}, P^*_{B,0}\} \geq 1/\sqrt{2}$ and $\max\{P^*_{A,1}, P^*_{B,1}\} \geq 1/\sqrt{2}$. In the strong bias of a coin flipping protocol, all four values are taken into account and thus the bias is bounded from below by $1/\sqrt{2} - 1/2$. However, in the case we want to assess the weak bias of a quantum coin flipping, we only look at pairs $(P^*_{A,0}, P^*_{B,1})$ *or* $(P^*_{A,1}, P^*_{B,0})$ depending on the preferences. As a result, Kitaev's proof does *not* apply on the weak bias of a protocol.

After Kitaev's bound, new results on the weak bias of quantum coin flipping protocol were found. One of these protocols by Mochon [14] in 2004 has a bias of approximately $0.19218\ldots$, which is strictly less than Kitaev's bound on the strong bias. This clearly indicated a difference between protocols with a good weak bias and those with a good strong bias. In 2007, the optimal lower bound for the weak bias of quantum coin flipping was completely resolved by Carlos Mochon. He showed that any arbitrarily small bias is possible.

Theorem 5.5.1 (C. Mochon, 2007) *There exist quantum coin flipping protocols for any weak bias* $\varepsilon_{WCF} > 0$.

Based on this result, in 2009 Chailloux and Kerenidis were able to prove that Kitaev's lower bound on the strong bias is tight by costructing protocols with a good strong bias from imbalanced protocols with a good weak bias.

Theorem 5.5.2 (Chailloux and Kerenidis, 2009 [17]) *For any* $\delta > 0$ *there exists a coin flipping protocol with strong bias*

$$\varepsilon < \frac{1}{\sqrt{2}} - \frac{1}{2} + \delta. \tag{5.5.1}$$

How both results on the weak and strong bias are connected can be found in Sikora's lecture notes [53]. Both optimal results on the weak and strong bias of quantum coin flipping show the difference between classical and quantum information theory. This observation is analogously to Bell's inequality [33].

5.6 Optimization of Secondary Preferences and Expectation

In the previous section, Proposition 5.4.4 finds a feasible cheating strategy that optimizes the probability of a particular chosen outcome. However, besides optimizing this chosen outcome, a player might also want to minimize the probability of being caught. We can relatively easy model this extra requirement.

For a given quantum coin flipping protocol we have four optimal cheating probabilities $P^*_{A,0}$, $P^*_{A,1}$, $P^*_{B,0}$ and $P^*_{B,1}$ as a result of the optimization programs in Proposition 5.4.4. Suppose Bob is a cheater and wants to enforce the outcome 1. All other scenarios are analogous. Solving this semidefinite program we get a feasible solution $\rho_{A,0}, \ldots, \rho_{A,N} \in D(\mathcal{A} \otimes \mathcal{M})$ with $\mathrm{Tr}((\Pi_{A,1} \otimes I_{\mathcal{M}})\rho_{A,N}) = P^*_{A,1}$. The optimum can be attained as we have seen in the proof of strong duality (Lemma 5.4.5). This solution may not be a unique solution that attains this optimum.

For all solutions $\rho_{A,0}, \ldots, \rho_{A,N}$ that yield an optimal probability of measuring the outcome 1, we can search for the optimal solution that also maximizes the probability of measuring the outcome 0. This is equivalent to minimizing the probability of aborting the protocol. Analogously, we can look for the solution that maximizes the probability of aborting the protocol and hence minimizes the probability of finding the outcome 0. This optimization problem is a simple extension of the primary optimization problem of Proposition 5.4.4 by changing the objective and

inserting the constraint

$$\mathrm{Tr}((\Pi_{A,1} \otimes I_{\mathcal{M}})\rho_{A,N}) = P^*_{A,1}, \tag{5.6.1}$$

into the semidefinite program. The secondary semidefinite program and its dual formulation are shown in Proposition 5.6.1.

Proposition 5.6.1 (Cheating strategy for secondary optimization) *Consider a quantum coin flipping protocol as in Definition 5.4.2 and suppose that Alice is honest. Let $P^*_{B,1}$ be the optimum of 5.4.4. The best strategy that minimizes the probability of aborting (or equivalently, maximizes the probability of measuring 0) whilst attaining optimal probability to enforce outcome one is given by the following semidefinite program:*

$$\begin{aligned}
\Gamma^*_{A,1} = \sup\ & \mathrm{Tr}((\Pi_{A,0} \otimes I_{\mathcal{M}})\rho_{A,N}) \\
\text{s. t. } & \mathrm{Tr}_{\mathcal{M}}(\rho_{A,0}) = |0\rangle\langle 0|_{\mathcal{A}} \\
& \mathrm{Tr}_{\mathcal{M}}(\rho_{A,1}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,1}\rho_{A,0}U^\dagger_{A,1}) \\
& \mathrm{Tr}_{\mathcal{M}}(\rho_{A,2}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,2}\rho_{A,1}U^\dagger_{A,2}) \\
& \vdots \\
& \mathrm{Tr}_{\mathcal{M}}(\rho_{A,N}) = \mathrm{Tr}_{\mathcal{M}}(U_{A,N}\rho_{A,N-1}U^\dagger_{A,N}) \\
& \mathrm{Tr}((\Pi_{A,1} \otimes I_{\mathcal{M}})\rho_{A,N}) = P^*_{A,1} \\
& \rho_{A,0}, \ldots, \rho_{A,N} \in \mathrm{Herm}(\mathcal{A} \otimes \mathcal{M}) \\
& \rho_{A,0}, \ldots, \rho_{A,N} \succeq 0.
\end{aligned} \tag{5.6.2}$$

Its dual is given by the semidefinite program over the operators $Z_{A,0}, \ldots, Z_{A,N} \in \mathrm{Herm}(\mathcal{A}), y \in \mathbf{R}$

$$\begin{aligned}
\Delta^*_{A,1} = \inf\ & \langle 0| Z_{A,0} |0\rangle + yP^*_{A,1} \\
\text{s. t. } & Z_{A,0} \otimes I_{\mathcal{M}} \succeq U^\dagger_{A,1}(Z_{A,1} \otimes I_{\mathcal{M}})U_{A,1} \\
& Z_{A,1} \otimes I_{\mathcal{M}} \succeq U^\dagger_{A,2}(Z_{A,2} \otimes I_{\mathcal{M}})U_{A,2} \\
& \vdots \\
& Z_{A,N-1} \otimes I_{\mathcal{M}} \succeq U^\dagger_{A,N}(Z_{A,N} \otimes I_{\mathcal{M}})U_{A,N} \\
& Z_{A,N} \otimes I_{\mathcal{M}} + y(\Pi_{A,1} \otimes I_{\mathcal{M}}) \succeq \Pi_{A,0} \otimes I_{\mathcal{M}} \\
& Z_{A,0}, \ldots, Z_{A,N} \in \mathrm{Herm}(\mathcal{A}), y \in \mathbf{R}\,.
\end{aligned} \tag{5.6.3}$$

Of course, Proposition 5.6.1 can also be applied to situation in which Bob is honest and all preferences of the outcomes 0,1 and $\varnothing$. This leads to the optimal values $\Gamma^*_{A,0}, \Gamma^*_{A,1}, \Gamma^*_{B,0}, \Gamma^*_{B,1}$. The maximum of these four values is called the *secondary bias*.

In this approach, the first preference is the most important, followed by the second and what is left is the least preferred. There might be situations in which this is a little bit more sophisticated any we do not necessarily value one outcome absolutely more than another. This can best be explained with an example. Suppose the coin flip leads to the following actions: If the outcome is 0, Bob has to pay one dollar to Alice. Similarly, if the outcome is 1, Alice has to pay one dollar to Bob. If a cheater gets caught, he or she has a to pay a fine of five dollar. Suppose that both players are honest, then their expected win is 0. Suppose that we have a coin flipping protocol and Bob executes a cheating strategy that leads to a 75% chance of the outcome 1. If this cheating strategy also has a 20% chance of being caught and 5% chance of finding the outcome 0, the expected amount of money is $75\% \cdot 1 + 5\% \cdot -1 + 20\% \cdot -5 = -0.30$ dollar. Although Bob is cheating and the chance of the favorable outcome increases, the expected win decreases

and we have to conclude that this is a bad cheating strategy. What we want to optimize is not the favorable outcome, but the expected amount of money.

This can also be written as a minor variation on the previous semidefinite programs. Let $\mathcal{F}_A(\mathcal{P})$ be the feasible set of cheating strategies $\rho_{A,0}, \ldots, \rho_{A,N} \in \mathrm{Herm}(\mathcal{A} \otimes \mathcal{M})$, when Alice is honest in a quantum coin flipping protocol $\mathcal{P} = (U_{A,1}, U_{B,1}, \ldots, U_{A,N}, U_{B,N}, \Pi_A, \Pi_B)$. Suppose that a protocol yields Bob the amounts $y_0, y_1, y_\varnothing \in \mathbf{R}$ for outcomes $0, 1$ and $\varnothing$ respectively. Let

$$P_{A,a}(X) = \mathrm{Tr}((\Pi_{A,a} \otimes I_{\mathcal{M}})\rho_{A,N}), \quad \text{for } a \in \{0, 1, \varnothing\}. \tag{5.6.4}$$

The expected yield is therefore

$$\begin{aligned} \mathbf{E}(Y) &= y_0 P_{A,0}(X) + y_1 P_{A,1}(X) + y_\varnothing P_{A,\varnothing}(X) \\ &= \mathrm{Tr}(((y_0\Pi_{A,0} + y_1\Pi_{A,1} + y_\varnothing\Pi_{A,\varnothing}) \otimes I_{\mathcal{M}})\rho_{A,N}) \end{aligned} \tag{5.6.5}$$

The cheating strategy that optimizes the expected yield is therefore given by the semidefinite optimization program

$$\sup\{\mathrm{Tr}((\Lambda \otimes I_{\mathcal{M}})\rho_{A,N} : \rho_{A,0}, \ldots, \rho_{A,N} \in \mathcal{F}_A(\mathcal{P})\}, \tag{5.6.6}$$

where $\Lambda = y_0\Pi_{A,0} + y_1\Pi_{A,1} + y_\varnothing\Pi_{A,\varnothing} \in \mathrm{Herm}(\mathcal{A})$.

Note that the optimization in which we do only optimize the particular outcome 1 in Proposition 5.4.4 is equivalent to letting $(y_0, y_1, y_\varnothing) = (0, 1, 0) \in \mathbf{R}^3$, because then $\Lambda = \Pi_{A,1}$.

The secondary optimization of Proposition 5.6.1 can also be cast into the form of Program 5.6.6 by letting $(y_0, y_1, y_\varnothing) = (1/n, 1, 0)$ for some $n \geq 1$ and solving the program as $n \to \infty$. In this way the outcome 1 is arbitrarily more important than the outcome 0 and the outcome 0 is infinitely more important than aborting the protocol. The optimal value will tend to $P^*_{A,1}$ and the optimal value of the secondary optimization program is

$$\Gamma^*_{A,1} = \mathrm{Tr}((\Pi_{A,0} \otimes I_{\mathcal{M}})\rho^*_{A,N}), \tag{5.6.7}$$

where $\rho^*_{A,0}, \ldots, \rho^*_{A,N}$ is optimal for Program 5.6.6 with $\Lambda_n = \frac{1}{n}\Pi_{A,0} + \Pi_{A,1}$ as $n \to \infty$. However, this does need a more formal description of what convergence means in this setting and a proof that this approach does indeed yield these solutions.

5.7 Strong Unbalanced Quantum Coin Flipping

When defining quantum coin flipping protocols, we mentioned that we want the coin to be balanced, i.e., both outcomes have to appear with equal probabilities. This lead to Kitaev's tight lower bound on the bias. Depending on the application, we may consider a trade-off between the probability of both outcomes of the coin if both players are fair and the minimum bias that can be achieved by protocols based on this (un)balanced coin. If a slightly unbalanced coin leads to protocols that have a substantially lower bias than Kitaev's bound, we might prefer this situation over balanced coin flipping. We show that this is not the case and a balanced coin leads to the lowest possible bias.

Suppose we have an unbalanced strong quantum coin flipping protocol, i.e., Alice and Bob want to generate the outcome of a coin with probability p for the outcome 1 and $1 - p$ for the outcome 0, for some $p \in [0, 1]$. We have to redefine the bias in terms of the difference with the corresponding outcome. For example, if Alice is a cheater, she can force outcome 0 with probability

$P^*_{B,0}$, then the deviation for this situation is $P^*_{B,0} - (1-p)$. We can do this similarly for the other three situations and define the bias by

$$\begin{aligned}\varepsilon_p &= \max\{P^*_{A,0} - (1-p), P^*_{B,0} - (1-p), P^*_{A,1} - p, P^*_{B,1} - p\} \\ &= \max\{\max\{P^*_{A,0}, P^*_{B,0}\} - (1-p), \max\{P^*_{A,1}, P^*_{B,1}\} - p\}.\end{aligned} \tag{5.7.1}$$

Note that if $p = 1/2$, this definition corresponds to Definition 5.4.2.

If we apply Kitaev's proof to unbalanced coin flipping we get the inequality $P^*_{A,0}P^*_{B,0} \geq 1 - p$ for outcome 0 and thus

$$\max\{P^*_{A,0}, P^*_{B,0}\} \geq \sqrt{1-p}. \tag{5.7.2}$$

Similarly, for outcome 1, we get the inequality $P^*_{A,1}P^*_{B,1} \geq p$ and thus

$$\max\{P^*_{A,1}, P^*_{B,1}\} \geq \sqrt{p}. \tag{5.7.3}$$

Combining this the definition of the bias in Eq. 5.7.1 we get

$$\varepsilon_p \geq \max\{\sqrt{1-p} - (1-p), \sqrt{p} - p\} =: \mu(p). \tag{5.7.4}$$

Note that this lower bound $\mu(p)$ has a local minimum at $p = 1/2$, corresponding to the situation of fair strong coin flipping.

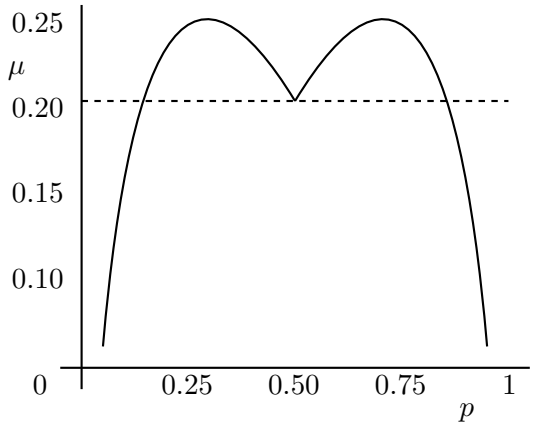


Figure 5.7: The lower bound for the bias as a result of Kiteav's proof for unbalanced coin flipping. The dotted line is the optimal bias for balanced coin flipping, where $\mu(1/2) = 1/\sqrt{2} - 1/2$.

The equation $\mu(p) = 1/\sqrt{2} - 1/2$ has three solutions: $p = 3/2 - \sqrt{2}, 1/2, \sqrt{2} - 1/2$. This means that in the regions $(0.0858, 1/2)$ and $(1/2, 0.9142)$ any quantum coin flipping protocol will perform worse than the balanced case where $p = 1/2$. Since the derivative of μ close to $p = 1/2$ is $\pm(1 - 1/\sqrt{2}) \approx \pm 0.2929$, we can conclude that the best way to have a low bias is to choose for balanced coin flipping, i.e., $p = 1/2$.

The regions outside the interval $(0.0858, , 0.9142)$ do have a lower bound on the bias less than $1/\sqrt{2} - 1/2$, but in this situation will be less useful in practical applications.

5.8 Quantum Coin Flipping as a Quantum Computing Circuit

Quantum coin flipping protocols such as the one of Ambainis [12] and Berlín et al. [15] are generally not stated in terms of the standard form of Definition 5.4.2. Protocols may also include procedures such as flipping private (classical) coins or communicating classical information.

By Stinespring's representation 4.3.4 this can be captured by unitary operations on a larger Euclidean space. We will make this explicit by viewing these operations in terms of quantum circuits. This contribution is new and allows for a flexible modelling of coin flipping protocols.

It is possible to create a private fair coin with a quantum circuit. We can simply do this by starting with a single qubit in the state $|0\rangle \in \mathbf{C}^2$, apply the Hadamard transform to obtain the state $(|0\rangle + |1\rangle)/\sqrt{2}$ and measure the result in the standard basis $\{\Pi_0 = |0\rangle\langle 0|, \Pi_1 = |1\rangle\langle 1|\}$. The result will be in one of the states $|0\rangle$ or $|1\rangle$ with probability $1/2$. The circuit is shown in Figure 5.8.

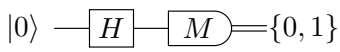


Figure 5.8: This circuit creates a fair, private coin flip through quantum randomness.

We can use the result of the measurement in the circuit to apply to other operations in the system. The trick to postpone this measurement, is to apply an operation that is controlled by the qubit in superposition and then measure the qubits afterwards. If the private coin collapses, the rest of the system will collapse to a state that corresponds to the action of the coin flip.

More explicitly, if we have state $|\psi\rangle$ and we want to apply an operation U to $|\psi\rangle$ based on a random coin flip, then a direct way of doing this is represented in Figure 5.9.

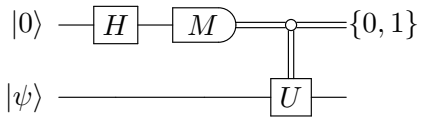


Figure 5.9: In this circuit we create a fair random coin flip by using quantum randomness as in Figure 5.8 and then use the outcome to decide whether or not we apply the gate U on the second qubit.

However, the circuit in Figure 5.9 requires us to first measure and then apply the operation U. Surprisingly, we can interchange the controlled operation and the measurement is we substitute the classical controlled operation by a quantum controlled operation. This is shown in Figure 5.10.

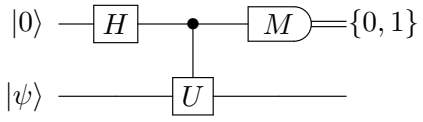


Figure 5.10: This system has the same outcome as the circuit in Figure 5.9. In this circuit the controlled gate U is applied before measuring the coin. After measuring the system, the results are indistinguishable.

To see why this works, we look ad the quantum controlled-U operation, defined by

$$cU = |0\rangle\langle 0| \otimes I_2 + |1\rangle\langle 1| \otimes U. \tag{5.8.1}$$

We show if the outcome of the coin flip is 0, then clearly

$$(I_2 \otimes I_2)(\Pi_0 \otimes I) = \Pi_0 \otimes I = (\Pi_0 \otimes I_2)(I_2 \otimes I_2). \tag{5.8.2}$$

If, on the other hand, the outcome of the coin flip is 1, then

$$(I_2 \otimes U)(\Pi_1 \otimes I) = \Pi_1 \otimes U = (\Pi_1 \otimes I_2)(cU). \tag{5.8.3}$$

This means we can incorporate a private coin into the protocol postpone the measurement to the end of the protocol.

We can also share information, such as private coins, by creating an EPR pair from our coin. This can be done by applying a CNOT operation to the private quantum coin as shown in Figure 5.11.

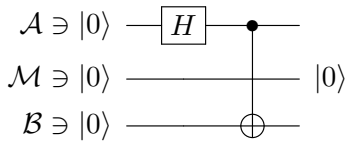


Figure 5.11: The system of Alice creates a private quantum coin and entangles with Bob. The measurement outcomes will be the same. However, this type of communication is not allowed because the spaces $\mathcal{A}$ and $\mathcal{B}$ are not allowed to communicate directly.

However, as stated in the definition of a quantum coin flipping protocol (Definition 5.4.2), it is not allowed to interact with the spaces $\mathcal{A}$ and $\mathcal{B}$ at the same time.

It is possible to share the coin to the message space and let Bob swap the message space $\mathcal{M}$ with an unused qubits in his private space. This is shown in the circuit in Figure 5.12

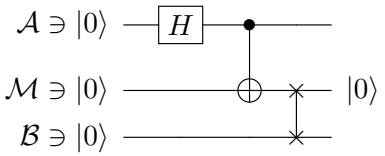


Figure 5.12: This circuit performs the same operation as the circuit in Figure 5.11 but only uses operations on $\mathcal{A} \otimes \mathcal{M}$ and $\mathcal{M} \otimes \mathcal{B}$ and is thus allowed in a coin flipping protocol.

With these basic tools and knowledge quantum computing we can determine protocols represented by quantum circuits.

5.9 Ambainis' protocol: Formulating and Solving the Semidefinite Program

We can now reformulate Protocol 5.4.1 by Ambainis with bias $1/4$, in terms of quantum operations and postponed measurements. This circuit is shown in Figure 5.13.

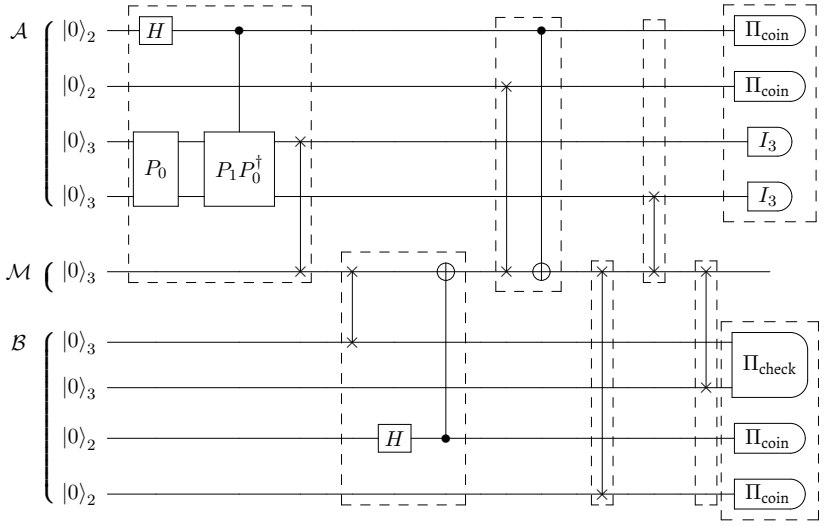


Figure 5.13: This circuit represents Ambainis' protocol (Protocol 5.4.1). It has three rounds and acts on the spaces $\mathcal{A} \cong \mathbf{C}^{36}$, $\mathcal{M} \cong \mathbf{C}^3$ and $\mathcal{B} \cong \mathbf{C}^{36}$. Note that $\mathcal{A}, \mathcal{M}$ and $\mathcal{B}$ consist of a mix of qubits an qutrits. Each box corresponds to a single unitary operation or measurements such as in Definition 5.4.2.

In Figure 5.13, the unitary gate P_i maps the pair of qutrits $|0\rangle_3 |0\rangle_3 \in \mathbf{C}^3 \otimes \mathbf{C}^3$ as

$$P_0 : |0\rangle_3 |0\rangle_3 \mapsto \frac{1}{\sqrt{2}}(|i\rangle_3 |i\rangle_3 + |2\rangle_3 |2\rangle_3), \quad \text{for } i \in \{0, 1\}. \tag{5.9.1}$$

If the private coin of Alice is 0, then $\frac{1}{\sqrt{2}}(|0\rangle_3 |0\rangle_3 + |2\rangle_3 |2\rangle_3)$ is the state we want, if the private coin is 1 however, we want the state $\frac{1}{\sqrt{2}}(|1\rangle_3 |1\rangle_3 + |2\rangle_3 |2\rangle_3)$, so consequently

$$P_1P_0^\dagger : \frac{1}{\sqrt{2}}(|0\rangle_3 |0\rangle_3 + |2\rangle_3 |2\rangle_3) \mapsto \frac{1}{\sqrt{2}}(|1\rangle_3 |1\rangle_3 + |2\rangle_3 |2\rangle_3). \tag{5.9.2}$$

Of course these operators are not uniquely defined on the $\mathbf{C}^3 \otimes \mathbf{C}^3$. One way to construct an operator explicitly that performs this action is by the *Gram-Schmidt orthogonalisation process*.

Lemma 5.9.1 (Gram-Schmidt orthogonalisation) *Let $|\psi_1\rangle, \ldots, |\psi_n\rangle$ be a set of linearly independent quantum states in the Hilbert space $\mathbf{C}^n$. Let γ be the renormalization map, defined by*

$$\begin{aligned} \gamma\colon \mathbf{C}^n \setminus \{0\} &\to \mathbf{C}^n \setminus \{0\} \\ z &\mapsto z/\|z\|. \end{aligned} \tag{5.9.3}$$

and

$$\begin{aligned} |\varphi_1\rangle &= |\psi_1\rangle, \\ |\varphi_2\rangle &= \gamma(|\psi_2\rangle - \langle\psi_2|\varphi_1\rangle |\varphi_1\rangle), \\ |\varphi_3\rangle &= \gamma(|\psi_3\rangle - \langle\psi_3|\varphi_1\rangle |\varphi_1\rangle - \langle\psi_3|\varphi_2\rangle |\varphi_2\rangle), \\ &\vdots \\ |\varphi_n\rangle &= \gamma(|\psi_n\rangle - \langle\psi_n|\varphi_1\rangle |\varphi_1\rangle - \cdots - \langle\psi_n|\varphi_{n-1}\rangle |\varphi_{n-1}\rangle). \end{aligned} \tag{5.9.4}$$

then $|\varphi_1\rangle, \ldots, |\varphi_n\rangle$ is a set of orthonormal vectors in $\mathbf{C}^n$.

Given such an orthonormal set, the operator

$$U = \sum_{i=1}^{n} |\varphi_i\rangle \langle i| \in L(\mathbf{C}^n) \tag{5.9.5}$$

is unitary. An explicit expression for P_0 and P_1 is provided in Appendix A.2.

In Appendix A.1 the matrix representation for a pair of n-dimensional systems is given. Also note that all qubits can be embedded in qutrits, because the space $\mathbf{C}^2$ is a subspace of $\mathbf{C}^3$. This makes it possible to use the message space as a 'qubit' by simply ignoring one dimension.

Measuring the coin is done in the standard basis

$$\Pi_{\text{coin}} = \{\Pi_{C,0}, \Pi_{C,1}\} = \{|0\rangle\langle 0|, |1\rangle\langle 1|\} \subseteq \text{Herm}(\mathbf{C}^2). \tag{5.9.6}$$

To check the two qutrit-state that Bob finally obtains, he measures with the following states:

$$\begin{aligned}\Pi_{\text{check}} &= \{\Pi_{Q,0}, \Pi_{Q,1}, \Pi_{Q,\varnothing}\} \\ &= \{|\psi_0\rangle\langle\psi_0|, |\psi_1\rangle\langle\psi_1|, I_3 \otimes I_3 - \Pi_{Q,0} - \Pi_{Q,1}\} \subseteq \text{Herm}(\mathbf{C}^3 \otimes \mathbf{C}^3).\end{aligned} \tag{5.9.7}$$

If Alice's coin is zero as a result of the protocol, she must have measured the pair $(0,0)$ or $(1,1)$. If her final outcome is 1, she must have measured one of the pairs $(0,1)$ or $(1,0)$. The qutrits are not taken into account and Alice does not have any (formal) possibility to abort. The leads to the set of measurement operators:

$$\begin{aligned}\Pi_A = \{\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\varnothing}\} = \{&(\Pi_{C,0} \otimes \Pi_{C,0} + \Pi_{C,1} \otimes \Pi_{C,1}) \otimes I_3 \otimes I_3, \\ &(\Pi_{C,0} \otimes \Pi_{C,1} + \Pi_{C,1} \otimes \Pi_{C,0}) \otimes I_3 \otimes I_3, \\ &0\} \subseteq \text{Herm}(\mathcal{A}).\end{aligned} \tag{5.9.8}$$

Bob measures the pair of coins as well as checking the two qutrit state. The two qutrit state has to agree with Alice's private coin. Similarly to Alice's final outcome, coins $(0,0)$ and $(1,1)$ will lead to the outcome 0 and $(0,1), (1,0)$ lead to 1. Furthermore, the quantum coin Bob received from Alice has to agree with the check. If the check leads to abort, then Bob will abort the protocol independent of the pair of quantum coins. These outcomes are represented in the following measurement for Bob:

$$\begin{aligned}\Pi_B = \{\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\varnothing}\} = \{&\Pi_{Q,0} \otimes \Pi_{C,0} \otimes \Pi_{C,0} + \Pi_{Q,1} \otimes \Pi_{C,1} \otimes \Pi_{C,1}, \\ &\Pi_{Q,1} \otimes \Pi_{C,0} \otimes \Pi_{C,1} + \Pi_{Q,0} \otimes \Pi_{C,1} \otimes \Pi_{C,0}, \\ &I_{\mathcal{B}} - \Pi_{B,0} - \Pi_{B,1}\} \subseteq \text{Herm}(\mathcal{B}).\end{aligned} \tag{5.9.9}$$

Note that every measurement agrees with the completeness axiom. We have now completely formulated Protocol 5.4.1 in the standard form of Definition 5.4.2 and we can therefore apply Proposition 5.4.4 to find the bias and the optimal cheating strategy.

The second protocol we will write as a quantum circuit is based on a two of EPR-pairs that Alice and Bob shares. We will first state the protocol and then state its bias and determine its representation as a quantum circuit.

Protocol 5.9.1 (EPR-based quantum coin flipping, [53]) *The following protocol is based on two EPR-pairs.*

1. *Alice creates two EPR-pairs, i.e., $|\psi\rangle \in \mathcal{A}_1 \otimes \mathcal{B}_1$ and $|\psi\rangle \in \mathcal{A}_2 \otimes \mathcal{B}_2$, where*

$$|\psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle).$$

   Alice then sends the qubits in the spaces $\mathcal{B}_1$ and $\mathcal{B}_2$ to Bob.

2. *Bob picks a random number i from $\{1,2\}$ and sends it to Alice.*

3. *Alice sends the qubit in the space $\mathcal{A}_i$ to Bob.*

4. *Bob measures the state in the space $\mathcal{A}_i \otimes \mathcal{B}_i$ in*

$$\Pi_{EPR} = \{\Pi_{accept}, \Pi_{abort}\} = \{|\psi\rangle\langle\psi|, I_2 \otimes I_2 - \Pi_{accept}\}.$$

5. *If Bob does not measure abort, both he and Alice measure the remaining qubit in the computational basis $\{|0\rangle, |1\rangle\}$.*

This protocol shares some similarities with the protocol of Ambainis. In both cases Alice is the only person to actually prepare quantum states and Bob is the only person who can detect a cheater in the protocol. Also the strong bias of the protocol is the same.

Lemma 5.9.2 ([53]) *Protocol 5.9.1 has optimal cheating probabilities $P^*_{A,0} = P^*_{A,1} = P^*_{B,0} = P^*_{B,1} = 3/4$ and thus the strong bias of this protocol is $\varepsilon = 1/4$.*

Using the tools from Section 5.8 we can write this protocol as a quantum circuit too. This is shown in Figure 5.14.

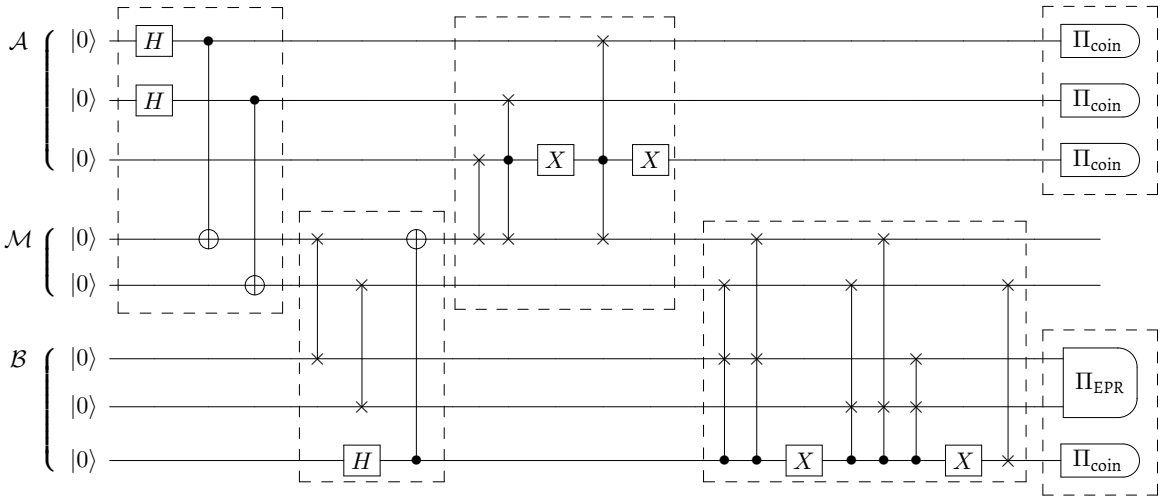


Figure 5.14: This protocol has two rounds and the spaces are $\mathcal{A} \cong \mathbf{C}^8$, $\mathcal{M} \cong \mathbf{C}^4$ and $\mathcal{B} \cong \mathbf{C}^8$. In this protocol Alice prepares two EPRs and Bob shares a random bit. Bob can perform checks.

In this protocol we have the following unitary operations.

$U_{A,1}$: Alice creates two EPR pairs and from each pair, she keeps one qubit herself and shares the other the message space $\mathcal{M}$.

$U_{B,1}$: Bob swaps both qubits that Alice send him to his own private space and shares a private coin with Alice through the message space.

$U_{A,2}$: Depending on Bob's coin, Alice swaps either the remaining qubits from the first or the second EPR pair with the message space.

$U_{B,2}$: Depending on Bob's coin, he has the first or second complete EPR pair.

The measurements in this protocol are described as follows.

Π_A : Alice measures Bob's private coin to determine from which EPR-qubit is left. The outcome of this remaining EPR-qubit is her result of the coin flip.

Π_B : Bob checks whether the complete EPR-pair is in the state $|\psi\rangle$. If this test is positive, he measures the remaining qubit from the EPR-pair and this result is his coin flip.

If both players are honest, they will have the same outcome with 50% each and do not abort the protocol. The measurement of the coin is the same is in Ambainis protocol,

$$\Pi_{\text{coin}} = \{\Pi_{C,0}, \Pi_{C,1}\} = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}. \tag{5.9.10}$$

We can check whether the EPR-qubit is correct by measuring in the operators

$$\Pi_{\text{EPR}} = \{\Pi_{\text{accept}}, \Pi_{\text{abort}}\} = \{|\psi\rangle\langle\psi|, I_2 \otimes I_2 - \Pi_{\text{accept}}\}. \tag{5.9.11}$$

Alice's measurement is determined by the single qubit left from one of the EPR-pairs. This results in the measurement

$$\begin{aligned}\Pi_A = \{\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\varnothing}\} = \{&I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,0} + \Pi_{C,0} \otimes I_2 \otimes \Pi_{C,1},\\ &I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,0} + \Pi_{C,1} \otimes I_2 \otimes \Pi_{C,1},\\ &0\}.\end{aligned} \tag{5.9.12}$$

Bob's measurement consists of a check of the complete EPR-pair he received and the remaining qubit he has left determines the coin

$$\begin{aligned}\Pi_B = \{\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\varnothing}\} = \{&\Pi_{\text{EPR,accept}} \otimes \Pi_{C,0},\\ &\Pi_{\text{EPR,accept}} \otimes \Pi_{C,1},\\ &I_{\mathcal{B}} - \Pi_{B,0} - \Pi_{B,1} = \Pi_{\text{EPR,abort}} \otimes I_2\}.\end{aligned} \tag{5.9.13}$$

This brings the protocol in the unitary form required by Definition 5.4.2.

5.10 Semidefinite Programming Implementation of Ambianis' Protocol

In Section 5.8 we showed how to write a coin flipping protocol as a quantum circuit with postponed measurements. This resulted in a list of unitary operations on the spaces $\mathcal{A} \otimes \mathcal{M}$ and $\mathcal{M} \otimes \mathcal{B}$ and a pair of measurements on the spaces $\mathcal{A}$ and $\mathcal{B}$. To determine the optimal cheating probabilities and strategies, we implemented the semidefinite program from Proposition 5.4.4 using MATLAB 2019b together with the CVX package [54]. The CVX package contains four solvers: SDPT3, SeDuMi, Mosek and Gurobi. The former two packages are free and the latter two are commercial. All solvers are capable of solving semidefinite programs and we used the SeDuMi solver with high precision for these problems. The scripts to solve the semidefinite programs are given in Appendices A.4 and A.5. In this implementation we use a mixed binary-ternary base system to enumerate states, that is for $i_1, i_2 \in \{0,1\}, j_3, j_4, j_5 \in \{0,1,2\}$, we write

$$|i_1\rangle|i_2\rangle|j_3\rangle|j_4\rangle|j_5\rangle = \left|2 \cdot 3^3 i_1 + 3^3 i_2 + 3^2 j_3 + 3 j_4 + j_5\right\rangle \in \mathbf{C}^{108} \tag{5.10.1}$$

This allows us to easily define SWAP and CNOT operations between every pair of qubits and/or qutrits in the spaces $\mathcal{A} \otimes \mathcal{M}$ and $\mathcal{M} \otimes \mathcal{B}$.

The constraints in the program in Proposition 5.4.4

$$\text{Tr}_{\mathcal{M}}(\rho_{A,j+1}) = \text{Tr}_{\mathcal{M}}(U_{A,j}\rho_{A,j}U_{A,j}^\dagger), \quad j = 0, \ldots, N-1, \tag{5.10.2}$$

can be rewritten by adding a new set of variables $\sigma_{A,j+1} \in \text{Herm}(\mathcal{A}\otimes\mathcal{M})$ for all $j \in \{0, \ldots, N-1\}$ and split constraint 5.10.2 into two constraints

$$\sigma_{A,j+1} = U_{A,j}\rho_{A,j}U_{A,j}^\dagger, \quad \text{and} \quad \text{Tr}_{\mathcal{M}}(\rho_{A,j+1}) = \text{Tr}_{\mathcal{M}}(\sigma_{A,j+1}). \tag{5.10.3}$$

Of course these formulations are equivalent, but using this pair of constraints does not require the program to calculate $U_{A,j}\rho_{A,j}U_{A,j}^\dagger$ in every term of the sum of the partial trace.

Solving the four semidefinite programs yields the values shown in Table 5.1.

Table 5.1: Results of solving the semidefinite programs applied to Ambainis' protocol. The status 'S' indicates solved within the specified accuracy and 'I' means inaccurate.

	Primal value	Dual value	Solution gap	Status	Number of iterations	Run time (sec)
$P^*_{A,0}$	0.75002403659	0.75001845344	$5.5832 \cdot 10^{-6}$	I/S	50	597.6
$P^*_{A,1}$	0.75002282927	0.75001820590	$4.6234 \cdot 10^{-6}$	I/S	48	582.3
$P^*_{B,0}$	0.75000071914	0.75000071665	$2.4899 \cdot 10^{-9}$	S	68	750.5
$P^*_{B,1}$	0.75000086773	0.75000073959	$1.2814 \cdot 10^{-7}$	S	68	767.9

Note that formally, all solutions are infeasible, since the optimal (feasible) solution has a primal objective value less than or equal to 0.75. The solutions are very close to the optimal value of $3/4$. We can calculate the strong bias based of this protocol

$$\varepsilon = \max\{P^*_{A,0}, P^*_{A,1}, P^*_{B,0}, P^*_{B,1}\} - \frac{1}{2} = 0.25002403659. \tag{5.10.4}$$

This result differs from the theoretical value by $2.40 \cdot 10^{-5}$. By using even higher accuracy it is possible to decrease this value even more. Whether higher accuracy is required depends mainly on the application.

Using the result optimal cheating strategy of this semidefinite program we can calculate, it is easy to calculate the probability of the complementary outcome or aborting the protocol. Alice does not abort to protocol so it is immediate that this probability is zero and the remaining probability (theoretically 25%) is for measuring the opposite outcome.

If Bob is honest, and we look at the cheating probabilities of Alice we find different results. The operator of measuring 'abort' is not trivial in Bob's case. We use Equation 5.4.20, which in Bob case is $\mathrm{Tr}((I_{\mathcal{M}} \otimes \Pi_{B,\varnothing})\rho_{B,N})$. These results are shown in Table 5.2.

Table 5.2: This table shows the probabilities when Alice cheats optimally according to the SDP. Note that in both cases the probability of aborting the protocol is approximately 23.6% and around 1.4% for the cheater's opposite outcome.

	Probability of outcome '0'	Probability of outcome '1'	Probability of aborting
$P^*_{B,0}$	0.75000	0.013889	0.23611
$P^*_{B,1}$	0.013889	0.75000	0.23611

We see that, contrary to Alice's case, Bob will abort the protocol with a relatively high probability of 23.6%. Depending on Alice's preference, this makes the protocol good or bad.

Solving the optimization programs for the protocol on EPR-pairs 5.9.1, we find exactly the same results, apart from some minor numerical deviations. As stated before both the bias and a lot of the structure of both protocols are similar and hence this result is no surprise.

5.11 Secondary Optimization of Ambainis' Protocol

Using this semidefinite program we confirmed that the optimal cheating strategy leads to a probability of $3/4$ to enforce a chosen outcome. This means that the sum of the probability of measuring the complementary outcome and aborting the protocol is $1/4$.

By applying the semidefinite program of Proposition 5.6.1 to Ambainis' protocol where Bob is honest[2], we have four possible preferences in the outcomes. The optimal solutions are shown in Tables 5.3 and 5.4.

Table 5.3: In this situation, Bob is honest and Alice wants to enforce the outcome '1'. There are two possible secondary preferences. This situation leads to an asymmetry.

Preference	Probability of outcome '0'	Probability of outcome '$\varnothing$'	Status	Number of iterations	Run time (sec)
$1, 0, \varnothing$	0.083723	0.16628	S	73	905.5
$1, \varnothing, 0$	$1.1965 \cdot 10^{-13}$	0.25000	S	33	368.8

Table 5.4: The primary outcome Alice wants to enforce is '0' and this can be done with probability $3/4$. With approximately the same probability as in 5.3 she can optimize the probabilities of the outcomes of the remaining preferences.

Preference	Probability of outcome '1'	Probability of outcome '$\varnothing$'	Status	Number of iterations	Run time (sec)
$0, 1, \varnothing$	0.083722	0.16628	S	74	866.7
$0, \varnothing, 1$	$1.1988 \cdot 10^{-13}$	0.25000	S	33	388.0

First of all, we see that these result yield different probabilities compared to the results in Table 5.2. If aborting the protocol is the second favorable option, then we find a cheating strategy that has the remaining 25% chance of Bob aborting the protocol. On the other hand, if aborting the protocol is Alice's least favorable option, then the optimal cheating strategy results in 16.6% chance of aborting and 8.4% of finding the complementary preference. We find that the secondary bias is around 25%.

Also in the secondary optimization, the protocol based on EPR-pairs has the same results apart from some small numerical differences.

5.12 The Protocol of Berlín et al.: Formulation and Optimization

The protocol of Berlín et al. (Protocol 5.4.2) can also be formulated as a quantum computing circuit from which we can determine the unitary operations and measurements explicitly. In this protocol, both parties start by creating a pair of random coins. Alice will prepare one of the four basis states based on these two coins. She sends this basis state to Bob. One of Bob's private coins is used to measure the qubit he receives (which will be a postponed measurement). The other coin is shared with Alice. Alice stores Bob's shared coin and also shares both of her private coins with Bob. Both parties then measure the states to determine a coin flip. The circuit that represents these two rounds is shown in Figure 5.15.

[2]This analysis is trivial in Alice her case, since she cannot abort the protocol.

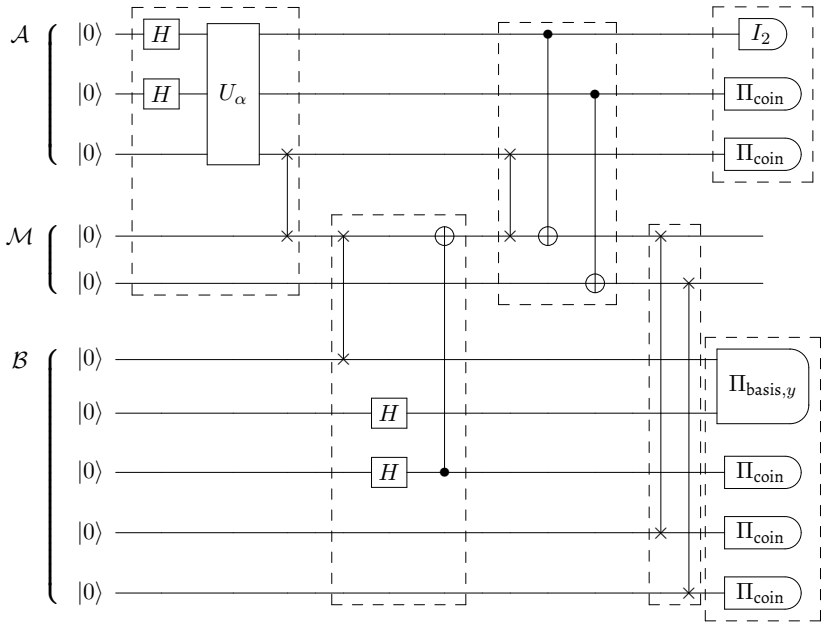


Figure 5.15: A quantum circuit representation of the protocol by Berlín et al. (Protocol 5.4.2) for a parameter $\alpha \in [0, \pi/4]$. This protocol takes two rounds and has spaces $\mathcal{A} \cong \mathbf{C}^8$, $\mathcal{M} \cong \mathbf{C}^4$ and $\mathcal{B} \cong \mathbf{C}^{32}$.

The unitary operation U_α in $U_{A,1}$ rotates a qubit in the state $|0\rangle$ to a basis state depending on the private coins, hence this operator is defined by

$$\begin{aligned} U_\alpha = |0\rangle\langle 0| \otimes |0\rangle\langle 0| \otimes P_\alpha + |0\rangle\langle 0| \otimes |1\rangle\langle 1| \otimes Q_\alpha^\dagger \in L(\mathbf{C}^2 \otimes \mathbf{C}^2 \otimes \mathbf{C}^2) \\ + |1\rangle\langle 1| \otimes |0\rangle\langle 0| \otimes P_\alpha^\dagger + |1\rangle\langle 1| \otimes |1\rangle\langle 1| \otimes Q_\alpha, \end{aligned} \tag{5.12.1}$$

here P_α and Q_α are the rotation operators

$$P_\alpha = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \in L(\mathbf{C}^2), \quad \text{and} \quad Q_\alpha = \begin{pmatrix} \sin\alpha & -\cos\alpha \\ \cos\alpha & \sin\alpha \end{pmatrix} \in L(\mathbf{C}^2). \tag{5.12.2}$$

The measurement in the standard basis for a single qubit is again $\Pi_{\text{coin}} = \{|0\rangle\langle 0|, |1\rangle\langle 1|\}$. The measurement operators are shown in Appendix A.3.

5.13 Results of the Berlín et al. Protocol

In this protocol we take $\alpha = \arctan(1/3)$. For this parameter the protocol is balanced, i.e., Alice and Bob can cheat equally much. The theoretical optimal values for these programs are

$$P^*_{A,0} = P^*_{A,1} = P^*_{B,0} \quad = P^*_{B,1} = 0.9. \tag{5.13.1}$$

Hence the bias of the protocol is 0.4. We solve the programs with in the same way as in Sections 5.10 and 5.11. We begin with calculating all the optimal cheating probabilities and strategies. These results are shown in Table 5.5.

Table 5.5: Results of the numerical solver of the Berlín et al. protocol in the balanced case. All values primal and dual values are close to the theoretical optimal value 0.9.

	Primal value	Dual value	Solution gap	Status	Number of iterations	Run time (sec)
$P^*_{A,0}$	0.90000000096	0.90000000051	$4.5 \cdot 10^{-10}$	I/S	13	6.9
$P^*_{A,1}$	0.90000000096	0.90000000044	$5.2 \cdot 10^{-10}$	I/S	13	5.8
$P^*_{B,0}$	0.90003520571	0.90002762325	$7.5825 \cdot 10^{-6}$	I/S	35	243.8
$P^*_{B,1}$	0.90002655184	0.90002142529	$5.1266 \cdot 10^{-6}$	I/S	35	245.8

Note that the pair of programs to determine $P^*_{A,0}$ and $P^*_{A,1}$ require significantly less iterations and run time compared to the programs $P^*_{B,0}$ and $P^*_{B,1}$. The results are also more accurate. This happens because $\dim(\mathcal{A} \otimes \mathcal{M}) = 2^5 = 32$, whereas $\dim(\mathcal{M} \otimes \mathcal{B}) = 2^7 = 128$. The bias based on these solutions is

$$\varepsilon = \max\{P^*_{A,0}, P^*_{A,1}, P^*_{B,0}, P^*_{B,1}\} - \frac{1}{2} = 0.40003520571, \tag{5.13.2}$$

which has an absolute difference of $3.52 \cdot 10^{-5}$ compared to the theoretical value.

The secondary optimization is also performed for this protocol. Again, we only have to consider four situations of Alice's preference as a cheater, since she herself cannot abort the protocol.

Table 5.6: Results of the secondary optimization of the cheating probabilities when Alice is a cheater and has as a first preference the outcome 1. The probability to abort is in both cases approximately 10% and the probability on the complementary outcome 0.

Preference	Probability of outcome '0'	Probability of outcome '∅'	Status	Number of iterations	Run time (sec)
$1, 0, \varnothing$	$3.2308 \cdot 10^{-5}$	0.099968	I/S	35	283.5
$1, \varnothing, 0$	$6.7074 \cdot 10^{-9}$	0.10000	I/S	19	159.1

Table 5.7: Results of the secondary optimization of the cheating probabilities when Alice is a cheater and has as a first preference the outcome 0. Just as in Table 5.6, the probability to abort is in both cases approximately 10% and the probability on the complementary outcome 0.

Preference	Probability of outcome '1'	Probability of outcome '∅'	Status	Number of iterations	Run time (sec)
$0, 1, \varnothing$	$1.5048 \cdot 10^{-5}$	0.099985	I/S	38	271.6
$0, \varnothing, 1$	$2.1893 \cdot 10^{-9}$	0.10000	I/S	20	141.9

All scenarios have an abort probability of approximately $1/10$ and the probability to find the complementary outcome is approximately 0. Presumably, these numbers are exact. Based on these values, the secondary bias is 0.10000.

If we compare the Ambainis' protocol and the Berlín et al. protocol, we see that bias of Ambainis protocol is better, whereas the secondary bias of the Berlín et al. protocol is lower. Depending on the application, both the bias and secondary bias can be taken into account to decide which protocol fits the best.

5.14 Multiparty Quantum Coin Flipping

So far we considered coin flipping protocol with two parties. We can generalize this by allowing for an arbitrary number of players that want to establish a single coin and none of the players trusts another. This means we have to generalize the theory we developed in the previous sections. For most concepts this follows in a natural way. The upper and lower bounds that have been proven in the two player also can be applied to the multiparty setting. In this section we establish upper and lower bounds for the optimal bias of multiparty quantum coin flipping protocols.

5.15 Upper Bounds on Multiparty Quantum Coin Flipping

Suppose we have $g = 1$ honest players in a group of size k players. We will derive that the optimal bias of a strong quantum coin flipping protocol is $1/2 - \Omega(1/k)$. We will first show a little bit weaker bound $1/2 - \Omega\left(1/k^{1+\delta}\right)$ for any $\delta > 0$. To do this, we give an explicit protocol.

We number players from 1 to k and make consecutive pairs. Player 1 and 2 flip a coin, player 3 and 4 and so on. If the outcome is 0, then the player with the lower id wins, if the outcome is 1, then the player with the higher id wins. Therefore this requires a coin flipping protocols with a good weak bias. If there are an odd number of players, the player who is left over automatically makes it to the next round. All winners then again form pairs and flip coins. This process goes on until there are two players left. They perform a strong coin flipping protocol and the result applies to the entire group. This protocol is schematically shown in Figure 5.17.

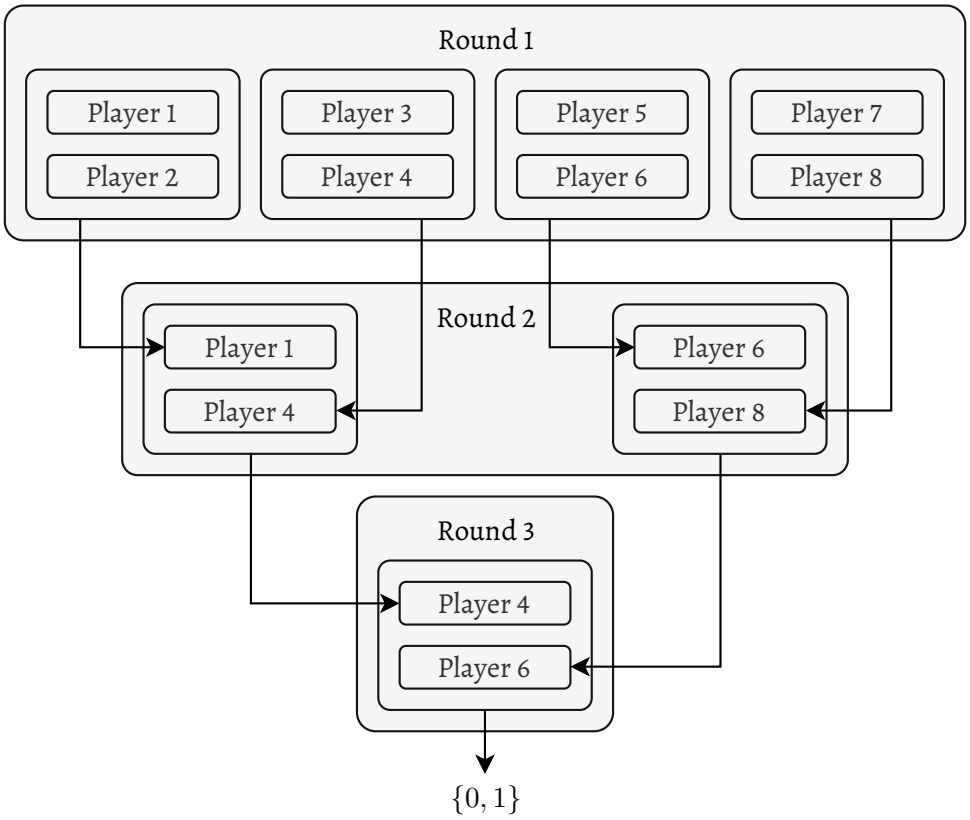


Figure 5.16: The protocol represented here allows to reduce more than two players to two player coin flipping protocols by introducing rounds. All rounds use quantum coin flipping with a good weak bias, except the last round where quantum coin flipping with a good strong bias leads to the outcome that holds for all players.

Note that there are $\lceil \log_2 k \rceil$ elimination rounds. Suppose the weak bias of quantum coin flipping protocol is ε_W and of the strong coin flipping protocol in the last round is ε_S.

Thus the probability that the single good player makes it to the last round is

$$\left(\frac{1}{2} - \varepsilon_W\right)^{\lceil -1 + \log_2 k \rceil}, \tag{5.15.1}$$

If the good player is indeed in the last round, the probability this player will win at least $1/2 - \varepsilon_S$.

Thus the probability the dishonest players will be able to determine the coin is

$$1 - \left(\frac{1}{2} - \varepsilon_S\right)\left(\frac{1}{2} - \varepsilon_W\right)^{\lceil -1+\log_2 k\rceil} \tag{5.15.2}$$

We will use the optimal result of Mochon [18] and lower bound of Kitaev [52], which is equivalent to $\frac{1}{2} - \varepsilon_S \leq 1 - \frac{1}{\sqrt{2}}$. Furthermore

$$\left(\frac{1}{2} - \varepsilon_W\right)^{\lceil -1+\log_2(k)\rceil} \geq \left(\frac{1}{2} - \varepsilon_W\right)^{\log_2(k)} = k^{\log_2(1/2-\varepsilon_W)} = \frac{1}{k^{1+\delta}}, \tag{5.15.3}$$

where

$$\delta = -\left(1 + \log_2\left(\frac{1}{2} - \varepsilon_W\right)\right). \tag{5.15.4}$$

Combining Equation 5.15.3 with Kitaevs lower bound gives the upper bound on the probability of successfully cheating

$$1 - \left(1 - \frac{1}{\sqrt{2}}\right)\frac{1}{k^{1+\delta}}, \quad \text{for any } \delta > 0. \tag{5.15.5}$$

We know that for every $\varepsilon_W > 0$, there exists a protocol with bias $\varepsilon < \varepsilon_W$, and thus δ can be arbitrarily close to 0. This means that an optimal multiparty protocol has a bias

$$\varepsilon \leq 1 - \left(1 - \frac{1}{\sqrt{2}}\right)\frac{1}{k^{1+\delta}} - \frac{1}{2} = \frac{1}{2} - \left(1 - \frac{1}{\sqrt{2}}\right)\frac{1}{k^{1+\delta}} = \frac{1}{2} - \Omega\left(\frac{1}{k^{1+\delta}}\right), \tag{5.15.6}$$

for every $\delta > 0$.

Remark 5.15.1 *The upper bound $1/2 - \Omega\left(1/k^{1+\delta}\right)$ for any $\delta > 0$ does not imply the upper bound $1/2 - \Omega\left(1/k\right)$. One way to show this is by considering the fact that the function $\log(x)$ grows more slowly than any positive power of x: $\log(x) = O(x^{\varepsilon})$ for any $\varepsilon > 0$. This is equivalent to the fact that $\exp(x) = \Omega(x^n)$ for any $n > 0$, which can easily be seen by its Taylor series.*

In the paper by Ambainis et al. [7], they suggest to use the quantum coin flipping protocol Spekkens and Rudolph [55] with weak bias $1/\sqrt{2} - 1/2$ and the strong coin flipping protocol by Ambainis [12], this results the upper bound $1/2 - \Omega\left(1/k^{1.78}\right)$. The best known weak protocol in 2004 was by Mochon [14] with bias 0.192 that implies the upper bound $1/2 - \Omega\left(1/k^{1.7}\right)$.

To obtain the better upper bound $\frac{1}{2} - \Omega\left(\frac{1}{k}\right)$, we have to sophisticate the protocol. This can be done by considering that the main risk the cheaters have is to get caught by the honest player. If any cheater is caught, the protocol stops and everyone is a loser. More specifically, this means that in the earlier rounds, the cheaters have to be more careful. This is when the probability is relatively high that the good player is still present in the game. Later on, when the probability the good player is still in the games has decreased, the cheaters can act more risky to achieve their goal.

This refined protocol can be formalized using quantum coin flipping with penalty. Such a protocol works for two players and has three possible scenarios. If the outcome is 0, Alice wins 1 dollar, if the outcome is 1, Bob wins 1 dollar. However, if a cheater is present and caught, he or she has to pay a specified amount of money. We will refer to this as the penalty. If we want the cheaters to be risk averse, we will make the penalty high. On the other hand, if we want the cheaters to take more risk, the penalty will be lower. Since the probability of the good player being present in game depends on the specific elimination round, the penalty is solely dependent on the number of the round.

We start with $k = 2^n$ players, for some integer $n \geq 1$, of whom only a single player is honest. The protocol will have the same structure as our previous multiparty coin flipping protocol, meaning that we form pairs and every winner advances to the next round to form new pairs. By construction, we will have n rounds that we number by $1, \ldots, n$. In the first $n-3$ rounds, we apply a quantum coin flipping protocol with penalty for cheating. In particular in round $i \in \{1, \ldots, n-3\}$, the penalty is

$$\pi_i = 2^{n-i} - 1. \tag{5.15.7}$$

Let Q_π be the maximum expected win in a two party quantum coin flipping protocol with penalty π. In the last three rounds, $n-2, n-1, n$, we apply coin flipping without penalties. In round $n-2$ there are 8 players left and if we use a protocol with weak bias $1/4$. Therefore the probability of a cheater winning from the honest player in a single round is at most $3/4$. Over these three rounds the maximum probability of cheaters forcing the outcome is $63/64$.

Suppose the honest player won the first $n-j$ round and now advances to round $n-j+1$, where there are $2^j - 1$ dishonest players present. We denote the maximum probability that in this round the dishonest players can force the outcome by P_j. We have the following relation.

Lemma 5.15.1 *For every $j \in \{2, \ldots, n\}$ we have*

$$1 - P_j \geq (1 - P_{j-1})(1 - Q_{\pi_{n-j+1}}). \tag{5.15.8}$$

Proof: The honest player has three possible outcomes: losing, winning or detecting a cheater and we denote the corresponding probabilities with p_w, p_ℓ and p_c respectively. Of course $p_w + p_\ell + p_c = 1$. If the cheaters are able to fix the coin, this is the result of either the honest player losing in the current round, or advancing to the next round and losing in one of the next rounds. This means

$$P_j \leq p_\ell + p_w P_{j-1}, \tag{5.15.9}$$

which we can rewrite to

$$\begin{aligned} P_j &\leq p_\ell + (1 - p_\ell - p_c)P_{j-1} = P_{j-1} + (1 - P_{j-1})p_\ell - P_{j-1}p_c \\ &= P_{j-1} + (1 - P_{j-1)}\left(p_\ell - \frac{P_{j-1}}{1 - P_{j-1}}p_c\right). \end{aligned} \tag{5.15.10}$$

Of course the dishonest players have the possibility to play honest as a strategy, this means $P_{j-1} \geq 1 - 1/2^{j-1}$, so $P_{j-1}/(1 - P_{j-1}) \geq 2^{j-1} - 1$. We can substitute this in the previous inequality to get

$$P_j \leq P_{j-1} + (1 - P_{j-1})(p_\ell - (2^{j-1} - 1)p_c). \tag{5.15.11}$$

Note that the term $p_\ell - (2^{j-1} - 1)p_c$ can be regarded as the expected win, so

$$P_j \leq P_{j-1} + (1 - P_{j-1})Q_{\pi_{n-j+1}}, \tag{5.15.12}$$

which is equivalent to $1 - P_j \geq (1 - P_{j-1})(1 - Q_{\pi_{n-j+1}})$. □

We can now apply Lemma 5.15.1 repeatedly to determine the probability that the dishonest can not force the chosen outcome

$$1 - P_n \geq (1 - P_3) \prod_{j=4}^{n} (1 - Q_{\pi_{n-j+1}}). \tag{5.15.13}$$

In [56], the authors choose a protocol for coin flipping with penalty π, which is a variation of Ambainis' protocol (Protocol 5.4.1). In Lemma 3 of this paper, it is shown that the probability of

Bob winning is at most $1/2 + 1/\sqrt{\pi}$ and in particular the expected win is at most $1/2 + 1/\sqrt{\pi}$. For our choice of penalty, this gives

$$\begin{aligned} 1 - P_n &\geq \frac{1}{64} \prod_{j=3}^{n-1} \left(\frac{1}{2} - \frac{1}{\sqrt{2^j - 1}} \right) = \frac{1}{8 \cdot 2^3} \frac{1}{2^{n-3}} \prod_{j=3}^{n-1} \left(1 - \frac{2}{\sqrt{2^j - 1}} \right) \\ &\geq \frac{1}{8 \cdot 2^n} \prod_{j=3}^{\infty} \left(1 - \frac{2}{\sqrt{2^j - 1}} \right) = \frac{M}{2^n}, \end{aligned} \tag{5.15.14}$$

where

$$M = \frac{1}{8} \prod_{j=3}^{\infty} \left(1 - \frac{2}{\sqrt{2^j - 1}} \right) = 0.0037317\ldots \in (0, 1). \tag{5.15.15}$$

We conclude that $P_n \leq 1 - M/2^n = 1 - M/k$ and thus

$$\varepsilon \leq P_n - \frac{1}{2} \leq \frac{1}{2} - \frac{M}{k}, \tag{5.15.16}$$

so $\varepsilon = 1/2 - \Omega(1/k)$. Asymptotically, this bound is better than the previous bound based on weak coin flipping without penalty. However, the constant of our first approach is better, which means that in smaller groups, this bound will be smaller. For example, if we use the protocol by Rudolph and Spekkens [11] with weak bias $\varepsilon_W = 1/\sqrt{2} - 1/2$, we find that the second bound is lower than the first when the group size is $k \geq 269$. The protocol by Rudolph and Spekkens is not optimal, by using protocol with even better bias this becomes even more apparent.

Ambainis et al. [56] also describe how to extend this result to more than one honest player. Suppose that a group of size k consists of g honest players. We can reduce this number using a protocol that creates a group size k/g of whom one player is honest with a probability of at least $1/2$. The bounds for a single honest player can now be applied and we find a bias $1/2 - \Omega(g/k)$. By taking this selection protocol into account, an upper bound on an optimal multiparty quantum coin flipping protocol is $1/2 - 1/2 \cdot \Omega(g/k) = 1/2 - \Omega(g/k)$. In Table 5.8, we find explicit upper bounds of small groups with variable number of honest parties.

Table 5.8: Explicit upper bounds for the optimal bias of group of size $0 \leq g \leq k \leq 10$. The asterisks $(*)$ indicates impossible scenarios.

k, g	1	2	3	4	5	6	7	8	9	10
10	0.4634	0.4634	0.4268	0.4268	0.3536	0.3536	0.3536	0.3536	0.3536	0
9	0.4634	0.4634	0.4268	0.4268	0.3536	0.3536	0.3536	0.3536	0	$*$
8	0.4268	0.4268	0.4268	0.3536	0.3536	0.3536	0.3536	0	$*$	$*$
7	0.4268	0.4268	0.4268	0.3536	0.3536	0.3536	0	$*$	$*$	$*$
6	0.4268	0.4268	0.3536	0.3536	0.3536	0	$*$	$*$	$*$	$*$
5	0.4268	0.4268	0.3536	0.3536	0	$*$	$*$	$*$	$*$	$*$
4	0.3536	0.3536	0.3536	0	$*$	$*$	$*$	$*$	$*$	$*$
3	0.3536	0.3536	0	$*$	$*$	$*$	$*$	$*$	$*$	$*$
2	0.2072	0	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
1	0	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$

Due to the selection protocols when there are multiple honest players, we have to round to get a sufficiently small reduction. The same reduction leads to the same upper bounds and thus a lot of the same values can be seen in this table.

5.16 Lower Bounds on Multiparty Quantum Coin Flipping

Lower bounds on multiparty quantum coin flipping can directly be extend from Kitaev's proof of the lower bound for two party quantum coin flipping. This requires to redefine a formal definition of a quantum coin protocol in the multiparty setting.

Definition 5.16.1 (Quantum coin flipping protocol, k parties) *A quantum coin flipping protocol with k parties is defined the following collection of structures and rules*

1. *$k+1$ complex Euclidean spaces $\mathcal{A}_1, \ldots, \mathcal{A}_k, \mathcal{M}$ respectively the spaces in which the quantum information of all players $1, \ldots, k$ exist and the message space that is shared between amongst all players. We denote the whole space of the system by*

$$\mathcal{X} = \mathcal{A}_1 \otimes \cdots \otimes \mathcal{A}_k \otimes \mathcal{M}. \tag{5.16.1}$$

2. *A positive integer N, denoting the number of rounds in the protocol.*

3. *The initial state of the system, that is a state of the following tensor form*

$$|\psi_0\rangle = |0\rangle_{\mathcal{A}_1} \otimes \cdots \otimes |0\rangle_{\mathcal{A}_k} \otimes |0\rangle_{\mathcal{M}} \in \mathcal{X}. \tag{5.16.2}$$

4. *A set of unitary operations $\{U_{j,r} : j \in \{1, \ldots, k\}, r \in \{1, \ldots, N\}\}$ on the space $\mathcal{X}$ such that $U_{j,r}$ is the identity on $\mathcal{X}$ except for possibly $\mathcal{A}_j$ and $\mathcal{M}$, i.e., U_j can be written as*

$$U_{j,r} = \sum_{a \in \Sigma} I_{\mathcal{A}_1} \otimes \cdots \otimes I_{\mathcal{A}_{j-1}} \otimes V_{j,r,a} \otimes I_{\mathcal{A}_{j+1}} \otimes \cdots \otimes I_{\mathcal{A}_k} \otimes W_{j,r,a} \in L(\mathcal{X}), \tag{5.16.3}$$

   for some unitary operators $V_{j,r,a}$ on $\mathcal{A}_j$ and $W_{j,r,a}$ on $\mathcal{M}$ and finite index set Σ.

5. *A family of k measurements $\{\Pi_{j,0}, \Pi_{j,1}, \Pi_{j,\varnothing}\}$ on $\mathcal{A}_j$ for each $j \in \{1, \ldots, k\}$. These measurements satisfy*

   (a) *The probabilty of two players measuring different outcomes is zero, i.e., for any $p, q \in \{1, \ldots, k\}$, where $p \neq q$:*

$$\begin{aligned} I_{\mathcal{A}_1} \otimes \cdots \otimes I_{\mathcal{A}_{p-1}} \otimes \Pi_{p,0} \otimes I_{\mathcal{A}_{p+1}} \otimes \cdots \\ \cdots \otimes I_{\mathcal{A}_{q-1}} \otimes \Pi_{q,1} \otimes I_{\mathcal{A}_{q+1}} \otimes \cdots \otimes I_{\mathcal{A}_k} |\psi_N\rangle = 0. \end{aligned} \tag{5.16.4}$$

   (b) *If all players are honest, the outcome has an equal probability for 0 and 1*

$$\langle\psi_N| \Pi_{1,0} \otimes \cdots \otimes \Pi_{k,0} \otimes I_{\mathcal{M}} |\psi_N\rangle = \langle\psi_N| \Pi_{1,1} \otimes \cdots \otimes \Pi_{k,1} \otimes I_{\mathcal{M}} |\psi_N\rangle = \frac{1}{2}. \tag{5.16.5}$$

   where $|\psi_N\rangle = U_{k,N} \cdots U_{1,N} \cdots U_{k,1} \cdots U_{1,1} |\psi_0\rangle \in \mathcal{X}$ is the state of the system after N rounds.

For $k = 2$ Definition 5.16.1 agrees with Definition 5.4.2. A graphical representation is shown in Figure 5.17 for four players that execute a protocol with three rounds.

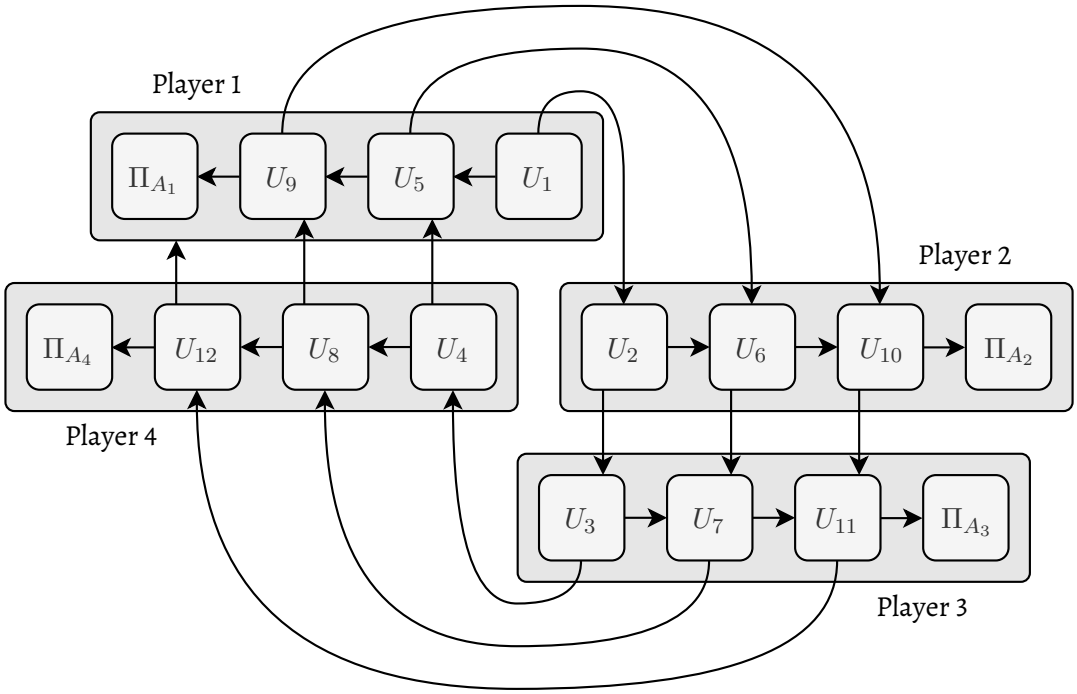


Figure 5.17: A graphical representation of a quantum coin flipping protocol with five players. The message space is carried over to each next player. The final measurements are not shown here.

To find lower bounds on this bias for any k player coin flipping protocol, we apply the same procedure as Kiteav's lower bound for two players. For every possible player $1, \ldots, k$ we have an SDP of the optimal strategy to cheat. The optimal value is the probability a cheater is able to generate a 1 as an outcome, denoted by $P^*_{r,1}$, when player r is honest. Let $|\psi_j\rangle$ be the state of the system after j rounds if every player is honest and $Z_{r,0}, \ldots, Z_{r,N}$ be a dual feasible solution if player r is honest for which $P^*_{r,1} + \delta = \langle 0| Z_{r,0} |0\rangle$, for $\delta > 0$ arbitrary. We then define

$$F_j = \langle \psi_j| Z_{1,j} \otimes \cdots \otimes Z_{k,j} \otimes I_{\mathcal{M}} |\psi_j\rangle \in \mathbf{R}, \quad \text{for } j = 0, \ldots, N. \tag{5.16.6}$$

Similar to the two player case, we can derive the following inequalities by applying strong duality and using dual constraints:

$$\begin{aligned} (P^*_{1,1} + \delta) \cdot \ldots \cdot (P^*_{k,1} + \delta) &= F_0, \\ F_j &\geq F_{j+1}, \quad \text{for } j \in \{0, \ldots, N-1\}, \\ F_N &\geq \frac{1}{2}. \end{aligned} \tag{5.16.7}$$

Thus we conclude $(P^*_{1,1} + \delta) \cdot \ldots \cdot (P^*_{k,1} + \delta) \geq 1/2$. It follows that $\max\{P^*_{1,1}, \ldots, P^*_{k,1}\} \geq (1/2)^{1/k}$. We can use the Taylor series of $(1/2)^{1/k}$ to show that

$$\max\{P^*_{1,1}, \ldots, P^*_{k,1}\} \geq \left(\frac{1}{2}\right)^{1/k} = \sum_{n=0}^{\infty} \frac{(-\log(2))^n}{k^n n!} = 1 - \frac{\log 2}{k} - O\left(\frac{1}{k^2}\right). \tag{5.16.8}$$

Therefore, the bias of any multiparty quantum coin flipping protocol satisfies

$$\varepsilon \geq \left(\frac{1}{2}\right)^{1/k} - \frac{1}{2}. \tag{5.16.9}$$

Which gives us the asymptotic lower bound. $\varepsilon = 1/2 - O(1/k)$. For a single honest player, we see that this lower bound is asymptotically the same as the upper bound and thus the result is

asymptotically tight with optimal bias $\varepsilon = 1/2 - \Theta(1/k)$.

For different values of k Table 5.9 shows the lower bound from equation 5.16.9 and upper bound from Equation 5.15.6.

Table 5.9: This table show the results bounds for quantum coin flipping protocols for a group of k players of whom 1 is honest and the absolute difference between these bounds.

k	3	10	30	100	300	1000	3000	10000
Lower bound	0.2937	0.4330	0.4771	0.4930	0.4976	0.4993	0.4997	0.4999
Upper bound	0.3536	0.4634	0.4817	0.4955	0.4989	0.4995	0.4999	0.5000
Difference (10^{-2})	5.9853	3.0355	0.4534	0.2331	0.1164	0.0121	0.0088	0.0034

Clearly, both bounds converge to $1/2$ as $k \to \infty$. This is due to the fact that a single honest player has less influence on the protocol and thus the probability of a cheat being successful increases.

If there are more honest players g in a group of size k, we can form groups and treat these groups like a single player. This reduces the situation to a single honest player in a group of size k/g. The bias of this protocol is $\varepsilon \geq 1/2 - \Theta(1/(k/g)) = 1/2 - \Theta(g/k)$. In Table 5.10, explicit bounds for groups with fewer than ten parties are shown.

Table 5.10: Explicit lower bounds for the optimal bias of group of size $0 \leq g \leq k \leq 10$. The asterisks ($*$) indicates impossible scenarios.

k, g	1	2	3	4	5	6	7	8	9	10
10	0.4330	0.3706	0.2937	0.2071	0.2071	0	0	0	0	0
9	0.4259	0.3409	0.2071	0.2071	0	0	0	0	0	$*$
8	0.4170	0.3409	0.2071	0.2071	0	0	0	0	$*$	$*$
7	0.4057	0.2937	0.2071	0	0	0	0	$*$	$*$	$*$
6	0.3909	0.2937	0.2071	0	0	0	$*$	$*$	$*$	$*$
5	0.3706	0.2071	0	0	0	$*$	$*$	$*$	$*$	$*$
4	0.3409	0.2071	0	0	$*$	$*$	$*$	$*$	$*$	$*$
3	0.2937	0	0	$*$	$*$	$*$	$*$	$*$	$*$	$*$
2	0.2071	0	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$
1	0	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$	$*$

If we compare Table 5.8 and 5.10, we see that in the situations where g gets relatively big compared to k, the bounds differ a lot. This is due to the fact that we have to round in different ways. For the situations where $g = 1$, the bounds are relatively close.

Chapter 6

Conclusions

Quantum coin flipping is an interesting subject that shows the difference between classical and quantum communication. Quantum coin flipping allows for protocols that are impossible in the classical setting. However, quantum communication does not have unlimited capabilities with respect to quantum coin flipping. The combination of quantum information theory and semidefinite programming leads a semidefinite program that encodes the optimal cheating strategy for a given protocol. By using both the primal and dual semidefinite programs for both potential cheaters, together with strong duality we can elegantly prove Kiteav's lower bound on the strong bias of any quantum coin flipping protocol.

Both classical and quantum coin flipping protocols can be built from bit commitment protocols. There exist bit commitment protocols that are post-quantum safe, and therefore coin flipping protocols based on a post-quantum bit commitment scheme is secure against cheaters.

Balanced coin flipping, i.e., if both players are honest, then both outcomes have equal probability, results in the lowest possible strong bias. This means that there do exist balanced quantum coin flipping protocols with a bias arbitrarily close to $1/\sqrt{2} - 1/2$, and every imbalanced quantum coin flipping protocol has a higher strong bias. From a practical and theoretical viewpoint it is therefor preferable to consider balanced quantum coin flipping.

The semidefinite program that encodes the optimization of the cheating strategy of quantum coin flipping was solved for a number of protocols and confirmed the theoretical values of the bias. Some of the theoretical proofs of the bias rely on sophisticated reasoning, use advanced inequalities in quantum information theory and *good guesses* of the primal and dual feasible solutions. The advantage of using this semidefinite programming approach is that solving the program only requires one to know the standard form description as a list of unitary operations and measurements to determine the optimal cheating probability strategy. This approach makes it possible to find the optimal cheating strategy and corresponding cheating probability and furthermore in a flexible way allows for variations, such as the secondary optimization and expected value optimization by simply changing the objective or adding constraints.

The new bounds on the optimal bias of multiparty quantum coin flipping do not yield the same asymptotical results as the bound presented in Ambainis et al. [56]. However, by using new results on two-player coin flipping, it is possible to establish bounds that perform in explicit cases better for small groups. By using protocols that are not only based on two parties, it is likely that these bounds can be improved.

However, when a protocol gets big, that is, when the number of rounds N increases or the dimensions of the private and message spaces $\mathcal{A}$, $\mathcal{M}$ or $\mathcal{B}$ gets big, the program becomes increasingly

hard to solve. Although semidefinite programs can be solved in polynomial time in the input size, it is the input size itself that grows exponentially as the number of qubits increases linearly. Problems with numerical stability of the algorithms might increase the error in the solutions considerably for protocols with more rounds or larger spaces.

6.1 Recommendations for Future Research

In this thesis we answered some new questions and opened some new problems in the field of quantum coin flipping. Some questions remain open and are theoretically interesting.

We will pose some of these questions that might be interesting for future research:

1. Suppose that we have a multiplayer setting with at least three players, including honest players, dishonest players that prefer the outcome 0, and dishonest players that prefer the outcome 1.

   If all players or groups act independently, what will be the possible outcomes of the protocol?

2. For a fixed number of rounds N and dimensions of the spaces $\mathcal{A}$, $\mathcal{M}$ and $\mathcal{B}$, we can consider a probability distribution on the set of unitary operators and measurements on the spaces $\mathcal{A} \otimes \mathcal{M}$ and $\mathcal{M} \otimes \mathcal{B}$.

   What will be the distribution of the bias when considering a random protocol according to the probability distribution of operators?

3. *How does the iterated optimisation apply to quantum dice throwing? Can this distribution be used to efficiently sample protocols with a good bias?*

   More formally, suppose $(a_1, \ldots, a_n, \varnothing)$ is a preference, i.e., $(a_1, \ldots, a_n)$ is a permutation of $\{1, \ldots, n\}$, of an n-sided dice. Let P^*_{A,a_1} be the maximum probability of measuring a_1, given by the semidefinite program

   $$P^*_{A,a_1} = \sup\{\mathrm{Tr}((\Pi_{A,a_1} \otimes I_{\mathcal{M}})\rho_{A,N}) : (\rho_{A,0}, \ldots, \rho_{A,N}) \in \mathcal{F}_A(\mathcal{P})\}. \tag{6.1.1}$$

   Furthermore, for all $i \in \{2, \ldots, n\}$, let P^*_{A,a_i} be the maximum probability of measuring a_i, given that the probability of measuring a_j is P^*_{A,a_j} for all $j \in \{1, \ldots, i-1\}$, i.e.,

   $$\begin{aligned} P^*_{A,a_i} = \sup\{ &\mathrm{Tr}((\Pi_{A,a_i} \otimes I_{\mathcal{M}})\rho_{A,N}) : (\rho_{A,0}, \ldots, \rho_{A,N}) \in \mathcal{F}_A(\mathcal{P}), \\ &\mathrm{Tr}((\Pi_{A,a_j} \otimes I_{\mathcal{M}})\rho_{A,N}) = P^*_{A,a_j}, j \in \{1, \ldots, i-1\}\}. \end{aligned} \tag{6.1.2}$$

   What can we say about the sequence $P^*_{A,a_1}, \ldots, P^*_{A,a_N}$?

4. The semidefinite program that encodes the cheating strategy leads to a lower bound on the strong bias. The semidefinite program that find the optimal secondary cheating strategy is similar to this first program, apart from its objective and an extra constraint.

   Does there exist a lower bound on secondary optimization, similar to Kitaev's lower bound on the strong bias?

5. Some protocols, such as Ambianinis' protocol, do have the same optimal cheating probability in all situations, but differ in the probablity of aborting the protocol.

   Is it possible to make a given protocol symmetric, i.e., a protocol such that both parties have the same cheating probabilities in all situations?

6. The protocols we considered had a limited number of rounds and the dimensions of the spaces $\mathcal{A}$, $\mathcal{M}$ and $\mathcal{B}$ were relatively small. Optimal protocols such as constructed by Mochon, Chailloux and Kerenidis require increasingly more resources when the bias gets closer to the optimal value.

   Which values for the strong bias of a quantum coin flipping protocol are possible if we add limitations on the dimensions of the private spaces, message spaces, number of rounds or quantum complexity of preparing the cheating strategy?

7. The optimal cheating strategies we found all lead to a non-zero probability of aborting the protocol.

   Does there exist a quantum coin flipping protocol that has a non-trivial cheating strategy for any of the parties that has zero probability of being detected?

Chapter 7

Bibliography

[1] R. W. Meredith, J. E. Janečka, J. Gatesy, O. A. Ryder, C. A. Fisher, E. C. Teeling, A. Goodbla, E. Eizirik, T. L. L. Simão, T. Stadler, D. L. Rabosky, R. L. Honeycutt, J. J. Flynn, C. M. Ingram, C. Steiner, T. L. Williams, T. J. Robinson, A. Burk-Herrick, M. Westerman, N. A. Ayoub, M. S. Springer, and W. J. Murphy, "Impacts of the cretaceous terrestrial revolution and kpg extinction on mammal diversification," *Science*, vol. 334, no. 6055, pp. 521–524, 2011.

[2] A. Shamir, R. L. Rivest, and L. M. Adleman, "Mental poker," in *The mathematical gardner*, pp. 37–43, Springer, 1981.

[3] C. Döscher and M. Keyl, "An introduction to quantum coin tossing," *Fluctuation and Noise Letters*, vol. 2, no. 04, pp. R125–R137, 2002.

[4] J. Von Neumann, O. Morgenstern, and H. W. Kuhn, *Theory of games and economic behavior (commemorative edition)*. Princeton university press, 2007.

[5] M. Blum, "Coin flipping by telephone a protocol for solving impossible problems," *SIGACT News*, vol. 15, pp. 23–27, Jan. 1983.

[6] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, pp. 1484–1509, Oct. 1997.

[7] A. Ambainis, H. Buhrman, Y. Dodis, and H. Rohrig, "Multiparty quantum coin flipping," in *Proceedings of the 19th IEEE Annual Conference on Computational Complexity*, CCC '04, (Washington, DC, USA), pp. 250–259, IEEE Computer Society, 2004.

[8] C. H. Bennett and G. Brassard, "Quantum cryptography: public key distribution and coin tossing.," *Theor. Comput. Sci.*, vol. 560, no. 12, pp. 7–11, 2014.

[9] D. Aharonov, A. Ta-Shma, U. Vazirani, and A. Yao, "Quantum bit escrow," *Proceedings of 32nd Annual ACM Symposium on Theory of Computing*, 2000.

[10] R. W. Spekkens and T. Rudolph, "Optimization of coherent attacks in generalizations of the bb84 quantum bit commitment protocol," *arXiv preprint quant-ph/0107042*, 2001.

[11] R. W. Spekkens and T. Rudolph, "Quantum protocol for cheat-sensitive weak coin flipping," *Physical Review Letters*, vol. 89, no. 22, p. 227901, 2002.

[12] A. Ambainis, "A new protocol and lower bounds for quantum coin flipping," *J. Comput. Syst. Sci.*, vol. 68, pp. 398–416, 2004.

[13] C. Mochon, "Large family of quantum weak coin-flipping protocols," *Physical Review A*, vol. 72, no. 2, p. 022341, 2005.

[14] C. Mochon, "Quantum weak coin-flipping with bias of 0.192," in *45th Annual IEEE Symposium on Foundations of Computer Science*, pp. 2–11, Oct 2004.

[15] G. Berlin, G. Brassard, F. Bussieres, and N. Godbout, "Fair loss-tolerant quantum coin flipping," *Physical Review A*, vol. 80, no. 6, p. 062321, 2009.

[16] G. Berlin, G. Brassard, F. Bussieres, N. Godbout, J. A. Slater, and W. Tittel, "Experimental loss-tolerant quantum coin flipping," *Nature communications*, vol. 2, p. 561, 2011.

[17] A. Chailloux and I. Kerenidis, "Optimal quantum strong coin flipping," in *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 527–533, IEEE, 2009.

[18] C. Mochon, "Quantum weak coin flipping with arbitrarily small bias," *arXiv preprint arXiv:0711.4114*, 2007.

[19] A. S. Arora, J. Roland, and S. Weis, "Quantum weak coin flipping," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pp. 205–216, ACM, 2019.

[20] A. Chailloux, I. Kerenidis, and J. Sikora, "Lower bounds for quantum oblivious transfer," *arXiv preprint arXiv:1007.1875*, 2010.

[21] J. Sikora, "Simple, near-optimal quantum protocols for die-rolling," *Cryptography*, vol. 1, no. 2, p. 11, 2017.

[22] S. Lang, *Linear Algebra*. Springer Undergraduate Texts in Mathematics and Technology, Springer, 1987.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[24] M. Laurent and F. Vallentin, "A course on semidefinite optimisation, lecture notes," February 2018.

[25] M. F. Anjos and J. B. Lasserre, *Handbook on semidefinite, conic and polynomial optimization*, vol. 166. Springer Science & Business Media, 2011.

[26] J. Watrous, *The Theory of Quantum Information*. Cambridge University Press, 2018.

[27] D. V. Pasechnik, "Bipartite sandwiches," *preprint*, 1999.

[28] R. Peeters, "The maximum edge biclique problem is np-complete," *Discrete Applied Mathematics*, vol. 131, no. 3, pp. 651–654, 2003.

[29] R. M. Karp, "Reducibility among combinatorial problems," in *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations, held March 20–22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York, and sponsored by the Office of Naval Research, Mathematics Program, IBM World Trade Corporation, and the IBM Research Mathematical Sciences Department* (R. E. Miller, J. W. Thatcher, and J. D. Bohlinger, eds.), The IBM Research Symposia Series, pp. 85–103, Springer US, 1972.

[30] F. Alizadeh, "Interior point methods in semidefinite programming with applications to combinatorial optimization," *SIAM J. Optim.*, vol. 5, no. 1, pp. 13–51, 1995.

[31] F. G. Brandao and K. M. Svore, "Quantum speed-ups for solving semidefinite programs," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 415–426, IEEE, 2017.

[32] J. Van Apeldoorn, A. Gilyén, S. Gribling, and R. de Wolf, "Quantum sdp-solvers: Better upper and lower bounds," in *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 403–414, IEEE, 2017.

[33] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[34] D. Griffiths, *Introduction to Quantum Mechanics*. Cambridge University Press, 2016.

[35] B. Hensen, H. Bernien, A. E. Dréau, A. Reiserer, N. Kalb, M. S. Blok, J. Ruitenberg, R. F. L. Vermeulen, R. N. Schouten, C. Abellán, W. Amaya, V. Pruneri, M. W. Mitchell, M. Markham, D. J. Twitchen, D. Elkouss, S. Wehner, T. H. Taminiau, and R. Hanson, "Loophole-free bell inequality violation using electron spins separated by 1.3 kilometres," *Nature*, vol. 526, p. 682, 2015.

[36] W. F. Stinespring, "Positive functions on c*-algebras," *Proceedings of the American Mathematical Society*, vol. 6, no. 2, pp. 211–216, 1955.

[37] Y. C. Eldar, A. Megretski, and G. C. Verghese, "Designing optimal quantum detectors via semidefinite programming," *IEEE Transactions on Information Theory*, vol. 49, pp. 1007–1012, April 2003.

[38] R. H. Tütüncü, K.-C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical programming*, vol. 95, no. 2, pp. 189–217, 2003.

[39] D. Bruß, "Optimal eavesdropping in quantum cryptography with six states," *Physical Review Letters*, vol. 81, no. 14, p. 3018, 1998.

[40] G. Brassard, C. Crépeau, D. Mayers, and L. Salvail, "Defeating classical bit commitments with a quantum computer," *arXiv preprint quant-ph/9806031*, 1998.

[41] J. Watrous, "Semidefinite programming in quantum information, lecture notes." `https://cs.uwaterloo.ca/~watrous/CS867.Winter2017/`, December 2017.

[42] S. Wehner, "Tsirelson bounds for generalized clauser-horne-shimony-holt inequalities," *Phys. Rev. A*, vol. 73, p. 022110, Feb 2006.

[43] A. Molina and J. Watrous, "Hedging bets with correlated quantum strategies," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 468, no. 2145, pp. 2614–2629, 2012.

[44] O. Regev and T. Vidick, "Quantum xor games," *ACM Transactions on Computation Theory (ToCT)*, vol. 7, no. 4, p. 15, 2015.

[45] B. W. Reichardt, "Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function," in *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pp. 544–551, Oct 2009.

[46] R. Duan, S. Severini, and A. Winter, "Zero-error communication via quantum channels, noncommutative graphs, and a quantum lovász number," *IEEE Transactions on Information Theory*, vol. 59, pp. 1164–1174, Feb 2013.

[47] M. Laurent and T. Piovesan, "Conic approach to quantum graph parameters using linear optimization over the completely positive semidefinite cone," *SIAM Journal on Optimization*, vol. 25, no. 4, pp. 2461–2493, 2015.

[48] A. Molina, T. Vidick, and J. Watrous, "Optimal counterfeiting attacks and generalizations for wiesner's quantum money," in *Theory of Quantum Computation, Communication, and Cryptography* (K. Iwama, Y. Kawano, and M. Murao, eds.), (Berlin, Heidelberg), pp. 45–64, Springer Berlin Heidelberg, 2013.

[49] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[50] N. A. Alkadri, "Post-quantum commitment schemes," *Master Thesis*, 2015.

[51] I. Damgård and C. Lunemann, "Quantum-secure coin-flipping and applications," in *Advances in Cryptology – ASIACRYPT 2009* (M. Matsui, ed.), (Berlin, Heidelberg), pp. 52–69, Springer Berlin Heidelberg, 2009.

[52] A. Kitaev, "Quantum coin-flipping. presentation at the 6th workshop on," *Quantum Information Processing*, 2002.

[53] J. Sikora, "Semidefinite programming & quantum information (winter 2015) - jamie sikora," 2015.

[54] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1." `http://cvxr.com/cvx`, Dec. 2018.

[55] R. W. Spekkens and T. Rudolph, "Quantum protocol for cheat-sensitive weak coin flipping.," *Physical review letters*, vol. 89 22, p. 227901, 2002.

[56] A. Ambainis, H. Buhrman, Y. Dodis, and H. Rohrig, "Multiparty quantum coin flipping," in *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pp. 250–259, IEEE, 2004.

Part II

Quantum Circuit Design

Chapter 8

Distributed Quantum Computing

"In theory, there is no difference between theory and practice.
But, in practice, there is."

- Attributed to Jan L. A. van de Snepscheut

This second part of this master thesis is the result of a collaboration between J. Mulderij, T. Attema, I. Chiscop and F. Phillipson and me. During my graduation period, Jesse Mulderij was also a gradutation intern at TNO in the department of Cyber Security and Robustness. His master thesis project focused on minimising the number SWAP-gates in a quantum algorithm whilst satisfying the *nearest neighbour constraint*. This constraint allows only for interactions between *neighbouring* qubits.

In my previous internship project at TNO, I investigated the effect of noisy connections in a network of quantum computers that is used to perform an algorithm in a distributed way.

Combining both subjects lead to the a number of interesting questions of which we formalized some into optimisation programs. These kinds of problems were new in the field of quantum computing and we therefore decided to publish the results in a the journal Quantum Information Processing.

Mathematical Formulation of Quantum Circuit Design Problems in Networks of Quantum Computers

R. van Houte J. Mulderij T. Attema I. Chiscop F. Phillipson

Received: date / Accepted: date

Abstract

In quantum circuit design, the question arises how to distribute qubits, used in algorithms, over the various quantum computers, and how to order them within a quantum computer. In order to evaluate these problems, we define the global and local reordering problems for distributed quantum computing. We formalise the mathematical problems and model them as integer linear programming problems, to minimise the number of SWAP gates or the number of interactions between different quantum computers. For global reordering, we analyse the problem for various geometries of networks: completely connected networks, general networks, linear arrays and grid-structured networks. For local reordering, in networks of quantum computers, we also define the mathematical optimisation problem.

Keywords: Nearest neighbour compliant, Quantum computation architectures and implementations, Distributed quantum computing

1 Introduction

The early quantum computers have a (very) limited number of qubits [32]. This is the result of the conditions that are required to store quantum information, and means required to manipulate the information. It is possible to connect multiple quantum computers to form a network and do computations together. This is, analogously to current methods in ICT, called *distributed quantum computing* [4, 7]. In such a system we require the network to be able to share both classical and quantum information. If the network is set up correctly, the collection of quantum computers will behave as one big computer [37], and thus greatly increase the possibilities and practical instances that it can be used for.

To act as one big quantum computer, two quantum computers are connected by an entangled pair of qubits. Depending on the topology of the network, we may have a situation where two computers are not connected directly, but indirectly, via other computers in the network. We can apply a method called *entanglement swapping* [14] to create an entangled pair of qubits between these computers. This procedure requires all consecutive computers along the path to have the shared entangled state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Consider the computers that are not at the endpoints of the path. Were those computers to measure in the Bell basis and then communicate their outcome (this requires two bits of information) to their neighbours along the path, they can perform Pauli gates on their qubits to create a shared entangled pair. This pair would again be in the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. By repeating this process for all computers along the path, we end up with a shared entangled state between two computers at the endpoints. This procedure indicates that it is

preferable to use the shortest path in a network. Computers that perform entanglement swapping will need two extra qubits to store and measure information.

Next, if we want to perform a calculation on a distributed quantum computer, we have to partition the calculations into parts and assign those parts to the individual quantum computers, in such a way that communication between parts of the calculation is possible and can be done efficiently. This means that we have to design a quantum circuit for the total network of quantum computers. Already on a single quantum computer quantum circuit design is not trivial. There are a couple of considerations on how to compile a circuit. The nearest neighbour constraint is one of them. This constraint imposes a restriction on quantum gates, such that gates can only act on two adjacent qubits. Given the locations on which the qubits are present, one might need to change the locations of the qubits before a gate can be applied. Changes to the locations of qubits can be made using so-called SWAP gates [24]. SWAP gates interchange the position of two qubits, but since they are also quantum gates, they can only act on two adjacent qubits. The SWAP gates are considered overhead because they do not directly contribute to the calculation that is being performed. SWAP gates do not only require resources, but also increase the running time significantly. Since coherence times are currently very low, information on qubits can only be held stable for a short amount of time, after which the information is lost due to interaction with the environment [9]. It is therefore important to minimise the running time of the circuit and hence the size of the overhead. In quantum algorithm design, minimising the number of required SWAP gates in order for a circuit to comply with the nearest neighbour constraints has become a research topic of its own. So far though, the focus has been on architectures that involve only a single quantum computer.

There are two main strategies of coping with the minimisation of the number of SWAP gates: global reordering and local reordering [36]. In global reordering, one is only concerned with finding an optimal initial qubit placement without focusing on the micromanagement of swapping the qubits into the right positions after every gate, which is what local reordering entails. Both strategies can be done on a single quantum computer or on a network of quantum computers, leading to four areas of research and applications as indicated in Tab. 1.

Table 1: Four areas of research minimising calculation overhead.

	Global	Local
Single	I	II
Distributed	III	IV

For the single quantum computer (Areas I and II) a variety of research is available. Area I was studied, mostly because of its relative simplicity, in [16, 17, 28, 29, 36]. All kinds of qubit architectures have been considered in the more popular Area II: qubits are placed on a linear array in [3, 5, 13, 15, 18, 20, 25, 27, 31, 35, 36], on a 2D grid in [1, 2, 6, 8, 12, 19, 26, 30], on a 3D grid in [11], or, more recently, on the IBM QX architectures in [10, 33, 38, 39].

Areas III and IV have (as far as the authors are aware of) not been studied before. The contribution of this paper lies in the definition of this research area and the first mathematical formulation of the problems of minimising the number of SWAP gates in the distributed computing areas III and IV.

Area III can be viewed in two ways. If we are interested in the order of all qubits on all quantum computers, we have *Complete distributed global reordering*. This can be seen as global single reordering with two different cost values for the SWAP gates between

qubits on different computers and SWAP gates between qubits on the same computer. Next we have, as we will call it, *Celestial Reordering*. Here one allocates qubits to quantum computers while trying to minimise the number of interactions between computers. This addressed because of the high costs that come with setting up the required entanglement between the computers. The order of the qubits within the computers is not considered. The problem is related to the well known graph partitioning problem as will be shown in Sec. 2.

Our contribution comprises of Integer Linear Programming (ILP) models for the proposed problems. The size of the models is reflected by the number of variables and the number of constraints that they contain. In Table 2, an overview is provided. All the models provide optimal solutions given that the circuit and the gate decomposition are both optimal.

Table 2: The order $\mathcal{O}(\cdot)$ of variables and and constraints of each model is shown. Here, n resembles the number of qubits, m is the number of quantum gates, and M is the number of quantum computers. The grid dimensions, where applicable, are indicated by m_1 and m_2. The dimension of the grid is denoted by p.

ILP model sizes for different problems			
Research area	Network/qubit architecture	Variables	Constraints
Area I	Linear array [29]	n^2	n^2
Area II	Linear array [22]	n^2m	n^2m
	2D grid [21]	n^4m	n^4m
	3D grid [21]	n^4m	n^4m
Area III	Complete	$nM + n^2$	Mn^2
	General	n^2M^2	n^2M^2
	Linear array	n^2	$n^2 + M$
	2D grid	$nM + n^2(m_1 + m_2)$	$n^2(m_1 + m_2)M$
	General grid	$nM + n^2pM^{(p-1)/p}$	$M + n^2pM^{(p-1)/p}$
Area IV	Linear array	$n^2m + nMm$	$n^2m + nMm$

In this paper we define the 'Celestial Reordering' problem (from research area III) and present the mathematical problem formulation for minimising the number of SWAP gates in specific topologies of quantum computer networks. We include ILP models that are suited for exact solution methods. After that, in Sec. 3, the problem of local reordering in the context of distributed quantum computing (Area IV) is formulated and explored. Here we minimise, using a weighted objective function, the number of required SWAP gates within a computer and the number of required SWAP gates between computers. An integer linear programming model is also provided, such that the problem can be solved with exact methods. We end in Sec. 4, with concluding remarks and suggestions for future research.

2 Celestial Reordering of Qubits in a Distributed Quantum Circuit

In this section we will introduce the problem of Celestial reordering. In Celestial reordering, given a quantum circuit consisting of qubits, quantum gates acting on the qubits and a number of quantum computers with given capacities, the task is to assign the qubits to the computers in such a way that the number of gate operations on pairs of qubits on different computers is minimised. We assume that the cost of setting up entanglement between two computers is significantly more costly than applying gates within a

computer. Therefore, we neglect costs related to gates that are applied on qubits that are located on the same computer.

It is of great importance how the quantum computers are connected in a network. In this section we consider the most straightforward geometries: the completely connected network, the general network, the linear array, the two-dimensional grid and the general grid. For each of the networks, we formalise and visualise the problem, and model it as an integer linear program (also ILP).

First we introduce some notation that we will use throughout the paper.

i n denotes the total number of qubits in the quantum algorithm. In diagrams, vertices that represent qubits are denoted by circles.

ii M denotes the number of quantum computers. In diagrams, quantum computers are represented by rounded squares.

iii K is used to denote the effective capacity of each quantum computer. That is, the maximum number of qubits that an individual quantum computer can use and store in working memory. This does not include the qubits that are necessary for communication or entanglement swapping. This adds an extra number of qubits per computer, depending on the network architecture.

Suppose we have a quantum algorithm acting on n qubits, that is represented by a series of unitary gates. We allow for unitary operations on single qubits or controlled gates on two qubits. The unitary operations on single qubits will be ignored. All other operations are assumed to be decomposed into this set of gates [24]. One way to ensure sufficient capacity is to take $K \geq \lceil n/M \rceil$ for every computer. If necessary, this quantity may vary per computer as long as the total capacity exceeds n.

2.1 Completely connected network

We start out in the setting where we have all-to-all coupling between the different quantum computers.

Consider the complete graph K_n, where the vertices are labelled $[n] := \{1, \ldots, n\}$ and each vertex corresponds to a qubit in the algorithms. We can count the number of controlled gates that are applied to each pair of qubits. Similar to the model of single quantum computer global reordering [17], we create a cost function $c : E(K_n) \to \mathbb{Z}_{\geq 0}$ by letting $c_{ij} = c(\{i, j\})$ be the number of controlled gates between qubits i and j. This graph is called the *interaction graph*.

Our goal now is to find an assignment of qubits to computers $f : \{1, \ldots, n\} \to \{1, \ldots, M\}$ such that the total number of controlled gates between all different pairs of computers is minimal.

For a qubit $i \in [n]$ and computer $k \in [M]$ let

$$x_{ik} = \begin{cases} 1 & \text{if qubit } i \text{ is assigned to computer } k \\ 0 & \text{otherwise.} \end{cases} \tag{2.1}$$

For each computer, we thus want to limit the total number of assigned qubits by the computer's total capacity K, so

$$\sum_{i=1}^{n} x_{ik} \leq K, \quad \forall k \in [M]. \tag{2.2}$$

Furthermore, every qubit can be assigned to only one computer, thus

$$\sum_{k=1}^{M} x_{ik} = 1, \quad \forall i \in [n]. \tag{2.3}$$

The objective is

$$\min \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \sum_{k\in[M]} \frac{|x_{ik} - x_{jk}|}{2}, \tag{2.4}$$

since for a given $i, j \in [n]$, $i \neq j$. The second summation in the objective is given by

$$\sum_{k\in[M]} \frac{|x_{ik} - x_{jk}|}{2} = \begin{cases} 1 & \text{if qubit } i \text{ and } j \text{ are assigned to different computers} \\ 0 & \text{otherwise.} \end{cases} \tag{2.5}$$

The 2 in the denominator is to compensate for counting twice that a qubit is on a computer where the other qubit is not. This constant can be taken out of the sums.

We can remove the absolute value in the objective by introducing the variable L_{ijk} and add an extra pair of constraints $-L_{ijk} \leq x_{ik} - x_{jk} \leq L_{ijk}$ for every $i, j \in [n], i < j$ and $k \in [M]$. Since any optimal solution will have integer values for L_{ijk}, this variable does not necessarily have to be formulated as integer. This gives us an MILP (mixed integer linear program) of the form

$$\begin{aligned}
\min\ & \frac{1}{2} \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \sum_{k\in[M]} L_{ijk} \\
\text{s.t.}\ & \sum_{i=1}^{n} x_{ik} \leq K, \quad \forall k \in [M] \\
& \sum_{k=1}^{M} x_{ik} = 1, \quad \forall i \in [n] \\
& \left.\begin{aligned} x_{ik} - x_{jk} &\leq L_{ijk} \\ x_{ik} - x_{jk} &\geq -L_{ijk} \end{aligned}\right\}, \quad \forall i, j \in [n], i < j, \forall k \in [M] \\
& x_{ik} \in \{0, 1\}, \quad \forall i \in [n], k \in [M] \\
& L_{ijk} \in \mathbb{R}, \quad \forall i, j \in [n], i < j, \forall k \in [M].
\end{aligned} \tag{2.6}$$

The total number of integer variables is nM and the total number of continuous variables is $\binom{n}{2}m = n(n-1)m/2$. There are $M + n + n(n-1)M = \mathcal{O}(Mn^2)$ constraints in this problem.

It is possible to extend the celestial reordering model by allowing different capacities of computers. This can easily be done by replacing the capacity constraints by

$$\sum_{i=1}^{n} x_{ik} \leq K_k \quad \forall k \in [M], \tag{2.7}$$

where the capacity K_k is now computer specific.

2.2 General networks of quantum computers

Suppose the network of quantum computers is represented by a connected graph $G = (V, E)$, where quantum computers are represented by nodes. A pair of quantum computers can communicate directly if and only if their corresponding nodes are connected by an edge in the graph. If two quantum computers are not connected directly, we can indirectly connect them via intermediate connections with other computers. We can do this by applying *entanglement swapping*. In this case, we search for the shortest path between the pair of computers.

We let the vertex set $V = [M]$ be labelled by the computers and define $w_{k\ell}$ as the length (i.e. the number of edges) of the shortest path between vertices k and ℓ in G. We can therefore consider the problem on the complete graph K_M with edge weights $w_{k\ell}$ for all $k, \ell \in [M], k \neq \ell$. In Fig. 1, an example is given for clarification.

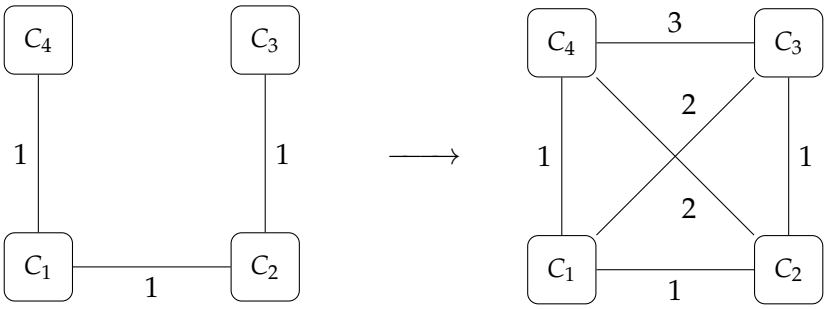


Figure 1: The conversion of a general graph to a complete graph with edge weights corresponding to the distance of the shortest path between each pair of nodes. In this example, the shortest path between C_3 and C_4 in the left graph is 3, therefore, the weight on the edge $\{C_3, C_4\}$ in the right graph is equal to 3.

Here, we see that for a network of four computers, we can construct a complete graph where every computer is connected to every other computer. The weights on the edges now indicate the length of the path from one computer to another. Pairs of qubits which are placed on different computers contribute to the costs if they interact with each other. The cost per interaction is equal to the distance between the computers on which the interacting qubits are located, since that counts the number of times an entangled pair of ancillary qubits is required.

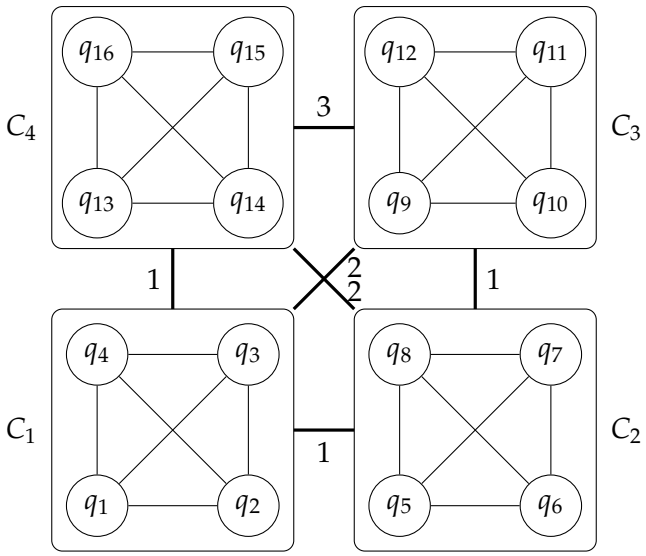


Figure 2: A complete graph on a network of four quantum computers, indicated by the C's. The computers each have a capacity of four qubits.

We consider the same decision variables $x_{ik}, i \in [n], k \in [M]$ as in Sec. 2.1. Our objective will be

$$\min \sum_{\substack{i,j \in [n] \\ i<j}} c_{ij} \sum_{\substack{k,\ell \in [M] \\ k \neq \ell}} w_{k\ell} x_{ik} \cdot x_{j\ell}, \tag{2.8}$$

and since x_{ik} and x_{jl} are both binary, their product is

$$x_{ik} \cdot x_{j\ell} = \begin{cases} 1 & \text{if qubit } i \text{ is on computer } k \text{ and qubit } j \text{ is on computer } \ell \\ 0 & \text{otherwise.} \end{cases} \tag{2.9}$$

The contribution of an assigned pair of qubits depends on two factors: the path length between the computers in the network and the number of interactions in-between the qubits in the algorithms. The product of these quantities is the number of EPR-pairs that is required for this pair of computers.

The constraints are the same as in the original Program 2.6. We thus obtain a quadratic binary optimisation program. This quadratic problem has nM variables and $n + M$ constraints.

To transform the quadratic program into an ILP, we introduce a variable $z_{ijk\ell} \in \{0, 1\}$ for $i, j \in [n], i < j$ and $k, \ell \in [M], k \neq \ell$, that satisfies the inequality

$$z_{ijk\ell} \geq x_{ik} + x_{j\ell} - 1, \quad \forall i, j \in [n], i < j \text{ and } k, \ell \in [M], k \neq \ell. \tag{2.10}$$

If x_{ik}, $x_{j\ell}$ or both are equal to 0, then $z_{ijk\ell} \geq 0$ and since we are minimising over an increasing function this yields $z_{ijk\ell} = 0$. Only if $x_{ik} = x_{j\ell} = 1$, then $z_{ijk\ell} = 1$ is required.

We are left with the equivalent program

$$\begin{aligned}
\min \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \sum_{\substack{k,\ell\in[M]\\ k\neq\ell}} w_{k\ell} z_{ijk\ell} \\
\text{s.t. } z_{ijk\ell} &\geq x_{ik} + x_{j\ell} - 1, \quad \forall i,j \in [n], i<j \text{ and } k,\ell \in [M], k \neq \ell \\
\sum_{k\in[M]} x_{ik} &= 1, \quad \forall i \in [n] \\
\sum_{i\in[n]} x_{ik} &\leq K, \quad \forall k \in [M] \\
x_{ik} &\in \{0,1\}, \quad \forall i \in [n], k \in [M] \\
z_{ijk\ell} &\in \{0,1\}, \quad \forall i,j \in [n], i<j \text{ and } k,\ell \in [M], k \neq \ell.
\end{aligned} \tag{2.11}$$

This is an ILP with $nM + \binom{n}{2}M(M-1) = \mathcal{O}(n^2M^2)$ variables and $\mathcal{O}(n^2M^2)$ constraints. Notice that the number of variables and constraints has increased by turning the quadratic program into an ILP.

An interesting question is how much the objective can vary as the capacity K of each computer changes. We can illustrate this with the example of the graphs K_6 and K_4 that are connected by one edge, see Fig. 3.

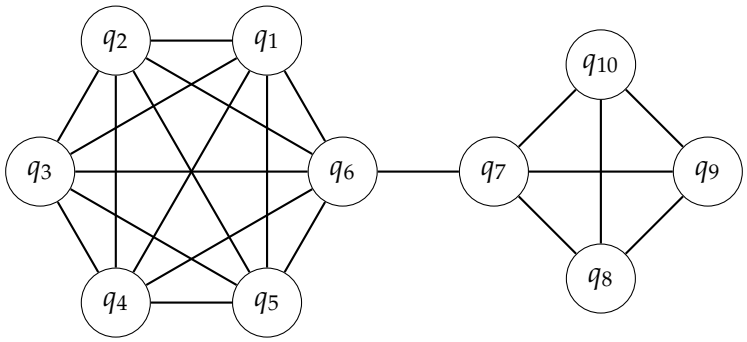


Figure 3: Here we see an interaction graph that consists of two complete graphs that are connected to each other by one edge. The graph contains ten qubits. Each edge represents a single interaction between a pair of qubits for some quantum algorithm on ten qubits.

If we have $M = 2$ computers, both with capacity $K = 5$, we are required to make a cut of at least 5 edges. We partition the interaction graph into $\{1,\ldots,5\}$ and $\{6,\ldots,10\}$ forming two computers.

However, if we were somehow able to increase the capacity of both computers to $K = 6$ qubits, we can partition the graph into $\{1,\ldots,5\}$ and $\{6,\ldots,10\}$. This requires a cut of only one edge. This example and generalisations to more qubits show that the capacity of the computers by a small amount can yield a big difference in the number of EPR pairs required.

2.3 Example of a quantum network and distributed algorithm

We consider an example of a quantum network of four computers between four cities in The Netherlands: Amsterdam (A), Delft (D), Leiden (L) and The Hague (G). The cities of Leiden, Delft and The Hague are all mutually connected while Amsterdam is only

connected to Leiden. The shortest distance between every pair of cities is represented in Tab. 3.

Table 3: The shortest distances between the cities of Amsterdam (A), Delft (D), Leiden (L) and The Hague (G).

w	A	D	L	G
A	-	2	1	2
D	2	-	1	1
L	1	1	-	1
G	2	1	1	-

Each computer has an effective capacity of four qubits to execute the circuit. Each computer also has one extra qubit that is used for communication. This qubit is not assigned to qubits in the circuit and is not taken into account in the optimal assignment. On this network of quantum computers, we want to execute a quantum circuit on fifteen qubits that counts the number of qubits in state $|1\rangle$. The circuit is shown in Fig. 4. It is called the "rd84_143" circuit and was obtained from the reversible circuit library *RevLib* [34].

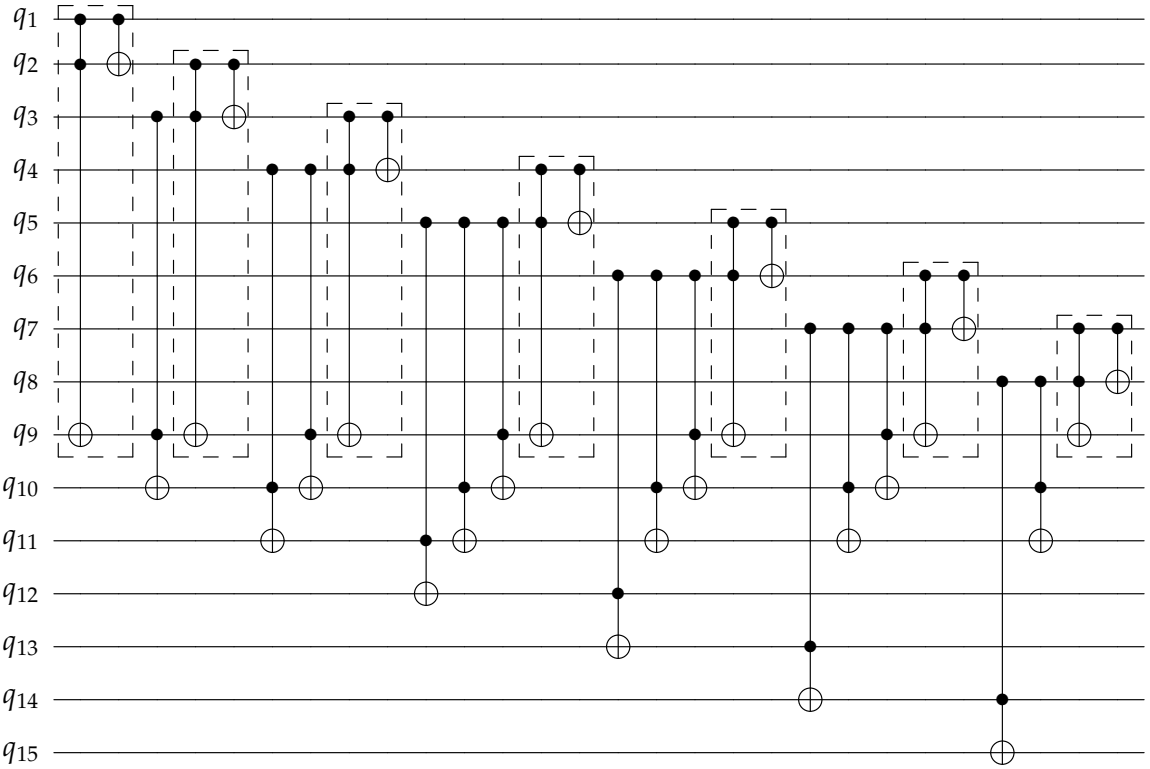


Figure 4: The "rd84_143" circuit. The circuit consists of fifteen qubits. After the Toffoli and Peres gates are decomposed, 98 two-qubit gates remain.

This circuit consists of fifteen CNOT gates and ten Toffoli gates. The Toffoli gate acts on three qubits and can be decomposed in five controlled gates as shown in Fig. 5.

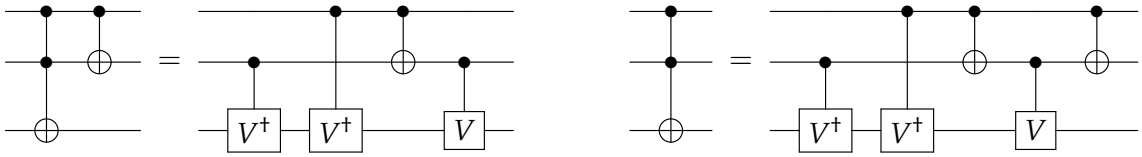


Figure 5: On the left, the decomposition of the Peres gate. On the right, the Sleator-Weinfurter decomoposition of the Toffoli gate [23].

The V-gate is the square root of the Pauli X-gate:

$$V = \sqrt{X} = \frac{1}{2}\begin{pmatrix} 1+i & 1-i \\ 1-i & 1+i \end{pmatrix}. \tag{2.12}$$

From the circuit, the cost c_{ij} is obtained for every pair of qubits, by counting the number of gates act on the qubit pair i, j.

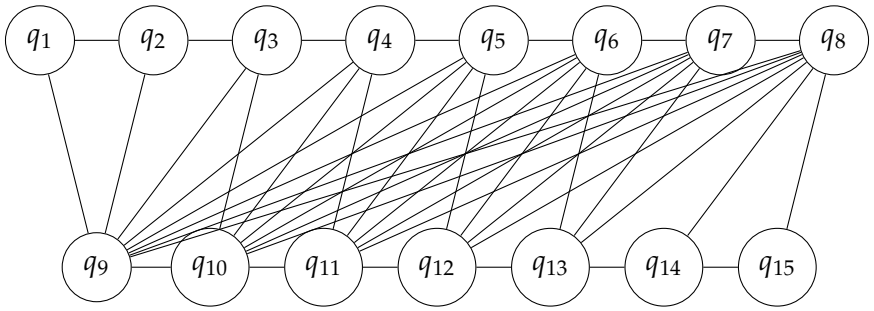


Figure 6: The interaction graph of counting circuit of Fig. 4 on fifteen qubits. An edge represents one or more controlled gates between each pair of qubits. An optimal assignment of qubits to computers in not immediately clear.

The ILP was constructed and solved to optimality using the Python API of CPLEX. The solver was run on a computer with 2 GB of RAM, and completed its Branch & Bound search in 0.39 seconds. The optimal qubit assignment is shown in Fig. 7.

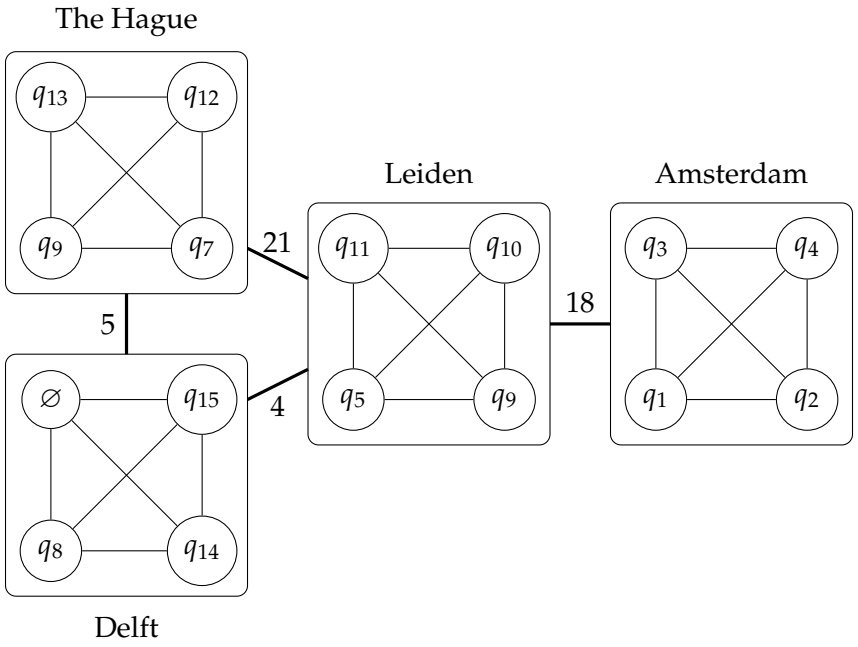


Figure 7: The graph shows the optimal qubit configuration, where each qubit is assigned to a computer. The capacity of the computers is not exceeded. The number of gates between every pair of computers is shown on the edges between computers.

The costs of communication between every two computers are shown on the edges in Fig. 7. The sum of these costs, which is the objective function of the optimisation program, is 48. There was no communication between computers with a distance of two between them.

2.4 Linear array

In this section, we consider a different network of quantum computers. In this network all computers are arranged on a line, and each one of them is connected to its one or two neighbouring computers. This network is a special case of the general network and leads to a reduction in the number of variables and constraints in the resulting model because of the structure in the network.

If we associate the computers with the numbers $\{1, \ldots, M\}$ then quantum computer k can only communicate with computers $k-1$ and $k+1$, except at the boundaries. An example of such a network is given in Fig. 8.

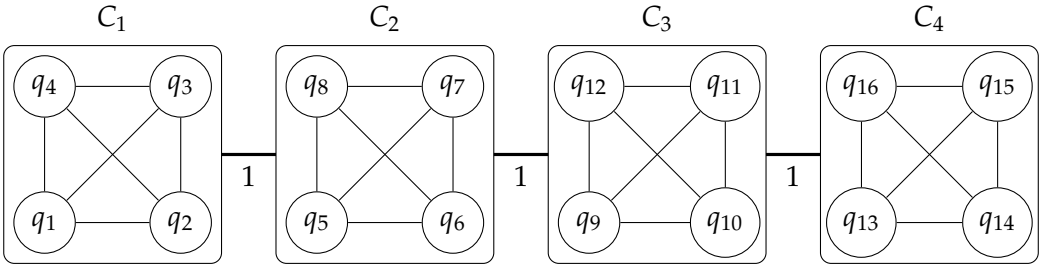


Figure 8: A line graph on a network of four quantum computers, indicated by the C's. The computers each have a capacity of four qubits, corresponding to the circular nodes.

This means that if two qubits are located on computer k and ℓ, then applying a two

qubit gate requires us to make $|k-\ell|$ non-local interactions by using the computers in-between. This changes the original objective in Eq. 2.4 to an objective that takes the distance between computers into account. For a given pair of qubits (i,j) this is given by the equation

$$\left|\sum_{k\in[M]} kx_{ik} - \sum_{k\in[M]} kx_{jk}\right| = \left|\sum_{k\in[M]} k(x_{ik}-x_{jk})\right|. \tag{2.13}$$

Again, we introduce a new variable to encode the absolute value as a linear constraint, analogous to the completely connected network of Sec. 2.1. The full mixed integer linear program now reads

$$\begin{aligned}
\min\ & \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij}L_{ij} \\
\text{s.t.}\ & \sum_{i=1}^{n} x_{ik} \le K, \quad \forall k\in[M] \\
& \sum_{k=1}^{M} x_{ik} = 1, \quad \forall i\in[n] \\
& \left.\begin{aligned} \sum_{k\in[M]} k(x_{ik}-x_{jk}) &\le L_{ij} \\ \sum_{k\in[M]} k(x_{ik}-x_{jk}) &\ge -L_{ij} \end{aligned}\right\}, \quad \forall i,j\in[n], i<j \\
& x_{ik}\in\{0,1\} \quad \forall i\in[n], k\in[M] \\
& L_{ij}\in\mathbb{R} \quad \forall i,j\in[n], i<j.
\end{aligned} \tag{2.14}$$

This MILP consists of nM integer variables and $\binom{n}{2} = \mathcal{O}(n^2)$ continuous variables and has $n+M+2\binom{n}{2} = \mathcal{O}(n^2+M)$ constraints.

2.5 Two-dimensional grid

In this section we consider a two-dimensional grid as network topology. Such a network allows for more connections between computers and directly extends the linear network of Sec. 2.4. Nevertheless, this network also leads to a reduction in the complexity in the assignment of qubits to computers.

We first have to introduce some tools to describe this network. Consider the metric based on the 1-norm[1] defined by

$$d(x,y) = \|x-y\|_1 = \sum_{i=1}^{p} |x_i - y_i|, \quad x,y\in\mathbb{Z}^p. \tag{2.15}$$

We first consider a (square) grid with side length m defined by $G_2 := [m_1]\times[m_2] \subseteq \mathbb{Z}^2$. Thus the number of quantum computers equals $M = m_1 m_2$. We say that two quantum computers are connected if and only if their distance is 1. A small example of such a network is shown in Fig. 9.
For qubit $i\in[n]$ and computer $(u,v)\in G_2$ we let

$$x_{i,uv} = \begin{cases} 1 & \text{if qubit } i \text{ is assigned to position } (u,v) \\ 0 & \text{otherwise.} \end{cases} \tag{2.16}$$

[1]This metric is also called the *taxicab distance* or *Manhattan distance* for its similarity to travelling along the shortest route between two points in the streets of Manhattan, New York.

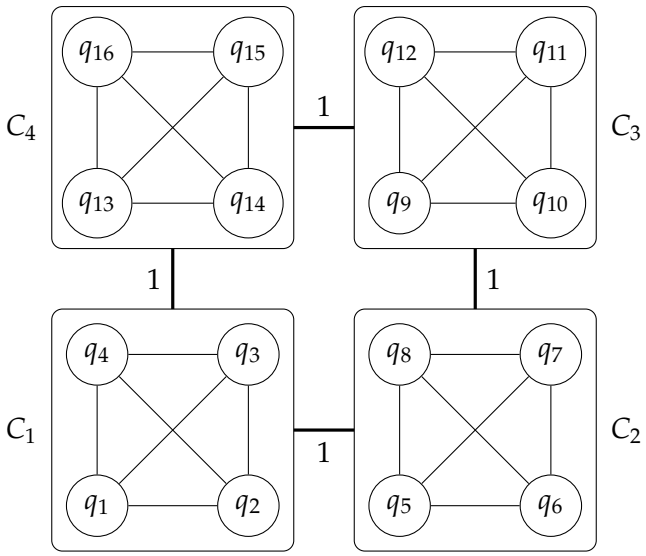


Figure 9: The graph of a two-dimensional grid on a network of four quantum computers, indicated by the C's. The computers each have a capacity of four qubits.

Then, similar to the constraints in Eqs. 2.2 and 2.3, we have the following constraints:

$$\sum_{i=1}^{n} x_{i,uv} \leq K, \quad \forall (u,v) \in G_2, \tag{2.17}$$

and

$$\sum_{(u,v)\in G_2} x_{i,uv} = 1, \quad \forall i \in [n]. \tag{2.18}$$

Furthermore, the objective is now a weighted sum. The weights are determined by the number of interactions between a pair of qubits. The weighted sum consists of terms given by the distance between computers in the network to which the qubits are assigned. The objective is

$$\begin{aligned}\sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \|f(i) - f(j)\|_1 = \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \Bigg(&\sum_{u\in[m_1]} \left| \sum_{v\in[m_2]} v x_{i,uv} - \sum_{v\in[m_2]} v x_{j,uv} \right| \\ &+ \sum_{v\in[m_2]} \left| \sum_{u\in[m_1]} u x_{i,uv} - \sum_{u\in[m_1]} u x_{j,uv} \right| \Bigg).\end{aligned} \tag{2.19}$$

Again, we introduce new variables to encode the absolute values, this is done by two families $L^{(1)}_{ij,u}$ and $L^{(2)}_{ij,v}$. Analogous to the complete linear network described in Sec. 2.4, these variables can be relaxed to real numbers. The mixed integer linear program then

reads

$$
\begin{aligned}
\min \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} &\left(\sum_{u\in[m_1]} L^{(1)}_{ij,u} + \sum_{v\in[m_2]} L^{(2)}_{ij,v} \right) \\
\text{s.t.} \sum_{i=1}^{n} x_{i,uv} &\le K, \quad \forall u \in [m_1], v \in [m_2] \\
\sum_{u\in[m_1]} \sum_{v\in[m_2]} x_{i,uv} &= 1, \quad \forall i \in [n] \\
&\left.\begin{aligned}
&\left.\begin{aligned}
\sum_{v\in[m_2]} v(x_{i,uv} - x_{j,uv}) &\le L^{(1)}_{ij,u} \\
\sum_{v\in[m_2]} v(x_{i,uv} - x_{j,uv}) &\ge -L^{(1)}_{ij,u}
\end{aligned}\right\} \quad \forall u \in [m_1] \\
&\left.\begin{aligned}
\sum_{u\in[m_1]} u(x_{i,uv} - x_{j,uv}) &\le L^{(2)}_{ij,v} \\
\sum_{u\in[m_1]} u(x_{i,uv} - x_{j,uv}) &\ge -L^{(2)}_{ij,v}
\end{aligned}\right\} \quad \forall u \in [m_1]
\end{aligned}\right\} \quad \forall i,j \in [n], i<j \\
x_{i,uv} &\in \{0,1\}, \quad \forall i \in [n], u \in [m_1], v \in [m_2] \\
L^{(1)}_{ij,u}, L^{(2)}_{ij,v} &\in \mathbb{R}, \quad \forall u \in [m_1], v \in [m_2], \forall i,j \in [n], i<j.
\end{aligned}
\tag{2.20}
$$

This MILP has $nm_1m_2 = nM$ integer variables and $\binom{n}{2}(m_1+m_2) = \mathcal{O}(n^2(m_1+m_2))$ continuous variables. The program contains $m_1m_2 + n + 2\binom{n}{2}(m_1+m_2) = \mathcal{O}(n^2(m_1+m_2)+M)$ constraints. If the grid sizes are similar up to a constant, then $m_1 = \Theta(m_2)$. Furthermore, if the capacity of each computer is fixed and the least number of computers is used, then $n = \mathcal{O}(M)$, then the number of constraints is $\mathcal{O}(M^{2.5})$.

2.6 General grid

The two-dimensional network of Sec. 2.5 was a generalisation of the linear network of Sec. 2.4. Since the 1-norm allows for generalisation to any finite dimensional lattice, this section describes the most general case for grids.

We assume the dimensions of the p-dimensional grid are the same to provide a clearer description. However, similar to the two-dimensional grid, it is possible to use grids of different spatial proportions. Let $G_p = \underbrace{[m]\times\cdots\times[m]}_{p\text{ times}}$, and for $u = (u_1,\ldots,u_{p-1}) \in G_{p-1}, r \in [d], v \in [m]$ define

$$
u \oplus_r v = (u_1,\ldots,u_{r-1},v,u_r,\ldots,u_{p-1}) \in G_p, \tag{2.21}
$$

that is, in the integer string u we insert the number v at place r.

The general objective now becomes

$$
\sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \left(\sum_{r\in[p]} \left(\sum_{u\in G_{p-1}} \left| \sum_{v\in[m]} v(x_{i,u\oplus_r v} - x_{j,u\oplus_r v}) \right| \right) \right). \tag{2.22}
$$

Here, we can again introduce a family of variables $L_{ij,u}^{(r)}$ for all $i,j \in [n], i < j$, $u \in G_{d-1}$ and $r \in [m]$ to linearise the absolute value. This gives us the MILP

$$
\begin{aligned}
\min \quad & \sum_{\substack{i,j\in[n]\\ i<j}} c_{ij} \left(\sum_{r\in[p]} \left(\sum_{u\in G_{p-1}} L_{ij,u}^{(r)} \right) \right) \\
\text{s.t.} \quad & \sum_{i\in[n]} x_{i,\omega} \leq K, \quad \forall \omega \in G_p \\
& \sum_{\omega\in G_p} x_{i,\omega} = 1, \quad \forall i \in [n] \\
& \left.\begin{aligned} \sum_{v\in[m]} u(x_{i,u\oplus_r v} - x_{j,u\oplus_r v}) \leq L_{ij,u}^{(r)} \\ \sum_{v\in[m]} u(x_{i,u\oplus_r v} - x_{j,u\oplus_r v}) \geq -L_{ij,u}^{(r)} \end{aligned}\right\} \quad \forall u \in G_{p-1}, r \in [p], i,j \in [n], i < j \\
& x_{i,\omega} \in \{0,1\}, \quad \forall i \in [n], \omega \in G_p \\
& L_{ij,u}^{(r)} \in \mathbb{R}, \quad \forall u \in G_{p-1}, r \in [p], i,j \in [n], i < j.
\end{aligned}
\tag{2.23}
$$

The number of quantum computers in this network is $M = m^p$. This program contains nM integer variables and $\binom{n}{2} p m^{p-1} = \mathcal{O}(n^2 p M^{(p-1)/p})$ continuous variables. Furthermore, there are $m^p + n + 2\binom{n}{2} p m^{p-1} = \mathcal{O}(M + n^2 p M^{(p-1)/p})$ constraints. For $p = 2$ we indeed get the result of the previous section.

3 Local Reordering of Qubits in a Distributed Quantum Circuit

Now we switch our focus to the problem of local reordering in the distributed case. In local reordering, SWAP gates can be inserted before every quantum gate, such that the quantum gate acts on adjacent qubits. The goal is to find the minimal number of required SWAP gates. SWAP gates are considered overhead, costing precious calculation time and resources.

Suppose we have a quantum circuit, consisting of m unitary 2-qubit gates $g_{il} \in G$, acting on a total of n qubits $\{q_1, \ldots, q_n\} \equiv Q$. The physical locations of the qubits are distributed between N quantum computers in a linear fashion, i.e., locations $L_1 = (1, \ldots, k_1)$ belong to computer C_1 and locations $L_2 = (k_1 + 2, \ldots, k_1 + k_2 + 1)$ belong to computer C_2, locations $L_N = (k_1 + \ldots + k_{N-1} + N, \ldots, k_1 + \ldots + k_N + N - 1)$ to computer C_N, where k_i is the qubit capacity of computer C_i. Here one qubit location is skipped between every two consecutive computers, we will see this helps with the modelling later on. Also suppose we have to comply with nearest neighbour interaction constraints, where gates can only act on two qubits if the corresponding qubits are physically adjacent, so their locations are l_i, l_{i+1} respectively for some i.

The local reordering problem concerns the micromanagement of the qubit location at the gate level. Before each gate, the qubit order must be adjusted such that the nearest neighbour constraints are satisfied. However, there are costs involved with the reorganisation of the qubit order. SWAP gates are used to interchange the location of two qubits, but they also have to comply with the nearest neighbour constraints and can thus only interchange locations of two adjacent qubits.

Furthermore, interactions between computers are limited to the action of SWAP gates, where two qubits are exchanged between two quantum computers. The SWAP gates

between computers will likely be more expensive than the ones within a computer since entanglement between the computers in needed for this purpose, this assumption is however not required for the model to provide valid results.

The goal consists of two parts:

1 Minimise the number of SWAP gates between different computers, associated with a cost of α

2 Minimise the number of SWAP gates within each computer, associated with a cost of β

An illustration is provided for clarification in Fig. 10.

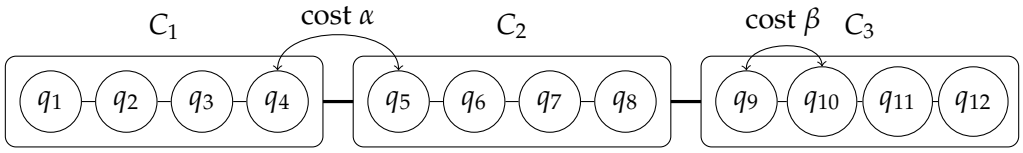


Figure 10: A line graph on a network of three quantum computers, indicated by the C's. The computers each have a capacity of four qubits which are also connected

In order to extend the ILP formulation of minimising the number of SWAP gates in the case of one computer [22] such that it also encapsulates the distributed variant of the local reordering problem, no big extension is required. The proposed mathematical model is presented below.

Let us first introduce variables x_i^t, indicating the location $l \in \cup_{i\in[N]} L_i$ of a qubit q_i just before gate g_t is applied. Note that this also specifies the quantum computer on which the qubit is located. To count the required number of SWAP gates when changing the qubit order between gates, variables $y_{il,t}$ are introduced which keep track of the pairwise ordering of two qubits q_i, q_l before gate g_t.

$$y_{il,t} = \begin{cases} 1 & \text{if } x_{i,t} > x_{l,t} \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Changes in the y's, when moving from one gate to the next, mean that two qubits have changed pairwise order. These pairwise changes in order are also known as inversions. Inversions exactly count the number of SWAP gates that are required to change one qubit order to the next[2].

The nearest neighbour constraints state that a gate can only act on two adjacent qubits, $|x_{i,t} - x_{l,t}| \leq 1$. This constraint is linearised using the inequalities

$$x_{i,t} - x_{l,t} \leq 1, \qquad \forall g_{il,t} \in G, \tag{3.2}$$
$$x_{i,t} - x_{l,t} \geq -1, \qquad \forall g_{il,t} \in G. \tag{3.3}$$

To keep track of the qubit order and to make sure the distance between qubits is at least 1, the following big-M type constraints are added.

[2]The number of required SWAP gates to go from one qubit order to the next is actually a metric on the corresponding elements of the symmetric group $\mathcal{S}_n$, called the Kendall tau metric.

$$x_{i,t} - x_{l,t} \leq My_{il,t} - 1, \qquad \forall i, l \in Q, i < l, t \in [m], \tag{3.4}$$
$$x_{l,t} - x_{i,t} \leq M(1 - y_{il,t}) - 1, \qquad \forall i, l \in Q, i < l, t \in [m], \tag{3.5}$$

where the constant M is chosen to be large enough, in this case $M = N + \sum_i k_i$ will suffice, to make one constraint trivially satisfied. The binary variable y determines for which of the two constraints that holds.

Recall the auxiliary locations left between the quantum computers which were supposed to help us. These borders b are locations indexed by the values

$$b_s = s + \sum_{j \in [s]} k_j, \qquad s \in [N-1], \tag{3.6}$$

where b_s is the location of the border between quantum computers s and $s+1$. Next, to keep track of qubits changing computer, variables $y_{is,t}$ are introduced.

$$y_{is,t} = \begin{cases} 1 & \text{if } x_{i,t} > b_s \\ 0 & \text{otherwise} \end{cases}, \tag{3.7}$$

They tell us on which side of the auxiliary location between two computers a qubit is located before gate g_t. If the qubit changes order with the auxiliary location, we can add a cost to the objective later on. The $y_{i,t}$ are binary and constrained in the following way:

$$x_{i,t} - b_s \leq My_{is,t} - 1, \qquad \forall i \in Q, i < l, t \in [m], s \in [N-1], \tag{3.8}$$
$$b_s - x_{i,t} \leq M(1 - y_{is,t}) - 1, \qquad \forall i \in Q, i < l, t \in [m], s \in [N-1]. \tag{3.9}$$

Here, the variable $y_{i,t}$ is 0 if the location $x_{i,t}$ of qubit q_i is smaller than the location of the border between computers s and $s+1$.
The absolute change (from gate to gate) in the y variables adds a cost to the objective function. To linearise the absolute values, variables p and r are introduced. The p variables are used for SWAP gates within a computer:

$$p_{il,t} = \begin{cases} 1 & \text{if the order of qubits } i \text{ and } l \text{ changed from gate } g_t \text{ to } g_{t+1} \\ 0 & \text{otherwise.} \end{cases}, \tag{3.10}$$

The r variables keep track of SWAP gates between different quantum computers:

$$r_{is,t} = \begin{cases} 1 & \text{if qubit } i \text{ crossed border } b_s \text{ between gates } g_t \text{ and } g_{t+1} \\ 0 & \text{otherwise.} \end{cases}, \tag{3.11}$$

The p's are constrained as

$$y_{il,t} - y_{il,t+1} \leq p_{il,t}, \qquad \forall i, l \in Q, i < l, t \in [m-1], \tag{3.12}$$
$$y_{il,t} - y_{il,t+1} \geq -p_{il,t}, \qquad \forall i, l \in Q, i < l, t \in [m-1], \tag{3.13}$$

and the r's are constrained as

$$y_{is,t} - y_{is,t+1} \leq r_{is,t}, \qquad \forall i \in Q, t \in [m-1], s \in [N-1], \tag{3.14}$$
$$y_{is,t} - y_{is,t+1} \geq -r_{is,t}, \qquad \forall i \in Q, t \in [m-1], s \in [N-1]. \tag{3.15}$$

Next, we formulate the objective function. The objective is of course to minimise the variables p and r, as they count the changes in qubit order and the qubits swapping to

another computer, respectively. Note that every time two qubits on different computers are swapped, both the corresponding r- and p-variables become 1. Swapping two qubits on different quantum computers should only cost β and not $\alpha + \beta$, the objective function is therefore

$$\min \sum_{t\in[m-1]} \left(\left(\frac{\alpha-\beta}{2} \sum_{i\in Q, s\in[N-1]} r_{is,t} \right) + \left(\beta \sum_{i,l\in Q, i<l} p_{il,t} \right) \right), \tag{3.16}$$

where the $(\alpha - \beta)$ term counteracts the extra counting of the SWAP gate with cost α and the factor of one half prevents us from counting the SWAP over the border between computers twice (once for both qubits).

The complete integer linear program then reads

$$\begin{aligned}
\min \sum_{t\in[m-1]} & \left(\left(\frac{\alpha-\beta}{2} \sum_{i\in Q, s\in[N-1]} r_{is,t} \right) + \left(\beta \sum_{i,l\in Q, i<l} p_{il,t} \right) \right) \\
\text{s.t. } & x_{i,t} - x_{l,t} \leq 1, \qquad \forall g_{il,t} \in G \\
& x_{i,t} - x_{l,t} \geq -1, \qquad \forall g_{il,t} \in G. \\
& x_{i,t} - x_{l,t} \leq M y_{il,t} - 1, \qquad \forall i,l \in Q, i < l, t \in [m] \\
& x_{l,t} - x_{i,t} \leq M(1 - y_{il,t}) - 1, \qquad \forall i,l \in Q, i < l, t \in [m] \\
& x_{i,t} - b_s \leq M y_{is,t} - 1, \qquad \forall i \in Q, t \in [m], s \in [N-1] \\
& b_s - x_{i,t} \leq M(1 - y_{is,t}) - 1, \qquad \forall i \in Q, t \in [m], s \in [N-1] \\
& y_{il,t} - y_{il,t+1} \leq p_{il,t}, \qquad \forall i,l \in Q, i < l, t \in [m-1] \\
& y_{il,t} - y_{il,t+1} \geq -p_{il,t}, \qquad \forall i,l \in Q, i < l, t \in [m-1] \\
& y_{is,t} - y_{is,t+1} \leq r_{is,t}, \qquad \forall i \in Q, t \in [m-1], s \in [N-1] \\
& y_{is,t} - y_{is,t+1} \geq -r_{is,t}, \qquad \forall i \in Q, t \in [m-1], s \in [N-1] \\
& x_{i,t} \in \cup_{i\in[N]} L_i, \qquad \forall i \in Q, t \in [m] \\
& y_{il,t} \in \{0,1\}, \qquad \forall i,l \in Q, i < l, t \in [m] \\
& y_{is,t} \in \{0,1\}, \qquad \forall i \in Q, t \in [m], s \in [N-1] \\
& r_{is,t} \in \{0,1\}, \qquad \forall i \in Q, t \in [m-1], s \in [N-1] \\
& p_{il,t} \in \{0,1\}, \qquad \forall i,l \in Q, i < l, t \in [m-1].
\end{aligned} \tag{3.17}$$

The size of the ILP model scales as a polynomial in the number of qubits, quantum gates and quantum computers in the instance. The number of variables and the number of constraints are both of the order $\mathcal{O}(n^2 m + nMm) = \mathcal{O}(n^2 m)$.

4 Concluding remarks and future research

In quantum circuit design, the step to distributed quantum networks gives rise to an extended area of research. How to distribute qubits over the various quantum computers, and how to order qubits within a quantum computer, are naturally arising problems on the interface of distributed quantum computing and nearest neighbour compliant quantum circuit design. These problems have not been discussed in literature before and are formally introduced in this paper. In order to evaluate these problems, we define the global and local reordering problems for distributed quantum computing. We formalise the mathematical problems and model them as integer linear programming problems, to minimise the number of SWAP gates or the number of interactions between different quantum computers. For global reordering, the problem we identify and analyse is called celestial reordering. In celestial reordering, only the initial distribution of qubits

between the quantum computers is optimised. We analyse the problem for various geometries of networks: completely connected networks, general networks, linear arrays and grid-structured networks. We provide an ILP model for each geometry. For local reordering, in networks of quantum computers, we also define the mathematical optimisation problem and we provide an ILP model. However, as these are NP-hard problems, the size of the instances that can be analysed, will be restricted by calculation times. Evaluation of existing or proposed quantum networks will lead to insights in capabilities of networks and algorithms. For development of large scale networks, these optimisation methods will be essential for efficient use. Further research on heuristic approaches for solving these integer linear programs is recommended by the authors.

Conflict of interest

The authors declare that they have no conflict of interest.

References

[1] AlFailakawi, M.G., Ahmad, I., Hamdan, S.: Harmony-search algorithm for 2d nearest neighbor quantum circuits realization. Expert Syst. with Appl. **61**, 16–27 (2016)

[2] Bhattacharjee, A., Bandyopadhyay, C., Wille, R., Drechsler, R., Rahaman, H.: A Novel Approach for Nearest Neighbor Realization of 2d Quantum Circuits. In: 2018 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 305–310. IEEE, Hong Kong (2018)

[3] Bhattacharjee, A., Bandyopadhyay, C., Wille, R., Drechsler, R., Rahaman, H.: Improved Look-Ahead Approaches for Nearest Neighbor Synthesis of 1d Quantum Circuits. In: 2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID), pp. 203–208. IEEE, Delhi, NCR, India (2019)

[4] Buhrman, H., Röhrig, H.: Distributed quantum computing. In: B. Rovan, P. Vojtáš (eds.) Mathematical Foundations of Computer Science 2003, pp. 1–20. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)

[5] Cheng, X., Guan, Z., Ding, W.: Mapping from multiple-control Toffoli circuits to linear nearest neighbor quantum circuits. Quantum Inf. Process. **17**(7), 169 (2018)

[6] Choi, B.S., Van Meter, R.: An $\Theta(\sqrt{n})$-depth Quantum Adder on a 2d NTC Quantum Computer Architecture. J. Emerg. Technol. Comput. Syst. **8**(3), 1–22 (2012)

[7] Denchev, V.S., Pandurangan, G.: Distributed quantum computing: A new frontier in distributed systems or science fiction? SIGACT News **39**(3), 77–95 (2008)

[8] Ding, J., Yamashita, S.: Exact Synthesis of Nearest Neighbor Compliant Quantum Circuits in 2d architecture and its Application to Large-scale Circuits. IEEE Trans. on Comput.-Aided Des. of Integr. Circuits and Syst. pp. 1–1 (2019)

[9] DiVincenzo, D.P., IBM: The Physical Implementation of Quantum Computation. Fortschr. der Phys. **48**(9-11), 771–783 (2000)

[10] Dueck, G.W., Pathak, A., Rahman, M.M., Shukla, A., Banerjee, A.: Optimization of Circuits for IBM's five-qubit Quantum Computers. In: 2018 21st Euromicro Conference on Digital System Design (DSD), pp. 680–684 (2018)

[11] Farghadan, A., Mohammadzadeh, N.: Mapping quantum circuits on 3d nearest-neighbor architectures. Quantum Sci. Technol. **4**(3), 035001 (2019)

[12] Hattori, W., Yamashita, S.: Quantum Circuit Optimization by Changing the Gate Order for 2d Nearest Neighbor Architectures. In: J. Kari, I. Ulidowski (eds.) Reversible Computation, Lecture Notes in Computer Science, pp. 228–243. Springer International Publishing (2018)

[13] Hirata, Y., Nakanishi, M., Yamashita, S., Nakashima, Y.: An efficient conversion of quantum circuits to a linear nearest neighbor architecture. Quantum Information & Computation **11**(1&2), 142–166 (2011)

[14] Kok, P., Braunstein, S.L.: Entanglement swapping as event-ready entanglement preparation. Fortschr. der Phys.: Prog. of Phys. **48**(5-7), 553–557 (2000)

[15] Kole, A., Datta, K., Sengupta, I.: A Heuristic for Linear Nearest Neighbor Realization of Quantum Circuits by SWAP Gate Insertion Using$N$-Gate Lookahead. IEEE J. on Emerg. and Sel. Top. in Circuits and Syst. **6**(1), 62–72 (2016)

[16] Kole, A., Datta, K., Sengupta, I.: A New Heuristic for N -Dimensional Nearest Neighbor Realization of a Quantum Circuit. IEEE Trans. on Comput.-Aided Des. of Integr. Circuits and Syst. **37**(1), 182–192 (2018)

[17] Kole, A., Datta, K., Sengupta, I., Wille, R.: Towards a Cost Metric for Nearest Neighbor Constraints in Reversible Circuits. Rev. Comput. **9138**, 273–278 (2015)

[18] Lin, C., Sur-Kolay, S., Jha, N.K.: PAQCS: Physical Design-Aware Fault-Tolerant Quantum Circuit Synthesis. IEEE Trans. on Very Large Scale Int. Syst. **23**(7), 1221–1234 (2015)

[19] Lye, A., Wille, R., Drechsler, R.: Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits. In: The 20th Asia and South Pacific Design Automation Conference, pp. 178–183 (2015)

[20] Matsuo, A., Yamashita, S.: Changing the Gate Order for Optimal LNN Conversion. In: A. De Vos, R. Wille (eds.) Reversible Computation, Lecture Notes in Computer Science, pp. 89–101. Springer Berlin Heidelberg (2012)

[21] Mulderij, J.: Nearest Neighbor Compliance. Master's thesis, Delft University of Technology, the Netherlands (2019)

[22] Mulderij, J., Aardal, K., Chiscop, I., Phillipson, F.: A polynomial size model with implicit swap gate counting for exact qubit reordering (2019). Submitted

[23] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010). DOI 10.1017/CBO9780511976667

[24] Nielsen, M.A., Chuang, I.L., Grover, L.K.: Quantum Computation and Quantum Information. Am. J. of Phys. **70**(5), 558–559 (2002)

[25] Pedram, M., Shafaei, A.: Layout Optimization for Quantum Circuits with Linear Nearest Neighbor Architectures. IEEE Circuits and Syst. Mag. **16**(2), 62–74 (2016)

[26] Pham, P., Svore, K.M.: A 2d Nearest-Neighbor Quantum Architecture for Factoring in Polylogarithmic Depth. arXiv:1207.6655 [quant-ph] (2012). ArXiv: 1207.6655

[27] Saeedi, M., Wille, R., Drechsler, R.: Synthesis of quantum circuits for linear nearest neighbor architectures. Quantum Inf. Process. **10**(3), 355–377 (2011)

[28] Shafaei, A., Saeedi, M., Pedram, M.: Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures. In: 2013 50th ACM/EDAC/IEEE Design Automation Conference (DAC), pp. 1–6 (2013)

[29] Shafaei, A., Saeedi, M., Pedram, M.: Qubit placement to minimize communication overhead in 2d quantum architectures. In: 2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 495–500 (2014)

[30] Shrivastwa, R.R., Datta, K., Sengupta, I.: Fast Qubit Placement in 2d Architecture Using Nearest Neighbor Realization. In: 2015 IEEE International Symposium on Nanoelectronic and Information Systems, pp. 95–100 (2015)

[31] Tan, Y.y., Cheng, X.y., Guan, Z.j., Liu, Y., Ma, H.: Multi-strategy based quantum cost reduction of linear nearest-neighbor quantum circuit. Quantum Inf. Process. **17**(3), 61 (2018)

[32] Wehner, S., Elkouss, D., Hanson, R.: Quantum internet: A vision for the road ahead. Sci. **362**(6412) (2018). DOI 10.1126/science.aam9288. URL `http://science.sciencemag.org/content/362/6412/eaam9288`

[33] Wille, R., Burgholzer, L., Zulehner, A.: Mapping Quantum Circuits to IBM QX Architectures Using the Minimal Number of SWAP and H Operations. In: Proceedings of the 56th Annual Design Automation Conference 2019 on - DAC '19, pp. 1–6. ACM Press, Las Vegas, NV, USA (2019)

[34] Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: An Online Resource for Reversible Functions and Reversible Circuits. In: 38th International Symposium on Multiple Valued Logic (ismvl 2008), pp. 220–225 (2008)

[35] Wille, R., Keszocze, O., Walter, M., Rohrs, P., Chattopadhyay, A., Drechsler, R.: Look-ahead schemes for nearest neighbor optimization of 1d and 2d quantum circuits. In: 2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 292–297. IEEE, Macao, Macao (2016)

[36] Wille, R., Lye, A., Drechsler, R.: Exact Reordering of Circuit Lines for Nearest Neighbor Quantum Architectures. IEEE Trans. on Comput.-Aided Des. of Integr. Circuits and Syst. **33**(12), 1818–1831 (2014)

[37] Yimsiriwattana, A., Lomonaco Jr, S.J.: Distributed quantum computing: A distributed shor algorithm. In: Quantum Information and Computation II, vol. 5436, pp. 360–372. International Society for Optics and Photonics (2004)

[38] Zulehner, A., Bauer, H., Wille, R.: Evaluating the Flexibility of A* for Mapping Quantum Circuits. In: M.K. Thomsen, M. Soeken (eds.) Reversible Computation, vol. 11497, pp. 171–190. Springer International Publishing, Cham (2019)

[39] Zulehner, A., Paler, A., Wille, R.: An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures. IEEE Trans. on Comput.-Aided Des. of Integr. Circuits and Syst. **38**(7), 1226–1236 (2019)

Appendix A

Appendix

A.1 The Generalized SWAP-gate

Suppose we have two quantum mechanical systems with associated complex Euclidean spaces of dimension n, i.e., $\mathcal{X} = \mathbf{C}^n$. Let $\{|0\rangle, \ldots, |n-1\rangle\}$ be a basis of this Hilbert space. Consider the following operator on $\mathcal{X} \otimes \mathcal{X}$

$$U = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} |i\rangle \langle j| \otimes |j\rangle \langle i| . \tag{A.1.1}$$

Then U is the SWAP-operation on the pair of systems, to see this, let $|k\rangle |\ell\rangle \in \mathcal{X} \otimes \mathcal{X}$, then

$$U |k\rangle |\ell\rangle = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (|i\rangle \langle j| \otimes |j\rangle \langle i|)(|k\rangle \otimes |\ell\rangle) \tag{A.1.2}$$

$$= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} |i\rangle \langle j|k\rangle \otimes |j\rangle \langle i|\ell\rangle \tag{A.1.3}$$

$$= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} |i\rangle \delta_{jk} \otimes |j\rangle \delta_{i\ell} = |\ell\rangle |k\rangle . \tag{A.1.4}$$

Note that $|i\rangle \langle j| = (|j\rangle \langle i|)^\dagger$. This means that if we want to generate a matrix representation of the SWAP_n operation we can build a $n^2 \times n^2$-matrix by joining together $n \times n$-matrices into an $n \times n$-matrix, where the element (i, j) is the matrix $|j\rangle \langle i|$, for example when $n = 2$ we have:

$$\mathrm{SWAP}_2 = \begin{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} , \tag{A.1.5}$$

and similarly when $n = 3$:

$$\mathrm{SWAP}_3 = \begin{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} & \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} . \tag{A.1.6}$$

A.2 The Operators P_0 and P_1 from Ambainis' Protocol

We apply the Gram-Schmidt orthogonalisation process to the set of linearly independent states

$$\frac{1}{\sqrt{2}}(|00\rangle + |22\rangle), |01\rangle, |02\rangle, |10\rangle, |11\rangle, |12\rangle, |20\rangle, |21\rangle \text{ and } |22\rangle. \tag{A.2.1}$$

This gives us the following operator

$$P_0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in L(\mathbf{C}^3 \otimes \mathbf{C}^3). \tag{A.2.2}$$

Similarly, we apply the process to the set of states

$$\frac{1}{\sqrt{2}}(|11\rangle + |22\rangle), |00\rangle, |01\rangle, |02\rangle, |10\rangle, |12\rangle, |20\rangle, |21\rangle \text{ and } |22\rangle, \tag{A.2.3}$$

and get the unitary operator

$$P_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sqrt{2} & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in L(\mathbf{C}^3 \otimes \mathbf{C}^3). \tag{A.2.4}$$

A.3 Measurement Operators in the Protocol of Berlín et al.

The measurement operators for Alice are relatively easy since she cannot check for abort and just outputs $x + y \mod 2$. The operators are

$$\begin{aligned} \Pi_A = \{\Pi_{A,0}, \Pi_{A,1}, \Pi_{A,\varnothing}\} = \{&I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,0} + I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,1}, \\ &I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,1} + I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,0}, \\ &0\} \subseteq \mathrm{Herm}(\mathcal{A}). \end{aligned} \tag{A.3.1}$$

In Bob's case, he has to measure both Alice her coins and his own coin and depending on this, he will check the state he received. If Bob does not abort the protocol, his output will be $x + y \mod 2$. The measurement operators are $\Pi_B = \{\Pi_{B,0}, \Pi_{B,1}, \Pi_{B,\varnothing}\} \subseteq \mathrm{Herm}(\mathcal{B})$, where

$$\begin{aligned} \Pi_{B,0} = &|\psi_{0,0}\rangle\langle\psi_{0,0}| \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \\ &+ I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \\ &+ I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \\ &+ |\psi_{1,0}\rangle\langle\psi_{1,0}| \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \\ &+ |\psi_{0,1}\rangle\langle\psi_{0,1}| \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \\ &+ I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \\ &+ I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,1} \\ &+ |\psi_{1,1}\rangle\langle\psi_{1,1}| \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,1}, \end{aligned} \tag{A.3.2}$$

and

$$\begin{aligned}\Pi_{B,1} = &|\psi_{0,0}\rangle\langle\psi_{0,0}| \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \\ &+ I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \\ &+ I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \\ &+ |\psi_{1,0}\rangle\langle\psi_{1,0}| \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,1} \otimes \Pi_{C,0} \\ &+ |\psi_{0,1}\rangle\langle\psi_{0,1}| \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \\ &+ I_2 \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \\ &+ I_2 \otimes \Pi_{C,0} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,1} \\ &|\psi_{1,1}\rangle\langle\psi_{1,1}| \otimes \Pi_{C,1} \otimes \Pi_{C,0} \otimes \Pi_{C,1} \otimes \Pi_{C,1},\end{aligned} \tag{A.3.3}$$

the operator that aborts the protocol is

$$\Pi_{B,\varnothing} = I_{\mathcal{A}\otimes\mathcal{M}} - \Pi_{B,0} - \Pi_{B,1}. \tag{A.3.4}$$

A.4 MATLAB Code for Bob's Optimal Cheating Strategy in Ambainis' Protocol

```
% ------------------------------------------------------------------------
%       *** Optimal cheating strategey Semidefinite Program ***
%               *** R. van Houte, November 22, 2019 ***
%                              ----
%                   *** Alice honest, Bob cheats ***
% ------------------------------------------------------------------------
clear all; close all; clc;

cvx_solver sedumi

% Size of the problem
N = 3;                  % Number of rounds in the protocol
dimA = 2*2*3*3;         % Dimension of Alice's private space (over C)
dimM = 3;               % Dimension of the Message space
dimB = 2*2*3*3;         % Dimension of Bob's private space (over C)

dimAM = dimA*dimM;      % Dimension of the space A(x)M
dimBM = dimB*dimM;      % Dimension of the space M(x)B

% Start state of the protocol |0><0| on D(A)
StartStateA = zeros(dimA,dimA);
StartStateA(1,1) = 1;

% ------------------------------------------------------------------------
%       *** Measurements in the protocol ***
% ------------------------------------------------------------------------

% Mearsuring a single qubit in the standard basis
PiC0 = [1;0]*[1;0]';    % Measureing the state |0>
PiC1 = [0;1]*[0;1]';    % Measureing the state |1>

% Alice's final coin measurement
PiA0 = superkron((kron(PiC0,PiC0)+kron(PiC1,PiC1)),eye(3),eye(3));
PiA1 = superkron((kron(PiC0,PiC1)+kron(PiC1,PiC0)),eye(3),eye(3));
PiAabort = zeros(dimA,dimA);    % Alice does not abort the protocol

% ------------------------------------------------------------------------
%       *** Alice's Unitary opererations in the protocol ***
% ------------------------------------------------------------------------

% Hadamard gate
```

```
H = 1/sqrt(2)*[1 1; 1 -1];

% Preparation states
P0 = eye(9);
P0(1,1) = 1/sqrt(2);
P0(1,9) = 1/sqrt(2);
P0(9,1) = 1/sqrt(2);
P0(9,9) = -1/sqrt(2);

P1 = zeros(9,9);
P1(5,1) = 1/sqrt(2);
P1(9,1) = 1/sqrt(2);
P1(5,9) = 1/sqrt(2);
P1(9,9) = -1/sqrt(2);
P1(1:4,2:5) = eye(4);
P1(6:8,6:8) = eye(3);

% Different SWAP gates used in Alice's unitary operations
% Swap qutrits 3 and 5
SWAP_3_and_5 = zeros(dimAM,dimAM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    t = q1*54+q2*27+q5*9+q4*3+q3;
                    SWAP_3_and_5(t+1,s+1)=1;
                end
            end
        end
    end
end

% Swap qubit 2 and qutrit 5
SWAP_2_and_5 = zeros(dimAM,dimAM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    t = q1*54+q5*27+q3*9+q4*3+q2;
                    % Only consider |0>,|1> of the qutrit state
                    if q5 == 0 || q5 == 1
                        SWAP_2_and_5(t+1,s+1)=1;
                    else
                        SWAP_2_and_5(s+1,s+1) = 1;
                    end
                end
            end
        end
    end
end

% Swap qutrits 4 and 5
SWAP_4_and_5 = zeros(dimAM,dimAM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    t = q1*54+q2*27+q3*9+q5*3+q4;
```

```
                    SWAP_4_and_5(t+1,s+1)=1;
                end
            end
        end
    end
end

% CNOT with control on qubit 1 and target on qutrit 5 on the states |0>,|1>
CNOT_1_on_5 = zeros(dimAM,dimAM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    % Only consider |0>,|1> of the qutrit state
                    if (q5 == 0 || q5 == 1)
                        t = q1*54+q2*27+q3*9+q4*3+q1*(1-q5)+(1-q1)*q5;
                        CNOT_1_on_5(t+1,s+1)=1;
                    else
                        CNOT_1_on_5(s+1,s+1) = 1;
                    end
                end
            end
        end
    end
end

% Alice's list of unitary operations
UA1_1 = superkron(H,eye(2),P0,eye(3));
UA1_2 = superkron(PiC0,eye(2),eye(3),eye(3),eye(3))...
        + superkron(PiC1,eye(2),P1*P0',eye(3));
UA1_3 = SWAP_3_and_5;

UA1 = UA1_3*UA1_2*UA1_1;          % First round Alice's unitary operation;
UA2 = CNOT_1_on_5*SWAP_2_and_5;   % Second round Alice's unitary
UA3 = SWAP_4_and_5;               % Third round Alice's unitary

% ------------------------------------------------------------------------
%  *** Optimisation program for finding the optimal cheating strategy ***
% ------------------------------------------------------------------------

cvx_begin sdp

cvx_precision high

% Varbiales: density matrices N+1
variable rhoA0(dimAM,dimAM) hermitian

variable rhoA1full(dimAM,dimAM) hermitian
variable rhoA1(dimAM,dimAM) hermitian

variable rhoA2full(dimAM,dimAM) hermitian
variable rhoA2(dimAM,dimAM) hermitian

variable rhoA3full(dimAM,dimAM) hermitian
variable rhoA3(dimAM,dimAM) hermitian

% Objective
maximize (trace(kron(PiA1,eye(dimM))*rhoA3))

% Partial trace constraints
    % The state on Alice's space remains |0><0|
    rhoA0(1:3:dimAM,1:3:dimAM)+rhoA0(2:3:dimAM,2:3:dimAM)...
```

```
        +rhoA0(3:3:dimAM,3:3:dimAM) == StartStateA;

    % Alice applies the operation UA1 to A(x)M
    rhoA1full == UA1*rhoA0*UA1';

    % Bob cannot access Alice her space
    rhoA1(1:3:dimAM,1:3:dimAM)+rhoA1(2:3:dimAM,2:3:dimAM)+...
        rhoA1(3:3:dimAM,3:3:dimAM) == rhoA1full(1:3:dimAM,1:3:dimAM)...
        +rhoA1full(2:3:dimAM,2:3:dimAM)+rhoA1full(3:3:dimAM,3:3:dimAM);

    % Alice applies the operation UA2 to A(x)M
    rhoA2full == UA2*rhoA1*UA2';

    % Bob cannot access Alice her space
    rhoA2(1:3:dimAM,1:3:dimAM)+rhoA2(2:3:dimAM,2:3:dimAM)+...
        rhoA2(3:3:dimAM,3:3:dimAM) == rhoA2full(1:3:dimAM,1:3:dimAM)+...
        rhoA2full(2:3:dimAM,2:3:dimAM)+rhoA2full(3:3:dimAM,3:3:dimAM);

    % Alice applies the operation UA3 to A(x)M
    rhoA3full == UA3*rhoA2*UA3';

    % Bob cannot access Alice her space
    rhoA3(1:3:dimAM,1:3:dimAM)+rhoA3(2:3:dimAM,2:3:dimAM)+...
        rhoA3(3:3:dimAM,3:3:dimAM) == rhoA3full(1:3:dimAM,1:3:dimAM)+...
        rhoA3full(2:3:dimAM,2:3:dimAM)+rhoA3full(3:3:dimAM,3:3:dimAM);

    % All quantum states are positive semidefinite
    rhoA0 >= 0;
    rhoA1 >= 0;
    rhoA2 >= 0;
    rhoA3 >= 0;

    % Positive semidefiniteness of rhoAifull is automatically the case for
    % feasible solutions

cvx_end

% ------------------------------------------------------------------------
%                       *** End of Script ***
% ------------------------------------------------------------------------
```

A.5 MATLAB Code for Alice's Optimal Cheating Strategy in Ambainis' Protocol

```
% ------------------------------------------------------------------------
%       *** Optimal cheating strategey Semidefinite Program ***
%               *** R. van Houte, November 22, 2019 ***
%                           ----
%                   *** Bob honest, Alice cheats ***
% ------------------------------------------------------------------------
clear all; close all; clc;

cvx_solver sedumi

% Size of the problem
N = 3;                  % Number of rounds in the protocol
dimA = 2*2*3*3;         % Dimension of Alice's private space (over C)
dimM = 3;               % Dimension of the Message space
dimB = 2*2*3*3;         % Dimension of Bob's private space (over C)

dimAM = dimA*dimM;      % Dimension of the space A(x)M
dimBM = dimB*dimM;      % Dimension of the space M(x)B
```

```
% Start state of the protocol |0><0| on D(B)
StartStateB = zeros(dimB,dimB);
StartStateB(1,1) = 1;

% ------------------------------------------------------------------------
%       *** Measurements in the protocol ***
% ------------------------------------------------------------------------

% Mearsuring a single qubit in the standard basis
PiC0 = [1;0]*[1;0]';    % Measureing the state |0>
PiC1 = [0;1]*[0;1]';    % Measureing the state |1>

% States Alice prepares in the beginning of the protocol that Bob will
% check
psi0 = 1/sqrt(2)*[1; 0; 0; 0; 0; 0; 0; 0; 1];   % (|00>+|22>)/sqrt(2)
psi1 = 1/sqrt(2)*[0; 0; 0; 0; 1; 0; 0; 0; 1];   % (|11>+|22>)/sqrt(2)

% Corresponding check measurements
PiQ0 = psi0*psi0';
PiQ1 = psi1*psi1';
PiQabort = kron(eye(3),eye(3))-PiQ0-PiQ1;

% Bob's final coin measurements on the space B
PiB0 = superkron(PiC0,PiC0,PiQ0)+superkron(PiC1,PiC1,PiQ1);
PiB1 = superkron(PiC0,PiC1,PiQ0)+superkron(PiC1,PiC0,PiQ1);
PiBabort = eye(dimB)-PiB0-PiB1;

% ------------------------------------------------------------------------
%       *** Bob's Unitary opererations in the protocol ***
% ------------------------------------------------------------------------

% Hadamard gate
H = 1/sqrt(2)*[1 1; 1 -1];

% Swap qutrits 4 and 5
SWAP_4_and_5 = zeros(dimBM,dimBM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    t = q1*54+q2*27+q3*9+q5*3+q4;
                    SWAP_4_and_5(t+1,s+1)=1;
                end
            end
        end
    end
end

% CNOT with control on qubit 2 and target on qutrit 5 on the states |0>,|1>
CNOT_2_on_5 = zeros(dimBM,dimBM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    % Only consider |0>,|1> of the qutrit state
                    if q5 == 0 || q5 == 1
                        t = q1*54+q2*27+q3*9+q4*3+q2*(1-q5)+(1-q2)*q5;
                        CNOT_2_on_5(t+1,s+1)=1;
                    else
```

```
                        CNOT_2_on_5(s+1,s+1) = 1;
                    end
                end
            end
        end
    end
end

% Swap qubit 1 and qutrit 5
SWAP_1_and_5 = zeros(dimBM,dimBM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    t = q5*54+q2*27+q3*9+q4*3+q1;
                    % Only consider |0>,|1> of the qutrit state
                    if q5 == 0 || q5 == 1
                        SWAP_1_and_5(t+1,s+1)=1;
                    else
                        SWAP_1_and_5(s+1,s+1) = 1;
                    end
                end
            end
        end
    end
end

% Swap qutrit 3 and qutrit 5
SWAP_3_and_5 = zeros(dimBM,dimBM);
for q1 = 0:1
    for q2 = 0:1
        for q3 = 0:2
            for q4 = 0:2
                for q5 = 0:2
                    s = q1*54+q2*27+q3*9+q4*3+q5;
                    t = q1*54+q2*27+q5*9+q4*3+q3;
                    SWAP_3_and_5(t+1,s+1)=1;
                end
            end
        end
    end
end

% Bob's list of unitary opertions
UB1_2 = superkron(eye(2),H,eye(3),eye(3),eye(3));

UB1 = CNOT_2_on_5*UB1_2*SWAP_4_and_5;   % First round Bob's unitary
UB2 = SWAP_1_and_5;                     % Second round Bob's unitary
UB3 = SWAP_3_and_5;                     % Third round Bob's unitary

% ------------------------------------------------------------------------
%  *** Optimisation program for finding the optimal cheating strategy ***
% ------------------------------------------------------------------------

cvx_begin sdp

cvx_precision high

% Varbiales: density matrices N+1
variable rhoB0(dimBM,dimBM) hermitian

variable rhoB1full(dimBM,dimBM) hermitian
```

```
variable rhoB1(dimBM,dimBM) hermitian

variable rhoB2full(dimBM,dimBM) hermitian
variable rhoB2(dimBM,dimBM) hermitian

variable rhoB3full(dimBM,dimBM) hermitian
variable rhoB3(dimBM,dimBM) hermitian


% Objective
maximize (trace(kron(PiB1,eye(dimM))*rhoB3))

% Partial trace constraints
    % The state in Bob's space remains |0><0|
    rhoB0(1:3:dimBM,1:3:dimBM)+rhoB0(2:3:dimBM,2:3:dimBM)+...
        rhoB0(3:3:dimBM,3:3:dimBM) == StartStateB;

    % Bob applies the operation UB1 to M(x)B
    rhoB1full == UB1*rhoB0*UB1';

    % Alice cannot acces Bob's space
    rhoB1(1:3:dimBM,1:3:dimBM)+rhoB1(2:3:dimBM,2:3:dimBM)+...
        rhoB1(3:3:dimBM,3:3:dimBM) == rhoB1full(1:3:dimBM,1:3:dimBM)+...
        rhoB1full(2:3:dimBM,2:3:dimBM)+rhoB1full(3:3:dimBM,3:3:dimBM);

    % Bob applies the operation UB2 to M(x)B
    rhoB2full == UB2*rhoB1*UB2';

    % Alice cannot acces Bob's space
    rhoB2(1:3:dimBM,1:3:dimBM)+rhoB2(2:3:dimBM,2:3:dimBM)+...
        rhoB2(3:3:dimBM,3:3:dimBM) == rhoB2full(1:3:dimBM,1:3:dimBM)+...
        rhoB2full(2:3:dimBM,2:3:dimBM)+rhoB2full(3:3:dimBM,3:3:dimBM);

    % Bob applies the operation UB3 to M(x)B
    rhoB3full == UB3*rhoB2*UB3';

    % Alice cannot acces Bob's space
    rhoB3(1:3:dimBM,1:3:dimBM)+rhoB3(2:3:dimBM,2:3:dimBM)+...
        rhoB3(3:3:dimBM,3:3:dimBM) == rhoB3full(1:3:dimBM,1:3:dimBM)+...
        rhoB3full(2:3:dimBM,2:3:dimBM)+rhoB3full(3:3:dimBM,3:3:dimBM);

    % All quantum states are positive semidefinite
    rhoB0 >= 0;
    rhoB1 >= 0;
    rhoB2 >= 0;
    rhoB3 >= 0;

    % Positive semidefiniteness of rhoAifull is automatically the case for
    % feasible solutions

cvx_end

% ------------------------------------------------------------------------
%                       *** End of Script ***
% ------------------------------------------------------------------------
```