Quantum Computation and Combinatorial Structures

by

Ryan L. Mann

A thesis submitted in satisfaction of the

requirements for the degree of

Doctor of Philosophy

in the

Faculty of Engineering and Information Technology

at the

University of Technology Sydney

February 2019

Abstract

Quantum Computation and Combinatorial Structures

by

Ryan L. Mann

Doctor of Philosophy

University of Technology Sydney

This thesis explores the relationship between quantum computation and combinatorial structures, with the goal of improving our understanding of the complexity of quantum computation. We begin by studying the case when the complexity of combinatorial structures can be used to provide evidence for the hardness of classically simulating quantum computations. To this end, we show that the complexity of evaluating multiplicative-error approximations of Jones polynomials can be used to bound the classical complexity of simulating random quantum computations. We then proceed by studying the contrary case, that is, when do the combinatorial structures allow for an efficient classical simulation of quantum computations? We establish an efficient deterministic approximation algorithm for the Ising model partition function with complex parameters when the interactions and external fields are absolutely bounded close to zero. This provides an efficient classical algorithm for simulating a class of quantum computations with bounded interactions between the qubits.

In the second part of this thesis, we present some independent results on the efficient preparation of Fock states with a high number of photons from a resource of single photons. These Fock states are a fundamental resource in many quantum information protocols.

For my mother.

Certificate of Original Authorship

I, Ryan L. Mann, declare that this thesis is submitted in satisfaction of the requirements for the degree of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

This research is supported by the Australian Government Research Training Program.

Production Note:
Signature removed prior to publication.

Signature of Author

Acknowledgements

I would like to begin by thanking my supervisors Michael Bremner, Peter Rohde, and Min-Hsiu Hsieh. They have shown me exceptional support, guidance, and patience throughout my PhD. They have greatly influenced the way I conduct research, view physics, mathematics, and computer science, and approach life in general, and for this, I am eternally grateful.

I would also like to thank my colleagues and collaborators for helpful discussions. In particular, I would like to thank Gavin Brennen for his willingness to meet and discuss ideas. I would also like to thank the theory group at Sydney University for allowing me to attend their group meetings and drink their beer. I thank Jonathan Dowling for numerous whiskies.

I have been fortunate to have some very good friends in Sydney. I would first like to thank all my friends at UTSOAC and SURMC for welcome distractions from this thesis. I especially thank Krisztina Katona for her support and patience during the writing of this thesis, and Catherine Pinfold and Llewellyn Kurtz for their generous hospitality. I would also like to thank Natalie, Vicki, and Dennis Harrold for their support during the first few years of my PhD. I owe special thanks to my long-time friends from home.

This thesis would, of course, not have been possible without the support of my family: my mother Sally, to whom this thesis is dedicated, my brother Liam, my uncle Derek, my grandparents Jennifer and Walter and my great aunt Kim, and our close family friends Paul and Barbara.

Contents

Chapter 1

Introduction

Quantum computation is a model of computation based on the postulates of quantum mechanics. Feynman [Fey82] famously suggested that such a quantum model of computation could provide an exponential improvement over classical computation. There has been some promising evidence of a separation between classical and quantum computation in the field of quantum algorithms. Most notably, Shor's algorithm [Sho99] for integer factorisation, which achieves an exponential improvement over the best-known classical algorithm. Another notable example is Grover's algorithm [Gro96] for searching an unstructured database, which achieves a provable quadratic speedup over any classical algorithm. However, to date, there is no provable exponential separation between classical and quantum computation.

The complexity of quantum computation is completely determined by the complexity of quantum probability amplitudes. These amplitudes are known to be computationally hard to evaluate exactly [FR98]. Unfortunately, quantum mechanics does not provide us with a method for directly measuring these amplitudes or their corresponding probabilities. We must instead infer additive-error approximations to them via repeated computations. These amplitudes can encode evaluations of combinatorial structures, such as Tutte polynomials [AAEL07, She10], Jones polynomials [AJL09], Ising model partition functions [DDVM11, ICBB14], and matrix permanents [Sch04, Rud09]. These structures are known to be computationally hard to evaluate exactly [JVW90] and even approximate up to a multiplicative error [Kup09]. Furthermore, additive-error approximations of such structures are known to completely capture the class of decision problems that can be efficiently solved by a quantum computer with bounded error.

Until recently, it was not obvious how to use the computational hardness of evaluating combinatorial structures, such as Tutte and Jones polynomials, to bound the classical complexity of simulating quantum computation. A seminal result of Aaronson and Arkhipov [AA11a], and independently, Bremner, Montanaro, and Shepherd [BMS16], showed that the average-case complexity of evaluating multiplicative-error approximations of certain combinatorial structures can be used to bound the classical complexity of approximately sampling from the output probability distribution of certain restricted classes of quantum computations. This leads to the following natural question: to what extent can the classical complexity of combinatorial structures improve our understanding of the complexity of quantum computation? In Part I of this thesis we make some partial progress towards answering this question.

Part I of this thesis is structured as follows. In Chapter 2, we introduce the required preliminaries, including computational complexity theory in Section 2.1 and quantum computation in Section 2.2. Specifically, in Section 2.1, we introduce asymptotic notation, computational problems, complexity classifications, and approximation algorithms, and, in Section 2.2, we introduce quantum states, quantum circuits, quantum algorithms, and the Hadamard test. More advanced topics are introduced as they are required.

In Chapter 3, we introduce the combinatorial structures that arise in this thesis, including the Tutte polynomial of graph and matroid theory in Section 3.1, the Jones polynomial of knot theory in Section 3.2, and the Ising model partition function of statistical physics in Section 3.3. We discuss the complexity of exactly and approximately evaluating each of these structures and their relation to quantum computation.

In Chapter 4, we show that the combinatorial structures that arise in the output probability amplitudes of quantum circuits can be used to provide evidence for the classical hardness of simulating random quantum computations. In particular, we show that the complexity of evaluating multiplicative-error approximations of Jones polynomials can be used to bound the classical complexity of approximately simulating random quantum computations. Specifically, we show that under the assumption that **(1)** the Polynomial Hierarchy does not collapse and **(2)** the average-case complexity of multiplicative-error approximations of the Jones polynomial matches the worst-case complexity, then there is no efficient classical algorithm for approximately sampling from the output probability distribution of random quantum computations.

In Chapter 5, we consider the contrary case, that is, when the combinatorial structures that arise in the output probability amplitudes of quantum circuits admit an efficient classical approximation scheme. Any efficient classical approximation scheme for these structures then directly implies an efficient classical algorithm for simulating the corresponding quantum computation. In this chapter, we establish a deterministic polynomial-time approximation scheme for the Ising model partition function when the interactions and external fields are absolutely bounded close to zero. We then proceed to show how our algorithm can be extended to approximate certain output probability amplitudes of quantum circuits.

In Part II, we present some independent results on the efficient preparation of Fock states, which are a key resource in many quantum information protocols. More precisely, in Chapter 6, we establish an efficient scheme for preparing Fock states with a high number of photons from a resource of single photons. Finally, we conclude in Chapter 7.

Part I

Quantum Computation and Combinatorial Structures

Chapter 2

Preliminaries

In this chapter, we cover the preliminaries required for this thesis. In particular, we introduce basic notions in computational complexity theory and quantum computation. More specific material will be introduced as is necessary throughout this thesis.

2.1 Computational Complexity Theory

Computational complexity theory is the study and classification of the resources required to solve computational problems. An important aspect of computational complexity theory is understanding the asymptotic behaviour of the resources required to solve a computational problem with the size of the input. In this section we review some basic notions in complexity theory.

2.1.1 Asymptotic Notation

We define the following standard asymptotic notation.

Definition 2.1 ($O(f(n))$). Let f and g be functions $f, g : \mathbb{N} \rightarrow \mathbb{Z}^+$. Say that $g(n) = O(f(n))$ if there exists positive integers c and n_0 such that, for all $n \geq n_0$, $g(n) \leq cf(n)$. When $g(n) = O(f(n))$ we say that that $f(n)$ is an *asymptotic upper bound* for $g(n)$.

Definition 2.2 ($\Omega(f(n))$). Let f and g be functions $f, g : \mathbb{N} \to \mathbb{Z}^+$. Say that $g(n) = \Omega(f(n))$ if there exists positive integers c and n_0 such that, for all $n \geq n_0$, $g(n) \geq cf(n)$. When $g(n) = \Omega(f(n))$ we say that that $f(n)$ is an *asymptotic lower bound* for $g(n)$.

Definition 2.3 ($\Theta(f(n))$). Let f and g be functions $f, g : \mathbb{N} \to \mathbb{Z}^+$. Say that $g(n) = \Theta(f(n))$ if there exists positive integers c_1, c_2, and n_0 such that, for all $n \geq n_0$, $c_1 f(n) \leq g(n) \leq c_2 f(n)$.

Definition 2.4 ($o(f(n))$). Let f and g be functions $f, g : \mathbb{N} \to \mathbb{Z}^+$. Say that $g(n) = o(f(n))$ if for any real number $c > 0$, there exists an integer n_0 such that, for all $n \geq n_0$, $g(n) \leq cf(n)$. This is equivalent to saying that $\lim_{n\to\infty} [g(n)/f(n)] = 0$.

Definition 2.5 ($\omega(f(n))$). Let f and g be functions $f, g : \mathbb{N} \to \mathbb{Z}^+$. Say that $g(n) = \omega(f(n))$ if for any real number $c > 0$, there exists an integer n_0 such that, for all $n \geq n_0$, $g(n) \geq cf(n)$. This is equivalent to saying that $\lim_{n\to\infty} [g(n)/f(n)] = \infty$.

2.1.2 Computational Problems

Computational problems are conveniently defined in terms of alphabets, strings, and languages. An *alphabet* is any non-empty finite set Σ. An element of an alphabet is called a *symbol*. A *string* over an alphabet Σ is a finite sequence of symbols from that alphabet. The *length* of a string x, denoted by $|x|$, is the number of symbols that it contains. The string of length zero is called the *empty string*. The set of all possible strings is denoted by Σ^*. A *language* over an alphabet Σ is a subset of Σ^*. If L is a language over an alphabet Σ, then its *complement* $\bar{L}$ is given by $\bar{L} = \Sigma^* \setminus L$.

A *computational problem* is a function that takes as input an *instance* and outputs a *solution*. We encode both instances and solutions as strings over an alphabet Σ, and so computational problems are functions mapping strings over Σ to strings over Σ. Typically we take Σ to be the binary alphabet, i.e., $\Sigma = \{0, 1\}$.

A fundamental class of computational problems are *decision problems*. A decision problem is a Boolean function, i.e., a function of the form $f : \Sigma^* \rightarrow \{0, 1\}$. We identify such a function f with the language $L_f := \{x \in \Sigma^* \mid f(x) = 1\}$. Any decision problem can be expressed as a language recognition problem, that is, given a string x, decide if x is in L_f. An algorithm solves a language recognition problem for a language L by *accepting* any input string in L and *rejecting* any input string not in L.

A closely related class of problems are *search problems*. In a search problem, given an input $x \in \Sigma^*$, we want to find a solution $y \in \Sigma^*$ that is in some relation to x, if such a solution exists. A search problem can be expressed as a relation $R \subseteq \Sigma^* \times \Sigma^*$, where $(x, y) \in R$ if and only if y is a solution to the input x; such a relation is called a *search relation*. A *counting problem* is a function $f : \Sigma^* \rightarrow \mathbb{N}$ that asks for the number of solutions to a given search problem. More precisely, for a search relation R, the corresponding counting problem is the function $f_R(x) := |\{y \mid (x, y) \in R\}|$.

2.1.3 Complexity Classifications

Complexity Classes

Complexity classes are used to classify problems by the resources required to solve them. The most notable complexity classes are **P** (Polynomial Time) and **NP** (Non-Deterministic Polynomial Time). Informally, **P** is the class of decision problems that can be solved in polynomial time and **NP** is the class of decision problems for which the accept instances can be verified in polynomial time.

Definition 2.6 (P). The class **P** consists of all languages L that have a polynomial-time algorithm A such that for any input $x \in \Sigma^*$,

- $x \in L \implies A(x)$ accepts.
- $x \notin L \implies A(x)$ rejects.

Definition 2.7 (NP). The class **NP** consists of all languages L that have a polynomial-time algorithm A such that for any input $x \in \Sigma^*$,

- $x \in L \implies \exists y \in \Sigma^*, |y| = O(\text{poly}(|x|))$: $A(x, y)$ accepts.
- $x \notin L \implies \forall y \in \Sigma^*$: $A(x, y)$ rejects.

Obviously **P** $\subseteq$ **NP**, since we can take y in the definition of **NP** to be the empty string. It is a famous open problem to decide if **P** is equal to **NP**. We shall now define the complexity class **coNP** (Complement of Non-Deterministic Polynomial Time), that is, the class of decision problems whose complement is in **NP**. Informally, **coNP** is the class of decision problems for which the reject instances can be verified in polynomial time.

Definition 2.8 (coNP). The class **coNP** consists of all languages L that have a polynomial-time algorithm A such that for any input $x \in \Sigma^*$,

- $x \in L \implies \forall y \in \Sigma^*$: $A(x, y)$ accepts.
- $x \notin L \implies \exists y \in \Sigma^*, |y| = O(\text{poly}(|x|))$: $A(x, y)$ rejects.

It follows from the definition of **coNP**, that **P** $\subseteq$ **coNP**. Note that **coNP** is not the complement of **NP**. In fact, they have non-empty intersection, since every problem in **P** is also in **NP** $\cap$ **coNP**. A notable example of a problem in **NP** $\cap$ **coNP** that is not known to be in **P** is integer factorisation.

An important complexity class is **#P**, which is the class of counting problems associated with decision problems in **NP**. More precisely, **#P** is the class of functions which count the number of accepting paths to a problem in **NP**.

Definition 2.9 (#P [Val79]). The class **#P** consists of all functions $f : \{0, 1\}^* \to \mathbb{N}$ for which there exists a polynomial $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time algorithm A such that for any $x \in \{0, 1\}^*$,

$$f(x) = \left| \left\{ y \in \{0, 1\}^{p(|x|)} \,\middle|\, A(x, y) \text{ accepts} \right\} \right|.$$

It is often useful for algorithms to employ randomness in their logic. This randomness allows us to reduce the complexity of solving a problem, however, this typically causes the success of the algorithm to become probabilistic. We now define the probabilistic complexity classes **RP** (Randomised Polynomial Time), **BPP** (Bounded-Error Probabilistic Polynomial Time), and **FBPP** (Function Bounded-Error Probabilistic Polynomial Time) that allow for randomised algorithms.

Definition 2.10 (RP [Gil77]). The class **RP** consists of all languages L that have a polynomial-time randomised algorithm A such that for any input $x \in \Sigma^*$,

- $x \in L \implies \mathbf{Pr}[A(x) \text{ accepts}] \geq \frac{1}{2}$.
- $x \notin L \implies \mathbf{Pr}[A(x) \text{ accepts}] = 0$.

RP is the class of decision problems solvable by a randomised algorithm in polynomial time such that accept instances are accepted with probability at least $1/2$ and reject instances are always rejected. Note that the error probability of $1/2$ is completely arbitrary. In fact, we could have chosen any constant non-zero probability less than one, since repeating the algorithm gives an exponentially small probability of error in the number of repetitions.

Definition 2.11 (BPP [Gil77]). The class **BPP** consists of all languages L that have a polynomial-time randomised algorithm A such that for any input $x \in \Sigma^*$,

- $x \in L \implies \mathbf{Pr}[A(x) \text{ accepts}] \geq \frac{2}{3}$.
- $x \notin L \implies \mathbf{Pr}[A(x) \text{ accepts}] \leq \frac{1}{3}$.

BPP is the class of decision problems solvable by a randomised algorithm in polynomial time such that accept instances are accepted with probability at least $2/3$ and reject instances are accepted with probability at most $1/3$. Again, the error probability of $1/3$ is completely arbitrary and can be replaced by any constant non-zero probability less than $1/2$. It is often useful to consider the functional version of **BPP**.

Definition 2.12 (FBPP). The class **FBPP** consists of all search problems $R \subseteq \Sigma^* \times \Sigma^*$ that have a polynomial-time randomised algorithm A such that for any input $x \in \Sigma^*$,

$$\mathbf{Pr}[(x, A(x)) \in R] \geq \frac{2}{3}.$$

We shall now introduce the quantum complexity class **BQP** (Bounded-Error Quantum Polynomial Time), which is the quantum analogue of the complexity class **BPP**. Essentially, **BQP** is the same as **BPP**, except that instead of a polynomial-time randomised algorithm we allow a polynomial-time quantum algorithm, which we introduce in Section 2.2. More precisely, **BQP** is the class of decision problems solvable by a quantum algorithm in polynomial time such that accept instances are accepted with probability at least 2/3 and reject instances are accepted with probability at most 1/3.

Definition 2.13 (BQP [BV97]). The class **BQP** consists of all languages L that have a polynomial-time quantum algorithm A such that for any input $x \in \Sigma^*$,

- $x \in L \implies \mathbf{Pr}[A(x) \text{ accepts}] \geq \frac{2}{3}$.
- $x \notin L \implies \mathbf{Pr}[A(x) \text{ accepts}] \leq \frac{1}{3}$.

Oracles

It is often useful to consider *oracles* which can solve certain computational problems in a single operation. These oracles are an essential tool for investigating the relationship between complexity classes. We shall consider complexity classes of problems solvable by an algorithm with access to an oracle.

Definition 2.14 (Oracle Complexity Class). For complexity classes **A** and **O**, we define the complexity class $\mathbf{A}^{\mathbf{O}}$ to be the problems solvable by an algorithm in **A** with access to an oracle that can solve problems in **O**.

Reductions

A *reduction* is an algorithm for mapping one problem into another problem in such a way that we can obtain a solution to the first problem by solving the second problem. We shall be interested in *polynomial-time reductions*, i.e., reductions that run in polynomial time.

Definition 2.15 (Polynomial-Time Reduction). Given two functions $f, g : \Sigma^* \mapsto \mathbb{N}$ we say that f is polynomial-time reducible to g if there is an algorithm with oracle access to g that computes f in time polynomial in the size of the input. We say that such an algorithm is a polynomial-time reduction from f to g.

Hardness and Completeness

Rather remarkably, there exists problems such that any problem in a given complexity class can be reduced to it. We say that these problems are *hard* for the complexity class. If these problems are also contained in that class then we say that these problems are *complete* for that class. For example, the Boolean satisfiability problem is **NP-complete** since every problem in **NP** can be reduced to it [Coo71, Lev73].

Definition 2.16 (Hardness). A problem is said to be hard for a complexity class if every problem in that class can be reduced to it.

Definition 2.17 (Completeness). A problem is said to be complete for a complexity class if it is hard for that class and contained in that class.

The Polynomial Hierarchy

The Polynomial Hierarchy (**PH**) is an infinite tower of complexity classes that generalise **P**, **NP**, and **coNP** to oracle machines.

Definition 2.18 (Polynomial Hierarchy [Pap03]). The Polynomial Hierarchy is an infinite set of complexity classes $\left\{\Delta_k^{\mathbf{P}}, \Sigma_k^{\mathbf{P}}, \Pi_k^{\mathbf{P}} \mid k \in \mathbb{N}\right\}$, such that $\Delta_0^{\mathbf{P}} = \Sigma_0^{\mathbf{P}} = \Pi_0^{\mathbf{P}} = \mathbf{P}$ and for all $k \geq 1$,

- $\Delta_k^{\mathbf{P}} = \mathbf{P}^{\Sigma_{k-1}^{\mathbf{P}}}$.
- $\Sigma_k^{\mathbf{P}} = \mathbf{NP}^{\Sigma_{k-1}^{\mathbf{P}}}$.
- $\Pi_k^{\mathbf{P}} = \mathbf{coNP}^{\Sigma_{k-1}^{\mathbf{P}}}$.

The complexity class **PH** is defined by $\mathbf{PH} := \bigcup_k \Sigma_k^{\mathbf{P}}$.

It follows from the definition of the Polynomial Hierarchy, that the first level comprises the classes $\mathbf{\Delta_1^P} = \mathbf{P}$, $\mathbf{\Sigma_1^P} = \mathbf{NP}$, and $\mathbf{\Pi_1^P} = \mathbf{coNP}$. The second level of the Polynomial Hierarchy comprises the classes $\mathbf{\Delta_2^P} = \mathbf{P^{NP}}$, $\mathbf{\Sigma_2^P} = \mathbf{NP^{NP}}$, and $\mathbf{\Pi_2^P} = \mathbf{coNP^{NP}}$. We have the following relations between the complexity classes. For each $k \in \mathbb{N}$,

- $\mathbf{\Sigma}_k^{\mathbf{P}} \subseteq \mathbf{\Delta}_{k+1}^{\mathbf{P}} \subseteq \mathbf{\Sigma}_{k+1}^{\mathbf{P}}$.
- $\mathbf{\Pi}_k^{\mathbf{P}} \subseteq \mathbf{\Delta}_{k+1}^{\mathbf{P}} \subseteq \mathbf{\Pi}_{k+1}^{\mathbf{P}}$.

Furthermore, if there is any $k \in \mathbb{N}$ such that $\mathbf{\Sigma}_k^{\mathbf{P}} = \mathbf{\Sigma}_{k+1}^{\mathbf{P}}$, then it follows that for all $l > k$, $\mathbf{\Sigma}_k^{\mathbf{P}} = \mathbf{\Delta}_k^{\mathbf{P}} = \mathbf{\Pi}_k^{\mathbf{P}} = \mathbf{\Sigma}_l^{\mathbf{P}}$. In this case, we say that the Polynomial Hierarchy has *collapsed* to the k^{th} level. It is widely believed that the Polynomial Hierarchy does not collapse. A classic result of Toda [Tod91] states that the Polynomial Hierarchy is contained in $\mathbf{P^{\#P}}$.

Theorem 2.19 (Toda [Tod91]).

$$\mathbf{PH} \subseteq \mathbf{P^{\#P}}.$$

Worst-Case and Average-Case Complexity

Definition 2.20 (Worst-Case Complexity). Worst-case complexity is the complexity of solving a computational problem for the worst instance.

Definition 2.21 (Average-Case Complexity). Average-case complexity is the complexity of solving a computational problem for an average instance.

Worst-case complexity tells us how many resources are required to solve any instance of a problem. Whereas, average-case complexity is useful for understanding how hard a typical instance of a problem is to solve. The average-case and worst-case complexity of a problem are equivalent if a problem is *random self-reducible.* Informally, a problem is random self-reducible if solving any instance of the problem can be reduced to solving one or more random instances of the problem.

2.1.4 Approximation Algorithms

Many computational problems are too hard to solve exactly and often an approximation is sufficient. This is especially relevant in quantum computation as **BQP** can be expressed as an approximation problem.

Notions of Approximation

There are several types of approximations that one might consider. The most common of these are *additive approximations* and *multiplicative approximations*.

Definition 2.22 (Additive ϵ-Approximation). Let α, $\hat{\alpha}$, and ϵ be positive real numbers. Say that $\hat{\alpha}$ is an additive ϵ-approximation of α if $|\alpha - \hat{\alpha}| \leq \epsilon$.

Definition 2.23 (Multiplicative ϵ-Approximation). Let α, $\hat{\alpha}$, and ϵ be positive real numbers. Say that $\hat{\alpha}$ is a multiplicative ϵ-approximation of α if $|\alpha - \hat{\alpha}| \leq \epsilon\alpha$.

Approximating Complex Numbers

To obtain a multiplicative ϵ-approximation of a complex number we require that we have a multiplicative ϵ-approximation of the norm and an additive ϵ-approximation of the argument. This is natural as when we multiply two complex numbers together the norms are multiplied and the arguments are added, and so we have the usual property that multiplicative approximations are preserved under multiplication.

Approximation Schemes

We shall now define the following classes of approximation algorithms.

Definition 2.24 (FPTAS). A fully polynomial-time approximation scheme (FPTAS) for a counting problem $f : \Sigma^* \longrightarrow \mathbb{N}$ is an algorithm A that takes as input an instance $x \in \Sigma^*$ and a positive real number ϵ and outputs a multiplicative ϵ-approximation of $f(x)$ in time polynomial in $|x|$ and $1/\epsilon$.

Definition 2.25 (FPRAS). A fully polynomial-time randomised approximation scheme (FPRAS) for a counting problem $f : \Sigma^* \rightarrow \mathbb{N}$ is an algorithm A that takes as input an instance $x \in \Sigma^*$ and a positive real number ϵ and outputs a multiplicative ϵ-approximation of $f(x)$ with probability at least $2/3$ in time polynomial in $|x|$ and $1/\epsilon$.

We can extend the above definitions to complex-valued functions by requiring that the algorithm produces a multiplicative ϵ-approximation to the complex numbers.

2.2 Quantum Computation

In this section we introduce some basic notions in quantum computation.

2.2.1 Quantum States

A *pure state* is described by a unit vector $|\psi\rangle$ in a complex Hilbert space. We denote its dual vector by $\langle\psi|$. The inner product between two vectors $|\psi\rangle$ and $|\phi\rangle$ is denoted by $\langle\phi|\psi\rangle$. If $|\psi\rangle$ is pure state then the *probability amplitude* for observing the system in the state $|\phi\rangle$ is given by the inner product $\langle\phi|\psi\rangle$. The probability of observing the system in that state is given by the absolute value of the probability amplitude squared, i.e., $|\langle\phi|\psi\rangle|^2$.

Any pure state can be expressed as a linear combination of basis elements of the Hilbert space. More precisely, if $\{|k\rangle\}_k$ is a basis of the Hilbert space, then any pure state in that Hilbert space can be written in the form $|\psi\rangle = \sum_k \alpha_k |k\rangle$ and its dual vector expressed as $\langle\psi| = \sum_k {\alpha_k}^* \langle k|$. Then, for two pure states $|\psi\rangle = \sum_k \alpha_k |k\rangle$ and $|\phi\rangle = \sum_k \beta_k |k\rangle$, the inner product is given by $\langle\phi|\psi\rangle = \sum_k \alpha_k {\beta_k}^*$. Quantum states combine though the *tensor product*, which, for two vectors $|\psi\rangle$ and $|\phi\rangle$, is defined by $(|\psi\rangle \otimes |\phi\rangle)_{ij} = \psi_i\phi_j$. A *mixed state* ρ is a probabilistic mixture of pure states $\rho = \sum_k p_k |\psi_k\rangle \langle\psi_k|$ with $\sum_k p_k = 1$.

The fundamental object in quantum computation is the *quantum bit* or *qubit*. A qubit is a vector in a two-dimensional complex Hilbert space and can therefore be represented as a linear combination of basis elements of the Hilbert space. We shall define the *computational basis* to be such a basis given by the set of vectors $\{|0\rangle, |1\rangle\}$, where $|0\rangle = [1,0]^{\intercal}$ and $|1\rangle = [0,1]^{\intercal}$. Another useful basis is the *Hadamard basis*, given by the set of vectors $\{|+\rangle, |-\rangle\}$, where $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. The above notions may of course be generalised to *qudits*, i.e., a vector in a d-dimensional complex Hilbert space.

2.2.2 Quantum Circuits

A *quantum circuit* is a unitary operator described by a sequence of elementary quantum gates, which each represent a unitary operation. A quantum circuit C takes as input an n-qubit pure state $|\psi\rangle$ and outputs the n-qubit pure state $C|\psi\rangle$. We can express the output state of the quantum circuit as a linear combination of computational basis states $C|\psi\rangle = \sum_{x\in\{0,1\}^n} \alpha_x |x\rangle$. The probability amplitude for observing the system in the state $|x\rangle$ is then given by $\alpha_x = \langle x| C |\psi\rangle$ and the probability by $|\alpha_x|^2 = |\langle x| C |\psi\rangle|^2$. This observation is called a measurement of the system in the *computational basis*. Note that the above definitions can be generalised to qudits in the natural way.

An important class of quantum circuits are those comprising gates from a *universal set*. We say that a set of gates G is universal if any unitary $U \in \mathrm{SU}(d^n)$ can be approximated to error $\epsilon > 0$ by a quantum circuit on n qudits comprising gates from G. The Solovay-Kitaev theorem [KSV02] guarantees that universal gate sets can efficiently approximate each other. More precisely, the Solovay-Kitaev states that any gate that acts on a constant number of qudits can be approximated to error $\epsilon > 0$ by a sequence of gates of length $\mathrm{polylog}(1/\epsilon)$ drawn from any universal set.

The output probability amplitudes of quantum circuits are of particular interest as they are **#P-hard** to evaluate [FR98]. Furthermore, they can encode the solution to several combinatorial structures, including the Tutte polynomial, the Jones polynomial, and the Ising model partition function. Unfortunately, it is not possible to directly access these amplitudes; we can, however, obtain an additive-error approximation to them by using the Hadamard test.

2.2.3 Quantum Algorithms

A *quantum algorithm* can be expressed as a quantum circuit applied to an initial pure state and terminating with a measurement. The result of the measurement is the output of the algorithm. We are typically interested in polynomial-time quantum algorithms, i.e., quantum algorithms where the quantum circuit is described by a sequence of quantum gates that has length polynomial in size of the input.

A notable example of a quantum algorithm is Shor's algorithm for prime factorisation [Sho99]. Given an integer $n = p \times q$, which is the product of two unknown primes p and q, prime factorisation is the problem of determining p and q. Shor's algorithm solves this problem in polynomial time, which is a substantial improvement over the best known heuristic classical algorithm — the general number field sieve [BLP93] — which requires sub-exponential time. This has important implications for public-key cryptosystems, in particular, the RSA (Rivest-Shamir-Adleman) cryptosystem [RSA78], which relies on the hardness of prime factorisation. Shor's algorithm implies that the RSA cryptosystem is insecure against quantum computation.

Another notable quantum algorithm is Grover's algorithm for unstructured search [Gro96]. Given a function $f : \{0,1\}^n \to \{0,1\}$, unstructured search is the problem of finding an x such that $f(x) = 1$, if such an x exists. It is easy to see that any classical algorithm that solves the unstructured search problem requires $\Omega(N)$ evaluations of f, where $N = 2^n$. Rather remarkably, Grover's algorithm solves this problem using $O(\sqrt{N})$ evaluations of f. Furthermore, it is known that any quantum algorithm that solves the unstructured search problem requires $\Omega(\sqrt{N})$ evaluations of f [BBBV97], and therefore, Grover's algorithm is asymptotically optimal. There are many other interesting quantum algorithms — we refer the reader to Ref. [Mon16] for a broad overview.

2.2.4 The Hadamard Test

The *Hadamard test* is a method for producing a random variable with expectation value equal to either the real or imaginary part of a quantum probability amplitude. This allows us to obtain an additive-error approximation to any quantum probability amplitude. Let U be a unitary operator, then the Hadamard test allows us to produce a random variable with expectation value $\mathrm{Re}\left(\langle 0^n| U |0^n\rangle\right)$ by the following procedure.

(1) Prepare the state $|+\rangle |0^n\rangle$.

(2) Apply a controlled-U operation to obtain

$$\frac{1}{\sqrt{2}}(|0\rangle |0^n\rangle + |1\rangle U |0^n\rangle) = \frac{1}{2}\left[|+\rangle (|0^n\rangle + U |0^n\rangle) + |-\rangle (|0^n\rangle - U |0^n\rangle)\right].$$

(3) Measure the first qubit in the $|\pm\rangle$ basis.

(4) For a measurement outcome of ± output ±1.

The measurement outcome probabilities are given by

$$\begin{aligned}\mathbf{Pr}(\pm) =& \frac{1}{4}\left||0^n\rangle \pm U |0^n\rangle\right|^2 \\ =& \frac{1}{2}\left[1 \pm \mathrm{Re}\left(\langle 0^n| U |0^n\rangle\right)\right].\end{aligned}$$

Therefore, the expectation value of the output is $\mathrm{Re}\left(\langle 0^n| U |0^n\rangle\right)$. To get the expectation value of $\mathrm{Im}\left(\langle 0^n| U |0^n\rangle\right)$, repeat the procedure except begin by preparing the state $\frac{1}{\sqrt{2}}(|0\rangle \pm i |1\rangle) |0^n\rangle$. It then follows from the Chernoff-Hoeffding bound that repeating this procedure a polynomial number of times in the number of qubits gives a $(1/\mathrm{poly}(n))$-additive approximation to $\langle 0^n| U |0^n\rangle$.

It is worth noting that when the unitary operator is a quantum circuit of polynomial length, the Hadamard test completely captures the complexity class **BQP**. Furthermore, the Hadamard test allows us to obtain an additive-error approximation to a number of interesting combinatorial structures.

Chapter 3

Combinatorial Structures

In the previous chapter, we showed that quantum computation can be expressed as an approximation problem. This suggests a natural question: how can we describe the functions that are approximated by quantum computation?

In this chapter, we briefly review the combinatorial structures that arise in this thesis, this includes the Tutte polynomial, the Jones polynomial, and the Ising model partition function. We note that the Jones polynomial and the Ising model partition function are specialisations of the Tutte polynomial, and so, any result that applies to these structures also applies to the Tutte polynomial for certain classes of graphs and parameters. These combinatorial structures are interesting from a quantum computational perspective as they emerge in the output probability amplitudes of quantum circuits. Furthermore, these structures have the property that they are **BQP-hard** to approximate up to an additive error. Furthermore, they are known to be **#P-hard** to compute exactly and even up to a multiplicative factor.

3.1 The Tutte Polynomial

The Tutte polynomial is a combinatorial structure with important applications in graph and matroid theory. We now define the Tutte polynomial of a graph and a matroid. We then discuss the classical and quantum complexity of evaluating the Tutte polynomial.

The Tutte polynomial is a bivariate polynomial defined for graphs [Tut47], and more generally, matroids [Cra69].

Definition 3.1 (Tutte Polynomial of a Graph). Let $G = (V, E)$ be a finite graph. Define $k(A)$ to be the number of connected components in the subgraph (V, A). Then the Tutte polynomial of G is a polynomial in x and y, defined by

$$\mathrm{T}(G; x, y) := \sum_{A \subseteq E} (x-1)^{k(A)-k(E)}(y-1)^{k(A)+|A|-|V|}.$$

The *multivariate Tutte polynomial* generalises the Tutte polynomial of a graph by assigning a weight to each edge in the graph.

Definition 3.2 (Multivariate Tutte Polynomial of a Graph [Tut47]). Let $G = (V, E)$ be a finite graph with the weights $\Omega = \{\omega_e\}_{e \in E}$ assigned to its edges. Define $k(A)$ to be the number of connected components in the subgraph (V, A). Then the multivariate Tutte polynomial of G is a polynomial in Ω and an extra variable q, defined by

$$\mathrm{T_M}(G; \Omega, q) := \sum_{A \subseteq E} q^{k(A)} \prod_{e \in A} \omega_e.$$

When the weights are all equal to a constant ω, the standard Tutte polynomial can be recovered by

$$\mathrm{T}(G; x, y) = (x-1)^{-k(E)}(y-1)^{-|V|}\mathrm{T_M}(G; \omega, q),$$

where $\omega = y - 1$ and $q = (x-1)(y-1)$.

We shall now briefly introduce the theory of matroids. The interested reader is referred to the classic textbook of Welsh [Wel76] for a detailed treatment. Matroids were introduced by Whitney [Whi35] as a structure that generalises the notion of linear dependence. There are many equivalent ways to define a matroid. We shall define a matroid by the independence axioms.

Definition 3.3 (Matroid [Wel76]). A matroid is a pair $M = (\mathcal{S}, \mathcal{I})$ consisting of a finite set $\mathcal{S}$, known as the *ground set*, and a collection $\mathcal{I}$ of subsets of $\mathcal{S}$, known as the *independent sets*, such that the following axioms are satisfied.

(1) The empty set is a member of $\mathcal{I}$.

(2) Every subset of a member of $\mathcal{I}$ is a member of $\mathcal{I}$.

(3) If A and B are members of $\mathcal{I}$ and $|A| > |B|$, then there exists an $x \in A \setminus B$ such that $B \cup \{x\}$ is a member of $\mathcal{I}$.

The rank of a subset A of $\mathcal{I}$ is given by the *rank function* $r : 2^{\mathcal{I}} \to \mathbb{N}$ of the matroid defined by $r(A) := \max(|X| \mid X \subseteq A, X \in \mathcal{I})$. The *rank* of a matroid M, denoted $r(M)$, is the rank of the set S.

Every finite graph $G = (V, E)$ induces a matroid M_G as follows. Let the ground set of M_G be the set of edges E and let the independent sets of M_G be the subsets of E that are a forest, i.e., they do not contain a simple cycle. It is easy to check that M_G satisfies the independence axioms. We call such a matroid a *graphic matroid*. Note that not all matroids are graphic.

We shall now define the Tutte polynomial of a matroid, which is such that the Tutte polynomial of a graph G is the Tutte polynomial of the graphic matroid $M(G)$

Definition 3.4 (Tutte Polynomial of a Matroid [Cra69]). Let $M = (\mathcal{S}, \mathcal{I})$ be a finite matroid. Define $r(A)$ to be the rank of the submatroid A. Then the Tutte polynomial of M is a polynomial in x and y, defined by

$$\mathrm{T}(M; x, y) := \sum_{A \subseteq \mathcal{S}} (x-1)^{r(M)-r(A)} (y-1)^{|A|-r(A)}. \tag{3.1}$$

A classic result of Jaeger, Vertigan, and Welsh [JVW90] showed that exactly evaluating the Tutte polynomial is **#P-hard**, except when (x, y) are some special points.

Theorem 3.5 (Jaeger, Vertigan, and Welsh [JVW90]). *The problem of evaluating the Tutte polynomial of a graph at a point in the (x, y)-plane is* **#P-hard** *except when*

$(x-1)(y-1) = 1$ *or when* (x, y) *equals* $(1, 1)$, $(-1, -1)$, $(0, -1)$, $(-1, 0)$, $(i, -i)$, $(-i, i)$, (j, j^2), (j, j^2), *or* (j^2, j), *where* $j = \exp(2\pi i/3)$.

Aharonov et al. [AAEL07] established a quantum algorithm for additively approximating the multivariate Tutte polynomial for certain classes of planar graphs with complex edge weightings and a complex parameter q. The algorithm involves encoding these multivariate Tutte polynomials in the output probability amplitudes of quantum circuits. The Hadamard test is then used to approximate the amplitude and, therefore, the multivariate Tutte polynomial. Furthermore, Aharonov et al. [AAEL07] proved that for many of these classes, approximating the multivariate Tutte polynomial is universal for quantum computation, i.e., **BQP-hard**.

Theorem 3.6 (Aharonov et al. [AAEL07]). *There exists several classes of complex weights and complex parameters q, for which additively approximating the multivariate Tutte polynomial to within a certain scale is* **BQP-hard**.

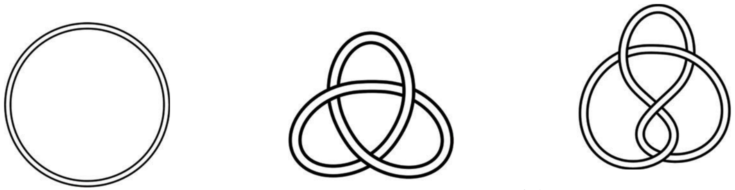
It is not known whether a classical algorithm exists for additively approximating the Tutte polynomial to the same scale achieved by the quantum algorithm.

3.2 The Jones Polynomial

The Jones polynomial is an important knot invariant in topology discovered by Vaughan Jones [Jon85]. We briefly introduce the theory of knots, braids, and the Jones polynomial. We then discuss the classical and quantum complexity of evaluating the Jones polynomial.

Definition 3.7 (Knot). A knot K is subset of points in $\mathbb{R}^3$ that is homeomorphic to a circle.

Informally, a knot is a tangled strand of string with the open ends closed to form a loop. Much like the everyday knots that we use when we tie our shoelaces, ties, and so on — mathematical knots are exactly that, except that the open ends are fused together. The most simple knot you can think of is the *unknot*, also called the *trivial knot*, which is a closed loop without a knot (Fig. 3.1a). Other examples of knots include the *trefoil knot* (Fig. 3.1b), and the *figure eight knot* (Fig. 3.1c).



(a) The unknot. **(b)** The trefoil knot. **(c)** The figure eight knot.

Figure 3.1: Examples of basic knots.

We have seen how a knot is an embedding of a circle in $\mathbb{R}^3$. We can now generalise this idea by considering an embedding of multiple circles in $\mathbb{R}^3$.

Definition 3.8 (Link). A link L is a finite disjoint union of knots $L = \bigcup_i K_i$. Each knot K_i in the union is called a *component* of the link.

Definition 3.9 (Oriented Link). An oriented link is a link in which each component is assigned an orientation.

We can now see that a knot is a link of only one component. The generalisation of the unknot to a link on n components is called the *unlink*, which is a collection of n unknots that are not interlinked. An example of a slightly more interesting link is the *Borromean rings* link (Fig. 3.2), which has the property that removing any single component of the link gives the two component unlink.

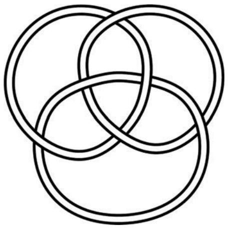


Figure 3.2: The Borromean rings link.

An central problem in knot theory is the *link recognition problem* — given two links are they the same? To answer this, we must first ask, what does it mean for two links to be the same?

Definition 3.10 (Link Equivalence). Two links L_1 and L_2 are said to be equivalent if there exists a orientation-preserving homeomorphism $f : \mathbb{R}^3 \to \mathbb{R}^3$ so that $f(L_1) = L_2$.

Essentially, two links are equivalent if they can be deformed into one another. We can prove that two links are equivalent by producing a set of instructions that will deform one link into the other. However, proving that two links are not equivalent is much more difficult, as we would need to prove that no set of instructions exist.

Link invariants are an important concept in knot theory as they allow us to study the link recognition problem.

Definition 3.11 (Link Invariant). A link invariant is a function from the set of links to some other set, such that the output of the function depends only on the equivalence class of the link.

Definition 3.12 (Jones Polynomial [Jon85]). The Jones polynomial $V_L(\omega)$ is a link invariant, which assigns to each oriented link a Laurent polynomial in the variable $\omega^{1/2}$ with integer coefficients, that is, a polynomial in the variables $\omega^{1/2}$ and $\omega^{-1/2}$ with integer coefficients.

The Jones polynomial is characterised by the *skein relation* and the normalisation that the Jones polynomial of the unknot $V_{\bigcirc}(\omega) = 1$.

Definition 3.13 (Skein Relation). Given three links L_-, L_0, and L_+ that are identical, except for a local region where they differ according to Fig. 3.3, then the following skein relation holds

$$(\omega^{1/2} - \omega^{-1/2})V_{L_0}(\omega) = \omega^{-1}V_{L_+}(\omega) - \omega V_{L_-}(\omega).$$

The skein relation is sufficient for a recursive computation of the Jones polynomial of a link. It follows that the Jones polynomial of a link can be computed in

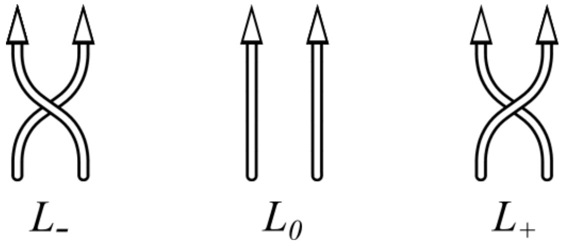


Figure 3.3: Diagrams for the skein relation.

time exponential in the number of crossings. A classic result of Jaeger, Vertigan, and Welsh [JVW90] states that exactly computing the Jones polynomial $V_L(\omega)$ of a link is **#P-hard** except when ω is one of a few special points. Bordewich et al. [BFLW05] showed that it is **BQP-hard** to approximate the Jones polynomial up to an additive error. Kuperberg [Kup09] proved that it remains **#P-hard** to approximate the Jones polynomial up to a multiplicative error.

Theorem 3.14 (Jaeger, Vertigan, and Welsh [JVW90]). *Evaluating the Jones polynomial $V_L(\omega)$ of a link is* **#P-hard** *except when $\omega = \pm\exp(2\pi i/k)$ with $k \in \{1, 2, 3, 4, 6\}$ when it can be evaluated in polynomial time.*

Thistlethwaite [Thi87] showed that along the hyperbola $xy = 1$ the Tutte polynomial of a planar graph specialises to the Jones polynomial of an associated alternating link, i.e, a link where the crossings alternate over and under.

We now introduce the theory of braids, which provides us with a convenient way to represent any link.

Definition 3.15 (Braid). Let

$$A = \{(x, 0, 0) \mid x \in \mathbb{Z}^+, x \leq n\},$$
$$B = \{(x, 0, 1) \mid x \in \mathbb{Z}^+, x \leq n\}.$$

Then, an n-strand braid is a collection of non-intersecting smooth paths in $\mathbb{R}^3$ connecting the points in A to the points in B.

Informally, a braid is a collection of strands of string that may cross over and under each other, and must always move from left to right. An example of a braid is given in Fig. 3.4.

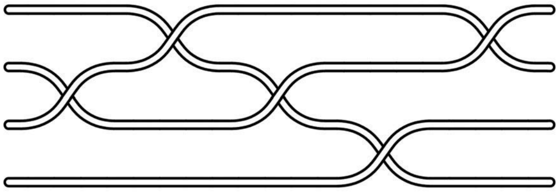


Figure 3.4: An example of a braid on 4 strands.

The set of all braids on n strands form an infinite group B_n, generated by the $n-1$ generators $\{\sigma_i\}$ and their inverses $\{\sigma_i^{-1}\}$. The generator σ_i crosses the i^{th} strand over the $(i+1)^{\text{th}}$ strand and its inverse σ_i^{-1} crosses the i^{th} strand under the $(i+1)^{\text{th}}$ strand.

Definition 3.16 (Braid Group). The braid group on n strands B_n is the group given by the Artin presentation

$$\left\langle \{\sigma_i\}_{i=1}^{n} \middle| \begin{array}{rll} \sigma_i\sigma_{i+1}\sigma_i = & \sigma_{i+1}\sigma_i\sigma_{i+1} & \text{for } 1 \leq i \leq n-2 \\ \sigma_i\sigma_j = & \sigma_j\sigma_i & \text{for } \ |i-j| \geq 2 \end{array} \right\rangle .$$

Each braid can be described by a *braid word*.

Definition 3.17 (Braid Word). A braid word is word on the set of generators $\{\sigma_i\}$ and their inverses $\{\sigma_i^{-1}\}$.

It will be convenient for us to define the length and depth of a braid.

Definition 3.18 (Braid Length). The length of a braid is the number of characters in its word.

Definition 3.19 (Braid Depth). The depth of a braid is the number of steps required to apply the generators of its braid word in order with the assumption that consecutive generators can be applied in parallel if they act on different strands.

We can connect the endpoints of any braid in a number of ways to form a link. For a braid with an even number of strands a natural way to do this is by the *plat closure*.

Definition 3.20 (Plat Closure). The plat closure of a $2n$-strand braid $b \in B_{2n}$ is the link formed by connecting pairs of adjacent strands on the left and the right of the braid. The link that is formed by the plat closure of the braid is often denoted b^{pl}.

Alexander [Ale23] showed that we can generate all possible links this way. Therefore, we can describe any link as the closure of a braid given by its braid word.

Theorem 3.21 (Alexander [Ale23]). *Every link can be represented by the closure of some braid.*

Freedman, Kitaev, and Wang [FKW02] established a quantum algorithm for additively approximating the Jones polynomial at any principal root of unity in polynomial time. This algorithm was later formalised by Aharonov, Jones, and Landau [AJL09]. Freedman, Larsen, and Wang [FLW02] proved that when $\omega = \exp(2\pi i/k)$ is a *principal non-lattice root of unity*, i.e. $k = 5$ or $k \geq 7$, the problem of additively approximating the Jones polynomial is universal for quantum computation. Aharonov and Arad [AA11b] extended this result to values of k that grow polynomially with the number of strands and crossings.

Theorem 3.22 (Aharonov and Arad [AA11b]). *Let ω be a principal non-lattice root of unity, and let $b \in B_{2n}$ be a braid. Then, the problem of additively approximating the Jones polynomial $V_{b^{pl}}(\omega)$ to within the same accuracy as the Aharonov-Jones-Landau algorithm* [AJL09] *is* **BQP-hard**.

In Chapter 4, we show that the complexity of evaluating multiplicative-error approximations of Jones polynomials can be used to bound the classical complexity of approximately simulating random quantum computations.

3.3 The Ising Model

The Ising model plays an vital role in combinatorics and statistical physics. The model is described by a graph $G = (V, E)$, with the vertices representing spins and the edges representing interactions between them. A set of edge weights $\{\omega_e\}_{e \in E}$ characterise the interactions and a set of vertex weights $\{\upsilon_v\}_{v \in V}$ characterise the external fields at each spin. A configuration of the model is an assignment σ of each spin to one of two possible states $\{-1, +1\}$. The *Ising model partition function* is defined as follows.

Definition 3.23 (Ising Model Partition Function). Let $G = (V, E)$ be a graph with the weights $\Omega = \{\omega_e\}_{e \in E}$ assigned to its edges and the weights $\Upsilon = \{\upsilon_v\}_{v \in V}$ assigned to its vertices. Then the Ising model partition function is defined by

$$\mathrm{Z}_{\mathrm{Ising}}(G; \Omega, \Upsilon) := \sum_{\sigma \in \{-1,+1\}^V} w_G(\sigma),$$

where

$$w_G(\sigma) = \exp\left(\sum_{\{u,v\} \in E} \omega_{\{u,v\}} \sigma_u \sigma_v + \sum_{v \in V} \upsilon_v \sigma_v \right).$$

The model is called *ferromagnetic* if $\omega_e > 0$ for all $e \in E$ and *anti-ferromagnetic* if $\omega_e < 0$ for all $e \in E$. Otherwise, the model is called *non-ferromagnetic*.

A classic result of Jerrum and Sinclair [JS93] establishes a fully polynomial-time randomised approximation scheme for the Ising model partition function for all graphs in the ferromagnetic regime with real vertex weights. In contrast, they showed that no such scheme could exists in the anti-ferromagnetic regime unless **RP**=**NP**. Furthermore, they showed that exactly computing the Ising model partition function in the anti-ferromagnetic regime is **#P-hard**.

The Tutte polynomial specialises to the Ising model partition function along the hyperbola $(x - 1)(y - 1) = 2$ [Wel93]. It then follows directly from the theorem of Jaeger, Vertigan, and Welsh [JVW90] that evaluating the Ising model partition function is **#P-hard** in general. The result of Jerrum and Sinclair then corresponds to fully polynomial-time randomised approximation scheme for the Tutte polynomial along the positive branch of the hyperbola $(x-1)(y-1)=2$.

Ising model partition functions with complex parameters arise naturally in the output probability amplitudes of quantum circuits. Furthermore, additive-error approximations of such partition functions are known to be **BQP-hard** [DDVM11, ICBB14]. Goldberg and Guo [GG17] showed that multiplicative-error approximations of these partition functions is **#P-hard**.

In Chapter 5, we establish a deterministic polynomial-time approximation scheme for the Ising model partition function with complex parameters when the interactions and external fields are absolutely bounded close to zero. We then show how our algorithm can be extended to approximate certain output probability amplitudes of quantum circuits.

Chapter 4

The Complexity of Random Quantum Computations

In this chapter, we show that the combinatorial structures that arise in the output probability amplitudes of quantum circuits can be used to provide evidence for the classical hardness of simulating random quantum computations. Specifically, we show that the complexity of evaluating multiplicative-error approximations of Jones polynomials can be used to bound the classical complexity of simulating random quantum computations. We prove that random quantum computations cannot be classically simulated up to a constant total variation distance, under the assumption that **(1)** the Polynomial Hierarchy does not collapse and **(2)** the average-case complexity of multiplicative-error approximations of the Jones polynomial matches the worst-case complexity over a constant fraction of random links. Our results provide a straightforward relationship between the approximation of Jones polynomials and the complexity of random quantum computations.

This chapter is based on joint work with Michael J. Bremner and is available as the preprint “On the Complexity of Random Quantum Computations and the Jones Polynomial” [MB17].

4.1 Introduction

As discussed previously, many quantum circuit classes can be associated with functions that are **#P-hard** to evaluate up to a multiplicative error. This feature has been used to construct arguments in favour of a separation between the power of classical and quantum computation (for a review on this topic see Ref. [LBR17] and Ref. [HM17]). While we do not believe that quantum computers can efficiently evaluate such functions, they play a vital role in defining the complexity of sampling from the output probability distribution of quantum circuits. Terhal and DiVincenzo [TD04] first used this feature to bound the capability of classical computers to simulate constant-depth quantum computations. This was later extended to the problem of sampling from linear optical networks [AA11a] and Instantaneous Quantum Polynomial-time (IQP) circuits [BJS10].

Aaronson and Arkhipov [AA11a] proved an important relationship between the complexity of approximate sampling and the average-case complexity of multiplicative-error approximations to counting problems. They showed that the complexity of evaluating multiplicative-error approximations to matrix permanents can be used to bound the classical complexity of sampling from random linear optical networks up to a constant total variation distance — a notion of approximation that is realistic for quantum computation. They conjecture that **(1)** the average-case complexity of the permanent of Gaussian matrices is **#P-hard** and **(2)** the permanent of Gaussian matrices satisfies a certain anti-concentration bound. Assuming that these conjectures are true, they show that the existence of an efficient classical algorithm which can approximately sample from these networks would imply the collapse of the Polynomial Hierarchy [AA11a]. A similar result was proven for IQP circuits [BMS16] — extending this argument to the quantum circuit model under a different average-case complexity conjecture, where the equivalent anti-concentration conjecture could be proven.

These sampling problems are not just a good candidate for proving a separation between classical and quantum computation, but also for providing experimental benchmarks [BIS$^+$18, HM17]. This has motivated the study of many other sampling problems. Each of these conjecture the equivalence of the average-case and worst-case complexity of multiplicative-error approximations of a given function. These include: **(1)** the permanent of Gaussian matrices [AA11a], **(2)** the gap of degree-three polynomials over $\mathbb{F}_2$ [BMS16, MSM17], **(3)** output probabilities of conjugated Clifford circuits [BFK17], and **(4)** complex-temperature Ising model partition functions over dense [BMS16], sparse [BMS17], and bounded degree graphs [BIS$^+$18, GWD17, BVHS$^+$18, HBVSE18].

These average-case complexity conjectures are each associated with a class of quantum circuits. These quantum circuits are not thought to be universal for quantum computation, with the exception of some of the bounded degree Ising models, but nonetheless become universal under post-selection. Understanding the distinctions between these conjectures is essential for understanding the relationship between these classes of quantum circuits. However, resolving such conjectures would require non-relativising techniques [AC16]. We therefore expect this to be a hard open problem. Recent work by Bouland et al. [BFNV18] proved average-case hardness for the output probability amplitudes of random quantum computations drawn from one of a family of discretisations of the Haar measure.

In this chapter, we show that the complexity of evaluating multiplicative-error approximations of Jones polynomials can be used to bound the classical complexity of approximately simulating random quantum computations. Under the assumption that **(1)** the Polynomial Hierarchy does not collapse and **(2)** the average-case complexity of multiplicative-error approximations of the Jones polynomial matches the worst-case complexity over a constant fraction of random links (Conjecture 4.19), we prove that random quantum computations cannot be classically simulated up to a constant total variation distance (Theorem 4.18). This argument follows as a natural extension to those given for IQP circuits [BJS10, BMS16] and for other classes of random quantum circuits [BIS+18], when combined with results on approximate designs [HL09, BHH16]. Our results provide a straightforward relationship between the approximation of Jones polynomials and the complexity of random quantum computations.

We begin by considering the problem of sampling from random quantum computations that are distributed according to an approximate unitary ($t \geq 2$)-design. We observe that these approximate unitary designs produce output probability distributions that satisfy an anti-concentration bound. This bound is used to prove that if there exists an efficient classical algorithm which can sample from these distributions up to a constant total variation distance, then Stockmeyer's Counting Theorem (Theorem 4.8) can be used to produce multiplicative-error approximations to a constant fraction of their output probabilities (Theorem 4.6). This same observation has been used to establish arguments for the complexity of random quantum circuits [BIS+18, HBVSE18] and conjugated Clifford circuits [BFK17].

We define a natural model of random links via the braid group. A random braid is generated by applying generators of the braid group uniformly at random. A random link is then the plat closure of a random braid. We show that the output probability amplitudes of random quantum computations are proportional to the Jones polynomial of a random link. Furthermore, we show that in the k^{th} path model representation with $k = 5$ or $k \geq 7$, random braids on $2n$ strands of length $\Omega[n(n + \log(1/\epsilon))]$ form an ϵ-approximate unitary 2-design (Corollary 4.17). This leads us to conjecture that it is **#P-hard** to approximate the Jones polynomial, up to a multiplicative error, on at least a constant fraction of random links (Conjecture 4.19). This provides a natural conjecture for bounding the classical complexity of simulating random quantum computations. Our results can be seen as an extension to arguments concerning the complexity of sampling from random quantum circuits [BIS+18, HBVSE18] to a topological model with a natural average-case complexity conjecture.

Finally, we study to what extent random braids can be performed in parallel. We consider applying the generators of a random braid in order and assume that consecutive generators can be applied in parallel if they act on different strands. Recall that the depth of a braid is the number of steps required to apply the generators in order with the assumption that consecutive generators can be applied in parallel if they act on different strands. We prove that random braids on n strands of length $t = \mathrm{poly}(n)$ have depth at most $O\left(\frac{t\log(n)}{n}\right)$ with probability at least $1 - \frac{1}{\mathrm{quasipoly}(n)}$.

This chapter is structured as follows. In Section 4.2, we provide an introduction to random quantum computations and approximate unitary designs. We then state our result on the classical simulation of random quantum computations, which we prove in Section 4.3. In Section 4.4, we review the relationship between Jones polynomials and quantum computing. In Section 4.5, we relate the complexity of random quantum computations to the complexity of approximating the Jones polynomial of random links. In Section 4.6, we investigate the parallelisation of random braids. Finally, we conclude in Section 4.7 with some remarks and open problems.

4.2 Random Quantum Computations

A *random quantum computation* is the action of **(1)** preparing an initial state, **(2)** applying a randomly chosen unitary matrix, and **(3)** measuring in the computational basis. This is equivalent to sampling from a probability distribution $\mathcal{D}_U$, where U is a randomly chosen unitary matrix.

Definition 4.1 ($\mathcal{D}_U$). For a $d \times d$ unitary matrix U, define $\mathcal{D}_U$ to be the probability distribution over integers $x \in [d]$, given by

$$\mathcal{D}_U[x] := |\langle x| U |0\rangle|^2 .$$

It is natural to consider unitary matrices drawn from the uniform distribution. The uniform distribution over the unitary group $\mathrm{U}(d)$ is defined by the *Haar measure*, which is the unique translation-invariant measure on the group. Unfortunately, random unitary matrices drawn from the Haar measure cannot be implemented efficiently by a quantum computer [Kni95].

For our purposes, it is important that the random quantum computations can be implemented efficiently. We achieve this by weakening the requirement that the unitary matrices are drawn from the Haar measure. Instead, we require only that the unitary matrices are drawn from a distribution that is close to the Haar measure.

A *unitary t-design* is a distribution over a finite set of unitary matrices which imitates the properties of the Haar measure up to the t^{th} moment. For convenience, let $\text{Hom}_{(t,t)}(\text{U}(d))$ be the set of polynomials homogeneous of degree t in the matrix elements of U and homogeneous of degree t in the matrix elements of U^*.

Definition 4.2 (Unitary t-Design [RS09]). A distribution $\mathcal{D} = \{p_i, U_i\}$ over unitary matrices in dimension d is a unitary t-design if, for any polynomial $f \in \text{Hom}_{(t,t)}(\text{U}(d))$,

$$\sum_{U_i \in \mathcal{D}} p_i f(U_i) = \int_{\text{U}(d)} f(U) dU.$$

It will be sufficient for us to consider unitary matrices drawn from an approximate unitary t-design, which are often much easier to construct than exact designs.

Definition 4.3 (ϵ-Approximate Unitary t-Design). A distribution $\mathcal{D} = \{p_i, U_i\}$ over unitary matrices in dimension d is an ϵ-approximate unitary t-design if, for any polynomial $f \in \text{Hom}_{(t,t)}(\text{U}(d))$,

$$(1-\epsilon) \int_{\text{U}(d)} f(U) dU \leq \sum_{U_i \in \mathcal{D}} p_i f(U_i) \leq (1+\epsilon) \int_{\text{U}(d)} f(U) dU.$$

Brandao, Harrow, and Horodecki [BHH16] showed that G-local random quantum circuits acting on n qudits composed of polynomially many gates form an approximate unitary poly(n)-design. Here, $G = \{g_i\}_{i=1}^m$ is a universal set of gates containing inverses with each $g_i \in \text{U}(d^2)$ composed of algebraic entries.

Definition 4.4 (G-Local Random Quantum Circuit). At each time step, two indices, i and j, are chosen uniformly at random from $[m]$ and $[n-1]$, respectively. The gate g_i is then applied to the two neighbouring qudits j and $j+1$.

Theorem 4.5 (Brandao, Harrow, and Horodecki [BHH16]). *Fix $d \geq 2$. Let $G = \{g_i\}_{i=1}^{m}$ be a universal set gates containing inverses with each $g_i \in \mathrm{U}(d^2)$ composed of algebraic entries. There exists a constant $\lambda = \lambda(G) > 0$ such that G-local random quantum circuits of length*

$$\lambda n \left\lceil \log_d(4t) \right\rceil^2 t^5 t^{3.1/\log(d)} \left[nt \log(d) + \log(1/\epsilon) \right]$$

form an ϵ-approximate unitary t-design.

We shall, therefore, restrict our attention to random quantum computations where the unitary matrices are drawn from an ϵ-approximate unitary ($t \geq 2$)-design. We are interested in a classical simulation of random quantum computations, for which we have the following result.

Theorem 4.6. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary ($t \geq 2$)-design and let $\mathcal{D}_U$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any U, samples from a probability distribution $\mathcal{D}'$, such that $||\mathcal{D}' - \mathcal{D}_U||_1 \leq \mu$. Then, for any γ such that $0 < \gamma < 1 - \epsilon$, there is an $\mathbf{FBPP}^{\mathbf{NP}^C}$ algorithm which approximates $|\langle 0| U |0\rangle|^2$ up to a multiplicative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of matrices.*

We prove Theorem 4.6 and several supporting lemmas in Section 4.3. Theorem 4.6 tells us that, if there exists an efficient classical algorithm which can approximately sample from any random quantum computation, then, there is an $\mathbf{FBPP}^{\mathbf{NP}}$ algorithm which can approximate $|\langle 0| U |0\rangle|^2$ up to a multiplicative error for a fraction of matrices U. Suppose that this algorithm solves a **#P-hard** problem, then, by Toda's Theorem, the Polynomial Hierarchy collapses to its third level.

In Section 4.5, we show that $|\langle 0| U |0\rangle|^2$ is proportional to the Jones polynomial of a random link, which is known to be **#P-hard** to approximate up to a multiplicative error in the worst case [Kup09]. We conjecture that this remains true in the average case.

4.3 Proof of Theorem 4.6

We now prove Theorem 4.6, which is restated below for convenience. Our proof requires several lemmas which we prove in the remainder of the section.

Theorem 4.6. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 2)$-design and let $\mathcal{D}_U$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any U, samples from a probability distribution $\mathcal{D}'$, such that $||\mathcal{D}' - \mathcal{D}_U||_1 \leq \mu$. Then, for any γ such that $0 < \gamma < 1 - \epsilon$, there is an* $\mathbf{FBPP}^{\mathbf{NP}^C}$ *algorithm which approximates $|\langle 0| U |0\rangle|^2$ up to a multiplicative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of matrices.*

Proof. Lemma 4.7 tells us that, for any $0 < \delta < 1$, there is an $\mathbf{FBPP}^{\mathbf{NP}^C}$ algorithm, which approximates $|\langle x| U |0\rangle|^2$, up to an additive error

$$O\left[(1 + o(1))\frac{\mu(1+\epsilon)}{\delta d} + \frac{|\langle x| U |0\rangle|^2}{\text{poly}(n)}\right],$$

with probability at least $1 - \delta$ over the choice of U. Combining this with Lemma 4.9 and setting $\delta = \frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$, it follows that there is an $\mathbf{FBPP}^{\mathbf{NP}^C}$ algorithm, which approximates $|\langle 0| U |0\rangle|^2$ up to a multiplicative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of matrices U. ■

We now prove Lemma 4.7, which relates the simulation of random quantum computations to approximating individual output probabilities. Our proof closely follows that of Lemma 4 from Ref. [BMS16].

Lemma 4.7. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 1)$-design and let $\mathcal{D}_U$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any U, samples from a probability distribution $\mathcal{D}'$, such that $||\mathcal{D}' - \mathcal{D}_U||_1 \leq \mu$. Then, for any δ such that $0 < \delta < 1$, there is an* $\mathbf{FBPP}^{\mathbf{NP}^C}$ *algorithm, which approximates $|\langle 0| U |0\rangle|^2$, up to an additive error*

$$O\left[(1 + o(1))\frac{\mu(1+\epsilon)}{\delta d} + \frac{|\langle 0| U |0\rangle|^2}{\text{poly}(n)}\right],$$

with probability at least $1 - \delta$ *over the choice of* U.

Proof. Define

$$Q_U := |\langle 0| U |0\rangle|^2, \quad T_U := \mathbf{Pr}[C \text{ outputs } 0 \text{ on input } U].$$

For any U, we can use Stockmeyer's Counting Theorem (Theorem 4.8) to obtain a multiplicative-error approximation T'_U to T_U in $\mathbf{FBPP}^{\mathbf{NP}^C}$,

$$\left|T_U - T'_U\right| \leq \frac{T_U}{\text{poly}(n)}.$$

Then,

$$\begin{aligned}
\left|Q_U - T'_U\right| &\leq |Q_U - T_U| + \left|T_U - T'_U\right| \\
&\leq |Q_U - T_U| + \frac{T_U}{\text{poly}(n)} \\
&\leq |Q_U - T_U| + \frac{(Q_U + |Q_U - T_U|)}{\text{poly}(n)} \\
&= |Q_U - T_U| \left(1 + \frac{1}{\text{poly}(n)}\right) + \frac{Q_U}{\text{poly}(n)}.
\end{aligned}$$

As C approximates $\mathcal{D}_U$ up to an l_1 error μ, it follows from Markov's inequality and the approximate design condition (Lemma 4.11) that, for any $0 < \delta < 1$,

$$\mathop{\mathbf{Pr}}_U \left[|Q_U - T_U| \geq \frac{\mu(1+\epsilon)}{\delta d}\right] \leq \delta.$$

Therefore,

$$\left|Q_U - T'_U\right| \leq \frac{\mu(1+\epsilon)}{\delta d}\left(1 + \frac{1}{\text{poly}(n)}\right) + \frac{Q_U}{\text{poly}(n)},$$

with probability at least $1 - \delta$ over the choice of U. ∎

The proof of Lemma 4.7 requires a classic result of Stockmeyer [Sto85], which allows us to approximately count in the Polynomial Hierarchy.

Theorem 4.8 (Stockmeyer's Counting Theorem [Sto85]). *Let* $f : \{0,1\}^n \to \{0,1\}$ *be a function, and let*

$$p = \mathop{\mathbf{Pr}}_x[f(x) = 1] = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x).$$

Then there exists an $\mathbf{FBPP}^{\mathbf{NP}^f}$ *algorithm, which, for any* $\epsilon = \Omega\left(\frac{p}{\mathrm{poly}(n)}\right)$*, outputs a value* α*, such that*

$$|\alpha - p| < \epsilon.$$

We now prove that unitary matrices distributed according to an ϵ-approximate unitary $(t \geq 2)$-design satisfy certain anti-concentration bounds. A similar result was proven independently by Hangleiter et al. [HBVSE18] in the context of qubit systems.

Lemma 4.9. *Let U be a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary $(t \geq 2)$-design, then, for any unit vectors $|\alpha\rangle$, $|\beta\rangle$ and a constant $0 \leq \gamma \leq 1 - \epsilon$, the following holds*

$$\Pr_U\left[|\langle\alpha| U |\beta\rangle|^2 > \frac{\gamma}{d}\right] \geq \frac{(1-\epsilon-\gamma)^2}{2(1+\epsilon)}.$$

Proof. The Paley-Zygmund inequality (Lemma 4.10) tells us that

$$\Pr_Z\left[Z > \frac{\gamma}{d}\right] \geq \left(1 - \frac{\gamma}{d\mathbb{E}[Z]}\right)^2 \frac{\mathbb{E}[Z]^2}{\mathbb{E}\left[Z^2\right]},$$

for any $0 \leq \gamma \leq d\mathbb{E}[Z]$. Setting $Z = |\langle\alpha| U |\beta\rangle|^2$, it follows from the approximate design condition (Lemma 4.11), that

$$\begin{aligned}\Pr_U\left[Z > \frac{\gamma}{d}\right] \geq& \frac{1}{2}\left(1 - \frac{\gamma}{(1-\epsilon)}\right)^2 \frac{(1-\epsilon)^2}{(1+\epsilon)} \frac{(d+1)}{d} \\ \geq& \frac{1}{2}\left(1 - \frac{\gamma}{1-\epsilon}\right)^2 \frac{(1-\epsilon)^2}{1+\epsilon} \\ =& \frac{(1-\epsilon-\gamma)^2}{2(1+\epsilon)},\end{aligned}$$

for any $0 \leq \gamma \leq 1 - \epsilon$. ∎

The proof of Lemma 4.9 combines the Paley-Zygmund inequality and the approximate design condition. The Paley-Zygmund inequality bounds the probability that a non-negative random variable is small in terms of its first and second moment.

Lemma 4.10 (Paley-Zygmund Inequality). *If $Z \geq 0$ is a random variable with finite variance, and if $0 \leq \theta \leq 1$, then*

$$\Pr_{Z}[Z > \theta\mathbb{E}[Z]] \geq (1-\theta)^2 \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]}.$$

We are interested in bounding the probability that the random variable $Z = |\langle\alpha| U |\beta\rangle|^2$ is small. In the case of an exact unitary ($t \geq 2$)-design the first and second moments of Z match those of the Haar measure. For an ϵ-approximate ($t \geq 2$)-design the approximate design condition bounds the distance of the first and second moments of Z from those of the Haar measure.

Lemma 4.11 (Approximate Design Condition [BH13]). *If U is a $d \times d$ unitary matrix distributed according to an ϵ-approximate unitary t-design, then, for any unit vectors $|\alpha\rangle$, $|\beta\rangle$ and an integer $k \leq t$,*

$$\frac{(1-\epsilon)}{\binom{k+d-1}{d-1}} \leq \mathbb{E}\left[|\langle\alpha| U |\beta\rangle|^{2k}\right] \leq \frac{(1+\epsilon)}{\binom{k+d-1}{d-1}}.$$

4.4 The Jones Polynomial and Quantum Computing

The Aharonov-Jones-Landau algorithm as in Section 3.2 is based on the *path model representation of the braid group* [Jon83, Jon85], which is unitary when $\omega = \exp(2\pi i/k)$ is a principal root of unity. For an integer k, the k^{th} path model representation of the braid group B_{2n} is defined on the vector space spanned by walks of length $2n$, on a $k-1$ vertex path graph G_k, which start and finish on the first vertex.

To calculate the dimension of this vector space it is sufficient to count the number of walks of length $2n$ on the graph G_k. From a combinatorial perspective, the walks on the graph G_k can be seen as *Dyck paths* of length $2n$, which never go above a height $k-2$. It is well known that the number of Dyck paths of length $2n$ is the n^{th} *Catalan number*, which provides an upperbound for the dimension of the vector space.

Definition 4.12 (Catalan number). The n^{th} Catalan number is defined by

$$C_n := \frac{1}{(n+1)}\binom{2n}{n}.$$

Claim 4.13. *For $n \geq 1$,*

$$C_n < 4^n.$$

Proof. The claim follows directly from Stirling's approximation for factorials. ∎

In this representation, each braid $b \in B_{2n}$ is mapped to a unitary matrix $\rho_k(b)$ composed of algebraic entries. These unitary matrices have the property that the expectation value $\langle 0| \rho_k(b) |0\rangle$ is proportional, up to an efficiently computable factor, to the Jones polynomial $V_{b^{pl}}(\omega)$ of the plat closure of b. Aharonov, Jones, and Landau [AJL09] showed that such representations can be implemented efficiently on a quantum computer.

In their construction, the unitary representation of each generator $\rho_k(\sigma_i^{\pm})$ of the braid group B_{2n} acts on a subspace of the Hilbert space of qudits. The Solovay-Kitaev theorem [KSV02] guarantees that these unitary matrices can be implemented efficiently. An entire braid $b \in B_{2n}$ is implemented efficiently by applying the corresponding unitary matrix of each generator in the order of the braid word of b.

4.5 Random Jones Polynomials

We now relate random quantum computations and the Jones polynomial of random links. We define a *random link* to be the plat closure of a *random braid.*

Definition 4.14 (Random Braid). A random braid on $2n$ strands is generated by uniformly at random choosing generators from the set $\{\sigma_i^{\pm}\}_{i=1}^{2n-1}$.

Definition 4.15 (Random Link). A random link is generated by the plat closure of a random braid.

In the k^{th} path model representation the generators of the braid group $\{\sigma_i^{\pm}\}$ are mapped to unitary matrices $\{\rho_k(\sigma_i^{\pm})\}$. In this representation, a random braid is equivalent to a product of random matrices chosen uniformly at random from the set

$\{\rho_k(\sigma_i^{\pm})\}$. Since each $\rho_k(\sigma_i^{\pm})$ acts on a subspace of the Hilbert space of qudits, a random braid is equivalent to a G-local random quantum circuit, with the number of strands proportional to the number of qudits. When $k = 5$ or $k \geq 7$ these gates are universal for quantum computation [FLW02, AA11b].

Theorem 4.16. *In the k^{th} path model representation with $k = 5$ or $k \geq 7$, there exists a constant $\lambda > 0$, such that random braids on $2n$ strands of length*

$$\lambda n \left\lceil \log_2(4t) \right\rceil^2 t^5 t^{3.1/\log(2)} \left[t \log\left(C_n\right) + \log(1/\epsilon) \right],$$

form an ϵ-approximate unitary t-design.

Proof. Since a random braid in the path model representation is equivalent to a G-local random quantum circuit, we can apply Theorem 4.5 to obtain an upperbound on the length of random braids required to form an ϵ-approximate unitary t-design. Combining this with the fact that the dimension of the vector space in the path model representation is bounded from above by the n^{th} Catalan number and that the local dimension is bounded from below by 2 gives the desired result. ■

We note that in the proof of Theorem 4.16, the length of a random braid is chosen to be sufficient for G-local random quantum circuits on a larger Hilbert space of qudits to form an ϵ-approximate unitary t-design, and so, is sufficient for a random braid in the path model representation to form an ϵ-approximate unitary t-design. This may be more than necessary, and, in fact, numerical evidence suggests that in the 5th path model representation, random braids are likely to form an ϵ-approximate unitary t-design in a shorter length than G-local random quantum circuits of certain other universal gate sets [YMKC18].

Corollary 4.17. *In the k^{th} path model representation with $k = 5$ or $k \geq 7$, there exists a constant $\lambda > 0$, such that random braids on $2n$ strands of length*

$$\lambda n \left[n + \log(1/\epsilon) \right],$$

form an ϵ-approximate unitary 2-design.

Proof. The proof follows from setting $t = 2$ in Theorem 4.16 and from the upperbound for the n^{th} Catalan number found in Claim 4.13. ■

We now relate the classical simulation of random quantum computations and the complexity of approximating the Jones polynomial of random links.

Theorem 4.18. *Fix $0 < \epsilon < 1$. Let $k = 5$ or $k \geq 7$ be an integer, and $\omega = \exp(2\pi i/k)$ its corresponding root of unity. Let $b \in B_{2n}$ be a random braid on $2n$ strands of length $\Omega\left[n(n + \log(1/\epsilon))\right]$. Let $\rho_k(b)$ be the k^{th} path model representation of b, and let $\mathcal{D}_{\rho_k(b)}$ be its corresponding probability distribution. Suppose that there is a classical polynomial-time algorithm C, which, for any b, samples from a probability distribution $\mathcal{D}'$, such that $\left\|\mathcal{D}' - \mathcal{D}_{\rho(b)}\right\|_1 \leq \mu$ and assume that Conjecture 4.19 holds. Then, there is a* **BPP$^{\mathbf{NP}}$** *algorithm for solving any problem in* **P$^{\#\mathbf{P}}$** *and by Toda's Theorem the Polynomial Hierarchy collapses to its third level.*

Proof. The proof follows from combining Theorem 4.6, Corollary 4.17, and Toda's Theorem. ■

Conjecture 4.19. *In the notation of Theorem 4.18. For some $0 < \gamma < 1 - \epsilon$, it is* **#P-hard** *to approximate the Jones polynomial $V_{b^{pl}}(\omega)$ up to a multiplicative error $\frac{4\mu(1+\epsilon)^2}{\gamma(1-\epsilon-\gamma)^2} + o(1)$ on at least a $\frac{(1-\epsilon-\gamma)^2}{4(1+\epsilon)}$ fraction of random braids.*

Conjecture 4.19 is based on the average-case complexity of multiplicative-error approximations of Jones polynomials. It is known that it is **#P-hard** to approximate the Jones polynomial up to a multiplicative error in the worst case [Kup09]. Therefore, Conjecture 4.19 states that this worst-case hardness result can be extended to an average-case hardness result.

Assuming that Conjecture 4.19 holds and the Polynomial Hierarchy does not collapse, Theorem 4.18 tells us that there is no efficient classical algorithm which can sample from any random quantum computation. This implies that random quantum computations can not be efficiently simulated by a classical computer.

Remark 4.20. It is worth noting that the 5th path model representation is equivalent to the *Fibonacci representation of the braid group* [SJ08]. Therefore, our results extend to the random braiding of Fibonacci anyons.

4.6 Parallelisation of Random Braids

We now study to what extent random braids can be performed in parallel. We prove that random braids on n strands of length $t = \text{poly}(n)$ have depth at most $O\left(\frac{t\log(n)}{n}\right)$ with probability at least $1 - \frac{1}{\text{quasipoly}(n)}$. Our proof closely follows that of Brown and Fawzi [BF12], who proved a similar result for random quantum circuits.

Definition 4.21 ($E_{n,m,k}$). Let $E_{n,m,k}$ be the event that a random braid on n strands of length m has depth at least k.

Theorem 4.22. *The probability that a random braid on n strands of length t* $= \text{poly}(n)$ *has depth at most* $O\left(\frac{t\log(n)}{n}\right)$ *is at least* $1 - \frac{1}{\text{quasipoly}(n)}$.

Proof. We begin by bounding the probability that a braid on n strands of length m forms a braid of depth at least k. Combining Lemma 4.23 with the fact that there are $\binom{m}{k}$ ways to choose a braid of length k from a braid of length m, it follows from the union bound that

$$\begin{aligned}\mathbf{Pr}[E_{n,m,k}] &\leq \binom{m}{k}\left(\frac{3}{n-1}\right)^{k-1} \\ &\leq \left(\frac{em}{k}\right)^{k}\left(\frac{3}{n-1}\right)^{k-1}.\end{aligned}$$

Setting $m = (n-1)/(3e)$ and $k = \log(n)$, we obtain

$$\begin{aligned}\mathbf{Pr}[E_{n,m,k}] &\leq \frac{(n-1)}{3n^{\log\log(n)}} \\ &\leq \frac{1}{n^{\log\log(n)-1}}.\end{aligned}$$

This proves that every random braid on n strands of length at most $(n-1)/(3e)$ has depth at most $\log(n)$ with probability at least $1 - \frac{1}{n^{\log\log(n)-1}}$. Suppose that we have a random braid on n strands of length t, then the braid has depth at most $O\left(\frac{t\log(n)}{n}\right)$ with probability at least $1-\frac{1}{\text{quasipoly}(n)}$. ■

Lemma 4.23. *The probability that a random braid on n strands of length k has depth k is at most $\left(\frac{3}{n-1}\right)^{k-1}$.*

Proof. The proof follows directly from Claim 4.24. ■

Claim 4.24. *For a random braid on n strands,*

$$\mathbf{Pr}[E_{n,k+1,k+1}] \leq \left(\frac{3}{n-1}\right)\mathbf{Pr}[E_{n,k,k}].$$

Proof. Let us begin with the conditional probability,

$$\mathbf{Pr}[E_{n,k+1,k+1}] = \mathbf{Pr}[E_{n,k+1,k+1}|E_{n,k,k}]\mathbf{Pr}[E_{n,k,k}].$$

Since each generator acts on two neighbouring strands and at most six of the $2(n-1)$ generators can act on these strands, we have

$$\mathbf{Pr}[E_{n,k+1,k+1}|E_{n,k,k}] \leq \left(\frac{3}{n-1}\right).$$

Therefore,

$$\mathbf{Pr}[E_{n,k+1,k+1}] \leq \left(\frac{3}{n-1}\right)\mathbf{Pr}[E_{n,k,k}].$$

This completes the proof. ■

4.7 Conclusion & Outlook

We have provided strong evidence that simulating random quantum computations is intractable for classical computers. Specifically, we have shown that if Conjecture 4.19 holds and the Polynomial Hierarchy does not collapse, then there is no efficient classical algorithm which can approximately sample from the output probability distribution of random quantum computations.

There are a number of natural problems that remain to be solved, the most obvious of which is to resolve Conjecture 4.19. Unfortunately, we are unaware of any proof techniques which are capable of extending the worst-case hardness result to an average-case hardness result. Moreover, the results of Aaronson and Chen [AC16] imply that any proof of this conjecture would require non-relativising techniques. Recent work by Bouland et al. [BFNV18] proved average-case hardness for the output probability amplitudes of random quantum computations drawn from one of a family of discretisations of the Haar measure. However, such a proof would need to hold for multiplicative-error approximations for random quantum computations drawn from an approximation to the Haar measure in order to resolve Conjecture 4.19.

Another natural problem is whether Corollary 4.17 can be strengthened to random braids of a shorter length. In Theorem 4.18, the length of a random braid is determined by the requirement that in the path model representation it is distributed according to an ϵ-approximate unitary 2-design. Therefore, any improvement to this bound yields a stronger version of Theorem 4.18. It is an open problem whether this bound can be improved.

It would also be interesting to adapt our results to other combinatorial structures, such as Tutte polynomials [AAEL07], Turaev-Viro invariants [AJKR10], and matrix permanents [Rud09], which are known to be **#P-hard** to compute in the worst case and **BQP-hard** to approximate up to an additive error.

Chapter 5

Approximation Algorithms for Complex-Valued Ising Models

In the previous chapter we showed that the complexity of combinatorial structures can be used to provide evidence for the classical hardness of simulating random quantum computations. In this chapter, we consider the contrary case, that is, when do the combinatorial structures allows for an efficient classical simulation of quantum computations?

In this chapter, we establish a deterministic polynomial-time approximation scheme for the Ising model partition function when the interactions and external fields are absolutely bounded close to zero. Furthermore, we prove that for this class of Ising models the partition function does not vanish. Our algorithm is based on an approach due to Barvinok for approximating evaluations of a polynomial based on the location of the complex zeros and a technique due to Patel and Regts for efficiently computing the leading coefficients of graph polynomials on bounded degree graphs. Finally, we show how our algorithm can be extended to approximate certain output probability amplitudes of quantum circuits.

This chapter is based on joint work with Michael J. Bremner and is available as the preprint "Approximation Algorithms for Complex-Valued Ising Models on Bounded Degree Graphs" [MB18].

5.1 Introduction

We study the problem of approximating the Ising model partition function in the complex parameter regime on bounded degree graphs. This work is motivated by the close relationship to quantum computation, where it can be shown that approximate evaluations of these partition functions can encode arbitrary quantum computations [DDVM11]. A classic result of Jaeger, Vertigan, and Welsh [JVW90] showed that exactly evaluating these partition functions is **#P-hard**. This was shown to remain true in the approximate case [GG17] and when restricted to graphs of bounded degree [FM17]. Therefore, it seems unlikely that an efficient algorithm exists for approximating the partition function for general parameters on bounded degree graphs. Furthermore, it has been conjectured that this problem remains hard on average over certain classes of interactions and external fields [GWD17, BIS$^+$18, BVHS$^+$18]. Resolving these conjectures is crucial for understanding the complexity of quantum computing.

We establish a deterministic polynomial-time approximation scheme for the Ising model partition function on bounded degree graphs when the interactions and external fields are absolutely bounded close to zero (Corollary 5.29). This provides a lower bound on when the interactions and external fields cause approximations of the Ising model partition function to transition from being contained in **P** to being **#P-hard**. Our algorithm is based on an approach due to Barvinok [Bar15, Bar16b] for approximating evaluations of a polynomial based on the location of the complex zeros and a technique due to Patel and Regts [PR17] for efficiently computing the leading coefficients of graph polynomials on bounded degree graphs.

Barvinok's approach considers the Taylor expansion of the logarithm of a polynomial about an easy to evaluate point. Suppose that we can show that the complex zeros of the polynomial lie in the exterior of a closed disc centred at this point, then it follows that a truncated Taylor expansion provides an additive approximation to the logarithm of the polynomial at any point in the interior of this closed disc. Now observe that an additive approximation of the logarithm of a polynomial corresponds to a multiplicative approximation of the polynomial.

To construct an algorithm from this approach we need to be able to compute the coefficients of the truncated Taylor expansion. Barvinok showed that computing these coefficients can be reduced to computing the leading coefficients of the polynomial itself. However, to achieve the accuracy required for an approximation scheme, we require a number of leading coefficients that is logarithmic in the degree of the polynomial. For many combinatorial structures, directly computing these coefficients requires quasi-polynomial time.

Patel and Regts [PR17] showed that, for several classes of graph polynomials on bounded degree graphs, the leading coefficients can be computed in polynomial time. Their approach is based on expressing the coefficients as linear combinations of connected induced subgraph counts of size logarithmic in the size of the graph. It then follows from a result due to Borgs et al. [BCKL13], which states that, for bounded degree graphs, we can efficiently enumerate all connected induced subgraphs of logarithmic size.

Barvinok and Soberón [BS17] established a deterministic quasi-polynomial time algorithm for approximating the multivariate graph homomorphism partition function on bounded degree graphs when the matrix entries are absolutely bounded close to one. In the case that all matrix entries are exactly equal to one the partition function is easy to evaluate. Barvinok and Soberón proved that for bounded degree graphs when the matrix entries are absolutely bounded close to one, the partition function does not vanish. Finally, they proved that the leading coefficients can be computed in quasi-polynomial time. Patel and Regts [PR17] improved this to a deterministic polynomial-time algorithm by showing that the coefficients can be expressed as linear combinations of connected induced subgraph counts.

In order to establish a polynomial-time approximation scheme for the Ising model partition function, we provide an approximation-preserving polynomial-time reduction to a restricted version of the multivariate graph homomorphism partition function (Proposition 5.27). We extend the results of Barvinok and Soberón [BS17] and Patel and Regts [PR17] to give an algorithm for approximating this restricted version of the multivariate graph homomorphism partition function on bounded degree graphs when the matrix entries are absolutely bounded close to one (Theorem 5.7). As a consequence, we obtain a deterministic polynomial-time approximation scheme for the Ising model partition function on bounded degree graphs when the interactions and external fields are absolutely bounded sufficiently close to zero. Furthermore, we prove that in this case the Ising model partition function does not vanish (Corollary 5.31). This may be of independent interest in statistical physics as the possible points of physical phase transitions are exactly the real limit points of complex zeros [S$^+$05].

Previous work by Liu, Sinclair, and Srivastava [LSS17] studied the problem of approximating the ferromagnetic Ising model partition function based on the location of complex zeros. They gave a deterministic polynomial-time approximation scheme for the Ising model partition function in the ferromagnetic regime for all complex external fields that are not purely imaginary. This can be seen as an algorithmic consequence of the classic Lee-Yang Theorem [LY52], which states that the ferromagnetic Ising model partition function does not vanish except when the external fields are purely imaginary. Peters and Regts [PR18] generalised this result by determining the exact location of zeros in the ferromagnetic and anti-ferromagnetic regime as a function of the inverse temperature and the maximum degree.

Further work has considered the problem of approximating the Ising model partition function on bounded degree graphs based on the decay of correlations property. Sinclair, Srivastava, and Thurley [SST14] established a deterministic polynomial-time approximation scheme for the anti-ferromagnetic Ising model partition function on graphs of maximum degree at most Δ when the interactions and external fields lie in the uniqueness region of the Gibbs measure on the infinite Δ-regular tree, which is exactly the region that the decay of correlation property holds. Sly and Sun [SS12] showed that for interactions outside of this region, unless **RP**=**NP**, there is no fully polynomial-time randomised approximation scheme for the anti-ferromagnetic Ising model partition function on graphs of maximum degree at most $\Delta \geq 3$. Independent work by Galanis, Štefankovič, and Vigoda [GŠV16] established a similar result in the case of no external field. Liu, Sinclair, and Srivastava [LSS18] showed that, in the case of no external field, the Ising model partition function has no zeros in a complex neighbourhood of the decay of correlation regime. This establishes a formal relationship between these two approaches.

Our final result is a polynomial-time algorithm for approximating certain output probability amplitudes of quantum circuits (Corollary 5.37). Our algorithm is based on the observation that complex-valued Ising model partition functions arise in the output probability amplitudes of quantum circuits [DDVM11, ICBB14]. We focus on a class of commuting quantum circuits, known as Instantaneous Quantum Polynomial-time (IQP) circuits [SB09], where the mapping to the Ising model partition function is the most straightforward [SB09, She10, FM17]. Our algorithm allows us to approximate a certain output probability amplitude of a quantum circuit when the corresponding graph has bounded degree and the interactions and external fields are absolutely bounded close to zero. Eldar and Mehraban [EM17] used a similar approach to derive a quasi-polynomial time algorithm for approximating the permanent of a random matrix with unit variance and vanishing mean in the context of linear optical quantum computing.

This chapter is structured as follows. In Section 5.2, we introduce the multivariate graph homomorphism partition function and establish a deterministic polynomial-time algorithm for approximating a restricted version of this partition function on bounded degree graphs when the matrix entries are absolutely bounded close to one.

To establish our algorithm, we require several lemmas, which we prove in Section 5.3 and Section 5.4. In Section 5.5, we provide an approximation-preserving polynomial-time reduction from the Ising model partition function to this restricted version of the multivariate graph homomorphism partition function. We then use this reduction to establish a deterministic polynomial-time approximation scheme for the Ising model partition function on bounded degree graphs when the interactions and external fields are absolutely bounded sufficiently close to zero. In this regime, we prove that the partition function does not vanish. In Section 5.6, we show how our algorithm can be extended to approximate certain output probability amplitudes of quantum circuits. Finally, we conclude in Section 5.7 with some remarks and open problems.

5.2 Graph Homomorphism Partition Functions

A *graph homomorphism* between two graphs G and H is an adjacency-preserving map between the vertex sets, i.e., a map $h : V(G) \to V(H)$ such that $\{u, v\} \in E(G)$ implies $\{h(u), h(v)\} \in E(H)$. Graph homomorphisms generalise the notion of graph colouring [HN04]; for example, a graph homomorphism from a graph G to the complete graph K_q is equivalent to a proper q-colouring of G.

Hell and Nešetřil [HN90] proved that the problem of deciding if a homomorphism between two graphs G and H exists is **NP-complete**. Dyer and Greenhill [DG00] showed that the corresponding counting problem is **#P-hard**, unless the graph has some special structure; otherwise it is in **P**. Furthermore, they showed that this problem remains **#P-hard** when restricted to graphs of bounded degree. The *graph homomorphism partition function* is defined as follows.

Definition 5.1 (Graph Homomorphism Partition Function). Let $G = (V, E)$ be a graph and let $A = (a_{ij})_{m\times m}$ be a $m \times m$ symmetric matrix. Then the graph homomorphism partition function is defined by

$$\mathrm{Hom}(G; A) := \sum_{\phi : V \to [m]} \prod_{\{u,v\} \in E} a_{\phi(u)\phi(v)},$$

where the sum is taken over all maps ϕ from the set of vertices V to the set of matrix indices $[m]$ and the product is taken over all edges E.

The graph homomorphism partition function evaluates to many important combinatorial quantities, including counting the number of graph homomorphisms, proper colourings, and independent sets [Bar16a]. For example, when A is the adjacency matrix of a graph H, $\mathrm{Hom}(G; A)$ counts the number of graph homomorphisms from G to H.

The complexity of computing graph homomorphism partition functions has been widely studied. Dyer and Greenhill [DG00] showed that computing $\mathrm{Hom}(G; A)$ when A is a fixed symmetric binary matrix is either in **P** or **#P-hard**. Moreover, they showed that these hardness results hold even for graphs of maximum degree three. These results were later generalised to non-negative symmetric matrices [BG05], real symmetric matrices [GGJT10], and complex symmetric matrices [CCL10]. Furthermore, the tractability criterion for the matrix is decidable in polynomial time.

The graph homomorphism partition function can be generalised by assigning a $m\times m$ symmetric matrix to each edge. The *multivariate graph homomorphism partition function* is defined as follows.

Definition 5.2 (Multivariate Graph Homomorphism Partition Function). Let $G = (V, E)$ be a graph with the $m \times m$ symmetric matrices $\mathcal{A} = \{(a^e_{ij})_{m\times m}\}_{e\in E}$ assigned to its edges. Then the multivariate graph homomorphism partition function is defined by

$$\mathrm{Hom}_{\mathrm{M}}(G; \mathcal{A}) := \sum_{\phi:V\to[m]} \prod_{\{u,v\}\in E} a^{\{u,v\}}_{\phi(u)\phi(v)}.$$

When the matrices are all equal, it is clear that the multivariate and standard graph homomorphism partition functions are equivalent.

For convenience, let us define the polydisc consisting of all sets of $m\times m$ symmetric matrices with matrix entries absolutely bounded close to one.

Definition 5.3 ($\mathcal{D}_{G,m}(\delta)$). For a graph $G = (V, E)$, $m \in \mathbb{Z}^+$, and $\delta > 0$, we define $\mathcal{D}_{G,m}(\delta)$ to be the closed polydisc consisting of all sets of $m \times m$ symmetric matrices $\mathcal{A} = \{(a^e_{ij})_{m\times m}\}_{e\in E}$, such that $\left|1 - a^e_{ij}\right| \leq \delta$ for all $e \in E$ and all $i, j \in [m]$.

Barvinok and Soberón [BS17] gave a quasi-polynomial time algorithm for approximating $\mathrm{Hom}_{\mathrm{M}}(G; \mathcal{A})$ when G is a graph of maximum degree at most Δ and $\mathcal{A}$ lies in the interior of the closed polydisc $\mathcal{D}_{G,m}(\delta_\Delta)$. Here, $\delta_\Delta > 0$ is an absolute constant. The absolute constants come from Barvinok's monograph [Bar16a], where a simpler proof was presented with better constants. Patel and Regts [PR17] improved this algorithm to run in polynomial time.

Definition 5.4 (δ_Δ). For $\Delta \in \mathbb{Z}^+$, we define the absolute constant δ_Δ by

$$\delta_\Delta := \max_{0<\alpha<\frac{2\pi}{3\Delta}} \left[\sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\alpha\Delta}{2}\right)\right].$$

Remark 5.5. A simple numerical search gives $\delta_3 = 0.18$, $\delta_4 = 0.13$, $\delta_5 = 0.11$, and $\delta_6 = 0.09$. In general, we have $\delta_\Delta = \Omega(1/\Delta)$.

We shall consider a restricted version of the multivariate graph homomorphism partition function, in which the sum is restricted to map a subset of vertices to a fixed index.

Definition 5.6 (Restricted Multivariate Graph Homomorphism Partition Function). Let $G = (V, E)$ be a graph with the $m \times m$ symmetric matrices $\mathcal{A} = \{(a^e_{ij})_{m\times m}\}_{e\in E}$ assigned to its edges. Further let $S \subseteq V$ be a subset of V and let $k \in [m]$ be an integer. Then the restricted multivariate graph homomorphism partition function is defined by

$$\mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A}) := \sum_{\substack{\phi: V\to[m] \\ \phi(s)=k, \forall s\in S}} \prod_{\{u,v\}\in E} a^{\{u,v\}}_{\phi(u)\phi(v)}.$$

The advantage of considering the restricted multivariate graph homomorphism partition function is that, when reduced from the Ising partition function, it will allows us to implement an external magnetic field. This reduction is described in detail in Section 5.5.

We extend the results of Barvinok and Soberón [BS17] and Patel and Regts [PR17] to give a deterministic polynomial-time approximation scheme for the restricted multivariate graph homomorphism partition function. We have the following theorem.

Theorem 5.7. *Fix $\Delta \in \mathbb{Z}^+$ and $0 < \delta < \delta_\Delta$. There is a deterministic polynomial-time approximation scheme for the restricted multivariate graph homomorphism partition function* $\mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A})$ *for all graphs $G = (V, E)$ of maximum degree at most Δ and all $\mathcal{A} = \{(a^e_{ij})_{m\times m}\}_{e\in E}$ in the closed polydisc $\mathcal{D}_{G,m}(\delta)$.*

Proof. Define $P(G; z) := \mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A}(z))$, with $\mathcal{A}(z) = \{(1 + z(a^e_{ij} - 1))_{m\times m}\}_{e\in E}$ and note that $\mathcal{A} = \mathcal{A}(1)$. By Lemma 5.8, we have that $P(G; z)$ does not vanish whenever $\mathcal{A}(z)$ lies in the closed polydisc $\mathcal{D}_{G,m}(\delta_\Delta)$. Since $\mathcal{A}(1)$ lies in the closed polydisc $\mathcal{D}_{G,m}(\delta)$, $P(G; z)$ does not vanish for all $|z| \le \delta_\Delta/\delta$. Let $\{r_i\}_{i=1}^{|E|}$ be the roots of $P(G; z)$. Then, by setting $C = (1 - \delta/\delta_\Delta)^{-1}$ in Lemma 5.9, we have that, for any $0 < \epsilon < 1$, there is a deterministic $(|V|/\epsilon)^{O(1)}$-time algorithm for computing $P(G, 0)$ and the inverse power sums $\left\{\sum_{i=1}^{|E|} r_i^{-j}\right\}_{j=1}^{m}$ for $m = (1 - \delta/\delta_\Delta)^{-1}\log(|V|/\epsilon)$. Then, it follows from Lemma 5.11 that there is a deterministic $(|V|/\epsilon)^{O(1)}$-time algorithm for computing a multiplicative ϵ-approximation to $P(G, z)$ for all $|z| < \delta_\Delta/\delta$. Since $\delta < \delta_\Delta$, we can take $z = 1$. Hence, we have a deterministic polynomial-time algorithm for computing a multiplicative ϵ-approximation to $\mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A})$. This completes the proof. ■

Our proof of Theorem 5.7 requires a result of Barvinok [Bar16a], which states that $\mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A})$ does not vanish on graphs of maximum degree at most Δ when $\mathcal{A}$ lies in the interior of the closed polydisc $\mathcal{D}_{G,m}(\delta_\Delta)$.

Lemma 5.8 (Barvinok [Bar16a]). *Fix $\Delta \in \mathbb{Z}^+$. For any graph $G = (V, E)$ of degree at most Δ and any $\mathcal{A} = \{(a^e_{ij})_{m\times m}\}_{e\in E}$ in the closed polydisc $\mathcal{D}_{G,m}(\delta_\Delta)$, the re-*

stricted multivariate graph homomorphism partition function does not vanish, i.e., $\mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A}) \neq 0$ *for all* $S \subseteq V$ *and all* $k \in [m]$.

Our proof also requires the following lemma, which states that we can efficiently compute the constant term and inverse power sums of the roots of $\mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A}(z))$. We prove Lemma 5.9 in Section 5.3.

Lemma 5.9. *Fix* $\Delta \in \mathbb{Z}^+$, $0 < \epsilon < 1$, *and* $C > 0$. *Let* $G = (V, E)$ *be a graph of maximum degree at most* Δ *with the* $m \times m$ *symmetric matrices* $\mathcal{A}(z) = \{(1 + z(a^e_{ij} - 1))_{m\times m}\}_{e\in E}$ *assigned to its edges. Further let* $\{r_i\}_{i=1}^{|E|}$ *be the roots of the polynomial* $P(G, S, k; z) := \mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A}(z))$. *Then there is a deterministic* $(|V|/\epsilon)^{O(1)}$*-time algorithm for computing* $P(G, 0)$ *and the inverse power sums* $\left\{\sum_{i=1}^{|E|} r_i^{-j}\right\}_{j=1}^{m}$ *for* $m = C\log(|V|/\epsilon)$.

For convenience, let us define the closed disc D of radius δ centred at the origin.

Definition 5.10 ($D(\delta)$). For $\delta > 0$, we define $D(\delta)$ to be the closed disc consisting of all complex numbers z, such that $|z| \leq \delta$.

Finally, we require the following lemma, which arises from the error analysis of Barvinok's interpolation method [Bar15, Bar16b] (see Barvinok [Bar16a]). The lemma states that, in order to get a multiplicative approximation to a polynomial inside its zero-free disc, it is sufficient to compute the constant term and inverse power sums of its roots.

Lemma 5.11 (Barvinok [Bar15, Bar16b, Bar16a]). *Fix* $0 < \epsilon < 1$. *Let* $\{r_i\}_{i=1}^{n}$ *be the roots of the polynomial* $p(z) := \sum_{k=0}^{n} a_k z^k$. *Suppose that, for some* $\delta > 0$, *the roots of* p *lie in the exterior of the closed disc* $D(\delta)$. *Suppose further that we can compute* a_0 *and the inverse power sums* $\left\{\sum_{i=1}^{n} r_i^{-j}\right\}_{j=1}^{m}$ *in time* $\tau(m)$. *Then, for any* t *in the interior of the closed disc* $D(\delta)$, *we can compute a multiplicative* ϵ*-approximation to* $p(t)$ *in time* $O\left[\tau\left(\frac{\log(n/\epsilon)}{1-|t|/\delta}\right)\right]$.

We prove Lemma 5.11 in Section 5.4.

5.3 Proof of Lemma 5.9

We shall now prove Lemma 5.9. Our proof follows from a generalisation of a result due to Patel and Regts [PR17] (Lemma 5.25) and an additional lemma (Lemma 5.26), which we prove in the remainder of the section.

Lemma 5.9. *Fix $\Delta \in \mathbb{Z}^+$, $0 < \epsilon < 1$, and $C > 0$. Let $G = (V, E)$ be a graph of maximum degree at most Δ with the $m \times m$ symmetric matrices $\mathcal{A}(z) = \{(1 + z(a^e_{ij} - 1))_{m\times m}\}_{e\in E}$ assigned to its edges. Further let $\{r_i\}_{i=1}^{|E|}$ be the roots of the polynomial $P(G, S, k; z) := \mathrm{Hom}_{\mathrm{M}}(G, S, k; \mathcal{A}(z))$. Then there is a deterministic $(|V|/\epsilon)^{O(1)}$-time algorithm for computing $P(G, 0)$ and the inverse power sums $\left\{\sum_{i=1}^{|E|} r_i^{-j}\right\}_{j=1}^{m}$ for $m = C\log(|V|/\epsilon)$.*

Proof. The proof follows from combining Lemma 5.25 and Lemma 5.26. ■

We shall begin with the following definitions.

Definition 5.12 ($\mathcal{G}_n$). For $n \in \mathbb{Z}^+$, define $\mathcal{G}_n$ to be the collection of all edge-coloured graphs on at most n vertices.

Definition 5.13 ($G[U]$). For a graph G and a subset of vertices $U \subseteq V(G)$, define $G[U]$ to be the subgraph induced by U.

Definition 5.14 ($\mathrm{Ind}_{\mathrm{C}}(G, H)$). For two edge-coloured graphs G and H, define $\mathrm{Ind}_{\mathrm{C}}(G, H)$ to be the number of induced subgraphs of G that are edge-colour isomorphic to H.

Definition 5.15 (Multiplicative Graph Polynomial). A graph polynomial $P(G; z)$ is said to be multiplicative if $P(\varnothing; z) = 1$ and $P(G \cup H; z) = P(G; z)P(H; z)$ for any two graphs G and H.

Patel and Regts [PR17] proved that, for any *edge-coloured bounded induced graph counting polynomial*, there is an efficient algorithm for computing the constant term and inverse power sums of its roots.

Definition 5.16 (Edge-Coloured Bounded Induced Graph Counting Polynomial [PR17]). Let $P(G;z)$ be a multiplicative graph polynomial defined by $P(G;z) := \sum_{n=0}^{d(G)} \alpha_{G,n} z^n$ with $P(G;0) = 1$. We say that $P(G;z)$ is an edge-coloured bounded induced graph counting polynomial if there exists constants $\mu, \nu \in \mathbb{Z}^+$, such that (1) the coefficients $\alpha_{G,n}$ satisfy $\alpha_{G,n} = \sum_{H \in \mathcal{G}_{\mu n}} \beta_{H,n} \mathrm{Ind}(H,G)$, for certain $\beta_{H,n}$ and (2) the coefficients $\beta_{H,n}$ can be computed in time $O\left(\nu^{|V(H)|}\right)$.

Lemma 5.17 (Patel and Regts [PR17]). *Fix $\Delta \in \mathbb{Z}^+$, $0 < \epsilon < 1$, and $C > 0$. Let $G = (V,E)$ be an edge-coloured graph of maximum degree at most Δ. Further let $P(G;z)$ be an edge-coloured bounded induced graph counting polynomial and let $\{r_i\}_{i=1}^{\deg(P)}$ be its roots. Then there is a deterministic $(|V|/\epsilon)^{O(1)}$-time algorithm for computing $P(G,0)$ and the inverse power sums $\left\{\sum_{i=1}^{\deg(P)} r_i^{-j}\right\}_{j=1}^{m}$ for $m = C\log(|V|/\epsilon)$.*

We shall now generalise the result of Patel and Regts [PR17] to the restricted case, that is, where the graph polynomial is restricted to map a subset of vertices to a fixed index. We begin by extending the previous definitions.

Definition 5.18 (Restricted Graph). A restricted graph is a pair (G,S), where $G = (V,E)$ is a graph and $S \subseteq V$ is a subset of V.

Definition 5.19 ($\mathcal{R}_n$). For $n \in \mathbb{Z}^+$, define $\mathcal{R}_n$ to be the collection of all edge-coloured restricted graphs on at most n vertices.

Definition 5.20 (Induced Restricted Subgraph). For a restricted graph (G,S) and a subset of vertices $U \subseteq V(G)$, the restricted subgraph induced by U is given by $(G[U], S \cap U)$.

Definition 5.21 (Isomorphic Restricted Graphs). Two restricted graphs (G,S) and (H,T) are said to be isomorphic if and only if there is an isomorphism φ from G to H and T is the image of S under φ.

Definition 5.22 ($\mathrm{Ind}_{\mathrm{C}}\left[(G,S),(H,T)\right]$). For two edge-coloured restricted graphs (G,S) and (H,T), define $\mathrm{Ind}_{\mathrm{C}}\left[(G,S),(H,T)\right]$ to be the number of induced restricted subgraphs of (G,S) that are edge-colour isomorphic to (H,T).

Definition 5.23 (Multiplicative Restricted Graph Polynomial). A restricted graph polynomial $P(G,S,k;z)$ is said to be multiplicative if $P(\emptyset,\emptyset,k;z) = 1$ and $P(G \cup H, S \cup T, k; z) = P(G,S,k;z)P(H,T,k;z)$ for any two restricted graphs (G,S) and (H,T) and integer $k \in \mathbb{Z}^+$.

Definition 5.24 (Edge-Coloured Bounded Induced Restricted Graph Counting Polynomial). Let $P(G,S,k;z)$ be a multiplicative restricted graph polynomial defined by $P(G,S,k;z) := \sum_{n=0}^{d(G)} \alpha_{G,S,k,n} z^n$ with $P(G,S,k;0) = 1$. We say that $P(G,S,k;z)$ is an edge-coloured bounded induced restricted graph counting polynomial if there exists constants $\mu, \nu \in \mathbb{Z}^+$, such that (1) the coefficients $\alpha_{G,S,k,n}$ satisfy $\alpha_{G,S,k,n} = \sum_{(H,T)\in\mathcal{R}_{\mu n}} \beta_{H,T,k,n} \mathrm{Ind}_{\mathrm{C}}\left[(G,S),(H,T)\right]$, for certain $\beta_{H,T,k,n}$ and (2) the coefficients $\beta_{H,T,k,n}$ can be computed in time $O\left(\nu^{|V(H)|}\right)$.

The restricted version of Lemma 5.17 is then obtained by following the proof of Patel and Regts [PR17] with the definitions extended in the natural way.

Lemma 5.25. *Fix $\Delta \in \mathbb{Z}^+$, $0 < \epsilon < 1$, and $C > 0$. Let $G = (V,E)$ be an edge-coloured graph of maximum degree at most Δ. Further let $P(G,S,k;z)$ be an edge-coloured bounded induced restricted graph counting polynomial with roots $\{r_i\}_{i=1}^{\deg(P)}$. Then there is a deterministic $(|V|/\epsilon)^{O(1)}$-time algorithm for computing $P(G,S,k,0)$ and the inverse power sums $\left\{\sum_{i=1}^{\deg(P)} r_i^{-j}\right\}_{j=1}^{m}$ for $m = C\log(|V|/\epsilon)$.*

Lemma 5.26. *Let $G = (V,E)$ be a graph with the $m \times m$ symmetric matrices $\mathcal{A}(z) = \{(1 + z(a_{ij}^e - 1))_{m\times m}\}_{e\in E}$ assigned to its edges and let each edge $e \in E$ be assigned a distinct colour. Further let $S \subseteq V$ be a subset of V and let $k \in [m]$ be an integer. Then, up to an efficiently computable factor, the restricted multivariate graph homomorphism partition function $\mathrm{Hom}_{\mathrm{M}}(G,S,k;\mathcal{A}(z))$ is an edge-coloured bounded induced graph counting polynomial.*

Proof. Define $P(G,S,k;z)$ by

$$P(G,S,k;z) := m^{-|V\setminus S|}\mathrm{Hom}_{\mathrm{M}}(G,S,k;\mathcal{A}(z)).$$

Then,

$$\begin{aligned}
P(G,S,k;z) &= m^{-|V\setminus S|}\sum_{\substack{\phi:V\to[m]\\ \phi(s)=k,\forall s\in S}}\prod_{\{u,v\}\in E}\left[1+z\left(a^{\{u,v\}}_{\phi(u)\phi(v)}-1\right)\right]\\
&= m^{-|V\setminus S|}\sum_{n=0}^{|E|}z^n\sum_{\substack{F\subseteq E\\ |F|=n}}\left[\sum_{\substack{\phi:V\to[m]\\ \phi(s)=k,\forall s\in S}}\prod_{\{u,v\}\in F}\left(a^{\{u,v\}}_{\phi(u)\phi(v)}-1\right)\right]\\
&= \sum_{n=0}^{|E|}z^n\sum_{\substack{F\subseteq E\\ |F|=n}}\left[m^{-|V(G[F])\setminus S|}\sum_{\substack{\phi:V(G[F])\to[m]\\ \phi(s)=k,\forall s\in(S\cap V(G[F]))}}\prod_{\{u,v\}\in F}\left(a^{\{u,v\}}_{\phi(u)\phi(v)}-1\right)\right],
\end{aligned}$$

where $G[F]$ is the subgraph of G induced by F. Since the number of vertices in $G[F]$ is at most $2|F|$, we can write

$$\begin{aligned}
P(G,S,k;z) = \sum_{n=0}^{|E|}z^n\sum_{\substack{(H,T)\in\mathcal{R}_{2n}\\ |E(H)|=n}}&\left[m^{-|V(H)\setminus T|}\sum_{\substack{\phi:V(H)\to[m]\\ \phi(t)=k,\forall t\in T}}\prod_{\{u,v\}\in E(H)}\left(a^{\{u,v\}}_{\phi(u)\phi(v)}-1\right)\right]\\
&\times\mathrm{Ind}_{\mathrm{C}}\left[(G,S),(H,T)\right].
\end{aligned}$$

Therefore, we have

$$P(G,S,k;z) = \sum_{n=0}^{|E|}\alpha_{G,S,k,n}z^n,$$

with

$$\alpha_{G,S,k,n} = \sum_{\substack{(H,T)\in\mathcal{R}_{2n}\\ |E(H)|=n}}\beta_{H,T,k,n}\mathrm{Ind}_{\mathrm{C}}\left[(G,S),(H,T)\right]$$

and

$$\beta_{H,T,k,n} = m^{-|V(H)\setminus T|}\sum_{\substack{\phi:V(H)\to[m]\\ \phi(t)=k,\forall t\in T}}\prod_{\{u,v\}\in E(H)}\left(a^{\{u,v\}}_{\phi(u)\phi(v)}-1\right).$$

It is clear that $P(G, S, k; z)$ is a multiplicative restricted graph polynomial with $P(G, S, k; 0) = 1$. Furthermore, for any restricted graph $(H, T) \in \mathcal{R}_{2n}$, the coefficients $\beta_{H,T,k,n}$ can be computed in time $O\left(m^{|V(H)\setminus S|}\right)$. Hence, $P(G, S, k; z)$ is an edge-coloured bounded induced restricted graph counting polynomial with constants $\mu = 2$ and $\nu = m$. This completes the proof. ■

5.4 Proof of Lemma 5.11

We shall now prove Lemma 5.11. The lemma is due to Barvinok [Bar15, Bar16b, Bar16a], however, our proof closely follows that of Patel and Regts [PR17].

Lemma 5.11 (Barvinok [Bar15, Bar16b, Bar16a]). *Fix $0 < \epsilon < 1$. Let $\{r_i\}_{i=1}^n$ be the roots of the polynomial $p(z) := \sum_{k=0}^n a_k z^k$. Suppose that, for some $\delta > 0$, the roots of p lie in the exterior of the closed disc $D(\delta)$. Suppose further that we can compute a_0 and the inverse power sums $\left\{\sum_{i=1}^n r_i^{-j}\right\}_{j=1}^m$ in time $\tau(m)$. Then, for any t in the interior of the closed disc $D(\delta)$, we can compute a multiplicative ϵ-approximation to $p(t)$ in time $O\left[\tau\left(\frac{\log(n/\epsilon)}{1-|t|/\delta}\right)\right]$.*

Proof. Define the function $f(z)$ on the closed disc $D(\delta)$ by

$$f(z) := \log(p(z)),$$

where the branch of the logarithm is chosen by taking the principal value at $p(0)$. By Taylor's Theorem about the point $t = 0$, for each t in the interior of the closed disc $D(\delta)$,

$$f(t) = \sum_{j=0}^{\infty} \frac{t^j}{j!} f^{(j)}(0).$$

Define the Taylor expansion truncated at order m by

$$T_m(f)(t) := f(0) + \sum_{j=1}^{m} \frac{t^j}{j!} f^{(j)}(0).$$

Now, let us write $p(z)$ in terms of its roots. By the Factor Theorem,

$$p(z) = a_n \prod_{i=1}^{n} (z - r_i).$$

Then,

$$f(z) = \log(a_n) + \sum_{i=1}^{n} \log(z - r_i).$$

Therefore,

$$f^{(j)}(0) = -(j-1)! \sum_{i=1}^{n} r_i^{-j}.$$

Let s_j be the j^{th} inverse power sum given by

$$s_j := \sum_{i=1}^{n} r_i^{-j}.$$

Then, by noting that $f(0) = \log(a_0)$,

$$T_m(f)(t) = \log(a_0) - \sum_{j=1}^{m} \frac{s_j t^j}{j}.$$

We shall now show that, for any $0 < \epsilon < 1$, the Taylor expansion truncated at order $m = O(\log(n/\epsilon))$ gives an additive ϵ-approximation to $f(t)$.

$$\begin{aligned} |f(t) - T_m(f)(t)| &\leq \left| \sum_{j=m+1}^{\infty} \frac{s_j t^j}{j} \right| \\ &\leq \frac{1}{m+1} \sum_{j=m+1}^{\infty} \left| s_j t^j \right|. \end{aligned}$$

Since the roots $\{r_i\}_{i=1}^n$ lie in the exterior of the closed disc $D(\delta)$, we have $\left|s_j\right| < n/\delta^j$. Therefore,

$$|f(t) - T_m(f)(t)| \leq \frac{n}{m+1} \sum_{j=m+1}^{\infty} \left(\frac{|t|}{\delta} \right)^j.$$

Since $|t| < \delta$, by the geometric series formula,

$$|f(t) - T_m(f)(t)| \leq \frac{n(|t|/\delta)^{m+1}}{(m+1)(1 - |t|/\delta)}.$$

Taking $m = (1 - |t|/\delta)^{-1} \log(n/\epsilon)$, it follows that

$$|f(t) - T_m(f)(t)| \leq \epsilon.$$

We shall now show that the truncated Taylor expansion is a multiplicative ϵ-approximation to $p(t)$. For the norm, we have

$$\begin{aligned}\left|e^{T_m(f)(t)-f(t)}\right| &\leq e^{|T_m(f)(t)-f(t)|}\\ &\leq e^{\epsilon},\end{aligned}$$

and

$$\left|e^{f(t)-T_m(f)(t)}\right| \leq e^{\epsilon}.$$

Now, for the argument,

$$\begin{aligned}\left|\mathrm{Arg}\left(e^{T_m(f)(t)-f(t)}\right)\right| &= \left|\mathrm{Im}\left[\log\left(e^{f(t)-T_m(f)(t)}\right)\right]\right|\\ &\leq \left|\log\left(e^{f(t)-T_m(f)(t)}\right)\right|\\ &\leq \epsilon.\end{aligned}$$

This completes the proof. ■

5.5 Ising Model Partition Functions

We shall extend the result of Theorem 5.7 to the Ising model partition function. This is achieved by an approximation-preserving polynomial-time reduction from the Ising model partition function to the restricted multivariate graph homomorphism partition function.

Proposition 5.27. *There is an approximation-preserving polynomial-time reduction from the Ising model partition function to the restricted multivariate graph homomorphism partition function.*

Proof. Let $G = (V, E)$ be a graph with the 2×2 symmetric matrices $\mathcal{A} = \{(a^e_{ij})_{2\times 2}\}_{e\in E}$ assigned to its edges. Let us construct a new graph G' from G by the following vertex gadget. For every vertex $v \in V$, add a new vertex s_v and an edge $e_v = \{v, s_v\}$ with a 2×2 symmetric matrix $(b^{e_v}_{ij})_{2\times 2}$ assigned to it. Let $S = \{s_v\}_{v\in V}$, and let $\mathcal{B} = \{(b^{e_v}_{ij})_{2\times 2}\}_{v\in V}$. Then,

$$\begin{aligned}\mathrm{Hom}_{\mathrm{M}}(G', S, 2; \mathcal{A} \cup \mathcal{B}) &= \sum_{\substack{\phi:V(G')\to[2]\\ \phi(s)=2,\forall s\in S}} \prod_{\{u,v\}\in E(G)} a^{\{u,v\}}_{\phi(u)\phi(v)} \prod_{v\in V(G)} b^{e_v}_{\phi(v)\phi(s_v)} \\ &= \sum_{\phi:V(G)\to[2]} \prod_{\{u,v\}\in E(G)} a^{\{u,v\}}_{\phi(u)\phi(v)} \prod_{v\in V(G)} b^{e_v}_{\phi(v)(2)}.\end{aligned}$$

Taking $a^e_{ij} = \exp\left[\omega_e(2i-3)(2j-3)\right]$ and $b^{e_v}_{ij} = \exp\left[v_v(2i-3)(2j-3)\right]$,

$$\begin{aligned}\mathrm{Hom}_{\mathrm{M}}(G', S, 2; \mathcal{A} \cup \mathcal{B}) &= \sum_{\phi:V(G)\to\{-1,+1\}} \exp\left(\sum_{\{u,v\}\in E(G)} \omega_{\{u,v\}}\phi(u)\phi(v) + \sum_{v\in V(G)} v_v\phi(v)\right) \\ &= \sum_{\sigma\in\{-1,+1\}^V} \exp\left(\sum_{\{u,v\}\in E(G)} \omega_{\{u,v\}}\sigma_u\sigma_v + \sum_{v\in V(G)} v_v\sigma_v\right) \\ &= \mathrm{Z}_{\mathrm{Ising}}(G; \Omega, \Upsilon),\end{aligned}$$

where $\Omega = \{\omega_e\}_{e\in E}$ and $\Upsilon = \{v_v\}_{v\in V}$. Hence, we have an approximation-preserving polynomial-time reduction from the Ising model partition function to the restricted multivariate graph homomorphism partition function. This completes the proof. ■

Let us define the following closed polyregion, which arises naturally from applying Proposition 5.27 to Theorem 5.7.

Definition 5.28 ($\mathcal{R}_G(\delta)$). For a graph $G = (V, E)$ and $\delta > 0$, we define $\mathcal{R}_G(\delta)$ to be the closed polyregion consisting of all sets of weights $\Omega = \{\omega_e\}_{e\in E}$ and $\Upsilon = \{v_v\}_{v\in V}$, such that $|1 - e^{\pm\omega_e}| \le \delta$ for all $e \in E$ and $|1 - e^{\pm v_v}| \le \delta$ for all $v \in V$.

We have the following corollary of Theorem 5.7 and Proposition 5.27.

Corollary 5.29. *Fix $\Delta \in \mathbb{Z}^+$ and $0 < \delta < \delta_{\Delta+1}$. There is a deterministic polynomial-time approximation scheme for the Ising model partition function $\mathrm{Z}_{\mathrm{Ising}}(G; \Omega, \Upsilon)$ for all graphs*

$G = (V, E)$ of maximum degree at most Δ and all $\Omega = \{\omega_e\}_{e \in E}$ and all $\Upsilon = \{v_v\}_{v \in V}$ in the closed polyregion $\mathcal{R}_G(\delta)$.

Proof. The proof follows directly from Theorem 5.7 and Proposition 5.27, while noting that the reduction from the Ising model partition to the restricted multivariate graph homomorphism partition function increases the maximum vertex degree by one. ■

Remark 5.30. It is possible to marginally increase the size of the polyregion by applying the *k-thickening* technique of Jaeger, Vertigan, and Welsh [JVW90].

It is worth mentioning that the bounds of Corollary 5.29 are not sharp in general. To see this, let us compare the results in the anti-ferromagnetic regime with no external field, to those of Sinclair, Srivastava, and Thurley [SST14]. In this case, Corollary 5.29 tells us that there is a deterministic polynomial-time approximation scheme for the Ising model partition function on graphs of maximum degree at most Δ when $\omega_e > -\log(\delta_\Delta + 1)$ for all $e \in E$ (noting that in the case of no external field the reduction preserves maximum degree). The results of Sinclair, Srivastava, and Thurley [SST14] give a deterministic polynomial-time approximation scheme when $\Delta \geq 3$ and $\omega_e > -\frac{1}{2}\log\left(\frac{\Delta}{\Delta-2}\right)$ for all $e \in E$. Hence, the bound of Corollary 5.29 is not sharp. It is an open problem to prove a sharp bound in the complex case.

We also have the following corollary concerning the location of the complex zeros of the Ising model partition function on bounded degree graphs.

Corollary 5.31. *Fix $\Delta \in \mathbb{Z}^+$. For any graph $G = (V, E)$ of degree at most Δ and any $\Omega = \{\omega_e\}_{e \in E}$ and $\Upsilon = \{v_v\}_{v \in V}$ in the closed polyregion $\mathcal{R}_G(\delta_{\Delta+1})$, the Ising model partition function does not vanish, i.e., $\mathrm{Z}_{\mathrm{Ising}}(G; \Omega, \Upsilon) \neq 0$.*

Proof. The proof follows directly from Lemma 5.8 and Proposition 5.27. ■

This may be of independent interest in statistical physics as the possible points of physical phase transitions are exactly the real limit points of such complex zeros [S+05].

5.6 Quantum Simulation

Complex-valued Ising model partition functions arise naturally in the output probability amplitudes of quantum circuits [DDVM11, ICBB14]. In particular, for the class of commuting quantum circuits, known as *Instantaneous Quantum Polynomial-time* (IQP) circuits [SB09, She10, FM17]. In this section we shall show how the results of Corollary 5.29 allow us to approximate output probability amplitudes of IQP circuits and, more generally, universal quantum circuits. First introduced by Shepherd and Bremner [SB09], IQP circuits comprise only gates that are diagonal in the Pauli-X basis. An IQP circuit is described by an *X-program.*

Definition 5.32 (X-Program). An X-program is a pair (P, θ), where $P = (p_{ij})_{m\times n}$ is a binary matrix and $\theta \in [-\pi, \pi]$ is a real angle. The matrix P is used to construct a Hamiltonian of m commuting terms acting on n qubits, where each term in the Hamiltonian is a product of Pauli-X operators,

$$\mathrm{H}_{(P,\theta)} := -\theta \sum_{i=1}^{m} \bigotimes_{j=1}^{n} X_j^{p_{ij}}.$$

Thus, the columns of P correspond to qubits and the rows of P correspond to interactions in the Hamiltonian.

An X-program induces a probability distribution $\mathcal{P}_{(P,\theta)}$ known as an *IQP distribution.*

Definition 5.33 ($\mathcal{P}_{(P,\theta)}$). For an X-program (P, θ) with $P = (p_{ij})_{m\times n}$, we define $\mathcal{P}_{(P,\theta)}$ to be the probability distribution over binary strings $x \in \{0, 1\}^n$, given by

$$\mathbf{Pr}[x] := \left|\langle x| \exp\left(-i\mathrm{H}_{(P,\theta)}\right) |0^n\rangle\right|^2 .$$

We shall consider X-programs that are induced by a weighted graph.

Definition 5.34 (Graph-Induced X-Program). For a graph $G = (V, E)$ with the weights $\{\omega_e \in [-\pi, \pi]\}_{e\in E}$ assigned to its edges and the weights $\{v_v \in [-\pi, \pi]\}_{v\in V}$

assigned to its vertices, we define the X-program induced by G to be an X-program $\mathcal{X}_G$ such that

$$\mathrm{H}_{\mathcal{X}_G} = -\sum_{\{u,v\}\in E} \omega_{\{u,v\}} X_u X_v - \sum_{v\in V} v_v X_v .$$

It will be convenient for us to define ψ_G as a specific probability amplitude induced by a weighted graph G.

Definition 5.35 (ψ_G). For a graph $G = (V, E)$ with the weights $\{\omega_e \in [-\pi, \pi]\}_{e\in E}$ assigned to its edges and the weights $\{v_v \in [-\pi, \pi]\}_{v\in V}$ assigned to its vertices, we define ψ_G to be the probability amplitude given by

$$\psi_G := \left\langle 0^{|V|} \middle| \exp\left(-i\mathrm{H}_{\mathcal{X}_G}\right) \middle| 0^{|V|} \right\rangle .$$

We note that any X-program can be efficiently represented by a graph-induced X-program [SB09]. Moreover, X-programs are known to become universal for quantum computation under postselection [BJS10]. Therefore, any quantum amplitude can be expressed in the form of ψ_G. The output probability amplitudes of such a graph-induced X-program are proportional to Ising model partition functions with imaginary weights.

Proposition 5.36. *Let $G = (V, E)$ be a graph with the weights $\Omega = \{\omega_e \in [-\pi, \pi]\}_{e\in E}$ assigned to its edges and the weights $\Upsilon = \{v_v \in [-\pi, \pi]\}_{v\in V}$ assigned to its vertices, then,*

$$\psi_G = \frac{1}{2^{|V|}} \mathrm{Z}_{\mathrm{Ising}}(G; i\Omega, i\Upsilon).$$

Proof. By definition,

$$\begin{aligned}
\psi_G &= \left\langle 0^{|V|} \middle| \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} X_u X_v + i \sum_{v\in V} v_v X_v \right) \middle| 0^{|V|} \right\rangle \\
&= \left\langle +^{|V|} \middle| \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} Z_u Z_v + i \sum_{v\in V} v_v Z_v \right) \middle| +^{|V|} \right\rangle \\
&= \frac{1}{2^{|V|}} \sum_{x,y\in\{0,1\}^{|V|}} \langle y| \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} Z_u Z_v + i \sum_{v\in V} v_v Z_v \right) |x\rangle
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{2^{|V|}} \sum_{x \in \{0,1\}^{|V|}} \exp\left(i \sum_{\{u,v\} \in E} \omega_{\{u,v\}} (-1)^{x_u \oplus x_v} + i \sum_{v \in V} \upsilon_v (-1)^{x_v} \right) \\
&= \frac{1}{2^{|V|}} \sum_{z \in \{-1,+1\}^{|V|}} \exp\left(i \sum_{\{u,v\} \in E} \omega_{\{u,v\}} z_u z_v + i \sum_{v \in V} \upsilon_v z_v \right) \\
&= \frac{1}{2^{|V|}} Z_{\text{Ising}}(G; i\Omega, i\Upsilon).
\end{aligned}
$$

This completes the proof. ■

We now apply Corollary 5.29 to Proposition 5.36 to achieve a deterministic polynomial-time approximation scheme for computing ψ_G for all graphs of bounded maximum degree with weights absolutely bounded sufficiently close to zero.

Corollary 5.37. *Fix $\Delta \in \mathbb{Z}^+$ and $0 < \delta < \delta_{\Delta+1}$. There is a deterministic polynomial-time approximation scheme for the probability amplitude ψ_G for all graphs $G = (V, E)$ of maximum degree at most Δ with the edge weights $\{\omega_e \in [-\pi, \pi]\}_{e \in E}$ satisfying $|\omega_e| \leq 2 \arcsin(\delta/2)$ for all $e \in E$ and the vertex weights $\{\upsilon_v \in [-\pi, \pi]\}_{v \in V}$ satisfying $|\upsilon_v| \leq 2 \arcsin(\delta/2)$ for all $v \in V$.*

Proof. It follows from Corollary 5.29 and Proposition 5.36 that we have a deterministic polynomial-time approximation scheme for computing ψ_G for all graphs of maximum degree at most Δ with $\Omega = \{i\omega_e\}_{e \in E}$ and $\Upsilon = \{i\upsilon_v\}_{v \in V}$ in the closed polyregion $\mathcal{R}_G(\delta)$. For weights in the range $[-\pi, \pi]$, this is achieved when $|\omega_e| \leq 2 \arcsin(\delta/2)$ for all $e \in E$ and $|\upsilon_v| \leq 2 \arcsin(\delta/2)$ for all $v \in V$. This completes the proof. ■

It is known that approximating ψ_G up to a multiplicative factor for bounded degree graphs with arbitrary weights in $[-\pi, \pi]$ is **#P-hard** [FM17], and so it seems unlikely that Corollary 5.37 can be extended to hold in this case. We note that Corollary 5.37 holds for classes of graphs with treewidth growing as the square root of the number of vertices; for example, square lattices. For classes of graphs with logarithmic treewidth a deterministic polynomial-time algorithm is known [MS08].

5.7 Conclusion & Outlook

We have established a deterministic polynomial-time approximation scheme for the Ising model partition function with complex parameters on bounded degree graphs when the interactions and external fields are absolutely bounded by a constant depending on the maximum degree of the graph. Furthermore, we have proven that the partition function does not vanish for this class of Ising models. Finally, we have shown how our algorithm can be extended to approximate certain output probability amplitudes of quantum circuits.

This work gives rise to many interesting open problems, the most obvious of which is to sharpen the bounds of Corollary 5.29. One approach would be to improve Lemma 5.8, i.e., prove that the restricted multivariate graph homomorphism partition function does not vanish on a polydisc of a greater radius. It may also be possible to prove sharper bounds for specific graphs of interest. An alternative approach would be to use decay of correlation based arguments [Wei06, Sly10, SST14]. It is an important open problem to understand the relationship between the location of complex zeros, decay of correlations, and the computational complexity of a function.

The work of Liu, Sinclair, and Srivastava [LSS18] showed that in the case of no external field, the Ising model partition function has no complex zeros in a complex neighbourhood of the regime where the decay of correlation property holds. This implies a deterministic polynomial-time approximation scheme for the Ising model partition function in the decay of correlation regime based on the location of complex zeros.

Part II

Other Results

Chapter 6

Efficient Preparation of Fock States from Single-Photon Sources

In this chapter, we establish an efficient scheme for preparing Fock states with a high number of photons from a resource of single photons. Our scheme achieves this by iteratively and non-deterministically fusing Fock states together via a beamsplitter with a number-resolved photo-detector on one of the output modes. We show that by recycling the output Fock states that arise from failed attempts, we are able to produce high-photon Fock states in time polynomial in the number of photons. Our scheme requires single-photon sources, beamsplitters, number-resolved photo-detectors, and an optical quantum memory.

This chapter is based on joint work with Keith R. Motes, Jonathan P. Olson, Nicholas M. Studer, E. Annelise Bergeron, Alexei Gilchrist, Jonathan P. Dowling, Dominic W. Berry, and Peter P. Rohde, and has been published previously as "Efficient recycling strategies for preparing large Fock states from single-photon sources — Applications to quantum metrology" [MMO+16].

6.1 Introduction

Fock states are a fundamental resource in quantum information [KL10] with applications in communication, cryptography, metrology [Yur86, Yue86, Dow98, GC01, KD07], information processing [BM95, KLM01, PJF01], and quantum walks [SRK15]. While there has been several advances in producing Fock states with a low number of photons, producing Fock states with a high number of photons remains challenging. It is of course possible to produce Fock states with an arbitrary number of photons using non-deterministic linear optics, however, in this case, the success probability decays exponentially with the number of photons.

In this chapter, we establish an efficient scheme for preparing Fock states with a high number of photons from a resource of single photons. The idea of our scheme is to prepare high-photon Fock states by iteratively and non-deterministically fusing lower-photon Fock states together via a beamsplitter with a number-resolved photo-detector on one of the output modes. Crucially, we show that by recycling the output Fock states that arise from failed attempts, we are able to produce high-photon Fock states in time polynomial in the number of photons. Our scheme requires many of the same resources as universal linear-optical quantum computing [KLM01], including single-photon sources, beamsplitters, number-resolved photo-detectors, and an optical quantum memory.

Fock states with a high number of photons are an essential resource for preparing NOON states [KD07], which are known to be the optimal state for quantum enhanced metrology [Dow08], i.e., they achieve the Heisenberg limit of phase sensitivity. Therefore, our scheme is an important milestone for the realisation of optimal quantum enhanced metrology.

This chapter is structured as follows. In Section 6.2, we review Fock state preparation by spontaneous parametric down-conversion with postselection. In Section 6.3, we present a naïve approach to preparing Fock states with a high number of photons by a single-shot linear optics network with postselection — an approach that requires exponential time and resource states. In Section 6.4, we present an improved bootstrapped approach, where high-photon Fock states are prepared by iteratively fusing lower-photon Fock states together. We then introduce state recycling, which allows us to efficiently prepare high-photon Fock states. In Section 6.5 we discuss several different approaches to fusing Fock states. In Section 6.6, we discuss a scheme for reducing the number of photons in a Fock state. We present our simulation results in Section 6.7. Finally, we conclude in Section 6.8

6.2 Spontaneous Parametric Down-Conversion with Postselection

The most common approach to preparing high-photon Fock states is to employ spontaneous parametric down-conversion with postselection. *Spontaneous parametric down-conversion* (SPDC) is a non-linear optical process that converts photons of higher energy from a coherent pump into pairs of photons of lower energy across two modes, known as the *signal* and *idler* mode. The interaction Hamiltonian of an SPDC process is given by $\mathrm{H}_{\mathrm{SPDC}} = \xi \hat{a}_{\mathrm{pump}} \hat{a}^{\dagger}_{\mathrm{signal}} \hat{a}^{\dagger}_{\mathrm{idler}} + \xi^{*} \hat{a}^{\dagger}_{\mathrm{pump}} \hat{a}_{\mathrm{signal}} \hat{a}_{\mathrm{idler}}$, where ξ denotes the interaction strength. This process gives an output state of the form $|\psi\rangle_{\mathrm{SPDC}} = \sqrt{1-|\lambda|^2} \sum_{n=0}^{\infty} \lambda^n |n\rangle_{\mathrm{signal}} |n\rangle_{\mathrm{idler}}$. Notice that there is perfect photon-number correlation between the signal and idler modes, and so, by postselecting on detecting n photons in one mode, we obtain a Fock state of exactly n photons in the other (Fig. 6.1). This approach has been experimentally demonstrated for Fock states of up to three photons [CWSS13].

We shall now investigate the efficiency of preparing high-photon Fock states using this approach. Suppose that our goal is to produce a Fock state with at least d photons, then, the probability of success using this approach is given by

$$\mathrm{P}_{\mathrm{success}}(d) = \left(1-|\lambda|^2\right) \sum_{n=d}^{\infty} |\lambda|^{2n} = |\lambda|^{2d} \,,$$

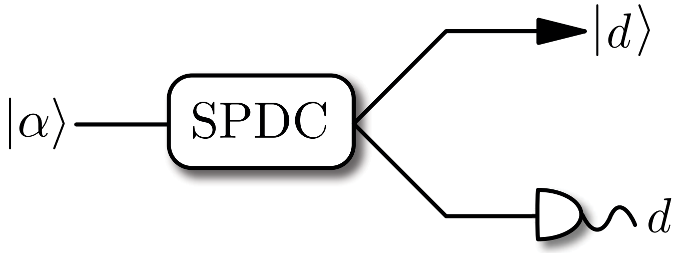


Figure 6.1: Preparation of an d-photon Fock state via a spontaneous parametric down-conversion process with postselection. A non-linear crystal is pumped with a coherent state $|\alpha\rangle$ giving a two-mode superposition with perfect photon-number correlations. Detecting d photons in the second mode guarantees an d-photon Fock state in the first.

which clearly decays exponentially with d and, therefore, typically takes an exponential number of trials to succeed. Furthermore, the experimental value of $|\lambda|^2$ is typically much less than one [FIJ+15], making this approach impractical for preparing Fock states with a high number of photons.

6.3 Single-Shot Linear Optics with Postselection

An alternative approach to preparing high-photon Fock states is via linear optics with postselection and a resource of single photons (Fig. 6.2). More precisely, consider an n-mode interferometer with exactly one photon in each input mode, i.e., a state of the form $|\psi_{\text{in}}\rangle = \left(\prod_{i=1}^{n} \hat{a}_i^\dagger\right) |0\rangle^{\otimes n}$, where $\hat{a}_i^\dagger$ is the photonic creation operator for the i^{th} mode. Now, apply a linear optical network described by the unitary map $\hat{U}\hat{a}_i^\dagger\hat{U}^\dagger \mapsto \sum_{j=1}^{n} U_{i,j}\hat{a}_j^\dagger$ on the photonic creation operators to obtain the output state $|\psi_{\text{out}}\rangle = \left(\prod_{i=1}^{n}\sum_{j=1}^{n} U_{i,j}\hat{a}_j^\dagger\right) |0\rangle^{\otimes n}$. Finally, by postselecting on n photons in the first mode, i.e., observing no photons in the other modes, we obtain the projected state $\left|\psi_{\text{projected}}\right\rangle = \sqrt{n!}\left(\prod_{i=1}^{n} U_{i,1}\hat{a}_1^\dagger\right)|n\rangle\,|0\rangle^{\otimes(n-1)}$. This succeeds with probability

$$\mathrm{P}_{\text{success}}(n) = n!\left|\prod_{i=1}^{n} U_{i,1}\right|^2,$$

which is maximised for a balanced interferometer, i.e., when $\left|U_{i,1}\right| = 1/\sqrt{n}$ for all i, and so, $\mathrm{P}_{\mathrm{success}}(n) \leq \frac{\sqrt{2\pi n}}{e^n} e^{\frac{1}{12n}}$. Therefore, this approach succeeds with probability inverse exponential in the number of photons.

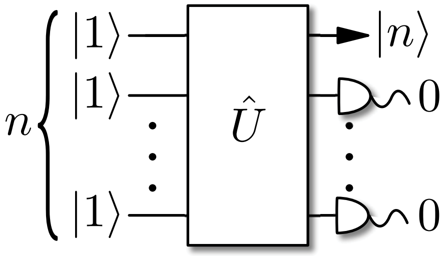


Figure 6.2: Preparation of an n-photon Fock state via single-shot linear optics with postselection. A photon is incident on each mode of an n-mode linear optical network. We postselect on obtaining n photons in the first mode by detecting no photons in any of the other modes.

6.4 Bootstrapped Linear Optics with Postselection

To improve upon the exponentially small success probability of the single-shot linear optics approach, we now consider a bootstrapped approach, where we prepare high-photon Fock states by iteratively and non-deterministically fusing lower-photon Fock states together via a beamsplitter with a number-resolved photo-detector on one of the output modes. More precisely, consider a beamsplitter with m photons in the first input mode and n photons in the second. Suppose that we detect s photons in the first output mode, then we obtain a Fock state of $m+n-s$ photons in the second output mode (Fig. 6.3). The input state to the fusion operation is the state $|\psi_{\mathrm{in}}\rangle = \frac{1}{\sqrt{m!n!}}(\hat{a}_1^\dagger)^m(\hat{a}_2^\dagger)^n$. Applying a beamsplitter with reflectivity η we obtain the output state

$$\begin{aligned}|\psi_{\mathrm{out}}\rangle &= \frac{1}{\sqrt{m!n!}}\left(\eta\hat{a}_1^\dagger + \sqrt{1-\eta^2}\hat{a}_2^\dagger\right)^m\left(\sqrt{1-\eta^2}\hat{a}_1^\dagger - \eta\hat{a}_2^\dagger\right)^n|0,0\rangle \\ &= \frac{1}{\sqrt{m!n!}}\sum_{j=0}^{m}\sum_{k=0}^{n}\binom{m}{j}\binom{n}{k}\eta^{n+j-k}\sqrt{1-\eta^2}^{m+k-j}(-1)^{n-k}(\hat{a}_1^\dagger)^{j+k}(\hat{a}_2^\dagger)^{m+n-j-k}|0,0\rangle\,.\end{aligned}$$

Now, suppose that we detect s photons in the first mode, then the projected state is

$$\left|\psi_{\text{projected}}\right\rangle = \sqrt{\frac{1}{m!n!}} \sum_{j=0}^{m} \binom{m}{j}\binom{n}{s-j} \eta^{n+2j-s} \sqrt{1-\eta^2}^{m+s-2j} (-1)^{n+j-s} (\hat{a}_1^\dagger)^s (\hat{a}_2^\dagger)^{m+n-s} \left|0,0\right\rangle .$$

The probability of detecting s photons is then

$$\mathrm{P}_{\text{fusion}}(s|m,n,\eta) = \frac{s!(m+n-s)!}{m!n!} \eta^{2(n-s)} \left(1-\eta^2\right)^{m+s} \left| \sum_{j=0}^{m} \binom{m}{j}\binom{n}{s-j} \left(\frac{\eta^2}{1-\eta^2}\right)^j \right|^2 .$$

This fusion operation will have been successful if the number of photons in the output state is greater than the number of photons in either of the inputs, i.e., when $s < m + n - \max(m, n)$. Therefore, the fusion operation succeeds with probability

$$\mathrm{P}_{\text{success}}(m,n,\eta) = \sum_{s=0}^{m+n-\max m,n-1} \mathrm{P}_{\text{fusion}}(s|m,n,\eta).$$

In the case that the Fock states are not recycled, we accept only the $s = 0$ outcome, and so we eliminate the sum and leave only the $s = 0$ term. The probability of success can be optimised over the choice of beamsplitter reflectivity η, and so, for each choice of m and n, we obtain the optimal beamsplitter reflectivity

$$\eta_{\text{optimal}}(m,n) := \underset{\eta}{\operatorname{argmax}} \left[\mathrm{P}_{\text{success}}(m,n,\eta)\right],$$

and the optimal success probability

$$\mathrm{P}_{\text{optimal}}(m,n) := \max_{\eta} \left[\mathrm{P}_{\text{success}}(m,n,\eta)\right].$$

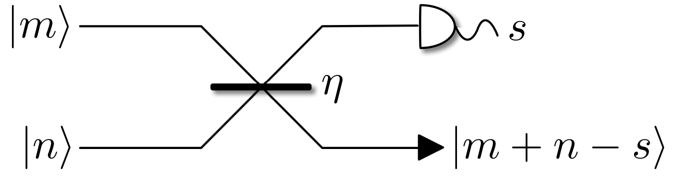


Figure 6.3: The Fock state fusion operation. Two Fock states $|m\rangle$ and $|n\rangle$ are incident on a beamsplitter with reflectivity η. By detecting s photons in the first output mode of the beamsplitter, we obtain the Fock state $|m + n - s\rangle$ at the second output mode.

6.5 Fusion

Numerically, we observe that the fusion success probability is maximised when fusing two Fock states of equal photon number. This suggests that the optimal strategy for performing the fusion operation is to always fuse together states of equal size, i.e., $m = n$. This is analogous with the cluster state literature [RB01, RBB03], where Rohde and Barrett [RB07] showed that bonding cluster states of equal size is optimal. In this instance, the only probabilities of interest are $\mathrm{P}_{\mathrm{fusion}}(s|m, m, \eta)$ and therefore the optimised success probability is $\mathrm{P}_{\mathrm{optimal}}(m, m) = 1/2$ for all m, which is very favourable for preparing high-photon Fock states. In the case that the Fock states are not recycled, we observe that the success probability is maximised only for fusing a Fock state with a single photon, i.e., $m = 1$ or $n = 1$.

Clearly, preparing a high-photon Fock state by iteratively fusing lower-photon Fock states will typically require time exponential in the number of photons. To improve on this approach, we borrow the concept of recycling from the cluster state literature [Nie04, BR05, GKE06, RB07] and the closely related parity-encoded scheme for linear optical quantum computation [GHR07]. We consider recycling the output state of a failed fusion operation and using as the input for future fusion attempts.

6.5.1 Generalised Fusion Protocol

To describe a generalised fusion protocol, we begin with the assumption that we can produce single-photon Fock states on demand. Now, suppose that we have a series of buckets (quantum memories) such that the n^{th} bucket contains only n-photon Fock states. Let $c_n(t)$ denote the number of n-photon Fock states in bucket n after the t^{th} fusion operation. By our first assumption, we set $c_1(0) = \infty$, and set all other buckets to be empty, i.e., $c_{i>1}(0) = 0$.

We then remove two Fock states from the buckets in accordance with our fusion strategy, and apply the fusion operation between them with beamsplitter reflectivity η. For Fock states drawn from bucket m and n, we obtain a Fock state with $m + n - s$

photons with probability $\mathrm{P}_{\text{fusion}}(s|m, n, \eta)$, which updates the buckets according to the transitions

$$\begin{aligned} c_m &\longrightarrow c_m - 1, \\ c_m &\longrightarrow c_m - 1, \\ c_{m+n-s} &\longrightarrow \begin{cases} c_{m+n-s} + 1 & \text{with recycling} \\ c_{m+n-s} + \delta_{s,0} & \text{without recycling.} \end{cases} \end{aligned}$$

Now, suppose that our goal is to prepare a resource of Fock states with photon number at least d. Then we are interested in the quantity $c_{\geq d}(t) = \sum_{j=d}^{\infty} c_j(t)$. The rate at which these state are prepared is then $r(d) = \lim_{t\to\infty} \frac{c_{\geq d}(t)}{t}$. We consider the $t \to \infty$ limit to establish the steady state flow dynamics of the states through the buckets.

6.5.2 Analytic Approximations

For certain schemes, we are able to establish analytic results that demonstrate an exponential improvement over the single-shot linear optics approach discussed in Section 6.3. Firstly, we consider a non-recycled scheme, where we attempt to construct a Fock state with 2^n photons. To achieve this, we fuse single-photon Fock states until we obtain 2-photon Fock states, we then fuse 2-photon Fock states until we obtain 4-photon Fock states, and so forth, until we obtain a 2^n-photon Fock state.

The success probability for this scheme is maximised for a balanced beamsplitter. To estimate the rate of producing 2^n-photon Fock states, we will estimate the average number of single-photons states needed to produced one 2^n-photon Fock state. The rate of preparing d-photon Fock states per fusion operation will then scale as the inverse of this number, since there can be no more than a factor of two between the number of single-photon states needed and the number of fusion operations.

To prove this, first consider the case where every fusion operation is successful. To prepare a 2^n-photon Fock state, we require 2^{n-1} fusion operations to fuse the single-photon Fock states, followed by 2^{n-2} fusion operations to fuse the 2-photon

Fock states, and so forth. This gives the total number of fusion operations required to be $2^n - 1$, which is one less than the number of single-photon Fock states. Now, when the success rate is decreased, the number of fusion operations can only be reduced for a given number of single photons. Therefore, the number of fusion operations cannot be higher than the number of single photons. Now, since the fusion operation is applied on all pairs of single photons, the number of fusion operations must be at least half of the number of single photons.

We shall now estimate the average number of single-photon Fock states required to prepare a 2^n-photon Fock state. The average number of attempts to fuse two 2^{n-1}-photon Fock states to prepare a 2^n-photon Fock state is given by $1/\mathrm{P}_{\mathrm{fusion}}(0|2^{n-1}, 2^{n-1}, 2^{-1/2})$. This gives the average number of 2^{n-1}-photon Fock states to be $2/\mathrm{P}_{\mathrm{fusion}}(0|2^{n-1}, 2^{n-1}, 2^{-1/2})$, since each fusion operation requires two states. Then, the average number of 2^{n-2}-photon Fock states required to prepare a 2^{n-1}-photon Fock state is given by $2/\mathrm{P}_{\mathrm{fusion}}(0|2^{n-2}, 2^{n-2}, 2^{-1/2})$. As a consequence, the expected number of 2^{n-2}-photon Fock states required to prepare a 2^n-photon Fock state is given by $4/[\mathrm{P}_{\mathrm{fusion}}(0|2^{n-2}, 2^{n-2}, 2^{-1/2}) \cdot \mathrm{P}_{\mathrm{fusion}}(0|2^{n-1}, 2^{n-1}, 2^{-1/2})]$. It then follows that the average number of single photons required to prepare a 2^n-photon Fock state is given by

$$2^{n-1} \prod_{k=1}^{n-1} \frac{1}{\mathrm{P}_{\mathrm{fusion}}(0|2^k, 2^k, 2^{-1/2})}.$$

To estimate this quantity, we observe that

$$\begin{aligned}\mathrm{P}_{\mathrm{fusion}}(0|d, d, 2^{-1/2}) &= \frac{(2d)!}{(2^{2d})(d!)^2} \\ &\sim \frac{1}{\sqrt{\pi d}},\end{aligned}$$

where the approximation follows from Stirling's formula. Then the average number of single photons required to prepare a 2^n-photon Fock state scales as

$$2^{n-1} \prod_{k=1}^{n-1} \sqrt{\pi 2^k} = \pi^{\frac{1}{2}(n-1)} 2^{\frac{1}{4}(n-1)(n+4)}.$$

The rate of preparing 2^n-photon Fock states $r(2^n)$ then scales as the inverse of this expression, that is,

$$r(2^n) \sim \frac{1}{\pi^{\frac{1}{2}(n-1)} 2^{\frac{1}{4}(n-1)(n+4)}}.$$

Therefore, we observe an exponential improvement in the preparation rate over the single-shot linear optics approach.

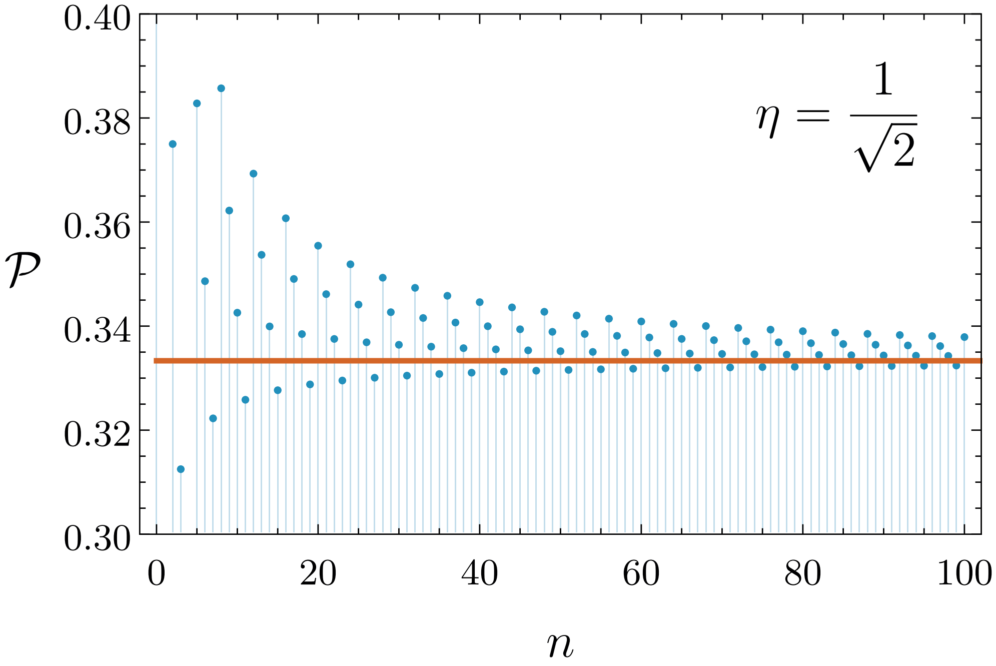


Figure 6.4: Probability $\mathcal{P}$ of a successful fusion operation of two n-photon Fock states, where we require that no more than $\lfloor n/2 \rfloor$ photons are lost. We observe that the probability of success approaches $\approx 1/3$ in the limit $n \longrightarrow \infty$.

To improve upon this further, we shall consider the case with limited recycling, that is, rather than requiring that no photons are lost at each fusion operation, we instead require that no more than $\lfloor n/2 \rfloor$ photons are lost when fusing two n-photon Fock states. The probability of a successful fusion operation is then given by

$$\mathcal{P}(n) = \sum_{s=0}^{\lfloor n/2 \rfloor} \mathrm{P}_{\text{fusion}}(s|n, n, 2^{-1/2}).$$

We numerically observe that the probability of success approaches $\approx 1/3$ in the limit $n \longrightarrow \infty$, as shown in Fig. 6.4. Equivalently, we require that a successful fusion gives a photon number of at least $\lceil 3n/2 \rceil$. If the photon number is higher than $\lceil 3n/2 \rceil$, then we can reduce the photon number with the Fock state reduction scheme described in Section 6.6. Now, to obtain a 2^n-photon Fock state we require a number of levels of fusion operations that scales as $n/\log_2(3/2)$. Taking the success probability to be $1/3$, we obtain that the average number of single-photon Fock states required to prepare a 2^n-photon Fock state scales as $6^{n/\log_2(3/2)}$. This corresponds to a preparation rate that scales as $1/6^{n/\log_2(3/2)}$, which is strictly polynomial in the number of photons.

6.5.3 Fusion Schemes

An analytic bound for more advanced recycling schemes is non-trivial, and so, we instead simulate these schemes as a classical Markov process between the buckets with probabilities given by $\mathrm{P}_{\mathrm{fusion}}(s|m, n, \eta)$ and the bucket transition rules. While we numerically observe that the balanced strategy may be optimal, we consider the following strategies.

(1) **Balanced:** Fuse the two highest available Fock states of equal size.

(2) **Modest:** Always fuse the highest available Fock state with a single photon.

(3) **Random:** Two Fock states are fused uniformly at random from buckets with available states.

(4) **Frugal:** The same as balanced, except that we do not attempt to fuse two equally sized states if $m = n > \lfloor d'/2 \rfloor$, where $d' \geq d$, and instead, we attempt to fuse available states such that $d \leq m + n \leq d'$. The intuition behind this is that because high-photon Fock states are costly to prepare, it is wasteful to fuse two states with total photon number in excess of the target d.

The optimisation technique for the frugal strategy differs from the other strategies in that if the total input photon number $m + n \geq d$, then we optimise η to maximise the probability of obtaining at least d photons, i.e., we maximise $\sum_{s=0}^{m+n-d} \mathrm{P}_{\mathrm{fusion}}(s|m, n, \eta)$.

Otherwise, if $m + n < d$, then we optimise η to maximise the photon number with an increased weighting for obtaining a higher photon number, i.e., we optimise

$$\sum_{s=0}^{m+n-\max(m,n)} [m + n - s - \max(m, n)] \cdot \mathrm{P}_{\text{fusion}}(s|m, n, \eta).$$

6.5.4 Hybrid Schemes

Previously, we have considered preparation schemes where a resource of single-photon Fock states is freely available. This is appropriate when our sources produce single photons, however, emerging technologies, such as quantum dot sources, have the ability to directly produce Fock states with a low number of photons. Intuitively, by starting with a resource of Fock states with higher photon number, we could further improve preparation rates. Suppose that we have a free resource of n-photon Fock states, then our framework easily accommodates for this by setting $c_n = \infty$.

6.6 Fock State Reduction

In our analysis, we have defined our state preparation rate $r(d)$ to be the rate at which Fock states of at least d photons are prepared. For many protocols that require high-photon Fock states, this is appropriate. However, other applications may require Fock states with exactly d photons. For these applications, we require a protocol for reducing the photon number of a Fock state.

This can be efficiently implemented using linear optics with postselection. We simply input the prepared Fock state into a beamsplitter with low reflectivity and vacuum at the other input mode. Due to the low reflectivity of the beamsplitter, with high probability no photons will be detected in the reflected mode. However, occasionally a single photon will be detected and with higher order probability more than one photon will be detected. By choosing a sufficiently small reflectivity, the higher order probabilities can be made arbitrarily small, so that, with probability close to one at most one photon is detected. When a single photon is detected we have reduced the photon number of the input Fock state by one. We repeat this procedure until the desired number of photons has be subtracted, giving the desired photon-number Fock state. Note that this protocol typically requires $O(s)$ beamsplitter operations to reduce the Fock state by s photons.

6.7 Results

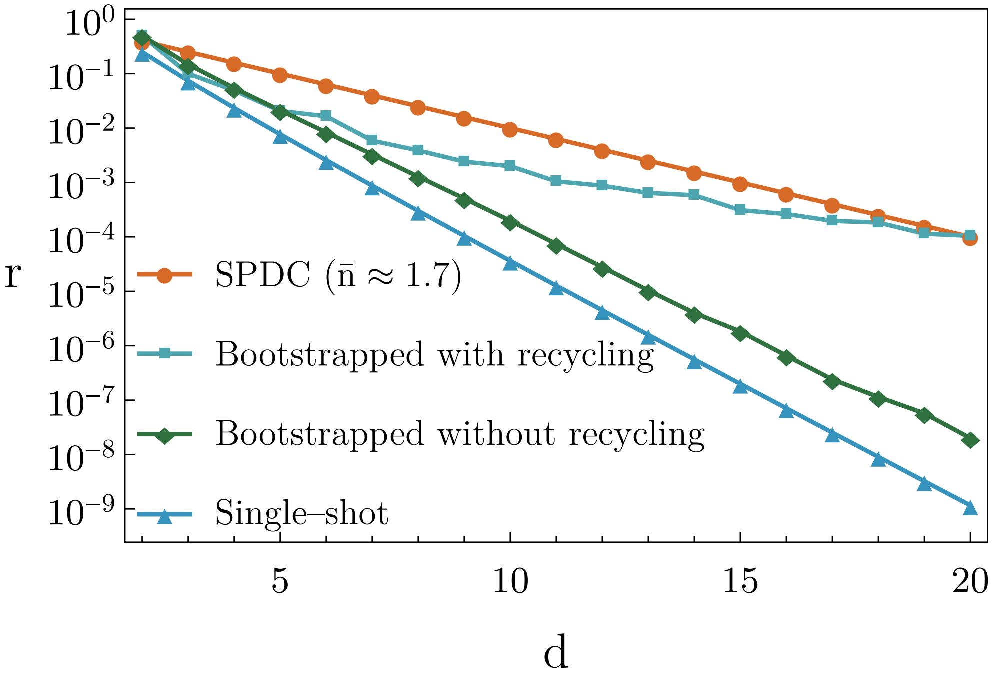


Figure 6.5: Rate of preparation r of Fock states with at least d photons for the recycled and non-recycled bootstrapped protocols, the single-shot linear optics protocol, and the SPDC protocol. We observe that, with the exception of the recycled bootstrapped protocol, these protocols exhibit an exponential decay in the preparation rate with d. It is evident that the recycled bootstrapped protocol gives an exponential improvement over the non-recycled or single-shot protocols. The exponential decays of SPDC protocol depends on the mean photon number $\bar{n}$ of the source, which is chosen such that the SPDC and recycled bootstrapped protocol have approximately the same 20-photon preparation rate.

In Fig. 6.5, we plot the rate $r(d)$ of preparing Fock states with at least d photons for the SPDC protocol (Section 6.2), the single-shot linear optics protocol (Section 6.3), and the recycled and non-recycled bootstrapped protocols (Section 6.4). In both bootstrapped protocols we employ the balanced strategy and assume an infinite resource of single photons. The cost of performing these bootstrapped protocols is measured by the number of fusion (beamsplitter) operations required.

In the case of the single-shot linear optics protocol, the rate of preparing states of at least d photons in terms of the number of interferometer operations is given by $d!/d^d$. To convert this into a cost in terms of the number of beamsplitter operations, we observe that a d-mode interferometer can be most easily constructed from d beamsplitters in a linear array. Therefore, the rate of preparing Fock states with at least d photons in terms of the number of beamsplitter operations is given by $d!/d^{d+1}$.

In the case of the SPDC protocol, there is no natural measure of the resource requirements in terms of beamsplitter operations. Instead, we shall measure the rate of preparing Fock states by the number of repetitions of the SPDC source, which is given by the pump repetition rate. Note that this is a different measure of the resource requirements than that used for other protocols. In Fig. 6.5, we have chosen the mean photon number $\bar{n}$ of the SPDC source to be such that the SPDC and recycled bootstrapped protocol have approximately the same 20-photon preparation rate. This occurs when $\bar{n} \approx 1.7$, which is far beyond what is typically achieved in present-day experiments.

It is clear from Fig. 6.5 that in all cases except the recycled bootstrapped scheme, we have an exponential decay in the preparation rate with d. Furthermore, the recycled bootstrapped scheme exhibits an exponential improvement in the preparation rate of these schemes. For example, the recycled bootstrapped protocol improves the 20-photon preparation rate by a factor of $\approx 10^4$ over the non-recycled bootstrapped approach.

In Fig. 6.6, we present the preparation rate for the fusion strategies introduced in Section 6.5. It is clear that the preparation rate exhibits a polynomial decay with the number of photons d for the frugal and balanced strategies, and an exponential decay for the random and modest strategies. This provides an exponential improvement in

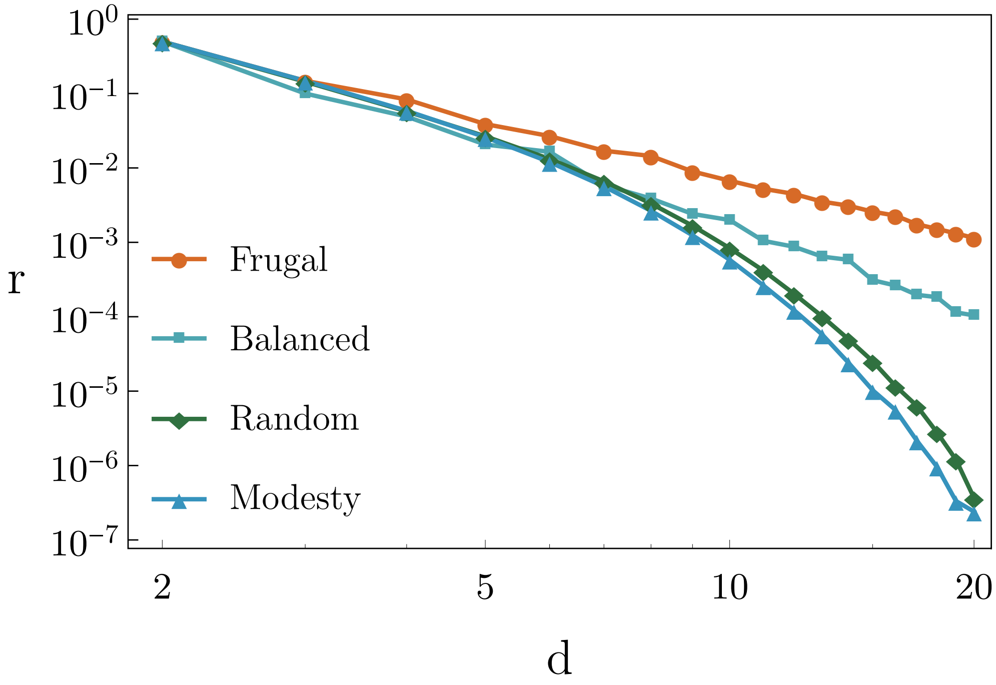


Figure 6.6: Comparison of the rate r of preparing Fock states with at least d photons for the frugal, balanced, random, and modest recycled fusion strategies. We observe that the frugal and balanced strategies exhibit a polynomial decay of preparation rate with d, whereas the random and modest strategies exhibit an exponential decay of the preparation rate with d.

the rate of state preparation for the recycled frugal and balanced strategies. The frugal strategy scales as $\sim 1/d^{2.8}$, while the balanced strategy scales as $\sim 1/d^{3.7}$.

In Fig. 6.7, we present the performance of hybrid schemes employing the frugal fusion strategy beginning with resource states of different photon number. We only include the results of the frugal strategy, since this hybrid scheme was observed to exhibit the most improved preparation rates.

6.8 Conclusion & Outlook

We have established a scheme for non-deterministically preparing Fock states with a high number of photons from a resource of single-photon Fock states by using linear optics with postselection to iteratively fuse low photon states into higher ones.

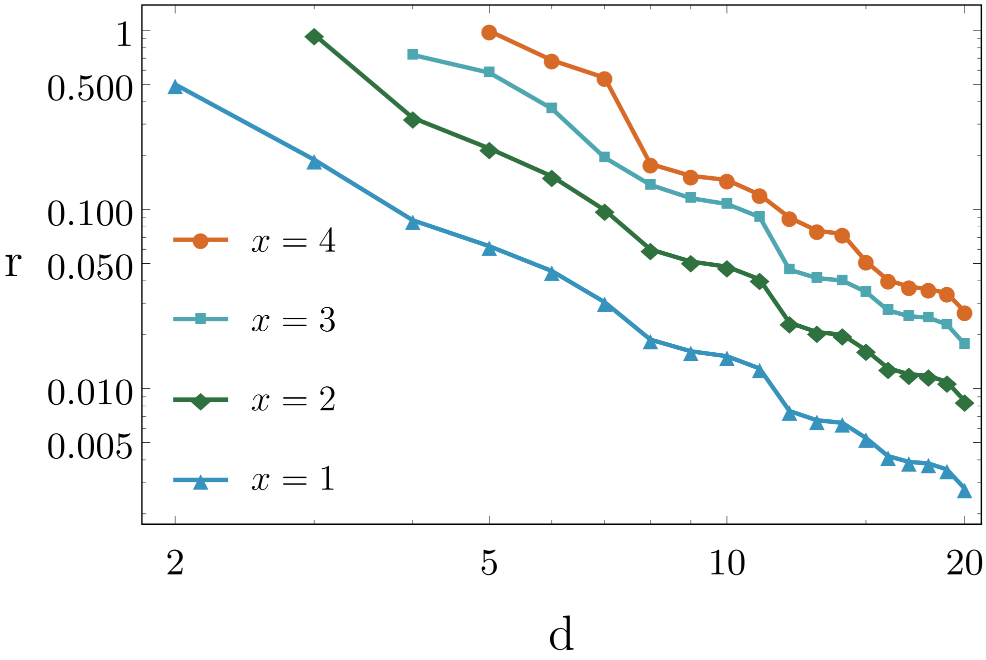


Figure 6.7: Rate of preparation r of Fock states with at least d photons for hybrid schemes employing the recycled frugal fusion strategy, where we begin with resource states of different photon numbers x, i.e., $c_x = \infty$. We observe an increased efficiency of the preparation rate with higher photon resource states.

We observe that by recycling Fock states we achieve an exponential improvement in the state preparation rate over conventional schemes. This allows us to efficiently prepare Fock states with a high number of photons. Our scheme requires many of the same resources as universal linear-optical quantum computing, including single-photon sources, beamsplitters, number-resolved photo-detectors, and an optical quantum memory.

In our analysis, we have assumed that there are no experimental imperfections, i.e., the resource of single photons are perfect Fock states, and the beamsplitters, number-resolved photo-detectors, and quantum memory have perfect efficiency. In practice, any experimental implementation will exhibit inefficiencies, which would result in an output state mixed in the photon-number basis, and so, future experimental implementations would need to take this into account.

Furthermore, we have assumed that the photons in the protocol have perfect mode overlap. In practice, photons will exhibit some extent of distinguishability, resulting in reduced visibility of the Hong-Ou-Mandel effect. This will alter the photon-number distribution at the the output modes of the fusion operation, causing a change in the state preparation rate and photon distinguishability. This can be easily modelled using the mode-operator formalism [RR05, RR06, RRM06, RMS07].

Chapter 7

Conclusion & Outlook

In this thesis, we studied the relationship between quantum computation and combinatorial structures. In particular, we have studied to what extent the classical complexity of combinatorial structures can improve our understanding of the complexity of quantum computation. We have made some partial progress towards resolving this, however, this fundamentally remains an open problem.

The results of Chapter 4 provide strong evidence that simulating random quantum computations is intractable for classical computers. Specifically, we showed that under the assumption that **(1)** the Polynomial Hierarchy does not collapse and **(2)** the average-case complexity of multiplicative-error approximations of the Jones polynomial matches the worst-case complexity, then there is no efficient classical algorithm for approximately sampling from the output probability distribution of random quantum computations. However, resolving this average-case complexity conjecture seems beyond the reach of existing techniques.

In Chapter 5, we considered the contrary case, that is, what is the strongest statement that can be made about efficiently simulating quantum computation by classical computation? We established a deterministic polynomial-time approximation scheme for the Ising model partition function when the interactions and external fields are absolutely bounded close to zero.

The results of this chapter add to existing results on approximating Ising model partition functions for restricted classes of graphs and parameters. However, it is clear that by taking all of these result into account, there is still a a large gap between the combinatorial structures that we can efficiently classically simulate and those required for an average-case hardness result. Furthermore, the behaviour of combinatorial structures where an efficient approximation scheme is known, is very different from that of a typical instance. It remains an interesting open problem to identify a complexity transition, at which, the structures that arise in this thesis transition from having an efficient approximation scheme to being **#P-hard**.

In Part II of this thesis, we established an efficient scheme for preparing Fock states with a high number of photons from a resource of single photons, which a fundamental resource in many quantum information protocols.

Bibliography

[AA11a] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, pages 333–342. ACM, 2011. doi:10.1145/1993636.1993682.

[AA11b] Dorit Aharonov and Itai Arad. The BQP-hardness of approximating the Jones polynomial. *New Journal of Physics*, 13(3):035019, 2011. doi:10.1088/1367-2630/13/3/035019.

[AAEL07] Dorit Aharonov, Itai Arad, Elad Eban, and Zeph Landau. Polynomial quantum algorithms for additive approximations of the Potts model and other points of the Tutte plane. *arXiv preprint quant-ph/0702008*, 2007, arXiv:quant-ph/0702008.

[AC16] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. *arXiv preprint 1612.05903*, 2016, arXiv:1612.05903.

[AJKR10] Gorjan Alagic, Stephen P Jordan, Robert König, and Ben W Reichardt. Estimating Turaev-Viro three-manifold invariants is universal for quantum computation. *Physical Review A*, 82(4):040302, 2010. doi:10.1103/physreva.82.040302.

[AJL09] Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the Jones polynomial. *Algorithmica*, 55(3):395–421, 2009. doi:10.1007/s00453-008-9168-0.

[Ale23] James Wadell Alexander. A lemma on systems of knotted curves. *Proceedings of the National Academy of Sciences*, 9(3):93–95, 1923. doi:10.1073/pnas.9.3.93.

[Bar15] Alexander Barvinok. Computing the partition function for cliques in a graph. *Theory of Computing*, 11(13):339–355, 2015. doi:10.4086/toc.2015.v011a013.

[Bar16a] Alexander Barvinok. *Combinatorics and complexity of partition functions*, volume 274. Springer, 2016. doi:10.1007/978-3-319-51829-9.

[Bar16b] Alexander Barvinok. Computing the permanent of (some) complex matrices. *Foundations of Computational Mathematics*, 16(2):329–342, 2016. doi:10.1007/s10208-014-9243-7.

[BBBV97] Charles H Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. doi:10.1137/s0097539796300933.

[BCKL13] Christian Borgs, Jennifer Chayes, Jeff Kahn, and László Lovász. Left and right convergence of graphs with bounded degree. *Random Structures & Algorithms*, 42(1):1–28, 2013. doi:10.1002/rsa.20414.

[BF12] Winton Brown and Omar Fawzi. Scrambling speed of random quantum circuits. *arXiv preprint 1210.6644*, 2012, arXiv:1210.6644.

[BFK17] Adam Bouland, Joseph F Fitzsimons, and Dax Enshan Koh. Quantum advantage from conjugated Clifford circuits. *arXiv preprint 1709.01805*, 2017, arXiv:1709.01805.

[BFLW05] Magnus Bordewich, Michael Freedman, László Lovász, and Dominic Welsh. Approximate counting and quantum computation. *Combinatorics, Probability and Computing*, 14(5-6):737–754, 2005. doi:10.1017/s0963548305007005.

[BFNV18] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. Quantum supremacy and the complexity of random circuit sampling. *arXiv preprint 1803.04402*, 2018, arXiv:1803.04402.

[BG05] Andrei Bulatov and Martin Grohe. The complexity of partition functions. *Theoretical Computer Science*, 348(2-3):148–186, 2005. doi:10.1016/j.tcs.2005.09.011.

[BH13] Fernando GSL Brandão and Michał Horodecki. Exponential quantum speed-ups are generic. *Quantum Information and Computation*, 13(11-12):901–924, 2013.

[BHH16] Fernando GSL Brandão, Aram W Harrow, and Michał Horodecki. Local random quantum circuits are approximate polynomial-designs. *Communications in Mathematical Physics*, 346(2):397–434, 2016. doi:10.1007/s00220-016-2706-8.

[BIS$^+$18] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. Characterizing quantum supremacy in near-term devices. *Nature Physics*, 14(6):595, 2018. doi:10.1038/s41567-018-0124-x.

[BJS10] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, page rspa20100301. The Royal Society, 2010. doi:10.1098/rspa.2010.0301.

[BLP93] Joe P Buhler, Hendrik W Lenstra, and Carl Pomerance. Factoring integers with the number field sieve. In *The development of the number field sieve*, pages 50–94. Springer, 1993.

[BM95] Samuel L Braunstein and A Mann. Measurement of the bell operator and quantum teleportation. *Physical Review A*, 51(3):R1727, 1995. doi:10.1103/physreva.51.r1727.

[BMS16] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical Review Letters*, 117(8):080501, 2016. doi:10.1103/physrevlett.117.080501.

[BMS17] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Achieving quantum supremacy with sparse and noisy commuting quantum computations. *Quantum*, 1:8, 2017. doi:10.22331/q-2017-04-25-8.

[BR05] Daniel E Browne and Terry Rudolph. Resource-efficient linear optical quantum computation. *Physical Review Letters*, 95(1):010501, 2005. doi:10.1103/physrevlett.95.010501.

[BS17] Alexander Barvinok and Pablo Soberón. Computing the partition function for graph homomorphisms. *Combinatorica*, 37(4):633–650, 2017. doi:10.1007/s00493-016-3357-2.

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/s0097539796300921.

[BVHS$^+$18] Juan Bermejo-Vega, Dominik Hangleiter, Martin Schwarz, Robert Raussendorf, and Jens Eisert. Architectures for quantum simulation showing a quantum speedup. *Physical Review X*, 8(2):021010, 2018. doi:10.1103/physrevx.8.021010.

[CCL10] Jin-Yi Cai, Xi Chen, and Pinyan Lu. Graph homomorphisms with complex values: A dichotomy theorem. In *International Colloquium on Automata, Languages, and Programming*, pages 275–286. Springer, 2010. doi:10.1007/978-3-642-14165-2_24.

[Coo71] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158. ACM, 1971. doi:10.1145/800157.805047.

[Cra69] Henry H Crapo. The Tutte polynomial. *Aequationes Mathematicae*, 3(3):211–229, 1969. doi:10.1007/bf01817442.

[CWSS13] Merlin Cooper, Laura J Wright, Christoph Söller, and Brian J Smith. Experimental generation of multi-photon Fock states. *Optics Express*, 21(5):5309–5317, 2013. doi:10.1364/oe.21.005309.

[DDVM11] Gemma De las Cuevas, Wolfgang Dür, Maarten Van den Nest, and Miguel A Martin-Delgado. Quantum algorithms for classical lattice models. *New Journal of Physics*, 13(9):093021, 2011. doi:10.1088/1367-2630/13/9/093021.

[DG00] Martin Dyer and Catherine Greenhill. The complexity of counting graph homomorphisms. *Random Structures and Algorithms*, 17(3-4):260–289, 2000. doi:10.1002/1098-2418(200010/12)17:3/4<260::AID-RSA5>3.0.CO;2-W.

[Dow98] Jonathan P Dowling. Correlated input-port, matter-wave interferometer: Quantum-noise limits to the atom-laser gyroscope. *Physical Review A*, 57(6):4736, 1998. doi:10.1103/physreva.57.4736.

[Dow08] Jonathan P Dowling. Quantum optical metrology–the lowdown on high-N00N states. *Contemporary Physics*, 49(2):125–143, 2008. doi:10.1080/00107510802091298.

[EM17] Lior Eldar and Saeed Mehraban. Approximating the permanent of a random matrix with vanishing mean. *arXiv preprint 1711.09457*, 2017, arXiv:1711.09457.

[Fey82] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982. doi:10.1007/bf02650179.

[FIJ$^+$15] Martin A Finger, Timur Sh Iskhakov, Nicolas Y Joly, Maria V Chekhova, and Philip St J Russell. Raman-free, noble-gas-filled photonic-crystal fiber source for ultrafast, very bright twin-beam squeezed vacuum. *Physical Review Letters*, 115(14):143602, 2015. doi:10.1103/physrevlett.115.143602.

[FKW02] Michael H Freedman, Alexei Kitaev, and Zhenghan Wang. Simulation of topological field theories by quantum computers. *Communications in Mathematical Physics*, 227(3):587–603, 2002. doi:10.1007/s002200200635.

[FLW02] Michael H Freedman, Michael Larsen, and Zhenghan Wang. A modular functor which is universal for quantum computation. *Communications in Mathematical Physics*, 227(3):605–622, 2002. doi:10.1007/s002200200645.

[FM17] Keisuke Fujii and Tomoyuki Morimae. Commuting quantum circuits and complexity of Ising partition functions. *New Journal of Physics*, 19(3):033003, 2017. doi:10.1088/1367-2630/aa5fdb.

[FR98] Lance Fortnow and John Rogers. Complexity limitations on quantum computation. In *Proceedings of the 13th IEEE Conference on Computational Complexity*, pages 202–209. IEEE, 1998. doi:10.1109/ccc.1998.694606.

[GC01] Christopher C Gerry and RA Campos. Generation of maximally entangled photonic states with a quantum-optical Fredkin gate. *Physical Review A*, 64(6):063814, 2001. doi:10.1103/physreva.64.063814.

[GG17] Leslie Ann Goldberg and Heng Guo. The complexity of approximating complex-valued Ising and Tutte partition functions. *Computational Complexity*, 26(4):765–833, 2017. doi:10.1007/s00037-017-0162-2.

[GGJT10] Leslie Ann Goldberg, Martin Grohe, Mark Jerrum, and Marc Thurley. A complexity dichotomy for partition functions with mixed signs. *SIAM Journal on Computing*, 39(7):3336–3402, 2010. doi:10.1137/090757496.

[GHR07] Alexei Gilchrist, AJF Hayes, and TC Ralph. Efficient parity-encoded optical quantum computing. *Physical Review A*, 75(5):052328, 2007. doi:10.1103/physreva.75.052328.

[Gil77] John Gill. Computational complexity of probabilistic Turing machines. *SIAM Journal on Computing*, 6(4):675–695, 1977. doi:10.1137/0206049.

[GKE06] David Gross, Konrad Kieling, and Jens Eisert. Potential and limits to cluster-state quantum computing using probabilistic gates. *Physical Review A*, 74(4):042343, 2006. doi:10.1103/physreva.74.042343.

[Gro96] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219. ACM, 1996. doi:10.1145/237814.237866.

[GŠV16] Andreas Galanis, Daniel Štefankovič, and Eric Vigoda. Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models. *Combinatorics, Probability and Computing*, 25(4):500–559, 2016. doi:10.1017/s0963548315000401.

[GWD17] Xun Gao, Sheng-Tao Wang, and L-M Duan. Quantum supremacy for simulating a translation-invariant Ising spin model. *Physical Review Letters*, 118(4):040502, 2017. doi:10.1103/physrevlett.118.040502.

[HBVSE18] Dominik Hangleiter, Juan Bermejo-Vega, Martin Schwarz, and Jens Eisert. Anticoncentration theorems for schemes showing a quantum speedup. *Quantum*, 2:65, 2018. doi:10.22331/q-2018-05-22-65.

[HL09] Aram W Harrow and Richard A Low. Random quantum circuits are approximate 2-designs. *Communications in Mathematical Physics*, 291(1):257–302, 2009. doi:10.1007/s00220-009-0873-6.

[HM17] Aram W. Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549:203–209, 2017. doi:10.1038/nature23458.

[HN90] Pavol Hell and Jaroslav Nešetřil. On the complexity of h-coloring. *Journal of Combinatorial Theory, Series B*, 48(1):92–110, 1990. doi:10.1016/0095-8956(90)90132-j.

[HN04] Pavol Hell and Jaroslav Nešetřil. *Graphs and homomorphisms*. Oxford University Press, 2004. doi:10.1093/acprof:oso/9780198528173.001.0001.

[ICBB14] S Iblisdir, M Cirio, O Boada, and GK Brennen. Low depth quantum circuits for Ising models. *Annals of Physics*, 340(1):205–251, 2014. doi:10.1016/j.aop.2013.11.001.

[Jon83] Vaughan FR Jones. Braid groups, Hecke algebras and type II1 factors. *Geometric Methods in Operator Algebras (Kyoto, 1983)*, 123:242–273, 1983.

[Jon85] Vaughan FR Jones. A polynomial invariant for knots via von Neumann algebras. *Bulletin of the American Mathematical Society*, 12(1):103–111, 1985. doi:10.1090/s0273-0979-1985-15304-2.

[JS93] Mark Jerrum and Alistair Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on Computing*, 22(5):1087–1116, 1993. doi:10.1137/0222066.

[JVW90] François Jaeger, Dirk L Vertigan, and Dominic JA Welsh. On the computational complexity of the Jones and Tutte polynomials. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 108, pages 35–53. Cambridge Univ Press, 1990. doi:10.1017/s0305004100068936.

[KD07] Kishore T Kapale and Jonathan P Dowling. Bootstrapping approach for generating maximally path-entangled photon states. *Physical Review Letters*, 99(5):053602, 2007. doi:10.1103/physrevlett.99.053602.

[KL10] Pieter Kok and Brendon W Lovett. *Introduction to optical quantum information processing*. Cambridge university press, 2010. doi:10.1017/cbo9781139193658.

[KLM01] Emanuel Knill, Raymond Laflamme, and Gerald J Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46, 2001. doi:10.1038/35051009.

[Kni95] Emanuel Knill. Approximation by quantum circuits. *arXiv preprint quant-ph/9508006*, 1995, arXiv:quant-ph/9508006.

[KSV02] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation*, volume 47. American Mathematical Society Providence, 2002. doi:10.1090/gsm/047.

[Kup09] Greg Kuperberg. How hard is it to approximate the Jones polynomial? *arXiv preprint 0908.0512*, 2009, arXiv:0908.0512.

[LBR17] AP Lund, Michael J Bremner, and TC Ralph. Quantum sampling problems, BosonSampling and quantum supremacy. *NPJ Quantum Information*, 3:1, 2017. doi:10.1038/s41534-017-0018-2.

[Lev73] Leonid Anatolevich Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

[LSS17] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. The Ising partition function: Zeros and deterministic approximation. *arXiv preprint 1704.06493*, 2017, arXiv:1704.06493.

[LSS18] Jingcheng Liu, Alistair Sinclair, and Piyush Srivastava. Fisher zeros and correlation decay in the Ising model. *arXiv preprint 1807.06577*, 2018, arXiv:1807.06577.

[LY52] Tsung-Dao Lee and Chen-Ning Yang. Statistical theory of equations of state and phase transitions. II. Lattice gas and Ising model. *Physical Review*, 87(3):410, 1952. doi:10.1103/physrev.87.410.

[MB17] Ryan L Mann and Michael J Bremner. On the complexity of random quantum computations and the Jones polynomial. *arXiv preprint 1711.00686*, 2017, arXiv:1711.00686.

[MB18] Ryan L Mann and Michael J Bremner. Approximation algorithms for complex-valued Ising models on bounded degree graphs. *arXiv preprint 1806.11282*, 2018, arXiv:1806.11282.

[MMO$^+$16] Keith R Motes, Ryan L Mann, Jonathan P Olson, Nicholas M Studer, E Annelise Bergeron, Alexei Gilchrist, Jonathan P Dowling, Dominic W Berry, and Peter P Rohde. Efficient recycling strategies for preparing large Fock states from single-photon sources: Applications to quantum metrology. *Physical Review A*, 94(1):012344, 2016. doi:10.1103/physreva.94.012344.

[Mon16] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2:15023, 2016. doi:10.1038/npjqi.2015.23.

[MS08] Igor L Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008. doi:10.1137/050644756.

[MSM17] Jacob Miller, Stephen Sanders, and Akimasa Miyake. Quantum supremacy in constant-time measurement-based computation: A unified architecture for sampling and verification. *arXiv preprint 1703.11002*, 2017, arXiv:1703.11002.

[Nie04] Michael A Nielsen. Optical quantum computation using cluster states. *Physical Review Letters*, 93(4):040503, 2004. doi:10.1103/physrevlett.93.040503.

[Pap03] Christos H Papadimitriou. *Computational complexity*. John Wiley and Sons Ltd., 2003.

[PJF01] TB Pittman, BC Jacobs, and JD Franson. Probabilistic quantum logic operations using polarizing beam splitters. *Physical Review A*, 64(6):062311, 2001. doi:10.1103/physreva.64.062311.

[PR17] Viresh Patel and Guus Regts. Deterministic polynomial-time approximation algorithms for partition functions and graph polynomials. *SIAM Journal on Computing*, 46(6):1893–1919, 2017. doi:10.1137/16m1101003.

[PR18] Han Peters and Guus Regts. Location of zeros for the partition function of the ising model on bounded degree graphs. *arXiv preprint 1810.01699*, 2018, arXiv:1810.01699.

[RB01] Robert Raussendorf and Hans J Briegel. A one-way quantum computer. *Physical Review Letters*, 86(22):5188, 2001. doi:10.1103/physrevlett.86.5188.

[RB07] Peter P Rohde and Sean D Barrett. Strategies for the preparation of large cluster states using non-deterministic gates. *New Journal of Physics*, 9(6):198, 2007. doi:10.1088/1367-2630/9/6/198.

[RBB03] Robert Raussendorf, Daniel E Browne, and Hans J Briegel. Measurement-based quantum computation on cluster states. *Physical Review A*, 68(2):022312, 2003. doi:10.1103/physrevlett.108.230508.

[RMS07] Peter P Rohde, Wolfgang Mauerer, and Christine Silberhorn. Spectral structure and decompositions of optical states, and their applications. *New Journal of Physics*, 9(4):91, 2007. doi:10.1088/1367-2630/9/4/091.

[RR05] Peter P Rohde and Timothy C Ralph. Frequency and temporal effects in linear optical quantum computing. *Physical Review A*, 71(3):032320, 2005. doi:10.1103/physreva.71.032320.

[RR06] Peter P Rohde and Timothy C Ralph. Error models for mode mismatch in linear optics quantum computing. *Physical Review A*, 73(6):062312, 2006. doi:10.1103/physreva.73.062312.

[RRM06] Peter P Rohde, Timothy C Ralph, and William J Munro. Practical limitations in optical entanglement purification. *Physical Review A*, 73(3):030301, 2006. doi:10.1103/physreva.73.030301.

[RS09] Aidan Roy and Andrew James Scott. Unitary designs and codes. *Designs, Codes and Cryptography*, 53(1):13–31, 2009. doi:10.1007/s10623-009-9290-2.

[RSA78] Ronald L Rivest, Adi Shamir, and Leonard Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

[Rud09] Terry Rudolph. Simple encoding of a quantum circuit amplitude as a matrix permanent. *Physical Review A*, 80(5):054302, 2009. doi:10.1103/physreva.80.054302.

[S$^+$05] Alan D Sokal et al. The multivariate Tutte polynomial (alias Potts model) for graphs and matroids. *Surveys in Combinatorics*, 327:173–226, 2005. doi:10.1017/cbo9780511734885.009.

[SB09] Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 465(2105):1413–1439, 2009. doi:10.1098/rspa.2008.0443.

[Sch04] Stefan Scheel. Permanents in linear optical networks. *arXiv prerint quant-ph/0406127*, 2004, arXiv:quant-ph/0406127.

[She10] Dan Shepherd. Binary matroids and quantum probability distributions. *arXiv preprint 1005.1744*, 2010, arXiv:1005.1744.

[Sho99] Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. doi:10.1137/s0036144598347011.

[SJ08] Peter W Shor and Stephen P Jordan. Estimating Jones polynomials is a complete problem for one clean qubit. *Quantum Information & Computation*, 8(8):681–714, 2008.

[Sly10] Allan Sly. Computational transition at the uniqueness threshold. In *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 287–296. IEEE, 2010. doi:10.1109/focs.2010.34.

[SRK15] Magdalena Stobińska, Peter P Rohde, and Paweł Kurzyński. Quantum leap: how to complete a quantum walk in a single step. *arXiv preprint 1504.05480*, 2015, arXiv:1504.05480.

[SS12] Allan Sly and Nike Sun. The computational hardness of counting in two-spin models on d-regular graphs. In *53rd Annual IEEE Symposium* *on Foundations of Computer Science (FOCS)*, pages 361–369. IEEE, 2012. doi:10.1109/focs.2012.56.

[SST14] Alistair Sinclair, Piyush Srivastava, and Marc Thurley. Approximation algorithms for two-state anti-ferromagnetic spin systems on bounded degree graphs. *Journal of Statistical Physics*, 155(4):666–686, 2014. doi:10.1007/s10955-014-0947-5.

[Sto85] Larry Stockmeyer. On approximation algorithms for #P. *SIAM Journal on Computing*, 14(4):849–861, 1985. doi:10.1137/0214060.

[TD04] Barbara M Terhal and David P DiVincenzo. Adaptive quantum computation, constant depth quantum circuits and Arthur-Merlin games. *Quantum Information & Computation*, 4(2):134–145, 2004.

[Thi87] Morwen B Thistlethwaite. A spanning tree expansion of the Jones polynomial. *Topology*, 26(3):297–309, 1987. doi:10.1016/0040-9383(87)90003-6.

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991. doi:10.1137/0220053.

[Tut47] WT Tutte. A ring in graph theory. In *Proceedings of the Cambridge Philosophical Society*, volume 43, page 26, 1947.

[Val79] Leslie G Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979. doi:10.1016/0304-3975(79)90044-6.

[Wei06] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing*, pages 140–149. ACM, 2006. doi:10.1145/1132516.1132538.

[Wel76] Dominic JA Welsh. *Matroid theory*. Academic Press, 1976.

[Wel93] Dominic JA Welsh. Complexity, knots, colourings and counting. *London Mathematical Society Lecture Note Series*, 186:372–390, 1993. doi:10.1017/cbo9780511752506.

[Whi35] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935. doi:10.2307/2371182.

[YMKC18] Zhi-Cheng Yang, Konstantinos Meichanetzidis, Stefanos Kourtis, and Claudio Chamon. Scrambling via braiding of nonabelions. *arXiv preprint 1804.01097*, 2018, arXiv:1804.01097.

[Yue86] Horace P Yuen. Generation, detection, and application of high-intensity photon-number-eigenstate fields. *Physical Review Letters*, 56(20):2176, 1986. doi:10.1103/physrevlett.56.2176.

[Yur86] B Yurke. Input states for enhancement of fermion interferometer sensitivity. *Physical Review Letters*, 56(15):1515, 1986. doi:10.1103/physrevlett.56.1515.