2011

# The Power Of Quantum Walk Insights, Implementation, And Applications

Chen Fu Chiang
*University of Central Florida*

# The Power of Quantum Walk: Insights, Implementation, and Applications

by

Chen-Fu Chiang
B.S. Computer Science, University of Central Florida, 2002
M.S.E. Computer and Information Science, University of Pennsylvania, 2003

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the Department of Electrical Engineering and Computer Science
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Fall Term
2011

Major Professor:
Pawel Wocjan

# ABSTRACT

In this thesis, I investigate quantum walks in quantum computing from three aspects: the insights, the implementation, and the applications. Quantum walks are the quantum analogue of classical random walks. For the insights of quantum walks, I list and explain the required components for quantizing a classical random walk into a quantum walk. The components are, for instance, Markov chains, quantum phase estimation, and quantum spectrum theorem. I then demonstrate how the product of two reflections in the walk operator provides a quadratic speed-up, in comparison to the classical counterpart.

For the implementation of quantum walks, I show the construction of an efficient circuit for realizing one single step of the quantum walk operator. Furthermore, I devise a more succinct circuit to approximately implement quantum phase estimation with constant precision controlled phase shift operators. From an implementation perspective, efficient circuits are always desirable because the realization of a phase shift operator with high precision would be a costly task and a critical obstacle.

For the applications of quantum walks, I apply the quantum walk technique along with other fundamental quantum techniques, such as phase estimation, to solve the partition function problem. However, there might be some scenario in which the speed-up of spectral gap is insignificant. In a situation like that that, I provide an amplitude amplification-based

approach to prepare the thermal Gibbs state. Such an approach is useful when the spectral gap is extremely small. Finally, I further investigate and explore the effect of noise (perturbation) on the performance of quantum walks.

Thesis Supervisor: Pawel M. Wocjan

Title: Associate Professor of Electrical Engineering and Computer Science

*To those wonderful people that appear in my journey to help me when I need it.*

*For William, Viba and Baiv: Thank you for being there all these years.*

# ACKNOWLEDGMENTS

I gratefully acknowledge the support of National Science Foundation grants CCF-0726771 and CCF-0746600 for my graduate study. I must express my sincere thanks to my advisor: Dr. Pawel Wocjan, for he guided me into the quantum computing research that has fascinated me throughout the years. His continuous support and his enthusiasm for research are among the motivations that make my choice of quantum research more worthwhile.

In summers of 2007, 2008 and 2009, I had the opportunity to participate in the Canadian summer school of quantum information program. Through this program, I had the chance to learn various topics in quantum computing, either theoretical or experimental. This great experience provided me the motivation and enthusiasm to study quantum computing. I must thank the higher education institutes in Canada for their dedication to quantum information science and their hospitality in welcoming researchers from all over the world, regardless of their nationality. In the meantime, I must also thank Dr. Michele Mosca from the Institute of Quantum Computing (IQC) for giving me the opportunity for a long visit in 2010. It was then that I felt the devotion people have in IQC for quantum research and saw the rigor it takes to conduct research in this field.

I want to thank my family members, especially my father who passed away in September 2008, for their understanding and the sacrifices they made so that I could study in the United States. Furthermore, I want to express my appreciation to the coauthors of my

publications: Dr. Anura Abeyesingh, Dr. Daniel Nagaj and Hamed Ahmadi. Their constant encouragement and guidance enriched my research and benefited my research skills. At last, but not at least, I must thank the department of computer science at the University of Central Florida (UCF) for providing me the environment and facility to do research and pursue my Ph.D.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1
# INTRODUCTION

## 1.1 Motivation

The modern computer is ubiquitous in our daily life. We probably can describe a computer from our knowledge based on its physical traits and internal components. However, such a description is restricted due to the limitation of current technology. From a more general viewpoint, a computer is a physical device that is capable of executing algorithms with much stronger computation ability than human beings. Such a description gives us the flexibility to define a computer more appropriately. An algorithm is a well-defined procedure with a finite description for solving some specific problems. To be exact, the process of executing an algorithm is an information-processing task.

For various hard problems, the hardness comes from the fact that the required computational resources, space and time, are overwhelming. For instance, the RSA problem is based on the hardness of factoring large numbers. By augmenting the input size of the problem, RSA would withstand attacks from bigger and faster computers because the required computational resources grow exponentially for a classical computer. In 1994, Shor invented a polynomial time algorithm for the factorization problem but the invention is based on the computational capability of a quantum computer. But what is a quantum computer? Devices that perform quantum information processing are known as quantum computers. Quantum information processing is the outcome of using quantum mechanical

systems. Quantum mechanics is a mathematical framework for the development of physical theories.

In the classical computing regime, the Markov Chain Monte Carlo (MCMC) method and random walks are centerpieces of many efficient classical algorithms. It allows us to approximately sample from a particular distribution $\pi$ over a large state space $\Omega$. Sampling from stationary distributions of Markov chains combined with the simulated annealing is the core of many clever classical approximation algorithms. For instance, approximating the volume of convex bodies [LV06], approximating the permanent of a non-negative matrix [JSV04], and the partition function of statistical physics models such as the Ising model [JS93] and the Potts model [BSVV08]. In addition, one can also use random walks to search for the *marked* state in the Markov chain, in which the *hitting time* is of interest. It is because hitting time indicates the time it requires to find the marked state. Given the promise of computational power of a quantum computer, the quantization of random walk into quantum walk was thus born. As expected and verified, quantum walk renders the solution more efficiently than its classical counterpart.

I ask and answer questions about the power of quantum walk by examining the essential ingredients of the walk, showing how to construct an efficient quantum circuit to simulate the quantum walk operator, and designing effective quantum walk based algorithms to solve problems that are classically considered hard. From a top level, Figure 1.1 depicts the scope of my research and illustrates the connections between my work and the fundamental quantum algorithm techniques.

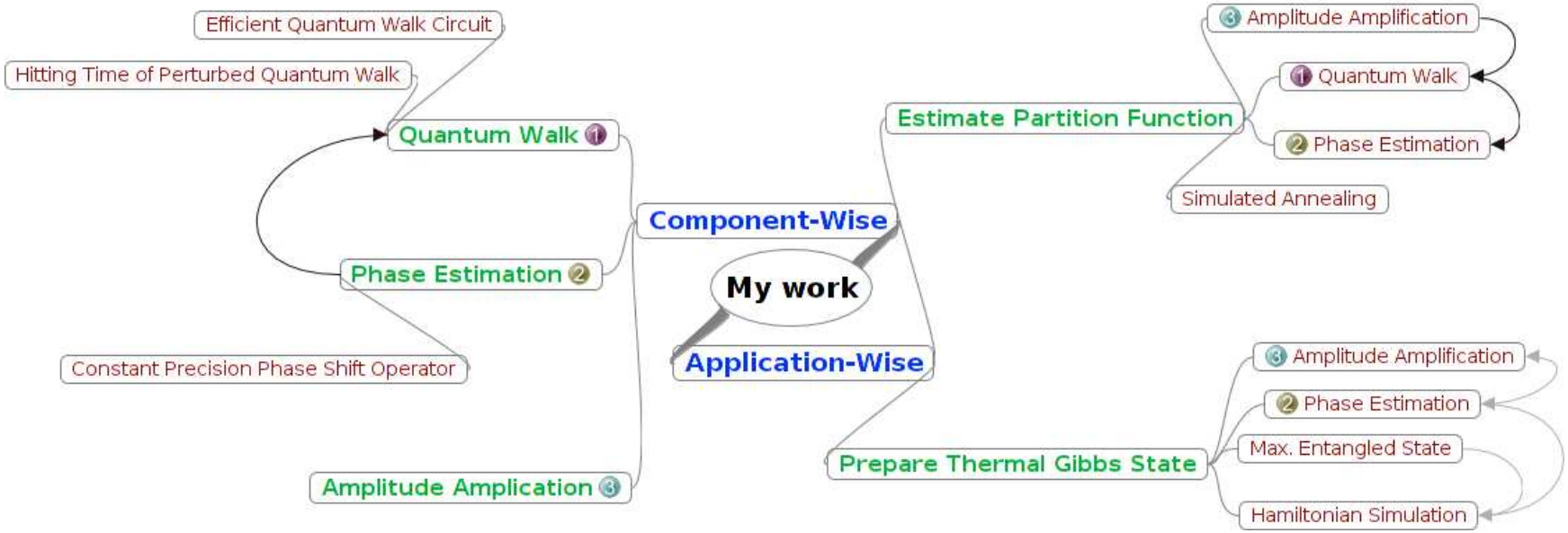


Figure 1.1: My Research Scope

## 1.2 Outline and Summary of Results

The content of this dissertation revolves around the discrete time quantum walk. To present this technique in a more pedagogical manner, we divide the dissertation into three parts – *Basics and the Insights, Quantum Circuit: Implementation, and Quantum Walk Applications.*

We outline and summarize the main results of each chapter and list them as follows:

**Part I : Basics and the Insights**

**Chapter 2**

We present some fundamental notions, such as circuit models and complexity, of classical computation and quantum computation that will form the basis for much of what follows in the remaining chapters. We also present important properties, such as the convergence rate and the stationary distribution, in classical Markov chains that later we can compare to justify the the improvement gained from quantum walk.

**Chapter 3**

In this chapter, we present the main ingredients in quantizing a quantum walk: the product of two reflections based on the given Markov transition matrix, the spectrum of this product, and the phase estimation technique. The core contribution from the quantum walk is the quadratic speedup that was made possible by the spectrum property of the walk operator

and the use of phase estimation.

## Part II : Quantum Circuit - Implementation

### Chapter 4

While Quantum phase estimation (QPE) is at the core of many quantum algorithms known to date, its physical implementation (algorithms based on quantum Fourier transform (QFT)) is highly constrained by the requirement of high-precision controlled phase shift operators, which remain difficult to realize. In this chapter, we introduce an alternative approach to approximately implement QPE with arbitrary constant-precision controlled phase shift operators.

The new quantum algorithm bridges the gap between QPE algorithms based on QFT and Kitaev's original approach. For approximating the eigenphase precise to the nth bit, Kitaev's original approach does not require any controlled phase shift operator. In contrast, QPE algorithms based on QFT or approximate QFT require controlled phase shift operators with precision of at least $\pi/2n$. The new approach fills the gap and requires only arbitrary constant-precision controlled phase shift operators. From a physical implementation viewpoint, the new algorithm outperforms Kitaev's approach.

### Chapter 5

We present an efficient general method for realizing a quantum walk operator corresponding to an arbitrary sparse classical random walk. Our approach is based on Grover and

Rudolph's method for preparing coherent versions of efficiently integrable probability distributions [GR02]. This method is intended for use in quantum walk algorithms with polynomial speedups, whose complexity is usually measured in terms of how many times we have to apply a step of a quantum walk [SZE04], compared to the number of necessary classical Markov chain steps. We consider a finer notion of complexity including the number of elementary gates it takes to implement each step of the quantum walk with some desired accuracy. The difference in complexity for various implementation approaches is that our method scales linearly in the sparsity parameter and poly-logarithmically with the inverse of the desired precision. The best previously known general methods either scale quadratically in the sparsity parameter, or polynomially in the inverse precision. Our approach is especially relevant for implementing quantum walks corresponding to classical random walks like those used in the classical algorithms for approximating permanents [JSV04, BSVV08] and sampling from binary contingency tables [BSSV06]. In those algorithms, the sparsity parameter grows with the problem size, while maintaining high precision.

## Part III : Quantum Walk Applications

### Chapter 6

We present a quantum algorithm based on classical fully polynomial randomized approximation schemes (FPRAS) for estimating partition functions that combine simulated annealing with the Monte-Carlo Markov Chain method and use non-adaptive cooling schedules. We achieve a twofold polynomial improvement in time complexity: a quadratic reduction with

respect to the spectral gap of the underlying Markov chains and a quadratic reduction with respect to the parameter characterizing the desired accuracy of the estimate output by the FPRAS. Both reductions are intimately related and cannot be achieved separately.

First, we use Grover's fixed point search, quantum walks and phase estimation to efficiently prepare approximate coherent encodings of stationary distributions of the Markov chains. The speed-up we obtain in this way is due to the quadratic relation between the spectral and phase gaps of classical and quantum walks. The second speed-up with respect to accuracy comes from generalized quantum counting, used instead of classical sampling to estimate expected values of quantum observables.

**Chapter 7**

While quantum walks may have many useful applications, there may be a situation in which the spectral gap of the corresponding transition matrix $P$ is extremely small. The quadratic speed-up might remain insignificant. In a situation such as this, we can use techniques like amplitude amplification. In a recent work [PW09], *Poulin* and *Wocjan* presented a quantum algorithm for preparing thermal Gibbs states of interacting quantum systems. This algorithm is based on Grover's technique for quantum state engineering, and its running time is dominated by the factor $\sqrt{D/\mathcal{Z}_\beta}$, where $D$ and $\mathcal{Z}_\beta$ denote the dimension of the quantum system and its partition function at inverse temperature $\beta$, respectively.

We present here a modified algorithm and a more detailed analysis of the errors that arise due to imperfect simulation of Hamiltonian time evolutions and limited performance

of phase estimation (finite accuracy and nonzero probability of failure). This modification together with the tighter analysis allows us to prove a better running time by the effect of these sources of error on the overall complexity. We think that the ideas underlying of our new analysis could also be used to prove a better performance of quantum Metropolis sampling by *Temme* et al. [TOVPV09].

**Chapter 8**

The hitting time is the required minimum time for a Markov chain-based walk (classical or quantum) to reach a target state in the state space. We investigate the effect of the perturbation on the hitting time of a quantum walk. We obtain an upper bound for the perturbed quantum walk hitting time by applying Szegedy's work and the perturbation bounds with Weyl's perturbation theorem on classical matrix. Based on the definition of quantum hitting time given in MNRS algorithm, we further compute the delayed perturbed hitting time (DPHT) and delayed perturbed quantum hitting time (DPQHT). We show that the upper bound for DPQHT is actually greater than the difference between the square root of the upper bound for a perturbed random walk and the square root of the lower bound for a random walk.

Based on an efficient quantum sample preparation approach invented in *speed-up via quantum sampling* and the perturbation bounds for stationary distribution for classical matrix, we find an upper bound for the total variation distance between the prepared quantum

sample and the true quantum sample.

## Chapter 9

The quantum walk is a useful technique that can be used to solve various problems [AMB04, CCDFGS03, CSV07, MSS05, WCNA09, KMOR10]. For some oracular problems, quantum walks render exponential speedups over its classical counterparts [CCDFGS03, CSV07]. For some other problems, such as NAND tree evaluation problem [FGG08] and the triangle finding problem [MSS05], quantum walks render polynomial speedup over classical algorithms. In this chapter, we will discuss possible problems for future study. Some of the problems have been tried quantumly, but there may be space left for improvement. Some of the problems have been tried out via classical random walk, but there is no quantum version yet. It would be of interest to many to investigate the possible speed-up (exponential, polynomial or none) that quantum walks can provide for those problems.

# CHAPTER 2
# PRELIMINARIES

In this chapter, a collection of common notations, definitions and theorems in quantum computing and classical computation is presented. The collection contains the introductions to classical computation, quantum computation, and the classical Markov chains. For classical computation, the core presentation covers the classical complexity quantifying approach and the classical circuit model. For quantum computation, the core presentation covers reversible computation, quantum circuit model, qubit operations and quantum complexity. For classical Markov chains, we introduce the necessary properties for a Markov chain to converge to a stationary distribution. To do so, we have this part of presentation cover definitions of required properties, such as irreducibility, aperiodicity and spectral gap.

These techniques and notations are useful for understanding the materials presented in later chapters. As we will, from time to time, discuss the complexity of algorithms, it is important to know the approach to quantify the complexity of a given algorithm. Furthermore, the complexity of an algorithm is a standard method for measuring the performance of an algorithm. This provides theorists the measure when comparing algorithms, that solve the same problem, in terms of their performance.

For each quantum algorithm, there exists an equivalent corresponding quantum circuit performing the computation that the quantum algorithm intends. Hence, it is important to

understand the quantum circuit model and the qubit operation inside the circuit.

For providing a thorough background in the preliminary, I consult various sources [KLM07, NC00, LPW09, GOL10] extensively for the fundamental materials. These references substantially contribute to the composition and the structure of the preliminaries.

## 2.1 Introduction to Computation

From a general viewpoint, a computation can be thought of as a process that modifies a relatively large environment through repeated applications of a simple and predetermined rule. Although the effect after each application of the rule is limited, the accumulated effect after many applications of the rule may be relatively, or maybe extremely, complex. What is of interest is the transformation of the environment effected by the computation. In computer science, it is common to encode the initial environment into binary strings and offer the binary string to the computation mechanism, the Turing Machine for instance, to execute the simple and predetermined rule. The result, or the end environment, is also rendered in the string format when the computation halts. In another word, the computation behaves like a function (can be partial) that maps from inputs to outputs, and such a mapping can be considered as solving a problem by using repeated applications of the rule, such as the transition function $\delta$ in the Turing Machine. We refer interested readers to the references [GOL08, GOL10] for the *Turing Machine* model of computation.

## 2.2 The Circuit Model of Computation

Circuits are networks composed of wires and gates. The wires carry bit values from the left through gates that perform some simple operations on the bits. The output after the operations from the gates will be delivered from the right to the output wires. A circuit $C_n$ has $n$ wires and an illustration of $C_6$ is shown in Figure 2.2. As seen in the figure, the wires carry the value of the bits, and the blocks $G_1$, $G_2$ are gates. The input bits are $i_1, i_2, i_3, i_4, i_5$ and the output bits are $o_1, o_2, o_3, o_4, o_5$. A family of circuits is a set of circuits $\{C_n | n \in cZ\}$. We say the family is *uniform* if each $C_n$ can be easily constructed. The reason for uniformity is to provide equal computation power in the definition of the circuits themselves. One important notion in circuits is universality. It is a set of different elementary gates that can be used to construct a circuit for any desired computation.

**Definition 1.** *[KLM07] For classical computation, a set of gates is universal if, for any positive integers $n, m$ and a function $f : \{0,1\}^n \to \{0,1\}^m$, a circuit can be constructed for computing $f$ using only gates from that set.*

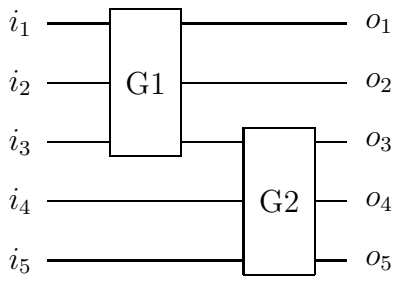


Figure 2.1: A Simple Circuit Diagram

A well-known universal set of gates for the classical computation is {NAND, FANOUT}. However, if being restricted to reversible gates, we would need more than just one-bit and two-bit gates. The *Toffoli* gate (see Figure 2.2) is a reversible three-bit gate that flips the value of the third bit when the first two bits are both 1. The Toffoli gate is universal when we are allowed to add ancillary bits to the circuit that can be initialized to either 0 or 1 as required.

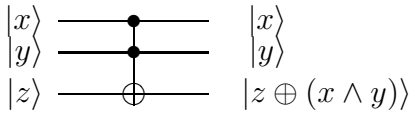


Figure 2.2: A Toffoli Gate

When we are given a circuit to perform certain types of computation task, it is natural to ask how good this circuit is. There might be several approaches to tackle the problem. That is why we need a way to quantify the required source to finish the computation task. The complexity of a circuit $C_n$ can be considered based on the *time* or the *space* that the circuit $C_n$ uses to complete the computation. One simple quantification would be the measure of the number of gates used in the circuit $C_n$. A second approach would be the depth of the circuit. This approach divides the circuit into a sequence of discrete time-slices and each time-slice is the time required by an application of a gate. Finally, the third approach would be the measure of space. This is the total number of wires required in the circuit. A good example is illustrated in Figure 2.3. In general, unless otherwise specified, the complexity of a circuit often refers to the depth of the circuit.

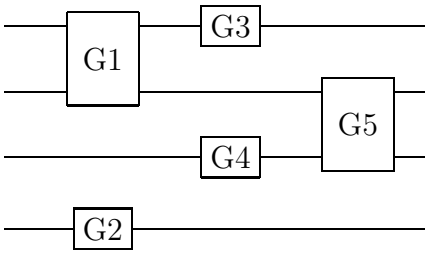


Figure 2.3: A Circuit of Depth 3, Space 4, and Size 5

### 2.2.1 Quantifying Computational Resources

Different models of computation lead to different resource requirements for computation. It is desirable to have a universal approach to quantify resource requirements that is independent of relatively trivial changes in the computational models. In the previous section we introduce how to count the required sources for a given circuit in terms of either *depth, space or gates*. Consider the case when the circuit is very complex and the required resources might be polynomial (or exponential) with respect to the input size of the problem. One well-known tool is asymptotic notation, which summarizes the essential behavior of a function. This asymptotic approach associates with three major notions – the **big-**$O$, the **big-**$\Omega$ and the **big-**$\Theta$.

The **big-**$O$ **notation** is used to bound the behavior of a function from the above. Suppose we have two functions, $f(n)$ and $g(n)$, that take on non-negative integers as the domain. We say $g(n)$ sets the upper bound on $f(n)$ if $f(n) \leq cg(n)$ for some constant $c$ and large $n$. Big-$O$ notation is useful for learning the worst-case behavior of an algorithm. Similarly, the **big-**$\Omega$ **notation** is used to bound the function from below. We say $g(n)$ sets the lower

bound on $f(n)$ if $cg(n) \leq f(n)$ for some constant and large $n$ in the domain. Finally, the **big-$\Theta$ notation** indicates that $f(n)$ behaves the same as $g(n)$ asymptotically, up to some unimportant factors. That is to say when $f(n)$ is $\Theta(g(n))$, it means $\Omega(f(n)) = c_1(g(n)) \leq f(n) \leq c_2(g(n)) = O(f(n))$ where $c_1$ and $c_2$ are some constants.

Let us consider an example to demonstrate the notation more explicitly. Let $f(n) = 5n^2 + 10n + 2$ and since $n^3 \geq 5n^2 + 10n + 2$ for large $n$, we can say $f(n)$ is bounded from above by $n^3$ (not a very tight bound though). On the other hand, it is clear that the function $f(n)$ is $\Omega(n^2)$.

With this asymptotic notation, we can classify problems into different categories. Computational complexity theory classifies the hardness of various computational problems. It determines the complexity of any well-defined task. Each classification represents a computational class. A computational class is a collection of computational problems that share some common features in terms of the required computational resources, such as time or space. An additional goal of complexity theory is the study of the relations between various computational classes. Interestingly, complexity theory has been more successful in the later (the study of the relations among the classes) than in the former (determination of the complexity of a well-defined task).

## 2.3 Quantum Computation

The theory of quantum computing is related to reversible computation. Reversible computing means that we can uniquely recover the input when given the output. For instance, a CNOT (Controlled NOT) is reversible (see Figure 2.4). We can uniquely determine $x, y$ by running CNOT gate again on the output $x$ and $y \oplus x$.

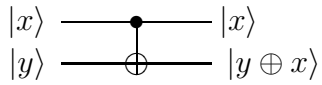


Figure 2.4: A Simple CNOT Gate

**Definition 2.** *An operator $U$ is called unitary if $U^\dagger = U^{-1}$, where $U^{-1}$ is the inverse of $U$.*

In the standard matrix representation, the matrix for $U^\dagger$ is the complex conjugate transpose of the matrix for $U$. In quantum computation, allowed operations are unitary matrices, which perfectly reserve the Euclidean norm of the input. Consequently, since a unitary operation can be undone by running the unitary backwards, quantum computations are reversible. The reversible property of a unitary operator can be further applied to a broader scale: the circuit. When we are given the output, we can run the circuit backwards to obtain the *unique input*.

### 2.3.1 Quantum Circuit Model

One of the most commonly used computational models for quantum computing is the *quantum circuit model*. The term quantum circuit refers to an acyclic network of quantum gates connected by wires. Similar to the classical circuit model in terms of the presentation format, the quantum circuit is also read from left to right as shown in Figure 2.5. $|w_i\rangle$ and $|z_i\rangle$ are equal iff $|\phi\rangle = |1\rangle$. The first gate is Toffoli controlled by 0 and the second gate is the regular Toffoli gate. Each line in the circuit represents a wire in the quantum circuit. The wire is not necessarily a physical wire; it might represent *the passage of time or just a physical particle*, such photon or electron, moving from one location to another through space.

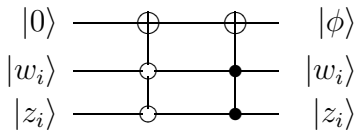


Figure 2.5: A Circuit That Tests Bit-wise Equality

### 2.3.2 Single Qubit Operation and Controlled Operations

A single qubit is a vector $|\psi\rangle = a|0\rangle + b|1\rangle$ described by two complex numbers satisfying $|a|^2 + |b|^2 = 1$. Since quantum computation is reversible, it is necessary for the operations on qubits to be unitary (that preserves the norm) and they can be described by a $2 \times 2$ matrix. The most important (and common) single qubit operators are the *Identity* matrix, the *Pauli matrices*:

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}; \quad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}; \quad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}; \quad I \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{2.1}$$

and the Hadamard gate (H), phase gate (S) and $\pi/8$ gate (T):

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \qquad S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; \qquad T \equiv \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/8) \end{bmatrix}. \tag{2.2}$$

For instance, when operator $X$ acts on the state $|\psi\rangle$, we would get $X|\psi\rangle = b|0\rangle + a|1\rangle$. When the Hadamard operator acts on $|\psi\rangle$, we have $H|\psi\rangle = \frac{a+b}{\sqrt{2}}|0\rangle + \frac{a-b}{\sqrt{2}}|1\rangle$.

The prototypical controlled operation is controlled-NOT, as seen in Fig 2.4. Since it only flips the target bit when the control bit is 1, in computational basis we can interpret such an action as

$$CNOT = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes X. \tag{2.3}$$

Similarly, suppose $U$ is an unitary operator that acts on one single qubit, then we can perform *controlled-U* operation with a control bit and a target bit. The controlled-$U$ presented by the quantum circuit is shown at Fig 2.6. The top line is the control qubit and the bottom line is the target qubit. When the top qubit $|a\rangle$ is $|1\rangle$, $U$ is applied to the target qubit $|b\rangle$, otherwise left alone. The matrix presentation of the controlled-$U$ can be written as follows: *controlled-U* $= |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$.

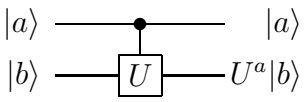


Figure 2.6: A Simple Controlled-U Gate

### 2.3.3 Universal Sets of Quantum Gates and N-Qubit Operation

The quantum gates we have seen so far are either 1-qubit or 2-qubit operators. An efficient quantum algorithm would, in most cases, require more complicated unitary operators acting on $n$-qubits. To implement $n$-qubit operators, we need to construct them by using the elementary gates from the universal set. The following definitions and theorems show that $n$-qubit operators can be efficiently simulated by 1-qubit and 2-qubit operators.

When the Pauli matrices are exponentiated, they become three useful classes of unitary matrices: the rotation operators $R_x, R_z$ and $R_z$. They are defined as follows:

$$R_x(\theta) \equiv e^{\frac{-i\theta X}{2}}, \quad R_y(\theta) \equiv e^{\frac{-i\theta Y}{2}}, \quad R_z(\theta) \equiv e^{\frac{-i\theta Z}{2}} \tag{2.4}$$

The following theorem tells us that any arbitrary 1-qubit unitary gate can be decomposed into a sequence of one rotation around the y-axis and two rotations around the z-axis along with some phase factor.

**Theorem 1.** *Suppose $U$ is a 1-qubit unitary gate. Then there exist real numbers, $\alpha, \beta, \gamma$, and $\delta$ such that*

$$U = e^{i\alpha} R_z(\beta) R_y(\gamma) R_z(\delta). \tag{2.5}$$

**Definition 3.** *A set of gates is said to be universal if, for any integer $n \geq 1$, any $n$-qubit operator can be approximated to arbitrary accuracy by a quantum circuit using only gates from that set.*

**Definition 4.** *A 2-qubit gate is said to be an entangling gate* [1] *if, for some input product state $|\psi\rangle|\phi\rangle$, the output of the gate is not a product state.*

**Theorem 2.** *A set composed of any 2-qubit entangling gate, together with all 1-qubit gates, is universal.*

An entangling 2-qubit gate with all 1-qubit gates enables us to implement an arbitrary $n$-qubit unitary exactly. However, a major drawback is that the universal sets of gates are infinite and that would not be helpful when it comes to implementation. We can start off the task of finding a finite universal set by searching for a finite set of 1-qubit gates that we can use to approximate an arbitrary 1-qubit gate to any desired accuracy.

**Definition 5.** *A set of gates is said to be universal for 1-qubit gates if any 1-qubit unitary gate can be approximated to arbitrary accuracy by a quantum circuit using only gates from that set.*

**Theorem 3.** *If a set of two 1-qubit $\varrho = \{R_l(\beta), R_m(\gamma)\}$ satisfies the conditions*
*(I) $l$ and $m$ are non-parallel axes of the Bloch sphere and*
*(II) $\beta, \gamma \in [0, 2\pi)$ are real numbers such that $\frac{\beta}{\pi}$ and $\frac{\gamma}{\pi}$ are not rational*
*then $\varrho$ is universal for 1-qubit gates.*

[1] *It is clear that CNOT gate is an entangling gate, simply let $|\psi\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $|\phi\rangle = |0\rangle$*

**Theorem 4.** *The set* $\{CNOT, H, T\}$ *is a universal set of gates.*

Theorem 4 guarantees us that we can approximate any $n$-qubit unitary $U$ implemented in an arbitrary quantum algorithm. We are interested in devising an efficient approach to approximate $U$ by using only a *polynomial* number of gates from the universal set. The polynomial term here means polynomial in $1/\epsilon$ and in the number of qubits $n$ where $\epsilon$ is the desired precision of the approximated $U$. The Solovay-Kitaev theorem has the answer to this question.

**Theorem 5.** *If* $\varrho$ *is a finite set of 1-qubit gates satisfying the conditions in Theorem 3 and (III) for any gate* $g \in \varrho$*, its inverse* $g^{-1}$ *can be implemented exactly by a finite sequence of gates in* $\varrho$*, then any 1-qubit gate can be approximated with error at most* $\epsilon$ *using* $O(\log^c(1/\epsilon))$ *gates from* $\varrho$*, where* $c$ *is a positive constant.*

The decomposition of $U$ with gates from the universal set is a two-fold process. The first step is to decompose $U$ into two-level unitary matrices. A two-level unitary matrix acts non-trivially only on two-or-fewer vector components. The second step is to efficiently approximate two-level matrices by using gates from the finite universal set.

**Theorem 6.** *For an arbitrary* $d \times d$ *unitary matrix* $U$*, it can be written as*

$$U = V_1 V_2 \dots V_k, \tag{2.6}$$

*where matrices* $V_i$ *are two-level unitary matrices and* $k \leq (d-1)+(d-2)+..+1 = d(d-1)/2$.

Suppose the unitary $U$ consists of several CNOT gates and $m$ 1-qubit gates after the first step of decomposition. If we want to approximate $U$ with precision $\epsilon$, by Solovay-Kitaev's theorem the approximation of those gates would require at most

$$O(m \log^c \frac{m}{\epsilon}) \tag{2.7}$$

gates from the universal set $\{CNOT, H, T\}$. This is only a poly-logarithmic increase over the size of the original circuit.

## 2.4 Classical Markov Chains

For many tasks, such as simulated annealing [KGV83, CER85], computing the volume of convex bodies [LV06] and approximating the permanent of a matrix [JSV04, BSVV08] (see references in [WCNA09] for more), the best approaches known today are randomized algorithms based on Markov chains (random walks) and sampling.

A classical Markov chain is a random process that moves among the elements in the state space $\Omega$. The movement of the process occurs in this manner: when a $x \in \Omega$, the next position is chosen basing on a fixed probability distribution $P(x, \cdot)$ where $P(x, \cdot)$ is the $x$-th row of transition (stochastic) matrix $P$. A matrix is recognized as transition (stochastic) matrix when the entries of each row of the matrix sum up to 1.

To be more exact, the moves based on the above manner can be viewed as a series of random variables $X = (X_0, X_1, \ldots)$. $X$ is considered the Markov chain with *transition matrix* $P$ and *state space* $\Omega$. Markov chain holds the *Markov Property*, namely that, given the current state, the future and the past states are independent. Formally that

$$\Pr(X_t = x_t | X_0 = x_0, X_1 = x_1, \ldots, X_{t-1} = x_{t-1}) = \Pr(X_t = x_t | X_{t-1} = x_{t-1}). \tag{2.8}$$

Because of this property, it is sufficient to use an $|\Omega| \times |\Omega|$ transition matrix $P$ to describe the moves of the Markov chain. When the state space $\Omega$ is finite, the corresponding Markov chain is *finite*.

**Definition 6.** *An $n \times n$ transition matrix $P$ is stochastic if $P$ only has non-negative entries such that the sum of entries of each row (column) is 1. It is called row-wise (column-wise) stochastic. A doubly stochastic matrix is when $P$ is both row-wise and column-wise stochastic.*

### 2.4.1 Irreducibility and Aperiodicity

To analyze the long-term behaviour of a Markov chain, it is important to classify its states. For a finite Markov chain, this is equivalent to examining the structure connectivity of the corresponding directed graph of the Markov chain.

**Definition 7.** *State $i$ is accessible from state $j$ if for some integer $n \geq o$ $P^n(x, y) > 0$. When two states are accessible from each other, we say they commute and they belong to the same communicating class.*

**Definition 8.** *A Markov chain is irreducible if all states belong to one communicating class.*

This means it is possible to get from any state to any other state using a sequence of positive transition probability.

**Definition 9.** *For any starting state $x$, let $\tau(x) := \{t \geq 1 : P^t(x, x) > 0\}$ be the set of times when the chain re-visit state $x$. The period of $x$ is defined as the greatest common divisor of $\tau(x)$.*

**Definition 10.** *A Markov chain is aperiodic when all states has period 1.*

The importance for a matrix $P$ to be aperiodic and irreducible leads to the following two propositions.

**Proposition 1.** *If $P$ is* aperiodic and irreducible*, then there is an integer $r$ such that $P^r(x, y) > 0$ for all $x, y \in \Omega$.*

**Proposition 2.** *[LPW09] If $P$ is the transition matrix of an irreducible markov chain, then there exists a probability distribution $\pi$ on $\Omega$ such that $\pi = \pi P$ and $\pi(x) > 0$ for all $x \in \Omega$.*

### 2.4.2 Stationary Distributions, Spectral Gap and Convergence

**Definition 11.** *A distribution $\pi$ (row vector) on $\Omega$ satisfying*

$$\pi P = \pi, \tag{2.9}$$

*is the stationary distribution of row-wise stochastic matrix $P$. Similarly, if $\pi$ is stationary distribution of a column-wise stochastic matrix $P$, then $P\pi = \pi$ where $\pi$ is a column vector.*

**Fact 1.** *Let $P$ be the* transition matrix of a finite Markov chain. *(i) If $\lambda$ is an eigenvalue of $P$, the $|\lambda| \leq 1$. (ii) If $P$ is irreducible and aperiodic, then $-1$ is* not *an eigenvalue of $P$.*

**Definition 12.** *Suppose $P$ is an $n \times n$ transition matrix with eigenvalues*

$$-1 \leq \lambda_n(P) \leq \ldots \leq \lambda_1(P) = 1. \tag{2.10}$$

*Define*

$$\lambda_\star = \max\{|\lambda_i(P)| : \lambda_i(P) \neq 1\}. \tag{2.11}$$

*The difference $\delta = 1 - \lambda_\star$ is the spectral gap.*

When $P$ is aperiodic and irreducible, Fact 1 implies $\delta > 0$.

**Theorem 7.** *(Convergence Theorem)[LPW09] Suppose that an $N \times N$ matrix $P$ is* irreducible and aperodic*, with stationary distribution $\pi$. Then there exist constants $\alpha \in (0, 1)$ and $C > 0$ such that*

$$\max_{x \in \Omega} |||P^t(x, \cdot) - \pi||| \leq C\alpha^t \tag{2.12}$$

*Given a stochastic $n \times n$ matrix $P$ that is irreducible and aperiodic, the time for any arbitrary starting state $|\phi\rangle$ to converge to the stationary distribution $|\pi\rangle$ of $P$ is proportional to $1/\delta$.*

### 2.4.3 Hitting Time

For the purpose of being complete, we adopt several important terms [MNRS09] that we use frequently in our context.

Let $P$ be a reversible and ergodic transition matrix with state space $\Omega$. Assume the Markov chain $(X_1, \ldots, X_n)$ under discussion has a finite state space $\Omega$ and transition matrix $P$. We define the following:

**Definition 13.** *For $z \in \Omega$, denote the* hitting time *for $z$*

$$\tau_z := HT(P, z) = \min\{t \geq 1 : X_t = z\}. \tag{2.13}$$

*$\tau_z$ is the expected number of transition matrix $P$ invocations to reach the state $z$ when started in the initial distribution $\pi$.*

Let $\pi$ be the stationary distribution of $P$. Let $P_{-z}$ be an $(n-1) \times (n-1)$ matrix where the column and row indexed by $z$ are removed. In the same manner, $v_{-z}$ denotes the vector $v$ with the $z$-coordinate eliminated. The $z$-hitting time of $P$ can then be computed via the formula [MNRS09] $HT(P, z) = \pi_{-z}^{\dagger}(I - P_{-z})^{-1}u_{-z}$, where $u$ is an all-ones vector. It is known that

$$\pi_{-z}^{\dagger}(I - P_{-z})^{-1}u_{-z} = \sqrt{\pi_{-z}}^{\dagger}(I - S_{-z})^{-1}\sqrt{\pi_{-z}} \tag{2.14}$$

where $S_{-z} = \sqrt{\prod_{-z}}P_{-z}\sqrt{\prod_{-z}}^{-1}$ with $\prod_{-z} = diag(\pi_i)_{i \neq x}$ and $\sqrt{\pi_{-z}}$ is the entry-wise square root of $\pi_{-z}$. Let $\{v_j : j \leq n-1\}$ be the set of normalized eigenvectors of $S_{-z}$ where the eigenvalue of $v_j$ is $\lambda_j = \cos\theta_j$ with $0 \leq \theta_j \leq \pi/2$. After reordering the eigenvalues, we have

$1 \geq \lambda_1 \geq \ldots \geq \lambda_{n-1} \geq 0$. We can express $\sqrt{\pi_{-z}} = \sum_j \nu_j v_j$ in the eigenbasis of $S_{-z}$, then the $x$-hitting time satisfies:

$$HT(P, x) = \sum_j \frac{\nu_j^2}{1 - \lambda_j} \tag{2.15}$$

# CHAPTER 3
# INSIGHTS

Current research in quantum walks has two distinct models: the discrete-time step quantum walk and the continuous time evolution quantum walk. In this work, we will focus on the discrete-time quantum walk. The discrete time quantum walk is a tool for generalizing classical discrete Markov chains [AAKV01]. The discrete time quantum walk can be considered as a sequence of of unitary operations where each operation has a non-zero transition amplitude only between adjacent vertices of the graph. Classically such a graph can be presented by an adjacency matrix with entries associated the transition probability as discussed in previous chapter.

In this chapter, we present the main ingredients in quantizing a quantum walk: quantum phase estimation (QPE), the product of two reflections based on the given Markov transition matrix and the spectrum of the product of two reflections. The core contribution of quantum walks is the quadratic speedup. It is made possible due to the spectrum gap of the walk operator and the application of phase estimation.

## 3.1 Phase Estimation

Quantum phase estimation (QPE) is a key tool in many important quantum algorithms, such as order finding and quantum walks. QPE approximates the phase of a given unitary

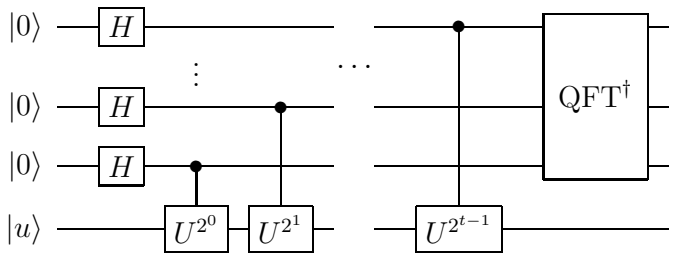


Figure 3.1: Standard Quantum Phase Estimation

operator. QPE based on quantum Fourier transform (QFT) is a standard approach used for such a task. The structure of this approach is depicted at Figure 3.1. A formal statement of the problem is as follows:

**Problem.** *Assume $U$ is a unitary matrix with eigenvalue $e^{2\pi i\varphi}$ and the corresponding eigenvector $|u\rangle$. The goal is to find $\tilde{\varphi}$ precise to the $n$-th bit such that*

$$\Pr(|\tilde{\varphi} - \varphi| < \frac{1}{2^n}) > 1 - \epsilon.$$

*Here $\varphi = 0.x_1 \dots x_n \dots$ is in binary representation.*

The standard QPE algorithm requires two registers and contains two stages. The first register is prepared as a composition of $t$ qubits initialized in the state $|0\rangle$. The second register is initially prepared in the state $|u\rangle$.

**Theorem 8.** *[NC00] To successfully obtain the eigenphase $\varphi$ accurate to $n$-th bit (precision of $1/2^n$) with probability success at least $1 - \epsilon$, we need to choose*

$$t = n + \lceil \log(2 + \frac{1}{2\epsilon}) \rceil.$$

*The number of invocations of the unitary $U$ for this algorithm is $2^t - 1$.*

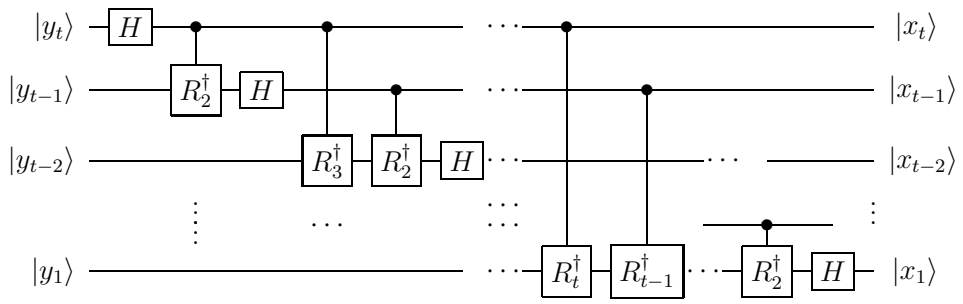


Figure 3.2: The Full-fledged Inverse QFT

In the first stage, known as the *phase kick-back stage*, the algorithm prepares a uniform superposition over all possible states and then applies the controlled-$U^{2^k}$ operations where $0 \leq k \leq t-1$. To sharpen our intuition about why QPE works, suppose $\varphi$ is of exactly of $t$ bits in binary presentation. Consequently, the state of the system for the first register after the first stage will be

$$\frac{1}{\sqrt{2^t}} \quad (|0\rangle + e^{2\pi i 0.x_t}|1\rangle)(|0\rangle + |e^{2\pi i 0.x_{t-1}x_t}|1\rangle) \ldots (|0\rangle + e^{2\pi i 0.x_1 x_2 \ldots x_t}|1\rangle) \tag{3.1}$$

$$= \quad |y_t\rangle \otimes |y_{t-1}\rangle \otimes \ldots \otimes |y_1\rangle \tag{3.2}$$

$$= \quad \frac{1}{2^{t/2}} \sum_{k=0}^{2^t-1} e^{2\pi i \varphi k}|k\rangle. \tag{3.3}$$

The second stage, known as *inverse QFT*, in the QPE algorithm is the $QFT^\dagger$ operation. The structure of the inverse Fourier transform is depicted at Figure 3.2. The rotation operator $R_k$ and the state $|y_i\rangle$ mentioned above are as follows:

$$R_k^\dagger \equiv \begin{pmatrix} 1 & 0 \\ 0 & e^{-2\pi i/2^k} \end{pmatrix}, \quad |y_i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_i \ldots x_t)}|1\rangle).$$

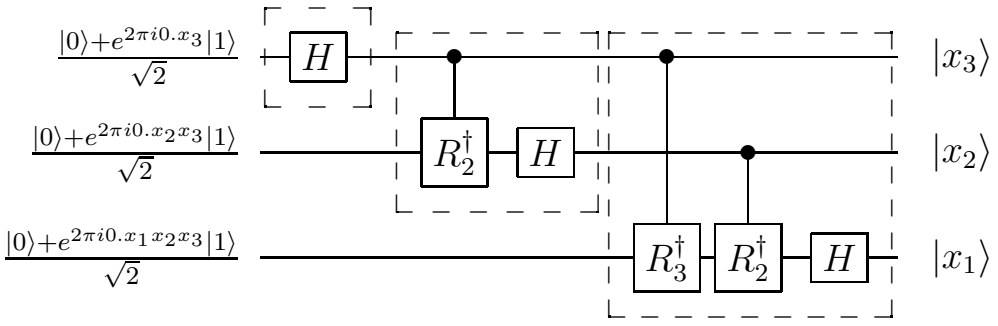


Figure 3.3: The Illustration of Inverse QFT When $t = 3$

Starting from the least significant bit, we apply the Hadamard operation to obtain the correct least significant bit $x_t$. To give readers a more clear idea, let us consider the case when $t = 3$ to illustrate how it works.

The first step is to recover the least significant bit correctly. In order to do so, we apply the Hadamard gate.

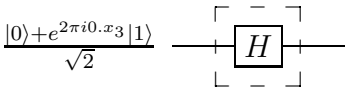


Figure 3.4: The Process to Obtain the Least Significant Bit $x_3$

By simple computation, we see that $x_3$ is obtained.

$$\frac{|0\rangle + e^{2\pi i 0.x_3}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + (-1)^{x_3}|1\rangle}{\sqrt{2}} \xrightarrow{H} |x_3\rangle \tag{3.4}$$

The second step is to recover the middle bit. We need to use the result, $x_3$, from previous step to do a controlled shift operation to reset the phase.

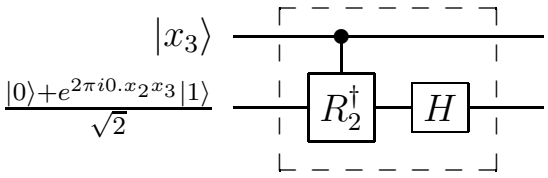


Figure 3.5: The Process to Obtain the Middle Bit $x_2$

Once the phase is properly reset, we can apply the Hadamard gate to obtain $x_2$ correctly.

$$|x_3\rangle \otimes \frac{|0\rangle + e^{2\pi i 0.x_2x_3}|1\rangle}{\sqrt{2}} \xrightarrow{ctrl\, R_2^\dagger} |x_3\rangle \otimes \frac{|0\rangle + e^{2\pi i 0.x_20}|1\rangle}{\sqrt{2}} \xrightarrow{I\otimes H} |x_3\rangle \otimes |x_2\rangle \tag{3.5}$$

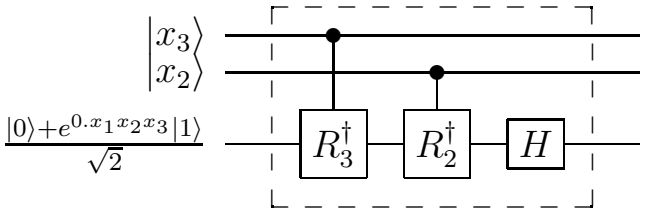


Figure 3.6: The Process to Obtain the Most Significant Bit $x_1$

When $x_3$ and $x_2$ are properly reset, we can obtain $x_1$. Then we successfully recover the eigenphase $0.x_1x_2x_3$.

$$|x_3\rangle \otimes |x_2\rangle \otimes \frac{|0\rangle + e^{2\pi i 0.x_1x_2x_3}|1\rangle}{\sqrt{2}} \xrightarrow{ctrl\, R_3^\dagger} |x_3\rangle \otimes |x_2\rangle \otimes \frac{|0\rangle + e^{2\pi i 0.x_1x_20}|1\rangle}{\sqrt{2}}$$
$$\xrightarrow{ctrl\, R_2^\dagger} |x_3\rangle \otimes |x_2\rangle \otimes \frac{|0\rangle + e^{2\pi i 0.x_100}|1\rangle}{\sqrt{2}} \xrightarrow{I\otimes I\otimes H} |x_3\rangle \otimes |x_2\rangle \otimes |x_1\rangle$$

We can generalize this idea by letting $t$ be any arbitrary positive integer greater than 3. By using the previously determined bits $x_{k+2}, \ldots, x_t$ and the action of corresponding controlled phase shift operators, we have

$$\frac{|0\rangle + e^{2\pi i 0.x_{k+1}0\ldots0}\,|1\rangle}{\sqrt{2}} = \frac{|0\rangle + (-1)^{x_{k+1}}\,|1\rangle}{\sqrt{2}}. \tag{3.6}$$

By applying a Hadamard gate to the state above, we obtain $|x_{k+1}\rangle$ at step $k$. Therefore, we can consider the inverse Fourier transform as a series of Hadamard tests. If $\varphi$ has an exact binary representation, the success probability at each step is 1. In the case that $\varphi$ cannot be exactly expressed in binary fraction, the success probability $P$ of the post-measurement state, at step $k$, is

$$P = \cos^2(\pi\theta) \quad \text{for} \quad |\theta| < \frac{1}{2^{k+1}} \tag{3.7}$$

Therefore, the success probability increases as we proceed. The following theorem gives us the success probability of the QFT algorithm.

**Theorem 9** ([KLM07]). *If $\frac{x}{2^n} \leq \varphi \leq \frac{x+1}{2^n}$, then the phase estimation algorithm returns one of $x$ or $x+1$ with probability at least $\frac{8}{\pi^2}$.*

## 3.2 Product of Two Reflections

In [SZE04], Szegedy defined a *quantum walk* as a quantum analog of a random walk of a classical Markov chain. Let $P$ be the underlying stochastic matrix for the classical random walk. Denote $p_{xy}$ as the probability of moving from state $x$ to state $y$. Since each step of the

quantum walk must be unitary, it is convenient to define the quantum walk on a quantum system with two registers $\mathcal{H} = \mathcal{H}_L \otimes \mathcal{H}_R$. The *quantum update rule*, defined in [MNRS07], is a unitary transformation $U$, based on a transition matrix $P$, that acts as

$$U \left|x\right\rangle_L \left|0\right\rangle_R = \left|x\right\rangle_L \sum_y \sqrt{p_{xy}} \left|y\right\rangle_R \tag{3.8}$$

on inputs of the form $\left|x\right\rangle_L \left|0\right\rangle_R$ for all $x \in \Omega$. (Its action on inputs $\left|x\right\rangle_L \left|y \neq 0\right\rangle_R$ can be chosen arbitrarily.) For details, please refer to Appendix A.1. Using such $U$, we define two subspaces of $\mathcal{H}$. First,

$$\mathcal{A} = \text{span}\{U \left|x\right\rangle_L \left|0\right\rangle_R\} \tag{3.9}$$

is the span of all vectors we get from acting with $U$ on $\left|x\right\rangle_L \left|0\right\rangle_R$ for all $x \in \Omega$, and second, the subspace $\mathcal{B} = S\mathcal{A}$ is the subspace we get by swapping the two registers of $\mathcal{A}$. Using the quantum update, we can implement a reflection about the subspace $\mathcal{A}$ as

$$\text{Ref}_{\mathcal{A}} = U \left(2 \left|0\right\rangle \left\langle 0\right|_R - \mathbb{I}\right) U^\dagger. \tag{3.10}$$

Szegedy defined a step of the quantum walk as the walk operator $W$

$$W = \text{Ref}_{\mathcal{B}} \cdot \text{Ref}_{\mathcal{A}}, \tag{3.11}$$

a composition of the two reflections about $\mathcal{A}$ and $\mathcal{B}$. This operation is unitary, and the state

$$\left|\psi_\pi\right\rangle = \sum_x \sum_y \sqrt{\pi_{xy}} \left|x\right\rangle_1 \left|y\right\rangle_2, \tag{3.12}$$

where $\pi$ is the stationary distribution of $P$, is an eigenvector of $W$ with eigenvalue 1. Szegedy [SZE04] proved[1] that when we parametrize the eigenvalues of $W$ as $e^{i\pi\theta_i}$, the second smallest

[1]Nagaj et al. give a simpler way to prove this relationship using Jordan's lemma in [NWZ09].

phase $\theta_1$ (after $\theta_0 = 0$) is related to the second largest eigenvalue $\lambda_1$ of $P$ as $|\theta_1| > \sqrt{1-\lambda_1}$. This can be viewed as a square-root relationship $\Delta > \sqrt{\delta}$ between the phase gap $\Delta = |\theta_1 - \theta_0|$ of the unitary operator $W$ and the spectral gap $\delta = |\lambda_0 - \lambda_1|$ of $P$. This relationship is at the heart of the quantum speedups of quantum walk-based algorithms over their classical counterparts.

Many recent quantum walk algorithms for searching [MNRS07, AMB04, AMB04a, MSS05], evaluating formulas and span programs [RS08, ACRSZ07, FGG08], quantum simulated annealing [SBBK08], quantum sampling [RIC07a, WA08] and approximating partition functions based on classical Markov chains [WCNA09] can be viewed in Szegedy's generalized quantum walk model. For all these algorithms, an essential step in implementing the quantum walk $W$ is the ability to implement the quantum update rule (3.8). For the basic search-like and combinatorial algorithms with low-degree underlying graphs, an efficient implementation of the corresponding quantum walks is straightforward. However, for complicated transition schemes coming from Markov chains like those for simulated annealing or for approximating partition functions of the Potts model, the situation is not so clear-cut. The standard polynomial speed-ups of these quantum algorithms are viewed in terms of how many times we have to apply the quantum walk operator versus the number of times we have to apply one step of the classical random walk (Markov chain). However, a finer notion of complexity including the number of elementary gates it takes to implement each step of the quantum walk is needed here. Our work addresses the question of whether it is possible

to apply the steps of these quantum walk-based algorithms efficiently enough so as not to destroy the polynomial speedups.

## 3.3 Quantization of Classical Random Walk

For better understanding the behaviour of a discrete-time quantum walk, it is crucial to compute the spectral decomposition of the walk operator. For the remainder of this chapter, let us assume $P$ is column-wise stochastic. A discrete-time classical random walk on an $N$ vertices graph is presented by an $N \times N$ transition matrix $P$. We introduce the state

$$|\psi_j\rangle = \sum_{k=1}^{N} \sqrt{P_{kj}} \, |j\rangle \, |k\rangle \tag{3.13}$$

for $j = 1, \ldots N$. The above state is normalized because $P$ is stochastic. Now let $\Pi$ denote the projection onto $\{|\psi_j\rangle \ : j = 1, \ldots, N\}$ where

$$\Pi = \sum_{j=1}^{N} |\psi_j\rangle \langle\psi_j| \, . \tag{3.14}$$

Let $S$ be the operator that swaps two registers and it is defined as

$$S = \sum_{j,k=1}^{N} |j,k\rangle \langle k,j| \, . \tag{3.15}$$

We can define a single step of the quantum walk operator as

$$U = S(2\Pi - \mathbb{I}). \tag{3.16}$$

**Theorem 10.** *[AC781] Given an $N \times N$ stochastic matrix $P$, let $\{|\lambda\rangle\}$ be a complete set of orthonormal eigenvectors of an $N \times N$ discriminant matrix $D$ where $D_{jk} = \sqrt{P_{jk}P_{kj}}$ with eigenvalues $\{\lambda\}$. Then the eigenvalues of the discrete-time quantum walk $U = S(2\Pi - \mathbb{I})$ corresponding $P$ are $\pm 1$ and $e^{\pm i arccos \lambda}$.*

*Proof.* Define an isometry

$$T = \sum_{j=1}^{N} |\psi_j\rangle \langle j| \tag{3.17}$$

that maps a state in $\mathbb{C}^n$ to $\mathbb{C}^n \otimes C^n$. We notice that

$$\begin{aligned}
TT^\dagger &= \sum_{j,k=1}^{N} |\psi_j\rangle \langle j|\, k\rangle \langle \psi_k| = \Pi && (3.18)\\
T^\dagger T &= \sum_{j,k=1}^{N} |j\rangle \langle \psi_j|\psi_k\rangle \langle k| = \sum_{j,k,l,m} \sqrt{P_{lj}P_{mk}}\, |j\rangle \langle j, l|\, k, m\rangle \langle k| && (3.19)\\
&= \sum_{j,l=1}^{N} P_{lj}\, |j\rangle \langle j| = \mathbb{I} && (3.20)\\
T^\dagger ST &= \sum_{j,k=1}^{N} |j\rangle \langle \psi_j|\, S\, |\psi_k\rangle \langle k| = \sum_{j,k,l,m} \sqrt{P_{lj}P_{mk}} |j\rangle\langle j, l| (\sum_{s,t} |s, t\rangle\langle t, s|) |k, m\rangle\langle k| && (3.21)\\
&= \sum_{j,k=1}^{N} \sqrt{P_{kj}P_{jk}} |j\rangle\langle k| = D. && (3.22)
\end{aligned}$$

Let $|\tilde{\lambda}\rangle = T\,|\lambda\rangle$. We apply $U$ to $|\tilde{\lambda}\rangle$ to get

$$U|\tilde{\lambda}\rangle = S(2TT^\dagger - \mathbb{I})T|\lambda\rangle = (2ST - ST)|\lambda\rangle = S|\tilde{\lambda}\rangle. \tag{3.23}$$

Then we apply U to $S|\tilde{\lambda}\rangle$ to get

$$US|\tilde{\lambda}\rangle = S(2TT^\dagger - \mathbb{I})ST|\lambda\rangle = (2STD - T)|\lambda\rangle = 2\lambda S|\tilde{\lambda}\rangle - |\tilde{\lambda}\rangle \tag{3.24}$$

It is clear that the subspace $\{|\tilde{\lambda}\rangle, S|\tilde{\lambda}\rangle\}$ is invariant under $U$ and that allows us to derive eigenvector within this subspace. Let $|\mu\rangle = |\tilde{\lambda}\rangle - \mu S|\mu\rangle$ be an eigenvector of $U$. We obtain the following

$$\begin{aligned} U|\mu\rangle &= S|\tilde{\lambda}\rangle - \mu(2\lambda S|\tilde{\lambda}\rangle - |\tilde{\lambda}\rangle) \\ &= \mu|\tilde{\lambda}\rangle + (1 - 2\lambda\mu)S|\tilde{\lambda}\rangle \end{aligned}$$

With the equation $-\mu^2 = -2\lambda\mu$ being satisfied, we have $\mu$ as the corresponding eigenvalue for $|\mu\rangle$. It means $\mu = \lambda \pm i\sqrt{1-\lambda^2} = e^{\pm i arccos\lambda}$.

## 3.4 Spectrum of the Product of Two Reflections

Let the subset $M$ be the set of marked elements that we are searching for. We can modify the original transition matrix $P$ into $\tilde{P}$ in the following manner:

$$\tilde{P}_{jk} = \begin{cases} 1 & k \in M \text{ and } j = k \\ 0 & k \in M \text{ and } j \neq k \\ P_{jk} & k \notin M \end{cases}$$

We can view $\tilde{P}$ in block structure as follows:

$$\tilde{P} = \begin{pmatrix} P_M & 0 \\ Q & I \end{pmatrix}. \tag{3.25}$$

When starting from a uniform distribution over unmarked elements, the probability of not reaching a marked element after $t$ steps would be

$$\frac{1}{N-|M|}\sum_{j,k\notin M}[P_M^t] \leq \|P_M\|^t.$$

Let $\|P_M\| = 1-\Delta$ then the probability of reaching a marked element after $t$ steps is at least $1-(1-\Delta)^t$. When $t = O(1/\Delta) = O(\frac{1}{1-\|P_M\|})$ is chosen, the probability of reaching a marked element approaches $\Omega(1)$.

**Lemma 1.** *[SZE04] If the second largest eigenvalue of $P$ (in absolute value) is at most $1-\delta$ and $|M| \leq \epsilon N$, then $\|P_M\| \leq 1-\delta\epsilon/2$.*

*Proof.* Let $|v\rangle \in \mathbb{R}^{N-|M|}$ be the principal eigenvector of $P_M$. Let $|w\rangle \in \mathbb{R}^N$ be the vector obtained by padding $|v\rangle$ with 0's for marked elements. Because $P$ is symmetric, the uniform vector $|V\rangle = \frac{1}{\sqrt{N}}\sum_j |j\rangle$ is its eigenvector corresponding to eigenvalue 1.

$$\begin{aligned}
\|P_M\|^2 \leq \|P|w\rangle\|^2 &= |\langle V|w\rangle|^2 + \sum_{\lambda\neq 1}\lambda^2|\langle\lambda|w\rangle|^2 && (3.26)\\
&\leq |\langle V|w\rangle|^2 + (1-\delta)\sum_{\lambda\neq 1}|\langle\lambda|w\rangle|^2 && (3.27)\\
&= 1-\delta(1-|\langle V|w\rangle|^2) && (3.28)
\end{aligned}$$

as $P = |V\rangle\langle V| + \sum_{\lambda\neq 1}\lambda|\lambda\rangle\langle\lambda|$ and $\mathbb{I} = |V\rangle\langle V| + \sum_{\lambda\neq 1}|\lambda\rangle\langle\lambda|$.

By Cauchy-Schwarze inequality,

$$|\langle V|w\rangle|^2 = |\langle V|\Pi_{V\setminus M}|w\rangle|^2 \leq \|\Pi_{V\setminus M}|V\rangle\|^2 \cdot \||w\rangle\|^2 = 1-\epsilon, \quad (3.29)$$

thus $\|P_M\| \leq \sqrt{1-\delta\epsilon} \leq 1-\delta\epsilon/2$

By the definition of discriminant matrix, we have $D = \begin{pmatrix} P_M & 0 \\ 0 & I \end{pmatrix}$. Clearly the eigenvalues of $D$ is the union of the eigenvalues of $P_M$ and 1 (from $I$). Since $\|P_M\| \leq 1 - \delta\epsilon/2$, then spectral gap of $D$ is greater than $\delta\epsilon/2$.

Let $|\psi\rangle = \frac{1}{\sqrt{N-|M|}} \sum |\psi_j\rangle$ be the initial input state. It can be easily prepared from the state $\frac{1}{\sqrt{N}} \sum |\psi_j\rangle$. Based on the cardinality of the marked elements set $M$, we have the following:

Case I: $|M| = 0$.

We have $\tilde{P} = P$ and $|\psi\rangle = \frac{1}{\sqrt{N}} \sum |\psi_j\rangle$. The phase estimation would yield phase 0 as $|\psi\rangle$ is eigenvector of $U$ with eigenvalue 1. QPE would always return 0 for the eigenphase.

Case II: $|M| > 0$.

From the spectrum theorem of quantum walk 10 the eigenvalues of $U$ are $\pm 1$ and $\lambda \pm i\sqrt{1-\lambda^2} = e^{iarccos\lambda}$ where $\lambda$ runs over the eigenvalues of $P_M$. To detect the existence of marked elements, that is to find an eigephase other than 0, we need to have QPE precision up to $O(\min_\lambda arccos\lambda)$. And since $arccos\lambda \geq \sqrt{1-\lambda}$, precision up to $\sqrt{1 - \|P_M\|}$ is sufficient. By *lemma 1*, we know $O(\frac{1}{\sqrt{\delta\epsilon}})$ invocations of the walk operator $U$ is required.

To summarize the quantization process of classical transition matrix $P$ to the corresponding quantum walk operator $U$, we can visualize the process in the following diagram:

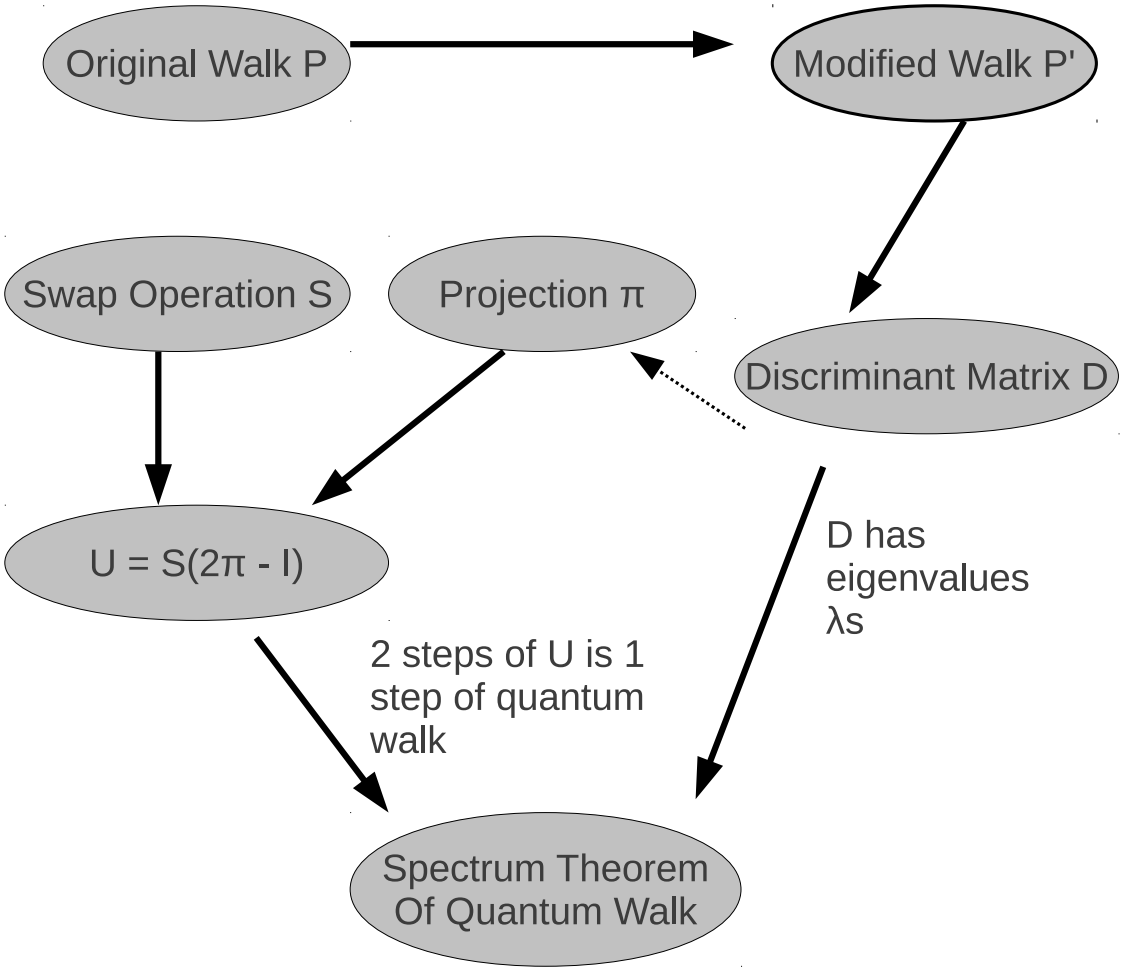


Figure 3.7: From A Random Walk to Spectrum Theorem for Quantum Walk

# CHAPTER 4
# EFFICIENT CIRCUIT FOR PHASE ESTIMATION

Quantum Phase Estimation (QPE) plays a core role in many quantum algorithms [HAL02, SHO94, SHO05, SZE04, WCNA09]. Some interesting algebraic and theoretic problems can be addressed by QPE, such as prime factorization [SHO94], discrete-log finding [SHO05], and order finding.

**Problem.** ***[Phase Estimation]*** *Let $U$ be a unitary matrix with eigenvalue $e^{2\pi i\varphi}$ and corresponding eigenvector $|u\rangle$. Assume only a single copy of $|u\rangle$ is available, the goal is to find $\widetilde{\varphi}$ such that*

$$\Pr(|\widetilde{\varphi} - \varphi| < \frac{1}{2^n}) > 1 - c, \tag{4.1}$$

*where $c$ is a constant less than $\frac{1}{2}$.*

In this chapter we investigate a more general approach for the QPE algorithm. This approach completes the transition from Kitaev's original approach that requires no controlled phase shift operators to QPE with approximate quantum Fourier transform (AQFT). The standard QPE algorithm utilizes the complete version of the inverse QFT. The disadvantage of the standard phase estimation algorithm is the high degree of phase shift operators required. Since implementing exponentially small phase shift operators is costly or physically not feasible, we need an alternative way to use lower precision operators. This was the motivation for AQFT being introduced — for lowering the cost of implementation while preserving high success probability.

In AQFT the number of required phase shift operators drops significantly with the cost of lower success probability. Such compromise demands repeating the process extra times to achieve the final result. The QPE algorithm has a success probability of at least $\frac{8}{\pi^2}$ [KLM07]. Phase estimation using AQFT instead, with phase shift operators up to degree $m$ where $m > \log_2(n) + 2$, has success probability at least $\frac{4}{\pi^2} - \frac{1}{4n}$ [BEST96, CHE04].

On the other hand, Kitaev's original approach requires only the first phase shift operator (as a single qubit gate not controlled). Comparing the existing methods, there is a gap between Kitaev's original approach and QPE with AQFT in terms of the degree of phase shift operators needed. In this chapter our goal is to fill this gap and introduce a more general phase estimation algorithm such that it is possible to realize a phase estimation algorithm with any degree of phase shift operators in hand. In physical implementation of the phase estimation algorithm the depth of the circuit should be small to avoid decoherence. Also, higher degree phase shift operators are costly to implement and in many cases it is not physically feasible.

We can visualize the precision gap in Figure 4.1. Our goal is to fill the gap by using lower precision (or constant precision) shift operators while preserving the depth of the circuit and success probability.

In this chapter, we assume only one copy of the eigenvector $|u\rangle$ is available. This implies a restriction on the use of controlled-$U$ gates that all controlled-$U$ gates should be applied on one register. Thus, the entire process is a single circuit that can not be divided into parallel processes. Due to results by Griffiths and Niu, who introduced semi-classical quantum

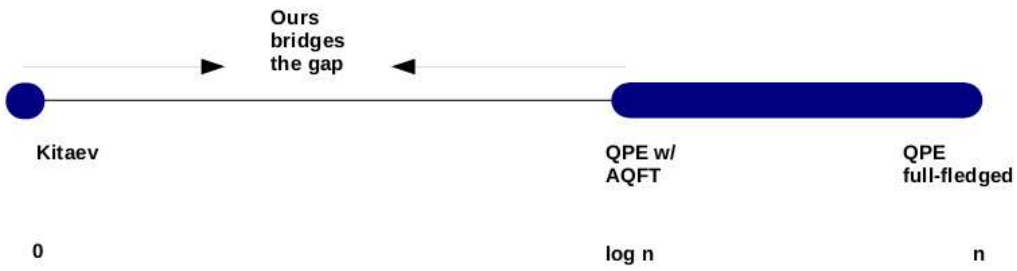


Figure 4.1: Required Precision at the Bottom: $2^0$, $2^{\log n}$ and $2^n$

Fourier transform [GN96], quantum circuits implementing different approaches discussed in this chapter would require the same number of qubits.

The structure of this chapter is organized as follows. In Sec. 4.1 we give a brief overview on existing approaches, such as Kitaev's original algorithm and standard phase estimation algorithm based on QFT and AQFT. In Sec. 4.2 we introduce our new approach and discuss the requirements to achieve the same performance output (success probability) as the methods above. Finally, we make our conclusion and compare with other methods.

## 4.1 Quantum Phase Estimation Algorithms

### 4.1.1 Kitaev's Original Approach

Kitaev's original approach is one of the first quantum algorithms for estimating the phase of a unitary matrix [KSV02]. Let $U$ be a unitary matrix with eigenvalue $e^{2\pi i\varphi}$ and corresponding

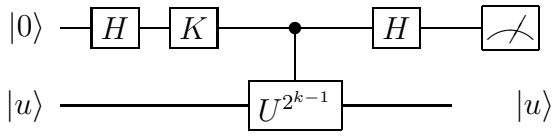


Figure 4.2: Hadamard Test with Extra Phase Shift Operator

eigenvector $|u\rangle$ such that

$$U|u\rangle = e^{2\pi i\varphi}|u\rangle. \tag{4.2}$$

In this approach, a series of Hadamard tests are performed. In each test the phase $2^{k-1}\varphi$ $(1 \leq k \leq n)$ will be computed up to precision 1/16. Assume an $n$-bit approximation is desired. Starting from $k = n$, in each step the $k$th bit position is determined consistently from the results of previous steps.

For the $k$th bit position, we perform the Hadamard test depicted in Figure 4.2, where the gate $K = I_2$. Denote $\varphi_k = 2^{k-1}\varphi$, the probability of the post measurement state is

$$\Pr(0|k) = \frac{1+\cos(2\pi\varphi_k)}{2}, \quad \Pr(1|k) = \frac{1-\cos(2\pi\varphi_k)}{2}. \tag{4.3}$$

In order to recover $\varphi_k$, we obtain more precise estimates with higher probabilities by iterating the process. But this does not allow us to distinguish between $\varphi_k$ and $-\varphi_k$. This can be solved by the same Hadamard test in Figure 4.2, but instead we use the gate

$$K = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}. \tag{4.4}$$

The probabilities of the post-measurement states based on the modified Hadamard test become

$$\Pr(0|k) = \frac{1-\sin(2\pi\varphi_k)}{2}, \quad \Pr(1|k) = \frac{1+\sin(2\pi\varphi_k)}{2}. \tag{4.5}$$

Hence, we have enough information to recover $\varphi_k$ from the estimates of the probabilities.

In Kitaev's original approach, after performing the Hadamard tests, some classical post processing is also necessary. Suppose $\varphi = 0.x_1x_2\ldots x_n$ is an exact $n$-bit. If we are able to determine the values of $\varphi$, $2\varphi, \ldots, 2^{n-1}\varphi$ with some constant-precision (1/16 to be exact), then we can determine $\varphi$ with precision $1/2^n$ efficiently [KIT95, KSV02].

Starting with $\varphi_n$ we increase the precision of the estimated fraction as we proceed toward $\varphi_1$. The approximated values of $\varphi_k\,(k = n, \ldots, 1)$ will allow us to make the right choices.

For $k = 1, \ldots, n$ the value of $\varphi_k$ is replaced by $\beta_k$, where $\beta_k$ is the closest number chosen from the set $\{\frac{0}{8}, \frac{1}{8}, \frac{2}{8}, \frac{3}{8}, \frac{4}{8}, \frac{5}{8}, \frac{6}{8}, \frac{7}{8}\}$ such that

$$|\varphi_k - \beta_k|_{\text{mod } 1} < \frac{1}{8}. \tag{4.6}$$

The result follows by a simple iteration. Let $\beta_n = \overline{0.x_nx_{n+1}x_{n+2}}$ and proceed by the following iteration:

$$x_k = \begin{cases} 0 & \text{if } \overline{0.0x_{k+1}x_{k+2}} - \beta_k|_{\text{mod } 1} < 1/4 \\ 1 & \text{if } \overline{0.1x_{k+1}x_{k+2}} - \beta_k|_{\text{mod } 1} < 1/4 \end{cases} \tag{4.7}$$

for $k = n-1, \ldots, 1$. By using simple induction, the result satisfies the following inequality:

$$|\overline{0.x_1x_2\ldots x_{n+2}} - \varphi|_{\text{mod } 1} < 2^{-(n+2)}. \tag{4.8}$$

In Eq. 4.6, we do not have the exact value of $\varphi_k$. So, we have to estimate this value and use the estimate to find $\beta_k$. Let $\widetilde{\varphi_k}$ be the estimated value and

$$\epsilon = |\widetilde{\varphi_k} - \varphi_k|_{\text{mod } 1} \tag{4.9}$$

be the estimation error. Now we use the estimate to find the closest $\beta_k$. Since we know the exact binary representation of the estimate $\widetilde{\varphi_k}$, we can choose $\beta_k$ such that

$$|\widetilde{\varphi_k} - \beta_k|_{\text{mod } 1} \leq \frac{1}{16}. \tag{4.10}$$

By the triangle inequality we have,

$$|\varphi_k - \beta_k|_{\text{mod } 1} \leq |\widetilde{\varphi_k} - \varphi_k|_{\text{mod } 1} + |\widetilde{\varphi_k} - \beta_k|_{\text{mod } 1} \leq \epsilon + \frac{1}{16}. \tag{4.11}$$

To satisfy Eq. 4.6, we need to have $\epsilon < 1/16$, which implies

$$|\widetilde{\varphi_k} - \varphi_k|_{\text{mod } 1} < \frac{1}{16}. \tag{4.12}$$

Therefore, it is required for the phase to be estimated with precision $1/16$ at each stage.

In the first Hadamard test (Eq. 4.3), in order to estimate $\Pr(1|k)$ an iteration of Hadamard tests should be applied to obtain the required precision of $1/16$ for $\varphi_k$. This is done by counting the number of states $|1\rangle$ in the post measurement state and dividing that number by the total number of iterations performed.

The Hadamard test outputs $|0\rangle$ or $|1\rangle$ with a fixed probability. We can model an iteration of Hadamard tests as Bernoulli trials with success probability (obtaining $|1\rangle$) being $p_k$. The best estimate for the probability of obtaining the post measurement state $|1\rangle$ with $t$ samples

is

$$\widetilde{p_k} = \frac{h}{t}, \tag{4.13}$$

where $h$ is the number of ones in $t$ trials. This can be proved by Maximum Likelihood Estimation (MLE) methods [HS98].

In order to find $\sin(2\pi\varphi_k)$ and $\cos(2\pi\varphi_k)$, we can use estimates of probabilities in Eq. 4.3 and EQ. 4.5. Let $s_k$ be the estimate of $\sin(2\pi\varphi_k)$ and $t_k$ the estimate of $\cos(2\pi\varphi_k)$. It is clear that if

$$|\widetilde{p_k} - p_k| < \epsilon_0, \tag{4.14}$$

then

$$|s_k - \sin(2\pi\varphi_k)| < 2\epsilon_0, \quad |t_k - \cos(2\pi\varphi_k)| < 2\epsilon_0. \tag{4.15}$$

Since the inverse tangent function is more robust to error than the inverse sine or cosine functions, we use

$$\widetilde{\varphi_k} = \frac{1}{2\pi} \arctan\left(\frac{s_k}{t_k}\right) \tag{4.16}$$

as the estimation of $\varphi_k$. By Eq. 4.12 we should have

$$\left|\varphi_k - \frac{1}{2\pi} \arctan\left(\frac{s_k}{t_k}\right)\right|_{\text{mod } 1} < \frac{1}{16}. \tag{4.17}$$

The inverse tangent function can not distinguish between the two values $\varphi_k$ and $\varphi_k \pm 1/2$. However, because we find estimates of the sine and cosine functions as well, it is easy to determine the correct value. The inverse tangent function is most susceptible to error when $\varphi_k$ is in the neighborhood of zero and the reason is that the derivative is maximized at zero.

Thus, if

$$|s_k - \sin(2\pi\varphi_k)| = \epsilon_1 \quad \text{and} \quad |t_k - \cos(2\pi\varphi_k)| = \epsilon_2, \tag{4.18}$$

considering the case where $\varphi_k = 0$, then we have

$$\frac{1}{2\pi}\left|\arctan\left(\frac{\epsilon_1}{1 \pm \epsilon_2}\right)\right| < \frac{1}{16}. \tag{4.19}$$

By simplifying the above inequality, we have

$$\left|\frac{\epsilon_1}{1 \pm \epsilon_2}\right| < \tan(\frac{\pi}{8}). \tag{4.20}$$

With the following upper bounds for $\epsilon_1$ and $\epsilon_2$, the inequality above is always satisfied when

$$|\epsilon_1| < 1 - \frac{1}{\sqrt{2}} \quad \text{and} \quad |\epsilon_2| < 1 - \frac{1}{\sqrt{2}}. \tag{4.21}$$

Therefore, in order to estimate the phase $\varphi_k$ with precision $1/16$, the probabilities in Eq. 4.3 and Eq. 4.5 should be estimated with error at most $(2-\sqrt{2})/4$ which is approximately 0.1464. In other words, it is necessary to find the estimate of $\Pr(1|k)$ such that

$$\left|\Pr(1|k) - \frac{h}{t}\right| < \frac{2-\sqrt{2}}{4} \approx 0.1464. \tag{4.22}$$

There are different ways we can guarantee an error bound with constant probability. The first method, used in [KSV02], is based on the Chernoff bound. Let $X_1, \ldots, X_m$ be Bernoulli random variables. By Chernoff's bound we have

$$\Pr\left(\left|\frac{1}{m}\sum_{i=0}^{m} X_i - p_k\right| \geq \delta\right) \leq 2e^{-2\delta^2 m}, \tag{4.23}$$

where in our case the estimate is $\widetilde{p_k} = \frac{1}{m}\sum_{i=0}^{m} X_i$. Since we need an accuracy up to 0.1464, we get

$$\Pr\left(|\widetilde{p_k} - p_k| > 0.1464\right) < 2e^{-(0.0429)m}. \tag{4.24}$$

In order to obtain

$$\Pr\left(|\widetilde{p_k} - p_k| < 0.1464\right) > 1 - \frac{\varepsilon}{2}, \tag{4.25}$$

a minimum of $m_1$ trials is sufficient when

$$\begin{aligned} m_1 &\approx 24 \ln \frac{4}{\varepsilon} \\ &\approx 33 + 24 \ln \frac{1}{\varepsilon} \end{aligned} \tag{4.26}$$

This is the number of trials for each Hadamard test, as we have two Hadamard tests at each stage. Therefore, in order to have

$$\Pr\left(|\widetilde{\varphi_k} - \varphi_k| < \frac{1}{16}\right) > 1 - \varepsilon. \tag{4.27}$$

we require a minimum of

$$\begin{aligned} m &= 2m_1 \\ &\approx 47 \ln \frac{4}{\varepsilon} \\ &\approx 66 + 47 \ln \frac{1}{\varepsilon} \end{aligned} \tag{4.28}$$

many trials.

In the analysis above, we used the Chernoff bound, which is not a tight bound. If we want to obtain the result with a high probability, we need to apply a large number of Hadamard tests. In this case, we can use an alternative method to analyze the process by employing methods of statistics [SIV96].

Iterations of Hadamard tests have a binomial distribution which can be approximated by a normal distribution. This is a good approximation when $p$ is close to $1/2$ or $mp > 10$ and

$m(1-p) > 10$, where $m$ is the number of iterations and $p$ the success probability. In other words, if we see 10 successes and 10 failures in our process, we can use this approximation to obtain a better bound.

In Kitaev's algorithm each Hadamard test has to be repeated a sufficient number of times to achieve the required accuracy with high probability. Because only one copy of $|u\rangle$ is available, all controlled-$U$ gates have to be applied to one register. Therefore, all the Hadamard tests have to be performed in sequence, instead of parallel, during one run of the circuit. A good example for this case is the order finding algorithm. We refer the reader to [NC00] for more details.

In Kitaev's approach, there are $n$ different Hadamard tests that should be performed. Thus, if the probability of error in each Hadamard test is $\varepsilon_0$, by applying the union bound, the error probability of the entire process is $\varepsilon = n\varepsilon_0$. Therefore, in order to obtain

$$\Pr(|\varphi - \widetilde{\varphi}| < \frac{1}{2^n}) > 1 - \varepsilon, \tag{4.29}$$

for approximating each bit we need $m$ trials where

$$m = 47 \ln \frac{4n}{\varepsilon}. \tag{4.30}$$

Since all of these trials have to be done in one circuit, the circuit consists of $mn$ Hadamard tests. Therefore the circuit involves $mn$ controlled-$U^{2^k}$ operations. As a result, if a constant success probability is desired, the depth of the circuit will be $O(n \log n)$.

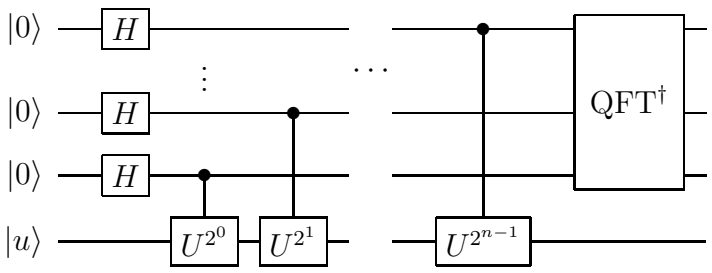


Figure 4.3: Standard Quantum Phase Estimation

### 4.1.2 Approach Based on QFT

One of the standard methods to approximate the phase of a unitary matrix is QPE based on QFT. The structure of this method is depicted at Figure 4.3. The QPE algorithm requires two registers and contains two stages. If an $n$-bit approximation of the phase $\varphi$ is desired, then the first register is prepared as a composition of $n$ qubits initialized in the state $|0\rangle$. The second register is initially prepared in the state $|u\rangle$. The first stage prepares a uniform superposition over all possible states and then applies controlled-$U^{2^k}$ operations. Consequently, the state will become

$$\frac{1}{2^{n/2}} \sum_{k=0}^{2^n-1} e^{2\pi i \varphi k} |k\rangle. \tag{4.31}$$

The second stage in the QPE algorithm is the QFT$^\dagger$ operation.

There are different ways to interpret the inverse Fourier transform. In the QPE algorithm, the post-measurement state of each qubit in the first register represents a bit in the final approximated binary fraction of the phase. Therefore, we can consider computing each bit as a step. The inverse Fourier transform can be interpreted such that at each step (starting

from the least significant bit), using the information from previous steps, it transforms the state

$$\frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^k \varphi}|1\rangle) \tag{4.32}$$

to get closer to one of the states

$$\begin{aligned} \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.0}|1\rangle) &= \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \\ \text{or} & \\ \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 0.1}|1\rangle) &= \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \end{aligned} \tag{4.33}$$

Assume we are at step $k$ in the first stage. By applying controlled-$U^{2^k}$ operators due to phase kick back, we obtain the state

$$\frac{|0\rangle + e^{2\pi i 0.x_{k+1}x_{k+2}\ldots x_n}|1\rangle}{\sqrt{2}}. \tag{4.34}$$

Shown in Figure 4.4, each step (dashed-line box) uses the result of previous steps, where phase shift operators are defined as

$$R_k \equiv \begin{bmatrix} 1 & 0 \\ 0 & e^{2\pi i/2^k} \end{bmatrix} \tag{4.35}$$

for $2 \leq k \leq n$.

By using the previously determined bits $x_{k+2}, \ldots, x_n$ and the action of corresponding controlled phase shift operators (as depicted in Figure 4.4) the state in Eq. 4.34 becomes

$$\frac{|0\rangle + e^{2\pi i 0.x_{k+1}0\ldots 0}|1\rangle}{\sqrt{2}} = \frac{|0\rangle + (-1)^{x_{k+1}}|1\rangle}{\sqrt{2}}. \tag{4.36}$$

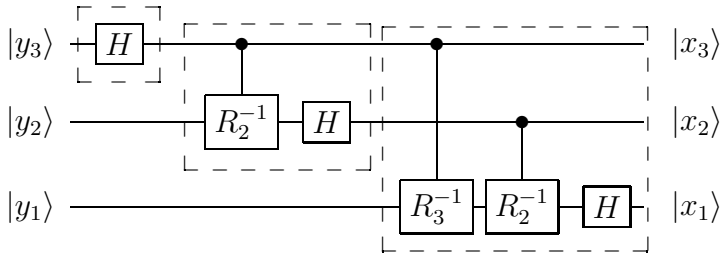


Figure 4.4: 3-qubit Inverse QFT Where $1 \leq i \leq 3$, $|y_i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_i \ldots x_3)} |1\rangle)$

Thus, by applying a Hadamard gate to the state above we obtain $|x_{k+1}\rangle$. Therefore, we can consider the inverse Fourier transform as a series of Hadamard tests.

If $\varphi$ has an exact $n$-bit binary representation the success probability at each step is 1. While, in the case that $\varphi$ cannot be exactly expressed in $n$-bit binary fraction, the success probability $P$ of the post-measurement state, at step $k$, is

$$P = \cos^2(\pi\theta) \quad \text{for} \quad |\theta| < \frac{1}{2^{k+1}} \tag{4.37}$$

Detailed analysis obtaining similar probabilities are given in Sec. 4.2.

Therefore, the success probability increases as we proceed. The following theorem gives us the success probability of the QFT algorithm.

**Theorem 11** ([KLM07]). *If $\frac{x}{2^n} \leq \varphi \leq \frac{x+1}{2^n}$, then the phase estimation algorithm returns one of $x$ or $x+1$ with probability at least $\frac{8}{\pi^2}$.*

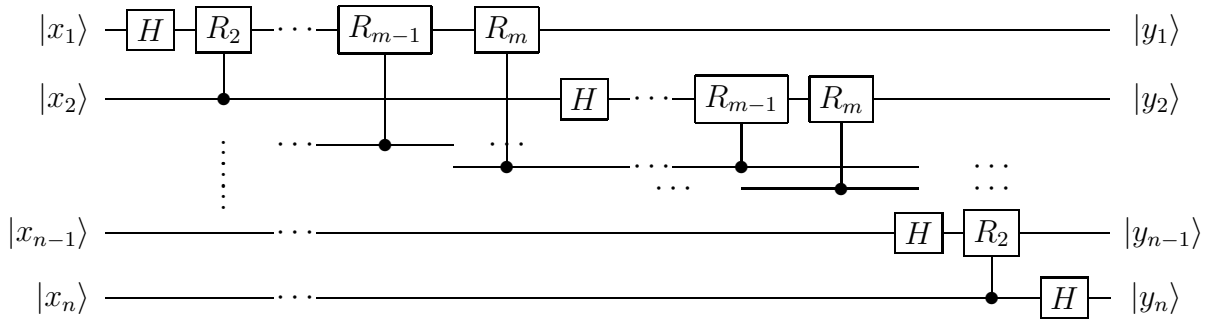


Figure 4.5: Quantum Circuit for AQFT

### 4.1.3 Approach Based on AQFT

AQFT was first introduced by Barenco, et al [BEST96]. It has an advantage in algorithms that involve periodicity estimation. Its structure is similar to regular QFT but differs by eliminating higher precision phase shift operators. The circuit of AQFT is shown in Figure 4.5. At the RHS of the circuit, for $n - m < i \leq n$

$$|y_i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_i \ldots x_n)} |1\rangle) \tag{4.38}$$

and for $1 < i \leq n - m$,

$$|y_i\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_i \ldots x_{i+m-1})} |1\rangle). \tag{4.39}$$

Let $0.x_1x_2 \ldots x_n$ be the binary representation of eigenphase $\varphi$. For estimating each $x_p$, where $1 \leq p \leq n$, AQFT$_m$ requires at most $m$ phase shift operations. Here $m$ is defined as the degree of the AQFT$_m$.

Therefore, phase shift operations in AQFT$_m$ requires precision up to $e^{2\pi i/2^m}$. The probability $P$ of gaining an accurate output using AQFT$_m$, when $m \geq \log_2 n + 2$, is at least

[BEST96]

$$P \geq \frac{8}{\pi^2}(\sin^2(\frac{\pi}{4}\frac{m}{n})). \tag{4.40}$$

The accuracy of $\text{AQFT}_m$ approaches the lower bound for the accuracy of the full QFT, which is $\frac{8}{\pi^2}$. A better lower bound is also achieved by Cheung in [CHE04]

$$P \geq \frac{4}{\pi^2} - \frac{1}{4n}. \tag{4.41}$$

Moreover, this indicates that the logarithmic-depth AQFT provides an alternative approach to replace the regular QFT in many quantum algorithms. The total number of the phase shift operator invocations in $\text{AQFT}_m$ is $O(n \log_2 n)$, instead of $O(n^2)$ in the QFT. The phase shift operator precision requirement is only up to $e^{2\pi i/4n}$, instead of $e^{2\pi i/2^n}$.

By using the AQFT instead of the QFT we trade off smaller success probability with smaller degrees of phase shift operators and a shorter circuit.

## 4.2 New Approach with Constant Degree Phase Shift Operators

In this section we introduce our new approach for QPE. Our approach draws a trade-off between the highest degree of phase shift operators being used and the depth of the circuit. As a result, when smaller degrees of phase shift operators are used, the depth of the circuit increases and vice versa.

As pointed out in Sec. 4.1.2, by using information of previous qubits, the full-fledged inverse QFT transforms the phase such that the phase of the corresponding qubit gets closer

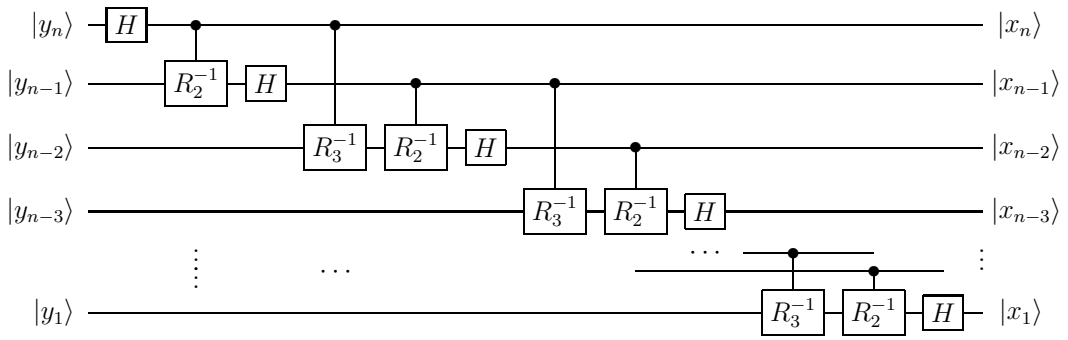


Figure 4.6: QPE with Only Two Controlled Phase Shift Operations

to one of the states $|+\rangle$ or $|-\rangle$. For our approach, we first consider the case where only the controlled phase shifts operators $R_2$ and $R_3$ are used (Eq. 4.35). In this case, we only use the information of the two previous qubits (see Figure 4.6). In such a setting, we show that it is possible to perform the QPE algorithm with arbitrary success probability.

The first stage of our algorithm is similar to the first stage of QPE based on QFT. Assume the phase is $\varphi = 0.x_1x_2x_3\ldots$ with an infinite binary representation. At step $k$, the phase after the action of the controlled gate $U^{2^k}$ is $2^k\varphi = 0.x_{k+1}x_{k+2}\ldots$ and the corresponding state is

$$|\psi_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i 2^k\varphi}\,|1\rangle). \tag{4.42}$$

By applying controlled phase shift operators $R_2$ (controlled by the $(k-1)$th qubit) and $R_3$ (controlled by the $(k-2)$th qubit) to the state above, we obtain

$$\left|\widetilde{\psi_k}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i\widetilde{\varphi}}\,|1\rangle), \tag{4.43}$$

where

$$\widetilde{\varphi} = 0.x_{k+1}00x_{k+4}\ldots. \tag{4.44}$$

It is easy to see that

$$|\widetilde{\varphi} - 0.x_{k+1}| < \frac{1}{8}. \tag{4.45}$$

Hence, we can express

$$\widetilde{\varphi} = 0.x_{k+1} + \theta \tag{4.46}$$

where $|\theta| < \frac{1}{8}$. Therefore, the state $\left|\widetilde{\psi_k}\right\rangle$ can be rewritten as

$$\left|\widetilde{\psi_k}\right\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_{k+1}+\theta)}\,|1\rangle). \tag{4.47}$$

In order to approximate the phase $\varphi$ at this stage ($k$th step), we need to find the value of $x_{k+1}$ by measuring the $k$th qubit. In this regard, we first apply a Hadamard gate before the measurement to the state $\left|\widetilde{\psi_k}\right\rangle$. The post-measurement state will determine the value of $x_{k+1}$ correctly with high probability. The post measurement probabilities of achieving $|0\rangle$ or $|1\rangle$ in the case where $x_{k+1} = 0$ is

$$\begin{aligned} \Pr(0|k) &= \cos^2(\pi\theta) \\ \Pr(1|k) &= \sin^2(\pi\theta). \end{aligned} \tag{4.48}$$

Therefore,

$$\begin{aligned} \Pr(0|k) &\geq \cos^2(\frac{\pi}{8}) \approx 0.85 \\ \Pr(1|k) &\leq \sin^2(\frac{\pi}{8}) \approx 0.15 \end{aligned} \tag{4.49}$$

In the case where $x_{k+1} = 1$, the success probability is similar.

By iterating this process a sufficient number of times and then letting the majority decide, we can achieve any desired accuracy. The analysis is similar to Sec. 4.1.1. In this case, all we

require is to find the majority. Therefore, by a simple application of the Chernoff's bound

$$\Pr\left(\frac{1}{m}\sum_{i=0}^{m} X_i \leq \frac{1}{2}\right) \leq e^{-2m(p-\frac{1}{2})^2}, \tag{4.50}$$

where in this case $p = \cos^2(\pi/8)$. It is easy to see that if a success probability of $1-\varepsilon$ is required, then we need at least

$$m = 4\ln(\frac{1}{\varepsilon}) \tag{4.51}$$

many trials for approximating each bit.

By comparing Eq. 4.30 and Eq. 4.51 (Table 4.1), we see that while preserving the success probability, our new algorithm differs by a constant and scales about 12 times better than Kitaev's original approach in terms of the number of Hadamard tests required (Figure 4.7). In physical implementations this is very important, especially in the case where only one copy of the eigenvector $|u\rangle$ is available and all Hadamard tests should be performed during one run of the circuit.

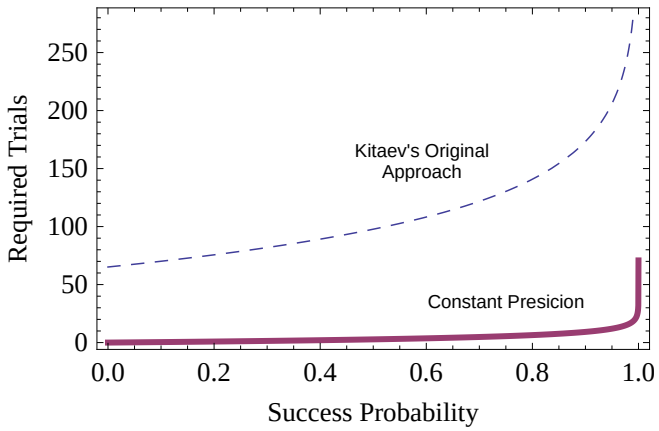


Figure 4.7: Required Trials Comparison between ours and Kitaev's

## 4.3 Discussion

In the algorithm introduced above, only phase shift operators $R_2$ and $R_3$ are used. When higher phase shift operators are used in our algorithm, the success probability of each Hadamard test will increase. As a result, fewer trials are required in order to achieve similar success probabilities. As pointed out in Sec. 4.1.3, the QPE based on AQFT requires phase shift operators of degree at least $2 + \log n$. With this precision of phase shift operators in hand, the success probability at each step would be high enough such that there is no need to iterate each step. In such scenario, one trial is sufficient to achieve an overall success probability of a constant.

Table 4.1: Required Trials by Using Chernoff's Bound

| **S**uccess **P**robability | **K**itaev's **O**riginal Approach | **C**onstant **P**recision |
|---|---|---|
| 0.50000 | 98 | 3 |
| 0.68269 | 120 | 5 |
| 0.95450 | 211 | 13 |
| 0.99730 | 344 | 24 |
| 0.99993 | 515 | 39 |

Recall the phase estimation problem stated in the introduction. If a constant success probability greater than $\frac{1}{2}$ is required, the depth of the circuit for all the methods mentioned

in this chapter (except the QPE based on full fledged QFT, which is $O(n^2)$), would be $O(n \log n)$ (assuming the cost of implementing the controlled-$U^{2^k}$ gates are all the same). This means the depth of the circuits differ only by a constant. However, the disadvantage of Kitaev's original approach to our new approach is the large number of Hadamard tests required for each bit in the approximated fraction.

Therefore, the new method introduced in this chapter provides the flexibility of using any available degree of controlled phase shift operators while preserving the success probability and the length of the circuit up to a constant. The gap is thus filled by using lower precision (or constant precision) shift operators while the depth of the circuit and success probability are preserved as shown in Figure 4.8.

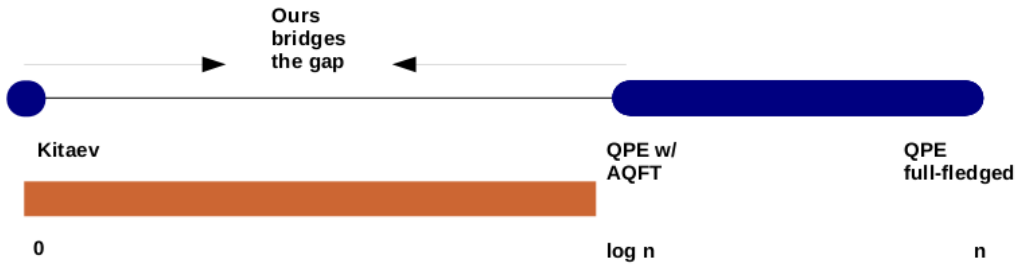


Figure 4.8: With Variable Reset Bit Chosen, Our Approach Bridges the Gap

# CHAPTER 5
# EFFICIENT CIRCUIT FOR QUANTUM WALKS

For many tasks, such as simulated annealing [KGV83, CER85], computing the volume of convex bodies [LV06] and approximating the permanent of a matrix [JSV04, BSVV08] (see references in [WCNA09] for more), the best approaches known today are randomized algorithms based on Markov chains (random walks) and sampling. Since quantum walk provides quadratic speed-up over its classical counter part, it is natural to devise a circuit implementing the quantum walk operator.

We present an efficient general method for realizing a quantum walk operator corresponding to an arbitrary sparse classical random walk. Our approach is based on Grover and Rudolph's method for preparing coherent versions of efficiently integrable probability distributions [GR02]. This method is intended for use in quantum walk algorithms with polynomial speedups, whose complexity is usually measured in terms of how many times we have to apply a step of a quantum walk [SZE04], compared to the number of necessary classical Markov chain steps. We consider a finer notion of complexity including the number of elementary gates it takes to implement each step of the quantum walk with some desired accuracy. The difference in complexity for various implementation approaches is that our method scales linearly in the sparsity parameter and poly-logarithmically with the inverse of the desired precision. The best previously known general methods either scale quadratically in the sparsity parameter, or polynomially in the inverse precision. Our approach is especially relevant for implementing quantum walks corresponding to classical random walks

like those used in the classical algorithms for approximating permanents [JSV04, BSVV08] and sampling from binary contingency tables [BSSV06]. In those algorithms, the sparsity parameter grows with the problem size, while maintaining high precision is required.

## 5.1 Introduction

In Section 5.2, we review the recent alternative approaches to the implementation of $U$, such as those relying on efficient simulation of sparse Hamiltonians [BACS07]. We find that they either scale quadratically in the sparsity parameter $d$, or polynomially in $\frac{1}{\epsilon}$, where $\epsilon$ is the allowed error in the implementation of $U$. When there is only a small number of neighbors connected to each state $x$, or we do not need to use many steps of the quantum walk so that we can tolerate more implementation error, one could use these methods. However, the subtle algorithms like [WCNA09] require many precise uses of $U$ which couple many (a number growing with the system size) neighboring states. In Appendix B.6 we show a particular example (a first step towards a possible future quantum version of the classical algorithm for approximating the permanent [JSV04, BSVV08]), where the alternative approaches to $U$ destroy the polynomial speedup of the quantum algorithm. This is why we developed our new method, scaling linearly in the sparsity parameter $d$ and polynomially in $\log \frac{1}{\epsilon}$.

Our general approach to the implementation of quantum walks based on sparse classical Markov chains is based on Grover and Rudolph's method of preparing states corresponding to efficiently integrable probability distributions [GR02]. In our case, the quantum samples

we need to prepare correspond to probability distributions that are supported on at most $d$ states of $\mathcal{E}$, which implies that they are efficiently integrable. Thus, we can use the method [GR02] to obtain an efficient circuit for the quantum update. The basic trick underlying Grover and Rudolph's method, preparing superpositions by subsequent rotations, was first proposed by Zalka [ZAL98]. Note that Childs [AC08], investigating the relationship between continuous-time [FG98] and discrete-time [KEM03] quantum walks, also proposed to use [GR02], also for some quantum walks with non-sparse underlying graphs.

This is our main result about the quantum update rule $U$, the essential ingredient in the implementation of the quantum walk defined as the quantum analogue of the original Markov chain:

**Theorem 12.** *(An Efficient Quantum Update Rule) Consider a reversible Markov chain on the state space $\mathcal{E}$, with $|\mathcal{E}| = 2^m$, with a transition matrix $P = (p_{xy})_{x,y\in\mathcal{E}}$. Assume that*

1. *there are at most $d$ possible transitions from each state ($P$ is sparse),*
2. *the transition probabilities $p_{xy}$ are given with $t$-bit precision, with $t = \Omega\left(\log\frac{1}{\epsilon} + \log d\right)$,*
3. *we have access to a reversible circuit returning the list of (at most $d$) neighbors of the state $x$ (according to $P$), which can be turned into an efficient quantum circuit $N$:*

$$N\left|x\right\rangle\left|0\right\rangle\cdots\left|0\right\rangle = \left|x\right\rangle\left|y_0^x\right\rangle\cdots\left|y_{d-1}^x\right\rangle, \tag{5.1}$$

4. *we have access to a reversible circuit which can be turned into an efficient quantum circuit $T$ acting as*

$$T|x\rangle|0\rangle\cdots|0\rangle = |x\rangle\left|p_{xy_0^x}\right\rangle\cdots\left|p_{xy_{d-1}^x}\right\rangle. \tag{5.2}$$

*Then there exists an efficient quantum circuit $\tilde{U}$ simulating the quantum update rule*

$$U|x\rangle|0\rangle = |x\rangle\sum_y \sqrt{p_{xy}}\,|y\rangle, \tag{5.3}$$

*where the sum over $y$ is over the neighbors of $x$, and $p_{xy}$ are the elements of $P$, with precision*

$$\left\|\left(U-\tilde{U}\right)|x\rangle\otimes|0\rangle\right\| \leq \epsilon \tag{5.4}$$

*for all $x\in\mathcal{E}$, with required resources scaling linearly in $m$, polynomially in $\log\frac{1}{\epsilon}$ and linearly in $d$ (with an additional poly$(\log d)$ factor).*

In Section 5.2, we describe the alternative approaches one could take to implement the quantum update and discuss their efficiency. In Section 5.3 we present our algorithm based on Grover & Rudolph's state preparation method. We conclude our discussion in Section 7.3. In Appendix B.1, we give an example where our approach is better than the alternative methods, and finally, we present the remaining details for the quantum update circuit, its required resources, and its implementation in Appendix B.1.

## 5.2 Alternative Ways of Implementing the Quantum Update

Before we give our efficient method, we review the alternative approaches in more detail. We know of three other ways how one could think of implementing the quantum update.

The first two are based on techniques for simulating Hamiltonian time evolutions, while the third uses a novel technique for implementing combinatorially block-diagonal unitaries.

The first method is to directly realize the reflection $\mathrm{Ref}_{\mathcal{A}}$ as $\exp(-i\Pi_{\mathcal{A}}\tau)$ for time $\tau = \frac{\pi}{2}$, where the projector $\Pi_{\mathcal{A}}$ onto the subspace $\mathcal{A}$ turns out to be a sparse Hamiltonian. Observe that the projector

$$\Pi_{\mathcal{A}} = \sum_{x\in\mathcal{E}} |x\rangle\langle x| \otimes \sum_{y,y'\in\mathcal{E}} \sqrt{p_{xy}}\sqrt{p_{xy'}}|y\rangle\langle y'|$$

is a sparse Hamiltonian provided that $P$ is sparse. Thus, we can approximately implement the reflection $\mathrm{Ref}_{\mathcal{A}}$ by simulating the time evolution according to $H = \Pi_{\mathcal{A}}$ for the time $\tau = \frac{\pi}{2}$. The same methods apply to the reflection $\mathrm{Ref}_{\mathcal{B}}$, so we can approximately implement the quantum walk $W(P)$, which is a product of these two reflections. The requirements of this method scale *polynomially* in $\frac{1}{\epsilon}$, where $\epsilon$ is the desired accuracy of the unitary quantum update. Moreover, the number of gates used in each $U$ scales at least linearly with $d$ and $m$.

The second approach is to apply novel general techniques for implementing arbitrary row-and-column-sparse unitaries, due to Childs [AC09] and Jordan and Wocjan [JW09]. Similarly to the first method, it relies on simulating a sparse Hamiltonian for a particular time. However, the complexity of this method again scales *polynomially* in $\frac{1}{\epsilon}$ (and linearly in $d$ and $m$).

The third alternative is to utilize techniques for implementing combinatorially block-diagonal unitary matrices. A (unitary) matrix $M$ is called combinatorially block-diagonal if

there exists a permutation matrix $P$ (i.e., a unitary matrix with entries 0 and 1) such that

$$PMP^{-1} = \bigoplus_{b=1}^{B} M_b$$

and the sizes of the blocks $M_b$ are bounded from above by some small $d$. The method works as follows: each $x \in \mathcal{E}$ can be represented by the pair $\{b(x), p(x)\}$, where $b(x)$ denotes the block number of $x$ and $p(x)$ denotes the position of $x$ inside the block $b(x)$. The unitary $M$ can then be realized by

1. the basis change $|x\rangle \mapsto |b(x)\rangle \otimes |p(x)\rangle$,

2. the controlled operation $\sum_{b=1}^{B} |b\rangle\langle b| \otimes M_b$ , and

3. the basis change $|b(x)\rangle \otimes |p(x)\rangle \mapsto |x\rangle$.

The transformations $M_b$ can be implemented using $O(d^2)$ elementary gates based on the decomposition of unitaries into a product of two-level matrices [RZBB94]. The special case $d = 2$ is worked out in the paper by Aharonov and Ta-Shma [AT03]. The reflection $\mathrm{Ref}_{\mathcal{A}} = 2\Pi_{\mathcal{A}} - \mathbb{I}$ then has the form

$$\mathrm{Ref}_{\mathcal{A}} = \sum_{x \in \mathcal{E}} |x\rangle\langle x| \otimes \left( \sum_{y,y' \in \mathcal{E}} \sqrt{p_{xy}}\sqrt{p_{xy'}} |y\rangle\langle y'| - \delta_{y,y'} \right) ,$$

where $\delta_{y,y'} = 1$ for $y = y'$ and 0 otherwise. Viewed in this form, we see that $\mathrm{Ref}_{\mathcal{A}}$ is a combinatorially block-diagonal unitary matrix, with a block decomposition with respect to the 'macro' coordinate $x$. Inside each 'macro' block labeled by $x$, we obtain a 'micro' block of size $d$ corresponding to all $y$ with $p_{xy} > 0$ and many 'micro' blocks of size 1 corresponding to all $y$ with $p_{xy} = 0$ after a simple permutation of the rows and columns. The disadvantage

of this way of implementing quantum walks is that its complexity scales *quadratically* with $d$ (and linearly in $m$ and $\log \frac{1}{\epsilon}$), the maximum number of neighbors for each state $x$.

In the next Section, we show how to implement the quantum update rule by a circuit with the number of operations scaling *linearly* with the sparsity parameter $d$ (with additional poly($\log d$) factors), linearly in $m = \log |\mathcal{E}|$ and *polynomially* in $\log \frac{1}{\epsilon}$.

## 5.3 Overview of the Efficient Circuit Structure

Our efficient circuit for the Quantum Update Rule

$$U |x\rangle_L |0\rangle_R = |x\rangle_L \sum_{i=0}^{d-1} \sqrt{p_{xy_i^x}} |y_i^x\rangle_R \tag{5.5}$$

works in the following way:

1. Looking at $x$ in the 'left' register, put a list of its (at most $d$) neighbors $y_i^x$ into an extra register and the corresponding transition probabilities $p_{xy_i^x}$ into another extra register.

2. Using the list of probabilities, prepare the superposition

$$\sum_{i=0}^{d-1} \sqrt{p_{xy_i^x}} |i\rangle_S \tag{5.6}$$

   in an extra 'superposition' register $S$.

3. Using the list of neighbors, put $\sum_{i=0}^{d-1} \sqrt{p_{xy_i^x}} |y_i^x\rangle_R |i\rangle_S$ in the registers $R$ and $S$.

4. Clean up the $S$ register using the list of neighbors of $x$ and uncompute the transition probability list and the neighbor list.

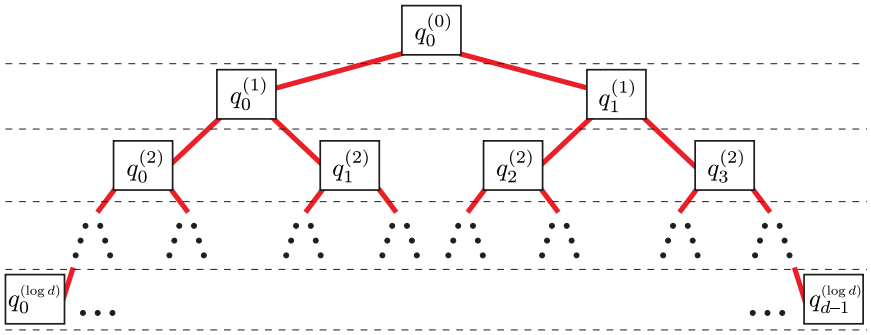


Figure 5.1: The Scheme for Preparing the Superposition in $\log d$ Rounds

We already assumed we can implement Step 1 of this algorithm efficiently. The second, crucial step is described in Section 5.4. Additional details for steps 3 and 4 are spelled out in Appendix B.1. Finally, the cleanup step 4 is possible because of the unitarity of step 1.

## 5.4 Preparing Superpositions à la Grover and Rudolph

The main difficulty is the efficient preparation of (5.6). We start with a list of transition probabilities $\{p_{xy_i^x}, 0 \leq i \leq d-1\}$ with the normalization property $\sum_{i=0}^{d-1} p_{xy_i^x} = 1$. Our approach is an application of the powerful general procedure of [GR02]. The idea is to build the superposition up in $\log d$ rounds of doubling the number of terms in the superposition (see Figure 5.1). Each round involves one of the qubits in the register $S$, to which we apply a rotation depending on the state of qubits which we have already touched.

For simplicity, let us first assume all points $x$ have exactly $d$ neighbors and that all transition probabilities $p_{xy_i^x}$ are nonzero, and deal with the general case in Section 5.4.1.

To clean up the notation, denote $q_i = p_{xy_i^x}$. Working up from the last row in Figure 5.1 where $q_i^{(\log d)} = q_i$, we first compute the $d-1$ numbers $q_i^{(k)}$ for $i = 0, \ldots, 2^k - 1$ and $k = 0, \ldots, (\log d) - 1$ from

$$q_i^{(k-1)} = q_{2i}^{(k)} + q_{2i+1}^{(k)}. \tag{5.7}$$

The transition probabilities sum to 1, so we end with $q_0^{(0)} = 1$ at the top.

Our goal is to prepare $|\psi_{\log d}\rangle = \sum_{i=0}^{d-1} \sqrt{q_i}\,|i\rangle$. We start with $\log d$ qubits in the state

$$|\psi_0\rangle = |0\rangle_1 |0\rangle_2 \cdots |0\rangle_{\log d}\,. \tag{5.8}$$

In the first round we prepare

$$|\psi_1\rangle = \left(\sqrt{q_0^{(1)}}\,|0\rangle_1 + \sqrt{q_1^{(1)}}\,|1\rangle_1\right) |0\rangle_2 \cdots |0\rangle_{\log d} \tag{5.9}$$

by applying a rotation to the first qubit. A rotation

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{5.10}$$

by $\theta_0^{(1)} = \cos^{-1}\sqrt{q_0^{(1)}}$ does this job. In the second round, we apply a rotation to the second qubit. However, the amount of rotation now has to depend on the state of the first qubit. When the first qubit is $|0\rangle$, we apply a rotation by

$$\theta_0^{(2)} = \cos^{-1}\sqrt{\frac{q_0^{(2)}}{q_0^{(1)}}}, \tag{5.11}$$

Analogously, when the first qubit is $|1\rangle$, we choose

$$\theta_1^{(2)} = \cos^{-1}\sqrt{\frac{q_2^{(2)}}{q_1^{(1)}}}. \tag{5.12}$$

Observe that the second round turns (5.9) into

$$|\psi_2\rangle = \left(\sqrt{q_0^{(2)}}\,|00\rangle_{1,2} + \sqrt{q_1^{(2)}}\,|01\rangle_{1,2} + \sqrt{q_2^{(2)}}\,|10\rangle_{1,2} + \sqrt{q_3^{(2)}}\,|11\rangle_{1,2}\right)|0\rangle_3 \cdots |0\rangle_{\log d}\,. \quad (5.13)$$

Let us generalize this procedure. Before the $j$-th round, the qubits $j$ and higher are still in the state $|0\rangle$, while the first $j-1$ qubits tell us where in the tree (see Figure 5.1) we are. In round $j$, we thus need to rotate the $j$-th qubit by

$$\theta_i^{(j)} = \cos^{-1}\sqrt{\frac{q_{2i}^{(j)}}{q_i^{(j-1)}}}, \quad (5.14)$$

depending on the state $|i\rangle$ which is encoded in binary in the first $j-1$ qubits of the 'superposition' register $S$.

Applying $\log d$ rounds of this procedure results in preparing the desired superposition (5.6), with the states $|i\rangle$ encoded in binary in the $\log d$ qubits.

### 5.4.1 A Nonuniform Case

In Section 5.4, we assumed each $x$ had exactly $d$ neighbors it could transition to. To deal with having fewer neighbors (and zero transition probabilities), we only need to add an extra 'flag' register $F_i$ for each of the $d$ neighbors $y_i^x$ in the neighbor list. This 'flag' will be 0 if the transition probability $p_{xy_i^x}$ is zero. Conditioning the operations in steps 2-4 of our algorithm (see Section 5.3) on these 'flag' registers will deal with the nonuniform case as well.

### 5.4.2 Precision Requirements

We assumed that each of the probabilities $p_{xy_i^x}$ was given with $t$-bit precision. Our goal was to produce a quantum sample (5.6) whose amplitudes would be precise to $t$ bits as well. Let us investigate how much precision we need in our circuit to achieve this.

For any $x$, the imperfections in $q_i^{\log d} = p_{xy_i^x}$ (see Section 5.4) come from the $\log d$ rotations by imperfectly calculated angles $\theta$. The argument of the inverse cosine in (5.14)

$$a_i^{(j)} = \sqrt{\frac{q_{2i}^{(j)}}{q_i^{(j-1)}}} \tag{5.15}$$

obeys $0 \le a_i^{(j)} \le 1$. The errors in the rotations are the largest for $a_i^{(j)}$ close to 0 or 1 (i.e. when the $\theta$'s are close to $\frac{\pi}{2}$ or 0). To get a better handle on these errors, we introduce extra flag qubits signaling $a_i^{(j)} = 0$ or $a_i^{(j)} = 1$ (see Appendix B.1 for details). In these two special cases, the rotation by $\theta$ becomes an identity or a simple bit flip. On the other hand, because the $q$'s are given with $t$ bits, for $a$'s bounded away from 0 and 1, we have

$$\sqrt{\frac{2^{-t}}{1}} \le a \le \sqrt{\frac{1-2^{-t}}{1}}. \tag{5.16}$$

We choose to use an $n$-bit precision circuit for computing the $a$'s, guaranteeing that $|\tilde{a} - a| \le 2^{-n}$. Using the Taylor expansion, we bound the errors on the angles $\theta$:

$$|\tilde{\theta} - \theta| = |\cos^{-1}\tilde{a} - \cos^{-1} a| = \left|(\tilde{a} - a)\frac{d\cos^{-1} a}{da} + \ldots\right| \le c_1 \frac{2^{-n}}{\sqrt{1-a^2}} \le c_1 2^{-n+\frac{t}{2}}, \tag{5.17}$$

because $a$ is bounded away from 1 as (5.16).

Each amplitude in (5.6) comes from multiplying out $\log d$ terms of the form $\cos\theta_i^j$ or $\sin\theta_i^j$. For our range of $\theta$'s, the error in each sine or cosine is upper bounded by

$$|\sin\tilde{\theta} - \sin\theta| \leq |\tilde{\theta} - \theta|, \qquad |\cos\tilde{\theta} - \cos\theta| \leq |\tilde{\theta} - \theta|. \tag{5.18}$$

Therefore, the final error in each final amplitude is upper bounded by

$$\Delta_i = \left|\sqrt{\tilde{q}_i} - \sqrt{q_i}\right| \leq c_1(\log d)2^{-n+\frac{t}{2}}. \tag{5.19}$$

Note that the factor $\log d$ is small. Therefore, to ensure $t$-bit precision for the final amplitudes, it is enough to work with $n = \frac{3}{2}t + \Omega(1)$ bits of precision during the computation of the $\theta$'s. We conclude that our circuit can be implemented efficiently and keep the required precision.

## 5.5 Discussion

The problem of constructing *explicit* efficient quantum circuits for implementing *arbitrary* sparse quantum walks has not been considered in detail in the literature so far. We were interested in an efficient implementation of a step of a quantum walk and finding one with a favorable scaling of the number of required operations with $d$ (the sparsity parameter) and the accuracy parameter $\frac{1}{\epsilon}$. Its intended use are algorithms based on quantum walks with polynomial speedups over their classical Markov Chain counterparts.

We showed how to efficiently implement a *general*[1] quantum walk $W(P)$ derived from an arbitrary sparse classical random walk $P = (p_{xy})_{x,y\in\mathcal{E}}$. We constructed a quantum circuit $\tilde{U}$ that approximately implements the quantum update rule (5.3) with circuit complexity scaling only *linearly* (with additional logarithmic factors) in $d$, the degree of sparseness of $P$, *linearly* in $m = \log|\mathcal{E}|$ and *polynomially* in $\log\frac{1}{\epsilon}$, where $\epsilon$ denotes the desired approximation accuracy (5.4).

It has been known that quantum walks could be implemented using techniques for simulating Hamiltonian time evolutions. However, the complexity would grow *polynomially* in $\frac{1}{\epsilon}$ if we were to rely on simulating Hamiltonian dynamics (see Section 5.2). This would be fatal for quantum algorithms such as the one for estimating partition functions in [WCNA09] or future algorithms for approximating the permanent, losing the polynomial quantum speed-ups over their classical counterparts. An alternative for implementing quantum walks whose running complexity scales logarithmically in $\frac{1}{\epsilon}$ exists. It relies on the implementation of combinatorially block-diagonal unitaries. However, its running time grows *quadratically* in $d$ (see Section 5.2). When the sparsity of the walk $d$ grows with the system size $n$, this brings an extra factor of $n$ to the complexity of the algorithms, destroying or decreasing its polynomial speedup. This is true e.g.for the example given in Appendix B.6. Therefore, our approach to the quantum update is again more suitable for this task.

[1] Of course, much more efficient approaches exist for specific walks (e.g. those on regular, constant-degree graphs).

# CHAPTER 6
# ESTIMATING PARTITION FUNCTION

We present a quantum algorithm based on classical fully polynomial randomized approximation schemes (FPRAS) for estimating partition functions that combine simulated annealing with the Monte-Carlo Markov Chain method and use non-adaptive cooling schedules. We achieve a twofold polynomial improvement in time complexity: a quadratic reduction with respect to the spectral gap of the underlying Markov chains and a quadratic reduction with respect to the parameter characterizing the desired accuracy of the estimate output by the FPRAS. Both reductions are intimately related and cannot be achieved separately.

First, we use Grover's fixed point search, quantum walks and phase estimation to efficiently prepare approximate coherent encodings of stationary distributions of the Markov chains. The speed-up we obtain in this way is due to the quadratic relation between the spectral and phase gaps of classical and quantum walks. The second speed-up with respect to accuracy comes from generalized quantum counting, used instead of classical sampling to estimate expected values of quantum observables.

Quantization of classical Markov chains has been crucial in the design of efficient quantum algorithms for a wide range of search problems that outperform their classical counterparts. We refer the reader to the survey article [SAN08] for a detailed account of the rapidly growing collection of quantum-walk-based search algorithms. In this context, we also point to the work [SBBK08], where the authors apply quantized Markov chains to speed up search

algorithms based on simulated annealing for finding low-energy states of (classical) Hamiltonians.

In this chapter, we extend the scope of use of quantized Markov chains beyond search problems. We show how to employ them to speed up fully polynomial-time randomized approximation schemes for partition functions, based on simulated annealing and the Monte Carlo Markov Chain (MCMC) method. To achieve this improvement, we rely on Szegedy's general method to quantize classical Markov chains [SZE04, MNRS07]. This method gives us a unitary *quantum walk* operator $W(P)$ corresponding to one update step of the classical Markov chain $P$. The complexity of the classical algorithms we are speeding up is measured in the number of Markov chain invocations. Similarly, we express the complexity of our quantum algorithm as the number of times we have to apply a quantum walk operator. As shown in [CNW10], in the circuit model of quantum computation, this operator can be implemented precisely and efficiently.

Sampling from stationary distributions of Markov chains combined with simulated annealing is at the heart of many clever classical approximation algorithms. Notable examples include the algorithm for approximating the volume of convex bodies [LV06], the permanent of a non-negative matrix [JSV04], and the partition function of statistical physics models such as the Ising model [JS93] and the Potts model [BSVV08]. Each of these algorithms is a *fully polynomial randomized approximation scheme* (FPRAS), outputting a random number $\hat{Z}$ within a factor of $(1 \pm \epsilon)$ of the real value $Z$, with probability greater than $\frac{3}{4}$, i.e.

$$\Pr\left[(1-\epsilon)Z \leq \hat{Z} \leq (1+\epsilon)Z\right] \geq \frac{3}{4}, \tag{6.1}$$

in a number of steps polynomial in $1/\epsilon$ and the problem size.

We show how to use a quantum computer to speed up a class of FPRAS for estimating partition functions that rely on simulated annealing and the Monte Carlo Markov Chain method (e.g. [JS93, BSVV08]). Let us start with an outline of these classical algorithms. Consider a physical system with state space $\Omega$ and an energy function $E : \Omega \to \mathbb{R}$, assigning each state $\sigma \in \Omega$ an energy $E(\sigma)$. The task is to estimate the Gibbs partition function

$$Z(T) = \sum_{\sigma \in \Omega} e^{-\frac{E(\sigma)}{kT}} \tag{6.2}$$

at a desired (usually very low) temperature $T_F$. We would like to know the value of $Z$ at zero temperature because it is equal to the number of the system configurations with zero energy[1], and this could be a hard counting problem.

The partition function $Z(T)$ encodes the thermodynamical properties of the system in equilibrium at temperature $T$, where the probability of finding the system in state $\sigma$ is given by the Boltzmann distribution

$$\pi_i(\sigma) = \frac{1}{Z(T)} e^{-\frac{E(\sigma)}{kT}}. \tag{6.3}$$

It is hard to estimate $Z(T)$ directly. The schemes we want to speed up use the following trick. Consider a sequence of decreasing temperatures $T_0 \geq T_1 \geq \cdots \geq T_\ell$, where $T_0$ is a very high starting temperature and $T_\ell = T_F$ is the desired final temperature. Then, $Z(T_F)$

[1]This relationship is used e.g. in the algorithm [JSV04] for approximating the permanent of a non-negative matrix – one can find the value of the permanent by counting the number of perfect matchings of a particular bipartite graph, which in turn is equal to the zero-temperature partition function of a certain spin system.

can be expressed as a telescoping product

$$\begin{aligned} Z(T_F) &= Z_0 \frac{Z_1}{Z_0} \cdots \frac{Z_{\ell-1}}{Z_{\ell-2}} \frac{Z_\ell}{Z_{\ell-1}} \\ &= Z_0 \underbrace{(\alpha_0 \alpha_1 \cdots \alpha_{\ell-2} \alpha_{\ell-1})}_{\alpha}, \end{aligned} \tag{6.4}$$

where $Z_i = Z(T_i)$ stands for the Gibbs partition function at temperature $T_i$ and $\alpha_i = Z_{i+1}/Z_i$. It is easy to calculate the partition function $Z_0 = Z(T_0)$ at high temperature. Next, for each $i$, we can estimate the ratio $\alpha_i$ by sampling from a distribution that is sufficiently close to the Boltzmann distribution $\pi_i$ (6.3) at temperature $T_i$ (see Section 6.1 for more detail). This is possible by using a rapidly-mixing Markov chain $P_i$ whose stationary distribution is equal to the Boltzmann distribution $\pi_i$.

To be efficient, these classical schemes require that

1. we use a cooling schedule such that the resulting ratios $\alpha_i = Z(T_{i+1})/Z(T_i)$ are lower bounded by a constant $c^{-1}$ (to simplify the presentation, we use $c = 2$ from now on),

2. the spectral gaps of the Markov chains $P_i$ are bounded from below by $\delta$.

The time complexity of such FPRAS, i.e., the number of times we have to invoke an update step for a Markov chain from $\{P_1, \ldots, P_{\ell-1}\}$, is

$$\tilde{O}\left(\frac{\ell^2}{\delta \cdot \epsilon^2}\right), \tag{6.5}$$

where $\tilde{O}$ means up to logarithmic factors.

Our main result is a general method for 'quantizing' such algorithms. Note that the method we present in this chapter does not yet allow us to speed up the more complicated

classical algorithm for the permanent (which requires to sample from the stationary distributions of the previously used Markov chains to decide which Markov chain to use next). Together with the algorithms using adaptive cooling schedules, it is a direction for further research.

**Theorem 13.** *Consider a classical FPRAS for approximating the Gibbs partition function of a physical system at temperature $T_F$, satisfying the above conditions. Then, there exists a fully polynomial quantum approximation scheme that uses*

$$\tilde{O}\left(\frac{\ell^2}{\sqrt{\delta}\cdot\epsilon}\right) \tag{6.6}$$

*applications of a controlled version of a quantum walk operator from $\{W(P_1),\ldots,W(P_{\ell-1})\}$.*

The reduction in complexity for our quantum algorithm (in comparison to the classical FPRAS) is twofold. First, we reduce the factor $1/\delta$ to $1/\sqrt{\delta}$ by using quantum walks instead of classical Markov chains, and utilizing the quadratic relation between spectral and phase gaps. As observed in [MNRS07], this relation is at the heart of many quantum search algorithms based on quantum walks (see e.g. [SAN08] for an overview of such quantum algorithms). Second, we speed up the way to determine the ratios $\alpha_i$ by using the quantum phase estimation algorithm in a novel way. This results in the reduction of the factor $1/\epsilon^2$ to $1/\epsilon$.

The quantum algorithm we present builds upon the previous work [WA08], where *Wocjan* and *Abeyesinghe* have shown how to use quantum walks to approximately prepare coherent

encodings

$$|\pi_i\rangle = \sum_{\sigma \in \Omega} \sqrt{\pi_i(\sigma)}\, |\sigma\rangle \tag{6.7}$$

of stationary distributions $\pi_i$ of Markov chains $P_i$, provided that the Markov chains are slowly-varying. Recall that a sequence of Markov chains is called slowly-varying if the stationary distributions of two adjacent chains are sufficiently close to each other. As we will see later, this condition is automatically satisfied for Markov chains that are used in FPRAS for approximating partition functions.

Note that our objective of approximately preparing coherent encodings of stationary distributions is different from the objective in [RIC07], where the author seeks to speed up the process of approximately preparing density operators encoding stationary distributions. For our purposes, we have to work with coherent encodings because otherwise we could not achieve the second reduction from $1/\epsilon^2$ to $1/\epsilon$.

This chapter is organized as follows. In Section 6.1 we review the classical FPRAS in more detail. We present our quantum algorithm in two steps. First, in Section 6.2.2 we explain how our quantum algorithm works, assuming that we can perfectly and efficiently prepare coherent encodings of the distributions (6.3). Then, in Section 6.2.3 we describe the full quantum algorithm, dropping the assumption of Section 6.2.2 and using approximate procedures for quantum sample preparation and readout, which are based on the quantum walks. We perform a detailed analysis of accumulation of error due to the approximation procedures and show that the success probability remains high, establishing Theorem 13.

Finally, in Section 6.3 we conclude with a discussion of open questions, the connection of our algorithm to simulated annealing, and the directions for future research.

## 6.1 Structure of the Classical Approach

Here we describe the classical approximation schemes in more detail, following closely the presentation in [BSVV08, Section 2.1]. Choosing a sequence of temperatures $T_0 \geq T_1 \geq \cdots \geq T_\ell$ starting with $T_0 = \infty$, and ending with the desired final (low) temperature $T_\ell = T_F$, we can express the Gibbs partition function (6.2) as a telescoping product (6.4). At $T_0 = \infty$, the partition function $Z_0$ is equal to

$$Z_0 = |\Omega|, \tag{6.8}$$

the size of the state space. On the other hand, for each $i = 0, \ldots, \ell - 1$, we can estimate the ratio

$$\alpha_i = \frac{Z_{i+1}}{Z_i} \tag{6.9}$$

in (6.4) as follows. Let $X_i \mathrm{sim} \pi_i$ denote a random state chosen according to the Boltzmann distribution $\pi_i$, i.e.,

$$\Pr(X_i = \sigma) = \pi_i(\sigma)\,. \tag{6.10}$$

Define a new random variable $Y_i$ by

$$Y_i = e^{-(\beta_{i+1} - \beta i)\, E(X_i)}, \tag{6.11}$$

where $\beta_i = (kT_i)^{-1}$ is the inverse temperature ($k$ is the Boltzmann constant). This $Y_i$ is an unbiased estimator for $\alpha_i$ since

$$\begin{aligned} \mathbf{E}\left(Y_i\right) &= \sum_{\sigma\in\Omega} \pi_i(\sigma)\, e^{-(\beta_{i+1}-\beta i)\, E(\sigma)} && (6.12)\\ &= \sum_{\sigma\in\Omega} \frac{e^{-\beta_i E(\sigma)}}{Z_i}\, e^{-(\beta_{i+1}-\beta i)\, E(\sigma)} && (6.13)\\ &= \sum_{\sigma\in\Omega} \frac{e^{-\beta_{i+1}\, E(\sigma)}}{Z_i} = \frac{Z_{i+1}}{Z_i} = \alpha_i. && (6.14) \end{aligned}$$

Assume now that we have an algorithm for generating states $X_i$ according to $\pi_i$. We draw

$$m := 64\ell/\epsilon^2 \tag{6.15}$$

samples of $X_i$ and take the mean $\overline{Y}_i$ of their corresponding estimators $Y_i$. Then, the mean $\overline{Y}_i$ satisfies

$$\frac{\mathbf{Var}\left(\overline{Y}_i\right)}{\left(\mathbf{E}\left(\overline{Y}_i\right)\right)^2} = \frac{\epsilon^2}{64\ell}\,\frac{\mathbf{Var}\left(Y_i\right)}{\left(\mathbf{E}\left(Y_i\right)\right)^2} \leq \frac{\epsilon^2}{16\ell}\,. \tag{6.16}$$

(We have used the assumption $\frac{1}{2} \leq \alpha_i \leq 1$.) We can now compose such estimates of $\alpha_i$. Define a new random variable $\overline{Y}$ by

$$\overline{Y} = \overline{Y}_{\ell-1}\overline{Y}_{\ell-2}\cdots\overline{Y}_0 \tag{6.17}$$

Since all $\overline{Y}_i$ are independent, we have

$$\begin{aligned} \mathbf{E}\left(\overline{Y}\right) &= \mathbf{E}\left(Y_{\ell-1}\right)\mathbf{E}\left(Y_{\ell-2}\right)\cdots\mathbf{E}\left(Y_0\right)\\ &= \alpha_{\ell-1}\alpha_{\ell-2}\cdots\alpha_0 = \alpha, && (6.18) \end{aligned}$$

Moreover, $\overline{Y}$ has the property

$$\begin{aligned}
&\frac{\mathbf{Var}\left(\overline{Y}\right)}{\left(\mathbf{E}\left(\overline{Y}\right)\right)^2} \\
&= \frac{\mathbf{E}\left(\overline{Y}_{\ell-1}^2\right)\cdots\mathbf{E}\left(\overline{Y}_0^2\right) - \mathbf{E}\left(\overline{Y}_{\ell-1}\right)^2\cdots\mathbf{E}\left(\overline{Y}_0\right)^2}{\mathbf{E}\left(\overline{Y}_{\ell-1}\right)^2\cdots\mathbf{E}\left(\overline{Y}_0\right)^2} \\
&= \left(1+\frac{\mathbf{Var}\left(\overline{Y}_{\ell-1}\right)}{\left(\mathbf{E}\left(\overline{Y}_{\ell-1}\right)\right)^2}\right)\cdots\left(1+\frac{\mathbf{Var}\left(\overline{Y}_0\right)}{\left(\mathbf{E}\left(\overline{Y}_0\right)\right)^2}\right) - 1 \\
&\leq \left(e^{\epsilon^2/16\ell}\right)^\ell - 1 \\
&\leq \epsilon^2/8\,, \qquad (6.19)
\end{aligned}$$

where we used $1+x \leq e^x$ (true for all $x$) and $e^x - 1 \leq 2x$ (true for all $x \in [0,1]$) in the last two steps, respectively. Chebyshev's inequality now implies that the value of $\overline{Y}$ is in the interval $[(1-\epsilon)\alpha, (1+\epsilon)\alpha]$ with probability at least $\frac{7}{8}$.

Of course, we are not able to obtain perfect samples $X_i$ from $\pi_i$. Assume now that we have $X_i'$ that are from a distribution with a variation distance from $\pi_i$ smaller than

$$d := \epsilon^2/(512\ell^2). \qquad (6.20)$$

Let $\overline{Y}'$ be defined as $\overline{Y}$ as above, but instead of $X_i$ we use $X_i'$. Then, with probability at least $\frac{7}{8}$, we have $\overline{Y} = \overline{Y}'$. To derive this, observe that the algorithm can be thought to first take a sample from a product probability distribution $\pi$ on the $(m\ell)$-fold direct product of $\Omega$. We denote the probability distribution in the case of imperfect samples by $\pi'$. The total variation distance between $\pi$ and $\pi'$ is then bounded from above by

$$d \cdot m \cdot \ell = \frac{\epsilon^2}{512\ell^2} \cdot \frac{64\ell}{\epsilon^2} \cdot \ell = \frac{1}{8}\,. \qquad (6.21)$$

Therefore, $\overline{Y}'$ is in the interval $[(1-\epsilon)\mathbf{E}(Y), (1+\epsilon)\mathbf{E}(Y)]$ with probability at least $\frac{3}{4}$.

We obtain the samples $X_i'$ by applying Markov chains $P_i$ whose limiting distributions are equal to $\pi_i$. Constructing such rapidly-mixing Markov chains is a hard task, but it has been done for the Ising model [JS93] and the Potts model [BSVV08].

## 6.2 Structure of Our Quantum Approach

### 6.2.1 Overview

The classical FPRAS we described in Section 6.1 consists of

1. preparing many samples from a distribution close to $\pi_i$ by letting a suitable Markov chain mix,
2. using these samples to approximate the ratios $\alpha_i$ in (6.4), and
3. composing these estimates of $\alpha_i$ into an estimate of the partition function.

We build our quantum algorithm on this scheme, with two novel quantum ingredients. First, instead of letting a Markov chain $P_i$ mix towards its stationary distribution $\pi_i$, we choose to approximately prepare the state $|\pi_i\rangle = \sum_\sigma \sqrt{\pi_i(\sigma)}\,|\sigma\rangle$, a coherent encoding of the Boltzmann distribution. We use a preparation method [WA08] based on Grover's $\frac{\pi}{3}$-fixed-point search [GRO05], efficiently driving the state $|\pi_0\rangle$ towards the desired state $|\pi_i\rangle$ through a sequence of intermediate states.

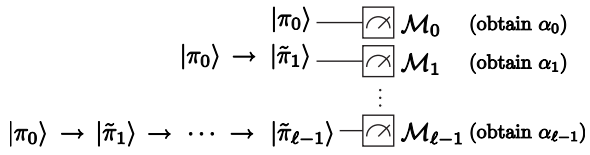


Figure 6.1: Structure of the Quantum Algorithm

Second, instead of using classical samples from the distribution $\pi_i$, we approximate $\alpha_i$ by phase-estimation of a certain unitary on the state $|\pi_i\rangle$. This is a new concept, going beyond the previous work [WA08]. This phase-estimation subroutine can be efficiently (albeit only approximately) applied by utilizing quantum walks.

The structure of our algorithm is depicted in Fig. 6.1. It consists of successive approximate preparations of $|\pi_i\rangle$ followed by a quantum circuit outputting a good approximation to $\alpha_i$ (with high probability). Our main result is the construction of a fast quantum version of a class of classical algorithms, summed in Theorem 13.

We arrive at our quantum algorithm in two steps. First, in Section 6.2.2, we explain how to quantize the the classical algorithm in the perfect case, assuming that we can take perfect samples $X_i$ from $\pi_i$. Then, in Section 6.2.3 we release this assumption and describe the full quantum algorithm.

### 6.2.2 Perfect Case

To estimate the ratios $\alpha_i$ in (6.4), the classical algorithm generates random states $X_i$ from $\pi_i$ and computes the mean $\overline{Y}_i$ of the random variables $Y_i$. The process of generating a random state $X_i$ from $\pi_i$ is equivalent to preparing the mixed state

$$\rho_i = \sum_{\sigma \in \Omega} \pi_i(\sigma) |\sigma\rangle\langle\sigma| \,. \tag{6.22}$$

Instead of this, we choose to prepare the pure states

$$|\pi_i\rangle = \sum_{\sigma \in \Omega} \sqrt{\pi_i(\sigma)} |\sigma\rangle \,. \tag{6.23}$$

We call these states *quantum samples* since they coherently encode the probability distributions $\pi_i$. In this Section, we assume that we can prepare these exactly and efficiently.

The random variable $Y_i$ can be interpreted as the outcome of the measurement of the observable

$$A_i = \sum_{\sigma \in \Omega} y_i(\sigma) |\sigma\rangle\langle\sigma| \tag{6.24}$$

in the state $\rho_i$, where

$$y_i(\sigma) = e^{-(\beta_{i+1} - \beta_i) E(\sigma)} \,. \tag{6.25}$$

With this interpretation in mind, we see that to estimate $\alpha_i$ classically, we need to estimate the expected value $\mathrm{Tr}(A_i \rho_i)$ by repeating the above measurement several times and outputting the mean of the outcomes.

We now explain how to quantize this process. We add an ancilla qubit to our quantum system in which the quantum samples $|\pi_i\rangle$ live. For each $i = 0, \ldots, \ell - 1$, we define the

unitary

$$V_i = \sum_{\sigma\in\Omega} |\sigma\rangle\langle\sigma| \otimes \begin{pmatrix} \sqrt{y_i(\sigma)} & \sqrt{1-y_i(\sigma)} \\ -\sqrt{1-y_i(\sigma)} & \sqrt{y_i(\sigma)} \end{pmatrix}. \tag{6.26}$$

This $V_i$ can be efficiently implemented, it is a rotation on the extra qubit controlled by the state of the first tensor component. Let us label

$$|\psi_i\rangle = V_i\big(|\pi_i\rangle \otimes |0\rangle\big). \tag{6.27}$$

Consider now the expected value of the projector

$$P = \mathbb{I} \otimes |0\rangle\langle 0| \tag{6.28}$$

in the state $|\psi_i\rangle$. We find

$$\langle\psi_i|P|\psi_i\rangle = \langle\pi_i|A_i|\pi_i\rangle = \alpha_i\,. \tag{6.29}$$

We now show how to speed up the process of estimating $\alpha_i$ with a method that generalizes quantum counting [BHT98]. As noted in the beginning of this Section, we assume efficient preparation of $|\pi_i\rangle$, which in turn implies that we can efficiently implement the reflections

$$R_i = 2|\pi_i\rangle\langle\pi_i| - \mathbb{I}\,. \tag{6.30}$$

The result of this Section, the existence of a quantum FPRAS for estimating the partition function assuming efficient and perfect preparation of $|\pi_i\rangle$, is summed in Theorem 14:

**Theorem 14.** *There is a fully polynomial quantum approximation scheme $\mathcal{A}$ for the partition function $Z$. Its output $Q$ satisfies*

$$\Pr\left[(1-\epsilon)Z \le Q \le (1+\epsilon)Z\right] \ge \frac{3}{4}\,. \tag{6.31}$$

*For each $i = 0, \ldots, \ell - 1$, the scheme $\mathcal{A}$ uses*

$$O\left(\log \ell\right) \tag{6.32}$$

*perfectly prepared quantum samples $|\pi_i\rangle$, and applies the controlled-$R_i$ operator*

$$O\left(\frac{\ell}{\epsilon} \log \ell\right) \tag{6.33}$$

*times, where $R_i$ is as in (6.30).*

To prove Theorem 14, we need the following three technical results.

**Lemma 2** (Quantum ratio estimation)**.** *Let $\epsilon_{pe} \in (0, 1)$. For each $i = 0, \ldots, \ell - 1$ there exists a quantum approximation scheme $\mathcal{A}'_i$ for $\alpha_i$. Its output $Q'_i$ satisfies*

$$\Pr\left[(1 - \epsilon_{pe})\alpha_i \leq Q'_i \leq (1 + \epsilon_{pe})\alpha_i\right] \geq \frac{7}{8}. \tag{6.34}$$

*The scheme $\mathcal{A}'_i$ requires one copy of the quantum sample $|\pi_i\rangle$ and invokes the controlled-$R_i$ operator $O\left(\epsilon_{pe}^{-1}\right)$ times, where $R_i$ is as in (6.30).*

*Proof.* Let

$$G = \left(2\left|\psi_i\right\rangle\left\langle\psi_i\right| - \mathbb{I}\right)\left(2P - \mathbb{I}\right). \tag{6.35}$$

Define the basis states

$$\left|\gamma_1\right\rangle = \frac{\left(\mathbb{I} - P\right)\left|\psi_i\right\rangle}{\sqrt{1 - \alpha_i}}, \quad \text{and} \quad \left|\gamma_2\right\rangle = \frac{P\left|\psi_i\right\rangle}{\sqrt{\alpha_i}}. \tag{6.36}$$

Restricted to the plane spanned by $|\gamma_1\rangle$ and $|\gamma_2\rangle$, $G$ acts as a rotation

$$G\big|_{\{|\gamma_1\rangle, |\gamma_2\rangle\}} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}, \tag{6.37}$$

where $\theta \in [0, \frac{\pi}{2}]$ satisfies

$$\cos\theta = 2\alpha_i - 1. \tag{6.38}$$

The eigenvectors and eigenvalues of $G$ are

$$|G_\pm\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ \pm i \end{bmatrix}, \quad \lambda_\pm = e^{\pm i\theta}. \tag{6.39}$$

We do not have direct access to one of these eigenvectors, as the state $|\psi_i\rangle$ is in a superposition of $|G_+\rangle$ and $|G_-\rangle$. Thus, when we apply the phase estimation circuit for the unitary $G$ to the state $|\psi_i\rangle$, we will sometimes obtain an estimate of $\theta$, and sometimes an estimate of $2\pi - \theta$. However, this is not a problem since both $\theta$ and $2\pi - \theta$ plugged into (6.38) yield the same result for $\alpha_i$.

We require that the estimate $\theta'$ satisfies

$$|\theta' - \theta| \leq 2\epsilon_{pe}\,\alpha_i \leq \epsilon_{pe} \tag{6.40}$$

with probability at least $\frac{7}{8}$. Using the phase estimation circuit in [NC00], this means that $\frac{\theta'}{2\pi}$ has to be an $n_a = \log\frac{2\pi}{\epsilon_{pe}}$ bit approximation of the phase and the failure probability $p_f$ has to be less than $\frac{1}{8}$. To achieve this, it suffices to use a phase estimation circuit (see Fig. 6.2) with

$$t = \log\frac{2\pi}{\epsilon_{pe}} + \log\left(2 + \frac{1}{2\,p_f}\right) = O\left(\log\epsilon_{pe}^{-1}\right)$$

ancilla qubits. This circuit invokes the controlled-$G$ operation $O(2^t) = O\left(\epsilon_{pe}^{-1}\right)$ times.

Let $\alpha_i'$ denote the value we compute from the estimate $\theta'$. We have

$$|\alpha_i - \alpha_i'| = \frac{1}{2}\,|\cos\theta - \cos\theta'| \leq \frac{1}{2}\,|\theta - \theta'| \leq \epsilon_{pe}\,\alpha_i\,, \tag{6.41}$$

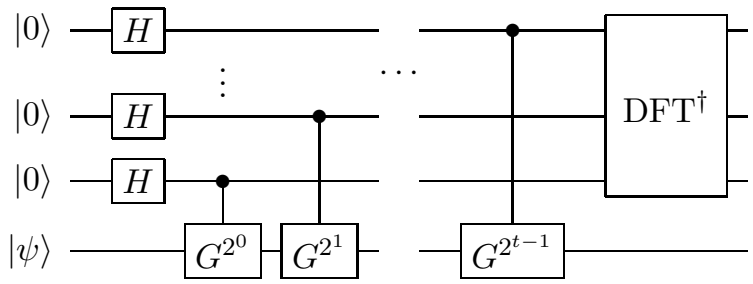


Figure 6.2: A Basic Phase Estimation Circuit with $t$ Ancilla Qubits

showing that the estimate $\alpha_i'$ is within $\pm\epsilon_{pe}\alpha_i$ of the exact value $\alpha_i$ with probability at least $\frac{7}{8}$. This completes the proof that the random variable $Q_i'$ corresponding to the output satisfies the desired properties on estimation accuracy and success probability. □

We can boost the success probability of the above quantum approximation scheme for the ratio $\alpha_i$ by applying the *powering lemma* from [JVV86], which we state here for completeness:

**Lemma 3** (Powering lemma for approximation schemes)**.** *Let $\mathcal{B}'$ be a (classical or quantum) approximation scheme whose estimate $W'$ is within $\pm\epsilon_{pe}q$ to some value $q$ with probability $\frac{1}{2}+\Omega(1)$. Then, there is an approximation scheme $\mathcal{B}$ whose estimate $W$ satisfies*

$$\Pr\left[(1-\epsilon_{pe})q \leq W \leq (1+\epsilon_{pe})q\right] \geq 1-\delta_{boost}\,. \tag{6.42}$$

*It invokes the scheme $\mathcal{B}'$ as a subroutine $O\left(\log \delta_{boost}^{-1}\right)$ times.*

With the help of Lemma 3, we now have the constituents required to compose the individual estimates of $\alpha_i$ into an approximation for the partition function (6.4).

**Lemma 4.** *Let $\epsilon > 0$. Assume we have approximation schemes $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{\ell-1}$ such that their estimates $Q_0, Q_1, \ldots, Q_{\ell-1}$ satisfy*

$$\Pr\Big[\Big(1 - \frac{\epsilon}{2\ell}\Big)\alpha_i \le Q_i \le \Big(1 + \frac{\epsilon}{2\ell}\Big)\alpha_i\Big] \ge 1 - \frac{1}{4\ell}\,. \tag{6.43}$$

*Then, there is a simple approximation scheme $\mathcal{A}$ for the product $\alpha = \alpha_0\alpha_1\cdots\alpha_{\ell-1}$. The result $Q = Q_0Q_1\cdots Q_{\ell-1}$ satisfies*

$$\Pr\big[(1-\epsilon)\alpha \le Q \le (1+\epsilon)\alpha\big] \ge \frac{3}{4}\,. \tag{6.44}$$

*Proof.* For each $i = 0, \ldots, \ell - 1$, the failure probability for estimating $\alpha_i$ is smaller than $1/(4\ell)$. The union bound implies that the overall failure probability is smaller than $1/4$, proving the lower bound $\frac{3}{4}$ on the success probability in (6.44).

To obtain the upper bound on the deviation, we now assume that each $Q_i$ takes the upper bound value. We have

$$\begin{aligned}\frac{Q-\alpha}{\alpha} &\le \prod_{i=0}^{\ell-1}\Big(1 + \frac{\epsilon}{2\ell}\Big) - 1 = \Big(1 + \frac{\epsilon}{2\ell}\Big)^{\ell} - 1\\ &\le e^{\epsilon/2} - 1 \le \epsilon\,,\end{aligned}$$

where we have used $1 + x \le e^x \le 1 + 2x$, which is true for all $x \in [0, 1]$. Thus, in the case of success, we have $Q \le (1 + \epsilon)\alpha$.

To obtain the lower bound on the deviation, we assume that each $Q_i$ takes its lower bound value. We have

$$\frac{\alpha - Q}{\alpha} \le 1 - \prod_{i=0}^{\ell-1}\Big(1 - \frac{\epsilon}{2\ell}\Big) \le \sum_{i=0}^{\ell-1}\frac{\epsilon}{2\ell} \le \epsilon\,, \tag{6.45}$$

where we have used $|\prod_i x_i - \prod_i y_i| \leq \sum_i |x_i - y_i|$, true for arbitrary $x_i, y_i \in [0,1]$. Thus, in the case of success, we have $(1-\epsilon)\alpha \leq Q$. □

We are now ready to prove Theorem 14:

*Proof of Theorem 14.* For each $i = 0, \ldots, \ell - 1$, we can apply Lemma 2 with the state $|\psi_i\rangle$ (6.27) and the projector $P$ (6.28). This gives us a quantum approximation scheme for $\alpha_i$. Note that to prepare $|\psi_i\rangle$, it suffices to prepare $|\pi_i\rangle$ once. Also, to realize a controlled reflection around $|\psi_i\rangle$, it suffices to invoke the controlled reflection around $|\pi_i\rangle$ once.

We now use the reflection $2|\psi_i\rangle\langle\psi_i| - \mathbb{I}$ and set $\epsilon_{pe} = \epsilon/(2\ell)$ in Lemma 2. With these settings, we can apply Lemma 3 to the resulting approximation scheme for $\alpha_i$ with $\delta_{boost} = 1/(4\ell)$. This gives us approximation schemes $\mathcal{A}_i$ outputting $Q_i$ with high precision and probability of success that can be used in Lemma 4. The composite result $Q = Q_0 \cdots Q_{\ell-1}$ is thus an approximation for $\alpha = \alpha_0 \cdots \alpha_{\ell-1}$ with the property

$$\Pr\left[(1-\epsilon)\alpha \leq Q \leq (1+\epsilon)\alpha\right] \geq \frac{3}{4}. \tag{6.46}$$

Finally, we obtain the estimate for $Z$ by multiplying $Q$ with $Z_0$. Let us summarize the costs from Lemmas 2-4. For each $i = 0, \ldots, \ell - 1$, this scheme uses $\log \delta_{boost}^{-1} = O(\log \ell)$ copies of the state $|\pi_i\rangle$, and invokes $\left(\log \delta_{boost}^{-1}\right) \epsilon_{pe}^{-1} = O\left(\frac{\ell}{\epsilon} \log \ell\right)$ reflections around $|\pi_i\rangle$. □

### 6.2.3 Quantum FPRAS

In the previous Section we have assumed that we can prepare the quantum samples $|\pi_i\rangle$ and implement the controlled reflections $R_i = 2|\pi_i\rangle\langle\pi_i| - \mathbb{I}$ about these states perfectly and efficiently. We now release these assumptions and show how to approximately accomplish both tasks with the help of quantum walks operators. We then show that the errors arising from these approximate procedures do not significantly decrease the success probability of the algorithm. This will wrap up the proof of our main result, Theorem 13.

In [WA08], *Wocjan* and *Abeyesinghe* show how to approximately prepare quantum samples $|\pi_i\rangle$ of stationary distributions of slowly-varying Markov chains. Using the fact that the consecutive states $|\pi_i\rangle$ and $|\pi_{i+1}\rangle$ are close, we utilize Grover's $\frac{\pi}{3}$ fixed-point search [GRO05] to drive the starting state $|\pi_0\rangle$ towards the desired state $|\pi_i\rangle$ through multiple intermediate steps. Moreover, to be able to perform this kind of Grover search, we have to be able to apply selective phase shifts of the form $S_i = \omega|\pi_i\rangle\langle\pi_i| + (\mathbb{I} - |\pi_i\rangle\langle\pi_i|)$ for $\omega = e^{i\pi/3}$ and $\omega = e^{-i\pi/3}$. This is another assumption of Section 6.2.2 that we have to drop here. Nevertheless, an efficient way to apply these phase shifts approximately, based on quantum walks and phase estimation, exists [WA08].

Our task is to show that the approximation scheme from Lemma 2 works even with approximate input states and using only approximate reflections about the states $|\pi_i\rangle$. Let us start with addressing the approximate state preparation. To be able to use the results of [WA08], we first have to establish an important condition. For their method to be efficient,

the overlap of two consecutive quantum samples $|\pi_i\rangle$ and $|\pi_{i+1}\rangle$ has to be large. This is satisfied when $\alpha_i = Z_{i+1}/Z_i$ is bounded from below by $\frac{1}{2}$, since

$$\begin{aligned} |\langle \pi_i | \pi_{i+1} \rangle|^2 &= \left| \sum_{\sigma \in \Omega} \frac{\sqrt{e^{-\beta_i E(\sigma)}\, e^{-\beta_{i+1} E(\sigma)}}}{\sqrt{Z_i\, Z_{i+1}}} \right|^2 \\ &\geq \left| \frac{\sum_{\sigma \in \Omega} e^{-\beta_{i+1} E(\sigma)}}{\sqrt{2 Z_{i+1}}\, \sqrt{Z_{i+1}}} \right|^2 = \frac{1}{2}\,. \end{aligned}$$

The following lemma then directly follows from the arguments used in [WA08, Theorem 2].

**Lemma 5.** *For $\epsilon_S > 0$ arbitrary and each $i = 1, \ldots, \ell - 1$, there is a quantum method preparing a state $|\tilde{\pi}_i\rangle$ with*

$$\| |\tilde{\pi}_i\rangle - |\pi_i\rangle |0\rangle^{\otimes a} \| \leq \epsilon_S\,, \tag{6.47}$$

*where $a = O\left(\frac{\ell}{\epsilon_S \sqrt{\delta}}\right)$ is the number of ancilla qubits. The method invokes a controlled version of a walk operator from the set $\{W(P_1), \ldots, W(P_{\ell-1})\}$*

$$O\left(\frac{\ell}{\sqrt{\delta}} \log^2 \frac{\ell}{\epsilon_S}\right)\,. \tag{6.48}$$

*times.*

We choose the preparation method from Lemma 5 with $\epsilon_S = \frac{1}{32}$. The cost for this precision $\epsilon_S$ is

$$O\left(\frac{\ell}{\sqrt{\delta}} \log^2 \ell\right) \tag{6.49}$$

applications of the quantum walk operator. Recall that when we used Lemma 2 in Section 6.2.2 with the state $|\psi_i\rangle$ (coming from the perfect quantum sample $|\pi_i\rangle$) as input, the success probability of the resulting scheme was greater than $\frac{7}{8}$. We now use the method given in

Lemma 2 on the approximate input $|\tilde{\psi}_i\rangle = V_i(|\tilde{\pi}_i\rangle \otimes |0\rangle)$. With our chosen precision for preparing $|\tilde{\pi}_i\rangle$, the success probability of the approximation scheme of Lemma 2 cannot decrease by more than $2 \cdot \frac{1}{32}$.

The second assumption of Lemma 2 we need to drop is the ability to perfectly implement the reflections $R_i = 2|\pi_i\rangle\langle\pi_i| - \mathbb{I}$. We now show how to approximately implement these reflections. The following lemma follows directly from the arguments in [WA08, Lemma 2 and Corollary 2].

**Lemma 6.** *For $\epsilon_R > 0$ arbitrary and each $i = 1, \ldots, \ell - 1$, there is an approximate reflection $\tilde{R}_i$ such that*

$$\tilde{R}_i\Big(|\varphi\rangle \otimes |0\rangle^{\otimes b}\Big) = \Big(R_i|\varphi\rangle\Big) \otimes |0\rangle^{\otimes b} + |\xi\rangle \tag{6.50}$$

*where $|\varphi\rangle$ is an arbitrary state, $b = O\left(\log \epsilon_R^{-1} \log \frac{1}{\sqrt{\delta}}\right)$ is the number of ancilla qubits, and $|\xi\rangle$ is some error vector with $\||\xi\rangle\| \leq \epsilon_R$. It invokes the controlled version of a walk operator from $\{W(P_1), \ldots, W(P_{\ell-1})\}$*

$$O\left(\frac{1}{\sqrt{\delta}} \log \frac{1}{\epsilon_R}\right) \tag{6.51}$$

*times.*

Recall that in Lemma 2, the controlled reflection $R_i$ is invoked $O(1/\epsilon_{pe})$ times. We now run this approximation scheme with $\tilde{R}_i$ instead of $R_i$. The norm of the accumulated error vector is

$$O\left(\frac{1}{\epsilon_{pe}}\right) \cdot \epsilon_R. \tag{6.52}$$

We choose

$$\epsilon_R = \Omega(\epsilon_{pe}) \tag{6.53}$$

to bound the norm of the accumulated error from above by $\frac{1}{32}$. The success probability can then decrease by at most $2 \cdot \frac{1}{32}$.

Combining these arguments establishes a variant of Lemma 2 without the unnecessary assumptions of Section 6.2.2:

**Lemma 7.** *Let $\epsilon_{pe} \in (0,1)$. For each $i = 0, \ldots, \ell - 1$, there exists a quantum approximation scheme $\mathcal{A}_i''$ for $\alpha_i$. Its estimate $Q_i''$ satisfies*

$$\Pr\left[(1-\epsilon_{pe})\alpha_i \leq Q_i'' \leq (1+\epsilon_{pe})\alpha_i\right] \geq \frac{3}{4}. \tag{6.54}$$

*This scheme invokes the controlled version of a walk operator from $\{W(P_1), \ldots, W_{\ell-1}\}$*

$$O\left(\frac{\ell}{\sqrt{\delta}} \log^2 \ell + \frac{1}{\epsilon_{pe}\sqrt{\delta}} \log \epsilon_{pe}^{-1}\right). \tag{6.55}$$

*Proof.* The success probability of the scheme in Lemma 2 was greater than $\frac{7}{8}$. Both the approximate state preparation and using approximate reflections reduce the overall probability of success by at most $\frac{1}{16}$. Thus the probability of success of the method given in Lemma 2 after dropping the unnecessary assumptions is at least $\frac{3}{4}$. □

We can finally complete the proof of Theorem 13 by following the procedure that led to the proof of Theorem 14 in Section 6.2.2.

*Proof of Theorem 14.* For each $i = 0, \ldots, \ell - 1$, we proceed as follows. We use the approximation scheme $\mathcal{A}_i''$ from Lemma 7 with precision $\epsilon_{pe} = \epsilon/(2\ell)$. We then boost the

success probability of each $\mathcal{A}_i''$ to $1 - \frac{1}{4\ell}$ by applying the powering lemma (Lemma 3) with $\delta_{boost} = 1/(4\ell)$. This step increases the cost in (6.55) by the factor $O(\log \ell)$. This resulting scheme $\mathcal{A}_i$ now satisfies the properties required for Lemma 4. We can thus use it to obtain a composite approximation scheme whose output satisfies

$$\Pr\left[(1-\epsilon)Z \leq Q \leq (1+\epsilon)Z\right] \geq \frac{3}{4}. \tag{6.56}$$

The resulting cost of this scheme (the number of times we have to invoke the controlled quantum walk operators) is

$$\begin{aligned} O\left(\frac{\ell^2}{\sqrt{\delta}}\,\log^3 \ell + \frac{\ell^2}{\epsilon\sqrt{\delta}}(\log \ell)\left(\log \ell + \log \epsilon^{-1}\right)\right) \\ = \tilde{O}\left(\frac{\ell^2}{\epsilon\sqrt{\delta}}\right). \end{aligned} \tag{6.57}$$

□

## 6.3 Discussion

We have shown that in the quantum circuit model, we can speed up a class of classical FPRAS for approximating partition functions, as measured in the number of times we have to invoke[2] a step of a quantum walk (instead of classical Markov chains). We obtained two reductions in complexity: $1/\delta \to 1/\sqrt{\delta}$ and $1/\epsilon^2 \to 1/\epsilon$. These two reductions are intimately related; they cannot occur separately. If we used quantum samples merely to obtain classical samples (i.e., if we tried to estimate the ratios without phase estimation), then this would

[2]When the classical Markov chains can be implemented efficiently, each step of the corresponding quantum walks can also be applied efficiently and precisely, as shown e.g. in [CNW10].

lead to $O(\ell^3)$ dependence (for $\epsilon \propto \ell^{-1}$). This is because we would have to take $O(\frac{\ell}{\epsilon^2})$ classical samples for each $i$ and producing a quantum sample costs at least $O(\ell)$. The advantage of our approximation procedure based on quantum phase estimation is that it requires only one quantum sample (or more precisely, $\log \ell$, after using the powering lemma to boost the success probability). We cannot obtain the second speed-up without using quantum samples (as mentioned in the introduction, this prevents us from using a procedure such as [RIC07] that prepares density operators encoding stationary distributions). Also, the arguments employed in the error analysis in the quantum case are quite different from those in the classical error analysis.

Each classical FPRAS we speed up uses the telescoping trick (6.4), a particular cooling schedule (decreasing sequence of temperatures), and slowly-varying Markov chains which mix rapidly, with stationary distributions equal to the Boltzmann distributions at the intermediate temperatures. The classical FPRAS is useful only when we have the Markov chains with the required properties. Moreover, the cooling schedules need to be such that the ratios $\alpha_i$ (6.9) are lower bounded by some $c^{-1}$. In [SVV07], the authors show that it is possible to use a cooling schedule $T_0 = \infty > T'_1 > \ldots > T'_{\ell'-1} = T_F$ for estimating the partition function $Z(T_F)$ as long as for each $i$,

$$\frac{\mathbf{E}\left(Y_i^2\right)}{\left(\mathbf{E}\left(Y_i\right)\right)^2} \leq b, \tag{6.58}$$

where $b$ is some constant. Such a cooling schedule is called a Chebyshev cooling schedule. Note that the above condition is automatically satisfied in the situation we consider in this chapter, but not vice versa (recall that we assume that we have a cooling schedule such

that $\mathbf{E}(Y_i)$ is bounded from below by a constant for each $i$; we set it to $\frac{1}{2}$ for simplicity of presentation). The advantage of Chebyshev cooling schedules is that they are provably shorter. The authors present an adaptive algorithm for constructing Chebyshev cooling schedule. We plan to explore if it is possible to speed up this process. But even if this is possible, a potential obstacle remains. It is not clear whether we can still obtain the reduction from $\frac{1}{\epsilon^2}$ to $\frac{1}{\epsilon}$ when we only know that the condition (6.58) is satisfied. It seems that the condition $\mathbf{E}(Y_i) > c^{-1}$ with $c$ some constant is absolutely necessary for phase estimation to yield the quadratic speed-up with respect to the accuracy parameter $\epsilon$.

The combination of simulated annealing and the Monte Carlo Markov Chain method used in approximating partition functions is the central piece of the best currently known algorithm for estimating permanents with non-negative entries [BSVV08]. We therefore plan to explore where our techniques can be used to speed up this breakthrough classical algorithm.

# CHAPTER 7
# PREPARING THERMAL GIBBS STATE

While quantum walk may have lots of useful applications, there may be a situation in which the spectral gap of the corresponding transition matrix $P$ is extremely small. The quadratic speed-up might remain insignificant. In a situation such as this, we can use techniques like amplitude amplification.

The ability to efficiently prepare thermal Gibbs states of arbitrary quantum systems at arbitrary temperatures on a quantum computer would lead to a multitude of applications in condensed matter, quantum chemistry and high energy physics [PW09, TOVPV09, TD00]. For example, we could estimate partition and correlation functions of fermionic and frustrated systems. For these systems, the approach of first applying the "quantum-to classical map" [SUZ88] and then using the classical Monte Carlo method fails because the mapping does not conserve the positivity of statistical weights.

We consider an arbitrary Hamiltonian $H$ with spectral decomposition

$$H = \sum_{a=1}^{D} E_a |\psi_a\rangle\langle\psi_a| \,. \tag{7.1}$$

The *thermal Gibbs state* of the system at inverse temperature $\beta$ is given by

$$\rho_\beta := \sum_a \frac{e^{-\beta E_a}}{\mathcal{Z}_\beta} |\psi_a\rangle\langle\psi_a| \tag{7.2}$$

where $\mathcal{Z}_\beta := \sum_a e^{-\beta E_a}$ denotes the partition function.

The formal definition of preparing thermal Gibbs states is as follows:

**Problem** (Thermalizing quantum states)**.** *Let $H$ be a Hamiltonian, $\beta$ an inverse temperature and $\epsilon \in (0,1)$ a parameter describing the desired accuracy. We consider the problem to prepare a state $\tilde{\rho}_\beta$ that is $\epsilon$-close to the thermal Gibbs state $\rho_\beta$ with respect to trace distance*[1]*, i.e.,*

$$\|\rho_\beta - \tilde{\rho}_\beta\|_{\mathrm{tr}} \leq \epsilon. \tag{7.3}$$

*We refer to the process of preparing such state as thermalizing the quantum system. We seek to determine efficient quantum circuits that realize such thermalizing process.*

We assume that the energies satisfy $E_a \in [0, \frac{\pi}{4}]$. If we initially only know that the spectrum of the Hamiltonian $H$ is contained in the interval $[\ell, u]$, then the shifted and rescaled Hamiltonian $4(H - \ell I)/(\pi(u - \ell))$ satisfies the condition of this assumption. The thermal Gibbs state is invariant under shifting of the spectrum. Thus, we have to rescale the inverse temperature by multiplying it by $u - \ell$ when working with the new Hamiltonian.

There are two types of quantum algorithms for preparing thermal Gibbs states. The first is a generalization of the Metropolis algorithm. The Metropolis algorithm [MRRTT53] can be applied to the special case of classical systems, i.e., systems whose Hamiltonian $H = \sum_a E_a |a\rangle\langle a|$ is diagonal in the computational basis. It offers great flexibility for constructing Markov chains whose limiting distributions are equal to the desired thermal Gibbs distributions. The number of times we have to apply the Markov chain scales like $1/\delta$, where $\delta$ is its spectral gap. Bounding the spectral gap from below for arbitrary systems and

[1] *Recall that the trace distance is defined to be $\frac{1}{2}\mathrm{tr}\sqrt{XX^\dagger}$, where $X = \rho_\beta - \tilde{\rho}_\beta$.*

neighborhood structures is very difficult. However, it is possible to prove that the gap is sufficiently large for many practically relevant cases.

Recently, *Temme* et al. presented an extension of the Metropolis algorithm to quantum sytems [TOVPV09]. Their quantum Metropolis sampling makes it possible to implement quantum maps such that their fixed-points are approximately equal to the desired thermal Gibbs states. Analogously to classical case, the number of time we have to apply the quantum map depends on its spectral gap. The difficulty of bounding the gap from below remains for general systems and neighborhood structures. However, numerical experiments in [TOVPV09] show that the gap scales like $1/N$ for the spin-chain Hamiltonian $H = \sum_k X_k X_{k+1} + Y_k Y_{k+1} + g Z_k$ on $N$ spins.

The second type of algorithm is due to *Poulin* and *Wocjan* [PW09]. This algorithm behaves like a Las Vegas algorithm, i.e., it always produces a correct output $\tilde{\rho}$ satisfying the requirements of the problem definition. The time it takes this algorithm to terminate is a random variable. However, we can bound the expected value. It is dominated by the factor $\sqrt{D/\mathcal{Z}_\beta}$. This square root term occurs because this algorithm is based on an extension of Grover's state engineering technique.

Once the Grover sampling has terminated we know that we have prepared a state that is close to the desired thermal Gibbs state. In contrast, we can only guarantee that quantum Metropolis sampling yields a good approximation if we have a lower bound on the spectral gap. But, of course, quantum Metropolis sampling has the potential to outperform the Grover sampling for certain quantum systems.

We modify this Grover sampling and analyze the errors that arise due to imperfect simulation of Hamiltonian time evolutions and limited performance of phase estimation (finite accuracy and nonzero probability of failure) in more detail. This modification together with the tighter analysis allows us to prove a better running time. We show that expressing the effect of these sources of error on the overall complexity is smaller than in the original algorithm. We also think that the ideas underlying of our new analysis could also be used to prove a better performance of the above quantum Metropolis sampling.

This chapter is organized as follows. In section 7.1 we present the structure of the algorithm. We identify three sources of errors that arise due to (i) imperfect simulation of Hamiltonian time evolution, (ii) limited precision of phase estimation, and (iii) nonzero failure probability of phase estimation. In section 7.2.1 and 7.2.2 we analyze how the complexity increases when we seek to keep the errors small. Finally, we make our conclusion in Section 7.3.

## 7.1 Quantum Algorithm – Idealized Setting

To better explain the intuition behind the quantum algorithm, we first ignore all sources of error. We assume that the unitary $U = \exp(2\pi i H)$ can be implemented perfectly and efficiently. The eigenvalues $E_a$ of $H$ correspond to the eigenphases $E_a$ of $U$, using the convention that the phase of $e^{2\pi i E_a}$ is $E_a$. We assume that phase estimation (PE) makes it possible to perfectly resolve the eigenphases, i.e., there is an efficient quantum circuit

mapping $|\psi_a\rangle \otimes |00\ldots0\rangle$ onto $|\psi_a\rangle \otimes |E_a\rangle$ (this is the case as long as the energy $E_a$ can be written as binary fractions). The realistic case is analyzed in detail in the following section.

The algorithm prepares a purified Gibbs state of the form

$$|\beta\rangle = \sum_{a=1}^{D} \sqrt{\frac{e^{-\beta E_a}}{\mathcal{Z}_\beta}} \underbrace{|\psi_a\rangle}_{A} \otimes \underbrace{|\varphi_a\rangle}_{B} \otimes \underbrace{|E_a\rangle}_{\text{energy}} \otimes \underbrace{|0\rangle}_{\text{anc}} . \tag{7.4}$$

The states $|\varphi_a\rangle$ form an orthonormal basis on the $D$-dimensional subsystem $B$. The $|E_a\rangle$ are computational basis states of the energy register, which consists of multiple qubits. These basis states encode the eigenvalues $E_a$ of the eigenvectors $|\psi_a\rangle$ of $H$. The ancilla register consists of a single qubit. We obtain the thermal Gibbs state $\rho_\beta$ from $|\beta\rangle$ by tracing out the subsystems $B$, energy, and anc (see eqn. (7.4))

$$\rho_\beta = \mathrm{tr}_{\bar{A}}(|\beta\rangle\langle\beta|) , \tag{7.5}$$

where we use $\bar{A}$ to denote the collection of the above three subsystems (the complement of $A$).

The algorithms consists of the following steps:

**Algorithm 1** Thermal Gibbs State Preparation at Inverse Temperature $\beta$

**Input:** Prepare the maximally entangled state $|\nu\rangle = \frac{1}{\sqrt{D}} \sum_a |a\rangle|a\rangle$ on the subsystem $AB$.

**Step I:** Run phase estimation of $U$ on the $A$-part of $|\nu\rangle$. Write the eigenphase into the energy register.

**Step II:** Apply the controlled rotation $R = \sum_E |E\rangle\langle E| \otimes R_E$ where

$$R_E = \begin{pmatrix} \sqrt{e^{-\beta E}} & -\sqrt{1 - e^{-\beta E}} \\ \sqrt{1 - e^{-\beta E}} & \sqrt{e^{-\beta E}} \end{pmatrix}.$$

The control is the energy register and the target is the ancilla qubit that is initialized in $|0\rangle$. Denote the resulting state by $|\Psi\rangle$.

**Step III:** Use a variant of Grover to project $|\Psi\rangle$ onto the subspace in which the ancilla qubit is in $|0\rangle$. Denote the projector onto this subspace by $\Pi_0$. The Grover iteration is given by

$$G = (2|\Psi\rangle\langle\Psi| - I)(I - 2\Pi_0)\,.$$

**Output:** The density matrix $\rho_\beta$ of final state $|\beta\rangle$ by tracing out $\bar{A}$.

Let $V$ be an arbitrary unitary. The maximally entangled state $|\nu\rangle$ is invariant under the action of $V \otimes \bar{V}$, i.e., $(V \otimes \bar{V})|\nu\rangle = |\nu\rangle$. Consequently, we can rewrite $|\nu\rangle$ as

$$|\nu\rangle = \frac{1}{\sqrt{D}} \sum_a |\psi_a\rangle \otimes |\varphi_a\rangle \tag{7.6}$$

by setting $V = \sum_a |\psi_a\rangle\langle a|$ and $|\varphi_a\rangle = \bar{V}|a\rangle$. In step I, we obtain the state

$$|\Phi\rangle = \frac{1}{\sqrt{D}} \sum_a \Big(|\psi_a\rangle \otimes |\varphi_a\rangle\Big) \otimes |E_a\rangle \otimes |0\rangle. \tag{7.7}$$

In step II, we obtain the state

$$|\Psi\rangle = \frac{1}{\sqrt{D}} \sum_a \Big( |\psi_a\rangle \otimes |\varphi_a\rangle \Big) \otimes |E_a\rangle \otimes \Big( \sqrt{e^{-\beta E_a}} |0\rangle + \sqrt{1 - e^{-\beta E_a}} |1\rangle \Big). \tag{7.8}$$

Note that the desired purified Gibbs state $|\beta\rangle$ is equal to

$$\frac{\Pi_0 |\Psi\rangle}{\|\Pi_0 |\Psi\rangle\|}, \tag{7.9}$$

where $\|\Pi_0|\Psi\rangle\| = \sqrt{\frac{\mathcal{Z}_\beta}{D}}$. We apply the variant of Grover algorithm [BBH96], which makes it possible to prepare $|\beta\rangle$ with an expected number of Grover iterations $O(1/\|\Pi_0|\psi\rangle\|)$. It is important that we do not need to know the overlap $\|\Pi_0|\Psi\rangle\|$. This shows that we obtain

$$|\beta\rangle = \sum_a^D \sqrt{\frac{e^{-\beta E_a}}{\mathcal{Z}_\beta}} |\psi_a\rangle \otimes |\varphi_a\rangle \otimes |E_a\rangle \otimes |0\rangle \tag{7.10}$$

in step III.

## 7.2 Analysis for Imperfect Setting

### 7.2.1 Analysis of Simulation Error

The first source of error is the inability to implement $U = \exp(2\pi i H)$ perfectly for general $H$. Using techniques [BACS07, LLO96, ZAL98] for simulating Hamiltonian time evolutions, we can only implement a unitary $U_{\text{sim}}$ with $\|U - U_{\text{sim}}\| \leq \epsilon_{\text{sim}}$. The resources grow inversely with the desired accuracy $\epsilon_{\text{sim}}$.

To bound the error arising from imperfect simulation, we use the following result, which follows the discussion in [PW09, Appendix A].

**Lemma 8.** *Let $H$ be a Hamiltonian whose eigenvalues are contained in the interval $[0, \frac{\pi}{4}]$. Let $U = exp(2\pi i H)$ and $U_{\text{sim}}$ be a unitary with $\|U - U_{\text{sim}}\| \leq \epsilon_{\text{sim}}$. Then, there exists an effective Hamiltonian $H_{\text{sim}}$ such that $U_{\text{sim}} = \exp(2\pi i H_{\text{sim}})$ and $\|H - H_{\text{sim}}\| \leq \kappa \epsilon_{\text{sim}}$ where $\kappa$ is a constant.*

Assume phase estimation could perfectly resolve the eigenphases of $U_{\text{sim}}$. Then, if we ran the algorithm using $U_{\text{sim}}$ instead of $U$, then we would prepare the thermal state with respect to the effective Hamiltonian $H_{\text{sim}}$ instead of $H$. Thus, it remains to determine how close the corresponding thermal states are close to each other with respect to trace norm.

**Lemma 9.** *Let $H$ and $H_{\text{sim}}$ be as above. Then, the corresponding thermal states*

$$\rho := \frac{\exp(-\beta H)}{\text{tr}(\exp(-\beta H))} \quad \textit{and} \quad \rho_{\text{sim}} := \frac{\exp(-\beta H_{\text{sim}})}{\text{tr}(\exp(-\beta H_{\text{sim}}))} \tag{7.11}$$

*satisfy*

$$\|\rho - \rho_{\text{sim}}\|_{\text{tr}} \leq \frac{\epsilon}{2} \tag{7.12}$$

*provided that $\epsilon_{\text{sim}} \leq \epsilon^2/(8\kappa\beta)$.*

*Proof.* The fidelity of $\rho$ and $\rho_{\text{sim}}$ is given by

$$F(\rho, \rho_{\text{sim}}) = \text{tr}\sqrt{\sqrt{\rho}\,\rho_{\text{sim}}\,\sqrt{\rho}}\,. \tag{7.13}$$

Using [FG99, Proposition 4] we bound the trace distance between $\rho$ and $\rho_{\text{sim}}$ as follows

$$\|\rho - \rho_{\text{sim}}\|_{\text{tr}} \leq \sqrt{1 - F(\rho, \rho_{\text{sim}})^2}\,. \tag{7.14}$$

The analysis in [PW09, Appendix C] shows that

$$F(\rho, \rho_{\text{sim}}) \geq e^{-\beta\kappa\epsilon_{\text{sim}}}, \tag{7.15}$$

and thus

$$\|\rho - \rho_{\text{sim}}\|_{\text{tr}} \leq \sqrt{1 - e^{-2\beta\kappa\epsilon_{\text{sim}}}} \leq \sqrt{2\beta\kappa\epsilon_{\text{sim}}} \leq \frac{\epsilon}{2}. \tag{7.16}$$

The rightmost inequality follows from $1 + x \leq e^x$ for all $x \in \mathbb{R}$. □

From now on, we measure the complexity in terms of how many times we have to invoke a controlled version of $U_{\text{sim}}$. If we wish to determine the complexity in terms of elementary gates, we have to look at the simulation technique more closely.

### 7.2.2 Analysis of Errors in Phase Estimation

We now show how to prepare a state $\tilde{\rho}$ such that $\|\rho_{\text{sim}} - \tilde{\rho}\|_{\text{tr}} \leq \epsilon/2$, implying that $\|\rho - \tilde{\rho}\|_{\text{tr}} \leq \epsilon$ as desired. We analyze the three phases of the algorithm.

#### 7.2.2.1 Phase I

We need to run a special variant of phase estimation [NWZ09] of $U_{\text{sim}}$ on $|\nu\rangle$. We briefly explain how it works. To avoid new definitions, we use $|\psi_a\rangle$ and $E_a$ to refer to the eigenvectors and eigenphases of $U_{\text{sim}}$, respectively.

The usual phase estimation algorithm consists of the following steps [KLM07]. The energy register consists of $n = \lceil \log_2(1/\epsilon_{\text{prec}}) \rceil$ qubits. We apply the Hadamard transform to each

of the qubits of the energy register, the controlled-$U_{\text{sim}}^{2^j}$ gates (controlled by the $j$th qubit of the energy register) on the $A$-part of $\nu$, and the inverse quantum Fourier transform $F^\dagger$ on the energy register. We measure the $n$ qubits of the energy register in the computational basis and interpret the outcome $b \in [0, 2^n - 1]$ as the binary fraction $\hat{E}_b := b/2^n$, which is a very good estimate for $E_a$. More precisely, the probability of obtaining the estimate $\hat{E}_b$ is given by

$$\Pr(E_a, \hat{E}_b) = \frac{1}{2^{2n}} \frac{\sin^2(\pi 2^n (E_a - \hat{E}))}{\sin^2(\pi(E_a - \hat{E}_b))} \,. \tag{7.17}$$

We use $|\hat{E}_b\rangle$ to denote the computational basis state $|b\rangle$, which encodes the energy value $\hat{E}_b$. Let $E_a^\pm$ denote the binary fractions that are closest to $E_a$, where we use the convention $E_a^- \leq E_a < E_a^+$. It follows that the probability of obtaining $E_a^+$ or $E_a^-$ is greater or equal to $\frac{8}{\pi^2} \geq \frac{3}{4}$. Thus, the probability of failure, i.e., the probability of not obtaining one of the closest $n$-bit fractions, is less than $\frac{1}{4}$.

To reduce the probability of failure to $\epsilon_{\text{fail}}$, we repeat this quantum circuit $k = \lceil \log_2(1/\epsilon_{\text{fail}}) \rceil$ times, each time recording the estimate into a new energy register and adjoin a median register that consists of $n$ qubits. This yields the state $|\Upsilon\rangle$

$$\frac{1}{\sqrt{D}} \sum_a |\psi_a\rangle |\varphi_a\rangle \otimes \Big( \sum_{b_1} c_{E_a, \hat{E}_{b_1}} |\hat{E}_{b_1}\rangle_{\text{energy}} \otimes \ldots \otimes \sum_{b_k} c_{E_a, \hat{E}_{b_k}} |\hat{E}_{b_k}\rangle_{\text{energy}} \Big) \otimes |0 \ldots 0\rangle_{\text{median}} \otimes |0\rangle_{\text{anc}} \,, \tag{7.18}$$

where the amplitudes $c_{E_a, \hat{E}_{b_\ell}}$ satisfy $|c_{E_a, \hat{E}_{b_\ell}}|^2 = \Pr(E_a, \hat{E}_{b_\ell})$ for $\ell = 1, \ldots, k$.

The median circuit determines the median of $\hat{E}_{b_1}, \ldots, \hat{E}_{b_k}$ and writes it into the median register. Reordering the registers, we may write the resulting states as

$$|\tilde{\Upsilon}\rangle = \frac{1}{\sqrt{D}} \sum_a |\psi_a\rangle|\varphi_a\rangle \otimes \left( c_a^{\pm} |E_a^{\pm}\rangle_{\text{median}} \otimes |\mu_a^{\pm}\rangle_{\text{energy}^{\otimes n}} + |\xi_a\rangle_{\text{median} \otimes \text{energy}^{\otimes n}} \right) \otimes |0\rangle_{\text{anc}} \,. \quad (7.19)$$

where

- the states $|\mu_a^{\pm}\rangle$ are supported only on the states $|\hat{E}_{b_1}\rangle \otimes \cdots \otimes |\hat{E}_{b_k}\rangle$ such
- that the median of $\hat{E}_{b_1}, \ldots, \hat{E}_{b_k}$ is equal to $E_a^{\pm}$, and
- the states $|\xi_a\rangle$ are orthogonal to $|E_a^{\pm}\rangle \otimes |\mu_a^{\pm}\rangle$.

It follows from the analysis in [NWZ09, JVV86] that the amplitudes $c_a^{\pm}$ satisfy

$$1 - \epsilon_{\text{fail}} \leq |c_a^{+}|^2 + |c_a^{-}|^2 \leq 1 \,, \quad 0 < \| |\xi_a\rangle \|^2 \leq \epsilon_{\text{fail}} \,. \quad (7.20)$$

This means that the probability of the median not being one of the closest binary fractions $E_a^{\pm}$ to $E_a$ is less than or equal to $\epsilon_{\text{fail}}$. The advantage of combining phase estimation with the powering technique for approximation algorithms is that we only need to invoke a controlled version of $U_{\text{sim}}$

$$\lceil (1/\epsilon_{\text{prec}}) \log(1/\epsilon_{\text{fail}}) \rceil \quad (7.21)$$

instead of $O((1/\epsilon_{\text{prec}})(1/\epsilon_{\text{fail}}))$ when using phase estimation alone [NC00].

To keep the notation simple, we use $|E_a^{\pm}\rangle$ to denote the tensor product $|E_a^{\pm}\rangle \otimes |\mu_a^{\pm}\rangle$. Using this convention, we write the state after step I (phase estimation) as

$$|\tilde{\Phi}\rangle = \frac{1}{\sqrt{D}} \sum_a |\psi_a\rangle|\varphi_a\rangle \otimes (c_a^{\pm} |E_a^{\pm}\rangle + |\xi_a\rangle) \otimes |0\rangle \,. \quad (7.22)$$

#### 7.2.2.2 Phase II

The $R$-operation is controlled by the energy value contained in the median register. After step II, $|\tilde{\Phi}\rangle$ evolves to the state

$$|\tilde{\Psi}\rangle = \underbrace{\frac{1}{\sqrt{D}}\sum_a |\psi_a\rangle|\varphi_a\rangle \otimes c_a^{\pm}|E_a^{\pm}\rangle \otimes \left(\sqrt{e^{-\beta E_a^{\pm}}}|0\rangle + \sqrt{1-e^{-\beta E_a^{\pm}}}|1\rangle\right)}_{|\psi\rangle} + \tag{7.23}$$

$$\underbrace{\frac{1}{\sqrt{D}}\sum_a |\psi_a\rangle|\varphi_a\rangle \otimes R(|\xi_a\rangle \otimes |0\rangle)}_{|\xi\rangle}\ . \tag{7.24}$$

As a consequence of the property $\langle\xi|\psi\rangle = 0$, we have

$$\||\xi\rangle\|^2 = \frac{1}{D}\sum_a \langle\xi_a|\xi_a\rangle \leq \epsilon_{\text{fail}} \quad \text{and} \quad 1-\epsilon_{\text{fail}} \leq \||\psi\rangle\|^2 \leq 1. \tag{7.25}$$

#### 7.2.2.3 Phase III

Let $|\tilde{\beta}\rangle$ be the state obtained by applying Grover's algorithm to $|\tilde{\Psi}\rangle$, i.e.,

$$|\tilde{\beta}\rangle = \frac{\Pi_0|\tilde{\Psi}\rangle}{\|\Pi_0|\tilde{\Psi}\rangle\|}. \tag{7.26}$$

Let $\tilde{\rho}$ be the reduced density operator of $|\tilde{\beta}\rangle$ over $\bar{A}$. We need to bound $\|\Pi_0|\tilde{\Psi}\rangle\|$ from below to obtain an upper bound on the expected number of Grover iterations. We also need to show that $\tilde{\rho}$ is close to $\rho_{\text{sim}}$. This is done in the following lemma.

**Lemma 10.** *Let $\tilde{\rho} = \text{tr}_{\bar{A}}(|\tilde{\beta}\rangle\langle\tilde{\beta}|)$. This density operator has the form*

$$\tilde{\rho} = \frac{\text{tr}_{\bar{A}}\big(\Pi_0|\psi\rangle\langle\psi|\Pi_0\big)}{\langle\tilde{\Psi}|\Pi_0|\tilde{\Psi}\rangle} + \frac{\text{tr}_{\bar{A}}\big(\Pi_0|\xi\rangle\langle\xi|\Pi_0\big)}{\langle\tilde{\Psi}|\Pi_0|\tilde{\Psi}\rangle} \tag{7.27}$$

*that satisfies*

$$\|\rho_{\text{sim}} - \tilde{\rho}\|_{\text{tr}} \leq \|\rho_{\text{sim}} - \frac{\text{tr}_{\bar{A}}(\Pi_0(|\psi\rangle\langle\psi|)\Pi_0)}{\langle\tilde{\Psi}|\Pi_0|\tilde{\Psi}\rangle}\|_{\text{tr}} + \|\frac{\text{tr}_{\bar{A}}(\Pi_0(|\xi\rangle\langle\xi|)\Pi_0)}{\langle\tilde{\Psi}|\Pi_0|\tilde{\Psi}\rangle}\|_{\text{tr}} \leq \frac{\epsilon}{4} + \frac{\epsilon}{4} = \frac{\epsilon}{2}\,, \quad (7.28)$$

*provided that* $\epsilon_{\text{fail}} = e^{-\beta}\,\epsilon^2$ *and* $\epsilon_{\text{prec}} = \epsilon/(32\beta)$.

*Proof.* Observe that the off-diagonal terms $\Pi_0(|\psi\rangle\langle\xi| + |\xi\rangle\langle\psi|)\Pi_0$ in $\left(\Pi_0|\tilde{\Psi}\rangle\langle\tilde{\Psi}|\Pi_0\right)$ vanish when we trace $|\tilde{\beta}\rangle$ over $\bar{A}$.

Set $N := \text{tr}(\Pi_0|\tilde{\Psi}\rangle\langle\tilde{\Psi}|\Pi_0)$ and define the operator

$$\sigma := \text{tr}_{\bar{A}}(\Pi_0|\psi\rangle\langle\psi|\Pi_0) = \frac{1}{D}\sum_a (|c_a^+|^2 e^{-\beta E_a^+} + |c_a^-|^2 e^{-\beta E_a^-})|\psi_a\rangle\langle\psi_a|\,. \quad (7.29)$$

We can express $N = \text{tr}(\sigma) + \text{tr}(\Pi_0|\xi\rangle\langle\xi|\Pi_0)$. Since $\text{tr}(\Pi_0|\xi\rangle\langle\xi|\Pi_0) = \langle\xi|\Pi_0|\xi\rangle \leq \|\xi\|^2 \leq \epsilon_{\text{fail}}$, by (7.25) we can bound $N$ and $\sigma$'s trace as follows

$$(1 - \epsilon_{\text{fail}})\,\frac{\mathcal{Z}_\beta}{D}\,e^{-\beta\epsilon_{\text{prec}}} \leq \text{tr}(\sigma) \leq \frac{\mathcal{Z}_\beta}{D}\,e^{\beta\epsilon_{\text{prec}}}\,, \quad (7.30)$$

$$(1 - \epsilon_{\text{fail}})\,\frac{\mathcal{Z}_\beta}{D}\,e^{-\beta\epsilon_{\text{prec}}} \leq N < \epsilon_{\text{fail}} + \frac{\mathcal{Z}_\beta}{D}\,e^{\beta\epsilon_{\text{prec}}}. \quad (7.31)$$

Because $E_a \in [0, \frac{\pi}{4}]$, we can bound the ratio $\mathcal{Z}(\beta)/D$ as follows

$$1 \geq \mathcal{Z}(\beta)/D \geq e^{-\beta}. \quad (7.32)$$

By choosing the lower bound on $N$ and the upper bound on $\|\text{tr}_{\bar{A}}(\Pi_0(|\xi\rangle\langle\xi|)\Pi_0)\|_{\text{tr}}$, we obtain

$$\|\frac{\text{tr}_{\bar{A}}(\Pi_0(|\xi\rangle\langle\xi|)\Pi_0)}{N}\|_{\text{tr}} \leq \frac{\epsilon_{\text{fail}} \cdot e^{\beta\epsilon_{\text{prec}}}}{(1 - \epsilon_{\text{fail}})\frac{\mathcal{Z}_\beta}{D}} \leq \frac{\epsilon^2 \cdot e^{\beta\epsilon_{\text{prec}}}}{(1 - \epsilon_{\text{fail}})} \leq 2\epsilon^2 e^{\beta\epsilon_{\text{prec}}} \leq \frac{\epsilon}{4}.$$

The last inequality is obtained because $\epsilon$ is small and $e^x < 1 + 2x$ for $x \in [0, 1]$. The term

$$\|\rho_{\text{sim}} - \frac{\text{tr}_{\bar{A}}(\Pi_0(|\psi\rangle\langle\psi|)\Pi_0)}{N}\|_{\text{tr}} = \sum_{i=1}^{d} |\frac{e^{-\beta E_i}}{\sum_j e^{-\beta E_j}} - \frac{\text{tr}(\sigma)}{N}| \leq \frac{\epsilon}{4} \tag{7.33}$$

is still satisfied even when examining the following two extreme cases

$$\text{(I) Lower bound on } N \text{ and upper bound on } \text{tr}(\sigma): \quad \frac{e^{\beta\epsilon_{\text{prec}}}}{(1-\epsilon_{\text{fail}})e^{-\beta\epsilon_{\text{prec}}}} - 1\,, \tag{7.34}$$

$$\text{(II) Upper bound on } N \text{ and lower bound on } \text{tr}(\sigma): \quad 1 - \frac{(1-\epsilon_{\text{fail}})e^{-\beta\epsilon_{\text{prec}}})}{\frac{D}{\mathcal{Z}_\beta}\epsilon_{\text{fail}} + e^{\beta\epsilon_{\text{prec}}}}. \tag{7.35}$$

We know that

$$\sum_{i=1}^{d} |\frac{e^{-\beta E_i}}{\sum_j e^{-\beta E_j}} - \frac{\text{tr}(\sigma)}{N}| \leq \max\left\{ \frac{e^{\beta\epsilon_{\text{prec}}}}{(1-\epsilon_{\text{fail}})e^{-\beta\epsilon_{\text{prec}}}} - 1, 1 - \frac{(1-\epsilon_{\text{fail}})e^{-\beta\epsilon_{\text{prec}}})}{\frac{D}{\mathcal{Z}_\beta}\epsilon_{\text{fail}} + e^{\beta\epsilon_{\text{prec}}}} \right\}. \tag{7.36}$$

Because $1 + 2x > e^x$ for $\forall x \in (0,1)$, we derive

$$\frac{e^{\beta\epsilon_{\text{prec}}}}{(1-\epsilon_{\text{fail}})e^{-\beta\epsilon_{\text{prec}}}} - 1 \leq \frac{1+\frac{\epsilon}{8}}{1-\epsilon_{\text{fail}}} - 1 \leq \frac{\epsilon}{4}. \tag{7.37}$$

In the second case because $1 + x \leq e^x$ for $\forall x \in \mathbb{R}$ and $1 \leq D/\mathcal{Z}(\beta) \leq e^\beta$, we have

$$1 - \frac{(1-\epsilon_{\text{fail}})e^{-\beta\epsilon_{\text{prec}}})}{\frac{D}{\mathcal{Z}_\beta}\epsilon_{\text{fail}} + e^{\beta\epsilon_{\text{prec}}}} \leq 1 - \frac{e^{-\beta\epsilon_{\text{prec}}}}{\epsilon^2 + e^{\beta\epsilon_{\text{prec}}}} \leq 1 - \frac{1-\frac{\epsilon}{16}}{\epsilon^2+1} \leq \frac{\epsilon}{4} \tag{7.38}$$

for small $\epsilon$. □

## 7.3 Discussion

**Theorem 15.** *Let $H$ be a Hamiltonian, $\beta$ an inverse temperature and $\epsilon \in (0,1)$ a parameter describing the desired accuracy. Let $U_{\text{sim}}$ be the Hamiltonian simulation such that $\|U -$*

$U_{\text{sim}}\| \leq \epsilon_{\text{sim}}$ *where* $U = \exp(2\pi i H)$. *Our algorithm prepares a state* $\tilde{\rho}_\beta$ *that is* $\epsilon$*-close to the thermal Gibbs state* $\rho_\beta$*, i.e.,*

$$\|\rho_\beta - \tilde{\rho}_\beta\|_{\text{tr}} \leq \epsilon, \tag{7.39}$$

*provided that* $\epsilon_{\text{sim}} \leq \epsilon^2/(8\beta\kappa)$, $\epsilon_{\text{prec}} = \epsilon/(32\beta)$ *and* $\epsilon_{\text{fail}} = e^{-\beta}\epsilon^2$. *The complexity of our algorithm scales like*

$$O\Big(\sqrt{\frac{D}{\mathcal{Z}_\beta}}\,\frac{\beta}{\epsilon}\,(\log\frac{1}{\epsilon} + \beta)\Big) \tag{7.40}$$

*in terms of the number of invocations of the controlled-*$U_{\text{sim}}$ *operation.*

*Proof.* The requirements for $\epsilon_{\text{sim}}$, $\epsilon_{\text{prec}}$ and $\epsilon_{\text{fail}}$ are immediate by *Lemma 9* and *Lemma 10*. By (7.21) the cost for performing one Grover iteration scales as

$$O\Big(\frac{\beta}{\epsilon}\,(\log\frac{1}{\epsilon} + \beta)\Big). \tag{7.41}$$

The number of Grover iterations [BBH96] is determined by $O(\frac{1}{\text{tr}(\sigma)}) = O(\sqrt{\frac{D}{\mathcal{Z}_\beta}})$ when using the lower bound of $\text{tr}(\sigma)$ in (7.30).

# CHAPTER 8
# THEORY OF PERTURBED QUANTUM WALK

Markov chains and random walks have been useful tools in classical computation. One can use random walks to obtain the final stationary distribution of a Markov chain to sample from. In such an application the time the Markov chain takes to converge, i.e., *convergence time*, is of interest because shorter convergence time means lower cost in generating a sample. Sampling from stationary distributions of Markov chains combined with simulated annealing is the core of many clever classical approximation algorithms. For instance, approximating the volume of convex bodies [LV06], approximating the permanent of a non-negative matrix [JSV04], and the partition function of statistical physics models such as the Ising model [JS93] and the Potts model [BSVV08]. In addition, one can also use the random walks to search for the *marked* state, in which the *hitting time* is of interest because hitting time indicates the time it requires to find the marked state.

In comparison to classical random walks, quantum walks provide a quadratic speed-up in hitting time. Quantum walks have been applied to solving many interesting problems [SAN08], such as searching problems, group commutativity, element distinctness, restricted range associativity, triangle finding in a graph, and matrix product verification. Perturbations of classical Markov chains are widely studied with respect to hitting time and stationary distribution. Since a quantum system is susceptible to the environmental noise, we are interested to know what effect perturbation has on currently existing quantum walk based

algorithms.

This chapter is organized as follows. In section 8.1 we present the deviation effect of perturbation on the spectral gap of a classical Markov chain. In section 8.2 we discuss how the hitting time would be affected because of the perturbation. We explore the upper bounds for the perturbed hitting time quantumly and the time difference (delayed perturbed hitting time) both quantumly and classically. In section 8.3 we investigate the effect of perturbation on the quantum sample prepared by quantum walks. Finally in section 8.4, we make our conclusion.

## 8.1 Classical Spectral Gap Perturbation

Given a stochastic symmetric matrix $P \in \mathbb{C}^{n\times n}$, we can quantize the Markov chain [SZE04]. P. Wocjan, D. Nagaj and I showed that the implementation of one step of a quantum walk [CNW10] can be achieved efficiently. However, the above settings always are under the assumption of perfect scenarios. In real life there are many sources of errors that would perturb the process. Noise might be propagated along with the input source or might be introduced during the process. Here we look solely at the noise that are introduced at the beginning of the process.

The noise can be introduced due to the precision limitation and the noisy environment. For instance, not all numbers have a perfect binary representation and the approximated numbers would cause perturbation. Suppose our input decoding mechanism can always take the input matrix and represent it in a symmetric transition matrix $Q$, where $Q$ can be perfectly represented and this is the matrix closest to the original matrix $P$ that the system can prepare.

Let $E$ be the noise that is introduced because of system's precision limitation and the environment, we can express the transition matrix as

$$Q = P + E. \tag{8.1}$$

Classically, much research [IN09, CM01, GL96, PAR98, BF60, EI99] has focused on the spectral gaps and stationary distributions of the matrices with perturbation. In a recent work by Ipsen and Nadler [IN09] , they refined the perturbation bounds for eigenvalues of Hermitians. Throughout the rest of the chapter, $\|\cdot\|$ always denotes the $l_2$ norm, unless otherwise specified. Based on their result, we summarized the following:

**Corollary 1.** *Suppose $P$ and $Q \in \mathbb{C}^{n\times n}$ are Hermitian symmetric transition matrices with respective eigenvalues*

$$0 < \lambda_{n-1}(P) \leq \ldots \leq \lambda_0(P) = 1, \qquad 0 < \lambda_{n-1}(Q) \leq \ldots \leq \lambda_0(Q) = 1, \tag{8.2}$$

*and $Q = P + E$, then*

$$\max_{0 \le i \le n-1} |\lambda_i(P) - \lambda_i(Q)| \le \|E\|. \tag{8.3}$$

*Furthermore, the spectral gap $\delta$ of $P$ and the spectral gap $\Delta$ of $Q$ have the following relationship*

$$\delta - \|E\| \le \Delta \le \delta + \|E\|. \tag{8.4}$$

*Proof.* Eq. (8.3) is a direct result from the Weyl's Perturbation Theorem. The *Weyl's Perturbation Theorem* bounds the worst-case absolute error between the $i$th exact and the $i$th perturbed eigenvalues of Hermitian matrices in terms of the $l_2$ norm [GL96, PAR98]. And since

$$1 - \lambda_1(P) = \delta, \qquad 1 - \lambda_1(Q) = \Delta,$$

by eq. (8.3) we have $|\delta - \Delta| \le \|E\|$. Therefore, in general we can bound the perturbed spectral gap $\Delta$ as

$$\delta - \|E\| \le \Delta \le \delta + \|E\|.$$

Generally speaking, the global norm of $E$ might be very large when the dimensions $n >> 1$ [JOH01]. However, in our case because $E$ is the difference between two very close stochastic symmetric matrices, its global norm would never become large.

## 8.2 Hitting Time of Markov Chain Based Walks

For the purpose of being complete, we need to cite several definitions and results used in the MNRS algorithm [MNRS09] in this section. We recommend interested readers to reference [MNRS09] for details.

Let $P$ be a reversible and ergodic transition matrix with state space $\Omega$ and positive eigenvalues. Suppose $P$ is column-wise stochastic and $|\Omega| = n$, then let the Markov chain $(X_1, \ldots, X_n)$ under discussion have a finite state space $\Omega$ and transition matrix $P$.

**Definition 14.** *For $x \in \Omega$, denote the* hitting time *for $x$*

$$HT(P, x) = \min\{t \geq 1 : X_t = x\}. \tag{8.5}$$

*$HT(P, x)$ is the expected number of transition matrix $P$ invocations to reach the state $x$ when started in the initial distribution $\pi$.*

**Definition 15.** *For an $n \times n$ matrix $P$, $P_{-x}$ denotes the $(n-1) \times (n-1)$ matrix of $P$ where the row and column indexed by $x$ are deleted. For a vector $v$, $v_{-x}$ is the vector that omits the $x$-coordinate of $v$. Similarly, suppose $\{M\} = \{x_1, \ldots, x_m\}$, then $P_{-\{M\}}$ denotes the $(n-m) \times (n-m)$ matrix of $P$ where the rows and columns indexed by $x_1, x_2, \ldots,$and $x_m$ are deleted.*

**Definition 16.** *Denote the vector space* $\mathcal{H} = \mathbb{C}^{|\Omega|\times|\Omega|}$. *For a state* $|\psi\rangle \in \mathcal{H}$, *define* $\Pi_\psi = |\psi\rangle\langle\psi|$ *as the orthogonal projector onto* $Span(|\psi\rangle)$. *Let* $\mathcal{A} = Span(|y\rangle|p_y\rangle : y \in \Omega)$ *be the vector subspace of* $\mathcal{H}$ *where*

$$|p_y\rangle = \sum_{z\in\Omega} \sqrt{p_{zy}}|z\rangle. \tag{8.6}$$

$\mathcal{A}$ *is spanned by a set of mutually orthogonal states* $\{|\psi_i\rangle : i = 1, 2\ldots, |\Omega|\}$, *then let* $\Pi_\mathcal{A} = \sum_i \Pi_{\psi_i}$. *Similarly,* $\mathcal{A}_{-x} = Span\{|y\rangle|p_y\rangle : y \in \Omega\backslash\{x\}\}$.

**Definition 17.** *The unitary operation* $W(P) = (S\cdot(2\Pi_\mathcal{A} - I))^2$ *defined on* $\mathcal{H}$ *is the quantum analog of* $P$. *Similarly, the unitary operation* $W(P, x) = (S\cdot(2\Pi_{\mathcal{A}_{-x}} - I))^2$ *defined on* $\mathcal{H}$ *is the quantum analog of* $P_{-x}$. $S$ *is the swap operation defined by* $S|y\rangle|z\rangle = |z\rangle|y\rangle$.

**Fact 2.** *[MNRS09, SZE04] Let* $x \in \Omega$ *and* $|\mu\rangle = |x\rangle|p_x\rangle$. *Let* $U_2 = S(2\Pi_\mathcal{A} - I)$ *and* $U_1 = I - 2|\mu\rangle\langle\mu|$. *When* $P$ *is reversible, then* $U_2^2 = W(P)$ *and* $(U_2U_1)^2 = (S(2\Pi_{\mathcal{A}_{-x}} - I))^2 = W(P, x)$.

*Proof.* Since $\Pi_\mathcal{A} = \left(\sum_{y=1,y\neq x}^{|\Omega|} |y\rangle|p_y\rangle\langle y|\langle p_y|\right) + |\mu\rangle\langle\mu|$, then we have

$$\begin{aligned} U_2U_1 &= S(2\Pi_\mathcal{A} - I)(I - 2|\mu\rangle\langle\mu|) \\ &= S(2\Pi_\mathcal{A} - 2|\mu\rangle\langle\mu| - I) \\ &= S(2\Pi_{\mathcal{A}_{-x}} - I) \end{aligned}$$

### 8.2.1 Classical Hitting Time

By [MNRS09], the $x$-hitting time of $P$ can be expressed as $HT(P,x) = \pi^{\dagger}(I - P_{-x})^{-1}u_{-x}$, where $u$ is an all-ones vector. It is known that

$$\pi_{-x}^{\dagger}(I - P_{-x})^{-1}u_z = \sqrt{\pi_{-x}}^{\dagger}(I - S_{-x})^{-1}\sqrt{\pi_{-x}} \tag{8.7}$$

where $S_{-x} = \sqrt{\Pi_{-x}}P_{-x}\sqrt{\Pi_{-x}}^{-1}$ with $\Pi_{-x} = diag(\pi_i)_{i\neq x}$ and $\sqrt{\pi_{-x}}$ is the entry-wise square root of $\pi_{-x}$. Let $\{v_j : j \leq n-1\}$ be the set of normalized eigenvectors of $S_{-x}$ where the eigenvalue of $v_j$ is $\lambda_j = \cos\theta_j$ with $0 \leq \theta_j < \pi/2$. By reordering the eigenvalues, let us assume that $1 > \lambda_1 \geq \ldots \geq \lambda_{n-1} > 0$. When $\sqrt{\pi_{-x}} = \sum_j \nu_j v_j$ is the decomposition of $\sqrt{\pi_{-x}}$ in the eigenbasis of $S_{-x}$, the $x$-hitting time satisfies

$$HT(P,x) = \sum_j \frac{\nu_j^2}{1-\lambda_j}. \tag{8.8}$$

In a similar manner, let $\tilde{S}_{-x} = \sqrt{\Pi_{-x}}Q_{-x}\sqrt{\Pi_{-x}}^{-1}$ for a perturbed matrix $Q$ where $Q = P + E$. Then the $x$-hitting time for $Q$ satisfies

$$HT(Q,x) = \sum_j \frac{\tilde{\nu}_j^2}{1-\tilde{\lambda}_j}, \tag{8.9}$$

where $\tilde{\lambda}_j$ are the eigenvalues of $\tilde{S}_{-x}$ and $\sqrt{\pi_{-x}} = \sum_j \tilde{\nu}_j\tilde{v}_j$ is the decomposition of $\sqrt{\pi_{-x}}$ in the eigenbasis, $\{\tilde{v}_j\}$, of $\tilde{S}_{-x}$.

Three simple facts can be observed from above description of classical hitting time.

**Fact 3.** *$S_{-x}$ and $P_{-x}$ are similar, they have the **same eigenvalues**.*

**Fact 4.** *$\tilde{S}_{-x}$ and $Q_{-x}$ are* similar*, they have the* ***same eigenvalues****.*

**Fact 5.** *Since the entries of distribution $\pi$ sum up to 1 , i.e. $\sum_i(\pi_i) = 1$, then it is obvious that $\sqrt{\pi_{-x}}^{\dagger}\sqrt{\pi_{-x}} = \sum_i(\pi_i)_{i\neq x} \leq 1$. Hence we know that $\sum_{\mathbf{i}} \boldsymbol{\nu_i^2} = \sum_{\mathbf{i}} \tilde{\boldsymbol{\nu}}_{\mathbf{i}}^{\mathbf{2}} \leq \mathbf{1}$.*

### 8.2.2 Delayed Perturbed Hitting Time

In this subsection, we define the delayed perturbed hitting time and its upper bound as the following.

**Lemma 11.** *For a Markov transition matrix $P$ with state space $\Omega$ and limiting distribution $\pi$. Assume $|\Omega| = n$ and let $|v_i\rangle$ be the eigenvector with corresponding eigenvalue $\lambda_i$ of $P_{-x}$. Suppose the eigenvalues of $P_{-x}$ are ordered such that $1 > \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1} > 0$. The $x$-hitting time satisfies*

$$HT(P, x) = \sum_{j=1} \frac{\nu_j^2}{1 - \lambda_j}$$

*where $\sqrt{\pi_{-x}} = \sum_{j=1}^{|\Omega|-1} \nu_j |v_i\rangle$. When given a perturbed matrix $Q$ where $\|Q - P\| \leq \|E\|$, denote the Delayed Perturbed Hitting Time (DPHT(P,Q,x)) that*

$$DPHT(P, Q, x) = HT(Q, x) - HT(P, x).$$

*$DPHT(P, Q, x)$ can be bounded from above by*

$$\frac{1}{1 - \lambda_1 - \|E\|_2} - \frac{1}{1 - \lambda_1 + \gamma} \tag{8.10}$$

*where* $\lambda_1 - \lambda_{n-1} = \gamma$.

*Proof.* Let the eigenvalues of $Q_{-x}$ be $\tilde{\lambda}_i$. By the fact $\|Q-P\| \leq \|E\|$ and Weyl's perturbation theorem, we know that $\|Q_{-x} - P_{-x}\| \leq \|E\|$ and $|\lambda_i - \tilde{\lambda}_i| \leq \|E\|$. The delayed hitting time due to perturbation is thus

$$\begin{aligned} DPHT(P,Q,x) &= HT(Q,x) - HT(P,x) \\ &= \sum_{i\in\Omega} \Big( \frac{\tilde{\nu}_i^2}{1-\tilde{\lambda}_i} - \frac{\nu_i^2}{1-\lambda_i} \Big) \\ &\leq \Big( \sum_{i\in\Omega} \frac{\tilde{\nu}_i^2}{1-\lambda_1 - \|E\|} \Big) - \Big( \sum_{i\in\Omega} \frac{\nu_i^2}{1-\lambda_{n-1}} \Big) \\ &\leq (\frac{1}{1-\lambda_1 - \|E\|} - \frac{1}{1-\lambda_1+\gamma}), \end{aligned} \tag{8.11}$$

the last inequality is a result from *Fact 5*.

### 8.2.3 Upper Bound for Perturbed Quantum Hitting Time

Given two Hermitian stochastic matrices, $P$ and $Q$, we explore the difference between walk operators, $W(P)$ and $W(Q)$, with respect to their hitting time. Denote the set of marked elements as $|M|$. Based on the result from *Corollary 1*, we have the following:

**Corollary 2.** *Given two symmetric reversible ergodic transition matrices* $P$ *and* $Q \in \mathbb{C}^{n\times n}$*, where* $Q = P + E$*, let* $W(P)$ *and* $W(Q)$ *be quantum walks based on* $P$ *and* $Q$*, respectively. Let* $M$ *be the set of marked elements in the state space. Denote* $QHT(P)$ *as the hitting time of walk* $W(P)$ *and* $QHT(Q)$ *as the hitting time of walk* $W(Q)$*. Suppose* $|M| = \epsilon N$*. If the*

*second largest eigenvalues of $P$ and $Q$ are at most $1-\delta$ and $1-\Delta$, respectively, then in general*

$$QHT(P) = O\Big(\sqrt{\frac{1}{\delta\epsilon}}\Big), \qquad QHT(Q) = O\Big(\sqrt{\frac{1}{(\delta - \|E\|)\epsilon}}\Big) \tag{8.12}$$

*where $\delta - \|E\| \leq \Delta \leq \delta + \|E\|$.*

*Proof.* Suppose the Markov chain $P$, $Q$ and matrix $E$ are in the following block structure

$$P = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}, \quad Q = \begin{pmatrix} Q_1 & Q_2 \\ Q_3 & Q_4 \end{pmatrix}, \quad E = \begin{pmatrix} E_1 & E_2 \\ E_3 & E_4 \end{pmatrix} \tag{8.13}$$

where we order the elements such that the marked ones come last, i.e., $P_4$, $Q_4$ and $E_4 \in \mathbb{C}_{|M|\times|M|}$. The corresponding modified Markov chains [SZE04] would be

$$\tilde{Q} = \begin{pmatrix} Q_1 & 0 \\ Q_3 & I \end{pmatrix} = \begin{pmatrix} P_1 + E_1 & 0 \\ P_3 + E_3 & I \end{pmatrix}. \tag{8.14}$$

By [SZE04], we have $QHT(P) = O(\sqrt{\frac{1}{1-\|P_1\|}})$ and $QHT(Q) = O(\sqrt{\frac{1}{1-\|Q_1\|}})$. Since we know

$$\|P_1\| \leq 1 - \frac{\delta\epsilon}{2} \quad \text{and} \quad \|Q_1\| \leq 1 - \frac{\Delta\epsilon}{2} \tag{8.15}$$

by [SZE04] and by Cauchy's interlacing theorem we have $\|E\| \geq \|E_1\|$ [BHA97, Cor.III.1.5], we then obtain

$$\|Q_1\| \leq \min\left\{\|P_1\| + \|E\|, 1 - \frac{(\delta - \|E\|)\epsilon}{2}\right\} \tag{8.16}$$

as $\delta - \|E\| \leq \Delta \leq \delta + \|E\|$. Therefore, the hitting times for $P$ and $Q$ are derived.

From the corollary above, it is clear that noise increases the quantum hitting time. By a simple comparison with the classical hitting time, we have the following fact.

**Fact 6.** *When given a perturbed quantum walk $W(Q)$, where the magnitude of noise is $\|E\|$, the quadratic speed-up gained from the quantum walk will be annihilated when $\|E\| \geq \Omega(\delta(1-\delta\epsilon))$.*

*Proof.* Given $\|E\| = \delta(1-\delta\epsilon)$, then by *collorary 2* we have $QHT(Q) = O(\sqrt{\frac{1}{\delta(1-(1-\delta\epsilon))\epsilon}}) = O(\frac{1}{\delta\epsilon})$.

### 8.2.4 Quantum Hitting Time Based on MNRS Algorithm

Let $U = U_2U_1$ be a unitary matrix with real entries. Let $|\mu\rangle$ (see Fact 2) be the marked element where $U_1 = I - 2|\mu\rangle\langle\mu|$ and $U_2$ is a real unitary matrix with a unique 1-eigenvalue $|\phi\rangle$. Similar to the classical case, let $|\phi\rangle_{-\mu} = |\phi\rangle - \langle\phi|\mu\rangle|\mu\rangle$.

The potential eigenvalues for $U$ are then $\pm 1$ and conjugate complex numbers $(e^{i\alpha_j}, e^{-i\alpha_j})$. Let $|\phi\rangle_{-\mu}$ be the input state for the phase estimation of $U$, then $|\phi\rangle_{-\mu}$ can be uniquely decomposed in the eigenbasis of $U$ as

$$|\phi\rangle_{-\mu} = \delta_0|\omega_0\rangle + \sum_j \delta_j|\omega_j^{\pm}\rangle + \delta_{-1}|\omega_{-1}\rangle \tag{8.17}$$

where $U|\omega_0\rangle = |\omega_0\rangle$, $U|\omega_{-1}\rangle = -|\omega_{-1}\rangle$ and $U|\omega_j\rangle = e^{\pm i\alpha_j}|\omega_j\rangle$. Let $QH$ be the random variable which takes the value $1/\alpha_j$ with probability $\delta_j^2$ and the value $1/\pi$ with probability $\delta_{-1}^2$.

**Definition 18.** *[MNRS09] The quantum $|\mu\rangle$-hitting time of $U_2$ is the expectation of QH, that is*

$$QHT(U_2, |\mu\rangle) = 2\sum_i \frac{\delta_j^2}{\alpha_j} + \frac{\delta_{-1}^2}{\pi}. \tag{8.18}$$

Hence, in order to compute the quantum hitting time of $U_2$, it is important to compute the spectral decomposition of $U$. It is shown in the following theorem.

**Theorem 16.** *[SZE04] Fix an $n \times n$ column-wise stochastic matrix $\tilde{P}$ [1], and let $\{|\lambda\rangle\}$ denote a complete set of orthonormal eigenvectors of the $n \times n$ matrix $D$ with entries $D_{jk} = \sqrt{\tilde{P}_{jk}\tilde{P}_{kj}}$ with eigenvalue $\{\lambda\}$. Then the eigenvalues of the discrete-time quantum walk $U = S(2\Pi_{\mathcal{A}} - I)$ corresponding to $\tilde{P}$ are $\pm 1$ and $\lambda \pm i\sqrt{1-\lambda^2} = e^{\pm i arccos\lambda}$ [2].*

Let the subset $M$ be the set of marked elements that we are searching for. The discrete-time quantum walk $U_{-\{M\}} = S(2\Pi_{\mathcal{A}_{-\{M\}}} - I)$ satisfies the above theorem when we modify the original transition matrix $P$ into $\tilde{P}$ in the following manner:

$$\tilde{P}_{jk} = \begin{cases} 1 & k \in M \text{ and } j = k \\ 0 & k \in M \text{ and } j \neq k \\ P_{jk} & k \notin M \end{cases}$$

[1] *$\tilde{P}$ is the modified stochastic matrix of $P$ defined in eq. 8.19*

[2] *Eigenvalues of $\tilde{D}$ in eq. 8.20 are exactly the eigenvalues of $\tilde{P}_{-\{M\}}$ and eigenvalue 1*

We can view $\tilde{P}$ in block structure as follows:

$$P = \begin{pmatrix} P_{-\{M\}} & P_2 \\ P_3 & P_4 \end{pmatrix} \longrightarrow \quad \tilde{P} = \begin{pmatrix} P_{-\{M\}} & 0 \\ P_3 & I \end{pmatrix}, \tag{8.19}$$

then the corresponding discriminant matrix $\tilde{D}$ is

$$\tilde{D} = \begin{pmatrix} P_{-\{M\}} & 0 \\ 0 & I \end{pmatrix}. \tag{8.20}$$

**Fact 7.** *Now let us set $M = \{x\}$. Then $\pm 1$ and $e^{\pm i\alpha_j}$ are eigenvalues of $U_{-x}$ where $\lambda_j$ are the eigenvalues of $P_{-x}$. Since $\lambda_j = \cos\theta_j$ (see sect. 8.2.1), and by use of* theorem 16*, we know that $\theta_j = \alpha_j$.*

Furthermore, by *Fact 2* we know the unitary $W(P,x) = U_{-x}^2$. The eigenvectors of $U_{-x}$ remain the eigenvectors of $W(P,x)$ but the eigenvalues of $W(P,x)$ would be $e^{2i\alpha_j}$. Given $|\phi\rangle_{-\mu}$ as the input state, we run phase estimation of $W(P,x)$ and the corresponding quantum hitting time would be

$$QHT(P,x) = 2\sum_{j=1}^{n-1} \frac{\delta_j^2}{2\alpha_j} = \sum_{j=1}^{n-1} \frac{\delta_j^2}{\theta_j}, \tag{8.21}$$

the term $\frac{\delta_{-1}^2}{\pi}$ in *def. 18* disappears because the corresponding eigenphase becomes 0.

### 8.2.5 Delayed Perturbed Quantum Hitting Time

In this subsection, we define the Delayed Perturbed Quantum Hitting Time (DPQHT) and its upper bound as the following.

**Fact 8.** *[MNRS09] When $P$ is an ergodic Markov transition with positive eigenvalues, then the $x$-quantum hitting time for the unitary $W(P,x)$ is*

$$QHT(P,x) = \sum_{j=1}^{n-1} \frac{\nu_j^2}{\theta_j} \tag{8.22}$$

*Proof.* Since the length of the projection of $|\phi\rangle_{-u}$ to the eigenspace corresponding to $\alpha_j$ is $\nu_j^2$ [MNRS09], then by eq. 8.21 we have the result as shown in eq. 8.22.

**Lemma 12.** *Given $QHT(P,x)$ and $QHT(Q,x)$ with $\|P-Q\| = \|E\|$, denote the Delayed Perturbed Quantum Hitting Time $DPQHT(P,Q,x)$ that*

$$DPQHT(P,Q,x) = QHT(Q,x) - QHT(P,x).$$

*By use of Fact 8, we have $DPQHT(P,Q,x)$ bounded from above by*

$$\frac{1}{\sqrt{1-\lambda_1-\|E\|_2}} - \frac{1}{2\sqrt{1-\lambda_1+\gamma}}.$$

*The eigenvalues of $P_{-x}$ are ordered such that $1 > \lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_{n-1} > 0$ and $\lambda_1 - \lambda_{n-1} = \gamma$.*

*Proof.* Based on Fact 8, we have

$$\begin{aligned} DPQHT(P,Q,x) &= QHT(Q,x) - QHT(P,x) \\ &= \sum_{i\in\Omega} \Big(\frac{\tilde{\nu}_i^2}{\tilde{\theta}_i} - \frac{\nu_i^2}{\theta_i}\Big) \\ &\leq \Big(\sum_{i\in\Omega} \frac{\tilde{\nu}_i^2}{\cos^{-1}\tilde{\lambda}_1}\Big) - \Big(\sum_{i\in\Omega} \frac{\nu_i^2}{\cos^{-1}\lambda_{n-1}}\Big) \\ &\leq \frac{1}{\sqrt{1-\lambda_1-\|E\|}} - \frac{1}{2\sqrt{1-\lambda_1+\gamma}}. \end{aligned} \tag{8.23}$$

The last inequality is a simple result from *Fact 5* and the fact that $2\sqrt{1-\lambda} > \cos^{-1}\lambda > \sqrt{1-\lambda}$ for all $\lambda \in (0,1)$.

## 8.3 Sample Perturbation

In this section we adapt the results from the work [CM01] to bound the stationary distribution $\pi(Q)$ of a perturbed matrix $Q$ with respect to the perturbation $E$ and the true stationary distribution $\pi(P)$, i.e.,

$$Q \cdot \pi(Q) = \pi(Q), \quad P \cdot \pi(P) = \pi(P). \tag{8.24}$$

Let $\Omega$ be the state space and $\Omega' = \Omega \cup \{0\}$. The *total variation distance* between two probability distributions over $\Omega$ is defined as

$$D(\pi(P), \pi(Q)) = \frac{1}{2}\sum_{x\in\Omega} \|\pi(P)_x - \pi(Q)_x\|_1 = \max_{S\subseteq\Omega'} |\pi(P)_S - \pi(Q)_S|. \tag{8.25}$$

Here $\pi(P)$ denotes the stationary distribution of matrix $P$, $\pi(P)_x$ is the $x$th element of $\pi(P)$ and $\pi(P)_S$ denotes the sum of $\pi(P)_x$ where $x \in S$, i.e., $\sum_{x\in S}\pi(P)_x = \pi(P)_S$.

In [CM01] it is assumed that the transition matrix is *row-wise stochastic*. Our matrix is column-wise stochastic (see Eq. 8.24) but since it is symmetric, it is also row-stochastic. By choosing *condition number* $\kappa_5$ in [CM01], the *ergodicity coefficient*, using the $l_p$ norm, is defined as

$$\tau_p(P) = \sup_{\|v\|_p=1, v^T e=0} \|v^T P\|_p \tag{8.26}$$

where $e$ is a column vector of all ones. Since $P$ is a stochastic matrix, the ergodic coefficient satisfies $0 \leq \tau_1(P) \leq 1$. In case of $\tau_1(P) < 1$, we have a perturbation bound in terms of the ergodic coefficient of $P$:

$$D(\pi(P), \pi(Q)) = \frac{1}{2}\|\pi(P) - \pi(Q)\|_1 \leq \frac{1}{2(1 - (\tau_1(P)))}\|E\|_\infty. \tag{8.27}$$

While there are several methods that make use of Szegedy's quantum walk operators to prepare quantum samples [WA08, SBB07, SBBK08], we choose [WA08] as the main approach to analyze as it leads to an overall speed-up in the general case. The other approaches [SBB07, SBBK08] take advantage of the quantum Zeno effect but the problem is that the quantum Zeno effect would result in an exponential slow-down in the general case.

The work by Wocjan and Abeyesinghe [WA08] showed an approach to prepare the coherent stationary distribution of a Markov Chain via a modified quantum walk and Grover's $\frac{\pi}{3}$-amplitude amplification techniques. The theorem listed below is the main theorem in *Speed-up via Quantum Sampling*. We refer the interested readers to [WA08] for details on this algorithm for the construction techniques and the computational complexity.

**Theorem 17** (Speed-up via quantum sampling [WA08])**.** *Let $Q_0, Q_1, \ldots, Q_r = Q$ be a sequence of classical Markov chains with stationary distributions $\pi_0, \pi_1, \ldots, \pi_r$ and spectral gap $\delta_0, \ldots, \delta_r$. Assume that the stationary distributions of adjacent Markov chains are close to each other in the sense that $|\langle \pi_i | \pi_{i+1} \rangle|^2 \geq c$ where $c$ is some constant, for $i = 0, \ldots, r-1$.. Then for any $\eta > 0$, there is an efficient quantum sampling algorithm, making it possible to*

*sample according to a probability distribution $\tilde{\pi}_r$ that is close to $\pi_r$ with respect to the total variation distance, i.e., $D(\tilde{\pi}_r, \pi_r) \leq \eta$.*

Based on the theorem above, we can immediately conclude the following corollary:

**Corollary 3.** *When the coherent quantum sample based on the perturbed Markov chain is prepared by using techniques of [WA08] with precision $\eta$, the total variation distance between the prepared quantum sample $\tilde{\pi}(Q)$ and the true quantum sample $\pi(P)$ is less than $\eta + \frac{1}{2(1-(\tau_1 P))}\|E\|_\infty$.*

*Proof.* By *Theorem 17* we can efficiently construct a quantum sample $\tilde{\pi}(Q)_r$ that is $\eta$ close to $\pi(Q)$. Then by triangle inequality we obtain

$$D(\pi(P), \tilde{\pi}(Q)) \leq D(\pi(P), \pi(Q)) + D(\pi(Q) + \tilde{\pi}(Q)) \leq \frac{1}{2(1-(\tau_1 P))}\|E\|_\infty + \eta. \quad (8.28)$$

## 8.4 Discussion

By quantizing a perturbed symmetric stochastic $n \times n$ matrix $Q$ with noise $E$, we find an upper bound for the perturbed quantum hitting time. We also show, in *fact 6*, the lower bound for the magnitude of noise when the quadratic speed-up gained from the quantum walk will be annihilated by the noise.

Furthermore we compute the upper bound for the delayed perturbed quantum hitting time based on the definition of quantum hitting time. One cannot just directly apply

the square root speed-up from quantum walks to the delayed perturbed hitting time (see eq. 8.11). If one does so, one would obtain an upper bound for $DPQHT$ as

$$\frac{1}{\sqrt{1-\lambda_1-\|E\|}}-\frac{1}{\sqrt{1-\lambda_1+\gamma}}. \tag{8.29}$$

It would be incorrect. The second term of eq. 8.29 should the the minimum of $\sum_{i\in\Omega}\frac{\nu_i^2}{\cos^{-1}\lambda_{n-1}}$. But in eq. 8.29, the second term was actually the maximum. Thus, it is clear that the upper bound for DPQHT is actually greater than the difference between the square root of the upper bound for a perturbed random walk and the square root of the lower bound for a random walk.

In the meanwhile, we also showed that how the quantum sample prepared by using the approach in [WA08] would fluctuate from the true quantum sample when perturbation is present. The analysis is based on the assumption that we have a series of Markov chains $Q1,\ldots,Q_r=Q$. Hence, we have

$$D(\pi(P),\tilde{\pi}(Q))\leq\frac{1}{2(1-(\tau_1 P))}\|E\|_\infty+\eta.$$

Intuitively from the analysis we can see that the total variation distance for $D(\pi(\tilde{Q}),\pi(Q))$ is simply *additive* and $D(\pi(P),\pi(Q))$ cannot be eliminated. However, if the matrix $P=Q_i$ is inside the sequence $Q_1,\ldots,Q_r$ where $1<i<r$, can we invent a procedure to detect to avoid such overshoot? Future study is to find the relation between quantum mixing time, the time it takes to get $\eta$-close to the true stationary distribution, and the quantum hitting time. Furthermore, another possible analysis approach would be to assume that we have a

series of Markov chains $P_1, \ldots, P_r = P$ (without the noise). We can adapt the analysis in [WA08] to study how the noise would affect (i) accuracy when blindly preparing the quantum sample without acknowledging the existence of noise or (ii) complexity when the noise is acknowledged and desired accuracy must be achieved.

# CHAPTER 9
# CONCLUSION

As shown in previous chapters, quantum walk is a useful technique that can be used to solve various problems [AMB04, CCDFGS03, CSV07, MSS05, WCNA09, KMOR10]. For some oracular problems, quantum walk renders exponential speedups over its classical counterparts [CCDFGS03, CSV07]. For some other problems, such as NAND tree evaluation problem [FGG08] and the triangle finding problem [MSS05], quantum walk renders polynomial speedup over classical algorithms. In this chapter, we will discuss possible problems for future study. Some of the problems have been tried quantumly, but there might be space left for improvement. Some of the problems have been tried out via classical random walk, but there are no quantum version of them yet. It would be of interest to researchers to investigate the possible speed-up (exponential, polynomial or none) that quantum walk can provide for those problems.

## 9.1 Graph Problems

**Problem.** ***Complexity of graph problems****: Several graph problems, such as triangle finding [MSS05] and matching finding in a bipartite graph, remain open in the query model in terms of their complexity. By Turán's theorem [Turan41], the n-vertex triangle-free graph with the maximum number of edges is a complete bipartite graph that the numbers of vertices on each side of the bipartition are as equal as possible. Hence, could there an improvement*

*on the query complexity $O(N^{1.3})$ of the triangle algorithm [MSS05] to determine if a graph is triangle-free? Furthermore, would it be possible to have a quantum algorithm that uses only $N^2$ queries for finding a matching in a bipartite graph $G$ with $N$ vertices on each side, specified by $N^2$ variables?*

## 9.2 The Ising Model

As classical random walk also deals with many problems related to the *Ising model*, we would also like to explore the possible advantage of applying quantum walk to those problems. But before we go further, let us review this model.

The Ising model on a graph with vertex set $V$ at inverse temperature $\beta$ (see section 6.1). A spin system is a probability distribution on $\Omega = \{-1, 1\}^V$, where $V$ is the vertex set of a graph $G = (V, E)$. Let $v$ be an arbitrary node in graph $G$, i.e. $v \in V$. The value $\sigma(v)$ is the spin at $v$, which can be 1 or -1. $\sigma$ is the configuration that specifies the values of the nodes. The nearest-neighbor **Ising model** defines the **energy** of a configuration that

$$E(\sigma) = -\sum_{v,w \in V, v \text{sim} w} \sigma(v)\sigma(w). \tag{9.1}$$

The probability distribution at inverse temperature $\beta$ is thus

$$\pi(\sigma) = \frac{e^{-\beta E(\sigma)}}{Z(\beta)}. \tag{9.2}$$

**Problem.** ***Positive boundary conditions****[LPW09]: Given the Ising model on the $n \times n$ grid with the boundary forced to have all positive spins, it remains open for classical algo-*

*rithms to show the mixing time is at most polynomial in $n$ at any temperature. Classically, an upper bound on the relaxation time of $e^{n^{\frac{1}{2+\epsilon}}}$ was obtained by Martinelli. Could we devise a corresponding quantum walk that has at least a polynomial speedup in terms of the relaxation time?*

**Problem.** ***Ising on transitive graphs****[LPW09]: For the Ising model on transitive graphs, it remains open to show the relaxation time is of order $n$ if and only if the mixing time is of order $n \log n$. It is known to be true for the two-dimensional torus. Will the quantum algorithm hold a similar relation to the classical one, i.e. $n$ and $n \log n$, or will the quantum speedup break the relation?*

## 9.3 Black-box Hamiltonian Simulation and Unitary Implementation

Simulation of Hamiltonian dynamics is one major application of quantum computation. By the 2nd Postulate [NC00] of quantum mechanics, we know that the evolution of the state of a closed quantum system is described by the *Shrödinger equation*

$$i\hbar \frac{d|\psi\rangle}{dt} = H|\psi\rangle \tag{9.3}$$

where $\hbar$ is the *Planck's constant* and $H$ is a fixed Hermitian operator known as *Hamiltonian* of the closed quantum system. Hence, Hamiltonian simulation is the basis for simulating quantum systems.

A general approach [BC11] was considered to simulate Hamiltonian by using quantum walks. The complexity of their approach scales linearly with respect to the sparseness $D$ and the evolution time $t$ while other methods scales as $D^4$ and superlinearly in $t$. The corresponding $N \times N$ unitary $U$, based on the Hamiltonian $H$, thus can be implemented by using $O(N^{\frac{2}{3}}(\log\log N)^{\frac{3}{4}})$ queries to the matrix elements, instead of $N^2$.

Could there be an improvement on the query complexity of the unitary $U$ implementation? Furthermore, can we simplify this proposed Hamiltonian simulation algorithm to reach a tighter upper bound for the query complexity?

## 9.4 Perturbation Theory

**Problem.** ***Perturbed quantization****: In our research, we discussed the effect of a given perturbed classical Markov transition matrix on the quantum hitting time. This is under the assumption that the quantization process itself is noise-free. In practice, it is more likely that noise will also occur during the quantization process. If we have a higher degree of freedom such that the quantization is allowed to have error $E_q$, what would the effect on the quantum hitting time be? Will the DPQHT (delayed perturbed quantum hitting time) be a simple additive result such that*

$$DPQHT = \frac{1}{\sqrt{1-\lambda_1-\|E\|_2-\|E_q\|_2}} - \frac{1}{2\sqrt{1-\lambda_1+\gamma}} \tag{9.4}$$

*(see lemma 12 at section 8.2.5 for details)? Or will the DPQHT have a much longer delayed time?*

# APPENDIX A
# QUANTUM WALK UPDATE

## A.1 Quantum Walks from Classical Markov Chains

The class of classical approximation schemes that we speed up uses reversible, ergodic Markov chains $P_i$ with stationary distributions $\pi_i$. Here we briefly review the quantum analogue of a Markov chain, describing the *quantum walk operator* $W$ corresponding to the classical Markov Chain $P$.

In each step of a Markov chain $P$ with state space $\Omega$, the probability of a state $x$ to transition to another state $y$ is given by the element $p_{xy}$ of the $D \times D$ transition matrix, where $D = |\Omega|$. Following Szegedy [SZE04], for each such Markov Chain, we can define its quantum analogue. The Hilbert space on which this quantum operation acts is $\mathbb{C}^D \otimes \mathbb{C}^D$, with two $\mathbb{C}^D$ registers. We start by defining the states

$$|p_x\rangle = \sum_{y \in \Omega} \sqrt{p_{xy}} \, |y\rangle \, . \tag{A.1}$$

These states can be generated by a *quantum update* – any unitary $U$ that satisfies

$$U \, |x\rangle \, |0\rangle = |x\rangle \, |p_x\rangle \tag{A.2}$$

for some fixed state $0 \in \Omega$ and all $x \in \Omega$. The quantum analogue of a Markov chain is then defined as follows.

**Definition 19** (Quantum Walk)**.** *A quantum walk $W(P)$ based on a classical reversible Markov chain $P$ is a unitary operation acting on the space $\mathbb{C}^D \otimes \mathbb{C}^D$ as*

$$W(P) = R_{\mathcal{B}} \cdot R_{\mathcal{A}}, \tag{A.3}$$

*where $R_{\mathcal{B}}$ and $R_{\mathcal{A}}$ are reflections about the spaces*

$$\mathcal{A} = \text{span}\{|x\rangle |0\rangle : x \in \Omega\}, \tag{A.4}$$

$$\mathcal{B} = U^{\dagger} S U \mathcal{A}, \tag{A.5}$$

*and $S$ is a swap of the two registers.*

This particular definition of the quantum walk is suitable for making some of the proofs in [WA08] easier. It is equivalent to the standard definition of Szegedy [SZE04] up to conjugation by $U$. Therefore, the spectral properties of our $W$ and Szegedy's quantum walk are the same.

Let $\delta$ be the spectral gap of the classical Markov chain $P$. Let us write its eigenvalues as $\mu_0 = 1$ and $\mu_j = \cos(\theta_j)$, for $j = 1, \ldots, D-1$ and $\theta_j \in \left(0, \frac{\pi}{2}\right)$. According to Szegedy [SZE04], on the space $\mathcal{A}+\mathcal{B}$, the eigenvalues of the quantum walk $W(P)$ with nonzero imaginary part are $e^{\pm 2i\theta_j}$. The phase gap of the quantum walk $W(P)$ is then defined as $\Delta = 2\theta_1$ (with $\theta_1$ the smallest of $\theta_j$). When the Markov chain is ergodic and reversible, Szegedy proved that

$$\Delta \geq 2\sqrt{\delta}, \tag{A.6}$$

a quadratic relation between the phase gap $\Delta$ of the quantum walk $W(P)$ and the spectral gap $\delta$ of the classical Markov chain $P$. This quadratic relation is behind the speed-up of many of today's quantum walk algorithms.

# APPENDIX B
# IMPLEMENTATION OF QUANTUM WALK

## B.1 Additional Details for the Efficient Quantum Update Circuit

In this Appendix, we spell out additional details for our Quantum Update circuit as well as draw the circuit out for a $d = 4$.

The state space of the classical Markov chain $P$ is $\mathcal{E}$, with $|\mathcal{E}| = 2^m$. The entries of $P$ are $p_{xy}$, the transition probabilities from state $x$ to state $y$. We assume that $P$ is sparse, i.e. that for each $x \in \mathcal{E}$ there are at most $d$ neighbors $y_i^x$ such that $p_{xy_i^x} > 0$, and their number is small, i.e. $d \ll 2^m$. Since $d$ is a constant, we can assume without loss of generality that $d = 2^r$. We want to implement the quantum (5.3), where $|x\rangle \in \mathbb{C}^{2m}$.

## B.2 Preparation

Classically, our knowledge of $P$ can be encoded into efficient reversible circuits outputting the neighbors and transition probabilities for the point $x$. We will use quantum versions $N$ and $T$ of these circuits, with the following properties. The neighbor circuit $N$ acts on $d$ copies of $\mathbb{C}^{|\mathcal{E}|}$ and produces a list of neighbors of $x$ as

$$N \left|x\right\rangle_L \left|0\right\rangle^{\otimes d} = \left|x\right\rangle_L \otimes \left|y_0^x\right\rangle \cdots \left|y_{d-1}^x\right\rangle . \tag{B.1}$$

All the transition probabilities $p_{xy_i^x}$ are given with $t$-bit precision. The transition probability circuit $T$ acts on a register holding a state $|x\rangle$ and $d$ copies of $(\mathbb{C}^2)^{\otimes t}$, producing a list of

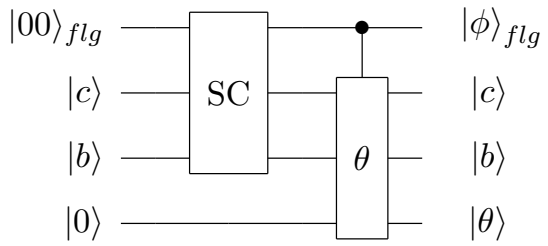


Figure B.1: The Determine Angle Circuit $DAC$

transition probabilities for neighbors of $|x\rangle$ as

$$T\,|x\rangle_L\,|0\rangle^{\otimes d} = |x\rangle_L \otimes |p_{xy_0^x}\rangle \cdots |p_{xy_{d-1}^x}\rangle. \tag{B.2}$$

To simplify the notation, let us label $q_i = p_{xy_i^x}$. We now prepare all the terms $q_i^{(k)}$, filling the tree in Figure (5.1). Starting from $q_i^{(\log d)} = q_i$, we use an adding circuit ($ADD$) doing the operation $q_i^{(k-1)} = q_{2i}^{(k)} + q_{2i+1}^{(k)}$. The probability distribution $\{q_i\}$ is efficiently integrable, so filling the tree of $q_i^{(k)}$ is easy, and we can use Grover and Rudolph's method [GR02] of preparing quantum samples for such probability distributions.

## B.3 Determining the Rotation Angles

After the preparation described in the previous Section, we need to compute the appropriate rotation angles $\tilde{\theta}_i^{(k)}$ for Grover and Rudolph's method. For this, we use the Determine Angle

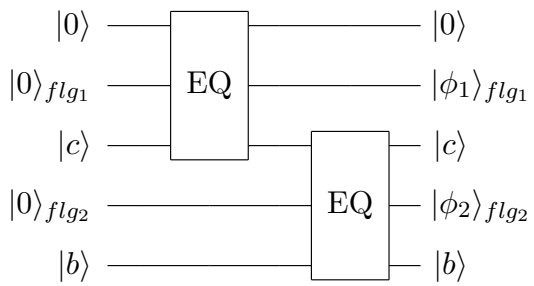


Figure B.2: The Circuit $SC$ Handling Special Cases

Circuit ($DAC$). This circuit produces

$$\theta_i^{(k)} = \cos^{-1}\sqrt{\frac{q_{2i}^{(k)}}{q_i^{(k-1)}}}, \tag{B.3}$$

while also handling the special cases $q_{2i}^{(k)} = q_i^{(k-1)}$ and $q_i^{(k-1)} = 0$. For simplicity, let us label $b = q_i^{(k-1)}$, $c = q_{2i}^{(k)}$. The $DAC$ circuit first checks the special cases, and then, conditioned on the state of the two two flag qubits, computes (B.3). We draw it in Figure B.1, with the special case-analysing circuit SC given in Figure B.2. Here $EQ$ is a subroutine testing whether two qubits (in computational basis states) are the same. The first $EQ$ tests the states $|0\rangle$ and $|c\rangle$, while the second $EQ$ runs the test on $|c\rangle$ and $|b\rangle$. We have the following four scenarios depending on the flag qubits

$$\begin{array}{ll} 00 & \text{the circuit } \theta \text{ computes normally ,} \\ 01, 11 & \text{the circuit } \theta \text{ does nothing (keeps angle} = 0, \text{ as } b = c) \text{ ,} \\ 10 & \text{the circuit } \theta \text{ outputs } \theta = \pi/2, \text{ as } c = 0. \end{array} \tag{B.4}$$

The third option corresponds to $c = 0$, when all the probability in the next layer of the tree is concentrated in the right branch. We then simply flip the superposition qubit, using $\theta = \frac{\pi}{2}$.

## B.4 Creating Superpositions and Mapping

After the angle is determined, we apply the corresponding rotation to the appropriate qubit in the superposition register $S$, as described in Section 5.4. We then uncompute the rotation angle.

Once the final superposition is created in $S$, we invoke a mapping circuit $M$. This $M$ acts on the register holding the names of the $d$ neighbors of $x$, the superposition register, and the output register $R$. It takes $y_j^x$, the name of the $j$-th neighbor of $x$, and puts it into the output register as

$$M|0\rangle_R \otimes |y_0^x\rangle \otimes \ldots \otimes |y_{d-1}^x\rangle \otimes |j\rangle_S = |y_j^x\rangle_R \otimes |y_0^x\rangle \otimes \ldots \otimes |y_{d-1}^x\rangle \otimes |j\rangle_S. \tag{B.5}$$

We can do this, because the names of the states in $\mathcal{E}$ are given as computational basis states. The next step is to uncompute the label $j$ in the last register with a cleaning circuit $C$ as

$$C|y_i^x\rangle_R \otimes |y_0^x\rangle \otimes \ldots \otimes |y_{d-1}^x\rangle \otimes |j\rangle = \begin{cases} |y_i^x\rangle_R \otimes |y_0^x\rangle \otimes \ldots \otimes |y_{d-1}^x\rangle \otimes |j\rangle & \text{if } i \neq j \\ |y_i^x\rangle_R \otimes |y_0^x\rangle \otimes \ldots \otimes |y_{d-1}^x\rangle \otimes |0\rangle & \text{if } i = j. \end{cases} \tag{B.6}$$

These two steps transferred the superposition from the register $S$ (with $r = \log d$ qubits), into the output register $R$ (which has $m$ qubits). The final step of our procedure is to uncompute (clean up) the lists of neighbors and transition probabilities.

Table B.1: Required Numbers of Qubits

| Register Type | Required number of qubits |
|---|---|
| $x$ (register $L$) | $m$ |
| $y$ (register $R$) | $m$ |
| $y_i^x$ (neighbor list) | $d \times m$ |
| $q_i$'s (probabilities) | $(2d-2) \times t$ |
| flag qubits | 2 |
| $\theta$ (rotation angle) | $n = \frac{3t}{2} + \Omega(1)$ |
| ancillae for computing $\theta$ | $a_\theta = \text{poly}(n) = \text{poly}(t)$ |
| superposition register $S$ | $r = \log d$ |

## B.5 The Required Resources

Let us count the number of qubits and operations required for our quantum update rule $U$ based on a $d$-sparse stochastic transition matrix $P$. The number of ancillae required is $\Omega(dm + dt)$, where $2^m$ is the size of the state space and $t$ is the required precision of the transition probabilities. Moreover, the required number of operations scales like $\Omega(d\, r\, m\, a_\theta)$, where $r = \log d$ and $a_\theta$ is the number of operations required to compute the angle $\theta$ with $n = \Omega(t)$-bit precision. Finally, when we have $t$-bit precision of the final amplitudes, the precision of the unitary we applied is

$$\left\| \left( U - \tilde{U} \right) |x\rangle \otimes |0\rangle \right\| \leq \epsilon, \tag{B.7}$$

for any $x \in \mathcal{E}$ when $t = \Omega\left(\log d + \log \frac{1}{\epsilon}\right)$. The total number of operations in our circuit thus scales like

$$\Omega\left(md\,\mathrm{poly}\left(\log d\right) + md\left(\log d\right)\mathrm{poly}\left(\log\frac{1}{\epsilon}\right)\right). \tag{B.8}$$

Besides the registers for the input $|x\rangle_L$ and output $|0\rangle_R$, we need $d$ registers (with $m$ qubits) to hold the names of the neighbors of $x$, and $2d-2$ registers (with $t$ qubits) to store the transition probabilities $q_i$. The $DAC$ circuit requires two extra flag qubits and a register with $n = \frac{3t}{2} + \Omega(1)$ qubits to store the angle $\theta$. Computing the angle $\theta$ requires a circuit with $poly(n)$ qubits. Finally, the superposition register $S$ holds $r$ qubits. These requirements are summed in Table B.1.

To conclude, we draw out the superposition-creating part of the quantum update for $d = 4$ in Figure B.3. The first two lines represent the superposition register $S$, in which we prepare

$$|\varphi\rangle \;=\; \sqrt{q_0^{(2)}}|00\rangle + \sqrt{q_1^{(2)}}|01\rangle + \sqrt{q_2^{(2)}}|10\rangle + \sqrt{q_3^{(2)}}|11\rangle = \sum_{i=0}^{3}\sqrt{q_i}\,|i\rangle\,. \tag{B.9}$$

## B.6 Approximating the Permanent

In this Appendix we present a particular example of a quantum algorithm with a polynomial speedup over its classical counterpart, requiring our efficient approach to implementing quantum walks. The example is a rather naïve quantization of the classical algorithm for approximating the permanent of a matrix

$$per(A) = \sum_{\sigma}\prod_{i=1}^{n} a_{i,\sigma(i)}, \tag{B.10}$$

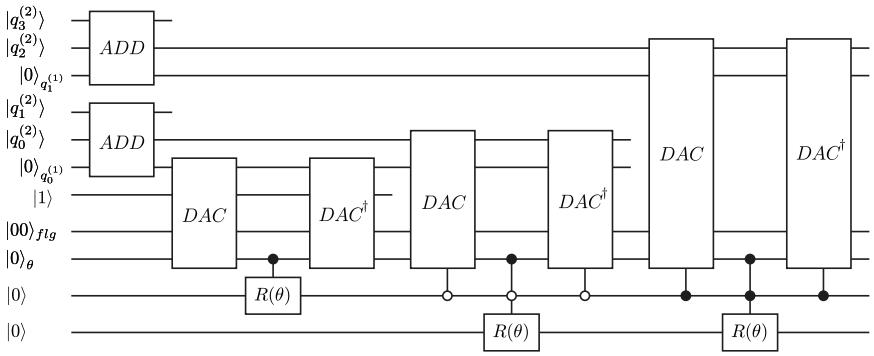


Figure B.3: Creating the Superposition for $d = 4$

where $\sigma$ runs all over the permutations of $[1, \ldots, n]$. For a 0/1 matrix $A$, the permanent of $A$ is exactly the number of perfect matchings in the bipartite graph with bipartite adjacency matrix $A$. A classical FPRAS (fully polynomial randomized approximation scheme) [BSVV08] for this task involves taking $O^*(n^7)$ steps of a Markov chain (here $O^*$ means up to logarithmic factors). It produces an approximation to the permanent within $[(1-\eta)\,per(A), (1+\eta)\,per(A)]$ by using

1. $\ell = O^*(n)$ stages of simulated annealing,
2. at each stage, generating $S = O^*(n^2)$ samples from a particular Markov chain,
3. $T = O^*(n^4)$ Markov chain invocations to generate a sample from its approximate steady state.

The failure probability of each stage is set to $\hat{\eta} = o(1/m^4)$ so that $\eta = \ell\hat{\eta}$ is small. Hence, the total complexity (number of Markov chain steps used) is $\ell ST = O^*(n^7)$.

The sparsity parameter $d$ of the Markov chains involved scales with the problem size $m$. Therefore, the dependence of the implementation of the corresponding quantum walk on $d$ becomes significant. Furthermore, because of the many stages of simulated annealing and sampling, the error $\epsilon$ in implementation of each quantum walk operator needs to smaller than one over the number of quantum walk steps involved.

The simplest quantized algorithm uses a quantum walk instead of the Markov Chain, and requires $O^*(n^5)$ steps of a quantum walk, as the mixing of the quantum walk requires only $\sqrt{T} = O^*(n^2)$ steps. However, it is important to choose an efficient circuit to implement each step of the quantum walk. A bad choice could destroy the speedup.

Let us compare what happens when this algorithms utilizes the different methods for quantum walk implementation as subroutines, counting the number of required elementary gates. Note that in this counting, all of the methods (classical and quantum) we will mention share a common factor $m$ (the log of the state space size). However, the scaling in $d$ (the sparsity parameter) and $\frac{1}{\epsilon}$ (precision) is what distinguishes them.

Let us look at the alternative approaches given in Section 5.2, and show that the small $n^2$ polynomial speedup is lost. The first two of these approaches scale with $\frac{1}{\epsilon}$. This brings an extra $\frac{1}{\epsilon} \propto \sqrt{T} \propto n^2$ factor to the complexity of the algorithm, destroying the speedup. The third alternative uses $O^*(d^2)$ elementary gates, adding an extra factor of $d^2 = n^2$, again destroying the speedup. On the other hand, our method uses only $O^*(d) = n$ gates (the scaling coming from precision requirements only adds logarithmic factors), and we thus retain some of the quantum advantage.

This example was just an illustration of a scenario, where our efficient implementation of a quantum walk (see Section 5.3) is necessary. However, we see its use in a future much better quantum algorithm for approximating the permanent, using not only quantum walks, but also quantizing the sampling/counting subroutine as in [WCNA09].

# APPENDIX C
# GLOSSARY OF NOTATIONS

# Nomenclature

| | |
|---|---|
| $\alpha_0, \alpha_1, \ldots$ | Complex numbers: $\alpha_j = \text{Real}(\alpha_j) + \text{Img}(\alpha_j)$ |
| $\alpha_0^*, \alpha_1^*, \ldots$ | Complex conjugates:: $\alpha_j^* = \text{Real}(\alpha_j)$ - $\text{Img}(\alpha_j)$ |
| $\mathbb{C}^n$ | $n$-dimensional vector space over the filed of complex numbers. |
| $\mathcal{H}_n$ | $n$-dimensional Hilbert space. |
| $\delta_{ij}$ | Kronecker's delta function |
| $\mathbf{E}[X]$ | The expected value of the random variable X (discrete case): $\mathbf{E}[X] = \sum_{i=1}^{n} x_i P_X(x_i)$ |
| $\mathbb{I}, \sigma_x, \sigma_y, \sigma_z$ | The identity and the Pauli matrices: $\mathbb{I} \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$; $\sigma_x \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$; $\sigma_y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$; $\sigma_z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$ |
| $\mathbb{I}_n$ | The $n \times n$ identity matrix |
| $\mathbb{R}$ | The field of real numbers |
| $a \in \Omega$ | Element $a$ in (belongs to) set $\Omega$ |
| $A^T$ | Transpose of matrix $A_{mn}$ |

$A^\dagger$ Complex conjugation of matrix $A_{mn}$. For instance, if

$$A = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix}, \text{ then } A^\dagger = \begin{pmatrix} \alpha_1^* & \alpha_2^* & \alpha_3^* \end{pmatrix}$$

$CNOT$ The controlled-NOT gate, its matrix representation is

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$e = 2.718\ldots$ Euler's number

$e^{i\alpha}$ Euler's formula: $e^{i\alpha} = \cos\alpha + i\sin\alpha$

$H$ Hadamard gate or Hamiltonian for a quantum system

$h$ Planck's constant: $h = 6.6262 \times Js$

$H, S, T$ The Hadamard ($H$), phase ($S$), and $\pi/8$ ($T$) matrices for one qubit gates:

$$H \equiv \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}; \qquad S \equiv \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}; \qquad T \equiv \begin{bmatrix} 1 & 0 \\ 0 & \exp(i\pi/8) \end{bmatrix}$$

$HT(P, x)$ The expected number of transition matrix $P$ invocations to reach the state $x$ when started in the initial distribution $\pi$

$i = \sqrt{-1}$ Imaginary number. Square root of -1.

$P_X(x_i)$ The probability of obtaining $x_i$ for random variable $X$

| | |
|---|---|
| $Var[X]$ | The variance of the random variable X (discrete case): $Var[X] = \sum_{i=1}^{n}(x_i - \mathbf{E}[X])^2 P_X(x_i)$ |
| $W(P)$ | Quantum Walk operator based on a classical transition matrix $P$ |
| FPRAS | Fully Polynomial Randomized Approximation Scheme |

# LIST OF REFERENCES


[AAKV01] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, *Quantum Walks on Graph*, Proceedings of ACM Symposium on Theory of Computing (STOC), pp. 50 - 59, New York, NY, 2001

[AC08] A. Childs, *On the Relationship between Continuous- and Discrete-time Quantum Walk*, Communications in Mathematical Physics, 2009

[AC09] A. Childs, *Personal Communication*, March 2009

[AC781] A. Childs, *Lecture Notes on Quantum Algorithms*, http://www.math.uwaterloo.ca/ amchilds/teaching/w08/co781.html

[ACRSZ07] A. Ambainis, A. Childs, B. Reichardt, R. Špalek and S. Zhang, *Any AND-OR Formula of Size N Can Be Evaluated in Time* $N^{1/2+o(1)}$ *on A Quantum Computer*, Proc of 48th FOCS, pp. 363-372, 2007

[AKR05] A. Ambainis, J. Kempe and A. Rivosh, *Coins Make Quantum Walks Faster*, Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1099 - 1108, 2005

[AMB04a] A. Ambainis , *Search via Quantum Walk*, ACM SIGACT News, Vol. 35 (2), 22, 2004

[AMB04] A. Ambainis, *Quantum Walk Algorithm for Element Distinctness*, Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 22–31, 2004

[AT03] D. Aharonov and A. Ta-Shma, *Adiabatic Quantum State Generation and Statistical Zero Knowledge*, Proc. of the 35th annual ACM symposium on Theory, pp. 20–29, 2003

[BACS07] D. Berry, G. Ahokas, R. Cleve and B. Sanders, *Efficient Quantum Algorithms for Simulating Sparse Hamiltonians*, Communications in Mathematical Physics, vol. 270, pp. 359–371, 2007

[BBCMW01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, R. Wolf, *Quantum lower bounds by Polynomials*, Journal of the ACM,, vol. 48, issue 4, pp. 778–797, 2001

[BBH96] M. Boyer, G. Brassard, P. Hoyer and Alain Tapp, *Tight Bounds on Quantum Searching*, Fortschritte Der Physik, vol. 46(4-5), pp. 493 – 505, 1998

[BC11] D. Berry and A. Child, *Black-box Hamiltonian Simulation and Unitary Implementation*, eprint: arXiv:0910.4157v3

[BEST96] A. Barenco, A. Ekert, K. Suominen and P. Törmä, *Approximate Quantum Fourier Transform and Decoherence*, vol. 54, issue 1, pp. 139 - 146, Phys. Rev. A, 1996

[BF60] F. Bauer and C. Fike, *Norms and Exclusion Theorems*, Numer. Math., vol. 2, pp. 137 - 141, 1960

[BKVPY81] M. Blum, R. Karp, O. Vornberger, C. Papadimitriou and M. Yannakakis, *The Complexity of Testing whether a Graph is a SuperConcentrator*, Information Processing Letters, vol. 13, No. 4,5, pp. 164–167, 1981

[BHA97] R. Bhatia, *Matrix Analysis*, Springer Verlag, New York, 1997

[BHT98] G. Brassard, P. Høyer, and A. Tapp, *Quantum Counting*, Proc. of 25th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science, vol. 1443, pp. 820–831, 1998

[BS08] S. Beigi and P. Shor, *On the Complexity of Computing Zero-Error and Holevo Capacity of Quantum Channels*, arXiv:abs/0709.2090

[BSSV06] I. Bezáková, A. Sinclair, D. Štefankovič and E. Vigoda, *Negative Examples for Sequential Importance Sampling of Binary Contingency Tables*, Proc. of Algorithms – ESA 2006, Lecture Notes in Computer Science,Springer Berlin/Heidelberg, vol. 4168, pp. 0302-9743, 2006

[BSVV08] I. Bezáková, D. Štefankovič, V. Vazirani and E. Vigoda, *Accelerating Simulated Annealing for the Permanent and Combinatorial Counting Problems*, SIAM Journal on Computing, vol. 37, no. 5, pp. 1429–1454, 2008

[BT98] B. Boghosian and W. Taylor, *Simulating Quantum Mechanics on a Quantum Computer*, Physica D, vol. 120, pp. 30 – 42, 1998

[CCDFGS03] A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. Spielman, *Exponential Algorithmic Seedup by Quantum Walk*, Proc. 35th ACM Symposium on Theory of Computing, pp. 59–68, 2003

[CHE04] Donny Cheung, *Improved Bounds for the Approximate QFT*, arXiv: abs/quant-ph/0403071, 2004

[CHI10] C. Chiang, *Sensitivity of Quantum Walks with Perturbation*, Proc. of the 10th Asian Conference on Quantum Information Science, pp. 209-210, 2010

[CEMM98] R. Cleve, A. Ekert, C. Macchiavello and M. Mosca, *Quantum Algorithms Revisited*, Proc. Royal Society A, vol. 454, no. 1969, pp. 339-354, 1998

[CER85] V. Černý, *A thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm*, Journal of Optimization Theory and Applications, vol. 45 , pp. 41–51, 1985

[CM01] G. Cho and C. Meyer, *Comparison of Perturbation Bounds for the Stationary Distribution of a Markov Chain*, Linear Algebra and Its Applications, vol. 335, issue 1-3, pp. 137 - 150, 2001

[CNW10] C.-F. Chiang, D. Nagaj, P. Wocjan, *Efficient Circuits for the Quantum Walks*, QIC vol. 10 no. 5&6 pp. 0420–0434, 2010

[CSV07] A. Childs, L. Schulman, and U. Vazirani, *Quantum Algorithms for Hidden Nonlinear Structures*, Proc. 48th IEEE Symposium on Foundations of Computer Science, pp. 395–404, 2007

[CW00] R. Cleve and J. Watrous, *Fast Parallel Circuits for the Quantum Fourier Transform*, IIII Symposium on Foundations of Computer Science, pp. 526 - 536, 2000

[CW10] C. Chiang and P. Wocjan, *Quantum Algorithm for Preparing Thermal Gibbs States - Detailed Analysis*, NATO Science for Peace and Security Series - D: Information and Communication Security, vol.26, pp. 138-147, 2010

[DFK91] M. Dyer, A. Frieze, and R. Kannan, *A Random Polynomial-Time Algorithm for Approximating the Body Volume of Convex Bodies*, Journal of the ACM, vol. 38, no. 1, pp. 1–17, 1991

[DLM92] P. Dagum, M. Luby, M. Mihail, *Polytopes, Permanents and Graphs with Large Factors*, Theoretical Computer Science, vol. 102, issue 2, pp. 283–305, 1992

[EI99] S. Eisenstat and I. Ipsen, *Three Absolute Perturbation Bounds for Matrix Eigenvalues Imply Relative Bounds*, SIAM Journal on Matrix Analysis and Applications, vol. 20 , issue 1, pp. 149 - 158, 1999

[FG98] E. Farhi, S. Gutmann, *Quantum Computation and Decision Trees*, Phys. Rev. A, vol: 58 pp. 915-928, 1998

[FG99] C. Fuchs and J. Graaf, *Cryptographic Distinguishability Measures for Quantum Mechanical States*, IEEE Transactions on Information Theory, vol. 45, issue 4, pp. 1216–1227, 1999

[FGG08] E. Farhi, S. Gutmann and S. Gutmann, *A Quantum Algorithm for the Hamiltonian NAND Tree*, Theory of Computing, vol. 4, pp. 169–190, 2008

[GL96] G. Golub and C. Loan, *Matrix Computations*, 3rd ed., The Johns Hopkins University Press, 1996

[GN96] R. Griffiths and C. Niu, *Semiclassical Fourier Transform for Quantum Computation*, Physical Review Letters, vol. 76, no. 17, 1996

[GOL08] O. Goldreich, *Computational Complexity. A Conceptional Perspective*, Cambridge University Press, 2008

[GOL10] O. Goldreich, *P, NP, and NP-Completeness. The Basics of Computational Complexity*, Cambridge University Press, 2010

[GRO96] L. K. Grover *A Fast Quantum Mechanical Algorithm for Database Search*, Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC), pp. 212 - 219, 1996

[GRO00] L. Grover, *Rapid Sampling through Quantum Computing*, Proc of the 32nd annual ACM symposium on Theory of Computing, pp. 618–626, 2000

[GRO05] L. K. Grover *A Different Kind of Quantum Search*, arXiv:abs/quant-ph/0503205, 2005

[GR02] L. Grover, T. Rudolph, *Creating Superpositions that Correspond to Efficiently Integrable Probability Distributions*, arXiv:abs/quant-ph/0208112, 2002.

[GS97] Ch. M. Grinstead, J. L. Snell, *Introduction to Probability*, American Mathematical Society, 1997

[HAL35] P. Hall, *On Representation of Subsets*, Journal of the London Mathematical Society, vol. 10 , pp. 26-30, 1935

[HAL02] S. Hallgren, *Polynomial-time Quantum Algorithms for Pell's Equation and the Principal Ideal Problem*. In Proceedings of the 34th ACM Symposium on Theory of Computing, 2002

[HOY05] P. Hoyer, *The phase matrix*, Proc. of 16th International Symposium on Algorithms and Computation, Lecture Notes on Computer Science, vol. 3827, pp. 308-317, 2005

[HS98] J. W. Harris and H. Stocker, *Maximum Likelihood Method*. Handbook of Mathematics and Computational Science. New York, Springer-Verlag, pp. 824, 1998

[IN09] I. Ipsen and B. Nadler, *Refined Perturbation Bounds for Eigenvalues of Hermitian and Non-Hermitian Matrices*, SIAM J. Matrix Anal. Appl., vol. 31, no. 1, pp. 40–53, 2009

[JER03] M. Jerrum, *Counting, Sampling and Integrating: Algorithms and Complexity*, Lectures in Mathematics, Birkhäuser, 2003

[JOH01] I. Johnstone, *On the Distribution of the Largest Eigenvalue in Principal Components Analysis*, Annals of Statistics, vol. 29, no. 2, pp. 295 - 327, 2001

[JS89] M. Jerrum and A. Sinclair, *Approximating the Permanent*, SIAM Journal on Computing, vol. 18, pp. 1149–1178, 1989

[JS93] M. Jerrum and A. Sinclair, *Polynomial-Time Approximation Algorithms for the Ising Model*, SIAM Journal on Computing, vol. 22, pp. 1087–1116, 1993.

[JSV04] M. Jerrum, A. Sinclair, and E. Vigoda, *A Polynomial-Time Approximation Algorithm for the Permanent of a Matrix Non-Negative Entries*, Journal of the ACM, vol. 51, issue 4, pp. 671–697, 2004

[JVV86] M. Jerrum, L. Valiant and V. Vazirani, *Random Generation of Combinatorial Structures from a Uniform Distribution*, Theoretical Computer Science, vol. 43, issue 2-3, pp. 169–188, 1986

[JW09] S. Jordan, P. Wocjan, *Efficient Quantum Circuits for Arbitrary Sparse Unitaries* , arXiv:abs/0904.2211, 2009

[KEM03] J. Kempe, *Quantum Random Walk Algorithms*, Contemp. Phys., vol. 44, pp. 302-327, 2003

[KGV83] S. Kirkpatrick, C. Gelatt and M. Vecchi, *Optimization by Simulated Annealing*, Science, New Series, vol. 220, No.4598, pp. 671 - 680, 1983

[KIT95] A. Kitaev, *Quantum Measurements and the Abelian Stabilizer Problem*. Technical Report, arXiv:quant-ph/9511026, 1995

[KLM07] P. Kaye, R. Laflamme and M. Mosca, *An Introduction to Quantum Computing*, Oxford University Press, 2007

[KMOR10] H. Krovi, F. Magniez, M. Ozols, and J. Roland, *Finding is as Easy as Detecting for Quantum Walks*. In 37th International Colloquium on Automata, Languages and Programming (ICALP'10), Lecture Notes in Computer Science, Springer, vol. 6198, pp. 540-551, 2010

[KSV02] A. Kitaev, A. Shen and M. Vyalyi, *Classical and Quantum Computation*, Graduate Studies in Mathematics, American Mathematical Society, vol. 47, 2002

[LLO96] S. Lloyd, *Universal Quantum Simulators*, Science, vol. 273. no. 5278, pp. 1073 - 1078, 1996

[LPW09] D. Levin, Y. Peres and E. Wilmer, *Markov Chains and Mixing Times*, American Mathematical Society, 2008

[LV06] L. Lovász and S. Vempala, *Simulated Annealing in Convex Bodies and an $O^*(n^4)$ Volume Algorithm*, Journal of Computer and System Sciences, vol. 72, issue 2, pp. 392–417, 2006

[ME99] M. Mosca and A. Ekert, *The Hidden Subgroup Problem and Eigenvalue Estimation on a Quantum computer*, arXiv: abs/quant-ph/9903071, 1999

[MRRTT53] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, *Equation of State Calculations by Fast Computing Machines*, J. Chem. Phys., vol. 21, pp. 1087–1092, 1953

[MNRS07] F. Magniez, A. Nayak, J. Roland, and M. Santha, *Search via Quantum Walk*, Proc. of the 39th Annual ACM Symposium on Theory of Computing, pp. 575–584, 2007

[MNRS09] F. Magniez, A. Nayak, P. Richter and M. Santha, *On the Hitting Times of Quantum versus Random Walks*, Proc. of the twentieth annual ACM-SIAM Symposium on Discrete Algorithms, pp. 86 - 95, 2009

[MR95] R. Motwani and P. Raghavan, *Randomized Algorithms*, Cambridge University Press, 1995

[MSS05] F. Magniez, M. Santha, M. Szegedy, *Quantum Algorithms for the Triangle Problem*, Proc. of the 16th ACM-SIAM symposium on Discrete algorithms, 1109, 2005

[MU05] M. Mitzenmacher and E. Upfal, Probability and Computing – Randomized Algorithms and Probabilistic Analysis, Cambridge University Press, 2005

[NC00] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000

[NWZ09] D. Nagaj, P. Wocjan, Y. Zhang, *Fast QMA Amplification*, QIC vol. 9 no. 11&12 pp. 1053–1068, 2009

[PAR98] B. Parlett, *The Symmetric Eigenvalue Problems*, SIAM, Philadelphia, 1998

[PW09] D. Poulin and P. Wocjan, *Thermalizing Quantum Systems and Evaluating Partition Functions with a Quantum Computer*, Physical Review Letter, vol. 103, pp. 220502, 2009

[RIC07] P. Richter, *Almost Uniform Sampling via Quantum Walks*, New Journal of Physics, vol. 9, article 72, 2007

[RIC07a] P. Richter, *Quantum Speed-Up of Classical Mixing Processes*, Physical Review A, vol. 76, 042306, 2007

[RS08] B. W. Reichardt, R. Špalek, *Span-program-based quantum algorithm for evaluating formulas*, Proceedings of the 40th STOC, 103, 2008

[RZBB94] M. Reck, A. Zeilinger, H. Bernstein, P. Bertani, *Experimental Realization of Any Discrete Unitary Operator*, Phys. Rev. Lett. vol. 73, no. 58, 1994

[SAN08] M. Santha, *Quantum Walk Based Search Algorithms*, Proc. of 5th Theory and Applications of Models of Computation (TAMC08), Lectures Notes on Computer Science, vol. 4978, pp. 31–46, 2008

[SBB07] R. Somma, S. Boixo, and H. Barnum, *Quantum Simulated Annealing*, arXiv:abs/0712.2008

[SBBK08] R. Somma, S. Boixo, H. Barnum, E. Knill, *Quantum Simulations of Classical Annealing Processes*, Phys. Rev. Lett. vol. 101, pp. 130504, 2008

[SHO94] P. Shor, *Algorithms for Quantum Computation: Discrete Logarithms and Factoring*. Proceedings of FOCS, pp. 124 - 134, 1994.

[SHO05] P. Shor, *Polynomial-time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, SIAM Journal on Computing, 26(5):1484-1509, 2005

[SIN92] A. Sinclair, *Algorithms for Generation & Counting – A Markov Chain Approach*, Birkhäuser, 1992

[SIP05] M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing Company, 2005

[SIV96] D. Sivia, *Data Analysis, a Bayesian Tutorial*, Oxford University Press, 1996

[SUZ88] M. Suzuki, *Quantum Monte Carlo Methods in Equilibrium and Nonequilibrium Systems* Springer Series in Solid-State Science, vol. 74 1988

[SVV07] D. Štefankovič, S. Vempala, and E. Vigoda, *Adaptive Simulated Annealing: A Near-Optimal Connection between Sampling and Counting*, Proc. of the 48th Annual IEEE Symbosium on Foundations of Computer Science, pp. 183-193, 2007

[SZE04] M. Szegedy, *Quantum Speed-up of Markov Chain Based Algorithms*, Proc. of 45th Annual IEEE Symposium on Foundations of Computer Science, pp. 32–41, 2004

[TD00] B. Terhal and D. DiVincenzo, *Problem of equilibration and the Computation of Correlation Functions on a Quantum Computer*, Physical Review A, vol. 61, pp. 022301, 2000

[TOVPV09] K. Temme, T. Osborne, K. Vollbrecht, D. Poulin and F. VerstraeteK, *Quantum Metropolis Sampling*, arXiv: abs/0911.3635, 2009

[Turan41] P. Turán, *On An Extreme Problem in Graph Theory*, Matematikai és Fizikai Lapok, vol.48, pp. 436–452, 1941

[VAL79] L. G. Valiant, *The Complexity of Computing the Permanent*, Theoretical Computer Science, vol. 8, pp. 189–201, 1979

[WA08] P. Wocjan and A. Abeyesinghe, *Speed-up via Quantum Sampling*, Physical Review A, vol. 78, pp. 042336, 2008

[WCNA09] P. Wocjan, C. Chiang, D. Nagaj and A. Abeyesinghe, *A Quantum Algorithm for Approximating Partition Functions*, Physical Review A, vol. 80, pp. 022340, 2009

[ZAL98] C. Zalka, *Simulating Quantum Systems on a Quantum Computer*, Proc. R. Soc. London, Ser. A, vol. 454, no. 1969, pp. 313 – 322, 1998