QUANTUM WALKS AND GROUND STATE PROBLEMS

BY PETER COURTLAND RICHTER

A dissertation submitted to the

Graduate School—New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Computer Science

Written under the direction of

Mario Szegedy

and approved by

____

____

____

____

New Brunswick, New Jersey

October, 2007

ABSTRACT OF THE DISSERTATION

Quantum walks and ground state problems

by Peter Courtland Richter

Dissertation Director: Mario Szegedy

Since the appearance of Shor's factoring algorithm in 1994, the search for novel quantum computer algorithms has proved surprisingly difficult. Two design approaches that have yielded some progress are quantum walks and adiabatic computing. The former has been shown to speed up algorithms whose complexity is related to the classical hitting time of a symmetric Markov chain, and there is evidence that the latter speeds up simulated annealing algorithms for computing ground states of classical Hamiltonians.

In this thesis, we look into the possibility of obtaining a quantum speedup for the mixing time of a symmetric Markov chain. We prove that by subjecting a quantum walk to a small amount of decoherence (typically the adversary of a quantum computer), it can be forced to mix to the correct stationary distribution, often considerably faster than its classical counterpart. A more general theorem to this effect would imply quantum speedups for a variety of approximation algorithms for #P-complete problems.

We conclude with some observations on adiabatic computing – a time-dependent generalization of the quantum walk framework – and the problem of estimating the ground state energy of a quantum Hamiltonian with local spin interactions.

Acknowledgements

I would like to thank Mario Szegedy for his support and guidance of my thesis research. His range and creativity as a researcher are as inspiring to me as they are impressive.

I have benefited greatly from the instruction, advice, and good nature of the Rutgers computer science faculty. In particular, I would like to thank Eric Allender, Ahmed Elgammal, Martin Farach-Colton, Michael Grigoriadis, Leonid Khachiyan, Joe Kilian, Casimir Kulikowski, Naftaly Minsky, Muthu Muthukrishnan, and Endre Szemerédi.

From among the many distinguished members and visitors of the computational complexity and quantum computing groups at Rutgers and NEC Labs, I would especially like to thank Xiaomin Chen, Lara Faoro, Sean Hallgren, Rajat Mittal, Martin Rötteler, Robert Špalek, and Fengming Wang for their contributions.

During the summers of 2005 and 2006, I had the pleasure of visiting the Institute for Quantum Computing (IQC) at the University of Waterloo and the Laboratoire de Recherche en Informatique (LRI) at Université Paris-Sud XI. For making these visits enjoyable ones, I would like to thank my hosts – Richard Cleve and Miklos Santha – and also Scott Aaronson, Andris Ambainis, Arvid Bessen, Christoph Dankert, Jordan Kerenidis, Sophie Laplante, Troy Lee, Frédéric Magniez, and Michele Mosca.

I am particularly grateful to Viv Kendon, whose research with Ben Tregenna led me to an idea that proved to be the starting point of my research for this thesis. She and Todd Brun were both helpful in providing suggestions on how to present my research to a physics audience and in proposing directions for future research.

Thanks to the outstanding mentors and collaborators I have had over the years, including Gary Miller, Steve Miller, Larry Peterson, Peter Sarnak, and Robert Tarjan.

And most of all, thanks to Mom, Dad, and Tim for their love and support.

Table of Contents

List of Tables

List of Figures

Chapter 0

Introduction

Quantum computing is an interdisciplinary area of research spanning physics, computer science, and mathematics. It is concerned with two fundamental questions:

1. *Can a scalable computer be built using quantum bits and logic gates?*
2. *What problems would become more tractable on such a computer?*

The former is of interest to physicists who view the challenge of building a *quantum computer* as a test for engineering and controlling large-scale quantum systems. The latter preoccupies computer scientists studying *quantum algorithms and complexity* and is the subject of this thesis.

Thus far, only three types of computational problems have been shown to be solvable more efficiently on a quantum computer than on a classical computer. The first of these is *algebraic and number-theoretic problems* – most prominently, the *integer factorization* and *discrete logarithm* problems, for which Peter Shor discovered efficient quantum algorithms in 1994 [98]. Both problems are believed to be intractable on a classical computer; in fact, the security of most encrypted electronic communications in use today relies on precisely this assumption.

The second is *combinatorial search problems.* Lov Grover proved in 1996 that brute-force quantum search of a database of N items can be accomplished using only $O(\sqrt{N})$ queries [52]. A simple extension of Grover's algorithm called *amplitude amplification* boosts the success probability of a randomized search algorithm to 0.99 from any $\epsilon > 0$ by running it only $O(\sqrt{1/\epsilon})$ times [28]. Each of these quantum algorithms is optimal and quadratically better than the best classical algorithm [25].

The third is *simulation of quantum systems.* Physicists typically identify a quantum system by its *Hamiltonian*, the Hermitian operator that drives Schrödinger's equation.

In a system consisting of n locally interacting particles with 2 or more spin states each, the Hamiltonian has a succinct (polynomial-size) matrix representation but the system state is described by an exponential-size vector. Richard Feynman suggested in 1982 [48] that a quantum computer could be used to simulate the dynamics of such a system efficiently. Since then, a line of research [44, 108, 33, 7, 32, 27] has confirmed this possibility and produced a considerable algorithmic framework surrounding Feynman's original idea.

The lack of orthogonal breakthroughs in the field has spurred researchers to extend these quantum algorithms to *paradigms* or meta-algorithms and to identify the most general classes of problems they can efficiently solve. Kitaev [67] showed how to use *phase estimation*, an extension of Shor's algorithm, to solve the more general *Abelian hidden subgroup problem.* Ambainis [16] showed that *quantum walks* – the quantum analogues of classical Markov chains – can solve *element distinctness* and other structured search problems more elegantly and efficiently than Grover's algorithm or amplitude amplification. Farhi et al. [44] proposed simulating the *adiabatic* dynamics of a time-dependent spin Hamiltonian in order to compute its *ground state* – the eigenvector whose eigenvalue is smallest – more efficiently than randomized local search strategies like simulated annealing.

The extension of Grover's algorithm to *quantum walk algorithms*, beginning with Ambainis' algorithm for the element distinctness problem, has proved particularly fruitful. Random walks are widely used in classical search ("hitting") algorithms and sampling ("mixing") algorithms, and researchers suspected prior to Ambainis' discovery that quantum walks might speed up these algorithms. Nayak et al. [81, 17] and Aharonov et al. [4] were the first to investigate whether quantum walks might speed up classical mixing algorithms, but their results were inconclusive: though quantum walks "mix" more quickly than their classical counterparts on one-dimensional cycles, their behavior on more complex graphs (likely to appear in algorithmic applications like approximate counting [41, 57, 58, 59]) is less obvious. On the other hand, Ambainis [16] and other researchers [75, 30, 73, 19] have found that quantum walks *do* in many cases speed up classical hitting algorithms.

Is it the case then that while quantum walks can speed up hitting algorithms, they cannot speed up mixing algorithms? In the author's opinion, there is sufficient reason to believe that this prospect is unlikely. Nevertheless, the quantum speedup of a classical mixing algorithm has yet to be demonstrated.

Overview of the dissertation. This thesis is focused on *quantum walks and their algorithmic applications* to search and sampling problems and on *quantum ground state problems* like the local Hamiltonian and their solution by adiabatic algorithms.

Our primary original contribution is a definition and analysis of the "quantum mixing time." We motivate this concept by introducing first the "quantum hitting time."

A *Markov chain* is a sequence of random variables distributed according to $\{P^t q\}_{t=1}^{\infty}$, where q is an initial probability distribution on the "states" $[N] := \{0, 1, \ldots, N-1\}$ and P is an $N \times N$ stochastic matrix of random "state transitions" $[N] \to [N]$. We frequently use the term "Markov chain" when referring to P as well. Suppose that a subset $M \subseteq [N]$ of the states is "marked," and let the *hitting time* h_u denote the expected time to transit to a marked state for the first time when starting from the uniform distribution $q = u$. Local search algorithms often reduce to this random search process, in which case their complexity is strongly related to the hitting time [13, 96]. Building on the work of several other researchers [109, 97, 16, 20], Szegedy [105] showed that if P (the matrix) is symmetric, there is a natural *quantum walk* W with coefficients derived from those of P whose *quantum hitting time* (the expected time for W to detect whether or not $|M| > 0$) is no more than $O(\sqrt{h_u})$. This generalizes both Grover's algorithm [52] and amplitude amplification [28], and it can be used to solve several structured search problems – including element distinctness – faster than any classical algorithm or (apparently) any quantum algorithm based only on amplitude amplification [75, 30, 73, 19].

If we run the Markov chain $\{P^t q\}_{t=1}^{\infty}$ from an initial state $x \in [n]$ *without* stopping at a marked state (or equivalently, if we let $|M| = 0$), it converges to its stationary (equilibrium) distribution π. The time required for $P^t q$ to become ϵ-close to π (in ℓ_1 distance) is the *mixing time* τ_x. Many *Markov chain Monte Carlo* algorithms –

most famously, approximation algorithms for #P-complete problems like evaluating the permanent of a nonnegative matrix [59] or the volume of a convex body [41] – reduce to this random process for (approximate) sampling from π, and their complexity is fundamentally tied to the mixing time. In practice, τ_x is often estimated by the inequality [11, 40, 99]

$$\delta^{-1} \leq \tau_x \leq \delta^{-1} \log 1/\pi(x) \tag{1}$$

where δ is the *spectral gap* of P – the difference between its two largest eigenvalues (in magnitude). But suppose that P is the (symmetric) simple random walk on a regular graph $G = (V, E)$, in which case $\pi = u$ is the uniform distribution and $\log 1/\pi(x) = \log N$. It can be shown [103] that the *diameter* $d(G)$ (the largest distance between any two vertices in G) is only:

$$O(\sqrt{\delta^{-1}} \log N) \tag{2}$$

The diameter is an obvious lower bound on the complexity of sampling uniformly at random from the vertices of G from a worst-case initial vertex x, so it is natural to ask: Is there is a sampling process with complexity (2) rather than (1)?

We prove that the answer is yes when G is the d-dimensional torus $\mathbb{Z}_n^d$ for any $n \geq 2$ and $d \geq 1$. This process is a *decoherent quantum walk* – i.e., a quantum walk subjected at random to repeated external measurements. Ironically, decoherence is widely acknowledged to be the chief physical obstacle to the implementation of a quantum computer. Kendon and Tregenna [65] first demonstrated in numerical experiments that decoherence can (even in small amounts) force a quantum walk to converge to the uniform distribution quite rapidly. On the other hand, Aharonov et al. [4] proved that the standard quantum walk on a regular graph G not subject to decoherence – which propagates more quickly than its classical random walk counterpart [81, 17, 4] – tends not to "converge" to the uniform distribution except in rare instances. We formulate a new notion of the *quantum mixing time* of a decoherent quantum walk derived from a classical Markov chain P, and we prove (a) that it is finite (i.e., that the limiting distribution is always uniform) if P is symmetric, (b) that it is robust (i.e., essentially invariant) under any "reasonable" model of decoherence, and (c) that it beats the upper bound (2) when P is the simple random walk on the torus $\mathbb{Z}_n^d$, indicating that the "uniform mixing" property of decoherent quantum walks observed by Kendon and Tregenna [65] can be combined with the "fast propagation" property of Aharanov et al. [4] to yield a quantum speedup of the classical mixing time. A more general theorem to this effect (extended to the class of so-called *reversible* Markov chains) would imply quantum speedups for a variety of approximation algorithms for #P-complete problems.

Our secondary contributions are as follows. First, we extend Szegedy's quantum speedup of the classical hitting time [105] (in slightly weaker form) from the class of symmetric Markov chains to the class of reversible Markov chains. More precisely, we show that the quantum hitting time of a reversible Markov chain P is at most

$$O(\sqrt{1/(1-||P_M||)}) \tag{3}$$

where P_M is the *leaking walk matrix* obtained from P by deleting all rows and columns indexed by the marked states M. Magniez et al. [74] showed using an elegant algorithm that the weaker upper bound

$$O(\sqrt{1/(\delta\varepsilon)}) \tag{4}$$

holds, where $\varepsilon := \sum_{x\in M}\pi(x)$ is the fraction of marked states weighted according to the stationary distribution π of P. This shows (in a common but not universal simplification of the underlying cost model) that Szegedy's algorithm remains faster, for example, on "low-dimensional" reversible Markov chains.

Second, we present some new observations on the the complexity of the *local Hamiltonian problem* – estimating the ground state energy of a spin Hamiltonian specified by its local interaction terms – and its connection to the *adiabatic computing* framework. We prove that the local Hamiltonian problem drops from QMA-complete to BQP-complete if we are promised that the Hamiltonian has a "low-energy" state whose "overlap" (inner product) with an efficiently computable quantum state is large (inverse-polynomial size). BQP is the class of decision problems *computable* by efficient quantum algorithms (the quantum analogue of the determistic and randomized complexity classes P and BPP) and QMA is the class of decision problems *verifiable*, or checkable, by efficient quantum algorithms (the quantum analogue of the classical nondeterministic classes NP and MA). Our BQP-completeness proof requires little modification to Kitaev's proof [68] that the ordinary local Hamiltonian problem is QMA-complete, but it has the desirable property of producing a ground state isolated from its excited states by a large spectral gap and relates conceptually to perturbation theory calculations and adiabatic algorithm design.

The outline of this thesis is as follows. In Part I, we introduce the standard quantum circuit model of a quantum computer and present quantum algorithms for the integer factorization and quantum simulation problems. Along the way, we review the discrete-time (product of unitaries) and continuous-time (Hamiltonian) descriptions of quantum mechanics and the basic framework of computational complexity. The textbooks by Nielsen and Chuang [82] and Kitaev, Shen, and Vyalyi [68] are excellent resources for additional background on quantum computing.

In Part II (the major one), we review Grover's algorithm and amplitude amplification (the starting points for the study of quantum walks), show how to construct quantum walks from classical random walks, prove that quantum walks can speed up the classical hitting time (and in turn, several search algorithms), and then prove that they can speed up the classical mixing time as well. Part II includes material from the author's short survey articles [92, 89] and original work of the author published in [91, 90].

In Part III, we turn our attention from quantum walks to quantum ground state problems. We explain the framework of adiabatic computing and a present a few adiabatic algorithms, and we review and offer some new observations on the closely related local Hamiltonian problem. This is the subject of the author's ongoing work, some of which is detailed in the final chapter.

Part I

Fundamentals of quantum computing

Chapter 1

Quantum circuits

In this chapter, we review the basic principles of discrete-time quantum mechanics and theoretical computer science, then introduce the *quantum circuit* model. For illustration, we present Kitaev's quantum circuit for *phase estimation* and show how it solves the factoring problem.

1.1 Discrete-time quantum mechanics

The evolution of a closed quantum system is governed by a *unitary process* acting on a complex vector, or *wavefunction*. Mathematically, a parallel can be drawn to the classical scenario in which a *probability distribution* is acted on by a *stochastic process*. However, the physical interpretations are very different.

1.1.1 Postulates of quantum mechanics

The following postulates are fundamental to quantum mechanics – see e.g., Neilsen and Chuang [82].

Quantum states. Let $\mathcal{S}$ be the set of all possible classical states of an isolated physical system – for example, the configurations of a collection of particles or the settings of a computer memory. We shall assume that $N := |\mathcal{S}| < \infty$ except where indicated otherwise. Denote by $\mathcal{H}_{\mathcal{S}}$, or simply $\mathcal{H}$, the N-dimensional complex Hilbert space generated by the *classical basis* $\mathcal{C}_{\mathcal{S}}$ of elementary column vectors with coordinates indexed by the elements of $\mathcal{S}$. The first postulate of quantum mechanics characterizes the states of the *quantum system* with classical states $\mathcal{S}$.

Postulate 1.1 *A* quantum state*, or* wavefunction*, is a vector* $|\psi\rangle \in \mathcal{H}$ *with* $\langle\psi|\psi\rangle = 1$.

Here $|\cdot\rangle$ is *Dirac's notation.* It is used widely in quantum mechanics to denote a column vector $|\psi\rangle$, its conjugate transpose $\langle\psi|$, and the inner and outer products $\langle\psi|\phi\rangle$ and $|\psi\rangle\langle\phi|$ of two vectors.

If $x \in \mathcal{S}$, we use $|x\rangle$ to denote the elementary vector from $\mathcal{C}_{\mathcal{S}}$ associated with x. The *amplitude* α_x of a wavefunction $|\psi\rangle$ at x is the inner product $\alpha_x = \langle x|\psi\rangle$. A quantum amplitude α_x is related to a classical probability in that $|\alpha_x|^2$ gives the probability of finding $|\psi\rangle$ in the state x if it is *measured* (or observed) in the classical basis.[1] However, a wavefunction $|\psi\rangle$ with nonzero amplitudes α_x and α_y is in *neither* classical state x nor classical state y prior to this observation, but rather a *superposition* of both states. And unlike probabilities, amplitudes can undergo *interference*, or destructive cancellation – a phenomenon exploited widely by quantum algorithms.

Unitary evolution. The second postulate of quantum mechanics states that the evolution of a closed quantum system is a *unitary process.*

Postulate 1.2 *The time evolution U of a closed quantum system from t to t' is unitary.*

An operator $U : \mathcal{H} \to \mathcal{H}$ is *unitary* if it satisfies $U^{-1} = U^{\dagger}$, where $\dagger$ is the conjugate transpose. By the spectral theorem of linear algebra, U has a spectral decomposition $\{\lambda_j, |\phi_j\rangle\}$ where $\lambda_j = e^{i\theta_j}$ and the $|\phi_j\rangle$ form an orthonormal basis of $\mathcal{H}$. Geometrically, a unitary transformation is a *change of basis* between two orthonormal bases.

Each of the $N!$ permutation matrices $P : \mathcal{H}_{\mathcal{S}} \to \mathcal{H}_{\mathcal{S}}$ is unitary. So is the *Fourier transform* $F_n : \mathcal{H}_{\mathbb{B}^n} \to \mathcal{H}_{\mathbb{B}^n}$, which is given by

$$F_n : |x\rangle \mapsto \sum_{y \in \mathbb{B}^n} e^{\frac{2\pi i x \cdot y}{2^n}} |y\rangle \tag{1.1}$$

where $x \in \mathbb{B}^n$ is an n-bit string and $\mathbb{B} := \{0, 1\}$ is a classical bit. Permutation matrices and Fourier transforms are ubiquitous in quantum computing.

Projective measurement. When a previously isolated quantum system interacts with its environment, its evolution is no longer unitary. This interaction is called *measurement.* The most basic type of measurement is *projective.*

[1] The concept of quantum measurement will be more fully explained shortly.

Postulate 1.3 Projective measurement *of a wavefunction* $|\psi\rangle$ *is performed with respect to a Hermitian operator or* observable $H = \sum_k m_k P_k$, *where the* m_k *are distinct real values and the* P_k *form a complete system of orthogonal projectors. If* $|\psi\rangle$ *is measured with respect to* H, *then with probability* $p_k = \langle\psi|P_k|\psi\rangle$, *the result of the measurement is the* outcome m_k *and the* collapse *of* $|\psi\rangle$ *to the state* $\sqrt{1/p_k} \cdot P_k|\psi\rangle$.

An operator $H : \mathcal{H} \to \mathcal{H}$ is *Hermitian* if $H = H^\dagger$. By the spectral theorem, H has a spectral decomposition $\{\lambda_j, |\phi_j\rangle\}$ where the λ_j are real and the $|\phi_j\rangle$ form an orthonormal basis of $\mathcal{H}$. The expression $H = \sum_k m_k P_k$ is just the spectral decomposition, with $P_k = \sum_{j:\lambda_j = m_k} |\phi_j\rangle\langle\phi_j|$.

If H has distinct eigenvalues, then $m_k = \lambda_k$ and $P_k = |\phi_k\rangle\langle\phi_k|$, and the result of the measurement is a collapse of $|\psi\rangle$ to the eigenstate $|\phi_k\rangle$ with probability $p_k = |\langle\phi_k|\psi\rangle|^2$. Measurement in the classical basis is obtained by choosing H to be diagonal so that its eigenstates are classical states.

Composite systems. The final postulate of quantum mechanics stipulates how a composite quantum system is described by its component systems. Let $\otimes$ denote the tensor product operation.

Postulate 1.4 *If the states* $|\phi\rangle \in \mathcal{H}$ *and* $|\phi'\rangle \in \mathcal{H}'$ *describe two quantum systems in isolation, then the state* $|\phi\rangle \otimes |\phi'\rangle \in \mathcal{H} \otimes \mathcal{H}'$ *describes their composition.*

For brevity, we often write $|\phi\rangle|\phi'\rangle$, $|\phi, \phi'\rangle$, or $|\phi\phi'\rangle$ in place of $|\phi\rangle \otimes |\phi'\rangle$.

A quantum computer memory $\mathcal{B}^{\otimes n} = \mathcal{H}_{\{0,1\}^n}$ is a composite system whose components are *qubits* $\mathcal{B} := \mathcal{H}_{\mathbb{B}}$. This does *not* mean that every state of a quantum computer is expressible as a tensor product of one-qubit states: the state $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \in \mathcal{B}^{\otimes 2}$ is a simple counterexample. Such states are *entangled*. Like interference, entanglement is a critical resource in quantum algorithms.

1.1.2 The density matrix formalism

The aforementioned postulates are sufficient to describe the evolution of any *pure* quantum state $|\psi\rangle$ from its creation to its collapse. But in practice, we must be prepared to deal with *mixed* quantum states specified by ensembles $\{p_j, |\psi_j\rangle\}$, where p_j is the probability that the quantum state is $|\psi_j\rangle$. Mixed states are encountered when interleaving quantum measurements or classical stochastic processes with unitary quantum processes. They are described elegantly by the *density matrix formalism* of quantum mechanics.

Density matrices. Let $\{p_j, |\psi_j\rangle\}$ be the ensemble of a mixed quantum state.

Definition 1.5 *The* density matrix ρ *of* $\{p_j, |\psi_j\rangle\}$ *is* $\rho := \sum_j p_j |\phi_j\rangle\langle\phi_j|$.

Denote by $\mathbf{L}(\mathcal{H})$ the space of linear operators $\mathcal{H} \to \mathcal{H}$. Clearly, a density matrix is an element of $\mathbf{L}(\mathcal{H})$ which is Hermitian positive semidefinite with trace one. Conversely, it follows from the spectral theorem that any positive Hermitian semidefinite operator with trace one is the density matrix of a mixed quantum state. The map from ensembles to density matrices is not injective; however, it is fairly simple to show that two ensembles map to the same density matrix if and only if they are indistinguishable by any quantum measurement.

A rank-one density matrix $\rho = |\psi\rangle\langle\psi| \in \mathbf{L}(\mathcal{H})$ describes a pure quantum state $|\psi\rangle \in \mathcal{H}$. The density matrix $\rho \in \mathbf{L}(\mathcal{H})$ of a classical distribution p is diagonal with $\rho_{jj} = p_j$. Two density matrices ρ, ρ' are composed by taking the tensor product $\rho \otimes \rho'$.

Quantum transformations. A general (non-unitary) quantum process is a linear map $\mathbf{L}(\mathcal{H}) \to \mathbf{L}(\mathcal{H}')$ between operator spaces, or a *superoperator*. In particular, it maps density matrices from $\mathbf{L}(\mathcal{H})$ to density matrices from $\mathbf{L}(\mathcal{H}')$.

Definition 1.6 *A* quantum transformation $T : \mathbf{L}(\mathcal{H}) \to \mathbf{L}(\mathcal{H}')$ *is a superoperator that is trace-preserving and completely positive.*

A superoperator $T : \mathbf{L}(\mathcal{H}) \to \mathbf{L}(\mathcal{H}')$ is *trace-preserving* if $\mathrm{tr}(T(\rho)) = \mathrm{tr}(\rho)$ for every $\rho \in \mathbf{L}(\mathcal{H})$ and *completely positive* if $T \otimes I_n$ maps the set of Hermitian positive semidefinite operators to itself for every $n \geq 1$.

Unitary evolution U is represented by the quantum transformation $T : \rho \mapsto U\rho U^\dagger$. Projective measurement $H = \sum_k m_k P_k$ of the density matrix ρ yields with probability

$p_k = \mathrm{tr}(\rho P_k)$ the outcome m_k and the density matrix $\frac{1}{p_k} \cdot P_k \rho P_k$.

1.2 Classical and quantum circuits

Both classical and quantum algorithms can be described by *circuits.* A circuit computes a function of its input by applying a sequence of logic gates. A unitary quantum gate can take as input a superposition of classical inputs and generate as output another superposition. Whether circuits built from quantum gates are more powerful than those built from classical gates is the central problem of *quantum complexity theory.*

1.2.1 Computational complexity

Before discussing quantum circuits, we briefly review some aspects of *computational complexity* to establish context. The textbooks of Papadimitriou [85] and Arora and Barak [22] are excellent resources for additional background.

Algorithms and computability. An *algorithm* is (loosely) a finite set of instructions, or program, for computing a function $f : \mathbb{B}^* \to \mathbb{B}^*$. A function for which an algorithm exists is called *computable.* Informally, a *Turing machine* T_f executes an algorithm for a function f using a finite-state automaton (processor) which reads an input x from a worktape (memory), performs calculations using the worktape as scratch space, and writes the output $f(x)$ to the worktape. The widely accepted *Church-Turing thesis* underpinning computer science postulates that *any* algorithm can be realized by a Turing machine:

Postulate 1.7 *For any computable function f, there exists a Turing machine T_f computing f.*

A family of *circuits* $\mathcal{C} = \{C_n : \mathbb{B}^n \to \mathbb{B}^*\}_{n \geq 1}$ composed of AND, OR, and NOT gates also computes a function $f : \mathbb{B}^* \to \mathbb{B}^*$ – in fact, any function f can be computed by a circuit family. But this does not mean that every function is computable. To qualify as an algorithm, a circuit family must be *uniform*: there must exist a Turing machine $T_{\mathcal{C}}$ which computes on input 1^n a description of the circuit C_n.

Efficiency and complexity. An algorithm is of little practical use unless it executes quickly. We say that an algorithm computing a function $f : \mathbb{B}^* \to \mathbb{B}^*$ has *time complexity* $t : \mathbb{N} \to \mathbb{R}$ if the algorithm terminates within $t(n)$ steps on any input $x \in \mathbb{B}^n$. When measuring time complexity (or any other function), we write $t(n) = O(s(n))$ if there is a constant c such that $t(n) \leq c \cdot s(n)$ for n sufficiently large. Moreover, we write $t(n) = \Omega(s(n))$ if $s(n) = O(t(n))$, and $t(n) = \Theta(s(n))$ if $t(n) = O(s(n))$ and $t(n) = \Omega(s(n))$. The complexity class P consists of those *predicates* (or *decision problems*) $f : \mathbb{B}^* \to \mathbb{B}$ having an *efficient* algorithm – i.e., one whose time complexity is bounded above by a fixed polynomial $p_f(n)$. Equivalently, a predicate is in P if the Turing machine $T_{\mathcal{C}}$ computing its uniform circuit family $\mathcal{C}$ is efficient. Predicates in P are considered to have efficient (deterministic) algorithms.

A *probabilistic Turing machine* (or *randomized algorithm*) for a function $f : \mathbb{B}^* \to \mathbb{B}^*$ computes a function $g : \mathbb{B}^{n+k} \to \mathbb{B}^*$ of an input $x \in \mathbb{B}^n$ to f and k uniformly distributed *random bits* r such that $\mathrm{E}_r[g(x, r) = f(x)] \geq 2/3$ for every x, where $\mathrm{E}[\cdot]$ denotes the expected value. The complexity class BPP consists of those predicates $f : \mathbb{B}^* \to \mathbb{B}$ having an *efficient* randomized algorithm – i.e., one whose time complexity is bounded above by a fixed polynomial $p_f(n)$. BPP trivially contains P. Bernstein and Vazirani [26] assert that modern computational complexity (as of 1993) rests on the following strengthening of the Church-Turing thesis:

Postulate 1.8 *Any algorithmic process can be simulated efficiently on a probabilistic Turing machine.*

Here *efficient* means *with polynomial slowdown.* Thus, it makes sense to think of BPP as representing the class of tractable problems.[2]

Suppose that a predicate is *efficiently verifiable* in the following sense: There is an algorithm (called a *verifier*) computing $f : \mathbb{B}^* \to \mathbb{B}$ in time polynomial in n provided it is given a *witness* (i.e., a hint) $w(x) \in \mathbb{B}^k$ in addition to the input $x \in \mathbb{B}^n$. This predicate is said to be in the complexity class NP. Clearly P $\subseteq$ NP, but it is strongly believed that P $\neq$ NP – i.e., that a predicate which is efficiently verifiable is not necessarily efficiently

[2] Surprisingly, there is strong evidence that P = BPP – but this is beyond the scope of this thesis.

computable. NP has *complete problems*, or predicates that are not in P unless P = NP. If we allow the verifier to be a randomized algorithm, we obtain the class MA. Note that BPP $\subseteq$ MA and NP $\subseteq$ MA. See Figure 1.1 for a diagram showing all of the complexity classes referenced in this thesis and their containment relationships.

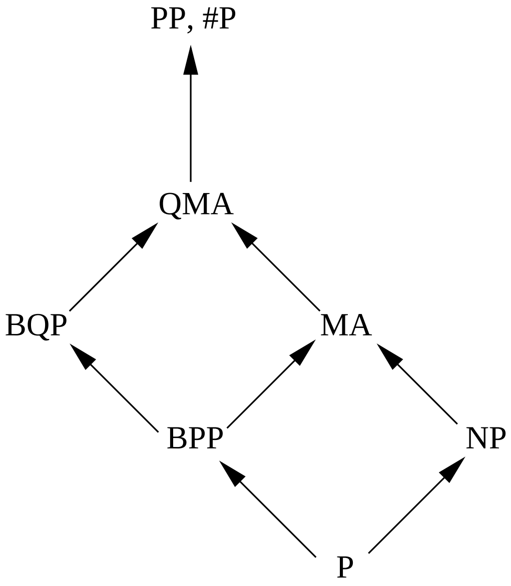

Figure 1.1: Known inclusions among classical and quantum complexity classes.

1.2.2 Quantum circuits and BQP

An n-qubit *quantum circuit* $U_n : \mathcal{B}^{\otimes n} \rightarrow \mathcal{B}^{\otimes n}$ is composed of local unitary gates rather than AND, OR, and NOT gates. Like any unitary transformation, a quantum circuit must be invertible, so we typically depict it using parallel *wires* without the *fan-in* of classical circuit diagrams. Just as AND, OR, and NOT (or just NAND) form a *universal gate set* for classical circuits, there are small sets of local unitary operators that form universal gate sets for quantum circuits [68]. However, *universal* in the latter case means that for every unitary operator $U : \mathcal{B}^{\otimes n} \rightarrow \mathcal{B}^{\otimes n}$ (of which there is a continuum) and every $\epsilon > 0$, a quantum circuit U' can be constructed satisfying:

$$||U - U'||_2 \leq \epsilon \tag{1.2}$$

The *Solovay-Kitaev theorem* gives an upper bound of $O(\log^4 \epsilon^{-1})$ elementary gates on the size of U'.

A *quantum algorithm* is a family of quantum circuits whose description can be generated uniformly by a (classical) Turing machine.[3] We say that a quantum circuit family $\{U_n\}_{n\geq 1}$ computes a function $f : \mathbb{B}^* \to \mathbb{B}^*$ if the value $f(x)$ is observed with probability at least $2/3$ when U_n is applied to the classical state $|x\rangle$ with $x \in \mathbb{B}^n$. The complexity class BQP consists of those predicates $f : \mathbb{B}^* \to \mathbb{B}$ having an *efficient* quantum algorithm – i.e., one whose time complexity is bounded above by a fixed polynomial $p_f(n)$. Bernstein and Vazirani [26] showed that BPP $\subseteq$ BQP.

Let us see a few simple but important examples of quantum circuits:

Reversible classical gates. An *oracle* is (loosely) a black-box subroutine that returns the value $f(x)$ of a predicate $f : \mathbb{B}^n \to \mathbb{B}$ when *queried* with the input x. The *query complexity* of an algorithm querying an oracle is the number of times the oracle is queried. The *XOR oracle* $X_f : \mathcal{B}^{\otimes n} \otimes \mathcal{B} \to \mathcal{B}^{\otimes n} \otimes \mathcal{B}$ given by

$$X_f : |x\rangle|y\rangle \mapsto |x\rangle|y \oplus f(x)\rangle, \tag{1.3}$$

where $\oplus$ is the XOR operation, is a reversible (unitary) classical oracle gate. So is the *Toffoli gate* $T : \mathcal{B}^{\otimes 3} \to \mathcal{B}^{\otimes 3}$ implementing the permutation

$$T : |a\rangle|b\rangle|c\rangle \mapsto |a\rangle|b\rangle|c \oplus ab\rangle. \tag{1.4}$$

Bennett [24] showed that the Toffoli gate is universal for *reversible* classical circuits, which are as powerful as circuits with fan-in. Every reversible classical circuit is a permutation on the set of classical basis states.

One-qubit quantum gates. A *phase shifter* $R(\theta) : \mathcal{B} \to \mathcal{B}$ implements the transformation:

$$R(\theta) : \begin{bmatrix} 1 & 0 \\ 0 & e^{i\theta} \end{bmatrix} \tag{1.5}$$

[3] Bernstein and Vazirani [26] first defined a quantum algorithm in terms of a *quantum Turing machine*, but Yao [110] later showed that this simpler definition is equivalent.

The *Hadamard gate* $H : \mathcal{B} \to \mathcal{B}$ implements the Fourier transform $F_1 : \mathcal{B} \to \mathcal{B}$ on a single bit:

$$H : \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{1.6}$$

Using the Hadamard gate and the classical XOR oracle gate (along with a single auxiliary qubit), we can build the *phase-flip oracle* $O_f : \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$ given by:

$$O_f : |x\rangle \mapsto (-1)^{f(x)}|x\rangle \tag{1.7}$$

Indeed, $X_f : |x\rangle(\frac{|0\rangle-|1\rangle}{\sqrt{2}}) \mapsto (-1)^{f(x)}|x\rangle(\frac{|0\rangle-|1\rangle}{\sqrt{2}})$. Together, the Hadamard and Toffoli gates form a universal gate set for quantum circuits.

Gates with quantum control. If $U : \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$ is a unitary gate, the *controlled-U* gate $\Lambda(U) : \mathcal{B} \otimes \mathcal{B}^{\otimes n} \to \mathcal{B} \otimes \mathcal{B}^{\otimes n}$ is the operator:

$$\Lambda(U) : |c\rangle|\psi\rangle \mapsto |c\rangle U^c|\psi\rangle \tag{1.8}$$

A *measuring operator* $V : \mathcal{H} \otimes \mathcal{H}' \to \mathcal{H} \otimes \mathcal{H}'$ generalizes this idea as follows: Let $U_j : \mathcal{H} \to \mathcal{H}$ be a collection of unitary operators and $P_j : \mathcal{H}' \to \mathcal{H}'$ be a complete system of orthogonal projectors. Then the unitary operator

$$V = \sum_j P_j \otimes U_j \tag{1.9}$$

is called a measuring operator. The controlled-U gate is recovered by choosing $P_0 = |0\rangle\langle 0|$, $P_1 = |1\rangle\langle 1|$, $U_0 = I$, and $U_1 = U$.

The quantum Fourier transform. The n-qubit Fourier transform (1.1) can be expressed in the tensor product form

$$|x_1\rangle|x_2\rangle \cdots |x_n\rangle \mapsto |\gamma_n\rangle \cdots |\gamma_2\rangle|\gamma_1\rangle \tag{1.10}$$

where $x = x_1x_2 \cdots x_n \in \mathbb{B}^n$ and:

$$|\gamma_k\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{2\pi i(0.x_k \cdots x_n)}|1\rangle) \tag{1.11}$$

It is not hard to see that up to a reverse-ordering of the n qubits, this is precisely the map computed by the quantum circuit depicted in Figure 1.2, in which boxes labeled H

and R_k denote Hadamard and $R(2\pi/2^k)$ gates, respectively, and vertical lines indicate quantum control on the auxiliary qubit.

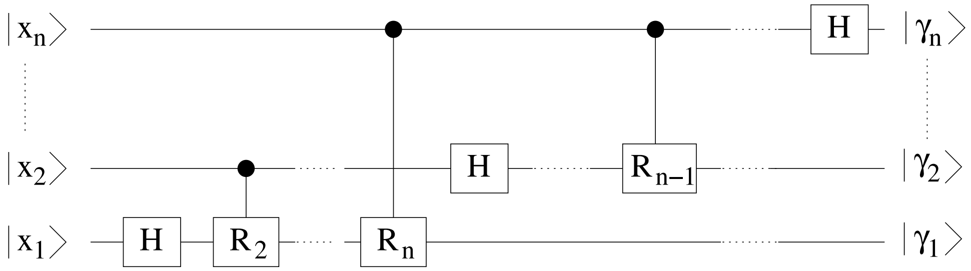

Figure 1.2: Quantum circuit for the n-qubit Fourier transform.

1.3 Phase estimation and factoring

Phase estimation is a powerful and widely applicable procedure for obtaining information about the eigenvalues of a unitary operator. Here we present the quantum circuit for phase estimation and show how it can be used on a reversible classical circuit to solve the factoring problem. Later we will use it to estimate the eigenvalues of more general unitary operators.

1.3.1 The phase estimation procedure

Let $U : \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$ be a unitary operator with spectrum $\{e^{2\pi i\theta_k}, |\phi_k\rangle\}$. Generalizing Kitaev [67], Cleve et al. [38] show:

Theorem 1.9 *There is a uniform quantum circuit*

$$Q(U) : |0^m\rangle|\phi_k\rangle \mapsto |\bar{\theta}_k\rangle|\phi_k\rangle, \tag{1.12}$$

accessing the black-box operators $U, U^2, \ldots, U^{2^{m-1}}$ such that: (a) if θ_k can be expressed exactly using an n-bit binary expansion, then $|\bar{\theta}_k\rangle$ will yield precisely this value when measured in the classical basis, and (b) if not, then measuring $|\bar{\theta}_k\rangle$ still yields the best m-bit approximation to θ_k with probability at least $4/\pi^2$.

Proof: The quantum circuit is depicted in Figure 1.3. The box labeled F_n^{-1} denotes the inverse Fourier transform – i.e., the quantum circuit from Figure 1.2 run in reverse.

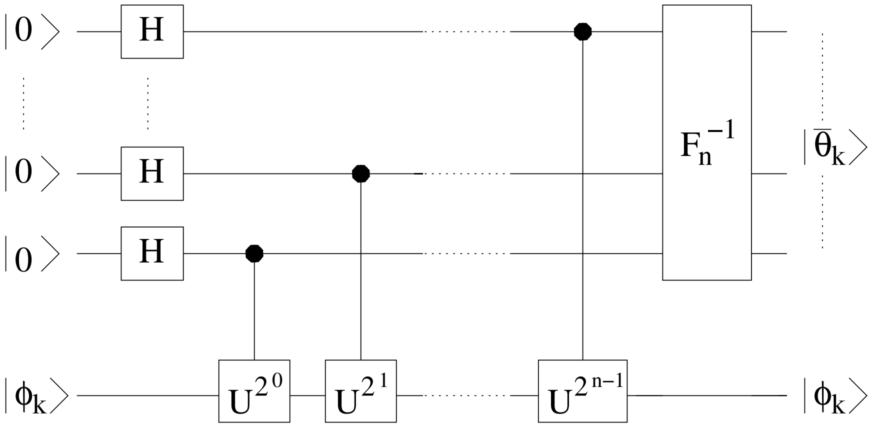

Figure 1.3: Quantum circuit for the phase estimation procedure.

By inspection, the first half of the circuit obtained by excluding F_n^{-1} maps the all-zero state $|0^n\rangle$ to the tensor product state:

$$\frac{1}{\sqrt{2^n}}(|0\rangle + e^{2\pi i 2^{n-1}\theta_k}|1\rangle)(|0\rangle + e^{2\pi i 2^{n-2}\theta_k}|1\rangle)(|0\rangle + e^{2\pi i 2^0\theta_k}|1\rangle) \tag{1.13}$$

If θ_k has an n-bit binary expansion $0.x_1x_2\cdots x_n$, then $2^j\theta_k = 0.x_{j+1}\cdots x_n$, in which case (1.13) simplifies to the right hand side of (1.10). This is mapped by F_n^{-1} to the state $|\bar{\theta}_k\rangle = |x\rangle$, proving claim (a). We prove claim (b) as in [38]. Suppose that $\frac{x}{2^n} = 0.x_1x_2\cdots x_n$ is the best n-bit estimate of θ_k, and let $\delta \in [-\frac{1}{2^{n+1}}, \frac{1}{2^{n+1}}]$ be the difference $|\theta_k - \frac{1}{2^n}|$. Rewrite (1.13) as $\frac{1}{\sqrt{2^n}}\sum_{y=0}^{2^n-1} e^{2\pi i\theta_k y}|y\rangle$. This state is mapped by F_n^{-1} to the state:

$$|\bar{\theta}_k\rangle = \frac{1}{\sqrt{2^n}}\sum_{z=0}^{2^n-1}\sum_{y=0}^{2^n-1} e^{\frac{-2\pi i z y}{2^n}} e^{2\pi i\theta_k y}|z\rangle = \frac{1}{\sqrt{2^n}}\sum_{z=0}^{2^n-1}\sum_{y=0}^{2^n-1} e^{\frac{2\pi i(x-z)y}{2^n}} e^{2\pi i\delta y}|z\rangle \tag{1.14}$$

The amplitude $\langle\bar{\theta}_k|x\rangle$ is the geometric series

$$\frac{1}{2^n}\sum_{y=0}^{2^n-1}(e^{2\pi i\delta})^y = \frac{1}{2^n}\left(\frac{1-(e^{2\pi i\delta})^{2^m}}{1-e^{2\pi i\delta}}\right) \tag{1.15}$$

A few easy calculations show the observation probability $|\langle\bar{\theta}_k|x\rangle|^2$ is at most $4/\pi^2$. ■

1.3.2 Factoring is in BQP

Once we have phase estimation in our arsenal, it is a relatively quick task to show that the factoring problem is in BQP. So let us do this now.

The *factoring problem* is to decompose an n-bit integer N into its prime factors. Both the predicate "Is there a prime factor less than m?" and its negation are in NP, which suggests that it is not NP-complete. Moreover, testing N for primality can be done in P [3]. Nevertheless, simply factoring the product of two large primes seems to be a computationally intractable problem.

The order finding problem. A problem closely related to factoring (as we shall see shortly) is *order finding*. The problem is to find the minimum positive integer r for which $a^r \bmod N = 1$, where N is an n-bit number and $a < N$ is coprime to N.

Theorem 1.10 *The order finding problem is in BQP.*

Proof: Define the permutation:

$$U : |x\rangle \mapsto |ax \bmod N\rangle \tag{1.16}$$

If the order of a mod N is r, then the spectrum $\{\lambda_k, |\phi_k\rangle\}$ of U is:

$$\lambda_k = e^{2\pi ik/r} \qquad |\phi_k\rangle = \frac{1}{\sqrt{r}} \sum_{j=0}^{r-1} e^{-2\pi ikj/r} |a^j \bmod N\rangle \tag{1.17}$$

The algorithm is as follows: Prepare the classical state $|1\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} |\lambda_k\rangle$, apply the phase estimation procedure to U, and measure the first register. By estimating several random eigenvalues of U to within $1/2r^2$, we can recover the fraction k/r – whose simplified denominator is a factor of r – using the classical method of continued fractions. To obtain this degree of precision using phase estimation, we need to construct the unitary operators U^{2^j} for $j = 0, 1, \ldots, O(\log r) = O(n)$. Fortunately, this modular exponentiation can be done efficiently (with j multiplications) by repeating squaring.■

Reducing factoring to order finding. There is an efficient randomized reduction from the factoring problem to the order finding problem [78].

Theorem 1.11 *There is a polynomial-time randomized reduction from factoring to order finding.*

Proof: To factor N odd, generate a number x uniformly at random modulo N and find its order r. Consider that:

$$(x^{r/2} - 1)(x^{r/2} + 1) = x^r - 1 = 0 \bmod N \tag{1.18}$$

If r is even and $x^{r/2} \neq -1 \bmod N$, then the greatest common divisor of $(x^{r/2} - 1)$ and N – which is efficiently computable by Euclid's algorithm – is a nontrivial divisor of N. In particular, if N has at least two distinct odd prime factors, then for at least half of the numbers x modulo N, r is even and $x^{r/2} \neq -1 \bmod N$. ■

Therefore, the factoring problem is in BQP. If factoring is as hard as we suspect for classical computers, then it seems that quantum computers (if physically constructible) are more powerful than classical computers – in particular, that BPP $\neq$ BQP, violating Postulate 1.8.

Chapter 2

Quantum Hamiltonians

Quantum mechanics is traditionally formulated in continuous time using the *Hamiltonian* framework. We use this chapter to review two basic concepts relevant to quantum computing – Schrödinger's equation and the adiabatic theorem – and to show how a quantum computer can be used to simulate or estimate the *ground state energy* of a quantum Hamiltonian.

2.1 Continuous-time quantum mechanics

Postulate 1.2 states that a closed quantum system evolves by a unitary process, but it does not specify how such a process is generated in continuous time. This is our present task.

2.1.1 Schrödinger's equation

Erwin Schrödinger proposed in 1925 the following equation governing the dynamics of pure quantum states:

$$\hbar \frac{d}{dt}|\psi(t)\rangle = -iH(t)|\psi(t)\rangle \tag{2.1}$$

Let us assume that *Planck's constant* $\hbar = 1$ in the underlying physical units. As long as the *Hamiltonian* H is a Hermitian operator, the process is unitary (and vice versa). In particular, if H is time-independent then Schrödinger's equation is easily solved:

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle \tag{2.2}$$

The time-dependent version is rarely this easy, although many clever techniques have been developed for special cases. One such technique, the adiabatic theorem, will be discussed shortly.

The Hamiltonian is more than just the driver of Schrödinger's equation. It is also an observable (recall Postulate 1.3) – in particular, it is the *energy* observable. The minimum eigenvalue λ_1 of H is the *ground state energy*, and an eigenvector of λ_1 is a *ground state*. If λ_1 is a repeated eigenvalue, then the subspace of ground states is *degenerate*. An eigenvector of $\lambda_j > \lambda_1$ is an *excited state*. We define the *spectral gap* of H by $\min_{j \neq 1} \lambda_j - \lambda_1$, and the *spectral gap above the ground space* by $\min_{j : \lambda_j \neq \lambda_1} \lambda_j - \lambda_1$. A *classical Hamiltonian* is one whose off-diagonal elements are zero, so its eigenstates are classical states.

2.1.2 The adiabatic theorem

The original *adiabatic theorem* of quantum mechanics, proved by Max Born and V.A. Fock in 1928, says the following:

> *A quantum system remains in its ground state if perturbed slowly enough by a Hamiltonian with nonzero spectral gap.*

Let $H(s)$ be a time-dependent Hamiltonian (the "adiabatic path") parametrized by $s \in [0,1]$, and denote by $|\phi(s)\rangle$ and $\delta(s)$ its ground state and spectral gap, respectively. Let $|\psi(s)\rangle$ be the system state and $\tau(s)$ be the *delay schedule*, or the rate (in real time) at which $H(s)$ is applied to $|\psi(s)\rangle$. Reparametrized, Schrödinger's equation reads:

$$\frac{d}{ds}|\psi(s)\rangle = -i\tau(s)H(s)|\psi(s)\rangle \tag{2.3}$$

In particular, the Hamiltonian $H(s)$ evolves from $H(0)$ to $H(1)$ in time $\int_{s=0}^{1} \tau(s)ds$. The folklore *quantitative* version of the adiabatic theorem (specifying what "slowly enough" means) says the following.

Theorem 2.1 *If $|\psi(0)\rangle = |\phi(0)\rangle$, then $|\psi(1)\rangle \approx |\phi(1)\rangle$ provided that:*

$$\tau(s) \approx \frac{||\frac{d}{ds}H(s)||}{\delta(s)^2} \tag{2.4}$$

Reichardt [86] proved the following rigorous formulation of Theorem 2.1 for all $k \geq 1$.

$$\tau(s) = O\left(\frac{\max_{1 \leq l \leq k+1} ||(\frac{d}{ds})^l H(s)||^{1+1/k}}{\delta(s)^{2+1/k}}\right) \tag{2.5}$$

In particular, for the linear adiabatic path $H(s) := (1-s)H_0 + sH_1$ a delay schedule of

$$\tau(s) = O\left(\frac{||H_1 - H_0||^{1+\epsilon}}{\delta(s)^{2+\epsilon}}\right) \tag{2.6}$$

works for any $\epsilon > 0$. Ambainis and Regev [21] proved a weaker statement than Reichardt's using different techniques.

2.2 Simulating Hamiltonian dynamics

The idea of using quantum computers to simulate quantum Hamiltonian dynamics goes back to Feynman [48]. Quantum simulation algorithms have been given by Farhi et al. [44], van Dam et al. [108], Childs et al. [33], Aharonov and Ta-Shma [7], and Childs [32]. Here we summarize a recent algorithm of Berry et al. [27] for simulating time-independent Hamiltonian dynamics.

Let $H : \mathcal{H} \to \mathcal{H}$ be a time-independent Hamiltonian. We wish to *simulate* H for time t – i.e., to approximate $U = e^{-iHt}$ by a quantum circuit U' with precision ϵ in the sense of (1.2). Let $\mathcal{H}$ have dimension at most 2^n, so its states can be represented using n qubits. We use n as the parameter with respect to which efficiency is measured – i.e., the simulation algorithm is *efficient* if and only if its time complexity is polynomial in n, ϵ^{-1}, and $\tau := ||H||t$.

Let H be such that each column has at most d nonzero entries. If d is polynomial in n, we say that H is *sparse*. We shall assume that the nonzero entries of a column $H(\cdot, j)$ are accessed by querying an oracle with the column label j.[1] Let $\log^*$ denote the iterated logarithm. Berry et al. [27] prove the following.

Theorem 2.2 *For any integer $k \geq 1$, there exists a uniform quantum circuit U' approximating $U = e^{-iHt}$ for $t = \tau/||H||$ with query complexity:*

$$O((\log^* n)d^2 5^{2k}(d^2\tau)^{1+1/2k}/\epsilon^{1/2k}) \tag{2.7}$$

Proof: The details are considerable, but the high-level ideas are not hard to convey. First, it is shown that H can be decomposed into a sum of $m = 6d^2$ "local" Hamiltonians

[1] We can think of this oracle as generating an adjacency list representation of H.

H_j, each with at most one nonzero entry per column. Each e^{-iH_jt} can be simulated using two queries to H_j [33], and each query to H_j can be simulated by $O(\log^* n)$ queries to H.[2]

The problem is thus reduced to simulating the Hamiltonian $H = \sum_{j=1}^m H_j$ by the local terms H_j. This is nontrivial because $e^{H_j+H_k} \neq e^{H_j}e^{H_k}$ unless H_j and H_k commute. The insight of [27] is to use the higher order *Suzuki recursion*

$$S_{2k}(\lambda) = [S_{2k-2}(p_k\lambda)]^2 S_{2k-2}((1-4p_k)\lambda)[S_{2k-2}(p_k\lambda)]^2 \tag{2.8}$$

where S_2 is the basic *Lie-Trotter formula* used to approximate $\sum_{j=1}^m H_j$ by interleaving the H_j in a round-robin fashion:

$$S_2(\lambda) = \prod_{j=1}^{m} e^{H_j\lambda/2} \prod_{j'=m}^{1} e^{H_{j'}\lambda/2} \tag{2.9}$$

In particular, they show that

$$||e^{-i\sum_{j=1}^m H_jt} - [S_{2k}(-it/r)]^r|| \leq \epsilon \tag{2.10}$$

provided that we choose $r = \lceil 4 \cdot 5^{k-1/2}(m\tau)^{1+1/2k}/\epsilon^{1/2k}\rceil$. The total number of exponentials $e^{-iH_jt/r}$ needed for the simulation is at most

$$2m5^{k-1}r \leq 2m5^{2k}(m\tau)^{1+1/2k}/\epsilon^{1/2k} \tag{2.11}$$

and the theorem follows. ■

A simple corollary is that H is efficiently simulable if it is sparse [7].

Suppose instead that we wish to simulate time-dependent dynamics along the linear adiabatic path $H(s) = (1-s)H_0 + sH_1$, $s \in [0,1]$. If H_0 and H_1 are efficiently simulable, then so is the entire path $H(s)$ [44, 108, 7]. The idea is to approximate $H(s)$ as a sequence of Hamiltonians $H(s_1), H(s_2), \ldots, H(s_m)$, then to simulate the time-independent dynamics of each of the $H(s_j)$ using Theorem 2.2.

[2] If we wish to measure time complexity rather than query complexity, then an additional factor $O(n\log^* n)$ is necessary here [27].

2.3 Estimating the ground state energy

We can use phase estimation (Theorem 1.9) and Hamiltonian simulation (Theorem 2.2) to verify that a sparse Hamiltonian has ground state energy below some threshold, given a copy of the ground state as a (quantum) witness.

Theorem 2.3 *There is an efficient quantum algorithm verifying that a sparse Hamiltonian H of dimension at most 2^n and spectral norm polynomial in n has ground state energy at most a, where a is specified to $O(\log n)$ bits of precision.*

Proof: Since H is sparse, it is efficiently simulable to time $t = \tau/||H||$ polynomial in n. In particular, there are efficient quantum circuits approximating the unitary evolutions e^{iH2^j} for each $j = 0, 1, \ldots, s = O(\log n)$. Thus, given an eigenstate of e^{iH}, the corresponding eigenvalue can be computed efficiently to $O(\log n)$ bits of precision using phase estimation. By rescaling H if necessary, we may assume that its spectrum $\{\lambda_k, |\phi_k\rangle\}$ satisfies $\pi/2 \leq \lambda_k \leq \pi/2$. Then the spectrum $\{e^{i\lambda_k}, |\phi_k\rangle\}$ of e^{iH} preserves the order of eigenvalues of H (mapping them from the real line to the unit circle), and the ground state $|\phi_1\rangle$ of H is a witness that the phase of the corresponding eigenvalue $e^{i\lambda_1}$ of e^{iH} (i.e., the ground state energy λ_1 of H) is at most a. ■

It is not necessary for the witness $|\xi\rangle$ to be the ground state $|\phi_1\rangle$, only "near" a "low-energy" state in the following sense: there should exist a state $|\psi\rangle$ of energy at most a such that the quantity $|\langle\psi|\xi\rangle|^2$ is at least inverse-polynomial in n.

Part II

Quantum walks and their applications

Chapter 3

Quantum search algorithms

The study of quantum walk algorithms begins with *Grover's algorithm* and *amplitude amplification.* While these are not quantum walk algorithms strictly speaking, they can be interpreted as such. In this chapter, we review these two algorithms and their applications.

3.1 Grover's algorithm

Grover's search algorithm [52] locates a marked item from a database of N items using $O(\sqrt{N})$ quantum queries to the database. It is trivial to prove that a classical randomized algorithm can do no better than $\Theta(N)$. Next to Shor's factoring algorithm, Grover's algorithm is the most well-known quantum algorithm. Though its quantum speedup is not exponential like Shor's, its applications are more numerous.

3.1.1 The quantum search framework

Grover's algorithm solves the search problem.

Definition 3.1 *Given oracle access to a predicate* $f : \mathbb{B}^n \to \mathbb{B}$*, the* search *problem is to find an* $x \in f^{-1}(1)$ *if one exists. The corresponding* decision *problem is to determine whether* $|f^{-1}(1)| > 0$*.*

We may assume that f is presented to us by the phase flip oracle (1.7), which we rewrite as:

$$O_f = I - 2 \sum_{x \in f^{-1}(1)} |x\rangle\langle x|. \tag{3.1}$$

Let $N = 2^n$. Define the uniform superposition $|u\rangle \in \mathcal{B}^{\otimes n}$ by

$$|u\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{B}^n} |x\rangle \tag{3.2}$$

and the *Grover diffusion operator* $G : \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$ by

$$G := 2|u\rangle\langle u| - I. \tag{3.3}$$

Both $|u\rangle$ and G are easily constructed:

$$|u\rangle = H^{\otimes n}|0^n\rangle \qquad G = H^{\otimes n}(2|0^n\rangle\langle 0^n| - I)H^{\otimes n} \tag{3.4}$$

Grover's algorithm is the circuit

$$(GO_f)^k|u\rangle \tag{3.5}$$

for k appropriately chosen. The proof that this solves the search problem with $k = O(\sqrt{N})$ is simple and elegant.

3.1.2 Proof of the quadratic search speedup

Let $M = |f^{-1}(1)|$. We show:

Theorem 3.2 *If $M > 0$, then Grover's algorithm finds an $x \in f^{-1}(1)$ using $O(\sqrt{N/M})$ quantum queries.*

Proof: Define the following orthogonal states:

$$|\alpha\rangle := \frac{1}{\sqrt{N-M}} \sum_{x \in f^{-1}(0)} |x\rangle \qquad |\beta\rangle := \frac{1}{\sqrt{M}} \sum_{x \in f^{-1}(1)} |x\rangle \tag{3.6}$$

Let $\mathcal{L} := \text{span}\{|\alpha\rangle, |\beta\rangle\}$. It is easy to check that $\mathcal{L}$ is an invariant space for both G and O_f, and that $|u\rangle \in \mathcal{L}$. Therefore, the action of Grover's algorithm lies entirely within the two-dimensional subspace $\mathcal{L}$. Geometrically, $G|_{\mathcal{L}}$ and $O_f|_{\mathcal{L}}$ are *reflections* about the vectors $|u\rangle$ and $|\alpha\rangle$. The product of two reflections is a rotation by twice the angle θ between the reflection axes (see Figure 3.1), so $GO_f|_{\mathcal{L}}$ is a rotation by:

$$2\theta = 2\cos^{-1}\langle\alpha|u\rangle = 2\sin^{-1}\langle\beta|u\rangle = 2\sin^{-1}\sqrt{M/N} \tag{3.7}$$

For $M \ll N$, $2\sin^{-1}\sqrt{M/N} \approx 2\sqrt{M/N}$. It follows that the state $(GO_f)^k|u\rangle$ has constant inner product with $|\beta\rangle$ for $k := \lceil \frac{\pi}{4}\sqrt{N/M}\rceil$. ■

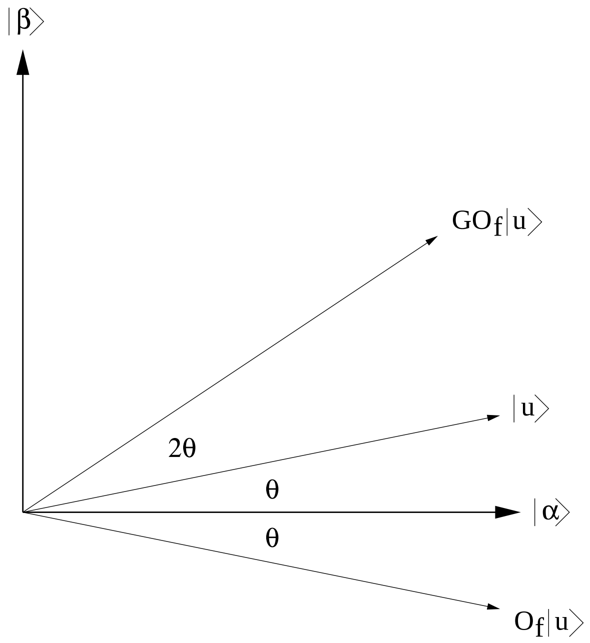

Figure 3.1: The product of two reflections in Grover's algorithm.

The quadratic search speedup of Grover's algorithm matches the lower bound proved earlier by Bennett et al. [25]. Thus, there is no efficient "brute-force" quantum algorithm for solving NP-complete problems.

Note that the eigenvalues $e^{\pm 2i\theta}$ of $GO_f|_{\mathcal{L}}$ reveal the number of solutions M to the search problem. Using phase estimation, we can approximate M more efficiently than possible classically [82].

3.2 Amplitude amplification

Suppose we have a classical randomized algorithm R_f computing a predicate $f : \mathbb{B}^n \to \mathbb{B}$ with *one-sided error*, meaning that R_f outputs 0 with certainty if $x \in f^{-1}(0)$ and 1 with *success probability* at least $\epsilon > 0$ if $x \in f^{-1}(1)$. By repeating R_f $O(1/\epsilon)$ times and outputting 1 if and only if any of the trials do, we can obtain an algorithm computing f with success probability at least 2/3.

Now suppose we have a quantum algorithm $A_f : \mathcal{B}^k \to \mathcal{B}^k$ computing f with one-sided error and success probability ϵ. Brassard et al. [28] show:

Theorem 3.3 *There is a quantum algorithm computing f with success probability at* *least 2/3 that queries A_f $O(1/\sqrt{\epsilon})$ times.*

Proof: Assume for simplicity that A_f maps the input register $|x\rangle$ and the (initially zero) work register $|0^{k-n}\rangle$ to the state

$$A_f|x\rangle|0^{k-n}\rangle = \cos\theta_x|\psi_{x,0}\rangle|0\rangle + \sin\theta_x|\psi_{x,1}\rangle|1\rangle \tag{3.8}$$

whose final qubit contains the “answer” to $f(x)$ – in particular, $\sin^2\theta_x$ is ϵ if $f(x) = 1$ and 0 if $f(x) = 0$. Define the reflection operators

$$G' = A_f(2|x, 0^{k-n}\rangle\langle x, 0^{k-n}| - I)A_f^{-1} \qquad O' = I - 2|\psi_{x,1}, 1\rangle\langle\psi_{x,1}, 1| \tag{3.9}$$

as analogues of the Grover diffusion operator (3.3) and the phase flip oracle (3.1), respectively. The *amplitude amplification* circuit is given by:

$$(G'O')^k|x, 0^{k-n}\rangle \tag{3.10}$$

The analysis is similar to that of Grover's algorithm. The algorithm is confined to the two-dimensional subspace $\mathcal{L} := \text{span}\{|\psi_{x,0}, 0\rangle, |\psi_{x,1}, 1\rangle\}$, and $G'|_{\mathcal{L}}$ and $O'|_{\mathcal{L}}$ are reflections about the axes $A_f|x, 0^{k-n}\rangle$ and $|\psi_{x,0}, 0\rangle$, respectively. Since these axes are separated by an angle θ_x, $G'O'|_{\mathcal{L}}$ is a rotation by

$$2\theta_x = 2\sin^{-1}\sqrt{\epsilon} \tag{3.11}$$

which for $\epsilon \ll 1$ is approximately $2\sqrt{\epsilon}$. It follows that the state $(G'0')^k|x, 0^{k-n}\rangle$ outputs 1 in the answer register with constant probability for $k = \lceil\frac{\pi}{4}\sqrt{1/\epsilon}\rceil$. ■

Høyer et al. [54] show that the success probabilities of quantum algorithms with *two-sided error* can also be amplified more efficiently than possible classically. The proof is more intricate than for one-sided error.

3.3 Algorithmic applications

Applications of Grover's algorithm and amplitude amplification in quantum algorithms are ubiquitous. We highlight just a few.

Constraint satisfaction. In a *constraint satisfaction problem* we are given n variables $\{x_i\}$, each of which can take any value from $[d]$, and m constraints (or clauses) $\{c_j\}$, each involving at most k of the n variables. The goal is to find an assignment $\{x_i\} \in [d]^n$ that simultaneously satisfies all m constraints. A well-known example is 3-SAT, where $d = 2$, $k = 3$, and the clauses are Boolean formulae composed of logical OR and negation operations. Most constraint satisfaction problems (including 3-SAT) are NP-complete. Grover's algorithm can be used to search for a satisfying assignment with a quadratic speedup over classical brute-force search through the space of assignments. However, brute-force search is not the optimal classical algorithm: Schöning [96] gave a randomized algorithm that in polynomial time finds a satisfying assignment with probability $\Omega(1/(d(1-1/k))^n)$. Using amplitude amplification rather than classical probability amplification, we can speed up Schöning's algorithm quadratically (Ambainis [15]).

Element distinctness. The *element distinctness* problem is to determine whether the elements $\{x_1, x_2, \ldots, x_n\}$ are distinct, or equivalently to decide whether a function on n variables is injective, using queries of the form "What is the value of x_i?" The classical query complexity is $\Theta(n)$. Aaronson and Shi [2] showed that the quantum query complexity is $\Omega(n^{2/3})$. Burhman et al. [29] gave the following quantum algorithm using $O(n^{3/4})$ queries. Select a subset $S \subseteq [n]$ of size $\sqrt{n}$ uniformly at random and query each x_i for $i \in S$, outputting "not distinct" if any two are equal. Then use Grover's algorithm to search for an index $j \notin S$ such that some $i \in S$ satisfies $x_i = x_j$, outputting "not distinct" if one is found. This algorithm uses $O(\sqrt{N})$ queries to either output "not distinct" or determine that the elements of S are distinct with each other and with $[n] \setminus S$. Since S has at least a $1/\sqrt{n}$ chance of intersecting any pair $x_i = x_j$ if one exists, repeating the algorithm $O(n^{1/4})$ times using amplitude amplification yields an $O(n^{3/4})$ algorithm for element distinctness.

Triangle finding. The *triangle finding* problem is to decide whether an n-vertex m-edge graph $G = (V, E)$ contains a *triangle* (clique of size three) using queries of the form “Are vertices u and v connected by an edge?” The classical query complexity is $\Theta(n^2)$. Searching the space of all potential triangles $\{(u,v),(v,w),(w,u) : u,v,w \in V\}$ by Grover’s algorithm uses $O(n^{3/2})$ queries. Burhman et al. [29] gave the following algorithm which is more efficient if G is *sparse* – i.e., if $m = o(n)$.[1] Use Grover’s algorithm to find (a) an edge $(u,v) \in E$ followed by (b) a vertex $w \in V$ such that $\{u,v,w\}$ is a triangle. The costs of steps (a) and (b) are $O(\sqrt{n^2/m})$ and $O(\sqrt{n})$ queries, respectively. If G contains a triangle Δ, then step (a) will find an edge (u,v) from Δ with probability $\Omega(1/m)$, and step (b) will find the third vertex with constant probability. By repeating the steps $O(\sqrt{m})$ times using amplitude amplification, a triangle is found with constant probability. The total cost in queries is $O(\sqrt{m}(\sqrt{n^2/m}+\sqrt{n})) = O(n+\sqrt{nm})$. Szegedy [104] gave a more involved algorithm based on amplitude amplification improving this to $\tilde{O}(n^{10/7})$.[2] For details, see the short survey [89].

Matrix product verification. The *matrix product verification* problem is to determine whether three $n \times n$ matrices A, B, C satisfy $AB = C$ using (adjacency matrix) queries of the form “What is the value of $A(i,j)$?” The classical time complexity is $\Theta(n^2)$ by Freivalds’ algorithm [50]: check whether $A(Bx) = Cx$ for a random vector x and output “equal” or “not equal” based on whether the test passes or fails. Ambainis et al. [18] used this idea to give the following quantum algorithm with query complexity $O(n^{7/4})$. Select a subset of column indices $S \subseteq [n]$ of size $\sqrt{n}$ uniformly at random. Compute $y = B|_S x$ and $z = C|_S x$ for a random vector x of length $\sqrt{n}$, and use Grover’s algorithm to search for a j such that $(Ay)_j \neq z_j$. Both of these steps can be done using $O(n^{3/2})$ queries. This gives a verification test with success probability $1/\sqrt{n}$, so repeating it $O(n^{1/4})$ times gives an algorithm succeeding with constant probability in $O(n^{7/4})$ queries. A lower bound of $\Omega(n^{3/2})$ queries can be proved by reducing the matrix verification problem to Boolean formula evaluation [18].

[1] Recall that $f(n) = o(g(n))$ if $f(n) = O(g(n))$ but $f(n) \neq \Theta(g(n))$.

[2] The notation $\tilde{O}(\cdot)$ is used instead of $O(\cdot)$ to suppress a constant number of logarithmic factors.

Chapter 4

Markov chains and quantum walks

A *Markov chain* is a stochastic process with time-independent dynamics. The *fundamental theorem of Markov chains* tells us that an ergodic Markov chain converges to a unique *stationary* (equilibrium) distribution π. Interpreted in the setting of statistical physics where it is postulated that all microstates of a closed system at equilibrium are equally likely – i.e., that π is the (maximum-entropy[1]) uniform distribution – the fundamental theorem is equivalent to the *second law of thermodynamics*:

> *The entropy of an isolated system tends to a maximum.*

In contrast, a unitary process with time-independent dynamics, or *quantum walk*, cannot converge to a fixed wavefunction. The complex behavior of quantum walks is already apparent in one dimension: Figure 4.1 superimposes plots of the probability distributions induced by the one-dimensional random and quantum walks. In this chapter, we review the theory of classical Markov chains and their algorithmic applications, then begin the theory of quantum walks by presenting the standard quantum walk constructions.

4.1 A quick review of Markov chains

There are many technical concepts in the basic theory of Markov chains, several of which play a role in the transition from Markov chains to quantum walks. We review them here. For more information on Markov chain theory, see Lovász [71] or Aldous and Fill [12].

[1] Recall that the *entropy* of a distribution p is $H(p) := -\sum_{x\in\mathcal{S}} p(x)\log p(x)$. In particular, the uniform distribution has entropy $\log|\mathcal{S}|$.

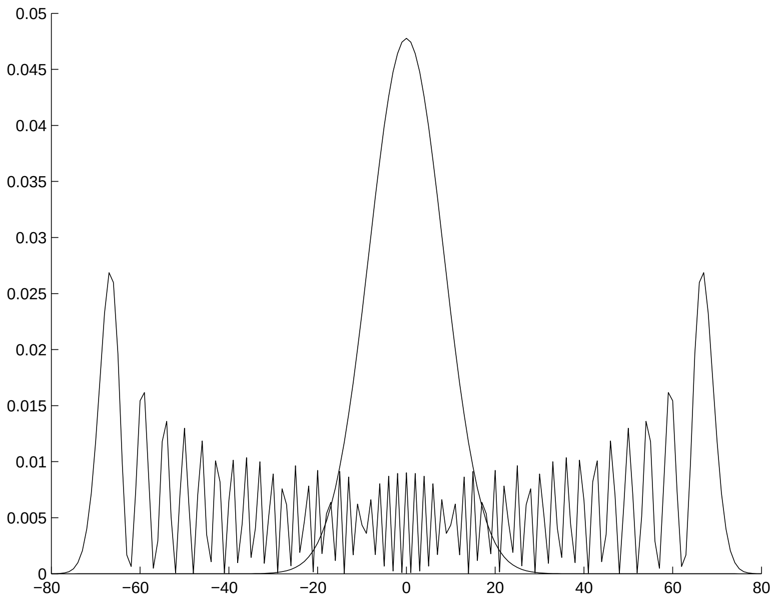

Figure 4.1: The behavior of random vs. quantum walks in one dimension.

4.1.1 Definitions and simple examples

Let $\mathcal{S}$ be a countable set of *states*. Unless otherwise stated, we take $N := |\mathcal{S}| < \infty$. An $\mathcal{S} \times \mathcal{S}$ matrix P is a *transition probability matrix* if it is column-stochastic; i.e., if each of its columns is a distribution (probability vector), or equivalently, if it preserves l_1 norm of nonnegative vectors. Let q be a distribution on $\mathcal{S}$. The sequence of random variables $(X_t)_{t\in[0..\infty)}$ distributed according to $P^t q$ is a *discrete-time Markov chain.* More frequently, we use the term *Markov chain* to refer to any valid transition probability matrix P, without specifying q. The family of random variables $(Y_t)_{t\in[0,\infty)}$ distributed according to $e^{-Qt}q$ is a *continuous-time Markov chain* with *transition rate matrix* $Q := I - P$.

Random walks on graphs and groups. We can think of any Markov chain as a *random walk* on a graph or digraph. The *simple random walk* on a graph $G = (V, E)$ is the Markov chain $P = A\Delta^{-1}$, where A is the adjacency matrix of G and Δ is the diagonal matrix of vertex degrees. We call G the graph *underlying* P. If G is a (nonnegatively) weighted graph, we define the *random walk on* G by $P = A\Delta^{-1}$ but with A now the weighted adjacency matrix of G and Δ the diagonal matrix with

$\Delta(x,x) = \sum_y A(y,x)$. If G is a digraph, the same construction again yields a Markov chain, and *any* Markov chain can be realized this way. However, we will see shortly that a nice spectral relation between A and P exists for walks on graphs but not on digraphs.

A well-studied class of Markov chains is random walks on groups. Let $X = \{g_1, \ldots, g_d\}$ be a set of distinct generators for some finite group H. The Markov chain with state space H whose transitions from an element $x \in H$ are $x \to xg_i$ with probability $1/d$ is denoted by $H[X]$. We interpret $H[X]$ as a random walk on the Cayley graph $\Gamma(H, X)$, which is undirected if and only if $g_i \in X \Leftrightarrow g_i^{-1} \in X$. Often X is clear from context and we refer only to the simple random walk on H. Examples include the infinite line $H = \mathbb{Z}$ with $X = \{\pm 1\}$, the hypercube $H = \mathbb{Z}_2^n$ with $X = \{e_i : 1 \leq i \leq n\}$, and the torus $H = \mathbb{Z}_n^d$ with $X = \{\pm e_i : 1 \leq i \leq d\}$.

Irreducibility and reversibility. A *stationary distribution* π for a Markov chain P is a probability vector satisfying $P\pi = \pi$. The Perron-Frobenius Theorem for nonnegative matrices guarantees the existence of a stationary distribution for any finite Markov chain. Moreover, uniqueness and strict positivity of the stationary distribution are guaranteed if P is *irreducible*; i.e., if the digraph G underlying P is strongly connected. The stationary distribution for the random walk on a graph $G = (V, E)$ is given by $\pi(x) = d(x)/2|E|$, where $d(x)$ is the degree of $x \in V$.

Let P be a Markov chain and $\pi > 0$ be a distribution satisfying the *detailed balance* condition $P(y,x)\pi(x) = P(x,y)\pi(y)$; or equivalently, suppose that the matrix

$$M(P) := \sqrt{R}^{-1} P \sqrt{R} \qquad \text{where } R := diag(\pi) \tag{4.1}$$

is symmetric. Then P is *reversible* – from π (necessarily its stationary distribution), it appears the same run forward or backward in time.[2] Moreover, the fact that it is similar to a real symmetric matrix implies that it has a complete system of orthogonal eigenvectors $\{v_k\}_{k=1}^N$ and real eigenvectors $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_N \geq -1$. For example, a Markov chain P that is *symmetric* (i.e., P equals its transpose) is reversible

[2] This notion of reversibility is not to be confused with the reversibility property of unitary processes.

with uniform stationary stationary distribution and satisfies $M(P) = P$.[3] The random walk on a weighted graph from earlier is also reversible; in fact, any reversible Markov chain can be realized as the random walk on a weighted graph.

A particularly famous example of a reversible Markov chain is the *Metropolis process*. Suppose we wish to sample from an arbitrary distribution $\pi > 0$ on a set $\mathcal{S}$ and are able to construct a single irreducible Markov chain P on $\mathcal{S}$. When $\mathcal{S}$ is a set of combinatorial objects (e.g., microstates in statistical physics or data structures in computer science), one typically obtains P by a local update rule (e.g., change a single spin or flip a single edge), called the *Glauber dynamics* in statistical physics. We obtain the Metropolis process P' using P as follows: at each state $x \in \mathcal{S}$, select an adjacent state y with probability $P(y, x)$ and move to y with probability $\min\{1, \frac{P(x,y)}{P(y,x)} \cdot \frac{\pi(y)}{\pi(x)}\}$; otherwise, stay put. We will see some applications of the Metropolis process in the next section.

4.1.2 Mixing and hitting times

The two Markov chain parameters most crucial to algorithmic applications are the mixing time and the hitting time.

Ergodic Markov chains. Under what circumstances does an irreducible Markov chain converge to its stationary distribution π from any initial distribution? Equivalently, when does it approach the limit matrix $P^\infty := [\pi\pi\cdots\pi]$? In the case of a continuous-time Markov chain, the answer is *always*:

$$e^{-Qt} = \sum_{s=0}^{\infty} \frac{e^{-t}t^s}{s!} P^s \to P^\infty \text{ as } t \to \infty \tag{4.2}$$

if and only if $P = I - Q$ is irreducible. In the discrete time case, the convergence $P^t \to P^\infty$ is guaranteed provided that P is *aperiodic*; i.e., the lengths of all closed walks in the graph G underlying P must be mutually coprime. If G is undirected, P is aperiodic precisely when G is non-bipartite. A Markov chain which is both irreducible and aperiodic is *ergodic*. The asymptotic behavior of an ergodic Markov chain is summarized by the *fundamental theorem of Markov chains*:

[3] A symmetric Markov chain appears the same run forward or backward in time from *any* distribution, not just π.

Theorem 4.1 *If P is an ergodic Markov chain, then it has a unique stationary distribution π, and $P^t \to P^\infty$ as $t \to \infty$.*

The speed at which an ergodic Markov chain P converges from an initial distribution e_x concentrated at $x \in \mathcal{S}$ to its stationary distribution π is captured by the *mixing time*

$$\tau_x := \min\{t : ||P^t e_x - \pi||_1 \leq 1/e\} \tag{4.3}$$

where $||\cdot||_1$ is the l_1 vector norm. The maximum mixing time over all initial states (or equivalently, distributions) is

$$\tau := \max_x \tau_x = \min\{t : ||P^t - P^\infty||_1 \leq 1/e\} \tag{4.4}$$

where $||\cdot||_1$ is the l_1 matrix norm. The parameter $1/e$ is somewhat arbitrary, in that we can change it to any $\epsilon > 0$ provided we multiply the mixing time by $O(\log 1/\epsilon)$.

The mixing time of a reversible Markov chain with spectrum $\{\lambda_k, v_k\}_{k=1}^N$ is closely related to its *spectral gap* $\delta := 1 - \lambda$, where $\lambda := \max\{\lambda_2, |\lambda_N|\}$ [11, 40, 99]:

Theorem 4.2 *Let P be a reversible, ergodic Markov chain with stationary distribution π and spectral gap δ. Then its mixing time satisfies (a) $\tau_x \leq \delta^{-1}(\log 1/\pi(x) + \log 2e)$, and (b) $\tau \geq \frac{1}{2}|\lambda|\delta^{-1}$.*

Often the spectral gap of a Markov chain can be estimated indirectly, for example by bounding its conductance (a geometric parameter of the chain) or the congestion of a multicommodity flow on the chain [99]. Occasionally it can be estimated directly: For example, the simple random walk on the periodic lattice (torus) $\mathbb{Z}_n^d$ is diagonalizable as $\{\frac{1}{d}\sum_{j=1}^d \cos(\frac{2\pi k_j}{n}), \frac{1}{\sqrt{n^d}}\sum_x e^{\frac{i2\pi k\cdot x}{n}}|x\rangle\}$, so its spectral gap is $\Theta(\frac{1}{dn^2})$.

Absorbing Markov chains. A Markov chain P is called *absorbing* if (a) the set $M = \{x \in \mathcal{S} : P(x,x) = 1\}$ is nonempty and (b) any $y \in \mathcal{S}$ can reach M by a directed walk in the graph underlying P. The states of M are *absorbing* and those of $\mathcal{S} \setminus M$ are *transient*. From any initial distribution, an absorbing Markov chain converges to a distribution with no support on the transient states. Define the *leaking walk matrix* P_M by deleting from P the rows and columns indexed by M. The reader may wish to verify that the convergence property just mentioned is equivalent to the following property of the leaking walk matrix:

Theorem 4.3 *Let P be an absorbing Markov chain with leaking walk matrix P_M. Then P_M^t tends to the all-zero matrix as $t \to \infty$.*

An absorbing Markov chain P' is obtained from an ergodic Markov chain P by *marking* a set of states $M \subseteq \mathcal{S}$; i.e., by replacing the transitions from each state $x \in M$ with the idle transition $P(x,x) = 1$. If q is an initial distribution on $\mathcal{S}$, q_M is obtained from q by deleting the entries indexed by M, and j is the column vector of ones, then the *hitting time*

$$h_q := \sum_{t=0}^{\infty} ||P_M^t q_M||_1 = j^\dagger (I - P_M)^{-1} q_M \tag{4.5}$$

is the expected time for P' to reach M. One can show:

Theorem 4.4 *Let P be a symmetric, ergodic Markov chain with uniform stationary distribution u and spectral gap δ. Let P' be the absorbing Markov chain formed by fixing a set of absorbing states $M \subseteq \mathcal{S}$ with $|M| = \varepsilon N$. Then the hitting time of P' satisfies $h_u \leq 1/(1 - ||P_M||_2) \leq 1/(\delta\varepsilon)$.*

The first inequality follows easily from Theorem 4.3 and the second inequality is shown in [105].

4.2 Applications of Markov chains

For many computational problems, the most efficient, elegant, or practical solution is a randomized algorithm that simulates a Markov chain. We describe four such problems. The textbooks of Motwani and Raghavan [80] and Jerrum [56] contain many more examples.

4.2.1 Local search algorithms

A *local search algorithm* explores the set of candidate solutions to a problem using local transitions based on myopic decision-making.

Graph reachability. Among the earliest Markov chain applications in computational complexity was the randomized logarithmic-space algorithm of Aleliunas et al. [13] for the *graph reachability* problem: given a graph $G = (V, E)$ and two vertices $s, t \in V$, decide whether s and t lie in the same connected component. The algorithm is almost trivial: run the simple random walk on G from s for $2|V||E|$ steps, and output true if and only if t is reached. Since the current vertex label and the walk step count fit into logarithmic-size registers, the algorithm uses only logarithmic space. If s and t do not lie in the same connected component, the random walk never reaches t and the output is correct with probability one. Otherwise, a simple hitting time calculation shows that the random walk reaches t with probability at least $1/2$. Until Reingold's recent breakthrough [87], it was not known how to solve the graph reachability problem in logarithmic space *deterministically.*

Constraint satisfaction. Markov chain algorithms are among the fastest algorithms known for solving constraint satisfaction problems. Consider the *k-satisfiability* (k-SAT) problem: given an n-variable Boolean formula ϕ with m conjunctive clauses (constraints) on k variables each, decide whether ϕ has an assignment satisfying all clauses simultaneously.

For $k = 2$, Papadimitriou [84] proposed the following algorithm for finding a satisfying assignment. Pick an initial assignment $x = x_1 x_2 \cdots x_n$, and repeat the following T times: if x does not satisfy ϕ, then select an unsatisfied clause, pick one of the two variables in the clause uniformly at random, and flip its value in the assignment x. If there is a satisfying assignment $y = y_1 y_2 \cdots y_n$, the algorithm's progress in discovering y is modeled by a random walk on the line $\{0, 1, \ldots, n\}$: the walk state is i if the Hamming distance between x and y is i, and each step either increases or decreases the walk state by one. In particular, it increases with probability at least $1/2$ since at least one of the two literals in a clause not satisfied by x must be flipped in y. Therefore, within $T = O(n^2)$ steps (the hitting time of the simple random walk on the line), the walk reaches state n and the algorithm finds y.

Papadimitriou's algorithm is not the fastest for solving the k-satisfiability problem with $k = 2$: there is a linear-time deterministic algorithm due to Aspvall et al. [23]. However, Schöning [96] showed that a similar algorithm is among the fastest for solving the (NP-complete) k-satisfiability problem with $k \geq 3$. The key observation is that running Papadimitriou's algorithm for only $T = 3n$ steps reaches y with probability at least $1/(2(1-\frac{1}{k}))^n$. Hence, repeating the algorithm $O((2(1-\frac{1}{k}))^n)$ times is enough to find y with probability at least $1/2$.

4.2.2 Simulated annealing and MCMC sampling

Simulated annealing and Markov chain Monte Carlo (MCMC) sampling are general methods for optimizing a function over and generating a random sample from a finite set, respectively. We motivate them using two examples from Sinclair [100].

Combinatorial optimization. *Simulated annealing* [66] is a Markov chain Monte Carlo method for solving combinatorial optimization problems (e.g., finding a minimum-energy configuration in statistical physics or a minimum-cost subset in computer science). The idea is as follows: Suppose we want to minimize the function $f(x)$ over $x \in \mathcal{S}$. For example, to find a perfect matching we consider $f(x) = -|x|$, where $\mathcal{S}$ is the set of matchings in a graph and $|x|$ is the size of the matching $x \in \mathcal{S}$. Connect the states $\mathcal{S}$ by any ergodic Markov chain P (e.g., a simple random walk), and define the *Gibbs distribution*

$$\pi(x) := \frac{1}{Z(\lambda)} \lambda^{-f(x)} \tag{4.6}$$

on the set $x \in \mathcal{S}$, where $\lambda \geq 0$ is the *activity* and the normalizing factor $Z(\lambda) = \sum_{x \in \mathcal{S}} \lambda^{-f(x)}$ is the *partition function.* The idea of simulated annealing is that by running the Metropolis process while gradually increasing λ from one to infinity – or rather, to a large finite value – we skew its stationary distribution π (initially uniform) toward states $x \in S$ of globally minimal energy. Note that we can compute the Metropolis transition probabilities $\frac{P(x,y)}{P(y,x)} \cdot \frac{\pi(y)}{\pi(x)} = \frac{P(x,y)}{P(y,x)} \cdot \lambda^{f(x)-f(y)}$ efficiently without evaluating the partition function Z at λ, which is key because this problem is #P-complete.[4]

[4] A problem is in #P if it counts the number of witnesses to an NP problem.

Approximate counting. The partition function Z and its derivatives reveal the *thermodynamic* (macroscopic) properties of a system, so the fact that it is typically #P-complete to evaluate is unfortunate. We may nevertheless be able to evaluate Z efficiently to within a factor $1 + \epsilon$ in polynomial time, for any fixed $\epsilon > 0$, using a randomized algorithm. This idea is called *approximate counting*, and it is solved by polynomial-time reduction to the problem of *approximate sampling*, or generating a random sample distributed very close to a target distribution [56]. *Markov chain Monte Carlo (MCMC) sampling* is the standard method for solving the approximate sampling problem. The MCMC paradigm is used to approximately evaluate the partition function of a monomer-dimer covering or ferromagnetic Ising system [57, 58], the permanent of a nonnegative matrix [59], and the volume of a convex body [41]. We illustrate the algorithm for monomer-dimer coverings.

Suppose we are given as input a graph $G = (V, E)$ and a positive value λ. Let $\mathcal{S}_k$ be the set of all k-matchings in G, where a *k-matching* is a k-subset of edges which are vertex-disjoint. A pair of vertices connected by an edge in the matching is a *dimer*; an isolated vertex is a *monomer*. Define the Gibbs distribution π on the set $x \in \mathcal{S} := \cup_k \mathcal{S}_k$ with $f(x) = -|x|$. Our computational task is to approximately evaluate the partition function $Z(\lambda)$.

Write $Z(\lambda)$ as a telescoping product

$$Z(\lambda) = \frac{Z(\lambda_r)}{Z(\lambda_{r-1})} \times \frac{Z(\lambda_{r-1})}{Z(\lambda_{r-2})} \times \cdots \times \frac{Z(\lambda_1)}{Z(\lambda_0)} \times Z(\lambda_0) \tag{4.7}$$

for some sequence $\lambda = \lambda_r > \lambda_{r-1} > \cdots > \lambda_1 > \lambda_0 = 0$. Then $Z(\lambda_0) = 1$ and $Z(\lambda_{i+1})/Z(\lambda_i) = \mathrm{E}_{x \leftarrow \pi_i}[(\lambda_{i+1}/\lambda_i)^{|x|}]$, where π_i is the Gibbs distribution corresponding to λ_i. It can be shown that r need not be very large to guarantee that the expectation values $E_{x \leftarrow \pi_i}[(\lambda_{i+1}/\lambda_i)^{|x|}]$ are bounded and well-estimated using relatively few samples from the distributions π_i. Thus we have reduced the problem of approximating $Z(\lambda)$ efficiently to the problem of sampling efficiently from the Gibbs distribution π_i.[5]

The sampling problem is solved by running the Metropolis process: at state $x \in \mathcal{S}_k$, select an edge $e = (u, v) \in E$ uniformly at random and (a) if $e \notin x$ and $x' := x \cup \{e\} \in$

[5] This is essentially the same trick used in the simulated annealing algorithm.

$\mathcal{S}_{k+1}$, goto $y := x'$; (b) if $e \notin x$ and $x' := x \cup \{e\} \notin \mathcal{S}_{k+1}$, goto the unique $y \in \mathcal{S}_k$ obtained from x' by removing the edge that conflicts with e; or (c) if $e \in x$, goto $y := x \setminus \{e\} \in \mathcal{S}_{k-1}$ with probability $\pi_i(y)/\pi_i(x) = 1/\lambda_i$. Jerrum and Sinclair [57] showed that the spectral gap of this Markov chain is $\Omega(\lambda_i^3|V||E|)$. By Theorem 4.2, the mixing time τ_x is $O(\lambda_i^3|E||V|\log 1/\pi_i(x))$. A maximum matching x has $\pi_i(x) \geq 1/N \geq 1/2^{|E|}$, so by starting the Markov chain at a maximum matching (which we can find in polynomial time), we can mix in $O(\lambda_i^3|E|^2|V|)$ steps.

4.3 Discrete-time quantum walks

A *discrete-time quantum walk* is a unitary process $\{U^t|\psi\rangle\}$, or more frequently the unitary operator U generating this process. Of particular relevance in algorithmic applications are those U whose coefficients are derived naturally and efficiently from a corresponding classical Markov chain P – e.g., with the same locality structure as U [9]. This turns out to be possible only if we are willing to let U operate on the Hilbert space generated by the state transitions (directed edges) of P rather than the states themselves [76, 77].[6] We describe the standard recipe for "quantizing" a Markov chain this way. For additional background on quantum walks, see the surveys of Ambainis [14], Kempe [61], and Kendon [64].

The Grover walk. The *Grover walk* was invented by Watrous [109] as a quantization of the simple random walk on a regular graph to show that quantum logspace machines can simulate their classical counterparts. It has since found application in numerous quantum algorithms. The idea of the walk is to apply the Grover diffusion operator (3.3) locally as a substitute for the classical state transitions.

To illustrate, consider first the following classical procedure for simulating the simple random walk on a d-regular graph $G = (V, E)$ on N vertices. Assume that G is *labeled*; i.e., the edges leaving each vertex (or equivalently, the neighboring vertices) are labeled from 1 to d in some arbitrary (but fixed) way, so we can refer unambiguously to the ith

[6] For example, there is no unitary matrix with tridiagonal nonzero structure, so there is no discrete-time quantum walk $U : \mathcal{H}_{\mathbb{Z}} \rightarrow \mathcal{H}_{\mathbb{Z}}$ on the line $\mathbb{Z}$.

neighbor (denoted x^i) of a vertex x.

```
1:  Pick a uniformly-distributed label $r$ from $[d]$.
2:  If the current vertex is $x$, change it to $x^r$.
```

This procedure uses two registers: a *coin* (edge label) register, which is overwritten by each coin flip C', and a *state* (vertex) register, which is overwritten by each state transition S'. Algebraically:

$$C' : |x\rangle|r\rangle \mapsto \frac{1}{d} \sum_{r' \in [d]} |x\rangle|r'\rangle \qquad S' : |x\rangle|r\rangle \mapsto |x^r\rangle|r\rangle \tag{4.8}$$

In order to *quantize* this classical procedure, it is natural to replace the classical diffusion C' on the coin register by the Grover diffusion:[7]

$$C : |x\rangle|r\rangle \mapsto -|x\rangle|r\rangle + \frac{2}{\sqrt{d}} \sum_{r' \in [d]} |x\rangle|r'\rangle \tag{4.9}$$

This changes the stochastic coin flip C' to a unitary coin flip C. The state transition operator S' is unitary (in fact, a permutation) if and only if G is *consistently labeled* [53]; i.e., if the edges *entering* (as well as those leaving) each vertex are assigned unique labels from $[d]$. Every undirected graph has a consistent labeling, but it can be hard to compute locally.[8] Hence, we replace S by the *rotation map* [88]:

$$S : |x\rangle|i\rangle \mapsto |y\rangle|j\rangle \qquad \text{where } y^j = x^i \tag{4.10}$$

For any undirected graph G, the rotation map S is a permutation. As we will see later, another key property of S (which will aid its analysis and in a sense confirm it as the "best" choice to use with the Grover coin flip) is that it is an *involution* (i.e., S^2 is the identity). Thus, we define the *Grover walk* by

$$W := SC \tag{4.11}$$

Notice that the uniform superposition

$$|u\rangle := \frac{1}{\sqrt{dN}} \sum_{x,r} |x\rangle|r\rangle \tag{4.12}$$

is a fixed point of W.

[7] Other quantum coin flips are possible, for example the Hadamard and DFT operators [61].

[8] It is not known whether a consistent labeling can be computed in logspace. Even when it is trivial to compute (e.g., when G is a Cayley graph) it can perform rather poorly [20].

The Szegedy walk. A generalization of the Grover walk to arbitrary Markov chains was obtained by Szegedy [105]. The idea is to replace each classical random walk transition from state x, selected from the distribution $p_x = P(\cdot, x)$, with a reflection about the vector $|p_x\rangle := \sum_{y \in \mathcal{S}} \sqrt{P(y,x)}|y\rangle$. When P is the simple random walk on a regular graph, this reduces to the Grover walk.

Let us modify our classical procedure for simulating the random walk on a d-regular graph $G = (V, E)$ so that it works for general Markov chains. It will be convenient for us to modify the coin register to hold the label of any possible vertex from V, rather than a label from $[d]$ of an outgoing edge (or neighboring vertex). That is, we keep track of each edge (transition) by its two endpoints rather than by one endpoint and an edge label.[9] With this change, our classical random walk simulation becomes:

```
1':  If the current vertex is $x$, pick a $p_x$-distributed vertex $y$.
2':  Exchange the vertices $x$ and $y$.
```

Contrast this general random walk simulation with the simple random walk simulation from earlier: Step 1' is the new "coin flip" C' generating a neighboring vertex label according to the specified transition probabilities $P(y,x)$, and Step 2' is the new "state transition" S' obtained by using the rotation map with labels from V rather than $[d]$. Algebraically:

$$C' := \sum_x |x\rangle\langle x| \otimes [p_x p_x \cdots p_x] \qquad S' := \sum_{x,y} |y,x\rangle\langle x,y| \tag{4.13}$$

Since Step 2' ensures that the "coin" (neighboring vertex) and "state" (current vertex) registers are treated symmetrically, we could equally well refer to them instead as the *left vertex* and *right vertex* registers.[10]

The natural quantization of the coin flip C' is a reflection C about the subspace $\mathcal{L} := \text{span}\{|x\rangle|p_x\rangle\}$:

$$C := \sum_x |x\rangle\langle x| \otimes (2|p_x\rangle\langle p_x| - I) \tag{4.14}$$

[9] Note that we could have expressed the Grover walk this way; in fact, Watrous [109] did so.

[10] In [105], this walk is interpreted as a "bipartite walk" taking place on the "double-cover" graph $G' = (V', E')$ with "left" and "right" vertices $V' := \{x^R, x^L : x \in V\}$ and edges $E' := \{(x^R, y^L) : (x,y) \in E\}$.

We use the state transition operator $S = S'$ in the quantum as well as the classical case. Since S is an involution it is self-inverse, and we can express two consecutive quantum walk steps as:

$$W_P := R_{\mathcal{L}'} R_{\mathcal{L}} \qquad \text{where } R_{\mathcal{L}} := C \text{ and } R_{\mathcal{L}'} := SCS = SCS^{-1} \tag{4.15}$$

This fact is crucial to the behavior and analysis of the quantum walk, for a simple reason: like Grover's search operator [52], W_P is the product of two reflections, $R_{\mathcal{L}}$ (about subspace $\mathcal{L}$) and $R_{\mathcal{L}'}$ (about subspace $\mathcal{L}' := \text{span}\{S|x\rangle|p_x\rangle\} = \text{span}\{|p_x\rangle|x\rangle\}$. To highlight this significance, we refer to the two-step walk operator W_P as the *quantization* of the classical Markov chain P [105].

Spectrum of the walk operator. The spectrum of W_P is expressed concisely by its *discriminant*, the Gram matrix

$$D(P) := A^\dagger B = \sum_{x,y \in \mathcal{S}} \sqrt{P(x,y)P(y,x)}|x\rangle\langle y| \tag{4.16}$$

of subspaces $\mathcal{L}$ and $\mathcal{L}'$, where $A : |x\rangle \mapsto |x\rangle|p_x\rangle$ and $B = SA$. In particular, if P is reversible then $D(P)$ is the (symmetric) matrix (4.1). The relationship between the discriminant $D(P)$ and the spectrum of W_P is revealed by the following theorem [105]:

Theorem 4.5 *Let $\{\lambda_k = \cos\theta_k\}_{k=1}^{l}$ be the singular values of $D(P)$ lying in the open interval $(0,1)$ and $\{v_k, w_k\}_{k=1}^{l}$ be the corresponding left/right singular vector pairs. Then W_P has l nontrivial invariant subspaces $\{\mathcal{L}_k\}_{k=1}^{l}$ (excluding those with eigenvalues ± 1), each of which is two-dimensional, and the spectrum of W_P inside $\mathcal{L}_k$ is:*

$$\{e^{\pm 2i\theta_k}, Aw_k - e^{\pm i\theta_k} Bv_k\} \tag{4.17}$$

Proof: By definition, any singular vector pair (v_k, w_k) of $D(P)$ with singular value λ_k must satisfy:

$$D(P)v_k = \lambda_k w_k \qquad D(P)^\dagger w_k = \lambda_k v_k \tag{4.18}$$

It is easy to check that all singular values are at most one, so we can write $\lambda_k = \cos\theta_k$ for $\theta_k \in [0, \frac{\pi}{2}]$ where θ_k is the angle between Aw_k and Bv_k. Moreover, each of the (at most) two-dimensional subspaces $\mathcal{L}_k = \text{span}\{Bv_k, Aw_k\}$ is an invariant subspace of

W_P on which W_P acts as a reflection about the axes Bv_k and Aw_k – i.e., as a rotation in $\mathcal{L}_k$ by the angle $2\theta_k$. If $\cos\theta_k \in (0,1)$, then $\mathcal{L}_k$ is exactly two-dimensional and the spectrum of $\mathcal{L}_k$ is as indicated above. If $\cos\theta_k = 1$, then $\mathcal{L}_k$ lies in $\mathcal{L} \cap \mathcal{L}'$ and W_P acts as the identity. If $\cos\theta_k = 0$, then $\mathcal{L}_k$ lies in either $\mathcal{L} \cap \mathcal{L}'^{\perp}$ or $\mathcal{L}^{\perp} \cap \mathcal{L}'$ and W_P acts as $-I$. We call the direct sum of all these subspaces $\mathcal{L}_k$ the *busy* subspace. Its dimension is at most $2N$. On the *idle* subspace $\mathcal{L}^{\perp} \cap \mathcal{L}'^{\perp}$, whose dimension is at least $N^2 - 2N$, W_P acts as the identity. ■

In particular, if P is reversible with stationary distribution π, then the eigenvector

$$|\tilde{\pi}\rangle := \sum_x \sqrt{\pi(x)}|x\rangle|p_x\rangle = \sum_x \sqrt{\pi(x)}|p_x\rangle|x\rangle \tag{4.19}$$

of W_P is a fixed point.

4.4 Continuous-time quantum walks

The *continuous-time quantum walk* $\{e^{-iHt}|\psi\rangle\}$ generated by a time-independent Hamiltonian H merits special focus beyond that of its discrete-time counterpart. This is because the precise relationship between discrete-time and continuous-time quantum walks, unlike that of their classical counterparts, is not well understood.

The Farhi/Gutmann and Aharonov/Ta-Shma walks. Obtaining a natural quantization of the simple random walk on a graph is an easier proposition in continuous time than in discrete time. Farhi and Gutmann [46, 35] noted that the *Laplacian matrix* $L := \Delta - A$ of a graph $G = (V, E)$ is symmetric and can therefore be used as a natural *walk Hamiltonian* on G. This construction was used to speed up classical random walk search procedures on binary decision trees [46] and other highly symmetric graphs [35]. The latter result was extended by Childs et al. [33] to an efficient quantum algorithm for an oracle problem that has no efficient classical algorithm. This was the first oracle separation of BQP and BPP that did not use the quantum Fourier transform.

Aharonov and Ta-Shma [7] observed that the matrix

$$H_P := I - M(P) = I - D(P) \tag{4.20}$$

can be used as a natural walk Hamiltonian for a reversible Markov chain, where $M(P)$ is the matrix (4.1) and $D(P)$ is the discriminant (4.16). Their walk reduces to the Farhi/Gutmann walk when P is the simple random walk on a regular graph. It is easy to see that the ground state of the Aharonov/Ta-Shma walk Hamiltonian is:

$$|\pi\rangle := \sum_x \sqrt{\pi(x)}|x\rangle \tag{4.21}$$

If we can somehow generate this ground state, we can measure it and output a random sample from the classical distribution π. Aharonov and Ta-Shma showed how to do this efficiently for a large class of Metropolis processes, including those used in MCMC approximate counting algorithms. Note that if the ratios $\frac{\pi(y)}{\pi(x)}$ are not efficiently computable, then neither are the coefficients of the Aharonov/Ta-Shma walk Hamiltonian, whereas those of Szegedy's walk operator (4.15) remain efficiently computable.

Spectrum of the walk Hamiltonian. Even more transparently than for the Szegedy walk, the spectrum of the Aharonov/Ta-Shma walk is determined by the discriminant $D(P)$ – and in turn, by P itself. Indeed, it is trivial to prove:

Theorem 4.6 *Let P be a reversible Markov chain with spectrum $\{\lambda_k, v_k\}$. Then the walk Hamiltonian $H = I - D(P)$ has spectrum $\{1-\lambda_k, \sqrt{R}^{-1} v_k\}$.*

Here R is defined as in (4.1). In particular, the spectral gap of H is the same as that of P.

See Table 4.1 for a brief summary of the walk constructions we have just covered.

Walk type	Operator	Eigenvalues
Discrete-time random	P	$\{\lambda_k\}$
Continuous-time random	e^{-Q}, $Q = I - P$	$\{e^{-(1-\lambda_k)}\}$
Discrete-time quantum	W_P^2, $W_P = R_{\mathcal{L}'} R_{\mathcal{L}}$ [109, 105]	$\{e^{\pm 2i \cos^{-1} \lambda_k}\}$ [105]
Continuous-time quantum	e^{-iH_P}, $H_P = I - D(P)$ [46, 7]	$\{e^{-i(1-\lambda_k)}\}$ [7]

Table 4.1: Random vs. quantum walks in discrete and continuous time.

Simulating the walk Hamiltonian. Suppose we design an algorithm that uses a continuous-time quantum walk. How do we simulate the walk by a quantum circuit? This issue was first addressed by Childs et al. [33] to translate their continuous-time quantum walk "algorithm" into an efficient quantum circuit.

First, recall how the classical continuous-time random walk $e^{-Lt} = e^{-(I-P)t}$ can be simulated by its discrete-time counterpart P. Write the continuous-time walk as a power series:

$$e^{-(I-P)t} = \sum_{s=0}^{\infty} \frac{e^{-t}t^s}{s!} P^s \tag{4.22}$$

From this expression one can see that continuous-time walk of length t is nothing more than the discrete-time walk P of length s, where s is Poisson-distributed with mean t. From this observation, an exact simulation is immediate.

Right away we see that in the quantum case, this approach to simulation breaks down. Although the continuous-time quantum walk can be expressed as a power series

$$e^{-iHt} = \sum_{s=0}^{\infty} \frac{e^{-t}(it)^s}{s!} H^s \tag{4.23}$$

it is not clear how this helps us since $\{\frac{e^{-t}(it)^s}{s!}\}_{s=0}^{\infty}$ is not an l_2-normalized wavefunction whose coefficients we could use as walk-length "amplitudes" and (more devastatingly) H is not a unitary operator which we could apply in discrete time. It appears that we cannot use a power series decomposition as the basis for a simulation.

Instead, we use the simulation method of Theorem 2.2, which uses the Lie-Trotter and Suzuki (approximate) product decompositions. Since most walk Hamiltonians are sparse, the slowdown introduced by the simulation is minimal.

Chapter 5

Quantum speedup of hitting algorithms

A *hitting algorithm* is a randomized algorithm whose complexity is related to the hitting time of a Markov chain. Randomized local search algorithms like Schöning's are typical examples. Some of these algorithms become more efficient when the Markov chain is replaced by a quantum walk. This discovery has prompted the definition of *quantum hitting time* and the study of how it relates to its classical counterpart.

5.1 Ambainis' algorithm

Recall from Chapter 3 that the element distinctness problem has quantum query complexity $\Omega(n^{2/3})$ [2], and that Burhman et al. [29] gave a quantum algorithm based on amplitude amplification using $O(n^{3/4})$ queries. Ambainis [16] showed that a matching upper bound of $O(n^{2/3})$ queries is achievable by an algorithm that performs a quantum walk. This was the first major algorithmic application of quantum walks and remains among the most important.

5.1.1 Random walk algorithm for element distinctness

Consider the following classical algorithm for solving the element distinctness problem. First, select a subset of r variables from $\{x_1, x_2, \ldots, x_n\}$ uniformly at random. Then, simulate a random walk on the *Johnson graph* $J(r, n)$ whose vertices are the r-size subsets of $\{x_1, x_2, \ldots, x_n\}$ and whose edges connect two subsets if and only if they differ by a single variable. After each walk step, check the current r-subset to see if it contains two variables mapping to the same value. If no such *collision* is found, output "distinct"; otherwise, output "not distinct."

This nontrivial algorithm does as well as brute-force search, which is already optimal. More importantly, its analysis parallels that of Ambainis' quantum algorithm.

Theorem 5.1 *Using the simple random walk on the Johnson graph $J(r, n)$, the element distinctness problem can be solved with $O(r + n^2/r)$ queries.*

Proof: If the variables $\{x_1, x_2, \ldots, x_n\}$ are not distinct, then the fraction of r-subsets with at least one collision is at least:

$$\varepsilon = \binom{n-2}{r-2} / \binom{n}{r} = \Theta(r^2/n^2) \tag{5.1}$$

Moreover, the spectral gap δ of the random walk on $J(r, n)$ is $\frac{n}{(n-r)r}$ [69], which for $r < \frac{n}{2}$ is $\Theta(1/r)$. By Theorem 4.4, the hitting time of the random walk is at most:

$$\frac{1}{\delta\varepsilon} = O(n^2/r) \tag{5.2}$$

Since the *setup cost* S of selecting the initial r-subset is r queries, the *update cost* U from each walk step is one query (to swap in the new variable), and the *checking cost* U to determine whether the current r-subset has a collision is zero new queries, the overall cost in queries is

$$S + \frac{1}{\delta\varepsilon}(U + C) = O(r + n^2/r) \tag{5.3}$$

and the theorem is proved. ■

As long as $r = \Omega(n)$, the algorithm uses $O(n)$ queries.

5.1.2 Quantum walk algorithm for element distinctness

The key insight of Ambainis [16] is that the operator $W^{\sqrt{r}}$, where W is the Grover walk (4.11) on the Johnson graph $J(r, n)$, works almost as effectively for quantum search as the Grover diffusion operator (3.3). The reason is that the uniform initial state (4.12) is a fixed point of $W^{\sqrt{r}}$, and all other eigenvalues of $W^{\sqrt{r}}$ (in the search algorithm's busy subspace) are bounded away from one.

Lemma 5.2 *Let $\mathcal{H}$ be an m-dimensional Hilbert space and $\{|\psi_1\rangle, \ldots, |\psi_m\rangle\}$ be a basis for $\mathcal{H}$. Let $|\psi_{start}\rangle, |\psi_{good}\rangle \in \mathcal{H}$ have real amplitudes and inner product $\langle\psi_{good}|\psi_{start}\rangle = \alpha$. Let U_1, U_2 be unitary operators $\mathcal{H} \to \mathcal{H}$ such that (i) U_1 (the "oracle") is a reflection about $|\psi_{good}\rangle$, and (ii) U_2 (the "diffusion") is a real-valued matrix in the basis $\{|\psi_1\rangle, \ldots, |\psi_m\rangle\}$, has $|\psi_{start}\rangle$ as a fixed point, and has every other eigenvalue $e^{i\theta}$ satisfying $\theta \in [\epsilon, 2\pi - \epsilon]$ where $\epsilon > 0$ is a fixed constant. Then there exists $t = O(1/\alpha)$ such that $\langle\psi_{good}|(U_2U_1)^t|\psi_{start}\rangle = \Omega(1)$.*

The proof [16] is more involved than that of Grover's algorithm (Theorem 3.2), which is the special case in which $\mathcal{H} = \mathcal{L}$, $m = 2$, $|\phi_{good}\rangle = |\beta\rangle$, $\phi_{start} = |u\rangle$, $\alpha = \Theta(1/\sqrt{N})$, $U_1 = 2|\alpha\rangle\langle\alpha| - I$, and $U_2 = 2|u\rangle\langle u| - I$.

Ambainis used Lemma 5.2 to prove the following.

Theorem 5.3 *Using the Grover walk on the Johnson graph $J(r, n)$, the element distinctness problem can be solved with $O(r + n/\sqrt{r})$ queries.*

Proof: The idea is as follows. The algorithm is exactly as in Lemma 5.2, with U_1 equal to the phase-flip oracle on the "marked" r-subsets containing at least one collision and U_2 equal to the operator $W^{\sqrt{r}}$. The entire algorithm stays within a 5-dimensional subspace $\mathcal{H}$, and the 4 eigenvalues of $W^{\sqrt{r}}$ inside $\mathcal{H} \cap |u\rangle^\perp$ – which are related to those of the simple random walk on $J(r, n)$ by Theorem 4.5 – are bounded away from one. A simple check shows that $\alpha = \sqrt{\varepsilon}$, where ε is the fraction of marked r-subsets given in (5.1). So the total cost in queries is

$$S + \frac{1}{\sqrt{\varepsilon}}(\sqrt{r} \cdot U + C) = O(r + n/\sqrt{r}) \tag{5.4}$$

proving the theorem. ■

Setting $r = n^{2/3}$ gives an algorithm using $O(n^{2/3})$ queries.

5.2 Szegedy's algorithm

Inspired by Ambainis' algorithm for element distinctness, Szegedy [105] defined the quantization (4.15) of a Markov chain and showed that its *quantum hitting time* is at most the square root of the hitting time of its classical counterpart.

5.2.1 The quantum hitting time

Ambainis' algorithm solves the following "spatial search" problem[1] for the special case in which P is the simple random walk on the Johnson graph $J(r, n)$ and f is the predicate that identifies marked r-subsets.

Definition 5.4 *Given oracle access to a $V \times V$ Markov chain P and a predicate $f : V \to \mathbb{B}$, the* spatial search *problem is to find a vertex $v \in f^{-1}(1)$ if one exists – with the stipulation that between successive queries u and v to f, P must be queried along a path from u to v in the digraph $G = (V, E)$ underlying P. The corresponding decision problem is to determine whether $|f^{-1}(1)| > 0$.*

There is an *update cost* U for querying P (i.e., for obtaining the nonzero coefficients of a column $P(\cdot, x)$) and a *checking cost* C for querying f. As in Ambainis' algorithm, we may also consider a *setup cost* S for auxiliary operations such as preprocessing.

There are several ways to solve the spatial search problem. The simplest is by simulating a random walk on P, whose cost U is given by the classical hitting time. Shenvi et al. [97] gave the first quantum walk algorithm for the spatial search problem (on the hypercube). Aaronson and Ambainis [1] gave a quantum algorithm for spatial search using nested application of Grover search. Ambainis et al. [20] gave a quantum walk algorithm for spatial search on the torus $\mathbb{Z}_n^d$ outperforming the algorithm of Aaronson and Ambainis in low dimensions.

Szegedy [105] generalized the quantum walk algorithms of [97, 16, 20] as follows. Suppose that P is symmetric. Let

$$P' := \begin{bmatrix} P_M & 0 \\ P'' & I \end{bmatrix} \tag{5.5}$$

be the absorbing Markov chain obtained from P by marking the states $v \in f^{-1}(1)$, and let $W_{P'}$ be its quantization (4.15). Szegedy's algorithm prepares a single qubit control register in the state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and a walk register in the (uniform) superposition $|\tilde{\pi}\rangle$

[1] Our definition of the spatial search problem is inspired by, but different than, that of Aaronson and Ambainis [1].

(4.19), then applies $W^t_{P'}$ with quantum control, creating the state:

$$\frac{1}{2}|0\rangle(|\tilde{\pi}\rangle + W^t_{P'}|\tilde{\pi}\rangle) + \frac{1}{2}|1\rangle(|\tilde{\pi}\rangle - W^t_{P'}|\tilde{\pi}\rangle) \tag{5.6}$$

Measuring the control register yields the value 0 with certainty if there are no marked states, since in this case $W_{P'} = W_P$ so $|\tilde{\pi}\rangle$ is a fixed point of $W_{P'}$. If t is chosen uniformly at random from $[0..T]$, a simple calculation shows that this same measurement yields the value 1 with probability at least $\frac{1}{8}$ if the fraction ε of marked states is less than $\frac{1}{2}$ and T is chosen so that:

$$\mathrm{E}_t[||W^t_{P'}|\tilde{\pi}\rangle - |\tilde{\pi}\rangle||_2^2] \geq \varepsilon \tag{5.7}$$

The minimum T satisfying (5.7) is the *quantum hitting time.*[2]

5.2.2 Proof of the quadratic hitting time speedup

Szegedy [105] showed that the quantum hitting time is at most the square root of the classical hitting time. The following lemma plays a key role in the proof.

Lemma 5.5 *Let P be any Markov chain and W_P be its quantization. Let $z = \sum_k \nu_k z_k$ be the decomposition of a unit vector z into unit vectors z_k from each of the two-dimensional invariant spaces $\mathcal{L}_k$ in Theorem 4.5. If $T \geq 100 \sum_k \nu_k^2/\theta_k$, then:*

$$E_t[\langle z|W^t_P|z\rangle] \leq \frac{1}{2} \tag{5.8}$$

The intuition is that within each subspace $\mathcal{L}_k$, it takes $t = \Theta(1/\theta_k)$ steps to fully rotate and cause cancellation. Using Lemma 5.5, Szegedy [105] proved the following.

Theorem 5.6 *Let P be a symmetric, ergodic Markov chain and let h_u be its hitting from the uniform distribution u to the marked set M. Then the quantum hitting time is $O(\sqrt{h_u})$.*

Proof: Suppose first that P is reversible – later we will focus on the special case where it is symmetric. Then the discriminant of P' is

$$D(P') = \begin{bmatrix} \sqrt{R_M}^{-1} P_M \sqrt{R_M} & 0 \\ 0 & I \end{bmatrix} \tag{5.9}$$

[2] This definition is not to be confused with the notion of quantum hitting time between two correlated vertices in a highly symmetric graph – e.g., antipodes of a hypercube [60].

where R_M is the restriction of R (4.1) to those rows and columns indexed by elements of $V \setminus M$. In particular, $D(P')$ is symmetric, so its left/right singular vector pairs (v_k, w_k) are eigenvectors $v_k = w_k$, and Theorem 4.5 implies that $W_{P'}$ has the nontrivial normalized eigenvectors:

$$\frac{(I - e^{\pm i\theta_k} S)Av_k}{\sqrt{1 - 2\cos\theta_k}} \tag{5.10}$$

Decompose $|\tilde{\pi}\rangle$ into its projections

$$|\tilde{\pi}_M\rangle = \sum_{x \in M} \sqrt{\pi(x)}|x\rangle|p_x\rangle \qquad |\tilde{\pi}_{V\setminus M}\rangle = \sum_{x \notin M} \sqrt{\pi(x)}|x\rangle|p_x\rangle \tag{5.11}$$

on the the marked and unmarked states, respectively. The latter can be expressed as

$$|\tilde{\pi}_{V\setminus M}\rangle = \sum_k \nu_k A v_k \tag{5.12}$$

where each Av_k is a unit vector in $\mathcal{L}_k$ and $\nu_k = \sum_{x \notin M} \sqrt{\pi(x)} v_k(x)$. We assume that the (weighted) fraction $\varepsilon := \sum_{x \in M} \pi(x)$ of marked states is at most $\frac{1}{2}$, otherwise the search problem is trivial. Apply Lemma 5.5 to the unit vector $z = \frac{1}{\sqrt{1-\varepsilon}}|\tilde{\pi}_{V\setminus M}\rangle$ for

$$T = \frac{100}{1-\epsilon} \sum_k \nu_k^2 \sqrt{\frac{1}{1-\lambda_k}} \geq \frac{100}{1-\epsilon} \sum_k \nu_k^2/\theta_k \tag{5.13}$$

where the inequality follows from the fact that $\theta_k \geq \sin\theta_k \geq \sqrt{1 - \cos\theta_k}$. We conclude that:

$$\mathrm{E}_t[||W_{P'}^t|\tilde{\pi}\rangle - |\tilde{\pi}\rangle||_2^2] = \mathrm{E}_t[2 - 2\langle\tilde{\pi}|W_{P'}^t|\tilde{\pi}\rangle] \geq 1 - \varepsilon \tag{5.14}$$

So the quantum hitting time is at most T. Since P is reversible, P_M has the same eigenvalues as $D(P')$ restricted to $V \setminus M$, so we can bound T by:

$$T \leq 200\sqrt{\frac{1}{1 - \max_k \lambda_k}} \leq 200\sqrt{\frac{1}{1 - ||P_M||}} \tag{5.15}$$

If P is symmetric, we can do even better: Since $\sum_k \nu_k^2 \leq 1$,

$$T \leq 200\sqrt{\sum_k \nu_k^2 \frac{1}{1-\lambda_k}} \tag{5.16}$$

and since π is the uniform distribution u, ν_k is the inner product of v_k with the (ℓ_2-normalized) uniform n-vector $|u\rangle$. By comparison with expression (4.5),

$$T \leq 200\sqrt{h_u} \tag{5.17}$$

proving the theorem. ■

Unlike its classical counterpart, the quantum hitting time is not an upper bound on the time to *find* a marked state – only to *detect* if one exists. However, it is sometimes the case that the walk register outputs a marked state with reasonable probability. Using the fact that (5.15) tightly bounds the quantum hitting time when P is state-transitive and $|M| = 1$, Szegedy [105] shows:

Theorem 5.7 *Suppose that P is state-transitive and $|M| = 1$. Then the walk register outputs a marked state with probability $\Omega(N/h_u)$ after $O(\sqrt{h_u})$ steps.*

Amplitude amplification can be used to recover the search upper bounds of Ambainis et al. [20] on the d-dimensional torus.

Theorem 5.6 can be used to solve the element distinctness problem as efficiently as Ambainis' algorithm. Indeed, the quantum hitting time on the Johnson graph $J(r, n)$ is at most

$$\sqrt{\frac{1}{\delta\varepsilon}} = O(n/\sqrt{r}) \tag{5.18}$$

where ε is the fraction of marked states and δ is the spectral gap of $J(r, n)$ as in (5.2). So the total cost in queries is

$$S + \frac{1}{\sqrt{\delta\varepsilon}}(U + C) = O(r + n/\sqrt{r}) \tag{5.19}$$

which is identical to (5.4).

5.3 The MNRS algorithm

Notice that after every walk step, Szegedy's algorithm checks the current state to see if it is marked. On the other hand, Ambainis' algorithm performs this check only once every $\sqrt{r}$ steps. Since the checking cost C is zero in the element distinctness application, the algorithms perform similarly. But in other applications with different costs S, U, and C to balance, the distinction is important. Magniez et al. [74] show that Szegedy's walk operator W_P can be used to gave a quantum algorithm (which we shall refer to as the MNRS algorithm) whose cost balance

$$S + \frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}U + C) \tag{5.20}$$

is as good as that of Ambainis' algorithm. The key insight is to interleave the checking step with a *global* reflection about the initial vector $|\tilde{\pi}\rangle$ (using phase estimation on W_P), rather than a *local* reflection about the vector $|p_x\rangle$ as in Szegedy's algorithm.

Theorem 5.8 *Let P be a reversible, ergodic Markov chain with stationary distribution π and spectral gap δ. Let M, the set of marked states, satisfy $\varepsilon := \sum_{x\in M} \pi(x) > 0$. Then there is a quantum algorithm solving the spatial search problem with cost (5.20). Moreover, the algorithm outputs a marked state with probability $\Omega(1)$.*

Proof: The *phase gap* $\Delta(P) := \min_{k:\theta_k>0} 2\theta_k$ of Szegedy's walk operator W_P satisfies

$$\Delta(P) = \Omega(\sqrt{\delta}) \tag{5.21}$$

by inspection of (5.13). So at a cost of $\frac{1}{\Delta(P)}U = \frac{1}{\sqrt{\delta}}U$, we can generate the operators $W_P^{2^j}$ for $j = 0, 1, \ldots, s = O(\log \Delta(P))$ and use phase estimation (Theorem 1.9) to distinguish the eigenvector $|\tilde{\pi}\rangle$ (whose eigenvalue has phase $\theta_k = 0$) from any other eigenvector (whose eigenvalue has phase satisfying $|\theta_k| \geq \Delta(P)$). By "distinguish," we mean that the output of the phase register is almost certainly zero on input $|\tilde{\pi}\rangle$ and almost certainly nonzero when the input is any other eigenvector. This allows us to implement an (approximate) reflection about $|\tilde{\pi}\rangle$:

$$R := Q(W_P)(2|0^s\rangle\langle 0^s| - I)Q(W_P)^{-1} \approx 2|\tilde{\pi}\rangle\langle\tilde{\pi}| - I \tag{5.22}$$

Define the superposition of marked states

$$|\mu\rangle := \frac{1}{\sqrt{\varepsilon}} \sum_{x\in M} \sqrt{\pi(x)}|x\rangle|p_x\rangle \tag{5.23}$$

and let O be the operator that flips the phase of any marked state. Inside the subspace $\mathcal{L} := \mathrm{span}\{|\tilde{\pi}\rangle, |\mu\rangle\}$, O is the reflection about $|\mu\rangle^{\perp}$. Modulo several details, we can use Lemma 5.2 with $\mathcal{H} = \mathcal{L}$, $m = 2$, $|\phi_{good}\rangle = |\mu\rangle$, $|\phi_{start}\rangle = |\tilde{\pi}\rangle$, $U_1 = O$, $U_2 = R$, and

$$\alpha = \langle\mu|\tilde{\pi}\rangle = \sin^{-1}\sqrt{\varepsilon} \approx \sqrt{\varepsilon} \tag{5.24}$$

to conclude that $(RO)^k$ finds a marked state for $k = O(1/\sqrt{\varepsilon})$. Since each iteration RO has a cost of $\frac{1}{\sqrt{\delta}}U + C$, the total cost is as indicated in (5.20). ■

The principal proof detail we have omitted is the use of amplitude amplification (specifically, the version of Høyer et al. [54] that handles two-sided error) to minimize the accumulation of errors from approximate (imperfect) reflection.

If we do not know ε a priori, it suffices to run the algorithm using a lower bound ε' on ε should the latter be nonzero. In particular, we can use the lower bound

$$\varepsilon' := \frac{1}{\sqrt{\pi_*}} \tag{5.25}$$

where $\pi_* := \min_x \pi(x)$.

The MNRS algorithm can sometimes underperform Szegedy's algorithm. For example, when $C \approx U$ and $h_u \ll 1/(\delta\varepsilon)$ – which is the case when P is the simple random walk on the two-dimensional torus – Szegedy's algorithm is significantly more efficient (and probably optimal). More generally, Szegedy's algorithm performs better when C is not much larger than U and the quantum hitting time T satisfies

$$T \approx \sqrt{\frac{1}{1 - ||P_M||}} \ll \sqrt{\frac{1}{\delta\varepsilon}} \tag{5.26}$$

by inspection of (5.15). This statement is true for any reversible Markov chain.

We summarize the quantum search methods we have seen thus far in Table 5.1.

Search method	First reflection	Second reflection
Grover [52]	$I - 2\sum_{x\in f^{-1}(1)} \vert x\rangle\langle x\vert$	$2\sum_{x\in\mathbb{B}^n} \vert x\rangle\langle x\vert - I$
Amp. Amp. [28]	$I - 2\vert\psi_{x,1},1\rangle\langle\psi_{x,1},1\vert$	$A_f(2\vert x,0^{k-n}\rangle\langle x,0^{k-n}\vert - I)A_f^{-1}$
Ambainis [16]	$I - 2\sum_{x\in f^{-1}(1)} \vert x\rangle\langle x\vert$	$W^{\sqrt{r}}$
Szegedy [105]	$C = \sum_x \vert x\rangle\langle x\vert \otimes (2\vert p_x\rangle\langle p_x\vert - I)$	SCS^{-1}
MNRS [74]	$I - 2\sum_{x\in f^{-1}(1)} \vert x\rangle\langle x\vert$	$Q(W_P)(2\vert 0^s\rangle\langle 0^s\vert - I)Q(W_P)^{-1}$

Table 5.1: Quantum search via the product of two reflections.

5.4 Algorithmic applications

We concluded Chapter 3 with a few applications of Grover's algorithm and amplitude amplification. Already we have seen that Ambainis' quantum walk algorithm solves the element distinctness problem optimally. Here we mention the success of quantum walk algorithms in solving a few other search problems.

Triangle finding. Magniez, Santha, and Szegedy [75] improved the $\tilde{O}(n^{10/7})$ quantum query algorithm of Szegedy [105] to an $\tilde{O}(n^{13/10})$ algorithm using quantum walks. The algorithm works by reduction to the *graph collision problem*: decide if there exists an edge $(u,v) \in E$ of a fixed graph $G = (V,E)$ on r vertices satisfying $f(u) = f(v) = 1$ for a predicate f. Ambainis' algorithm can be extended from element distinctness to solve the graph collision problem in $\tilde{O}(r^{2/3})$ queries to f.

The algorithm of [75] finds a triangle in a graph $G = (V,E)$ on n vertices as follows. First, it selects an initial subset $A \subseteq V$ of size r at random and determines $G|_A$ for a setup cost of $S = r^2$ queries. Then it performs a quantum walk on the Johnson graph of all r-subsets of V, alternating between checking and update steps. The checking step runs the graph collision subroutine on $G|_A$ with the oracle $f_w(v) = 1$ if and only if $(v,w) \in E$, where $w \in V$ is fixed. This finds a triangle $\{u,v,w\}$ with $(u,v) \in G|_A$, if there is one. This is repeated $O(\sqrt{n})$ times using amplitude amplification to find any triangle with two vertices in $G|_A$. The total checking cost C then is at most $\tilde{O}(r^{2/3})$ queries (for finding a graph collision) times $O(\sqrt{n})$ queries. The cost U of an update step is r queries to update $G|_A$ to $G|_{A'}$, where A' differs from A by one vertex. It follows from (5.1) and (5.4) that triangle finding can be solved with cost:

$$S + \frac{1}{\sqrt{\varepsilon}}(\sqrt{r} \cdot U + C) = \tilde{O}(r^2 + \frac{n}{r}(\sqrt{r} \cdot r + r^{2/3}\sqrt{n})) \tag{5.27}$$

This expression is optimized to $\tilde{O}(n^{13/10})$ by setting $r = n^{2/3}$. Using the MNRS algorithm one can bring this down to $O(n^{13/10})$ [74]. Childs and Eisenberg gave an algorithm for finding k-cliques and more general subsets of vertices [34]. See the short survey [89] for details.

Matrix product verification. Ambainis' algorithm can be applied to matrix product verification to give a quantum algorithm using $O(n^{5/3})$ queries and $\Theta(n^2)$ time, which already improves the upper bound of Ambainis et al. [18]. Burhman and Špalek [30] give an algorithm using $O(n^{5/3})$ time and queries by performing Szegedy's walk on the product of two Johnson graphs – one (R) for the row subsets of A and another (S) for the column subsets of B. At each walk step, random vectors p and q are chosen and the identity

$$(p|_R A|_R) \cdot (B|_S q|_S) = p|_R \cdot (C|_{R,S} \cdot q|_S) \tag{5.28}$$

is verified using amplitude amplification to locate an inconsistency if there is one. See [30] for details.

Group commutativity testing. In the *group commutativity testing* problem, we are given a black-box group specified by its k generators and are to decide if the group is commutative using queries of the form "What is the product of group elements g and h?" The classical query complexity is $\Theta(k)$ group operations. Magniez and Nayak [73] gave an (essentially optimal) $\tilde{O}(k^{2/3})$ quantum query algorithm by walking on the product of two graphs whose vertices are (ordered) l-tuples of distinct generators and whose transition probabilities are nonzero only where the l-tuples at two endpoints differ in at most one coordinate.

Boolean formula evaluation. Let ϕ be a Boolean formula (circuit) composed of AND, OR, and NOT (or without loss of generality, only NAND) gates and n input bits $\{x_1, x_2, \ldots, x_n\}$. We would like to determine whether ϕ evaluates to 0 or 1 using as few queries as possible of the form "What is the value of input x_i?"

Let T be the tree whose branching models the fan-in of ϕ. At one extreme, ϕ could be the OR function on all n inputs, which is equivalent to the search problem – with classical query complexity $\Theta(n)$ and quantum query complexity $\Theta(\sqrt{n})$ – and T has branching factor n and depth 1. At the other extreme, ϕ could be defined by a complete balanced binary tree T (branching factor 2 and depth $\log n$) with alternating levels of AND and OR gates. A clever classical randomized algorithm solves this instance in $O(n^{0.753})$ queries [94, 95], but for quite some time no quantum algorithm was known to perform better – nor was a lower bound better than $\Omega(\sqrt{n})$ known.

Using a technique from scattering theory, Farhi, Goldstone, and Gutmann [43] showed that a continuous-time quantum walk on T (augmented with a few cleverly chosen additional edges) can be used to evaluate ϕ in time $O(n\sqrt{\log n})$. Ambainis et al. [19] have obtained discrete-time quantum algorithms using $O(\sqrt{n})$ queries when T is at least "approximately balanced" and $O(n^{\frac{1}{2}+o(1)})$ queries for any ϕ by preprocessing T to "rebalance" it. This nearly proves that the square of the quantum query complexity of a formula ϕ is a lower bound on the formula size, a question raised by Laplante, Lee, and Szegedy [70].

Chapter 6

Quantum speedup of mixing algorithms

A *mixing algorithm* is a randomized algorithm whose complexity is related to the mixing time of a Markov chain. Most approximation algorithms for #P-complete problems are mixing algorithms that use Markov chain Monte Carlo (MCMC) sampling. A natural question is whether these algorithms can be made more efficient by replacing the Markov chain with a quantum walk. This question, unlike the corresponding question for hitting algorithms, remains unanswered. However, there has been some progress toward a reasonable definition of *quantum mixing time* that is in some cases significantly smaller than its classical counterpart.

6.1 MCMC sampling redux

Let us define the following problem common to MCMC sampling methods like the monomer-dimer covering algorithm from Chapter 4.

Definition 6.1 *Given a start vertex $s \in V$ and oracle access to a $V \times V$ Markov chain P with stationary distribution π, the* spatial sampling *problem is to output a random vertex v distributed ϵ-close to π in total variation distance – with the stipulation that P must be queried along a path from s to v in the digraph $G = (V, E)$ underlying P.*

Unlike the spatial search problem, there is no checking cost C – only an update cost U for querying $P(\cdot, x)$, and potentially also a setup cost S for preprocessing. MCMC methods exploit a tradeoff in which the setup cost (usually associated with building a simple data structure or initial configuration) and the update cost (for perturbing the data structure or configuration to a very similar one) are both quite small, and the number of updates T is *small enough* for the algorithm to be considered efficient. By simulating a random walk on P from initial state x, this number T is bounded from above by the mixing time τ_x (4.3). In turn, τ_x is often estimated using the inequality (4.2)

$$\delta^{-1} \leq \tau_x \leq \delta^{-1} \log 1/\pi(x) \tag{6.1}$$

where δ is the spectral gap of P.

Although this inequality is tight with respect to δ, it is somewhat unsatisfactory in the following sense. Let P be the simple random walk on a regular graph G on N vertices. It can be shown [103] that the diameter d of G satisfies:

$$d(G) = O(\sqrt{\delta^{-1}} \log N) \tag{6.2}$$

Clearly, $d(G)$ is a lower bound on the number of updates needed to solve the spatial sampling problem for P from a worst-case start vertex $s \in V$. But (6.1) shows that simulating a random walk on P does *significantly* worse than this. For example, consider the simple case in which P is the simple random walk on a line: its spectral gap is $\delta = \Theta(\frac{1}{N^2})$, so the random walk on P samples from (close to) its uniform distribution only after $\Theta(N^2)$ walk steps.

Fundamentally, the random walk's performance bottleneck in solving the spatial sampling problem is its Gaussian width. After t steps, most of the probability mass is supported on only the middle $\Theta(\sqrt{t})$ vertices of the line. What we would like is an algorithm that spreads mass nearly uniformly across $\Theta(t)$ vertices after t steps. More generally, we would like an algorithm for the spatial sampling problem on a regular graph requiring only

$$T = O(\sqrt{\delta^{-1}} \log N) \tag{6.3}$$

updates. Chen, Lovász, and Pak [31] achieved some progress toward this end by *lifting* P to a faster-mixing Markov chain P', but their algorithm requires both knowledge of the chain's global structure and its use in solving an NP-complete network flow problem to find low-congestion paths along which to "route" probability mass efficiently. A more promising idea that has received attention in recent years is replacing P by a quantum walk.

6.2 The quantum mixing time?

Figure 4.1 compares the behavior of random and quantum walks on the one-dimensional infinite line. In continuous time, a distribution p_t initally concentrated at the origin propagates by the simple random walk P according to

$$p_t(x) = e^{-(I-P)t} p_0 = e^{-2t} I_x(2t) \approx \frac{1}{\sqrt{4\pi t}} \exp(-x^2/4t) \tag{6.4}$$

where I_x is the modified Bessel function of order x and the latter expression is a Gaussian of width $\sqrt{2t}$. Using P as the Hamiltonian for a continuous-time quantum walk, a wavefunction $|\psi_t\rangle$ initially concentrated at the origin spreads as

$$\langle x|\psi_t\rangle = e^{-iPt}(-i)^{|x|} J_{|x|}(2t) \tag{6.5}$$

where $J_{|x|}$ is a Bessel function of order $|x|$. For $|x| \gg 1$ the quantity $|J_{|x|}(t)|$ is (a) exponentially small in $|x|$ for $t < (1-\epsilon)\cdot|x|$ and (b) of order $|x|^{-1/2}$ for $t > (1+\epsilon)\cdot|x|$ (Childs [32]). So in fact, the two "peaks" of $|\langle x|\psi_t\rangle|^2$ in Figure 4.1 are exponentially thin, and the more important property of $|\psi_t\rangle$ is that its amplitude is spread almost entirely and nearly uniformly over the interval between the two peaks. Nayak et al. [81, 17] showed that the discrete-time *Hadamard walk* on the line also puts amplitude $\Omega(1/\sqrt{t})$ on the vertices within an interval of length $\Theta(t)$ around the origin at time t.

Nayak et al. [81, 17] and Aharonov et al. [4] were the first to investigate whether quantum walks might speed up classical mixing processes. Nayak et al. [81, 17] showed that the spreading property of the Hadamard walk on the cycle $\mathbb{Z}_N$ yields a notion of *quantum mixing time* which is $\Theta(N)$ in contrast to the classical $\Theta(N^2)$. Aharonov et al. [4] proved an $O(N \log N)$ upper bound for the Hadamard walk on the cycle using a different notion of quantum mixing time. Moore and Russell [79] showed that the continuous-time Farhi/Gutmann walk and the discrete-time Grover walk on the hypercube $\mathbb{Z}_2^d$ both mix in time $O(d)$ as opposed to $O(d \log d)$ classically. The Grover walk also seems to mix quickly on the torus $\mathbb{Z}_n^2$, although this has not been verified analytically [55, 106, 72].

Absence of an ergodic theorem. Let U be a discrete-time quantum walk operator, or let $U = e^{-iH}$ where H is a continuous-time quantum walk Hamiltonian, and define the $N \times N$ stochastic matrix

$$P_t(y,x) := |\langle y|U^t|x\rangle|^2 \tag{6.6}$$

induced by starting the quantum walk from a classical state, running it for time t, and measuring the walk register in the classical basis.[1] A key observation of Aharonov et al. [4] is that while P_t does not converge to a limit (due to the underlying unitary dynamics), its time average does: if U is a unitary operator and ω_T is the uniform distribution on $[0..T]$, then the following limit exists.

$$\Pi := \lim_{T\to\infty} \mathrm{E}_{t\leftarrow\omega_T}[P_t] \tag{6.7}$$

However, it is typically the case that the quantum walk generated by a Markov chain P with stationary distribution π satisfies

$$\Pi \neq [\pi\pi\cdots\pi] = P^\infty \tag{6.8}$$

even though in the classical case we have $P^t \to P^\infty$ as in (4.2). Moreover, the columns of Π are not identical, so the limiting distribution of the time-averaged quantum walk depends on the initial state. Even when P is symmetric (so that π is the uniform distribution u), we rarely have an *ergodic theorem* (time-space average) $\Pi = [uu\cdots u]$: Aharonov et al. [4] were able to prove this equality only for quantum walks U on Cayley graphs of Abelian groups in which U has distinct eigenvalues. Gerhardt and Watrous [51] showed that

$$||\Pi - [uu\cdots u]||_1 \geq \frac{2}{n} - \frac{2}{n\cdot n!}\binom{2n-2}{n-1} \tag{6.9}$$

for a continuous-time quantum walk on the symmetric group S_n, and Moore and Russell [79] showed that there exists an $\epsilon > 0$ such that

$$||\Pi - [uu\cdots u]||_1 \geq \epsilon \tag{6.10}$$

for a continuous-time quantum walk on the hypercube $\mathbb{Z}_2^d$. Even more pathological are the Farhi/Gutmann and Grover walks on the complete graph K_N, for which Π has diagonal elements $\approx 1 - \frac{1}{N}$ and off-diagonal elements $\approx \frac{1}{N}$.

[1] If the walk is discrete-time and requires a second register, we consider its effect on a single register.

Lower bound arguments. One way to force the quantum walk to converge to π is to use a walk operator which depends on the initial state. For example, if we run the MNRS algorithm in reverse – using the initial state $|x\rangle$ as the "marked" state – and measure the output state $\approx |\tilde{\pi}\rangle$, we obtain a mixing algorithm with cost

$$T = O(1/\sqrt{\delta\pi(x)}) \tag{6.11}$$

whose dependence on δ is precisely what we seek. Unfortunately, the dependence on $\pi(x)$ is prohibitively costly. The prospects for using quantum hitting procedures as mixing procedures in this manner are severely limited by the $\Omega(\sqrt{N})$ quantum search lower bound of Bernstein et al. [25], which is typically beaten handily by the standard classical mixing procedure.

Aharanov et al. [4] proved lower bounds for several notions of quantum mixing time in terms of the *conductance* of the underlying Markov chain, a geometric parameter quadratically related to the spectral gap. However, their lower bounds do not rule out the possibility of an upper bound of the form (6.3) on the quantum mixing time. In particular, their lower bound for *non-unitary* quantum walks is relatively weak[2] – which brings us to the subject of decoherent quantum walks.

6.3 Decoherent quantum walks

Evidence in favor of a quantum speedup of the form (6.3) for the spatial sampling problem is presented in [91, 90] using quantum walks that *decohere* under repeated randomized measurements. Decoherence (in small amounts) was first identified as a way to improve spreading and mixing properties in numerical experiments performed by Kendon and Tregenna [65] and in analytical estimates by Fedichkin et al. [47, 101, 102]. On the other hand, high rates of decoherence in quantum walks have been shown to degrade mixing properties substantially by the quantum Zeno effect (Alagic and Russell [10]). For an excellent survey of these and other aspects of decoherent quantum walks, see Kendon [64].

[2] The lower bound of Aharonov et al. [4] for non-unitary quantum walks applies to the quantum "sampling time," which is in general a larger quantity than the quantum mixing time.

6.3.1 A quantum walk algorithm that mixes

Henceforth, we shall use $\langle U, \omega_T\rangle$ and $\langle H, \omega_T\rangle$ to denote the discrete-time quantum walk (with unitary walk operator U) and the continuous-time quantum walk (with walk Hamiltonian H). Here ω_T is the *measurement rule* – a T-parametrized family of probability mass (or density) functions on $[0, \infty)$ characterizing the (random) time at which a total measurement of the walk is performed in the classical basis.

We show that under a very mild condition on ω_T, the T'-repeated continuous-time Aharonov/Ta-Shma walk (4.20) $\langle H_P, \omega_T\rangle$ derived from a symmetric Markov chain P converges to the uniform limit $P^\infty = [uu \cdots u]$ [91, 90].[3] In particular, any reasonable ("smooth") measurement rule ω_T – whether chosen by design or used to model decoherence – can be expected to satisfy the condition. Thus, decoherent quantum walks (which are non-unitary) circumvent the barrier to uniform-mixing identified by Aharonov et al. [4].

First we prove a variant of Theorem 3.4 in Aharonov et al. [4].

Lemma 6.2 *Let P be a symmetric Markov chain and ω_T be a family of distributions satisfying $E_{t\leftarrow\omega_T}[e^{i\theta t}] \to 0$ as $T \to \infty$ for any $\theta \neq 0$. In the limit $T \to \infty$, the stochastic matrix $\hat{P}_T := E_{t\to\omega_T}[P_t]$ (6.7) generated by the quantum walk $\langle H_P, \omega_T\rangle$ approaches the matrix Π with entries*

$$\Pi(y, x) := \sum_j \Big| \sum_{k\in C_j} \langle y|\phi_k\rangle\langle\phi_k|x\rangle \Big|^2 \tag{6.12}$$

where $\{\lambda_k, |\phi_k\rangle\}$ is the spectrum of P and $\{C_j\}$ is the partition of these indices k obtained by grouping together the k with identical λ_k.

Proof: Decomposing the quantum walk along spectral components gives us:

$$\hat{P}_T(y, x) = E_{t\leftarrow\omega_T}[|\sum_k \langle y|\phi_k\rangle\langle\phi_k|x\rangle e^{i\lambda_k t}|^2] \tag{6.13}$$

Writing $|\cdot|^2$ as a product of complex conjugates, the right hand side becomes:

$$E_{\omega_T}[(\sum_k \langle y|\phi_k\rangle\langle\phi_k|x\rangle)(\sum_l \langle\phi_l|y\rangle\langle x|\phi_l\rangle)e^{i(\lambda_k-\lambda_l)t}] \tag{6.14}$$

[3] Though repeatedly measured quantum walks generate mixed states, our analysis is simplest without the use of density matrices.

By linearity of expectation, this is equivalent to:

$$(\sum_k \langle y|\phi_k\rangle\langle\phi_k|x\rangle)(\sum_l \langle\phi_l|y\rangle\langle x|\phi_l\rangle)E_{\omega_T}[e^{i(\lambda_k-\lambda_l)t}] \tag{6.15}$$

Now by assumption, $E_{\omega_T}[e^{i(\lambda_k-\lambda_l)t}]$ vanishes as $T \to \infty$ for all $\lambda_k \neq \lambda_l$, so we have

$$\begin{aligned} \hat{P}_T(y,x) &\to \sum_k \langle y|\phi_k\rangle\langle\phi_k|x\rangle(\sum_{l:\theta_l=\theta_k} \langle\phi_k|y\rangle\langle x|\phi_k\rangle) \\ &= \sum_j |\sum_{k\in C_j} \langle y|\phi_k\rangle\langle\phi_k|x\rangle|^2 = \Pi(y,x) \end{aligned} \tag{6.16}$$

in the limit $T \to \infty$. ■

Using Lemma 6.2, we prove that the T'-repeated quantum walk solves the spatial sampling problem for symmetric P using a *finite* number of updates [90].

Theorem 6.3 *Let P be a symmetric, irreducible Markov chain and ω_T be a family of distributions satisfying $E_{t\leftarrow\omega_T}[e^{i\theta t}] \to 0$ as $T \to \infty$ for any $\theta \neq 0$. For T sufficiently large (but fixed), the T'-repeated quantum walk $\langle H_P, \omega_T\rangle$ generates a Markov chain $(\hat{P}_T)^{T'}$ approaching $[uu\cdots u]$ in the limit $T' \to \infty$.*

Proof: We need to show that for T sufficiently large, the Markov chain $\hat{P}_T$ is ergodic with uniform stationary distribution.

That the uniform distribution is stationary is clear: each of the $P_t(y,x) := |\langle y|e^{iPt}|x\rangle|^2$ has uniform stationary distribution since the uniform classical state is invariant under unitary quantum operations and under total measurement of the system; thus, any probabilistic combination $\hat{P}_T$ of them has uniform stationary distribution.

To show that $\hat{P}_T$ is ergodic for all sufficiently large T, it is sufficient (by Lemma 6.2) to prove that Π is ergodic. (The latter implies the former because the ergodic matrices form an open subset of the set of stochastic matrices.) Why is Π ergodic? Because the 1-eigenspace of P is precisely the space spanned by u, so it follows from Lemma 6.2 (by consideration of only this nondegenerate eigenspace in the expression (6.12)) that $\Pi(y,x) \geq 1/N^2$ for every x, y.

In fact, each of the P_t (and so $\hat{P}_T$ and Π as well) is symmetric.[4] To see this, write

[4] Π is also positive semidefinite: it is the Gram matrix of $\{f_s\}$ with $f_s(kl) := \langle s|\phi_k\rangle\langle\phi_l|s\rangle$ if $\lambda_k = \lambda_l$, 0 otherwise.

out the Taylor series for e^{iPt} and note that every positive integer power P^k is symmetric (since $P^2(x,y) = \sum_z P(x,z) \cdot P(z,y) = \sum_z P(y,z) \cdot P(z,x) = P^2(y,x)$). This property will be quite useful in the next subsection: it will allow us to use Theorem 4.2 to relate the spectral gap and the mixing time of $\hat{P}_T$. ■

6.3.2 Robustness of the quantum mixing time

Theorem 6.3 showed that just about any measurement (or decoherence) rule ω_T can be expected to force the T'-repeated quantum walk $\langle H_P, \omega_T \rangle$ to mix to the uniform distribution if P is symmetric. In any algorithmic application, we would like to know which ω_T converges *fastest*. We settle this question by showing that "most" ω_T (those which are at least somewhat "smooth") are at least approximately equivalent. Hence, the notion of quantum mixing time is robust for a decoherent quantum walk – essentially independent of ω_T.

We illustrate this result using two concrete measurement rules ω_T that fall into this (approximate) equivalence class: the uniform distribution $\bar{\mu}_T := \frac{1}{T}\chi_{[0,T]}$, where χ is the characteristic function, and the exponential distribution $\tilde{\mu}_T(t) := \frac{1}{T}e^{-t/T}$. The uniform distribution is the measurement rule typically used in the computer science literature [4, 79, 51], since it corresponds most naturally to an algorithm. The exponential distribution is the measurement rule typically used in the physics literature [65, 10, 47], since it describes the interarrival time between measurements in a Poisson process with rate $\lambda = 1/T$. The proof of (approximate) equivalence applies more generally to any two measurement rules with finite expectation and significant overlap for most T.

Let P be a symmetric Markov chain. Let $\bar{P}_T := \mathrm{E}_{t \leftarrow \bar{\mu}_T}[P_t]$ and $\tilde{P}_T := \mathrm{E}_{t \leftarrow \tilde{\mu}_T}[P_t]$ be the stochastic matrices (6.7) generated by the quantum walks $\langle H_P, \bar{\mu}_T \rangle$ and $\langle H_P, \tilde{\mu}_T \rangle$, respectively, and let $\bar{\delta}_T$ and $\tilde{\delta}_T$ be their respective spectral gaps.

Lemma 6.4 *For any $k \geq 1$, we have the inequality:*

$$e^{-1}\bar{\delta}_T \leq \tilde{\delta}_T \leq k(1 - e^{-k}) \cdot \bar{\delta}_{kT} + 2e^{-k} \tag{6.17}$$

Proof: Suppose we want to simulate $\bar{P}_T$ by $\tilde{P}_T$. Scaling the distribution $\bar{\mu}_T$ by $\alpha := 1/e$ allows us to "fit it inside" the distribution $\tilde{\mu}_T$ (i.e., $e^{-1}\bar{\mu}_T \leq \tilde{\mu}_T$ pointwise), so we can express $\tilde{\mu}_T$ as the probabilistic combination $\alpha\bar{\mu}_T + (1-\alpha)\nu$ for some distribution ν, so that

$$\begin{aligned} \tilde{P}_T = E_{t \leftarrow \tilde{\mu}_T}[P_t] &= \alpha E_{\bar{\mu}_T}[P_t] + (1-\alpha)E_\nu[P_t] \\ &= \alpha \bar{P}_T + (1-\alpha)Q \qquad (6.18) \end{aligned}$$

where Q is stochastic with uniform stationary distribution. It follows that

$$||\tilde{P}_T|_{u^\perp}||_2 \leq 1/e||\bar{P}_T|_{u^\perp}||_2 + (1-1/e)||Q|_{u^\perp}||_2 \qquad (6.19)$$

which implies that $\tilde{\delta}_T \geq 1/e \cdot \bar{\delta}_T$ since $||Q|_{u^\perp}||_2 \leq 1$.

Suppose we want to simulate $\tilde{P}_T$ by $\bar{P}_{kT}$. Then the basic approach is the same, but since the support of $\tilde{\mu}_T$ is not compact we have to be careful. Scaling the distribution $\tilde{\mu}_T$ by $\beta := 1/k$ allows us to fit it inside the distribution $\bar{\mu}_{kT}$ up to the point $t = kT$, and the probability mass in $\tilde{\mu}_T$ past $t = kT$ is only $\Pr_{t \leftarrow \tilde{\mu}_T}[t > kT] = e^{-k}$. So we can write

$$\tilde{\mu}_T = (1-e^{-k}) \cdot \tilde{\mu}_T^{head} + e^{-k} \cdot \tilde{\mu}_T^{tail} \qquad (6.20)$$

where $\tilde{\mu}_T^{head}$ and $\tilde{\mu}_T^{tail}$ are the conditional distributions of $\tilde{\mu}_T$ such that $t \leq kT$ and $t > kT$, respectively; thus,

$$\tilde{P}_T = (1-e^{-k}) \cdot \tilde{P}_T^{head} + e^{-k} \cdot \tilde{P}_T^{tail} \qquad (6.21)$$

where $\tilde{P}_T^{head}$ and $\tilde{P}_T^{tail}$ are the expectations of P_t with respect to $\tilde{\mu}_T^{head}$ and $\tilde{\mu}_T^{tail}$, respectively. Since we can fit $\tilde{\mu}_T^{head}$ inside $\bar{\mu}_{kT}$ if we scale it by $1/k$, we can write

$$\bar{P}_{kT} = \frac{1}{k}\tilde{P}_T^{head} + (1-\frac{1}{k})Q \qquad (6.22)$$

where Q is stochastic with uniform stationary distribution. The above equations yield:

$$\bar{P}_{kT} = \frac{1}{k(1-e^{-k})}(\tilde{P}_T - e^{-k}\tilde{P}_T^{tail}) + (1-\frac{1}{k})Q \qquad (6.23)$$

From the triangle inequality, $||\bar{P}_{kT}|_{u^\perp}||_2$ is at most:

$$\frac{||\tilde{P}_T|_{u^\perp}||_2 + e^{-k}||\tilde{P}_T^{tail}|_{u^\perp}||_2}{k(1-e^{-k})} + (1-\frac{1}{k})||Q|_{u^\perp}||_2 \qquad (6.24)$$

Rearranging terms and simplifying, we have

$$\frac{(1-||\tilde{P}_T|_{u^\perp}||_2) - 2e^{-k}}{k(1-e^{-k})} \leq 1 - ||\bar{P}_{kT}|_{u^\perp}||_2 \qquad (6.25)$$

from which the lemma follows. ■

Say that the T'-repeated quantum walk *threshold-mixes* if the Markov chain $(\hat{P}_T)^{T'}$ it generates mixes (to the uniform distribution, in this case) in time $O(1)$. If the T'-repeated quantum walk of length T threshold-mixes, then it solves the spatial sampling problem using $O(T \cdot T')$ updates. The following theorem is shown in [90]:

Theorem 6.5 *If the T'-repeated quantum walk $\langle H_P, \bar{\mu}_T \rangle$ threshold-mixes, then the $T' \cdot O(\log N)$-repeated quantum walk $\langle H_P, \tilde{\mu}_T \rangle$ threshold-mixes. Conversely, if the T'-repeated quantum walk $\langle H_P, \tilde{\mu}_T \rangle$ threshold-mixes, then the $T' \cdot O(\log T' \log N)$-repeated quantum walk $\langle H_P, \bar{\mu}_{T \cdot O(\log T')} \rangle$ threshold-mixes.*

Proof: To see (a), note that our assumption implies that $\bar{P}_T$ mixes in time T'. Therefore, $\bar{\delta}_T = \Omega(1/T')$ by Theorem 4.2, and from Lemma 6.4 it follows that $\tilde{\delta}_T = \Omega(1/T')$. Applying Theorem 4.2 again, we obtain for $\tilde{P}_T$ a mixing time of $O(T' \log N)$.

The proof of (b) is almost as straightforward. Our assumption implies that $\tilde{P}_T$ mixes in time T', so $\tilde{\delta}_T = \Omega(1/T')$ by Theorem 4.2. Set k to be the smallest integer for which $\tilde{\delta}_T \geq 3e^{-k}$; in particular, $k = \Theta(\log \tilde{\delta}_T^{-1}) = O(\log T')$. By Lemma 6.4:

$$\bar{\delta}_{kT} \geq \frac{1}{k(1-e^{-k})}(\tilde{\delta}_T - 2e^{-k}) \geq \frac{1}{k(1-e^{-k})}(e^{-k}) \tag{6.26}$$

Asymptotically, the right hand side is:

$$\Theta\left(\frac{\tilde{\delta}_T}{\log \tilde{\delta}_T^{-1}}\right) = \Theta\left(\frac{1}{T' \log T'}\right) \tag{6.27}$$

Applying Theorem 4.2 again, we obtain for $\bar{P}_{kT}$ a mixing time of $O(T' \log T' \log N)$. ■

6.4 Quantum mixing on the torus

We turn our attention now to the special case in which P is the simple random walk on the d-dimensional periodic lattice (torus) $\mathbb{Z}_n^d$. First we prove an *approximate ergodic theorem* for the continuous-time quantum walk $\langle H_P, \bar{\mu}_T \rangle$: its time-average limit (the distribution $\Pi(\cdot, x)$) is approximately equivalent to its space-average (the uniform distribution). Thus, it converges to the same limit as its classical counterpart. Then we prove a quantum speedup for the mixing time. More precisely, we show that the T'-repeated quantum walk solves the spatial sampling problem using $T \cdot T' = O(nd \log d)$

update queries – beating the upper bound (6.3) – whereas the classical random walk simulation uses $O(n^2 d \log d)$ update queries.

6.4.1 An approximate ergodic theorem

Let Π be the stochastic matrix (6.7) generated by the quantum walk $\langle H_P, \bar{\mu}_T \rangle$ on the torus $\mathbb{Z}_p^d$ in the limit $T \to \infty$, where $p > 4d$ is prime. We show that every matrix entry of Π is bounded from below by $\Omega(1/p^d)$ [91]. Among other things, this implies (by Propositions 6.9, 6.10) that Π mixes in time $O(1)$.

First we bound the eigenvalue multiplicities of P.

Lemma 6.6 *Let P be the simple random walk on $\mathbb{Z}_p^d$ with $d \geq 1$ fixed and $p > 4d$ prime. Then each of the C_j in Theorem 6.2 consists of all $l \in \mathbb{Z}_p^d$ equivalent to a single $k \in \mathbb{Z}_p^d$ up to permutation and signing of the coordinates.*

Proof: Let $\omega = e^{2\pi i/p}$. Suppose $\lambda_k = \lambda_l$ and l is not equivalent to k. Then the vanishing sum

$$\sum_{j=1}^{d} \omega^{k_j} + \omega^{-k_j} - \omega^{l_j} - \omega^{-l_j} = 0 \tag{6.28}$$

is *not* simply $2d$ vanishing sums of length two $\omega^{k_j} - \omega^{k_j}$, $\omega^{-k_j} - \omega^{-k_j}$ over $j : 1 \leq j \leq d$. Observe that if we cannot decompose the above sum into length-two vanishing sums *in precisely this way*, then we cannot do so *at all* (since $\omega^{k_j} + \alpha = 0 \Rightarrow \alpha = -\omega^{k_j}$ is a $2p$th root of unity but *not* a pth root of unity). Hence there exists a minimal vanishing sum of length m ($3 \leq m \leq 4d$) of roots of unity ζ_i (without loss of generality, we may assume $\zeta_1 = 1$) of common order $r = p$. By Theorem 5 of Conway and Jones [39], $\sum_{s \, prime: \, s|r} (s-2) \leq m - 2$; so we have $p - 2 \leq 4d - 2$. ■

Next we prove a lemma which will be used to bound an exponential sum from below.

Lemma 6.7 *For any $y \in \mathbb{Z}_n^d$, there are $\Omega(n)$ different $x \in \mathbb{Z}_n$ satisfying $xy_i \bmod n \in [-n/8d, n/8d]$ for all $i : 1 \leq i \leq d$.*

Proof: Consider the map $f : \mathbb{Z}_n \to \mathbb{Z}_n^d$ given by $x \mapsto (xy_1, \ldots, xy_d)$. Thinking of $\mathbb{Z}_n^d$ as being divided into subgrids of side length n/m (with one of them, H_0, centered at 0), it is clear that one such subgrid (call it H) must contain at least n/m^d of the points in the image of f. Let $x'y$ be any of the image points in H. Then there are at least n/m^d different x for which $(x - x')y_i \in [-n/m, n/m]$ for all i. Let $m = 8d$; then there are at least $n/(8d)^d$ such x. ■

Finally we establish the main theorem [91].

Theorem 6.8 *Let P be the simple random walk on $\mathbb{Z}_p^d$ with $d \geq 1$ fixed, p prime, and $N = p^d$. Then each entry of Π is bounded below by $\Omega(1/N)$.*

Proof: Label the spectrum of P using indices $k \in \mathbb{Z}_p^d$ as follows:

$$\lambda_k := \frac{1}{d}\sum_{j=1}^{d} \cos(2\pi k_j/p) \qquad |\phi_k\rangle := \frac{1}{\sqrt{N}} \sum_{x \in \mathbb{Z}_p^d} e^{2\pi i k\cdot x/p}|x\rangle \tag{6.29}$$

Since $\mathbb{Z}_p^d$ is vertex-transitive, it suffices to show that $\Pi(y, 0) = \Omega(1/N)$; or, from Theorem 6.2:

$$\sum_j |\sum_{k \in C_j} e^{2\pi i k\cdot y/p}|^2 = \Omega(N) \tag{6.30}$$

This is a consequence of the preceding two lemmas: by Lemma 6.7, at least $(p/(8d)^d)^d = \Omega(N)$ of the $k \in \mathbb{Z}_p^d$ are such that: (i) $k_j y_i \bmod p \in [-p/8d, p/8d]\ \forall i, j$, thus (ii) $l \cdot y := \sum_i l_i y_i \bmod p \in [-p/8, p/8]$ for all l equivalent to k up to permutation and signing of the coordinates. So by Lemma 6.6, which in particular shows that $|C_j| \leq 2^d d!$ for all j, it follows that $(p/(8d)^d)^d/2^d d! = \Omega(N)$ of the C_j give a sum of at least 1. ■

6.4.2 Quantum speedup of the mixing time

The simple random walk P on the torus $\mathbb{Z}_n^d$ has spectral gap $\delta = \Theta(\frac{1}{dn^2})$ and mixing time $\Theta(n^2 d \log d)$. This stays under the target upper bound (6.3) only when $\mathbb{Z}_n^d$ is quite high-dimensional: in particular, when d is roughly of order n^2 or larger. That the low-dimensional case is a bottleneck is unsurprising considering the discussion immediately preceding (6.3).

We show that the T'-repeated quantum walk $\langle H_P, \bar{\mu}_T\rangle$ mixes in time $O(1)$ for $T = O(dn)$ and $T' = O(\log d)$, for a "quantum mixing time" of $T \cdot T' = O(nd \log d)$ [90].

This beats the target upper bound (6.3), and for $d = 1$ it proves a conjecture of Kendon and Tregenna [65] based on numerical experiments.[5] For $d \geq 1$, it extends the results of Fedichkin et al. [47, 101, 102] by confirming $O(n)$ and $O(d \log d)$ scaling (suggested by their analytical estimates and numerical experiments in regimes of both high and low decoherence) of the fastest-mixing walk, which they conjectured to be decoherent (repeatedly measured) rather than unitary (once measured).

Previously, mixing speedups were known only for the unitary quantum walks of Nayak et al. [81, 17] and Aharonov et al. [4] (on the cycle) and of Moore and Russell [79] (on the hypercube). (See Table 6.1.) Thus, our result shows that introducing a small amount of decoherence to a quantum walk can simultaneously force convergence to the uniform distribution while preserving a quantum mixing speedup, an advantageous combination for algorithmic applications.[6]

Graph	Spectral estimate (6.3)	Random walk	Quantum walk
$\mathbb{Z}_n$	$O(n \log n)$	$O(n^2)$	$O(n)$ [81, 17, 4]
$\mathbb{Z}_n^d$	$O(d\sqrt{d}n \log n)$	$O(n^2 d \log d)$	$O(nd \log d)$ [90]
$\mathbb{Z}_2^d$	$O(d\sqrt{d})$	$O(d \log d)$	$O(d)$ [79]

Table 6.1: Known mixing upper bounds for random vs. quantum walks.

First we need to establish a few properties of Markov chains. Define the *maximum pairwise column distance* of a Markov chain P by:

$$d'(P) := \max_{x,x'} ||P(\cdot, x) - P(\cdot, x')||_1 \tag{6.31}$$

It is related to the matrix ℓ_1 distance

$$d(P) := ||P - P^{\infty}||_1 \tag{6.32}$$

[5] In fact this is not quite true: Kendon and Tregenna [65] conjectured this for the discrete-time Hadamard walk. We prove it for both walks.

[6] Alagic and Russell [10] exhibit a decoherent quantum walk on the hypercube which converges to the uniform distribution and preserves the quantum mixing speedup proven by Moore and Russell [79], but in contrast to our work, it is their single (carefully chosen) final measurement that *forces* uniform convergence, not the decoherence itself (which only adds "noise" that if small enough does not destroy the speedup).

(assuming P is irreducible) by the inequality

$$\frac{1}{2}d(P) \leq d'(P) \leq d(P) \tag{6.33}$$

and it has the submultiplicativity property [12]:

$$d'(P^{t+t'}) \leq d'(P)^{t+t'} \tag{6.34}$$

The next two propositions can be used to estimate the mixing time of P given a common lower bound on most of the entries in each column.

Proposition 6.9 *If $d'(P) \leq \alpha < 1$, then $d(P^t) \leq \epsilon$ for some $t = O(\log \epsilon^{-1} / \log \alpha^{-1})$.*

Proof: If $d'(P) \leq \alpha$, then the rightmost inequality of

$$d(P^t) \leq 2d'(P^t) \leq 2d'(P)^t \leq \epsilon \tag{6.35}$$

holds provided that t is small enough so that:

$$t \log 1/d'(P) \leq \log 2/\epsilon \tag{6.36}$$

In particular, this holds for $t = O(\log \epsilon^{-1} / \log \alpha^{-1})$. ■

To prove that P mixes in time $O(1)$, it suffices to show that $d'(P)$ is at most some constant less than one. A sufficient condition for this property to hold is the following:

Proposition 6.10 *Suppose $\beta > \frac{1}{2}$ and $\gamma > 0$. If P is $N \times N$ and has at least βN entries in each column bounded below by γ / N, then $d'(P) \leq 1 - \gamma(1 - 2(1 - \beta)) < 1$.*

Proof: Recall that for any two distributions p and q we have $\frac{1}{2}||p - q||_1 = 1 - \sum_k \min\{p_k, q_k\}$. It follows that:

$$d'(P) \leq 1 - (1 - 2(1 - \beta))N \cdot \frac{\gamma}{N} = 1 - \gamma(1 - 2(1 - \beta)). \tag{6.37}$$

which proves the claim. ■

Define the point distribution $\delta_T(t) := \delta(t - T)$ where δ is the delta function, the (discrete-time) uniform distribution $\bar{\nu}_T := \frac{1}{T}\chi_{[0..T-1]}$, and the geometric distribution

$\tilde{\nu}_T(t) := \frac{1}{T}(1 - \frac{1}{T})^t$ – the discrete-time (Bernoulli interarrival) counterpart to the exponential (Poission interarrival) distribution $\tilde{\mu}_T$. The following lemma shows that a decoherent quantum walk using any of these measurement rules mixes quickly on the cycle $\mathbb{Z}_n$.

Lemma 6.11 *Let P be the simple random walk on $\mathbb{Z}_n$, the cycle on $n \geq 2$ vertices. The continuous-time quantum walks $\langle H_P, \omega_T \rangle$ with $\omega \in \{\delta, \bar{\mu}, \tilde{\mu}\}$ threshold-mix for any $T \in \mathcal{I} := [\frac{2}{3} \cdot \frac{n}{2}, \frac{n}{2}]$, and the Hadamard walks $\langle U_{Had}, \omega_T \rangle$ with $\omega \in \{\bar{\nu}, \tilde{\nu}\}$ threshold-mix for any $T \in \mathcal{J} := [\frac{2}{3} \cdot \frac{n}{\sqrt{2}}, \frac{n}{\sqrt{2}}]$.*

Proof: Consider first the continuous-time walk. To prove that it threshold-mixes with any of the measurement rules $\omega \in \{\delta, \bar{\mu}, \tilde{\mu}\}$ for any $T \in \mathcal{I}$, it suffices by Proposition 6.9 to show that for every $t \in \mathcal{I}' := [\frac{3}{5} \cdot \frac{n}{2}, \frac{4}{5} \cdot \frac{n}{2}]$, $d'(P_t)$ is bounded below one by a positive constant, where P_t is the Markov chain generated with measurement rule $\omega = \delta$. Indeed, this easily implies that $d'(\bar{P}_T)$ and $d'(\tilde{P}_T)$ are bounded below one by a smaller positive constant, where $\bar{P}_T$ and $\tilde{P}_T$ are the Markov chains generated with measurement $\omega = \bar{\mu}$ and $\omega = \tilde{\mu}$, respectively.

Let $|\phi_t\rangle$ and $|\psi_t\rangle$ be the wavefunctions at time t for the continuous-time walks on $\mathbb{Z}$ and $\mathbb{Z}_n$, respectively, starting from the origin (without loss of generality, since $\mathbb{Z}$ and $\mathbb{Z}_n$ are vertex-transitive). Then for each $\bar{y} \in \mathbb{Z}_n$ we have:

$$\langle \bar{y} | \psi_t \rangle = \sum_{y \equiv \bar{y} \bmod n} \langle y | \phi_t \rangle \tag{6.38}$$

Refer back to the estimates (a) and (b) of (6.5). For every $t < \frac{4}{5} \cdot \frac{n}{2}$, estimate (a) implies that the only term in the above summand that is non-negligible is the $\langle y | \phi_t \rangle$ with $|y| < \frac{n}{2}$ (call it $\hat{y}$ and note that $\bar{y} \leftrightarrow \hat{y}$ is a 1-1 correspondence), so we can use estimate (b) to conclude that up to a negligible correction

$$|\langle \bar{y} | \psi_t \rangle| \approx |\langle \hat{y} | \phi_t \rangle| = \Theta(1/\sqrt{n}) \tag{6.39}$$

for every $|\hat{y}| \gg 1$ and $t > (1 + \epsilon) \cdot |\hat{y}|$. In particular, the nearly $\frac{3}{5}n$ different $\bar{y}$ with $1 \ll |\hat{y}| \leq \frac{3}{5} \cdot \frac{n}{2}$ satisfy $|\langle \bar{y} | \psi_t \rangle| = \Omega(1/\sqrt{n})$, and therefore $P_t(\bar{y}, \bar{0}) = \Omega(1/n)$, for every $t \in \mathcal{I}'$. So by Proposition 6.10, $d'(P_t)$ is bounded below one by a positive constant.

In the case of the Hadamard walk (see [81, 17, 4] for its definition), the wavefunction is no longer characterized by Bessel functions, but it retains the same essential asymptotic spreading behavior as its continuous-time counterpart [81, 17], and the argument above works with little modification. A caveat is the emergence of a parity problem: if n is even, then $\mathbb{Z}_n$ is bipartite and the wavefunction is supported only on vertices of the same parity at each integer timestep. Hence the Hadamard walk with $\omega = \delta$ threshold-mixes only on vertices of the same parity, but with time-averaged measurement $\omega = \bar{\nu}$ or $\omega = \tilde{\nu}$ parity is broken and threshold-mixing occurs on the entire vertex set.

Although the argument above relies on the asymptotic behavior of the wavefunction as $n \to \infty$, this is clearly the difficult case: if n is bounded, then it suffices to show only that there exists a time (or a pair of consecutive timesteps, in the case of the Hadamard walk) in which the wavefunction is supported on at least 2/3 of the vertices. ■

For the Hadamard walk with $\omega = \tilde{\nu}$, Lemma 6.11 resolves a conjecture of Kendon and Tregenna [65] based on numerical experiments.

Let G^d denote the dth (Cartesian) power of a graph G. Examples are the d-dimensional standard lattice (the dth power of a line) and the d-dimensional periodic lattice (the dth power of a cycle). Let P and P' be the simple random walks on G and G^d, respectively. The following theorem shows how to extend a threshold-mixing result from H_P to $H_{P'}$ [90].

Theorem 6.12 *Suppose the continuous-time quantum walk $\langle H_P, \delta_T \rangle$ threshold-mixes. Then the $O(\log d)$-repeated quantum walk $\langle H_{P'}, \delta_{dT} \rangle$ threshold-mixes.*

Proof: The Hamiltonian $H' = P'$ is related to the Hamiltonian $H = P$ by the identity:

$$H' = \frac{1}{d} \sum_{j=1}^{d} I^{\otimes(j-1)} \otimes H \otimes I^{\otimes(d-j)} \tag{6.40}$$

Since H' commutes with the identity I, which can introduce at most a global phase factor to the system, the Markov chain P'_t generated by the walk $\langle H_{P'}, \delta_t \rangle$ is the dth tensor power of the Markov chain $P_{t/d}$ generated by the walk $\langle H_P, \delta_{t/d} \rangle$. By assumption, $d'(P_T) \leq \alpha$ for some constant $\alpha < 1$. By Proposition 6.9, we can choose $T' = O(\log d)$

to ensure that:

$$d(P_T^{T'}) \leq d(P_T^{T'}) \leq \frac{1}{6d^2} \tag{6.41}$$

Then at least $n\sqrt[d]{2/3}$ entries in each column of $P_T^{T'}$ are bounded below by $\frac{1-1/2d}{n}$, otherwise we would have the contradiction

$$\begin{aligned} d(P_T^{T'}) &= 1 - \sum_y \min\{P_T^{T'}(y,x), \frac{1}{n}\} \\ &> (1 - \sqrt[d]{2/3})\frac{1}{2d} \geq \frac{1}{6d^2} \end{aligned} \tag{6.42}$$

where the first equation uses the identity $\frac{1}{2}||p-q||_1 = 1 - \sum_k \min\{p_k, q_k\}$ for distributions p and q, and the last inequality uses simple algebra along with the fact that for any $d \geq 1$:

$$(1 - \frac{1/3}{d})^d \geq \frac{2}{3} \tag{6.43}$$

Since $(P'_{Td})^{T'} = (P_T^{T'})^{\otimes d}$, at least $(n\sqrt[d]{2/3})^d = \frac{2}{3}n^d$ of the entries in each column of $(P'_{Td})^{T'}$ are bounded below by $(\frac{1-1/2d}{n})^d \geq \frac{1}{2n^d}$. It follows from Proposition 6.10 that $(P'_{Td})^{T'}$ threshold-mixes in time $O(1)$. ■

We have the following corollary for the dth power $\mathbb{Z}_n^d$ of the cycle $\mathbb{Z}_n$.

Corollary 6.13 *Let P' be the simple random walk on $\mathbb{Z}_n^d$, the d-dimensional torus with $n \geq 2$ vertices per side. The $O(\log d)$-repeated continuous-time quantum walk $\langle H_{P'}, \omega_{nd/2}\rangle$ with $\omega \in \{\delta, \bar{\mu}, \tilde{\mu}\}$ threshold-mixes.*

Proof: Combining Lemma 6.11 with Theorem 6.12, we conclude that $\langle U_{ct}(P(\mathbb{Z}_n^d)), \delta_T\rangle$ threshold-mixes for any $T \in [\frac{2}{3} \cdot \frac{nd}{2}, \frac{nd}{2}]$. It is easy to see (cf. Lemma 6.11) that this is sufficient to imply the stated corollary not only for the measurement $\omega = \delta$ but also for the time-averaged measurements $\bar{\mu}$ and $\tilde{\mu}$. ■

Theorem 2.2 implies the quantum walk Hamiltonian H_P (which is sparse) can be simulated efficiently by a quantum circuit – in particular, without destroying the quantum speedup over the classical mixing time. Nevertheless, one wonders whether the discrete-time Grover walk W_P mixes just as well as H_P. There is some positive evidence for the torus $\mathbb{Z}_n^2$ [55, 106, 72] and a proof for the hypercube $\mathbb{Z}_2^n$ [79].

Part III

Quantum ground state problems

Chapter 7

Adiabatic quantum computing

Many of the quantum walk algorithms we have seen heretofore can be recast naturally as *time-independent quantum* procedures for finding the ground state $|x_*\rangle$ of a *classical Hamiltonian* with diagonal entries $\langle x|H|x\rangle = f(x)$, where $f : \mathbb{B}^n \to \mathbb{R}$ and $x_* := \operatorname{argmin} f(x)$. Recall that simulated annealing solves this same problem using a *time-dependent classical* procedure. An *adiabatic* (or quantum annealing) algorithm finds the ground state of a *quantum Hamiltonian* using a *time-dependent quantum* procedure. As one might expect, this is a powerful algorithmic technique – so much so, in fact, that *any* quantum circuit can be simulated efficiently by an adiabatic algorithm. Thus, an "adiabatic quantum computer" is no weaker than one built from quantum circuits.

7.1 Simulated annealing redux

Recall from Chapter 4 the simulated annealing algorithm. We are given a classical (diagonal) Hamiltonian $H : \mathbb{B}^n \to \mathbb{B}^n$ whose ground state energy

$$\lambda_* := \min_{x \in \mathbb{B}^n} \langle x|H|x\rangle \tag{7.1}$$

we want to compute. Let $|x_*\rangle$ be a classical ground state. If we reparametrize the Metropolis process by its *inverse temperature* $\beta = \ln \lambda$ (where λ is its activity), then it converges to the Gibbs distribution

$$\pi(x) = \frac{1}{Z(\beta)} \langle x|e^{-\beta H}|x\rangle \tag{7.2}$$

whose numerator is the *Boltzmann factor* and whose denominator is the partition function:

$$Z(\beta) = \operatorname{tr}(e^{-\beta H}) \tag{7.3}$$

Physically, $\pi(x)$ is the probability that the system, in equilibrium and at temperature $T = 1/\beta$, will be found in the classical state $|x\rangle$. At high temperatures ($T \to \infty$), $\pi(x)$ is approximately uniform; at low temperatures ($T \to 0$), it *freezes* to the ground state $|x_*\rangle$ of H.[1] The simulated annealing algorithm starts at any classical state $|x\rangle$ and runs the Metropolis process while gradually lowering the temperature according to a *cooling schedule*. Using perturbation theory, it can be shown that the algorithm outputs the state $|x_*\rangle$ with high probability if the cooling schedule is sufficiently gradual.

7.2 Adiabatic algorithms for search and 3-SAT

A quantum adiabatic algorithm [44] interpolates between two fixed Hamiltonians along an adiabatic path $H(s)$, much like the cooling schedule in classical simulated annealing. Theorem 2.1 guarantees that the algorithm will maintain the ground state $|\phi(s)\rangle$ of $H(s)$ if it proceeds sufficiently slowly relative to the spectral gap of $H(s)$. There is evidence that quantum adiabatic algorithms can sometimes "tunnel through" energy barriers surrounding local minima more quickly that classical simulated annealing algorithms can cross them [42].

The adiabatic computing framework was developed by Farhi et al. [44], who gave the following algorithm for the 3-satisfiability (3-SAT) problem from Chapter 4. Let $H_1 : \mathbb{B}^n \to \mathbb{B}^n$ be the classical Hamiltonian in which $\langle x|H_1|x\rangle$ gives the number of clauses violated in the Boolean formula ξ by the variable assignment $x = x_1 x_2 \cdots x_n \in \mathbb{B}^n$. This is the *problem Hamiltonian.* It is a *local* Hamiltonian in that it can be expressed in the bitwise product form

$$H_1 = \sum_{j=1}^{m} H_{C_j} \otimes I_{[n]\setminus C_j} \tag{7.4}$$

where $C_j \subseteq [n]$ contains the indices of the three bits in clause j, $H_{C_j} : \mathbb{B}^3 \to \mathbb{B}^3$ is the Hamiltonian acting on the bits in clause j whose diagonal entry $\langle x_i, x_{i'}, x_{i''}|H|x_i, x_{i'}, x_{i''}\rangle$ is 1 if the partial assignment $x_i, x_{i'}, x_{i''}$ violates clause C_j (and 0 otherwise), and $I_{[n]\setminus C_j}$ is the identity operator acting on the bits not in clause j. If there is a unique assignment

[1] That it does not concentrate *entirely* on $|x_*\rangle$ unless $T = 0$ is essentially the third law of thermodynamics.

x_* satisfying ξ, then the unique ground state of H_1 is $|x_*\rangle$. Let the *initial Hamiltonian* $H_0 : \mathcal{B}^{\otimes n} \to \mathcal{B}^{\otimes n}$ be given by

$$H_0 = \sum_{j=1}^{m} H'_{C_j} \otimes I_{[n]\setminus C_j} \tag{7.5}$$

where H'_{C_j} is a sum of three single-qubit Hamiltonians $H'_{\{i\}}$, $H'_{\{i'\}}$, $H'_{\{i''\}}$ acting on the bits x_i, $x_{i'}$, $x_{i''}$ by

$$H'_{\{i\}} = \frac{1}{2}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{7.6}$$

in the classical basis. Its unique ground state is the uniform superposition $|u\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\mathbb{B}^n} |x\rangle = H^{\otimes n}|0^n\rangle$, which is easily preparable.

Farhi et al. [44] suggest that there is an adiabatic algorithm $H(s) := (1-s)H_0 + sH_1$ with an efficient delay schedule $\tau(s)$ (2.4) – i.e., polynomial in n – that solves some 3-SAT instances thought to be hard for classical algorithms. A proof to this effect would require bounding the spectral gap of $H(s)$ (by Theorem 2.1), but techniques for doing so are quite limited.

There is at least one example of an adiabatic algorithm whose efficiency we *can* prove. Roland and Cerf [93] and van Dam et al. [108] designed an adiabatic algorithm for the search problem and proved that it achieves the same quantum speedup as Grover's algorithm. Using the initial and problem Hamiltonians

$$H_0 := H^{\otimes n}(I - |0^n\rangle\langle 0^n|)H^{\otimes n} \qquad H_1 := I - |x_*\rangle\langle x_*| \tag{7.7}$$

where x_* is the unique solution $f^{-1}(1)$ of a predicate $f : \mathbb{B}^n \to \mathbb{B}$, they show:

Theorem 7.1 *There is an adiabatic algorithm $H(s) = (1-s)H_0 + sH_1$ computing x_* in time $O(\sqrt{2^n})$.*

Proof: To estimate the required delay schedule $\tau(s)$, we need to bound the spectral gap $\delta(s)$. The algorithm, like Grover's, acts within a two-dimensional subspace spanned by the initial ground state $|u\rangle = H^{\otimes}|0^n\rangle$ and the final ground state $|x_*\rangle$. A simple calculation [108] shows that the spectral gap inside this subspace is:

$$\delta(s) = \sqrt{\frac{2^n + 4(2^n-1)(s^2-s)}{2^n}} \tag{7.8}$$

Applying Theorem 2.1, we find that the total time

$$\int_{s=0}^{1} \frac{ds}{g(s)^2} = 2^n \frac{\tan^{-1}\sqrt{2^n-1}}{\sqrt{2^n-1}} = O(\sqrt{2^n}) \tag{7.9}$$

is indeed achievable. ■

The continuous-time walk algorithm $e^{-i(H_0+H_1)t}$ often performs as well as the adiabatic algorithm $H(s) = (1-s)H_0 + sH_1$. For example, Grover's search speedup can be reproduced [45], as can several discrete-time quantum walk speedups [37, 36].

7.3 Universality of adiabatic quantum computing

In the adiabatic algorithms for search and 3-SAT, the problem Hamiltonian is classical and the initial Hamiltonian is diagonal in an efficiently constructible (e.g., Hadamard) basis. From the remarks following Theorem 2.2, it follows that these adiabatic algorithms are efficiently simulable by quantum circuits. What about the converse: Can any quantum circuit be implemented efficiently as an adiabatic algorithm with local Hamiltonians? Aharonov et al. [8] showed the answer is yes. The problem Hamiltonian in their algorithm is a clever one borrowed from Kitaev [68].

7.3.1 Kitaev's Hamiltonian

Kitaev defined the following reduction from a quantum circuit U to a time-independent local Hamiltonian H.[2] Let $U = U_L \cdots U_1$ be the n-qubit circuit, each gate of which acts on at most two qubits. Define the Hamiltonian H on n *circuit* qubits and L *clock* qubits by

$$H := H_{input} + H_{prop} + H_{clock} \tag{7.10}$$

where

$$H_{input} := \sum_{i=1}^{n} |\neg x_i\rangle\langle\neg x_i|_i \otimes |0\rangle\langle 0|_1 \tag{7.11}$$

[2] Kitaev credits the idea of considering such a reduction to Feynman [49].

is the *input* term,

$$\begin{aligned} H_{prop} \;\; := \;\; & \sum_{l=1}^{L} I \otimes |100\rangle\langle 100|_{l-1,l,l+1} - U_l \otimes |110\rangle\langle 100|_{l-1,l,l+1} \\ & -U_l^\dagger \otimes |100\rangle\langle 110|_{l-1,l,l+1} + I \otimes |110\rangle\langle 110|_{l-1,l,l+1} \end{aligned} \tag{7.12}$$

is the *propagator* term, and

$$H_{clock} := I \otimes \sum_{l=1}^{L} |01\rangle\langle 01|_{l,l+1} \tag{7.13}$$

is the (unary) *clock* term. The first and second halves of each tensor product refer to the circuit and clock qubits, respectively, and the notation $|a\rangle\langle a|_i$ denotes the projection onto the subspace in which the ith qubit is $|a\rangle$. The purpose of the Hamiltonian is to assign an *energy penalty* to any quantum state encoding an incorrect circuit state or state transition on input x, so that only the correct circuit is encoded in its ground state. Note that H is 5-local (cf. (8.4)) since each local term involves at most one 2-local gate U_l and at most one 3-local clock penalty.

The following lemma is abstracted from Aharonov et al. [8].

Lemma 7.2 *The ground state of H (with eigenvalue zero) is the* history state

$$|\eta\rangle := \frac{1}{\sqrt{L+1}} \sum_{l=0}^{L} U_l \cdots U_1 |x\rangle \otimes |1^l 0^{L-l}\rangle \tag{7.14}$$

and the spectral gap is $\Theta(1/L^2)$.

Proof: That $|\eta\rangle$ lies in the nullspace is easy to check. The eigenvalue zero is the smallest, as all terms are positive semidefinite. It remains to estimate the spectral gap.

Let $\mathcal{S}$ denote the subspace of dimension $(L+1)\cdot 2^n$ spanned by all legal clock states – i.e., all clock states of the form $|1^l 0^{L-l}\rangle$, representing time step l in unary. Note that $\mathcal{S}$ is invariant under H and so $H = \mathcal{S} \oplus \mathcal{S}^\perp$ and we can analyze the spectrum within these two subspaces. Note that all eigenvalues of $H|_{\mathcal{S}^\perp}$ are at least one since H_{clock} penalizes illegal clock states by this amount and all other terms are positive semidefinite.

Next we show that incorrect inputs are also sufficiently penalized. Write $\mathcal{S}$ as the direct sum of 2^n orthogonal subspaces $\mathcal{S}_0, \ldots, \mathcal{S}_{2^n-1}$ where $\mathcal{S}_y$ is the $L+1$ dimensional subspace spanned by the vectors $U_l \cdots U_1 |y\rangle \otimes |1^l 0^{L-l}\rangle$. Then each eigenvalue of $H|_{\mathcal{S}_y}$ for $y \neq x$ is at least one, since at least one input bit is penalized and all other terms are positive semidefinite. Last we show that incorrect transitions are sufficiently penalized; i.e., the spectral gap of $H|_{\mathcal{S}_x} = H_{prop}|_{\mathcal{S}_x}$ is $\Theta(1/L^2)$. This can be seen by changing the basis of H_{prop} by the measuring operator

$$W = \sum_{l=0}^{L} U_L \cdots U_1 \otimes |1^l 0^{L-l}\rangle\langle 1^l 0^{L-l}| \tag{7.15}$$

to yield the matrix

$$W^\dagger H_{prop} W = I \otimes T \tag{7.16}$$

where T is the $(L{+}1)\times(L{+}1)$ tridiagonal matrix with principal diagonal $(\frac{1}{2}, 1, 1, \ldots, 1, \frac{1}{2})$ and non-principal diagonals $(-\frac{1}{2}, -\frac{1}{2}, \ldots, -\frac{1}{2})$. The matrix $P = I - T$ is that of a random walk on a line, whose spectral gap below one is $\Theta(1/L^2)$. Thus, this is the spectral gap of T above zero. ■

7.3.2 5-local adiabatic quantum computing

Aharonov et al. [8] showed that a polynomial-size quantum circuit computation can be realized as an efficient adiabatic algorithm using 3-local qubit interactions. We present the somewhat simpler 5-local case.

The following lemma is proved in Kitaev et al. [68].

Lemma 7.3 *Let H_1, H_2 be Hamiltonians with ground spaces $\mathcal{L}_1, \mathcal{L}_2$, ground state energies a_1, a_2, and spectral gaps at least Λ above $\mathcal{L}_1, \mathcal{L}_2$. Then the ground state energy of $H_1 + H_2$ is at least*

$$a_1 + a_2 + 2\Lambda \sin^2(\theta/2) \tag{7.17}$$

where θ is the angle between $\mathcal{L}_1$ and $\mathcal{L}_2$.

We now sketch the proof of Aharonov et al. [8]. Note that the algorithm's output state $|\psi_1\rangle$ is "close enough" to the circuit's output state $U|x\rangle$: The correct output can be found by repeating the procedure $O(1/L)$ times until the state $|1^L\rangle$ is observed in the clock register, then returning the contents of the the circuit register.

Theorem 7.4 *Given a quantum circuit U on n qubits with L two-qubit gates, there is an adiabatic algorithm $H(s) := (1-s)H_0 + sH_1$ where $H(s) : \mathcal{B}^{\otimes(n+L)} \to \mathcal{B}^{\otimes(n+L)}$ is a 5-local Hamiltonian whose input state is $|\psi_0\rangle = |x, 0^L\rangle$, whose output $|\psi_1\rangle$ satisfies $|\langle\psi_1|U|x, 0^L\rangle|^2 = \Theta(1/L)$, and whose running time is polynomial in L.*

Proof: We want to show that we can adiabatically efficiently generate a state of large overlap with the final state of a polynomial-size quantum circuit using Kitaev's 5-local Hamiltonian. Specifically, set $H_0 = H_{input} + H_{clockinit} + H_{clock}$, where

$$H_{clockinit} := I \otimes |1\rangle\langle 1|_1 \tag{7.18}$$

is a *clock initialization* term that ensures the clock's state starts out $|0^L\rangle$. Now set the problem Hamiltonian H_1 to Kitaev's Hamiltonian.

Clearly, the ground state of H_0 (with eigenvalue zero) is $|x\rangle \otimes |0^L\rangle$. By similar arguments as in Lemma 7.2, we have the direct sum decompositions of the whole space into $\mathcal{S} \oplus \mathcal{S}^\perp$ and the subspace $\mathcal{S}$ into $\mathcal{S}_0, \ldots, \mathcal{S}_{2^n-1}$, with the eigenvalues of H outside $\mathcal{S}_x$ at least one. $H_0|_{\mathcal{S}_x} = H_{clockinit}|_{\mathcal{S}_x}$ is represented by the diagonal matrix with diagonal $(0, 1, 1, \ldots, 1)$. Thus, the ground state is unique and has a spectral gap of one.

By Lemma 7.2, H_1 has the history state $|\eta\rangle$ as its unique ground state, with eigenvalue zero and spectral gap $\Omega(1/L^2)$.

That the spectral gap of $H(s)$ for any $s \in [0, 1]$ is $\Omega(1/L^3)$ is argued as follows. First, note that the spectral gap of $H|_{\mathcal{S}_x}(s)$ is $\Omega(1/L^2)$. This is because, as [8] show: (i) for $s < 1/3$, it is close enough to $H_0|_{\mathcal{S}_x}$ (whose spectral gap is one) for us to show a lower bound of $1/3$ on the spectral gap using Gerschgorin's theorem; and, (ii) for $s \geq 1/3$, it is close enough to $H_1|_{\mathcal{S}_x} = H_{prop}|_{\mathcal{S}_x}$ to obtain a lower bound of $\Omega(1/L^2)$ by applying Markov chain techniques.

Now let us extend the lower bound on the spectral gap of $H(s)$ from the subspace $\mathcal{S}_x$ to the entire space. Since the ground state energy of $H_{\mathcal{S}}(s)$ is at most $1/2$ (as one can easily check) and the eigenvalues of $H|_{\mathcal{S}^\perp}(s)$ are at least one, it suffices to show that the spectral gap of $H_{\mathcal{S}}(s)$ is at least $\Omega(1/L^3)$. We note that

$$H|_{\mathcal{S}_y}(s) = H|_{\mathcal{S}_x}(s) + H_{input}|_{\mathcal{S}_y} \tag{7.19}$$

and apply Lemma 7.3 above. The spectral gap of $H|_{\mathcal{S}_x}(s)$ above its ground space is $\Omega(1/L^2)$ as we showed before. The matrix of $H_{input}|_{\mathcal{S}_y}$ for $y \neq x$ is all zeros except for the top left corner entry which equals the Hamming distance between x and y, so the spectral gap above its ground space is at least one. Moreover, one can show [8] that the angle θ between the two ground spaces satisfies $\cos\theta \leq 1 - 1/L$. Therefore, Lemma 7.3 implies that the ground state energy of $H|_{\mathcal{S}_y}(s)$ is higher than that of $H|_{\mathcal{S}_x}(s)$ by at least $\Omega(1/L^3)$.

Since the history state $|\eta\rangle$ satisfies $|\langle\eta|U|x, 0^L\rangle|^2 = 1/L$, the theorem follows. ■

Theorem 7.4 was improved to 3-local qubit interactions and to 2-local 6-state spin interactions on the two-dimensional planar lattice by Aharonov et al. [8], to 2-local qubit interactions by Kempe et al. [62], to 2-local qubits interactions on the two-dimensional planar lattice by Oliveira and Terhal [83], and to 2-local 9-state spin interactions on the one-dimensional line by Aharonov et al. [5].

In light of the remarks concluding the previous subsection, one wonders whether *adiabatic* continuous-time evolution is truly necessary to simulate the quantum circuit efficiently. More precisely, does the *time-independent* evolution $e^{-i(H_0+H_1)t}$ also generate a state of large overlap with the ground state of Kitaev's Hamiltonian? This is an open question.

Chapter 8

The local Hamiltonian problem

The *local Hamiltonian problem* (LH-MIN) is to estimate the ground state energy of an n-qubit Hamiltonian given its decomposition into local interactions. It is a central problem of computational physics and a natural generalization of the maximum k-satisfiability problem (MAX-k-SAT). Kitaev showed that LH-MIN is complete for the complexity class QMA, the generalization of NP obtained by allowing the witness and verifier to be quantum. By looking at natural restrictions of LH-MIN, we can obtain complete problems for several complexity classes inside QMA and potentially arrive at a better understanding of their computational power.

8.1 Spin glasses and MAX-2-SAT

In classical statistical physics, a *spin glass* is an n-spin system exhibiting *frustration*: it has many excited states of locally minimal energy just above that of the ground state. A typical spin glass has the classical Hamiltonian

$$H = -\sum_{(i,j)\in E} H_{\{i,j\}} \otimes I_{[n]\setminus\{i,j\}} - \sum_{i\in V} H_{\{i\}} \otimes I_{[n]\setminus\{i\}} \tag{8.1}$$

acting locally on a graph $G = (V, E)$ connecting the spins $x = x_1 x_2 \cdots x_n \in \mathbb{B}^n$, where

$$\langle x_i, x_j | H_{\{i,j\}} | x_i, x_j \rangle = (-1)^{x_i + x_j} J_{ij} \tag{8.2}$$

is determined by the *interaction strength* J_{ij} and

$$\langle x_i | H_{\{i\}} | x_i \rangle = B_i \tag{8.3}$$

is the *external field.* In the simple model proposed by Ernst Ising in 1925, $J_{ij} = J$ and $B_i = B$ are constant, so the ground state is either $|0^n\rangle$ or $|1^n\rangle$ (i.e., all of the spins are “aligned”) and the ground state energy is simple to calculate. But if we specify arbitrary J_{ij} and B_i as input, deciding whether the ground state energy is above or below some threshold is NP-complete. This remains true even when G is the two-dimensional planar lattice. Indeed, MAX-2-SAT is almost a special case of this problem (requiring only slightly more general interaction terms) and can easily be reduced to it in polynomial time. Estimating the ground state energy of an arbitrary 2-local classical Hamiltonian on n spins *does* include MAX-2-SAT as a special case.

8.2 The computational complexity of LH-MIN

A k-local quantum Hamiltonian has the form

$$H = \sum_{e \in E} H_e \otimes I_{V \setminus e} \tag{8.4}$$

where $G = (V, E)$ is a hypergraph whose vertices are n-state “qudits” (generalizing qubits, which have $n = 2$) and whose hyperedges are k-subsets of qudits. The *local Hamiltonian problem* (LH-MIN) is to estimate the ground state energy of H to $O(\log n)$ bits of precision. LH-MIN is clearly NP-hard, but it does not seem to be NP-complete. It turns out that LH-MIN is QMA-complete, where QMA is a natural generalization of NP to the quantum setting.

8.2.1 The complexity class QMA

A *promise problem* is a partial function $f : \mathbb{B}^* \to \mathbb{B}$. Inputs not in $f^{-1}(0) \cup f^{-1}(1)$ are “undefined” and an algorithm computing f may behave arbitrarily on these inputs. QMA [68] is the following class of promise problems.

Definition 8.1 *A promise problem $f : \mathbb{B}^* \to \mathbb{B}$ is in* QMA *if there exists a BQP verifier V and a polynomial p such that:*

$$x \in f^{-1}(1) \quad \Rightarrow \quad \exists |\xi\rangle \in \mathcal{B}^{\otimes p(|x|)} : \Pr[V(x, |\xi\rangle) = 1] \geq 1 - \epsilon \tag{8.5}$$

$$x \in f^{-1}(0) \quad \Rightarrow \quad \forall |\xi\rangle \in \mathcal{B}^{\otimes p(|x|)} : \Pr[V(x, |\xi\rangle) = 1] \leq \epsilon \tag{8.6}$$

Here the expression $\Pr[V(x, |\xi\rangle) = 1]$ denotes the probability that the verifier accepts on input x with witness $|\xi\rangle$. The classes NP and MA are obtained if the (quantum witness, BQP verifier) pair is replace by (classical witness, BPP verifier) and (classical witness, P verifier), respectively. As in the classical setting, the definition remains equivalent if the *completeness* parameter $1-\epsilon$ and the *soundness* parameter ϵ are replaced by values from $[\frac{1}{2}+1/p'(|x|), 1-1/p'(|x|)]$ and $[1/p'(|x|), \frac{1}{2}-1/p'(|x|)]$, respectively, where p' is any fixed polynomial.

Clearly QMA contains both BQP and MA, and it is known that QMA $\subseteq$ PP.[1] Not much else is known about QMA. Is it weakened if we require perfect completeness? Is it weakened if we require the witness $|\xi\rangle$ to be a classical state? Is it *self-reducible* in that a witness $|\xi\rangle$ can be computed efficently for any input $f^{-1}(1)$ using queries to an oracle deciding f? For a survey of these and many other aspects of QMA, see the survey of Aharonov and Naveh [6].

8.2.2 LH-MIN is QMA-complete

A precise definition of the local Hamiltonian problem is as follows.

Definition 8.2 *Given an n-qudit k-local Hamiltonian H whose ground state energy is promised to be at most a or at least $b > a$, where $b-a$ is at least inverse-polynomial in n, the* local Hamiltonian problem (LH-MIN) *is to decide which of these two possibilities is the case.*

If $k = O(1)$, then H has at most polynomially many local terms in n. We assume that each local term is specified by polynomially many bits and has operator norm bounded polynomially.

Kitaev [68] used the Hamiltonian (7.10) to show that LH-MIN is QMA-complete.

Theorem 8.3 *The n-qubit 5-local Hamiltonian problem is QMA-complete.*

Proof: Theorem 2.3 implies that the problem is in QMA. It remains to show that it is hard for QMA.

[1] A problem is in PP if there is an NP verifier that decides it correctly a majority of the time, but not a clear majority as with BPP. PP contains NP and has power similar to that of #P.

Let $V(x, |\xi\rangle) = U_L \cdots U_1|x, \xi\rangle$ be the verifier circuit. Let H' be Kitaev's Hamiltonian (7.10), modified by adding a second input register for the witness $|\xi\rangle$ (but not penalizing any state $|\xi\rangle$ by H_{input}) and by adding to H' an *output* term

$$H_{output} := |0\rangle\langle 0|_1 \otimes |1^L\rangle\langle 1^L| \tag{8.7}$$

to penalize the rejection of input $|x\rangle$ with witness $|\xi\rangle$.

Completeness is relatively easy to show. If input x with witness $|\xi\rangle$ is accepted by the circuit with probability $1-\epsilon$, then the history state $|\eta\rangle = \frac{1}{\sqrt{L+1}} \sum_{l=0}^{L} U_l \cdots U_1|x, \xi\rangle$ lies in the nullspace of every term but H_{output} and it follows that $\langle\eta|H|\eta\rangle \leq \epsilon$.

Soundness is more complicated. We will invoke Lemma 7.3 with $H_1 = H_{input} + H_{output}$ and $H_2 = H_{prop}$. The ground state energy lower bound will carry over to the entirety of H' (including H_{clock}) because the subspace spanned by all legal clock states is invariant under H' and the eigenvalues in the complement of this subspace are at least one due to penalty terms.

Since H_1 is a sum of commuting projectors, the spectral gap above its ground space is at least one. Moreover, we computed the spectral gap of H_2 above zero earlier: it is $\Theta(1/L^2)$. So it remains only to show that the angle between the two ground spaces is large. It is a technical exercise (see [68] for proof) that this angle has $\sin^2\theta \geq \frac{1-\sqrt{\epsilon}}{L+1}$. ■

This result was improved to 3-local qubit interactions by Kempe and Regev [63], to 2-local qubit interactions by Kempe et al. [62], to 2-local qubit interactions on the two-dimensional planar lattice by Oliveira and Terhal [83], and to 2-local 12-state qudit interactions on the one-dimensional line by Aharonov et al. [5]. At a glance, the final result is somewhat surprising since the corresponding maximum constraint satisfaction (MAX-CSP) problem is solvable in polynomial time.

8.3 Some observations and open problems

Many fundamental questions about the local Hamiltonian problem – and its connection to complexity classes and to adiabatic computing – remain unresolved. Here we consider a selection of them.

A BQP-complete restriction of LH-MIN. In practice, a good estimate for the ground state energy of a complicated Hamiltonian H' can sometimes be obtained by identifying a low-energy state $|\phi'\rangle$ of a nearby Hamiltonian H' and refining it to a low-energy state $|\phi\rangle$ of H using perturbation theory. In the computational setting, adiabatic algorithms have this same flavor. So does the remark following Theorem 2.3: If we know (i.e., can efficiently compute) a state $|\xi\rangle$ having large overlap with a low-energy state – i.e., one such that $|\langle\psi|\xi\rangle|^2$ is at least inverse-polynomial for some state $|\psi\rangle$ of energy at most a – then we can estimate the ground state energy efficiently. It turns out that this promise problem is complete for BQP:

Theorem 8.4 *With the additional promise that there there exists an efficiently computable state whose overlap with a low-energy state is large, LH-MIN is BQP-complete.*

Proof: The hardness proof follows by modifying Kitaev's Hamiltonian (7.10) to a Hamiltonian H' which includes the additional output term H_{output} (8.7). If x is accepted by the circuit with probability $1 - \epsilon$, then the history state $|\eta\rangle$ (7.14) is a good approximation to the ground state of H'; in particular, its energy is at most ϵ. If not, then a lower bound can be shown on the ground state energy of H' by the same argument as in Theorem 8.3. ■

By trivial modification of Lemma 7.2, the ground state of H' is *unique* and the spectral gap of H' is at least inverse-polynomial in the circuit length L. Since the circuit history state $|\eta\rangle$ is a good approximation to the ground state, we can recover it efficiently by applying the algorithm in Theorem 2.3 to H' using the trivially preparable input state $|x\rangle \otimes |0^L\rangle$ as the witness.

Hamiltonians with large spectral gaps. What is the complexity of computing the ground state – or at least solving LH-MIN – for a Hamiltonian whose spectral gap is "large"?[2] There is a well-studied analogue of this question in the classical setting: How hard is it to find the assignment satisfying a SAT formula if it is promised to be unique?[3]

[2] Note that if we can compute witnesses (low-energy states) for an LH-MIN instance promised to have large spectral gap, then we can compute its ground state.

[3] The spectral gap here is the number of clauses violated by the best non-satisfying assignment.

The Valiant-Vazirani [107] *isolation* technique randomly reduces any satisfiable SAT formula to one that is uniquely satisfiable (with high probability) in polynomial time. Thus, "unique" SAT is not solvable in polynomial time unless NP problems have efficient randomized algorithms. Since similar statements can be proven for "unique" k-SAT, it is unlikely that LH-MIN can be solved efficiently even if the spectral gap is large. It is not hard to envision how a reduction isolating the ground state of an arbitrary instance of LH-MIN might work – using randomly-chosen energy penalty terms – but the technical hurdles seem considerable.

It is plausible nonetheless that the *one-dimensional* version of LH-MIN is efficiently solvable if the spectral gap is large. Like the other direct reductions from QMA to LH-MIN (e.g., Theorem 8.3), the one-dimensional reduction of Aharanov et al. [5] does not produce a spectral gap. Computational physicists seem to believe that the one-dimensional case is easier with a spectral gap than without[4] – although many also believed that the one-dimensional problem was easy even without a spectral gap, which turned out to be incorrect. It is tempting to try to design an adiabatic algorithm for this problem since the problem Hamiltonian H_1 is already supplied, but identifying an "easy" class of initial Hamiltonians connected to H_1 by an efficient adiabatic path seems to be a major challenge even in one dimension.

Quantum statistical mechanics. Recall from Chapter 1 that a mixed quantum state $\{p_j, |\psi_j\rangle\}$ can be represented by a density matrix. The *Gibbs density matrix*

$$\frac{e^{-\beta H}}{Z(\beta)} \tag{8.8}$$

describes the (statistical) thermal equilibrium at temperature $T = 1/\beta$ of a quantum system whose Hamiltonian is H and whose partition function is:

$$Z(\beta) = \mathrm{tr}(e^{-\beta H}) \tag{8.9}$$

What is the complexity of sampling from the state of a quantum system in thermal equilibrium whose Hamiltonian is local? It is at least QMA-hard, since the problem

[4] The reason is that a large spectral gap typically suppresses long-range correlations in the ground state.

at zero temperature reduces to computing the ground state of H. How hard is it to evaluate (or at least estimate) the partition function? It is at least as hard to evaluate as the partition function of a classical Hamiltonian, which is #P-complete. What about approximation? Table 8.1 summarizes what we know about the complexity of these and several other problems we have considered.

Problem	Classical version	Quantum version
Ground state energy, 1D	P	QMA-complete [5]
Partition function, 1D	P	???
Ground state energy, $\geq$2D	NP-complete	QMA-complete [68, 63, 62, 83]
Partition function, $\geq$2D	#P-complete	???

Table 8.1: The complexity of ground state and partition function problems.

References

[1] S. Aaronson and A. Ambainis. Quantum search of spatial regions. *Theory of Computing*, 1(4):47–79, 2005.

[2] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the Association for Computing Machinery*, 51(4):595–605, 2004.

[3] M. Agrawal, N. Kayal, and N. Saxena. PRIMES is in P. *Annals of Mathematics*, 160(2):781–793, 2004.

[4] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum walks on graphs. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 50–59, 2001.

[5] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe. The power of quantum systems on a line. To appear in *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science*, 2007. arXiv.org e-prints quant-ph/0705.4067 and quant-ph/0705.4077.

[6] D. Aharonov and T. Naveh. Quantum NP – a survey. arXiv.org e-print quant-ph/0210077, 2002.

[7] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47–82, 2007.

[8] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev. Adiabatic quantum computation is equivalent to standard quantum computation. *SIAM Journal on Computing*, 37(1):166–194, 2007.

[9] Y. Aharonov, L. Davidovich, and N. Zagury. Quantum random walks. *Physical Review A*, 48:1687, 1993.

[10] G. Alagic and A. Russell. Decoherence in quantum walks on the hypercube. *Physical Review A*, 72:062304, 2005.

[11] D. Aldous. Some inequalities for reversible Markov chains. *Journal of the London Mathematical Society, Series 2*, 25:564–576, 1982.

[12] D. Aldous and J. Fill. *Reversible Markov chains and random walks on graphs.* Book in preparation. http://www.stat.berkeley.edu/ aldous/RWG/book.html.

[13] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovász, and C.W. Rackoff. Random walks, universal travelling sequences, and the complexity of maze problems. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science*, pages 218–223, 1979.

[14] A. Ambainis. Quantum walks and their algorithmic applications. *International Journal of Quantum Information*, 1:507–518, 2003.

[15] A. Ambainis. Quantum search algorithms. *ACM SIGACT News*, 35:22–35, 2004.

[16] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

[17] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous. One-dimensional quantum walks. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 37–49, 2001.

[18] A. Ambainis, H. Buhrman, P. Høyer, M. Karpinski, and P. Kurur. Quantum matrix verification. Unpublished manuscript, 2002.

[19] 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. To appear in *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science*, 2007. arXiv.org e-prints quant-ph/0703015 and quant-ph/0704.3628.

[20] A. Ambainis, J. Kempe, and A. Rivosh. Coins make quantum walks faster. In *Proceedings of the 16th ACM/SIAM Symposium on Discrete Algorithms*, pages 1099–1108, 2005.

[21] A. Ambainis and O. Regev. An elementary proof of the quantum adiabatic theorem. arXiv.org e-print quant-ph/0411152, 2004.

[22] S. Arora and B. Barak. *Computational complexity: a modern approach*. Book in preparation. http://www.cs.princeton.edu/theory/complexity/.

[23] B. Aspvall, M. Plass, and R. Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[24] C. Bennett. Logical reversibility of computation. *IBM Journal of Research and Development*, 17:525–532, 1973.

[25] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

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

[27] D. Berry, G. Ahokas, R. Cleve, and B. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications on Mathematical Physics*, 270:359, 2007.

[28] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics Series*. AMS, 2002.

[29] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, and R. de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005.

[30] H. Buhrman and R. Špalek. Quantum verification of matrix products. In *Proceedings of the 17th ACM/SIAM Symposium on Discrete Algorithms*, pages 880–889, 2006.

[31] F. Chen, L. Lovász, and I. Pak. Lifting Markov chains to speed up mixing. In *Proceedings of the 31st ACM Symposium on Theory of Computing*, pages 275–281, 1999.

[32] A. Childs. *Quantum information processing in continuous time.* PhD thesis, Massachusetts Institute of Technology, 2004.

[33] A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. Spielman. Exponential algorithmic speedup by quantum walk. In *Proceedings of the 35th ACM Symposium on Theory of Computing*, pages 59–68, 2003.

[34] A. Childs and J. Eisenberg. Quantum algorithms for subset finding. *Quantum Information and Computation*, 5:593, 2005.

[35] A. Childs, E. Farhi, and S. Gutmann. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, 1:35, 2002.

[36] A. Childs and J. Goldstone. Spatial search and the Dirac equation. *Physical Review A*, 70:042312, 2004.

[37] A. Childs and J. Goldstone. Spatial search by quantum walk. *Physical Review A*, 70:022314, 2004.

[38] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London, Series A*, 454:339–354, 1998.

[39] J. Conway and A. Jones. Trigonometric diophantine equations (On vanishing sums of roots of unity). *Acta Arithmetica*, 30(3):229–240, 1976.

[40] P. Diaconis and D. Strook. Geometric bounds for eigenvalues of Markov chains. *Annals of Applied Probability*, 1:36–61, 1991.

[41] M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *Journal of the Association for Computing Machinery*, 38:1–17, 1991.

[42] E. Farhi, J. Goldstone, and S. Gutmann. Quantum adiabatic evolution algorithms versus simulated annealing. arXiv.org e-print quant-ph/0201031, 2002.

[43] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. arXiv.org e-print quant-ph/0702144, 2007.

[44] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. arXiv.org e-print quant-ph/0001106, 2000.

[45] E. Farhi and S. Gutmann. Analog analogue of a digital quantum computation. *Physical Review A*, 57:2403, 1998.

[46] E. Farhi and S. Gutmann. Quantum computation and decision trees. *Physical Review A*, 58:915, 1998.

[47] L. Fedichkin, D. Solenov, and C. Tamon. Mixing and decoherence in continuous-time quantum walks on cycles. *Quantum Information and Computation*, 6:263–276, 2006.

[48] R.P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.

[49] R.P. Feynman. Quantum mechanical computers. *Optics News*, 11:11–20, 1985.

[50] R. Freivalds. Fast probabilistic algorithms. In *Proceedings of the 8th Symposium on Mathematical Foundations of Computer Science*, pages 57–69, 1979.

[51] H. Gerhardt and J. Watrous. Continuous-time quantum walks on the symmetric group. In *Proceedings of the 7th International Workshop on Randomization and Computation*, pages 290–301, 2003.

[52] L.K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 212–219, 1996.

[53] S. Hoory and A. Wigderson. Universal traversal sequences for expander graphs. *Information Processing Letters*, 46(2):67–69, 1993.

[54] P. Høyer, M. Mosca, and R. de Wolf. Quantum search on bounded-error inputs. In *Proceedings of the 30th International Colloquium on Automata, Languages, and Programming*, pages 291–299, 2003.

[55] N. Inui, Y. Konishi, and N. Konno. Localization of two-dimensional quantum walks. *Physical Review A*, 69:052323, 2004.

[56] M. Jerrum. *Counting, sampling, and integrating: algorithms and complexity.* Lectures in Mathematics, ETH Zurich. Birkhauser, 2003.

[57] M. Jerrum and A. Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18:1149–1178, 1989.

[58] M. Jerrum and A. Sinclair. Polynomial-time approximation algorithms for the Ising model. *SIAM Journal on Computing*, 22:1087–1116, 1993.

[59] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the Association for Computing Machinery*, 51:671–697, 2004.

[60] J. Kempe. Discrete quantum walks hit exponentially faster. In *Proceedings of the 7th International Workshop on Randomization and Computation*, pages 354–369, 2003.

[61] J. Kempe. Quantum random walks – an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.

[62] J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

[63] J. Kempe and O. Regev. 3-local Hamiltonian is QMA-complete. arXiv.org e-print quant-ph/0302079, 2003.

[64] V. Kendon. Decoherence in quantum walks – a review. arXiv.org e-print quant-ph/0606016, 2006.

[65] V. Kendon and B. Tregenna. Decoherence can be useful in quantum walks. *Physical Review A*, 67:042315, 2003.

[66] S. Kilpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[67] A. Kitaev. Quantum measurements and the Abelian stabilizer problem. ECCC technical report 96-003 and arXiv.org e-print quant-ph/9511026, 1995.

[68] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and quantum computation*, volume 47 of *Graduate Studies in Mathematics*. AMS, 2002.

[69] D. Knuth. Combinatorial matrices. In *Selected Papers on Discrete Mathematics*, volume 106 of *CSLI Lecture Notes*. Stanford University, 2003.

[70] S. Laplante, T. Lee, and M. Szegedy. The quantum adversary method and classical formula size lower bounds. *Computational Complexity*, 15:163–196, 2006.

[71] L. Lovász. Random walks on graphs: a survey. In *Combinatorics: Paul Erdos is Eighty*, volume 2 of *Bolyai Society Mathematical Studies*. Janos Bolyai Mathematical Society, 1993.

[72] T. Mackay, S. Bartlett, L. Stephenson, and B. Sanders. Quantum walks in higher dimensions. *Journal of Physics A*, 35:2745, 2002.

[73] F. Magniez and A. Nayak. Quantum complexity of testing group commutativity. *Algorithmica*, 48(3):221–232, 2007.

[74] F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, pages 575–584, 2007.

[75] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007.

[76] D. Meyer. From quantum cellular automata to quantum lattice gases. *Journal of Statistical Physics*, 85:551–574, 1996.

[77] D. Meyer. On the absence of homogeneous scalar unitary cellular automata. *Physics Letters A*, 223(5):337–340, 1996.

[78] G.L. Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300–317, 1976.

[79] C. Moore and A. Russell. Quantum walks on the hypercube. In *Proceedings of the 6th International Workshop on Randomization and Computation*, pages 164–178, 2002.

[80] R. Motwani and P. Raghavan. *Randomized algorithms.* Cambridge University Press, 1995.

[81] A. Nayak and A. Vishwanath. Quantum walk on the line. DIMACS technical report 2000-43 and arXiv.org e-print quant-ph/0010117, 2000.

[82] M. Nielsen and I. Chuang. *Quantum computation and quantum information.* Cambridge University Press, 2000.

[83] R. Oliveira and B. Terhal. The complexity of quantum spin systems on a two-dimensional square lattice. arXiv.org e-print quant-ph/0504050, 2005.

[84] C.H. Papadimitriou. On selecting a satisfying truth assignment. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pages 163–169, 1991.

[85] C.H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

[86] B. Reichardt. The quantum adiabatic optimization algorithm and local minima. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, pages 502–510, 2004.

[87] O. Reingold. Undirected st-connectivity in log-space. In *Proceedings of the 37th ACM Symposium on Theory of Computing*, pages 376–385, 2005.

[88] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2001.

[89] P.C. Richter. Quantum algorithms for triangle finding. To appear in *Encyclopedia of Algorithms*, Springer-Verlag, 2007.

[90] P.C. Richter. Quantum speedup of classical mixing processes. To appear in *Physical Review A*, 2007.

[91] P.C. Richter. Almost uniform sampling via quantum walks. *New Journal of Physics*, 9(72), 2007.

[92] P.C. Richter and M. Szegedy. Quantization of Markov chains. To appear in *Encyclopedia of Algorithms*, Springer-Verlag, 2007.

[93] J. Roland and N. Cerf. Quantum search by local adiabatic evolution. *Physical Review A*, 65:042308, 2002.

[94] M. Saks and A. Wigderson. Probabilistic Boolean decision trees and the complexity of evaluating game trees. In *Proceedings of the 27th IEEE Symposium on Foundations of Computer Science*, pages 29–38, 1986.

[95] M. Santha. On the Monte Carlo decision tree complexity of read-once formulae. *Random Structures and Algorithms*, 6(1):75–87, 1995.

[96] U. Schöning. A probabilistic algorithm for k-SAT and constraint satisfaction problems. In *Proceedings of the 40th IEEE Symposium on Foundations of Computer Science*, pages 410–414, 1999.

[97] N. Shenvi, J. Kempe, and K. Whaley. A quantum random walk search algorithm. *Physical Review A*, 67:052307, 2003.

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

[99] A. Sinclair. Improved bounds for mixing rates of Markov chains and multicommodity flow. *Combinatorics, Probability, and Computing*, 1:351–370, 1992.

[100] A. Sinclair. Lecture notes for *Markov Chain Monte Carlo: Foundations and Applications*. U.C. Berkeley, Fall 2006.

[101] D. Solenov and L. Fedichkin. Continuous-time quantum walks on a cycle graph. *Physical Review A*, 73:012313, 2006.

[102] D. Solenov and L. Fedichkin. Non-unitary quantum walks on hyper-cycles. *Physical Review A*, 73:012308, 2006.

[103] D. Spielman. Lecture notes for *Spectral Graph Theory and its Applications*. Yale University, Fall 2004.

[104] M. Szegedy. On the quantum query complexity of detecting triangles in graphs. arXiv.org e-print quant-ph/0310107, 2003.

[105] M. Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004.

[106] B. Tregenna, W. Flanagan, R. Maile, and V. Kendon. Controlling discrete quantum walks: coins and initial states. *New Journal of Physics*, 5(83), 2003.

[107] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. *Theoretical Computer Science*, 47:85–93, 1986.

[108] W. van Dam, M. Mosca, and U. Vazirani. How powerful is adiabatic quantum computation? In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, pages 279–287, 2001.

[109] J. Watrous. Quantum simulations of classical random walks and undirected graph connectivity. *Journal of Computer and System Sciences*, 62(2):376–391, 2001.

[110] A. Yao. Quantum circuit complexity. In *Proceedings of the 34th IEEE Symposium on Foundations of Computer Science*, pages 352–361, 1993.

Vita

Peter Courtland Richter

Education

2004-2007 Department of Computer Science, Rutgers University–New Brunswick
M.S., October 2006.

2002-2004 Department of Computer Science, Carnegie Mellon University

1998-2002 Department of Mathematics, Princeton University
A.B. *magna cum laude*, June 2002.

Employment

2007 Shannon Laboratory, AT&T Labs Research, Florham Park, New Jersey
Summer Intern.

2004-2007 Department of Computer Science, Rutgers University–New Brunswick
Research/Teaching Assistant.

2002-2004 Department of Computer Science, Carnegie Mellon University
Graduate Assistant.

Publications

2007 Peter C. Richter and Mario Szegedy. Quantization of Markov chains. To appear in *Encyclopedia of Algorithms*, Springer-Verlag, 2007.

Peter C. Richter. Quantum algorithms for triangle finding. To appear in *Encyclopedia of Algorithms*, Springer-Verlag, 2007.

Peter C. Richter. Quantum speedup of classical mixing processes. To appear in *Physical Review A*, 2007.

Peter C. Richter. Almost uniform sampling via quantum walks. *New Journal of Physics* 9 72, 2007.

2004 Gary L. Miller and Peter C. Richter. Lower bounds for graph embeddings and combinatorial preconditioners. In *Proceedings of the 16th ACM Symposium on Parallel Algorithms and Architectures*, pages 112–119, 2004.