Continuous-time Quantum Algorithms: Searching and Adiabatic Computation

by

Lawrence Mario Ioannou

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Mathematics
in
Combinatorics and Optimization

Waterloo, Ontario, Canada, 2002

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

One of the most important quantum algorithms is Grover's search algorithm [G96]. Quantum searching can be used to speed up the search for solutions to NP-complete problems e.g. 3SAT. Even so, the best known quantum algorithms for 3SAT are considered inefficient.

Soon after Grover's discovery, Farhi and Gutmann [FG98] devised a "continuous-time analogue" of quantum searching. More recently Farhi *et. al.* [FGGS00] proposed a continuous-time 3SAT algorithm which invokes the adiabatic approximation [M76]. Their algorithm is difficult to analyze, hence we do not know whether it can solve typical 3SAT instances faster than Grover's search algorithm can.

I begin with a review of the discrete- and continuous-time models of quantum computation. I then make precise the notion of "efficient quantum algorithms", motivating sufficient conditions for discrete- and continuous-time algorithms to be considered efficient via discussion of standard techniques for discrete-time simulation of continuous-time algorithms. After reviewing three quantum search algorithms [F00, FG98, G96], I develop the adiabatic 3SAT algorithm as a natural extension of Farhi and Gutmann's search algorithm. Along the way, I present the adiabatic search algorithm [vDMV01] and remark on its discrete-time simulation. Finally I devise a generalization of the adiabatic algorithm and prove some lower bounds for various cases of this general framework.

Acknowledgements

I am grateful to my supervisor Michele Mosca for all of his help, teaching, and advice related to this work; his ideas form the basis for much of section 5.2 and chapter 6.

Many thanks to Christof Zalka, who spent time to help me learn about this topic, and who guided me to believe things that are correct. His ideas helped form parts of section 2.4 and chapter 6.

I am indebted to NSERC Canada and the University of Waterloo, both of which funded this M.Math degree.

Special thanks to my parents, brother, family, and friends for their love and support.

Very special thanks to Sarah for her love and support.

Contents

Preface

Recently, Farhi, Gutmann, Goldstone, and Sipser (FGGS) [FGGS00] proposed a continuous-time quantum algorithm that solves 3SAT, invoking the adiabatic approximation [M76]. My work began with the intention of proving that this adiabatic algorithm does not provide any significant speed-up over Grover's algorithm on any instances of 3SAT. As time went on, this proved to be a rather lofty goal. I was able, however, to make some noteworthy observations and prove some relevant theorems:

- In chapter 4, I show that it is unclear whether the $O(\sqrt{N})$-time adiabatic search algorithm [vDMV01] can be simulated by a discrete-time algorithm using $o(N)$ resources via standard discretization methods.
- In chapter 6, I prove that any continuous-time algorithm following FGGS's adiabatic paradigm must use initial and final Hamiltonians each with two large mutually-orthogonal subspaces if its running time is required to be small.
- In chapter 6, I reproduce the optimality proof of Farhi and Gutmann's search algorithm in the context of a "generalized adiabatic scheme"; because of the constraints of this scheme, our result is stronger than the original.

In addition to the above, two other features of this work are

- the discussion in chapter 2 of what it means for a quantum algorithm to be "efficient", and

- the seamless development of FGGS's adiabatic algorithm from quantum searching.

Chapter 1 gives a concise introduction to quantum mechanics and quantum computation, where familiarity with linear algebra and tensor product of vector spaces is assumed. Chapter 2 gives *sufficient* conditions for discrete-time and continuous-time quantum algorithms to be considered efficient. Chapter 3 reviews three approaches to searching for an unknown element $w \in \{0,1\}^n$ – two continuous-time algorithms and Grover's celebrated discrete-time algorithm. Chapter 4 introduces the adiabatic theorem and adiabatic approximation and gives the adiabatic version of Farhi and Gutmann's search algorithm. Chapter 5 introduces 3SAT, and motivates the proposed adiabatic 3SAT algorithm as a generalization of Farhi and Gutmann's search algorithm. In Chapter 6, we cast the adiabatic 3SAT algorithm in a more general setting and prove some lower bounds.

Chapter 1

Introduction

The theory of quantum computation derives from the basic theory of quantum mechanics. Properties of quantum mechanical systems are usually described as varying continuously with the parameter t which corresponds to physical time. If we only care about the properties at specific points in time, say $t = t_0$ and $t = t_1$, then it is not beneficial to bother with the details of the properties at times *between* t_0 and t_1. This is evidenced in classical computation theory. A NOT-gate flips a bit-value stored by some physical medium. Presumably it takes time to flip the bit-value, but we do not care about what the medium is doing while we are waiting for it to change. Traditionally such matters are only important to engineers. This viewpoint has led to the conventional paradigm of quantum computation, where quantum algorithms are expressed as a sequence of discrete operations and physical time is ignored.

Recently, though, embracing the continuous-time nature of quantum mechanics has borne fruit. The proposed adiabatic 3SAT algorithm has provided a new paradigm for designing quantum algorithms. What other interesting continuous-time algorithms might be waiting to be discovered? This chapter contains a concise overview of quantum mechanics followed by a development of both frameworks.

1.1 Quantum Mechanics in a Nutshell

The following is based on the development of quantum mechanics given in Sakurai's book [S85]. See Dirac's book [D58] for a more rigorous development.

Consider a quantum physical system frozen in time. One axiom of quantum mechanics is that the (pure) states of a quantum physical system are in one-to-one correspondence with the unit vectors $|\psi\rangle$ of a *Hilbert space*. Since we will deal only with finite-dimensional Hilbert spaces throughout this work, we can take "Hilbert space" to mean just a finite-dimensional, complex vector space endowed with an inner product. We will use only the usual inner product (i.e. "Euclidean" or "dot-product"). It is conventional to call $|\psi\rangle$ a *ket*. The vector that is dual to $|\psi\rangle$ is called a *bra* and is denoted $\langle\psi|$. Thus the inner product of two kets, $|a\rangle$ and $|b\rangle$, is expressed in this notation as $\langle a|b\rangle$, which is $\langle a| \cdot |b\rangle$ with the "$|\cdot|$" reduced to a single vertical line; the "$\cdot$" can be thought of as regular matrix multiplication once a basis for the Hilbert space is fixed and $\langle a|$ and $|b\rangle$ are represented by a row-matrix and column-matrix respectively. For example, note that $\langle\psi|\psi\rangle = 1$ since $|\psi\rangle$ is a unit vector.

Suppose that a physical property A of $|\psi\rangle$ may be measured or *observed*. The result of such a measurement is a real number (in practice having finite representation dictated by the precision of the measurement apparatus). Another axiom of quantum mechanics is that all possible real outcomes of measuring property A form the spectrum of a Hermitian operator (which we also denote by "A"). For our purposes it is sufficient to assume that all such physical properties A are in one-to-one correspondence with the Hermitian operators acting on the Hilbert space, so that any Hermitian operator defines a physical property that can be measured. Suppose the Hilbert space has dimension N. Since the set of eigenkets (eigenvectors) of a Hermitian operator may be made into an orthonormal basis $\{|a_i\rangle : i = 1, 2, \ldots, N\}$ for the Hilbert space, $|\psi\rangle$ may be expressed in this orthonormal

eigenbasis as

$$|\psi\rangle = \sum_{i=1}^{N} c_{a_i} |a_i\rangle , \qquad c_{a_i} \in \mathbb{C}, \qquad \sum_{i=1}^{N} |c_{a_i}|^2 = 1,$$

where $|a_i\rangle$ is an eigenket of A with eigenvalue a_i, that is,

$$A |a_i\rangle = a_i |a_i\rangle ,$$

and the orthonormality condition holds:

$$\langle a_j | a_k \rangle = \delta_{jk} \equiv \begin{cases} 0 & \text{if } j \neq k \ , \\ 1 & \text{if } j = k \ . \end{cases}$$

When property A of $|\psi\rangle$ is measured in the laboratory, the *measurement axiom* dictates that with probability $|c_{a_i}|^2$ (1) the result of the measurement is a_i and (2) the state immediately after the measurement is $|a_i\rangle$. Such physical properties or Hermitian operators, A, are also called *observables*. The numbers c_{a_i} are called *probability amplitudes*.

Now allow the physical time parameter t to enter the picture. The quantum physical system is time dependent and its state is now more properly denoted $|\psi(t)\rangle$. Suppose the unit vector $|\psi(t_0)\rangle$ is the state at some initial time t_0. Another axiom of quantum mechanics is that, at a later time t, the state has evolved via some linear operation $U(t, t_0)$ from $|\psi(t_0)\rangle$ to $|\psi(t)\rangle$:

$$|\psi(t)\rangle = U(t, t_0) |\psi(t_0)\rangle , \qquad t > t_0. \tag{1.1}$$

The operator $U(t, t_0)$ is called the *time-evolution operator*.

Since $|\psi(t)\rangle$ is to remain a unit vector throughout the evolution, $U(t, t_0)$ is required to be unitary i.e. $U(t, t_0)^{-1} = U(t, t_0)^\dagger$, so that

$$\begin{aligned} \langle \psi(t) | \psi(t) \rangle &= (U(t, t_0) |\psi(t_0)\rangle)^\dagger U(t, t_0) |\psi(t_0)\rangle \\ &= \langle \psi(t_0)| U(t, t_0)^\dagger U(t, t_0) |\psi(t_0)\rangle \\ &= \langle \psi(t_0)| U(t, t_0)^{-1} U(t, t_0) |\psi(t_0)\rangle \\ &= \langle \psi(t_0) | \psi(t_0) \rangle \\ &= 1, \end{aligned}$$

where "$O^\dagger$" denotes the object whose matrix representation is the complex conjugate transpose of the matrix representing the object O. We have used the identities $|a\rangle^\dagger = \langle a|$ and $(XY)^\dagger = Y^\dagger X^\dagger$.

Since 1.1 holds for any initial time t_0, we have

$$U(t, t_0) = U(t, t')U(t', t_0), \quad \text{for all } t' \text{ such that } t_0 < t' < t. \tag{1.2}$$

Another reasonable requirement of the time-evolution operator is that for all $t \geq t_0$

$$\lim_{t' \to t} U(t', t) = \mathbb{I}, \quad t' \geq t,$$

where $\mathbb{I}$ denotes the identity transformation. Let $\epsilon > 0$ be considered in the limit as ϵ approaches 0. It turns out [S85] that all of the above requirements are satisfied if the time-evolution operator satisfies, for all $t \geq t_0$,

$$U(t + \epsilon, t) = \mathbb{I} - \frac{\mathrm{i}}{\hbar} H(t)\epsilon + \Xi(\epsilon), \quad ||\Xi(\epsilon)|| \in o(\epsilon), \tag{1.3}$$

for a Hermitian operator $H(t)$ called a *Hamiltonian*, where for a linear operator A

$$||A|| \equiv \max_{\langle\psi|\psi\rangle=1} ||A\,|\psi\rangle\,|| \tag{1.4}$$

(see section 8.1 of the Appendix for definitions and conventional usage of asymptotic notation). The constant $\hbar$ is *Planck's constant* divided by 2π. Since

$$U(t + \epsilon, t_0) = U(t + \epsilon, t)U(t, t_0)$$

by property 1.2, we get

$$\frac{U(t + \epsilon, t_0) - U(t, t_0)}{\epsilon} = -\frac{\mathrm{i}}{\hbar} H(t)U(t, t_0) + \frac{\Xi(\epsilon)}{\epsilon} U(t, t_0).$$

Taking the limit as ϵ approaches 0 of both sides of this equation gives the *Schrödinger equation* for the time-evolution operator,

$$\mathrm{i}\hbar \frac{\partial}{\partial t} U(t, t_0) = H(t)U(t, t_0).$$

Applying each side of the above equation to $|\psi(t_0)\rangle$ and noting that

$$\left(\frac{\partial}{\partial t}U(t,t_0)\right)|\psi(t_0)\rangle = \frac{\partial}{\partial t}\left(U(t,t_0)\,|\psi(t_0)\rangle\right)$$

gives the more common version of the Schrödinger equation,

$$\mathrm{i}\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = H(t)\,|\psi(t)\rangle\,.$$

The Hamiltonian $H(t)$ is an observable called the *total energy operator*: the result of a measurement at time t of the energy of a quantum system is one of the eigenvalues of $H(t)$. Thus the eigenvalues of $H(t)$ are often called *energy eigenvalues*. It is convenient to regard the Hamiltonian as being responsible for how the quantum physical system changes in time, given the initial condition of the system $|\psi(t_0)\rangle$. Thus we say that "$H(t)$ *generates* (or *induces*, or *effects*) the unitary evolution $U(T,0)$" or "$|\psi(t)\rangle$ *evolves under* $H(t)$."

1.2 Quantum Computation in a Nutshell

In quantum computation we usually use a Hilbert space of dimension $N = 2^n$, where n is the number of two-dimensional quantum systems that make up the entire N-dimensional system. Each two-dimensional quantum system is called a *quantum bit*, or *qubit*.

The standard orthonormal basis, or *computational basis*, for this space is $\{|z\rangle : z \in \{0,1\}^n\}$, i.e. the set of N kets labelled by the N classical bit-configurations of an n-bit classical computer register. In practice, the computational basis is chosen to be the set of eigenkets of the observable whose measurement is most convenient to carry out in the given physical scheme used to implement the qubits of the quantum register. Thus the state of an n-qubit quantum register may be expressed as a unit vector in the computational basis as

$$|\psi\rangle = \sum_{z\in\{0,1\}^n} c_z\,|z\rangle\,, \tag{1.5}$$

where

$$\sum_{z\in\{0,1\}^n} |c_z|^2 = \langle\psi|\psi\rangle = 1.$$

We write "$\sum_z$" for "$\sum_{z\in\{0,1\}^n}$". We use the word "state" in place of "ket".

Our N-dimensional Hilbert space is the n-fold tensor product of the n 2-dimensional spaces corresponding to the n qubits. For example, consider two qubits, a and b. If the state space for qubit a is spanned by $\{|0_a\rangle, |1_a\rangle\}$, and the state space for qubit b is spanned by $\{|0_b\rangle, |1_b\rangle\}$, then the state space for the two qubits is spanned by

$$\{|0_a\rangle \otimes |0_b\rangle, |0_a\rangle \otimes |1_b\rangle, |1_a\rangle \otimes |0_b\rangle, |1_a\rangle \otimes |1_b\rangle\}.$$

Writing $|01\rangle$ instead of $|0_a\rangle \otimes |1_b\rangle$ is the commonly used shorthand. Similarly, each binary digit of the label $z \in \{0,1\}^n$ in equation 1.5 corresponds to a particular qubit.

The measurement of an observable with N different eigenvalues and set of normalized eigenstates equal to the computational basis is called a *measurement in the computational basis.*

Recall that the quantum mechanical time-evolution operator is unitary. Without loss of generality, a conventional quantum algorithm involves three elements:

- an initial state $|\psi_0\rangle$;
- a finite sequence of non-trivial unitary operators, or *quantum gates*, $(U_j)_{j=0,1,\ldots,M-1}$, applied to $|\psi_0\rangle$; and
- a measurement in the computational basis on the final state $|\psi_M\rangle = U_{M-1}\cdots U_1 U_0 |\psi_0\rangle$.

The sequence of quantum gates is often called an *acyclic quantum gate array* or a *quantum network.* Because it resembles the Boolean circuit model of classical computation, this model is sometimes referred to as the *circuit model* of quantum computation.

Once a basis for the Hilbert space is fixed, circuit-model quantum algorithms are easily expressed in the language of matrices, where $N \times 1$ column matrices represent the state of the n-qubit quantum computer, and unitary matrices represent the quantum gates U_j. A $2^k \times 2^k$ unitary matrix is considered a *k-qubit gate*, as it acts non-trivially on a subspace of dimension at most the dimension of the state space of k qubits.

Though the circuit model of quantum computation effectively "abstracts away" Hamiltonians and the Schrödinger equation, we certainly should not forget them altogether! After all, each gate U_j will ultimately be implemented as the evolution induced by some Hamiltonian $H_j(t)$ for some amount of time Δt_j since physical systems do not actually evolve in discrete stages (at least in our model of quantum mechanics). Thus we have an equivalent definition of a quantum algorithm:

- an initial state $|\psi_0\rangle$;
- a Hamiltonian $H(t)$, $0 \leq t \leq T$; and
- a measurement in the computational basis on the final state $|\psi(T)\rangle$, where $|\psi(t)\rangle$ satisfies

$$\begin{cases} \frac{\partial}{\partial t}|\psi(t)\rangle &= -\frac{\mathrm{i}}{\hbar}H(t)|\psi(t)\rangle\,, \quad 0 \leq t \leq T \\ |\psi(0)\rangle &= |\psi_0\rangle\,. \end{cases}$$

We thus distinguish between conventional *discrete-time* quantum algorithms specified via quantum gates and *continuous-time* quantum algorithms specified via Hamiltonians. Since final measurements are restricted to be in the computational basis, a quantum algorithm may be denoted $(|\psi_0\rangle\,, (U_j)_{j=0,1,\ldots,M-1})$ or $(|\psi_0\rangle\,, H(t)_{0\leq t\leq T}, T)$ for a discrete-time or continuous-time algorithm respectively.

In this chapter we have reviewed the basic elements of quantum mechanics and intro-

duced the discrete-time and continuous-time models of quantum computation.

Chapter 2

Efficient Quantum Algorithms

One of the most difficult challenges in computer science is to find an efficient algorithm for NP-complete problems (see [GJ79]), or to prove that such an algorithm does not exist. The 3SAT problem, which we define in section 5.1, is an example of an NP-complete problem. The intended goal of this work was to analyze the proposed adiabatic 3SAT algorithm to determine whether it is indeed efficient. In this chapter we define what it means for an algorithm to be efficient. We motivate sufficient conditions for efficiency of quantum algorithms in both the discrete-time and continuous-time frameworks.

2.1 Definition of "Efficient Algorithm"

In evaluating the efficiency of an algorithm for a certain problem, we estimate the total number of physical resources required to implement the algorithm as a function of the size of an instance of the problem. In this work we shall define "problem" as in [GJ79]: a *problem* is a general question to be answered, usually possessing several parameters, or *free variables*, whose values are left unspecified; a problem is described by giving (1) a general description of all its free variables and (2) a statement of what properties the answer, or *solution*, is required to satisfy. An *instance* of a problem is obtained by

specifying particular values for all the free variables of the problem. Problem instances are generally considered to be given as binary encodings. The *size* of a given instance is defined as the length of the binary encoding that has the *minimum length necessary* to specify the problem instance. Let Π denote a particular problem. We assume that there is exactly one free variable v_Π of Π (with one degree of freedom) that determines the size of an instance of Π. Let S_Π denote a suitable (see paragraph with line 2.1 below) "size function"

$$S_\Pi : \{\text{all instances of } \Pi\} \longrightarrow \mathbb{Z}^+,$$

where $\mathbb{Z}^+ \equiv \{1, 2, \ldots\}$. Thus, for all instances I and J of Π,

$$S_\Pi(I) = S_\Pi(J) \quad \Leftrightarrow \quad v_\Pi(I) = v_\Pi(J),$$

where $v_\Pi(I)$ is the specified value of the free variable v_Π in instance I. Let $\sigma_\Pi : \mathbb{Z}^+ \to \mathbb{Z}^+$ be a *strictly increasing* function such that

$$\{S_\Pi(I) : \ I \text{ is an instance of } \Pi\} = \{\sigma_\Pi(n) : n \in \mathbb{Z}^+\}$$

(where we assume all instances have nonzero size). Thus, we enumerate all of the instances of a problem Π by the index n in increasing order of size such that

$$\Pi_n \equiv \{\text{all instances of } \Pi \text{ having size } \sigma_\Pi(n)\}$$

(so Π_1 denotes the smallest instances of Π, Π_2 the next largest instances of Π, etc.). For many problems Π, and all problems that we consider in this course, v_Π is identified with the index variable n.

Suppose that an algorithm for solving instances of a certain problem requires an amount of total resources (e.g. time, space, energy, experimental precision, amount of apparatus) equal to $r(\sigma)$ on instances of size σ. Then, the algorithm is considered *efficient* if and only if $r(\sigma) \in O(\text{poly}(\sigma))$. When the algorithm is efficient, we also say that it requires a *small* amount of physical resources to implement. More generally, the word

"small" is used to describe any quantity that is in $O(\text{poly}(\sigma))$, where σ is the size of the problem instance.

Evidently, there are many suitable "size functions" S_Π. In order to respect the definition of "efficient", we only require that any two "size functions" be *polynomially related*. Technically, this can be defined as follows. Let S_Π and S'_Π be any two size functions that, as detailed above, give rise to $\sigma_\Pi(n)$ and $\sigma'_\Pi(n)$ respectively. We say that $\sigma_\Pi(n)$ and $\sigma'_\Pi(n)$ (and S_Π and S'_Π) are *polynomially related* if and only if there exist $k \geq 0$ and $k' \geq 0$ such that

$$\sigma'_\Pi(n) \in O((\sigma_\Pi(n))^k) \quad \text{and} \quad \sigma_\Pi(n) \in O((\sigma'_\Pi(n))^{k'}). \tag{2.1}$$

If a quantum algorithm requires a Hilbert space of size $N = 2^n$ on a problem instance of size $\Omega(\text{poly}(n))$, then it uses a small amount of space. When, precisely, is a quantum algorithm considered efficient with respect to *total* resources?

To answer this question properly, we need to make more precise our definition of "algorithm". We will consider a (quantum) *algorithm* $\mathcal{A}_\Pi$ for a problem Π to be a uniform family of "instances of $\mathcal{A}_\Pi$"

$$\mathcal{A}_\Pi \equiv \{\mathcal{A}_{\Pi,n} : n = 1, 2, \ldots\}$$

where $\mathcal{A}_{\Pi,n}$ is the "instance of the algorithm $\mathcal{A}_\Pi$" that solves instances of Π of size $\sigma_\Pi(n)$. By "uniform" we mean that there exists a classical Turing Machine (see chapter 3 in [NC00]) that, given n, outputs a complete description of $\mathcal{A}_{\Pi,n}$ using $O(\text{poly}(\sigma_\Pi(n)))$ total resources. It will be clear from the presentation that all algorithms we consider satisfy this uniformity condition. In the case of discrete-time algorithms we have

$$\mathcal{A}_{\Pi,n} = \left(|\psi_0(n)\rangle\,,\ (U_j(n))_{j=0,1,\ldots,M(n)-1}\right), \qquad n = 1, 2, \ldots,$$

but for convenience we will often hide the dependence on n and just write

$$\mathcal{A}_\Pi = \{(|\psi_0\rangle\,,\ (U_j)_{j=0,1,\ldots,M-1})\}$$

with the understanding that we have a family of "algorithm instances" indexed by n. When referring to an element of an algorithm, for example U_j above, we are always implicitly referring to the family $\{U_j(n)\}$. We use the analogous convention in the case of continuous-time algorithms. For example, suppose we have a continuous-time algorithm $\mathcal{C}_\Pi = \{(|\psi_0\rangle, H(t), T)\}$; it is understood that the pure running time T is a function of n and, for the purpose of analyzing efficiency, a function of $\sigma_\Pi(n)$:

$$T = T(\sigma_\Pi(n)), \qquad n = 1, 2, \ldots .$$

We also define the function $\mathrm{SPACE}_{\mathcal{A}_\Pi}(n) : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that

$$\mathrm{SPACE}_{\mathcal{A}_\Pi}(n) \equiv \{\text{the total number of distinct qubits that } \mathcal{A}_{\Pi,n} \text{ uses}\}.$$

For convenience, in the following discussion about the efficiency of algorithms in general, we shall drop the subscript label which denotes the problem i.e. the subscript "Π" above. That is, we shall speak of a quantum algorithm $\mathcal{A}$ assuming that it solves some problem whose instances are indexed by n.

Before we continue with our discussion of efficiency, here is some notation and conventions that we will use for the rest of this work. If D is an operator, $\mathcal{B}$ is a basis for a complex inner product space, and D acts on span$\mathcal{B}$, then let $[D]_\mathcal{B}$ be the matrix representation of D in basis $\mathcal{B}$. If no basis is specified, assume the basis is the computational basis $\{|z\rangle : z \in \{0,1\}^n\}$. If A is a matrix of size $p \times q$, then sometimes we will write $A_{p\times q}$ to emphasize the size. If $A = \left[a_{ij}\right]$ is a $p \times q$ matrix, and B is an $r \times s$ matrix, then $A \otimes B$ is a $pr \times qs$ matrix, expressed in block form as,

$$A \otimes B = \left[a_{ij}B\right]_{pr\times qs}.$$

Suppose we have an n-qubit quantum register. In labelling each computational basis state with an n-bit string $z_n z_{n-1} \cdots z_1$, we establish a natural ordering of the qubits from right to left where each qubit has an index from 1 to n. We say that k qubits are *adjacent* if and only if their indices form a subset of $\{1, 2, \ldots, n\}$ that can be written

$\{I+1, I+2, \ldots, I+k\}$ for some $I \in \{1, 2, \ldots, n\}$. When we write

$$\bigotimes_{j=1}^{J} [D_j]_{2^{k_j} \times 2^{k_j}} \tag{2.2}$$

we mean that D_j acts on k_j qubits of the quantum register; we assume that we know which k_j qubits, but for convenience we will *not* assume that these qubits are adjacent.

2.2 Efficient Discrete-Time Algorithms

Suppose a quantum algorithm $\mathcal{D} = \{(|\psi_0\rangle\,,\ (U_j)_{j=1,2,\ldots,J})\}$ is specified in the discrete-time picture. The algorithm is efficient if (but not only if)

- $\mathrm{SPACE}_{\mathcal{D}}(n) \in O(\mathrm{poly}(\sigma(n)))$; (2.3)
- the initial state $|\psi_0\rangle$ requires only $O(\mathrm{poly}(\sigma(n)))$ many total physical resources to prepare; (2.4)
- $J \in O(\mathrm{poly}(\sigma(n)))$; and (2.5)
- each gate U_j acts *locally* on only $O(\log(\sigma(n)))$ qubits, i.e. $[U_j]$ may be written as

$$[U_j] = \bigotimes_{i=1}^{I_j} [U_{j,i}]_{2^{k_{j,i}} \times 2^{k_{j,i}}}, \qquad j = 1, 2, \ldots, J,$$

  where $k_{j,i} \in O(\log(\sigma(n)))$ for all i. (2.6)

Using the fact $A \otimes B = (A \otimes \mathbb{I}) \cdot (\mathbb{I} \otimes B)$ for matrices A and B, the algorithm $\{(|\psi_0\rangle\,, (U_j)_{j=0,1,\ldots,M-1})\}$ may be rewritten

$$\{(|\psi_0\rangle\,, (U_{j,i})_{j=1,2,\ldots,J;\ i=1,2,\ldots,I_j})\}.$$

What is significant about the size of the unitary matrices? How necessary is condition 2.6?

2.3 Direct Implementation of Gates via Natural Hamiltonians

The discrete-time picture of quantum algorithms effectively "abstracts away" the underlying inherent continuous-time nature of quantum physical processes (at least, according to our model of quantum mechanics). Ultimately, any quantum algorithm would be implemented in the laboratory in the continuous-time picture, where it is convenient to consider Hamiltonians as the generators of unitary operations. In this section, we briefly consider why the above sufficient conditions 2.5 and 2.6 make sense.

Suppose U is a quantum gate from the discrete-time algorithm $\mathcal{D}$ of the previous section. If U is to be efficiently implemented, it must be the induced time-evolution of some Hamiltonian H that can be efficiently prepared in the laboratory. Let us start by looking at what type of time-independent Hamiltonians ought to be efficiently implemented in the laboratory.

Up to an initial condition, the Hamiltonian is solely responsible for how a quantum physical system changes in time. It is reasonable to assume that the amount of information contained in the Hamiltonian is proportional to the physical complexity of the system it governs. Thus, intuition might suggest the rule that the number of resources required to directly implement a Hamiltonian H is lower-bounded by the minimum number of matrix elements required to describe $[H]$. Suppose $H_{\text{Structure}}$ is time independent and acts on a Hilbert space of dimension $2^{\text{SPACE}_{\mathcal{D}}(n)}$. Suppose $[H_{\text{Structure}}]$ has the tensor-product factorization

$$\begin{aligned}[H_{\text{Structure}}] \quad &= \quad \sum_{j=1}^{J} \bigotimes_{i=1}^{m_j} [H_{j,i}]_{2^{k_{j,i}} \times 2^{k_{j,i}}}, \\ &\qquad \text{with} \quad J \in O(\text{poly}(\sigma(n))),\ \max_{j,i}\{k_{j,i}\} \in O(\log(\sigma(n))),\end{aligned} \tag{2.7}$$

where no $[H_{j,i}]$ can be factorized further (with respect to tensor product). Then, employing our intuitive rule, the number of resources required to implement $H_{\text{Structure}}$ is

in

$$\Omega\left(\sum_{j=1}^{J}\sum_{i=1}^{m_j} 2^{k_{j,i}}\right) = \Omega(\text{poly}(\sigma(n))),$$

and so is lower-bounded by a polynomial in $\sigma(n)$.

However, even if the number of matrix elements gave an *upper* bound on the number of resources needed to directly implement $H_{\text{Structure}}$, we cannot rely on being able to directly implement Hamiltonians like $H_{\text{Structure}}$ for the purposes of implementing quantum gates: Hamiltonians of this form do not occur in Nature! In fact, most naturally-occurring Hamiltonians H acting on m qubits are believed to be of the form H_{Nature} where

$$H_{\text{Nature}} = \sum_{l=1}^{L} H_l, \qquad L \in O(\text{poly}(m)), \tag{2.8}$$

where each H_l acts on a maximum of 2 qubits (which implies that $L \in O(\text{poly}(m))$. Furthermore, in a typical quantum computer, technology restricts the total Hamiltonian to act on two or three qubits at a time [SS99]. Thus to guarantee that the unitary gate U can be efficiently implemented, U should act on at most 2 qubits so that it may be implemented by a natural Hamiltonian H_{Nature} with few terms, say, $L = 3$. This seems inconsistent with condition 6 but in light of the fact that any unitary gate U acting on k qubits of an m-qubit register can be so decomposed:

$$U = V_r V_{r-1} \ldots V_1, \qquad r \in O\left(\frac{2^k(2^k-1)}{2} m^2\right),$$

where each V_i acts on a maximum of two qubits (see sections 4.5.1 and 4.5.2 in [NC00]). Moreover, the only two-qubit gate required is the *controlled-sign* gate (CSIGN),

$$[\text{CSIGN}] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Thus, returning to our generic discrete-time algorithm $\mathcal{D}$, if $\text{SPACE}_{\mathcal{D}}(n) \in O(\text{poly}(\sigma(n)))$ and each $U_{j,i}$ is restricted to acting on only $O(\log(\sigma(n)))$ qubits as in

condition 6, $U_{j,i}$ may be decomposed into only $O(\text{poly}(\sigma(n)))$ many CSIGN and one-qubit gates $U_{j,i,l}$ i.e.

$$U_{j,i} = U_{j,i,L_{j,i}} U_{j,i,(L_{j,i}-1)} \cdots U_{j,i,1}, \qquad \text{where } L_{j,i} \in O(\text{poly}(\sigma(n))).$$

If each $U_{j,i,l}$ can be implemented by a simple, naturally-occurring Hamiltonian with $O(1)$ total physical resources, then conditions 5 and 6 are sound. It remains to show that there exist physical systems that support Hamiltonians effecting arbitrary one-qubit gates and the CSIGN gate with $O(1)$ total resources. We merely say that there do exist physical systems that support the implementation of one-qubit gates and the CSIGN gate via simple one- and two-qubit (time-varying) Hamiltonians with few terms (see chapter 7 in [NC00]).

We assume throughout this work that every one-qubit quantum gate that we need can be implemented exactly. In practice, we would likely be restricted to using a finite "universal" set of gates. However, it can be shown that there exists a finite "universal" set $\mathcal{G}$ of one-qubit gates such that any one-qubit gate can be implemented to accuracy ϵ with only $O(\text{poly}(\frac{1}{\epsilon}))$ gates from $\mathcal{G}$. If Y is the ideal one-qubit gate, and $\tilde{Y}$ is the approximation to Y made from gates in $\mathcal{G}$, then "$\tilde{Y}$ approximates Y to accuracy ϵ" means $\left\| Y - \tilde{Y} \right\| \leq \epsilon$. To see that $O(\text{poly}(\frac{1}{\epsilon}))$ is sufficiently small, suppose we wish to approximate a network of M one-qubit gates to overall error δ. It suffices that each gate is approximated to accuracy $\frac{\delta}{M}$. Thus the total number of gates needed will be $M' \in O\left(M \cdot \text{poly}(\frac{M}{\delta})\right)$; so if M is small, then M' will also be small. In fact, M. Solovay (in an unpublished manuscript in 1995) and Kitaev [K97] independently showed that we only need $O(\text{poly}(\log(\frac{1}{\epsilon})))$ gates from $\mathcal{G}$ to approximate any one-qubit gate to accuracy ϵ (see chapter 4 and appendix 3 in [NC00] for more details).

We note finally that condition 2.6 is too restrictive in that if a $\text{SPACE}_{\mathcal{D}}(n)$-qubit gate U happens to be the induced time-evolution of some easy-to-implement, naturally-occurring $\text{SPACE}_{\mathcal{D}}(n)$-qubit Hamiltonian in some particular physical implementation scheme, then U can be implemented efficiently. However, since *all* reasonable imple-

mentation schemes support very simple natural Hamiltonians (H_{Nature} with $L \in O(1)$), insisting on condition 2.6 is convenient.

2.4 Efficient Continuous-time Algorithms

So far we have justified the sufficient conditions for how to recognize an efficient discrete-time algorithm by explaining that each one- and two-qubit gate is directly implemented by a naturally-occurring Hamiltonian. Thankfully, this does not mean that all continuous-time algorithms $\{(|\psi_0\rangle, H(t)_{0\leq t\leq T}, T)\}$ must be restricted to only naturally-occurring Hamiltonians $H(t)_{0\leq t\leq T}$. There is a wide class of unnatural Hamiltonians whose induced time-evolution can be "efficiently simulated" or "approximated" by sequences of very simple, natural Hamiltonians, i.e. sequences of quantum gates each of constant size. Let $U(T,0)$ be the time-evolution induced by the Hamiltonian in the continuous algorithm $\mathcal{C} = \{(|\psi_0\rangle, H(t)_{0\leq t\leq T}, T)\}$. Let $\mathcal{D}(\epsilon) = \{(|\psi_0\rangle, (U_j(\epsilon))_{j=1,2,\ldots,M(\epsilon)})\}$ be a discrete-time algorithm whose "algorithm instances" are indexed by both $\epsilon > 0$ and n. Suppose further that each $U_j(\epsilon)$ is either a one-qubit gate or CSIGN. We say that $\mathcal{D}$ *efficiently approximates* (or *simulates*) $\mathcal{C}$ if and only if for all $\epsilon > 0$

$$\left\| \left(U(T,0) - U_{M(\epsilon)}(\epsilon) \cdot U_{M(\epsilon)-1}(\epsilon) \cdot \cdots \cdot U_1(\epsilon)\right) |\psi_0\rangle \right\| \leq \epsilon$$

and

$$M(\epsilon) \in O\left(\text{poly}\left(T, \max_t \|H(t)\|, \frac{1}{\epsilon}\right)\right).$$

Suppose $\mathcal{C} = \{(|\psi_0\rangle, H(t)_{0\leq t\leq T}, T)\}$ is a continuous-time algorithm. We would like a set of sufficient conditions which, if satisfied, would imply that $\mathcal{C}$ is efficient. Since our definition of efficiency should not change just because we are looking at continuous-time algorithms, "$\mathcal{C}$ is efficient" must mean "$\mathcal{C}$ can be implemented in the laboratory with $O(\text{poly}(\sigma(n)))$ total resources." Thus, the easiest way of stating a sufficient condition for $\mathcal{C}$ to be efficient is "$\mathcal{C}$ is efficient if $\mathcal{C}$ can be efficiently approximated by an efficient

discrete-time quantum algorithm." We seek conditions, analogous to 2.3, 2.4, 2.5, and 2.6, that are more tailored to the continuous-time picture. Clearly, conditions 2.3 and 2.4 carry over into the continuous-time picture unchanged. This section investigates the analogues of 2.5 and 2.6 in certain cases relevant to quantum searching and adiabatic quantum computation. For the rest of this section assume that $\mathcal{C}$ satisfies conditions 2.3 and 2.4 (with $\mathcal{C}$ in place of $\mathcal{D}$). For convenience we will use m instead of $\mathrm{SPACE}_{\mathcal{C}}(n)$ when discussing mathematical details.

2.4.1 Time Independence I: Hamiltonians With Known Diagonalizations

For this section, we assume that $\mathcal{C} = \{(|\psi_0\rangle, H, T)\}$ where H is time independent. Let H be unitarily diagonalized by U:

$$[H] = [U^\dagger][D][U],$$

where D is diagonal with respect to the computational basis. The associated induced time-evolution operator can now be decomposed so that the time-dependence of the evolution is factored out as the induced time-evolution of the diagonal Hamiltonian D:

$$e^{-\mathrm{i}Ht} = U^\dagger e^{-\mathrm{i}Dt} U,$$

where the constant $\hbar$ is suppressed and will be for the rest of this work. Thus, in evaluating the efficiency of the algorithm $\mathcal{C}$ it suffices to check that U can be decomposed into a small number of one- and two-qubit gates and that $e^{-\mathrm{i}DT}$ can be implemented efficiently. Of course this approach assumes that, given H, the matrices $[U]$ and $[D]$ are available for all n.

Suppose that

$$D = \sum_{z \in \{0,1\}^m} \lambda(z) |z\rangle\langle z|.$$

Under what sufficient condition can $e^{-\mathrm{i}DT}$ be implemented efficiently? Such a condition may be stated in terms of the eigenvalue function $\lambda(z)$:

- there is available an efficient quantum network $C_\lambda(n)$ that computes $\lambda(z)$.

We briefly show why this condition is sufficient. The operator $e^{-\mathrm{i}DT}$ puts a phase factor of $e^{-\mathrm{i}\lambda(z)T}$ in front of each basis state $|z\rangle$. Let "$\oplus$" denote "bit-wise sum modulo 2". Suppose C_λ operates on our principal m-qubit register and places the result $\lambda(z)$ into an ancillary m'-qubit register like this

$$C_\lambda : \qquad |z\rangle\,|b\rangle \mapsto |z\rangle\,|b \oplus \lambda(z)\rangle\,,$$

where the significant digits of $\lambda(z)$ are encoded as an m'-bit binary number, and $|b\rangle$ is the initial state of the ancillary register. Thus, $e^{-\mathrm{i}DT}$ can be implemented in three stages [Z98], ultimately ignoring the ancillary register:

$$|z\rangle\left|0^{m'}\right\rangle \;\mapsto\; |z\rangle\,|\lambda(z)\rangle \tag{2.9}$$

$$|z\rangle\,|\lambda(z)\rangle \;\mapsto\; e^{-\mathrm{i}\lambda(z)T}\,|z\rangle\,|\lambda(z)\rangle \tag{2.10}$$

$$e^{-\mathrm{i}\lambda(z)T}\,|z\rangle\,|\lambda(z)\rangle \;\mapsto\; e^{-\mathrm{i}\lambda(z)T}\,|z\rangle\left|0^{m'}\right\rangle \tag{2.11}$$

(see [CEMM98] for another method of computing eigenvalues into the phase). Steps 2.9 and 2.11 can be done directly and efficiently with the network C_λ. Step 2.10 is just the transformation

$$|\lambda\rangle \mapsto e^{-\mathrm{i}\lambda T}\,|\lambda\rangle$$

in the ancillary register. If, for some integer p,

$$\lambda = \frac{1}{2^p}\sum_{i=0}^{m'-1}\lambda_i 2^i, \qquad \lambda_i \in \{0,1\},$$

then

$$\begin{aligned}
& e^{-\mathrm{i}\lambda T}\left|\lambda\right\rangle \\
= \;& e^{-\mathrm{i}\left(\frac{1}{2^p}\sum_{i=0}^{m'-1}\lambda_i 2^i\right)T}\left|\lambda_{m'-1}\right\rangle \otimes \left|\lambda_{m'-2}\right\rangle \otimes \cdots \otimes \left|\lambda_0\right\rangle \\
= \;& e^{-\mathrm{i}\left(\lambda_{m'-1}2^{m'-1-p}\right)T}\left|\lambda_{m'-1}\right\rangle \otimes e^{-\mathrm{i}\left(\lambda_{m'-2}2^{m'-2-p}\right)T}\left|\lambda_{m'-2}\right\rangle \otimes \cdots \otimes e^{-\mathrm{i}\left(\lambda_0 2^{-p}\right)T}\left|\lambda_0\right\rangle \\
= \;& \left[e^{-\mathrm{i}\left(2^{m'-1-p}\right)T}\right]^{\lambda_{m'-1}}\left|\lambda_{m'-1}\right\rangle \otimes \left[e^{-\mathrm{i}\left(2^{m'-2-p}\right)T}\right]^{\lambda_{m'-2}}\left|\lambda_{m'-2}\right\rangle \otimes \cdots \otimes \left[e^{-\mathrm{i}\left(2^{-p}\right)T}\right]^{\lambda_0}\left|\lambda_0\right\rangle \\
= \;& R_{\hat{z}}(-2^{m'-1-p}T)\left|\lambda_{m'-1}\right\rangle \otimes R_{\hat{z}}(-2^{m'-2-p}T)\left|\lambda_{m'-2}\right\rangle \otimes \cdots \otimes R_{\hat{z}}(-2^{-p}T)\left|\lambda_0\right\rangle \\
= \;& \left[R_{\hat{z}}(-2^{m'-1-p}T) \otimes R_{\hat{z}}(-2^{m'-2-p}T) \otimes \cdots \otimes R_{\hat{z}}(-2^{-p}T)\right]\left|\lambda\right\rangle,
\end{aligned}$$

where

$$R_{\hat{z}}(\theta) \equiv e^{-\mathrm{i}Z\frac{\theta}{2}}, \qquad \theta \in \mathbb{R},$$

and Z is the simple, natural, one-qubit Hamiltonian

$$[Z] = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Each gate $R_{\hat{z}}(-2^{i-p}T) = R_{\hat{z}}(-2^{i-p}T \bmod 2\pi)$ can be carried out in constant time with respect to T. Thus step 2.10 can be implemented efficiently, making the simulation of the diagonal, time-independent Hamiltonian D efficient if we have the network C_λ computing $[D]$'s diagonal entries (eigenvalues) λ_i.

In summary, we have shown that the algorithm $\mathcal{C} = \{(\left|\psi_0\right\rangle, H, T)\}$ for time-independent H is efficient if (along with conditions 2.3 and 2.4)

- $H = U^\dagger D U$ for diagonal D, where $[U]$ has the form

$$[U] = \bigotimes_{j=1}^{J} [U_j]_{2^{k_j}\times 2^{k_j}},$$

  where $k_j \in O(\log(\sigma(n)))$ for all j, and we have efficiently computable formulae for the all the elements of $[U_j]$ for all n; and (2.12)

- there is an accessible, reversible, efficient quantum network C_λ that computes $[D]$'s diagonal entries $\lambda(z)$. (2.13)

Is there some structural condition on D that will guarantee condition 2.13? Suppose D has the tensor product structure

$$[D] = [D_1] \otimes [D_2] \otimes \cdots \otimes [D_I],$$

where each D_i acts on a subset S_i of the m qubits, and

$$D_i = \sum_{z_{S_i} \in \{0,1\}^{|S_i|}} \lambda_{S_i}(z_{S_i}) |z_{S_i}\rangle\langle z_{S_i}|.$$

To evaluate $\lambda(z)$, we can exploit the structure of D like this

$$\begin{aligned} D|z\rangle &= (D_1 \otimes D_2 \otimes \cdots \otimes D_I)|z_{S_1}\rangle \otimes |z_{S_2}\rangle \otimes \cdots \otimes |z_{S_I}\rangle \\ &= D_1|z_{S_1}\rangle \otimes D_2|z_{S_2}\rangle \otimes \cdots \otimes |z_{S_I}\rangle D_I \\ &= \lambda_{S_1}(z_{S_1})|z_{S_1}\rangle \otimes \lambda_{S_2}(z_{S_2})|z_{S_2}\rangle \otimes \cdots \otimes \lambda_{S_I}(z_{S_I})|z_{S_I}\rangle \\ &= \left[\prod_{i=1}^{I} \lambda_{S_i}(z_{S_i})\right] |z\rangle, \end{aligned}$$

so that $\lambda(z)$ can be computed using networks $C_{\lambda_{S_i}}$ that compute each function $\lambda_{S_i}(z_{S_i})$. Note that this method is efficient if I is small and if each network $C_{\lambda_{S_i}}$ is efficient. Note also that if the size of the domain of each λ_{S_i} is small, then $C_{\lambda_{S_i}}$ is efficient. Thus, the algorithm $\mathcal{C} = \{(|\psi_0\rangle, H, T)\}$ is efficient if

- $[H]$ may be decomposed as

$$[H] = \bigotimes_{i=1}^{I} [H_i]_{2^{k_i} \times 2^{k_i}},$$

  where $k_i \in O(\log(\sigma(n)))$ for all i; and (2.14)

- efficiently computable formulae for all the elements of $[U_i]$ and $[D_i]$, such that $[H_i] = [U_i^\dagger][D_i][U_i]$, are available for all n. (2.15)

2.4.2 Time Independence II: Small Sums of Efficiently Simulated Hamiltonians

For this section, it is useful to think of the unitary operator $e^{-\mathrm{i}Ht}$ as a rotation. For one-qubit gates, this is easy to do, as we now explain. We can write the state of one qubit as

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{\mathrm{i}\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle, \quad 0 \leq \theta \leq \pi, \;\; 0 \leq \phi < 2\pi \tag{2.16}$$

without loss (since global phases produce no measurable effects), and then visualize $|\psi\rangle$ as a unit vector in $\mathbb{R}^3$ whose tail is at the origin. Letting $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$, $\hat{\mathbf{k}}$ be the mutually-orthogonal unit vectors in $\mathbb{R}^3$ along the standard $\hat{x}$, $\hat{y}$, and $\hat{z}$ axes of $\mathbb{R}^3$ respectively, we visualize $|\psi\rangle$ as

$$\cos(\phi)\sin(\theta)\hat{\mathbf{i}} + \sin(\phi)\sin(\theta)\hat{\mathbf{j}} + \cos(\theta)\hat{\mathbf{k}}.$$

The unit sphere, centred at the origin, is often called the *Bloch sphere* in this setting. Note that if two states $|\psi\rangle$ and $|\psi'\rangle$ are orthogonal, then they do not correspond to perpendicular vectors in the Bloch sphere; for example, $|0\rangle$ and $|1\rangle$ are visualized on the Bloch sphere as $\hat{\mathbf{k}}$ and $-\hat{\mathbf{k}}$ respectively. Define the *Pauli operators* I, X, Y, and Z, all four of which are both unitary and Hermitian:

$$[I] \equiv \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad [X] \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad [Y] \equiv \begin{bmatrix} 0 & -\mathrm{i} \\ \mathrm{i} & 0 \end{bmatrix}, \quad [Z] \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{2.17}$$

Let $\hat{\mathbf{n}} = n_x\hat{\mathbf{i}} + n_y\hat{\mathbf{j}} + n_z\hat{\mathbf{k}}$ be a real unit vector in $\mathbb{R}^3$. Define the Hamiltonian

$$H_{\hat{\mathbf{n}}} \equiv \frac{1}{2}(n_x X + n_y Y + n_z Z). \tag{2.18}$$

It can be shown (see exercise 4.6 in [NC00]) that the action of the operator

$$R_{\hat{\mathbf{n}}}(\alpha) \equiv e^{-\mathrm{i}H_{\hat{\mathbf{n}}}\alpha} = \cos\left(\frac{\alpha}{2}\right) I - \mathrm{i}\cdot\sin\left(\frac{\alpha}{2}\right)(n_x X + n_y Y + n_z Z) \tag{2.19}$$

is to rotate the Bloch-sphere representation of a qubit state by an angle α about the axis defined by $\hat{\mathbf{n}}$ (any one-qubit gate can be written in the form $R_{\hat{\mathbf{n}}}(\alpha)$ up to global phase;

see exercise 4.8 in [NC00]). Thus, $R_{\hat{\mathbf{n}}}(\alpha)$ must have two mutually-orthogonal eigenstates, whose Bloch-sphere representations are $\hat{\mathbf{n}}$ and $-\hat{\mathbf{n}}$; call these eigenstates, which are only defined up to global phase, $|+\hat{\mathbf{n}}\rangle$ and $|-\hat{\mathbf{n}}\rangle$ respectively. It is easy to verify that

$$\begin{aligned} R_{\hat{\mathbf{n}}}(\alpha)\,|\pm\hat{\mathbf{n}}\rangle &= e^{\mp \mathrm{i}\alpha/2}\,|\pm\hat{\mathbf{n}}\rangle \\ \text{and} \qquad H_{\hat{\mathbf{n}}}\,|\pm\hat{\mathbf{n}}\rangle &= \pm\frac{1}{2}\,|\pm\hat{\mathbf{n}}\rangle\,. \end{aligned}$$

Thus, $e^{-\mathrm{i}H_{\hat{\mathbf{n}}}\alpha}$ is a rotation in the Bloch sphere about an *axis defined by the eigenstates of* $H_{\hat{\mathbf{n}}}$. Writing $e^{-\mathrm{i}H_{\hat{\mathbf{n}}}\alpha}$ in its spectral decomposition

$$\begin{aligned} e^{-\mathrm{i}H_{\hat{\mathbf{n}}}\alpha} &= e^{-\mathrm{i}\alpha/2}|+\hat{\mathbf{n}}\rangle\langle+\hat{\mathbf{n}}| \quad + \quad e^{+\mathrm{i}\alpha/2}|-\hat{\mathbf{n}}\rangle\langle-\hat{\mathbf{n}}| \\ &= e^{-\mathrm{i}\alpha/2}\cdot\left[\ |+\hat{\mathbf{n}}\rangle\langle+\hat{\mathbf{n}}| \quad + \quad e^{+\mathrm{i}\alpha}|-\hat{\mathbf{n}}\rangle\langle-\hat{\mathbf{n}}|\ \right] \end{aligned}$$

we see that the angle α is actually the magnitude of a relative rephasing of the eigenstates of $e^{-\mathrm{i}H_{\hat{\mathbf{n}}}\alpha}$. In the space of one qubit, the Bloch sphere allows us to visualize this relative rephasing as a classical rotation. Consider a general Hermitian operator H of rank N expressed in its spectral decomposition:

$$H = \sum_{i=1}^{N} \lambda_i |\lambda_i\rangle\langle\lambda_i|.$$

We can easily derive that

$$\begin{aligned} e^{-\mathrm{i}Ht} &= \sum_{i=1}^{N} e^{-\mathrm{i}\lambda_i t}|\lambda_i\rangle\langle\lambda_i| \\ &= e^{-\mathrm{i}\lambda_j t}\sum_{i=1}^{N} e^{-\mathrm{i}(\lambda_i-\lambda_j)t}|\lambda_i\rangle\langle\lambda_i|, \qquad \text{for any } j \in \{1,2,\ldots,N\}. \end{aligned}$$

We see that for dimensions higher than 2, $e^{-\mathrm{i}Ht}$ effects multiple relative rephasings whose magnitudes are determined by the difference between eigenvalues of H. In this case, we cannot, in any rigorous way, visualize these rephasings as one classical rotation. However, it will become apparent that $e^{-\mathrm{i}Ht}$ superficially behaves like one rotation. We clarify the analogy as follows. Define the general rotation operator $R_A(\theta)$ to mean "a rotation by

an angle θ about an axis A". Let $\Lambda = \{|\lambda_i\rangle : i = 1, 2, \ldots, N\}$ and define the maximum energy eigenvalue difference of H to be $\|\Delta H\|$:

$$\|\Delta H\| \equiv \max_{i,j} |\lambda_i - \lambda_j|. \tag{2.20}$$

The unitary operator $e^{-\mathrm{i}Ht}$ behaves like a rotation by an "angle" $\theta \leq \|\Delta H\|\, t$ about an "axis" defined by Λ:

$$e^{-\mathrm{i}Ht} \sim R_\Lambda(\theta), \qquad \theta \leq \|\Delta H\|\, t$$

(the purist can just think in terms of rephasing eigenvectors Λ by a phase angle of magnitude less than $\|\Delta H\|\, t$).

Let $\hat{a}$ and $\hat{b}$ be axes through the origin in Euclidean 3-space. We know from common experience, perhaps from moving furniture about a house, that $R_{\hat{a}}(\theta)$ and $R_{\hat{b}}(\theta)$ do not commute for every θ unless $\hat{a} = \hat{b}$. Thus, for two Hamiltonians H_1 and H_2 with set of eigenvectors Λ_1 and Λ_2, our above analogy suggests that $e^{-\mathrm{i}H_1 t}$ and $e^{-\mathrm{i}H_2 t}$ commute for all t if and only if $\Lambda_1 = \Lambda_2$. Recalling that H_1 and H_2 commute if and only if $\Lambda_1 = \Lambda_2$, our analogy suggests that

$$e^{-\mathrm{i}H_1 t} e^{-\mathrm{i}H_2 t} = e^{-\mathrm{i}H_2 t} e^{-\mathrm{i}H_1 t} \text{ for all } t \quad \Leftrightarrow \quad H_1 \text{ and } H_2 \text{ commute .} \tag{2.21}$$

In fact, for Hamiltonians H_1, H_2,..., H_J, the following are equivalent (see section 8.2 of the Appendix for a proof):

- $\prod_{j=1}^{J} e^{-\mathrm{i}H_{\tau(j)}t} = \prod_{j=1}^{J} e^{-\mathrm{i}H_{\tau'(j)}t}$ for all $t \geq 0$ and any permutations τ and τ' of $\{1, 2, \ldots, J\}$ (2.22)
- H_i and H_j commute for all i and j (2.23)
- $e^{-\mathrm{i}(H_1 + H_2 + \cdots + H_J)t} = e^{-\mathrm{i}H_1 t} e^{-\mathrm{i}H_2 t} \cdots e^{-\mathrm{i}H_J t}$ for all $t \geq 0$. (2.24)

Let us return to our algorithm $\mathcal{C} = \{(|\psi_0\rangle\,,\, H,\, T)\}$ where, in this section, H is time independent and is of the form $H = \sum_{j=1}^{J} H_j$. If the H_j commute, then we can implement the induced time-evolution $e^{-\mathrm{i}HT}$ as the right-hand side of equation 2.24 where for each $e^{-\mathrm{i}H_j T}$ we use the techniques of the last section. If each H_j satisfies the conditions

- $[H_j]$ may be decomposed as

$$[H_j] = \bigotimes_{i=1}^{I_j} [H_{j,i}]_{2^{k_i} \times 2^{k_{j,i}}},$$

where $k_{j,i} \in O(\log(\sigma(n)))$ for all i; and

- efficiently computable formulae for all elements of $[U_{j,i}]$ and $[D_{j,i}]$, such that $[H_{j,i}] = [U_{j,i}^\dagger][D_{j,i}][U_{j,i}]$, are available for all n,

then $\mathcal{C}$ can be implemented with $O(J\text{poly}(\sigma(n)))$ resources; if $J \in O(\text{poly}(\sigma(n)))$, then $\mathcal{C}$ is efficient.

If the H_j do not commute, then by statement 2.22 the operators $e^{-\mathrm{i}H_j T}$ do not commute in general. Recall from classical mechanics that although $R_{\hat{a}}(\theta)$ and $R_{\hat{b}}(\theta)$ do not commute for all θ, $R_{\hat{a}}(\theta)$ and $R_{\hat{b}}(\theta)$ *approximately* commute for extremely small angles θ. Thus, when H_1 and H_2 do not commute, our analogy suggests that $e^{-\mathrm{i}H_1 t}$ and $e^{-\mathrm{i}H_2 t}$ approximately commute for extremely small "angles" $\|\Delta H_1\| t$ and $\|\Delta H_2\| t$, that is,

$$e^{-\mathrm{i}H_1 t} e^{-\mathrm{i}H_2 t} \approx e^{-\mathrm{i}H_2 t} e^{-\mathrm{i}H_1 t} \quad \text{for sufficiently small } t.$$

By the equivalence of 23 and 24, this becomes

$$e^{-\mathrm{i}(H_1+H_2)t} \approx e^{-\mathrm{i}H_1 t} e^{-\mathrm{i}H_2 t} \quad \text{for sufficiently small } t.$$

In fact, one can easily derive the following asymptotic behaviour as $\epsilon > 0$ approaches 0:

$$e^{-\mathrm{i}(H_1+H_2+\cdots+H_J)\epsilon} = e^{-\mathrm{i}H_1\epsilon} e^{-\mathrm{i}H_2\epsilon} \cdots e^{-\mathrm{i}H_J\epsilon} + O\left(\epsilon^2 J^2 [\max_j \|H_j\|]^2\right); \tag{2.25}$$

more specifically, there exists a constant c (independent of both t and n) such that for all $\epsilon < 1/J \max_j \|H_j\|$,

$$\left\| e^{-\mathrm{i}(H_1+H_2+\cdots+H_J)\epsilon} - e^{-\mathrm{i}H_1\epsilon} e^{-\mathrm{i}H_2\epsilon} \cdots e^{-\mathrm{i}H_J\epsilon} \right\| < c\epsilon^2 J^2 [\max_j \|H_j\|]^2.$$

Recall that for a linear operator A

$$\begin{aligned} \|A\| &\equiv \max_{\langle\psi|\psi\rangle=1} \|A|\psi\rangle\| \\ &= \max\{|\lambda| : \quad \lambda \text{ is an eigenvalue of } A\}, \text{ if } A \text{ is Hermitian.} \end{aligned}$$

Noting that $\|\Delta H_j\| \leq 2\|H_j\|$, the error term implicitly depends on the size of the "angles" $\|\Delta H_j\|\,\epsilon$ as we would expect.

Suppose that $|\psi(t)\rangle$ evolves under H with $|\psi(0)\rangle = |\psi_0\rangle$. Ultimately, we would like to use the approximation

$$|\psi(T)\rangle \approx \left(e^{-\mathrm{i}H_1\Delta t}e^{-\mathrm{i}H_2\Delta t}\cdots e^{-\mathrm{i}H_J\Delta t}\right)^M |\psi_0\rangle\,, \qquad \Delta t = \frac{T}{M}.$$

For a desired fixed total error bound $\delta > 0$, we can calculate a sufficiently large M using the error term in equation 2.25 with $\epsilon = \frac{T}{M}$ as we now explain. From the discussion above, if

$$M > TJ \max_j \|H_j\| \tag{2.26}$$

then the total error incurred is at most $M \cdot c\left(\frac{T}{M}\right)^2 J^2[\max_j \|H_j\|]^2$. Setting this to be at most δ gives

$$M \geq \frac{T^2 J^2[\max_j \|H_j\|]^2}{\delta} c. \tag{2.27}$$

We assume that, for sufficiently large n, $TJ\max_j \|H_j\| > 1$ so that condition 2.27 implies 2.26. We highlight some points about inequality 2.27. Note that the efficiency of implementing the time-evolution $e^{-\mathrm{i}(H_1+H_2+\cdots+H_J)T}$ depends on the pure running time T; recall that the implementation of the time-evolution $e^{-\mathrm{i}DT}$ could always be done in constant time with respect to T. Note also the (implicit) dependence of M on the energy differences of H_j. Intuitively, using Hamiltonians that have larger energy differences must be more inefficient, since energy is a fundamental physical resource. Here we see where that inefficiency shows up in the simulation: the larger the energy differences in the Hamiltonians H_j, the more simulation steps required.

We have shown that $\mathcal{C}$ is efficient if

- each H_j satisfies conditions for H in 12 and 13; or 14 and 15; and (2.28)
- $TJ\max_j \|H_j\| \in O(\mathrm{poly}(\sigma(n)))$. (2.29)

Note that $H = \sum_{j=1}^{J} H_j$ is in principle diagonalizable and hence might be better viewed not as a sum but as a single-term Hamiltonian to be dealt with as in section 2.4.1. Whether H is best analyzed as a sum or not as a sum is dependent on the availability of efficiently computable formulae for elements of $[U]$ and $[D]$ such that $[H] = [U^\dagger][D][U]$ for diagonal matrix $[D]$. Also, equation 2.25 is not optimal: by carefully manipulating power series expansions, one can obtain [SS99] approximations that have error term $O(\epsilon^r J^r [\max_j \|H_j\|]^r)$ for $r = 3, 4, 5$ which lead to $M \sim (T^r J^r [\max_j \|H_j\|]^r)^{\frac{1}{r-1}}$. Still, these techniques for simulating the time-evolution of a Hamiltonian are generic; specific cases may have special structure that can be exploited to achieve a discrete-time algorithm having input and output the same as the given continuous-time algorithm but requiring even fewer physical resources than a generic simulation. An example is the case of Farhi and Gutmann's continuous-time search algorithm (see the end of section 3.2).

Note that we have come full circle in showing that indeed, under certain conditions, $H_{\text{Structure}}$ in equation 2.7 can be efficiently implemented in the laboratory as we initially suspected.

2.4.3 Time-Dependent Hamiltonians

Finally we consider continuous-time algorithms employing a time-dependent Hamiltonian $H(t)$, $0 \leq t \leq T$. Let $t_0 \equiv 0$ and $t_M \equiv T$, and partition the total time interval into M subintervals

$$[0, T] = [t_0, t_1) \cup [t_1, t_2) \cup \cdots \cup [t_{M-1}, t_M].$$

If $U(T, 0)$ is the total induced time-evolution operator, then we can write

$$U(T, 0) = U(t_M, t_{M-1}) \cdot U(t_{M-1}, t_{M-2}) \cdots U(t_{j-1}, t_j) \cdots U(t_2, t_1) \cdot U(t_1, t_0).$$

We approximate each factor above like this:

$$U(t_j, t_{j-1}) \approx e^{-\mathrm{i}H(t_{j-1})\Delta t_j}, \quad j = 1, 2, \ldots M, \tag{2.30}$$

where $\Delta t_j \equiv t_j - t_{j-1}$. Now let j "auto-increment" with time

$$j(t) \equiv \max\{i : \ t_i \leq t\}$$

and define the Hamiltonian $\tilde{H}(t)$

$$\tilde{H}(t) \equiv H(t_{j(t)}).$$

Thus $\tilde{H}(t)$ approximates $H(t)$ by a sequence of M time-independent Hamiltonians $(H(t_j))_{j=1,2,\ldots,M}$. Let $\tilde{U}(T,0)$ be the induced time-evolution operator corresponding to $\tilde{H}(t)$. The next fact [vDMV01] bounds the error incurred by using $\tilde{H}(t)$ instead of $H(t)$:

- if $\left\|H(t) - \tilde{H}(t)\right\| \leq \delta$ for all t, then $\left\|U(T,0) - \tilde{U}(T,0)\right\| \leq \sqrt{2T\delta}$. (2.31)

Below we work out the details for the specific form of time-dependent Hamiltonians employed by adiabatic quantum algorithms.

Let $s(t)$ be a smooth, nondecreasing, real function of time. Let H_0 and H_1 be two time-independent Hamiltonians. Let

$$H(t) = (1 - s(t))H_0 + s(t)H_1, \qquad s(0) = 0, \ \ s(T) = 1.$$

Consider approximating $H(t)$ on the interval $[t_{j-1}, t_j)$ as above, $t_{j-1} \leq t < t_j$:

$$\begin{aligned}
\left\|H(t) - \tilde{H}(t)\right\| &= (s(t) - s(t_{j-1}))\,\|H_0 - H_1\| \\
&\leq (s(t_j) - s(t_{j-1}))\,\|H_0 - H_1\| \\
&= \Delta t_j \cdot \frac{ds}{dt}(h) \cdot \|H_0 - H_1\|, \qquad \text{for some } h \in (t_{j-1}, t_j) \\
&\leq \Delta t_j \cdot \max_{t \in [t_{j-1}, t_j)} \left\{\frac{ds}{dt}(t)\right\} \cdot \|H_0 - H_1\|,
\end{aligned}$$

where we have used the Mean Value Theorem. Thus in solving for a sufficiently small time increment Δt_j, we must consider the size of the derivative of $s(t)$ in general. In the special case where $s(t) = \frac{t}{T}$, it is sufficient to partition the interval $[0,T]$ into equally-sized intervals of length $\Delta t = \frac{T}{M}$ and thus derive the bound

$$\left\|H(t) - \tilde{H}(t)\right\| \leq \frac{T}{M} \cdot \frac{1}{T} \cdot \|H_0 - H_1\| = \frac{1}{M} \cdot \|H_0 - H_1\|.$$

If the total error is to be less than $\delta > 0$, then by fact 31 we require

$$M > \frac{2T \left\| H_0 - H_1 \right\|}{\delta^2}. \tag{2.32}$$

Each term $e^{\mathrm{i}H(j\Delta t)\Delta t} = e^{\mathrm{i}\left(\left(1-\frac{j\Delta t}{T}\right)H_0 + \frac{j\Delta t}{T}H_1\right)\Delta t}$ must be approximated using the methods in the previous section. Recall that those approximation methods require M to be larger with respect to T, that is, they require $M \in \Omega(T^{\frac{r}{r-1}})$ for e.g. $r = 5$. Thus, we have shown that if H_0 and H_1 both satisfy conditions for H_j in 28 and $\max\{\left\|H_0\right\|, \left\|H_1\right\|\} \in O(\mathrm{poly}(\sigma(n)))$, then the algorithm $\{\left(|\psi_0\rangle,\ \left(1-\frac{t}{T}\right)H_0 + \frac{t}{T}H_1, T\right)\}$ is efficient if $T \in O(\mathrm{poly}(\sigma(n)))$.

In this chapter, we have defined what it means for a quantum algorithm to be efficient. We have given sufficient conditions for discrete- and continuous-time quantum algorithms to be considered efficient. Throughout the rest of this work we will refer to elements of this chapter when discussing efficiency or discrete-time simulations of continuous-time algorithms.

Chapter 3

Searching

The searching problem comes in a number of flavours, of which only one will be considered: Given a black-boxed quantum network, or *oracle*, C_f that computes the function $f : \{0,1\}^n \longrightarrow \{0,1\}$ such that

$$f(z) = \begin{cases} 1 & \text{if } z \neq w, \\ 0 & \text{if } z = w, \end{cases}$$

find w with high probability using a minimal number of queries to C_f (we seek an algorithm that works with this minimal number of queries for any $w \in \{0,1\}^n$). Other versions of the search problem may involve there being more than one solution to $f(z) = 0$, with the number of solutions being either known or unknown. Our focus is the case where it is known that there is only one solution: w. Note that C_f is given as a *black box*, that is, the internal workings of C_f are unknown – only queries to C_f are permitted. Each query to C_f is assumed to require $\Theta(1)$ physical resources.

The size of an instance of the black-box search problem can be taken to be n, that is, $\sigma_{\text{BLACK-BOX SEARCH}}(n) = n$. Thus an efficient algorithm for black-box searching would require only $O(\text{poly}(n))$ resources. It will be convenient for us to express the resource requirements of black-box search algorithms in terms of the parameter $N = 2^n$. This

reminds us that we are "searching through" N elements (for the problem of "unstructured database-search" (see section 6.5 in [NC00]), where we are given a database storing N elements, the parameter N more accurately represents the "size" of an instance; the black-box model hides all the complexity associated with maintaining a database).

If C_f is just a classical operator (which does not handle superpositions of basis states) and we restrict ourselves to classical computations, then the number of queries needed to discover w with sufficiently high probability is in $\Omega(N)$. However, using a quantum computer and having a quantum network C_f, a solution can be found with high probability using only $O(\sqrt{N})$ queries to C_f and, in fact, $O(\sqrt{N})$ total resources.

This chapter reviews three approaches to solving search-like problems using quantum mechanics. Grover's algorithm [G96] solves the search problem exactly as stated above and is a conventional discrete-time quantum algorithm: a sequence of quantum gates applied to a standard start state. The sequence can be written $(G, G, \ldots, G)$ for a gate G called the *Grover iterate* which incorporates the gate C_f. Farhi and Gutmann [FG98] solved a "continuous-time analogue" of the search problem, where, instead of a black-boxed quantum network C_f, we are given a black-boxed Hamiltonian $H_w \equiv E|w\rangle\langle w|$ with E known and w unknown. The problem is to find another Hamiltonian $H_{\mathrm{D}}(t)$ and minimize the total time needed to evolve under the total Hamiltonian $H_{\mathrm{D}}(t)+H_w$ in order to discover w with high probability. Finally, guided by simple intuition, Fenner [F00] discovered a Hamiltonian H_{Fenner} that depends on w such that, for a specific $|\psi_0\rangle$ that is independent of w, the algorithm $\{\big(|\psi_0\rangle\,,\,\frac{1}{N}H_{\mathrm{Fenner}},\,T\big)\}$ solves for w with $T \in O(\sqrt{N})$. Fenner's algorithm is the true continuous-time analogue of Grover's algorithm, as we explain shortly.

In each approach, there is a one-to-one correspondence between the computational basis states and the domain $\{0,1\}^n$ of f. As well, each algorithm uses an n-qubit principal quantum register (Grover's algorithm actually uses one ancillary qubit in addition to the n-qubit quantum register, but it can be ignored for now). If the instantaneous state of

the quantum computer expressed in the computational basis is $|\psi\rangle = \sum_{z=0}^{N-1} c_z |z\rangle$, then the result of a measurement in the computational basis will be w with probability $|c_w|^2$ (in general, the result will be the *eigenvalue* of the eigenket $|w\rangle$; but the observable can always be chosen so that the eigenvalue of $|z\rangle$ is the real number whose binary expansion is z, for all $z \in \{0,1\}^n$). Whatever the quantum mechanical solution, clearly its goal is to make $|c_w|^2$ as big as possible. If $|c_w|^2$ is made greater than, say, $\frac{1}{2}$, then the final measurement results in w with probability greater than $\frac{1}{2}$. This would render a quantum algorithm with probability of error less than $\frac{1}{2}$.

It is convenient at this point to define the uniform-amplitude superposition state

$$|u\rangle \equiv \frac{1}{\sqrt{N}} \sum_{z\in\{0,1\}^n} |z\rangle .$$

This state is used as the initial state in all of the following algorithms. Whatever the initial state is, it must be easy to prepare, so it may not, for example, depend on knowing anything about w, since w is unknown by assumption. The state $|u\rangle$ meets this criterion nicely. The *Walsh-Hadamard transform* (or just *Hadamard transform*), denoted W, maps the computational basis state $|0^n\rangle$ to $|u\rangle$, i.e.

$$|u\rangle = W|0^n\rangle .$$

The Hadamard transform may be written concisely as

$$W = \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} \sum_{y\in\{0,1\}^n} (-1)^{x\bullet y} |x\rangle\langle y|,$$

or, as a matrix in the computational basis,

$$[W] = \left(\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \right)^{\otimes n},$$

where "$x \bullet y$" denotes "$\sum_{i=1}^{n} x_i y_i$" if $x = x_n x_{n-1} \cdots x_1$ and $y = y_n y_{n-1} \cdots y_1$, and "$\otimes n$" denotes n-fold tensor product. The description of $[W]$ immediately implies W's efficient implementation as a quantum gate, since it is explicitly revealed to be the tensor product of n one-qubit gates. Note W is both unitary and Hermitian, thus $W^2 = \mathbb{I}$. The state $|0^n\rangle$ is considered easy to prepare, as is any element of the computational basis.

3.1 Intuition and Fenner's Hamiltonian

The simple idea of "making $|c_w|^2$ big" motivated Fenner [F00] to discover a Hamiltonian for quantum search. Fenner solves the problem,"Find a time-independent Hamiltonian (dependent on w) whose induced time-evolution maps $|u\rangle$ sufficiently close to $|w\rangle$ in a short time."

Recall that the infinitesimal time evolution governed by Hamiltonian H is $\mathbb{I} - iH\epsilon$ for $0 < \epsilon \ll 1$. Thus we can let this simple operator guide our intuition about how a continuous-time quantum algorithm behaves. We start with an intuitive idea about how a suitable algorithm might behave at each infinitesimally small time step, and then find a Hamiltonian H that implements the idea.

One way to pile lots of probability amplitude into $|w\rangle$ is, after each infinitesimally small period of time ϵ, to subtract amplitude proportional to $\epsilon \cdot c_w$ from all coefficients other than c_w, and add amplitude to c_w proportional to $\epsilon(\sum_{z \neq w} c_z)$. That is, we seek a Hamiltonian H_{Fenner} such that

$$(\mathbb{I} - \mathrm{i}H_{\text{Fenner}}\epsilon) \sum_z c_z |z\rangle = \sum_{z \neq w} (c_z - \epsilon \cdot c_w) |z\rangle + \left(c_w + \epsilon \sum_{z \neq w} c_z \right) |w\rangle .$$

It is easy to derive that

$$[H_{\text{Fenner}}] = \begin{bmatrix} 0 & \dots & 0 & -\mathrm{i} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & -\mathrm{i} & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & -\mathrm{i} & 0 & \dots & 0 \\ \mathrm{i} & \dots & \mathrm{i} & 0 & \mathrm{i} & \dots & \mathrm{i} \\ 0 & \dots & 0 & -\mathrm{i} & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & -\mathrm{i} & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & -\mathrm{i} & 0 & \dots & 0 \end{bmatrix},$$

where only the w_{th} row and column have nonzero entries. This Hamiltonian can be written concisely as

$$H_{\text{Fenner}} \equiv i\sqrt{N}(|w\rangle\langle u| - |u\rangle\langle w|).$$

Fenner proves that the unitary time evolution induced by $\frac{1}{N}H_{Fenner}$ maps $|u\rangle$ into $|w\rangle$ in time $T \in O(\sqrt{N})$. In fact, he proves that Grover's algorithm is a "true" simulation of his algorithm $\{\left(|u\rangle\,,\ \frac{1}{N}H_{\mathrm{Fenner}},\ T\right)\}$: there exists a time interval Δt_G such that

$$e^{-\mathrm{i}\frac{1}{N}H_{\mathrm{Fenner}}\Delta t_G} = G^2$$

where G is the Grover iterate.

3.2 Farhi and Gutmann's Search Problem

Farhi and Gutmann [FG98] solve a "continuous-time" version of the search problem. In place of the black box C_f, there is a Hamiltonian $H_w = E|w\rangle\langle w|$ where $w \in \{0,1\}^n$ is unknown but $E > 0$ is known. The goal is to discover w. In the physics laboratory, this situation corresponds to being given some conditions under which one must perform an experiment. For example, H_w may describe a magnetic field in which our qubits are immersed, where we are not allowed to control the electromagnet producing the magnetic field. To control the time-evolution of the system, the only recourse is to apply some other conditions to the system in addition to the given conditions. Mathematically, this corresponds to finding another Hamiltonian $H_{\mathrm{D}}(t)$ to add to H_w, for a total effective Hamiltonian $H_{\mathrm{D}}(t) + H_w$. The Hamiltonian $H_{\mathrm{D}}(t)$ is called the *driving Hamiltonian.* We can restate the problem as "Find $H_{\mathrm{D}}(t)$ such that the algorithm $\{(|u\rangle\,,\ H_{\mathrm{D}}(t) + H_w,\ T)\}$ solves for w in a minimal time T."

Farhi and Gutmann found that $H_{\mathrm{D}} = H_u \equiv E|u\rangle\langle u|$ solves the problem in time $T \in O(\sqrt{N})$. An outline of the proof follows.

Define the Hamiltonian H_{FG} as

$$H_{\mathrm{FG}} \equiv H_u + H_w.$$

Let $|\psi_{\mathrm{FG}}(t)\rangle$ be such that

$$\begin{cases} \frac{\partial}{\partial t}|\psi_{\mathrm{FG}}(t)\rangle & = -\mathrm{i}H_{\mathrm{FG}}|\psi_{\mathrm{FG}}(t)\rangle\,, \quad t \geq 0 \\ |\psi_{\mathrm{FG}}(0)\rangle & = |u\rangle\,. \end{cases}$$

It can be shown that $|\psi_{\mathrm{FG}}(t)\rangle$ remains in the 2-dimensional subspace $\mathrm{span}\{|w\rangle, |u\rangle\}$ for all t (we will see a proof of a result of this sort in section 6.2). The Gram-Schmidt procedure can be used to find an orthonormal basis $\{|w\rangle, |r\rangle\}$ for this subspace. Analyzing the time evolution in this basis is relatively easy, so the details are omitted. The result is that

$$|\psi_{\mathrm{FG}}(t)\rangle = e^{-\mathrm{i}Et}\left[\left(\frac{1}{\sqrt{N}}\cos\left(\frac{Et}{\sqrt{N}}\right) - \mathrm{i}\sin\left(\frac{Et}{\sqrt{N}}\right)\right)|w\rangle + c_r(t)\,|r\rangle\right],$$

for some coefficient $c_r(t)$. Thus, at time $t = \frac{\pi\sqrt{N}}{2E}$, a measurement in the computational basis renders w with probability 1.

Farhi and Gutmann also show that the choice $H_D(t) = H_u$ is in some sense optimal. This result is the continuous-time analogue of the lower bound of Bennett *et al.* [BBBV97]. Since $H_{\mathrm{D}}(t)$ is unrestricted, then it could be chosen such that the Hamiltonian $H_{\mathrm{D}}(t) + H_w$ induces the unitary operation corresponding to, say, the classical algorithm which tests $O(\mathrm{poly}(n))$ many elements of $\{0,1\}^n$ to see if one of them is the solution (see section 8.3 in the Appendix for the details of this algorithm). The resulting algorithm would solve the search problem in $O(\mathrm{poly}(n))$ time on the few instances where the solution is in this $O(\mathrm{poly}(n))$-sized subset of $\{0,1\}^n$. Thus the best optimality result we can hope for would say that no other driving Hamiltonian works faster than H_u on an appreciable fraction of search problem instances. Farhi and Gutmann show that any algorithm $\{(|u\rangle, H_D(t) + H_w, T)\}$ requires $T \in \Omega(\frac{\sqrt{N}}{E})$ if it is to work on *all* instances. We outline their proof in the following paragraph.

Define the states $|\psi_w(t)\rangle$ and $|\psi_D(t)\rangle$ by

$$\begin{cases} \frac{\partial}{\partial t}|\psi_w(t)\rangle & = -\mathrm{i}(H_D(t) + H_w)|\psi_w(t)\rangle\,, \quad 0 \leq t \leq T \\ |\psi_w(0)\rangle & = |\psi_0\rangle \end{cases}$$

and

$$\begin{cases} \frac{\partial}{\partial t}\left|\psi_D(t)\right\rangle & = -\mathrm{i}H_D(t)\left|\psi_D(t)\right\rangle, \quad 0 \le t \le T \\ \left|\psi_D(0)\right\rangle & = \left|\psi_0\right\rangle, \end{cases}$$

where w is the unique solution to the search problem instance. In order for the algorithm $\{(\left|\psi_0\right\rangle, H_D(t) + H_w, T)\}$ to work on all instances, we require $|| \left|\psi_w(T)\right\rangle - \left|\psi_D(T)\right\rangle ||^2 \ge \epsilon$ for some $\epsilon \in \Theta(1)$ for all but possibly one $w \in \{0,1\}^n$. Summing over all instances, we thus certainly require that

$$\sum_{w\in\{0,1\}^n} || \left|\psi_w(T)\right\rangle - \left|\psi_D(T)\right\rangle ||^2 \ge (N-1)\epsilon. \tag{3.1}$$

Using straightforward techniques, we can derive the bound

$$\frac{\partial}{\partial t} \sum_{w\in\{0,1\}^n} || \left|\psi_w(t)\right\rangle - \left|\psi_D(t)\right\rangle ||^2 \le 2E\sqrt{N}$$

which upon integrating from $t = 0$ to $t = T$ gives

$$\sum_{w\in\{0,1\}^n} || \left|\psi_w(T)\right\rangle - \left|\psi_D(T)\right\rangle ||^2 \le 2E\sqrt{N}T.$$

In order to satisfy 3.1, we thus require

$$T \ge \frac{\sqrt{N}\epsilon}{2E} - \frac{\epsilon}{2E\sqrt{N}}.$$

By allowing for algorithms that favour certain instances over others, we can get the slightly more general result that we hoped for. To do this, we require that $|| \left|\psi_w(T)\right\rangle - \left|\psi_D(T)\right\rangle ||^2 \ge \epsilon$ for only $g(N)$ instances, where $0 < g(N) < N$. Then 3.1 becomes

$$\sum_{w\in\{0,1\}^n} || \left|\psi_w(T)\right\rangle - \left|\psi_D(T)\right\rangle ||^2 \ge g(N)\epsilon$$

and the final bound on T is

$$T \ge \frac{g(N)\epsilon}{2E\sqrt{N}}. \tag{3.2}$$

We can consider a "useful" algorithm to be one that works on all but a small number of known elements which could be checked one by one after the algorithm completes. For

such an algorithm we thus require $N - g(N) \in O(\text{poly}(\log(N)))$ which makes the bound on T sufficiently large.

From the discussion on efficiency in section 2.4, we know that for a sufficiently small increment of time Δt, the following approximation is valid:

$$e^{-\mathrm{i}(H_u+H_w)\Delta t} \approx e^{-\mathrm{i}H_u\Delta t}e^{-iH_w\Delta t}.$$

Thus, if the system evolves for total time T, and $\Delta t \equiv \frac{T}{M}$, a valid simulation of Farhi and Gutmann's search algorithm is given by

$$|\psi(T)\rangle \approx \left(e^{-\mathrm{i}H_u\Delta t}e^{-\mathrm{i}H_w\Delta t}\right)^M |u\rangle\,,$$

for sufficiently large M. To implement this simulation, it suffices to be able to simulate $e^{-iH_u\Delta t}$ and $e^{-iH_w\Delta t}$ with quantum gates. Noting that $e^{-\mathrm{i}H_u\Delta t} = We^{-iE|0\rangle\langle 0|\Delta t}W$, the unitary operator $e^{-\mathrm{i}H_u\Delta t}$ can be efficiently implemented via the technique of section 2.4.1. But how can $e^{-\mathrm{i}H_w\Delta t} = e^{-\mathrm{i}E|w\rangle\langle w|\Delta t}$ be simulated without knowing w? Noting that H_w is diagonal in the computational basis, section 2.4.1 tells us that this operation can be simulated by a quantum network that has access to a quantum network C_λ that computes the eigenvalues of H_w; such a C_λ can easily be made out of the oracle C_f from the search problem, provided that the amount of precision needed to store the binary representation of E is not too large. We assume that $E \in O(\text{poly}(n))$ so that a sufficiently large M is polynomially related to the pure running time T. If we put $\Delta t = \pi/E$ in this simulation scheme, the resulting discrete-time algorithm is precisely Grover's algorithm which we look at next. Note, though, that in putting $\Delta t = \pi/E$ we no longer get a true simulation i.e. the state vector in Farhi and Gutmann's search algorithm strays far from the intermediate states reached in Grover's algorithm [F00].

3.3 Grover's Algorithm

From C_f we can construct the following efficient network:

$$O_f : |z\rangle \mapsto (-1)^{1-f(z)}\,|z\rangle\,, \qquad \text{for all } z \in \{0,1\}^n.$$

To do this, we can either use the technique detailed in section 2.4.1 which requires two C_f gates and one rotation gate on the ancillary qubit, or we can use just one C_f gate and have the ancillary qubit always in the state $(|0\rangle - |1\rangle)/\sqrt{2}$. Thus we can assume that the number of times the gate O_f appears in the final quantum algorithm is equal to the number of queries our algorithm makes to C_f.

Define the "inversion about mean" operator,

$$D \equiv 2|u\rangle\langle u| - \mathbb{I}.$$

Grover's algorithm:

- start with state $|u\rangle = W|0^n\rangle$;
- apply the operator DO_f approximately $\frac{\pi\sqrt{N}}{4}$ times; and
- perform a measurement in the computational basis.

Grover's algorithm is optimal in the following sense: Any sequence of quantum gates (applied to an easy-to-prepare start state) that solves the search problem via making queries to O_f must make $\Omega(\sqrt{N})$ such queries. The proof [BBBV97] is the discrete-time analogue of the optimality proof outlined in the previous section and is thus omitted.

To gain insight into why Grover's algorithm works, note first that D applied to a general state $\sum_{z\in\{0,1\}^n} \alpha_z |z\rangle$ gives

$$\sum_{z\in\{0,1\}^n} [-\alpha_z + 2\langle\alpha\rangle] |z\rangle ,$$

where $\langle\alpha\rangle \equiv \frac{1}{N}\sum_{z\in\{0,1\}^n} \alpha_z$ (the mean average of the coefficients). To illustrate what the algorithm does, it suffices to look carefully at how the first application of DO_f transforms $|u\rangle$. After the first time O_f is applied, all basis states have coefficient $\frac{1}{\sqrt{N}}$, except $|w\rangle$ which has coefficient $-\frac{1}{\sqrt{N}}$. Thus the mean average of the coefficients is $\frac{1}{\sqrt{N}} - \epsilon$, for some

$\epsilon \in \Theta(N^{-\frac{3}{2}})$. Applying D produces the state

$$\begin{aligned} DO_f |u\rangle &= \sum_{z \neq w} \left(-\frac{1}{\sqrt{N}} + 2\frac{1}{\sqrt{N}} - 2\epsilon \right) |z\rangle + \left(+\frac{1}{\sqrt{N}} + 2\frac{1}{\sqrt{N}} - 2\epsilon \right) |w\rangle \\ &= \sum_{z \neq w} \left(\frac{1}{\sqrt{N}} - 2\epsilon \right) |z\rangle + \left(3\frac{1}{\sqrt{N}} - 2\epsilon \right) |w\rangle . \end{aligned}$$

Thus, after only one iteration of DO_f applied to $|u\rangle$, probability amplitude has been added to $|w\rangle$ (note the similarity to the intuition behind Fenner's algorithm). By inspection, after each iteration of applying DO_f, the coefficient of $|w\rangle$ increases by roughly $\frac{2}{\sqrt{N}}$. Thus, it is plausible that after $O(\sqrt{N})$ iterations, the coefficient of $|w\rangle$ will be in $O(1)$ (for a different way of looking at the algorithm, see chapter 6 in [NC00]). The operator $G \equiv DO_f$ is the *Grover iterate*.

Briefly looking back to the simulation of Farhi and Gutmann's search algorithm $\{\left(|u\rangle, H_{\mathrm{FG}}, \frac{\pi\sqrt{N}}{2E}\right)\}$, note that for $\Delta t = \pi/E$, $e^{-iH_w \Delta t} = O_f$ and $e^{-iH_{\mathrm{D}} \Delta t} = -D$.

In this chapter, we have reviewed three quantum algorithms used to search for an element $w \in \{0,1\}^n$. One of the goals of this work is to exhibit a smooth transition from Farhi and Gutmann's search algorithm to the proposed adiabatic 3SAT algorithm.

Chapter 4

Adiabatic Quantum Computation

So far, all of the continuous-time quantum algorithms that have been discussed have used time-independent Hamiltonians. The adiabatic quantum computation paradigm uses time-varying Hamiltonians which are smooth deformations of one time-independent Hamiltonian into another time-independent Hamiltonian. In this chapter, we review the adiabatic theorem of quantum mechanics, the adiabatic approximation, and give an "adiabatic version" of Farhi and Gutmann's search algorithm.

4.1 The Adiabatic Theorem in Quantum Physics

The following is based on the presentation of the Adiabatic Theorem given in Messiah's book [M76]. Suppose a Hamiltonian $H(t)$ continuously deforms from an initial value H_0 at time t_0, into a final value H_1 at time t_1, that is

$$H(t_0) \equiv H_0 \qquad H(t_1) \equiv H_1.$$

An example of such a continuous deformation $H(t)$ is

$$H_{\text{e.g.}}(t) = \left(1 - \frac{t - t_0}{t_1 - t_0}\right) H_0 + \frac{t - t_0}{t_1 - t_0} H_1, \quad t_0 \leq t \leq t_1 \tag{4.1}$$

but, in general, the deformation does not have to be a weighted sum of H_0 and H_1. The goal is to analyze the unitary evolution induced by $H(t)$ as a function of the total duration $t_1 - t_0$. A cleaner way to do this analysis is to consider a family of reparametrisations of $H(t)$ as we now explain.

Imagine that H_0 and H_1 are distinct points in the space of all possible Hamiltonians and that $H(t)$ traces a curve H between the two points. Introduce the scaled time parameter $s(t)$ and the total duration T

$$s(t) \equiv \frac{t - t_0}{T} \qquad T \equiv t_1 - t_0.$$

Hence, reparametrize the curve H by changing the speed along it by a multiplicative constant, or *delay factor*, $\frac{ds}{dt} = \frac{1}{T}$, to get the curve $\tilde{H}$, defined by

$$\tilde{H}(s) \equiv H(t_0 + sT), \qquad 0 \leq s \leq 1,$$

where $\tilde{H}$ describes the same curve as H. Let the momentary unitary evolution induced by $H(t)$ and $\tilde{H}(s)$ be $U(t, t_0)$ and $U_T(s(t))$ respectively, so

$$U_T(s(t)) = U(t, t_0), \qquad t_0 \leq t \leq t_1.$$

Simply writing $U_T(s)$ for $0 \leq s \leq 1$, the unitary evolution may be analyzed as a function of T. The adiabatic theorem is a statement about $\lim_{T\to\infty} U_T(s)$, under certain conditions of $\tilde{H}(s)$, i.e. it considers what happens when the passage along the curve from H_0 to H_1 is made infinitely slowly. The "certain conditions" have to do with the eigenspaces and eigenvalues of $\tilde{H}(s)$. It is sufficient here to assume that $\tilde{H}(s)$ has a discrete spectrum of eigenvalues and that $\tilde{H}(s)$ acts on a Hilbert space of dimension N.

Let "diag$(d_1, d_2, \ldots, d_m)$" denote the $m \times m$ diagonal matrix with element d_i in the ith row (column). Define the eigenvalues $\tilde{\lambda}_i(s)$ of $\tilde{H}(s)$ for $i = 0, 1, \ldots N-1$ such that for all s

$$\tilde{H}(s) = U^\dagger(s) \cdot \text{diag}(\tilde{\lambda}_0(s), \tilde{\lambda}_1(s), \ldots, \tilde{\lambda}_{N-1}(s)) \cdot U(s) \quad \text{for some } U(s)$$

and

$$\tilde{\lambda}_0(s) \leq \tilde{\lambda}_1(s) \leq \cdots \leq \tilde{\lambda}_{N-1}(s).$$

Let $\tilde{P}_i(s)$ denote the projector onto the eigenspace corresponding to $\tilde{\lambda}_i(s)$ for $i = 0, 1, \ldots, N-1$.

Suppose that the following two conditions hold:

- $\tilde{\lambda}_0(s) < \tilde{\lambda}_1(s) < \cdots < \tilde{\lambda}_{N-1}(s)$ for all s (4.2)
- $\frac{d\tilde{P}_i(s)}{ds}$ and $\frac{d^2\tilde{P}_i(s)}{ds^2}$ are well defined and piecewise continuous for all i. (4.3)

Then the Adiabatic Theorem says that for all $i = 0, 1, \ldots, N-1$

$$\lim_{T\to\infty} \left(U_T(s)\tilde{P}_i(0)\right) = \tilde{P}_i(s) \lim_{T\to\infty} U_T(s).$$

If $|\phi_i(0)\rangle$ is an eigenstate of H_0, i.e. $\tilde{H}(s)\,|\phi_i(s)\rangle = \tilde{\lambda}_i(s)\,|\phi_i(s)\rangle$, then

$$\begin{aligned}
\lim_{T\to\infty} \left(U_T(s)\,|\phi_i(0)\rangle\right) &= \lim_{T\to\infty} \left(U_T(s)\tilde{P}_i(0)\,|\phi_i(0)\rangle\right) \\
&= \left[\lim_{T\to\infty} \left(U_T(s)\tilde{P}_i(0)\right)\right] |\phi_i(0)\rangle \\
&= \tilde{P}_i(s) \left[\lim_{T\to\infty} U_T(s)\right] |\phi_i(0)\rangle \\
&= \tilde{P}_i(s) \lim_{T\to\infty} \left(U_T(s)\,|\phi_i(0)\rangle\right).
\end{aligned}$$

In other words, in the limit $T \to \infty$, if the state is initially in the ith eigenstate of H_0, then the state remains in the corresponding ith eigenstate of $\tilde{H}(s)$ throughout the entire evolution $0 \leq s \leq 1$. If we only require this to occur for a particular $i = i_0$, then [K50] we only require less restrictive versions of conditions 4.2 and 4.3:

- $\tilde{\lambda}_{i_0-1}(s) < \tilde{\lambda}_{i_0}(s) < \tilde{\lambda}_{i_0+1}(s)$ for all s (4.4)
- $\frac{d\tilde{P}_{i_0}(s)}{ds}$ and $\frac{d^2\tilde{P}_{i_0}(s)}{ds^2}$ are well defined and piecewise continuous, (4.5)

where $\tilde{\lambda_{-1}}(s) \equiv -\infty$ and $\tilde{\lambda_N}(s) \equiv \infty$.

The *ground state* of a Hamiltonian H is the eigenstate of H corresponding to the eigenvalue of H which is less than or equal to all the other eigenvalues of H. In the case $i = i_0 = 0$, when the initial state is the ground state $|\phi_0(0)\rangle$ of the initial value $\tilde{H}(0) = H_0$ of the Hamiltonian, the adiabatic theorem can be illustrated by the following analogy. Liken the energy of the quantum state to a baby's level of excitement. Suppose the baby is rather unexcited, that is, the baby is sleeping in a crib. Now liken the baby's surroundings to the Hamiltonian governing the system. The baby is initially sleeping in the crib in the bedroom (analogous to H_0), and the parent wishes to move the baby's crib to the living room (analogous to H_1). The adiabatic theorem applied to this metaphor says that if the baby is a sufficiently sound sleeper (analogous to condition 4.4), and the path from the bedroom to the living room is sufficiently smooth (analogous to condition 4.5), then if the parent moves the crib infinitely slowly from the bedroom to the living room, the baby will not wake.

4.2 The Adiabatic Approximation

Moving the crib infinitely slowly is not a practical solution. Instead we know that it is sufficient that the crib is moved *slowly enough* so that the baby remains asleep. In quantum physics this idea is formalized in the *adiabatic approximation*.

Just how slowly does the parent have to move the baby? First we consider what are the sufficient conditions for the baby to remain asleep. Intuitively they should depend on the following three things: (1) *how* sound a sleeper the baby is, (2) *how* smooth the path from the bedroom to the living room is, and (3) *how* fast the baby is moved. As well, one might expect some sort of tradeoff among these three criteria: for example, the more sound a sleeper the baby is, then perhaps the less smooth the path from the bedroom from the living room has to be and the faster one can move the baby. Furthermore, if we always restrict our attention to paths that are, say, "very smooth" (analogous to

ensuring that condition 4.5 holds), then all that matter are (1) and (3). Returning to our mathematical problem, the above intuition is formalized below.

Suppose condition 3 above holds. Revert to the physical time parameter t so that the set of eigenvalues of $H(t)$ is denoted $\{\lambda_i(t) : i = 0, 1, \ldots\}$, that is,

$$\lambda_i(t) \equiv \tilde{\lambda}_i(\frac{t - t_0}{T}).$$

Let $|\phi_i(t)\rangle$ be an eigenstate of $H(t)$ corresponding to $\lambda_i(t)$:

$$H(t)\,|\phi_i(t)\rangle = \lambda_i(t)\,|\phi_i(t)\rangle\,, \qquad t_0 \leq t \leq t_1.$$

Let $|\psi(t)\rangle$ be the state of the system:

$$\mathrm{i}\frac{\partial}{\partial t}\,|\psi(t)\rangle = H(t)\,|\psi(t)\rangle\,, \qquad t_0 \leq t \leq t_1.$$

Suppose the system starts in an eigenstate of the initial value of the Hamiltonian, i.e. $|\psi(t_0)\rangle = |\phi_a(t_0)\rangle$. Consider the leakage of probability amplitude from the ath eigenstate into the ith eigenstate, $i \neq a$. Using time-dependent perturbation theory (see chapter 17 in [M76]), we get an approximation to the probability of finding the final state in the ith eigenstate:

$$|\,\langle\phi_i(t_1)|\psi(t_1)\rangle\,|^2 \approx \left| \int_{t_0}^{t_1} \frac{\langle\phi_i\,(t)|\left(\frac{dH(t)}{dt}\right)|\phi_a\,(t)\rangle}{(\lambda_i\,(t) - \lambda_a\,(t))} e^{\mathrm{i}\int_{t_0}^{t}(\lambda_i(t') - \lambda_a(t'))dt'} dt \right|^2 . \tag{4.6}$$

This integral can be upper-bounded by considering the minimum value of $|\lambda_i(t) - \lambda_a(t)|$ and maximum value of $|\,\langle\phi_i\,(t)|\left(\frac{dH(t)}{dt}\right)|\phi_a\,(t)\rangle\,|$ in the interval $[t_0, t_1]$ and assuming that these two quantities are approximately constant with respect to time. Thus, barring exceptions[1] [M76], we get an approximate bound on the total probability of finding the

[1]Such an exception may occur if

$$\max_t \sum_{i \neq a} \left|\langle\phi_i\,(t)|\left(\frac{dH\,(t)}{dt}\right)|\phi_a\,(t)\rangle\right|^2 \approx \sum_{i \neq a} \max_t \left|\langle\phi_i\,(t)|\left(\frac{dH\,(t)}{dt}\right)|\phi_a\,(t)\rangle\right|^2$$

is a bad approximation.

final state in some eigenstate other than $|\phi_a(t_1)\rangle$:

$$1 - |\langle\phi_a(t_1)|\psi(t_1)\rangle|^2 \lesssim \frac{\max_t \sum_{i\neq a} \left|\langle\phi_i(t)|\left(\frac{dH(t)}{dt}\right)|\phi_a(t)\rangle\right|^2}{\min_{i\neq a,t}|\lambda_i(t)-\lambda_a(t)|^4}$$

(where "$\lesssim$" denotes "is less than a quantity which is approximately equal to"). Noting that [vD02]

$$\begin{aligned}\sum_{i\neq a}\left|\langle\phi_i(t)|\left(\frac{dH}{dt}\right)|\phi_a(t)\rangle\right|^2 &\leq \sum_i \left|\langle\phi_i(t)|\left(\frac{dH}{dt}\right)|\phi_a(t)\rangle\right|^2 \\ &= \left\|\frac{dH}{dt}|\phi_a(t)\rangle\right\|^2 \\ &\leq \left\|\frac{dH}{dt}\right\|^2,\end{aligned}$$

we can simplify this bound and get a sufficient condition for ending up in the eigenstate $|\phi_a(t_1)\rangle$ with high probability:

$$\frac{\max_t \left\|\frac{dH(t)}{dt}\right\|^2}{\min_{i\neq a,t}|\lambda_i(t)-\lambda_a(t)|^4} \ll 1. \tag{4.7}$$

More precisely, for $0 < \epsilon < 1$, if

$$\frac{\max_t \left\|\frac{dH(t)}{dt}\right\|^2}{\min_{i\neq a,t}|\lambda_i(t)-\lambda_a(t)|^4} < \epsilon,$$

then

$$|\langle\phi_a(t_1)|\psi(t_1)\rangle|^2 \gtrsim 1-\epsilon.$$

Recall the example Hamiltonian $H_{\text{e.g.}}(t)$,

$$H_{\text{e.g.}}(t) = \left(1-\frac{t-t_0}{t_1-t_0}\right)H_0 + \frac{t-t_0}{t_1-t_0}H_1, \qquad t_0 \leq t \leq t_1.$$

Note that

$$\frac{dH_{\text{e.g.}}(t)}{dt} = \frac{-H_0+H_1}{t_1-t_0} = \frac{H_1-H_0}{T}$$

so that in the special case $H(t) = H_{\text{e.g.}}$, condition 4.7 becomes

$$\frac{\|H_1 - H_0\|}{\min_{i \neq a,\, t} |\lambda_i(t) - \lambda_a(t)|^2} \ll T; \tag{4.8}$$

that is, if

$$\frac{\|H_1 - H_0\|}{\epsilon \cdot \min_{i \neq a,\, t} |\lambda_i(t) - \lambda_a(t)|^2} < T$$

then

$$|\langle \phi_a(t_1) | \psi(t_1) \rangle|^2 \gtrsim 1 - \epsilon.$$

Since $\frac{1}{T}$ is the delay factor, we have answered the question "At what (nonzero) speeds can the parent move the baby to ensure that it likely remains asleep?" in the special case of $H(t) = H_{\text{e.g.}}(t)$.

4.3 Adiabatic Search

How can the adiabatic approximation result be used to search? In all of the continuous-time quantum search algorithms looked at so far, the start state is $|u\rangle$ and the final state is $|w\rangle$. Suppose we seek a Hamiltonian $H_{\text{AS}}(t)$, $0 \leq t \leq T$, that will accomplish searching for w in the spirit of the adiabatic approximation, starting with state $|u\rangle$. Then we require that $|u\rangle$ is an eigenstate of $H_{\text{AS}}(0)$ and that $|w\rangle$ is an eigenstate of $H_{\text{AS}}(T)$. We also require that $|w\rangle$ be derived from $|u\rangle$ by continuity: that is, we require the existence of a continuous function $\lambda_{\text{AS}}(t)$ such that, for all t, $0 \leq t \leq T$,

$$H_{\text{AS}}(t) \, |\lambda_{\text{AS}}(t)\rangle \;\; = \;\; \lambda_{\text{AS}}(t) \, |\lambda_{\text{AS}}(t)\rangle$$

for some eigenstate $|\lambda_{\text{AS}}(t)\rangle$ of $H_{\text{AS}}(t)$ satisfying

$$\begin{aligned} |\lambda_{\text{AS}}(0)\rangle &= |u\rangle \\ |\lambda_{\text{AS}}(T)\rangle &= |w\rangle . \end{aligned}$$

Noting that $|u\rangle$ is an eigenstate of $H_u \equiv E|u\rangle\langle u|$ and $|w\rangle$ is an eigenstate of $H_w \equiv E|w\rangle\langle w|$, we try the following for $H_{\mathrm{AS}}(t)$:

$$H_{\mathrm{AS}}(t) \equiv \left(1 - \frac{t}{T}\right) H_u + \frac{t}{T} H_w, \qquad 0 \leq t \leq T.$$

Note that $2H_{\mathrm{AS}}(T/2)$ equals H_{FG} from Farhi and Gutmann's search algorithm.

Since condition 4.3 clearly holds for $H_{\mathrm{AS}}(t)$, it remains to do an eigenvalue analysis to see if there is sufficient eigenvalue non-crossing in order to use the adiabatic approximation to solve for a sufficiently large running time T. Luckily, since the induced unitary evolution is restricted to the subspace $\mathrm{span}\{|u\rangle, |w\rangle\}$ as in Farhi and Gutmann's continuous-time search algorithm, the eigenvalue problem for $H_{\mathrm{AS}}(t)$ is easily solved. The result is that all but two eigenvalues are 0 for all t, with the two nonzero eigenvalues $\lambda^+(t)$ and $\lambda^-(t)$ such that

$$\lambda^{\pm}(t) = E\left(\frac{1}{2} \pm \sqrt{\frac{1}{4} - \left(1 - \frac{t}{T}\right)\left(\frac{t}{T}\right)\left(\frac{N-1}{N}\right)}\right)$$

and thus

$$\lambda^+(t) - \lambda^-(t) = E\sqrt{\frac{N - 4(N-1)(\frac{t}{T} - (\frac{t}{T})^2)}{N}} > 0, \qquad 0 \leq t \leq T, \tag{4.9}$$

with

$$\min_t(\lambda^+(t) - \lambda^-(t)) = \left(\lambda^+\left(\frac{T}{2}\right) - \lambda^-\left(\frac{T}{2}\right)\right) = \frac{E}{\sqrt{N}}. \tag{4.10}$$

Evidently, $\lambda^+(t)$ corresponds to the sought after function $\lambda_{\mathrm{AS}}(t)$ since

$$H_{\mathrm{AS}}(0)\,|u\rangle = \lambda^+(0)\,|u\rangle = E\,|u\rangle$$

and

$$H_{\mathrm{AS}}(T)\,|w\rangle = \lambda^+(T)\,|w\rangle = E\,|w\rangle\,.$$

Inequality 4.9 guarantees that there exists a sufficiently large value of T such that the adiabatic approximation is good. Combining equation 4.10 with condition 4.8 and noting

that $\|H_1 - H_0\| \leq \|H_1\| + \|H_0\| \leq 2E$, a sufficiently large value for T is $T \in \Omega(N/E)$. Thus we have an adiabatic search algorithm $\{(|u\rangle, H_{\mathrm{AS}}(t), T)\}$ with $T \in \Theta(N/E)$.

Assume that $E \in O(1)$ so that minimal overhead is incurred in the simulation of this algorithm (see section 2.4.2, inequality 2.27). Under this assumption, the pure running time of this adiabatic search algorithm is $T \in O(N)$. Using the simulation techniques of section 2.4, we can show that this adiabatic search algorithm requires $O(N^{\frac{r}{r-1}})$ resources to implement, for, say, $r = 5$.

A pure running time of $T \in \Omega(N)$ is significantly larger than the pure running time in $O(\sqrt{N})$ that was found in Farhi and Gutmann's quantum search algorithm. Can the adiabatic paradigm be used to achieve a continuous-time algorithm having pure running time in $O(\sqrt{N})$? The answer is yes [vDMV01]. To describe the method precisely, we use the scaled time parameter $s(t)$. Consider the Hamiltonian

$$H_{\mathrm{AS}'}(t) = (1 - s(t))H_u + s(t)H_w,$$

where instead of using a constant delay factor $\frac{ds}{dt} = \frac{1}{T}$ we allow $\frac{ds}{dt}$ to vary. That is we define the varying delay factor $\frac{1}{\tau(s)} \equiv \frac{ds}{dt}$ and the Hamiltonian

$$\tilde{H}_{\mathrm{AS}'}(s) \equiv H_{\mathrm{AS}'}(t_0 + \int_0^s \tau(s')ds').$$

Note that $s(t)$ is no longer a linear function of t, that is $H_{\mathrm{AS}'}(t)$ is *not* of the form $H_{\mathrm{e.g.}}$ from equation 4.1. We can imagine partitioning the interval $[t_0, t_1]$ into tiny subintervals and finding a sufficiently large constant delay factor for each subinterval using condition 4.8. In the limit as the length of all of the subintervals goes to 0, this process yields a time-varying delay factor defined on the entire interval $[t_0, t_1]$. Equation 4.9 gives the sufficiently small value of the delay factor $\frac{1}{\tau(s)}$ at each point in the evolution:

$$\tau(s) \gg \frac{\|H_w - H_u\|}{\left(\lambda^+(s) - \lambda^-(s)\right)^2} = \|H_w - H_u\| \frac{N/E^2}{N - 4(N-1)(s - s^2)}, \qquad 0 \leq s \leq 1.$$

Using this "form-fitting" delay factor, one recovers the pure running time of Farhi and

Gutmann's search algorithm, that is, for $E = 1$,

$$\int_0^1 \tau(s)ds \in O(\sqrt{N}).$$

Thus we have a revised adiabatic search algorithm $\{(|u\rangle\,,\, H_{\mathrm{AS}'}(t),\, T)\}$ that requires only $T \in \Theta(\sqrt{N})$.

Since $s(t) \neq \frac{t}{T}$, the number of resources needed to simulate this adiabatic algorithm is not given by the method in section 2.4.3. Because the derivative $\frac{ds}{dt}(t) = \frac{1}{\tau(s(t))}$ gets larger near the endpoints of $[0, T]$, this method requires that the time steps Δt_j near the endpoints be smaller causing more overhead. Setting $E = 1$ for convenience, note that we require $\frac{ds}{dt}(0) \ll 1$ and $\frac{ds}{dt}(T) \ll 1$. We can get a loose upper bound on the number of required time-steps M by calculating M assuming that $\Delta j = \frac{T}{M}$ and $\frac{ds}{dt} \in \Theta(1)$ throughout the entire interval $[0, T]$. Under these assumptions, inequality 2.32 gives $M \in \Omega(T^2)$. Thus, using this simulation technique, we could implement the algorithm $\{(|u\rangle\,,\, H_{\mathrm{AS}'}(t),\, T)\}$, $T \in \Theta(\sqrt{N})$, with $O(N)$ resources. A tighter bound on M is likely achievable. Perhaps there is an altogether better simulation technique for this algorithm. Or, perhaps the best we can hope for is just to use Grover's algorithm as a "simulation". We leave these issues for future work.

4.4 Adiabatic Searching by Staying in the Ground State

When using the adiabatic paradigm, it is convenient to work with algorithms $\{(|\psi_0\rangle\,,\, H(t),\, T)\}$ where $|\psi_0\rangle$ is the ground state of $H(0)$ i.e. where the state of the quantum computer remains in the ground state of the Hamiltonian throughout the entire evolution. All adiabatic algorithms considered hereafter will be of this type. The adiabatic search algorithm of the previous section can be converted into such an algorithm by substituting for H_u and H_w the Hamiltonians

$$H'_u = 0|u\rangle\langle u| + E\sum_{z \neq w} W|z\rangle\langle z|W$$

and

$$H'_w = 0|w\rangle\langle w| + E\sum_{z\neq w}|z\rangle\langle z|,$$

where $E > 0$. It is easy to verify that the distance between the smallest and next smallest eigenvalue of the new Hamiltonian $(1-s)H'_u + sH'_w$ is the same as the distance between $\lambda^+(s)$ and $\lambda^-(s)$.

We have reviewed the adiabatic approximation theory and developed an "adiabatic version" of Farhi and Gutmann's search algorithm.

Chapter 5

The Proposed Adiabatic 3SAT Algorithm

In this chapter, we show a natural progression from the adiabatic search algorithm to the proposed adiabatic 3SAT algorithm. We also briefly look at the classical intuition behind the adiabatic 3SAT algorithm in the spirit of Fenner (see section 3.1).

5.1 Definition of 3SAT

A Boolean formula f on the n Boolean variables $z_1, z_2, \ldots, z_n$ is said to be in *conjunctive normal form* (CNF) if and only if

$$f = \bigwedge_{j=1}^{m} \left(\bigvee_{k=1}^{K_j} a_{j,k} \right), \quad \text{where } a_{j,k} \in \{z_1, z_2, \ldots, z_n, \overline{z_1}, \overline{z_2}, \ldots, \overline{z_n}\}.$$

We assume that "$\overline{z_i}$" means "the logical negation of z_i", "$\bigvee$" means "logical OR", and "$\bigwedge$" means "logical AND". Further, f is said to be a *3-CNF* formula if and only if $K_j = 3$ for all j in the above expression. Each subformula $\left(\bigvee_{k=1}^{K_j} a_{j,k}\right)$ is called a *clause* of f.

I focus on the following special case of *3-CNF Satisfiability* (3SAT): Given a 3-CNF Boolean formula f on n variables $z_1, z_2, \ldots, z_n$ such that

- the number m of clauses comprising f is in $O(\text{poly}(n))$, and
- there is a unique satisfying Boolean assignment,

find the satisfying assignment (in most texts, "3SAT" is actually the corresponding *decision problem* (see [GJ79]) which is equivalent in difficulty to what we call "3SAT"). Recalling the terminology of section 2, $m \in O(\text{poly}(n))$ implies that we may take

$$\sigma_{3\text{SAT}}(n) = n,$$

that is, the size of a 3SAT instance is polynomially related to n. Thus an efficient algorithm for 3SAT must use $O(\text{poly}(n))$ total resources.

Corresponding to the formula f is the Boolean *function* mapping $\{0,1\}^n$ into $\{0,1\}$, where we employ the natural isomorphism between all truth assignments of the variables $z_1, \ldots, z_n$ and all n-bit binary strings $z \in \{0,1\}^n$ (we take 0 to correspond to TRUE and 1 to correspond to FALSE). Via the obvious abuse of notation, denote this Boolean function by $f(z)$. This isomorphism will be implicitly invoked throughout the discussion. For instance, let the unique satisfying assignment be $w \in \{0,1\}^n$.

5.2 From Searching to 3SAT

The above 3SAT problem looks a lot like the search problem. In both cases, there is a Boolean function $f(z)$ and the goal is to find the only solution to $f(z) = 0$. The difference is that, in 3SAT, we have full knowledge about how $f(z)$ computes its output: a quantum network C_f computing $f(z)$ need not be considered as a black box anymore since we may use the Boolean formula f to construct C_f.

Suppose we construct the quantum network C_f and treat it as a black box. Suppose we then try to solve 3SAT by "searching" for the solution w using a discrete-time quantum algorithm. From the outset, the optimality of Grover's algorithm dashes any hopes that we could find w with fewer queries to C_f than roughly $\sqrt{2^n}$. Since each query takes $\Theta(1)$ time, any search-type scheme using the black box C_f will take time $\Omega(\sqrt{2^n})$. Although Grover's algorithm is faster than any known classical 3SAT algorithm, the goal is to find a quantum algorithm that solves 3SAT in time $O(\text{poly}(n))$.

A different approach is to use the structure of the formula f to our advantage and not treat C_f as a black box. Instead of just asking "*Is f satisfied* by z?" we can ask "*How satisfied* is f by z?" That is, given z, we can quickly calculate how many clauses of f are violated by z. Thus we define a new function $v : \{0,1\}^n \longrightarrow \{0,1,\ldots\}$,

$$v(z) \equiv \text{the number of clauses of } f \text{ violated by assignment } z.$$

Let C_v be a quantum network computing the function $v(z)$. We can think of C_v as a better oracle than C_f. In fact, the query complexity of 3SAT given C_v (and given no other information about the formula f) has been shown [vDMV01] to be in $O(\text{poly}(n))$: with $O(n^3)$ queries to C_v, one can acquire enough information to construct the formula f. In contrast, one requires $O(\sqrt{2^n})$ queries to C_f to solve 3SAT. Can this apparent extra power of C_v be harnessed in a quantum algorithm?

We can rewrite the Hamiltonian $H'_w = 0|w\rangle\langle w| + E\sum_{z\neq w}|z\rangle\langle z|$ used in the adiabatic search as

$$H'_w = \sum_{z\in\{0,1\}^n} f(z)|z\rangle\langle z|,$$

where $f(z)$ is the function that defines that search problem and we have set $E = 1$ for clarity. We aim to construct a continuous-time adiabatic algorithm $\{(|u\rangle\,,\, H_{3\text{SAT}}(t),\, T)\}$ for 3SAT instances that takes advantage of any extra power that the function $v(z)$ has over the binary function $f(z)$ corresponding to the 3-CNF formula f. Thus replacing f

with v above gives a suitable candidate for the final value of the Hamiltonian as

$$H_{3\mathrm{SAT}}(T) = \sum_{z\in\{0,1\}^n} v(z)|z\rangle\langle z|,$$

whose ground state is indeed $|w\rangle$. Thus $H_{3\mathrm{SAT}}(T)$ encodes the function $v(z)$ just like H'_w encodes the function $f(z)$ in the search problem. What might be a suitable initial value $H_{3\mathrm{SAT}}(0)$? In the adiabatic search algorithm, we used $H'_u = 0|u\rangle\langle u| + E\sum_{z\neq w} W|z\rangle\langle z|W$ which can be rewritten

$$H'_u = \sum_{z\in\{0,1\}^n} h_{\mathrm{SEARCH}}(z)W|z\rangle\langle z|W,$$

where $h_{\mathrm{SEARCH}}(z) \in \{0,1\}$ and $h_{\mathrm{SEARCH}}(z) = 0$ if and only if $z = 0^n$. Note that, unlike the function $f(z)$, the codomain of the function $v(z)$ is not restricted to $\{0,1\}$ but is $\{0,1,\ldots\}$. Using these two ideas, a suitable candidate for the initial value of the Hamiltonian is

$$H_{3\mathrm{SAT}}(0) = \sum_{z\in\{0,1\}^n} h(z)W|z\rangle\langle z|W,$$

for some function $h : \{0,1\}^n \longrightarrow \{0,1,\ldots\}$ where $h(z) = 0$ if and only if $z = 0^n$ to ensure that the ground state of $H_{3\mathrm{SAT}}(0)$ is $|u\rangle = W|0\rangle$. Thus we have a candidate continuous-time adiabatic algorithm $\{(|u\rangle, H_{3\mathrm{SAT}}(t), T)\}$ where

$$H_{3\mathrm{SAT}}(t) \equiv \left(1 - \frac{t}{T}\right) \sum_{z\in\{0,1\}^n} h(z)W|z\rangle\langle z|W + \frac{t}{T} \sum_{z\in\{0,1\}^n} v(z)|z\rangle\langle z|, \qquad 0 \leq t \leq T.$$

Of course, we have not proved that there exists some value of T such that this algorithm is correct since we have not proved that the two smallest eigenvalues of $H_{3\mathrm{SAT}}(t)$ never meet in the interval $[t_0, t_1]$ (see condition 4.4 with $i_0 = 0$). Even if the smallest two eigenvalues do not meet, they may still come too close to each other for most 3SAT instances. This proposed adiabatic 3SAT algorithm is just like the adiabatic search algorithm but with the initial and final Hamiltonian values having more than two energy levels (distinct energy eigenvalues). It is unclear whether this added complexity helps to solve 3SAT instances faster than Grover's algorithm.

We stress the natural progression from Farhi and Gutmann's search algorithm (section 3.2), to the first adiabatic search algorithm (section 4.3), to the second adiabatic search algorithm (section 4.4), to this candidate 3SAT algorithm $\{(|u\rangle\,,\, H_{3\mathrm{SAT}}(t),\, T)\}$. Because of the connection with Farhi and Gutmann's search algorithm, we refer to the initial Hamiltonian $H_{3\mathrm{SAT}}(0)$ as the *driving Hamiltonian*.

From the discussion in section 2.4, the time-evolution induced by $H_{3\mathrm{SAT}}(t)$ can be simulated with $O([T \max_z v(z) \max_z h(z)]^{r/(r-1)})$ resources, for e.g. $r = 5$, as long as we have access to small quantum networks C_v and C_h that compute the energy eigenvalue functions $v(z)$ and $h(z)$. We know that C_v is easily constructed given the Boolean formula f. However, we still do not know how to choose the function $h(z)$ given the 3SAT instance. This development of the adiabatic 3SAT algorithm is not the one presented in the original paper [FGGS00] by Farhi, Gutmann, Goldstone, and Sipser (FGGS). Next we give FGGS's presentation which specifies $h(z)$.

5.3 FGGS's Definition of the Proposed Adiabatic 3SAT Algorithm

FGGS define the initial and final Hamiltonian values as sums of operators each acting on at most three qubits. Let C be a clause of the Boolean formula f involving the Boolean variables z_{i_C}, z_{j_C}, and z_{k_C} or their negations.

First we give FGGS's definition of the final value H_{F} of the Hamiltonian. For each clause C, define the function $h_C : \{0,1\} \times \{0,1\} \times \{0,1\} \longrightarrow \{0,1\}$

$$h_C(a,b,c) = \begin{cases} 1 & \text{if } z_{i_C} := a,\ z_{j_C} := b,\ z_{k_C} := c \text{ makes } C \text{ FALSE}, \\ 0 & \text{otherwise.} \end{cases}$$

Define the Hamiltonian $H_{\mathrm{F,C}}$ for each clause by its action on the computational basis $\{0,1\}^n$:

$$H_{\mathrm{F,C}}\,|z\rangle = h_C\left((z)_{i_C}, (z)_{j_C}, (z)_{k_C}\right)|z\rangle\,, \qquad z \in \{0,1\}^n,$$

where $(z)_i$ is the value of the ith bit (from the right) of the basis state label $z = z_n z_{n-1} \cdots z_1$, for $i = 1, 2, \ldots, n$. It is clear that each $H_{\mathrm{F,C}}$ acts only on three qubits. The final Hamiltonian is

$$H_{\mathrm{F}} = \sum_C H_{\mathrm{F,C}}.$$

It is straightforward to verify that this definition is equivalent to our definition of the final Hamiltonian i.e. that $H_{\mathrm{F}} = H_{3\mathrm{SAT}}(T)$:

$$\begin{aligned} H_{\mathrm{F}} |z\rangle &= \sum_C H_{\mathrm{F,C}} |z\rangle \\ &= \sum_C h_C \left((z)_{i_C}, (z)_{j_C}, (z)_{k_C} \right) |z\rangle \\ &= \left(\sum_C h_C \left((z)_{i_C}, (z)_{j_C}, (z)_{k_C} \right) \right) |z\rangle \\ &= v(z) |z\rangle, \qquad \text{for all } z \in \{0,1\}^n. \end{aligned}$$

Now we define the initial Hamiltonian H_{I}. Define the operator $Z^{(i)}$:

$$[Z^{(i)}] \equiv [\mathbb{I}]_{2^{n-i} \times 2^{n-i}} \otimes [Z] \otimes [\mathbb{I}]_{2^{i-1} \times 2^{i-1}},$$

which is the Z operator acting on the ith qubit and the identity operator acting on the rest of the qubits. Similarly, define $W^{(i)}$ as

$$[W^{(i)}] \equiv [\mathbb{I}]_{2^{n-i} \times 2^{n-i}} \otimes \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \otimes [\mathbb{I}]_{2^{i-1} \times 2^{i-1}}$$

which is the one-qubit Hadamard gate acting on the ith qubit and the identity operator acting on the rest of the qubits. Define the pairwise-commuting one-qubit Hamiltonians

$$H_{\mathrm{I}}^{(i)} \equiv W^{(i)} \frac{1}{2} \left(\mathbb{I} - Z^{(i)} \right) W^{(i)}, \qquad i = 1, 2, \ldots, n.$$

For each clause C define the following three-qubit Hamiltonian

$$H_{\mathrm{I,C}} \equiv H_{\mathrm{I}}^{(i_C)} + H_{\mathrm{I}}^{(j_C)} + H_{\mathrm{I}}^{(k_C)}$$

and finally the initial Hamiltonian

$$\begin{aligned} H_{\mathrm{I}} &\equiv \sum_{C} H_{\mathrm{I}}^{(i_C)} + H_{\mathrm{I}}^{(j_C)} + H_{\mathrm{I}}^{(k_C)} \\ &= \sum_{i=1}^{n} d_i H_{\mathrm{I}}^{(i)}, \end{aligned}$$

where d_i is the number of clauses in which the Boolean variable z_i appears. The following sequence of manipulations shows that H_{I} is of the form of our initial Hamiltonian value $H_{3\mathrm{SAT}}(0)$:

$$\begin{aligned} H_{\mathrm{I}} &= \sum_{i=1}^{n} d_i H_{\mathrm{I}}^{(i)} \\ &= \frac{1}{2} W \left[\sum_{i=1}^{n} d_i \left(\mathbb{I} - Z^{(i)} \right) \right] W \\ &= \frac{1}{2} W \left[\left(\sum_{i=1}^{n} d_i \right) \mathbb{I} - \sum_{i=1}^{n} d_i \overbrace{\sum_{z \in \{0,1\}^n} (-1)^{z_i} |z\rangle\langle z|}^{Z^{(i)}} \right] W \\ &= \frac{1}{2} \left[\left(\sum_{i=1}^{n} d_i \right) \sum_{z \in \{0,1\}^n} W|z\rangle\langle z|W - \sum_{z \in \{0,1\}^n} \sum_{i=1}^{n} d_i (-1)^{z_i} W|z\rangle\langle z|W \right] \\ &= \sum_{z \in \{0,1\}^n} \left(\frac{1}{2} \sum_{i=1}^{n} d_i \left(1 - (-1)^{z_i}\right) \right) W|z\rangle\langle z|W. \end{aligned} \tag{5.1}$$

Note that $\left(\frac{1}{2}\sum_{i=1}^{n} d_i \left(1 - (-1)^{z_i}\right)\right)$ is 0 if and only if $z = 0^n$, and is greater than 0 for any other $z \in \{0,1\}^n$; thus H_{I} is $H_{3\mathrm{SAT}}(0)$ with

$$h(z) := \frac{1}{2} \sum_{i=1}^{n} d_i \left(1 - (-1)^{z_i}\right) = \frac{1}{2} \sum_{i : z_i = 1} d_i.$$

Note that since the d_i are easily computable given the Boolean formula f, we can construct a small quantum network that computes $h(z)$. From f we can also construct a network computing $v(z)$, the number of clauses of f violated by assignment z. Thus the time-evolution induced by $H_{3\mathrm{SAT}}(t)$ is efficiently simulated (with respect to T) by the techniques of section 2.4.3. Since we are ultimately interested in solving practical

instances of 3SAT, it is this discrete-time simulation of the continuous-time adiabatic algorithm $\{(|u\rangle, H_{3\mathrm{SAT}}, T)\}$ that would ultimately be implemented. The continuous-time picture is really just a convenient way of looking at things. It is unclear whether one could actually construct the Hamiltonian H_{F} in the laboratory just using knowledge of the formula f in order to implement the proposed adiabatic algorithm directly.

5.4 Intuition and the Proposed Adiabatic 3SAT Algorithm

So far the only intuition we have about how the proposed 3SAT algorithm works is based in the adiabatic approximation. But what is the algorithm actually doing?

As in section 3.1, we can compute the action of $\mathbb{I} - \mathrm{i}H_{3\mathrm{SAT}}\epsilon$ in the computational basis to get some classical intuition behind the algorithm. For this we need to compute $[H_{3\mathrm{SAT}}(t)]$, the matrix of $H_{3\mathrm{SAT}}$ with respect to the computational basis $\{|i\rangle : i \in \{0,1\}^n\}$. Note that $[H_{\mathrm{F}}]$ is just a diagonal matrix. It is a little trickier to see what $[H_{\mathrm{I}}]$ looks like. From line 5.1 we can write

$$H_{\mathrm{I}} = \frac{1}{2}\left(\sum_{i=1}^{n} d_i\right)\mathbb{I} - \frac{1}{2}\sum_{i=1}^{n} d_i X^{(i)},$$

where

$$[X^{(i)}] \equiv [\mathbb{I}]_{2^{n-i}\times 2^{n-i}} \otimes \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes [\mathbb{I}]_{2^{i-1}\times 2^{i-1}}.$$

Define $M^{(i)}$ to be $X^{(i)}$ without the higher order identity operator:

$$[M^{(i)}] \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \otimes [\mathbb{I}]_{2^{i-1}\times 2^{i-1}}.$$

Thus the matrix of $M^{(i)}$ in block form is

$$[M^{(i)}] = \begin{bmatrix} [0]_{2^{i-1}\times 2^{i-1}} & [\mathbb{I}]_{2^{i-1}\times 2^{i-1}} \\ [\mathbb{I}]_{2^{i-1}\times 2^{i-1}} & [0]_{2^{i-1}\times 2^{i-1}} \end{bmatrix}_{2^i\times 2^i}.$$

Let $x = x_i x_{i-1} \cdots x_1 \in \{0,1\}^i$ and $y = y_i y_{i-1} \cdots y_1 \in \{0,1\}^i$ label a row and column of the matrix $[M^{(i)}]$ respectively. By inspection, the element in the xth row and yth column of $[M^{(i)}]$ is

$$\langle x| M^{(i)} |y\rangle = \begin{cases} 1 & \text{if } x \text{ and } y \text{ differ only in the } i\text{th bit,} \\ 0 & \text{otherwise.} \end{cases}$$

Clearly we have,

$$[X^{(i)}] = [\mathbb{I}]_{2^{n-i}\times 2^{n-i}} \otimes [M^{(i)}] = \operatorname{diag}\left(\overbrace{[M^{(i)}], [M^{(i)}], \ldots, [M^{(i)}]}^{2^{n-i} \text{ blocks}} \right)_{2^n \times 2^n}.$$

Let $x = x_n x_{n-1} \cdots x_1 \in \{0,1\}^n$ and $y = y_n y_{n-1} \cdots y_1 \in \{0,1\}^n$ label a row and column of the matrix $[X^{(i)}]$ respectively. Note that on each of the diagonal $[M^{(i)}]$-blocks of $[X^{(i)}]$, the $n-i$ higher-order (leftmost) bits of the matrix element labels x and y are constant. For an element off of the diagonal blocks, the row and column labels differ in at least one of these $n-i$ higher-order bits. Therefore

$$\langle x| X^{(i)} |y\rangle = \begin{cases} 1 & \text{if } x \text{ and } y \text{ differ only in the } i\text{th bit,} \\ 0 & \text{otherwise} \end{cases}$$

and

$$\langle x| \sum_{i=1}^{n} d_i X^{(i)} |y\rangle = \begin{cases} d_i & \text{if } x \text{ and } y \text{ differ only in the } i\text{th bit,} \\ 0 & \text{otherwise} \end{cases}$$

and finally

$$\begin{aligned} & \langle x| H_{3\mathrm{SAT}}(t) |y\rangle \\ = & \begin{cases} \frac{1}{2}\left(1 - \frac{t}{T}\right) \sum_{i=1}^{n} d_i + \frac{t}{T} v(z) & \text{if } x = y, \\ -\frac{1}{2}\left(1 - \frac{t}{T}\right) d_i & \text{if } x \text{ and } y \text{ differ only in the } i\text{th bit,} \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

With the matrix $[H_{3\mathrm{SAT}}(t)]$ calculated, we can take a look at how a general state $\sum_{z\in\{0,1\}^n} c_z(t)\,|z\rangle$ evolves under the Hamiltonian in small time increments:

$$c_z(t+\epsilon) \approx c_z(t) + \epsilon \cdot \mathrm{i}\left[\left(1-\frac{t}{T}\right)\frac{1}{2}\left(\sum_{i=1}^{n} d_i\left[c_{z^{(i)}}(t) - c_z(t)\right]\right) - \frac{t}{T}v(z)c_z(t)\right],$$

where $z^{(i)} \in \{0,1\}^n$ differs from z only in the ith bit. Thus $|z\rangle$ exchanges its probability amplitude only with its closest neighbouring basis states (basis states whose labels differ in exactly one bit-position). The imaginary constant i interferes with the intuition: iterating the above step would produce powers of i and corresponding sign changes making it hard to say whether probability amplitude is entering or leaving a basis state $|z\rangle$. Perhaps iterating this step several times would produce a pattern from which we could derive some clearer intuition. We leave this for future work. Still, we can make some more observations. As the algorithm progresses, and t/T gets larger, the $v(z)$ term plays a bigger role. States $|z\rangle$ whose labels violate many clauses exchange more and more probability amplitude as time progresses. Intuition suggests that it might be better that such states $|z\rangle$ exchange just as much probability amplitude nearer the beginning of the algorithm. Now look at the term containing $(1 - t/T)$. Near the beginning of the algorithm, the quantity $c_{z^{(i)}}(t) - c_z(t)$ is close to 0, so the term is ineffective. The term is also ineffective near the end of the algorithm where $(1 - t/T)$ is small. However, the term is certainly useful as it is the one which transfers probability amplitude across basis states. These two observations suggest that perhaps the algorithm $\{(|u\rangle\,,\, H_{3\mathrm{SAT}}(t),\, T)\}$ would be just as effective with the time-*independent* Hamiltonian $2H_{3\mathrm{SAT}}(T/2)$ in place of $H_{3\mathrm{SAT}}(t)$. Note that this was the case for the adiabatic search algorithm, where the adiabaticity did not provide any advantage over Farhi and Gutmann's search algorithm.

Hogg [H00] has independently developed a discrete-time quantum algorithm to solve 3SAT. His algorithm is a series of steps with amplitude adjustments varying linearly with the step and the number of clauses a basis state label violates, a common heuristic used in classical algorithms – and, as we saw from the intuitive analysis, the same heuristic that FGGS's adiabatic algorithm employs. His algorithm involves four free parameters

that can be chosen arbitrarily. For a particular value of these parameters, his algorithm has exactly the same form as the discrete-time simulation of FGGS's algorithm with all the d_i being equal (where we use the discrete-time simulation given in section 2.4).

This chapter motivated the definition of FGGS's proposed adiabatic 3SAT algorithm by developing it naturally from the adiabatic search algorithm. We briefly looked at the classical intuition behind the algorithm, but we could draw no solid conclusions about how efficient the algorithm is.

Chapter 6

Analysis

Farhi, Gutmann, and Goldstone [FGG00] and others [FGG+01] have run numerical simulations of the proposed adiabatic algorithm on random instances of EXACT COVER, an NP-complete problem very similar to 3SAT. Their data suggest that the adiabatic algorithm *may* be solving these instances in time $O(\text{poly}(n))$: they can fit a polynomial curve ($\text{poly}(n)$) to the data points on a graph of pure running time $T(n)$ versus input size $n \leq 20$.

The most direct approach to an analysis, as suggested by the adiabatic approximation, is to compute the two smallest eigenvalues of $H_{3\text{SAT}}(t)$ for $0 \leq t \leq T$ to see by how much they differ. For arbitrary 3SAT instances, such a closed-form analysis has not been possible.

This chapter contains some theorems about the behaviour of adiabatic algorithms of a certain general form that we define in the next section. We gain some insight into the limitations of certain adiabatic algorithms.

6.1 A Generalization of the Problem

It is convenient to cast the adiabatic algorithm in a more general notation, thus emphasizing some essential elements of its structure. Analyzing the algorithm in this general framework may produce insights into the limitations and capabilities of certain forms of Hamiltonians for adiabatic algorithms.

For convenience, define

$$|\overline{z}\rangle \equiv W|z\rangle, \qquad \text{for all } z \in \{0,1\}^n.$$

We define the *Hadamard basis* to be $\{|\overline{z}\rangle : z \in \{0,1\}^n\}$. Let $N = 2^n$.

Note that the functions $v(z)$ and $h(z)$ of the previous section each induce a partition on $\{0,1\}^n$:

$$\bigcup_{i=0}^{\max v(z)} \{z \in \{0,1\}^n : v(z) = i\} \;=\; \{0,1\}^n \;=\; \bigcup_{i=0}^{\max h(z)} \{z \in \{0,1\}^n : h(z) = i\},$$

where the above two unions are disjoint. Each nonempty set in the above unions defines a basis for an eigenspace of either H_{F} or H_{I}. We make a straightforward generalization of this below.

Let $p(n)$ and $q(n)$ be two nonnegative integer functions of n. Let $\mathcal{P} = \{P_j\}_{j=0,1,\ldots,p(n)}$ and $\mathcal{Q} = \{Q_k\}_{k=0,1,\ldots,q(n)}$ be two partitions of $\{0,1\}^n$, that is, the following two unions are disjoint:

$$\bigcup_{k=0}^{q(n)} Q_k = \{0,1\}^n = \bigcup_{j=0}^{p(n)} P_j.$$

It will soon be clear that each subset P_j defines a basis for some eigenspace of some Hamiltonian, and likewise for each Q_k. With this in mind, define the projectors onto these eigenspaces:

$$\hat{Q}_k \equiv \sum_{z \in Q_k} |\overline{z}\rangle\langle\overline{z}|, \qquad \hat{P}_j \equiv \sum_{z \in P_j} |z\rangle\langle z|.$$

Now define the two Hamiltonians H_0 and H_1 in terms of their spectral decompositions,

$$H_0 \equiv \sum_{k=0}^{q(n)} F_k \hat{Q}_k, \qquad H_1 \equiv \sum_{j=0}^{p(n)} E_j \hat{P}_j,$$

where the two sequences of energy eigenvalues $(E_j)_{j=0,1,\ldots,p(n)}$ and $(F_k)_{k=0,1,\ldots,q(n)}$ are strictly increasing sequences of real numbers. These two Hamiltonians H_0 and H_1 are, respectively, the generalized versions of the previously-defined initial and final Hamiltonians H_{I} and H_{F} which define $H(t)_{3\mathrm{SAT}}$. Assume further that $F_0 = 0 = E_0$ and that the start state $|u\rangle \equiv \left|\overline{0^n}\right\rangle = W|0^n\rangle$ is a ground state of H_0 i.e. $0^n \in Q_0$. Note that H_0, like H_{I}, is diagonal in the Hadamard basis and H_1, like H_{F}, is diagonal in the computational basis. So that we have convenient expressions for H_0 and H_1 in these bases, we will also write

$$H_0 = \sum_{z\in\{0,1\}^n} F(z)|\overline{z}\rangle\langle\overline{z}| \qquad \text{and} \qquad H_1 = \sum_{z\in\{0,1\}^n} E(z)|z\rangle\langle z|$$

for eigenvalue functions $E(z)$ and $F(z)$ which are defined by the partitions $\mathcal{P}$ and $\mathcal{Q}$ and the sequences (E_j) and (F_k).

Let $s(t)$ be a smooth, nondecreasing, real function of time t such that

$$s : [0, T] \longrightarrow [0, 1],$$

with $s(0) = 0$ and $s(T) = 1$, for some final time T. It is clear that

$$H_{\mathrm{GEN}}(t) \equiv (1 - s(t))\, H_0 + s(t) H_1, \qquad 0 \leq t \leq T,$$

is a Hamiltonian having suitable form for adiabatic quantum algorithms solving problems for which P_0 contains the solutions. Note that $H_{3\mathrm{SAT}}(t)$ is a special case of $H_{\mathrm{GEN}}(t)$. Note also that $H_{\mathrm{GEN}}(t)$ depends on many variables other than t (e.g. n, $p(n)$, $q(n)$, the partitions $\mathcal{P}$ and $\mathcal{Q}$, etc.), but for a less cumbersome notation only the dependence on time t is regularly explicated.

Let $\mathcal{A}_{\mathrm{GEN}}$ denote the algorithm $\{(|u\rangle\,,\, H_{\mathrm{GEN}}(t),\, T)\}$ for some T. We assume that we can easily construct the small quantum networks that compute $E(z)$ and $F(z)$ needed

for a discrete-time simulation of the time-evolution induced by H_0 and H_1. We also require that $E_j, F_k \in O(\text{poly}(n))$ so that the energy differences of H_0 and H_1 are small. This ensures that the complexity of the discrete-time simulation of $\mathcal{A}_{\text{GEN}}$ is polynomially related to the pure running time T.

The "ground eigenspace", $\text{span}\{|z\rangle : z \in P_0\}$, of H_1 shall be considered the solution space of a *generalized search problem*. That is, for every $w' \in P_0$, the string w' is a solution to the problem instance; the goal of the algorithm $\mathcal{A}_{\text{GEN}}$ is to rotate the initial state $|u\rangle$ into the subspace $\text{span}\{|w'\rangle : w' \in P_0\}$ so that a measurement in the computational basis renders a solution w'. As with 3SAT and BLACK-BOX SEARCH, we shall treat the parameter n as the size of the generalized search problem. Thus the qualifiers "exponential" and "polynomial" refer to quantities like 2^n and $\text{poly}(n)$ respectively.

With the goal of obtaining intuition about the power of FGGS's adiabatic algorithm, we consider some simpler adiabatic algorithms obtained from making restrictions on the general Hamiltonian $H_{\text{GEN}}(t)$.

6.2 Trying to Solve 3SAT by Just "Searching with a Better Oracle"

Consider the following problem: "What is the minimum time T needed such that the Hamiltonian

$$\begin{aligned} H_A(t) &\equiv H_{\text{GEN}}(q(n) := 1, |Q_0| := 1; t) \\ &= (1 - s(t)) \left(0|\overline{0^n}\rangle\langle\overline{0^n}| + F_1 \sum_{z \neq 0} |\overline{z}\rangle\langle\overline{z}| \right) + s(t) \sum_{j=0}^{p(n)} E_j \hat{P}_j \\ &= (1 - s(t))\, F_1 \left(\mathbb{I} - |u\rangle\langle u| \right) + s(t) \sum_{j=0}^{p(n)} E_j \hat{P}_j, \qquad 0 \leq t \leq T, \end{aligned}$$

(adiabatically) evolves the start state $|\psi_A(0)\rangle = |u\rangle$ to a final state $|\psi_A(T)\rangle$ that is mostly in the "solution subspace" $\text{span}\{|w'\rangle : w' \in P_0\}$?" We stress that the *proposed*

adiabatic algorithm $\mathcal{A}_A \equiv \{(|u\rangle\,,\, H_A(t),\, T)\}$ has not been proved correct for *any* T since we have not shown that the smallest two eigenvalues of $H_A(t)$ never meet in the interval $t \in [0, T]$. Note that $\mathcal{A}_A$ can be thought of as FGGS's proposed 3SAT algorithm but with H_{I} replaced by H'_u from the adiabatic search algorithm of section 4.4. Note that $\mathcal{A}_A$ can also be thought of as the continuous-time, adiabatic analogue of Grover's algorithm but with the oracle O_f replaced by the oracle O_v (hence the title of this section). Below, I show that $\mathcal{A}_A$ requires an exponentially large pure running time T.

Let $|\psi_A(t)\rangle$ be the state evolving under $H_A(t)$, that is, we define the initial value problem

$$\begin{cases} \frac{\partial}{\partial t}\,|\psi_A(t)\rangle & = -\mathrm{i}H_A(t)\,|\psi_A(t)\rangle\,, \quad 0 \le t \le T \\ |\psi_A(0)\rangle & = |u\rangle\,. \end{cases}$$

Let S_A be the subspace

$$S_A \equiv \mathrm{span}\{\hat{P}_j\,|u\rangle : \quad j = 0, 1, \ldots, p(n)\}.$$

Noting that $\sum_{j=0}^{p(n)} \hat{P}_j = \mathbb{I}$, we have $|u\rangle \in S_A$. Thus S_A is invariant under $H_A(t)$ since applying $H_A(t)$ to an arbitrary vector $\sum_{j=0}^{p(n)} a_j \hat{P}_j\,|u\rangle$ in S_A gives a vector in S_A:

$$\begin{aligned} & H_A \sum_{j=0}^{p(n)} a_j \hat{P}_j\,|u\rangle \\ = \quad & \left[(1-s)F_1(\mathbb{I} - |u\rangle\langle u|) + s\sum_{j=0}^{p(n)} E_j \hat{P}_j\right] \sum_{j=0}^{p(n)} a_j \hat{P}_j\,|u\rangle \\ = \quad & \sum_{j=0}^{p(n)} \left[(1-s)F_1 a_j\right] \hat{P}_j\,|u\rangle + \left[-\sum_{j=0}^{p(n)} (1-s)F_1 a_j\,\langle u|\,\hat{P}_j\,|u\rangle\right] |u\rangle + \sum_{j=0}^{p(n)} \left[sE_j a_j\right] \hat{P}_j\,|u\rangle \\ \in \quad & S_A. \end{aligned}$$

In general, suppose a subspace S of dimension d is invariant under a Hamiltonian $H(t)$ for all $0 \le t \le T$ and that $|\psi(t)\rangle$ is the state evolving under $H(t)$. Then S is invariant under $\mathbb{I} - \mathrm{i}H(t)c$ for all t and all constants c. Let $U(T, 0)$ be the total unitary time-evolution

operator induced by $H(t)$. From equations 1.2 and 1.3 we know that

$$U(t,0) = \lim_{M\to\infty} K_M(t), \tag{6.1}$$

where

$$K_M(t) \equiv \prod_{m=1}^{M} \left[\mathbb{I} - \mathrm{i}H\left((M-m)\frac{t}{M}\right)\frac{t}{M}\right].$$

Thus, if $|\psi(0)\rangle \in S$, then the sequence $(K_M(t)\,|\psi(0)\rangle)_{M=1,2,\ldots}$ is contained in S. The limit $|\psi(t)\rangle$ of this sequence exists. Because vectors in S are in one-to-one correspondence with points in $\mathbb{C}^d$, S can be viewed as a metric space relative to the norm metric (induced by the inner product on S). Since $\mathbb{C}^d$ is closed with respect to this metric, the limit of the sequence is in S. Thus $|\psi(t)\rangle \in S$ for all t. Thus the state $|\psi_A(t)\rangle$ remains in S_A for all t. Now proceed with an analysis in the orthonormal basis $B_A \equiv \{|u_j\rangle : \ j = 0, 1, \ldots, p(n)\}$ for S_A, where

$$|u_j\rangle \equiv \sqrt{\frac{N}{|P_j|}}\hat{P}_j\,|u\rangle\,, \qquad j = 0, 1, \ldots, p(n).$$

Noting that $|u\rangle = \frac{1}{\sqrt{N}}\sum_{j=0}^{p(n)}\sqrt{|P_j|}\,|u_j\rangle$, we can easily compute the outer product form of $H_A(t)$ *restricted to* S_A,

$$\begin{aligned}
& H_A(t)|_{S_A} \\
= \ & \left[(1-s(t))\,F_1\,(\mathbb{I} - |u\rangle\langle u|) + s(t)\sum_{j=0}^{p(n)} E_j\hat{P}_j\right]_{S_A} \\
= \ & (1-s(t))\,F_1\left(\mathbb{I} - \frac{1}{N}\sum_{j=0}^{p(n)}\sum_{l=0}^{p(n)}\sqrt{|P_j||P_l|}|u_l\rangle\langle u_j|\right) + s(t)\sum_{j=0}^{p(n)} E_j|u_j\rangle\langle u_j|. \qquad (6.2)
\end{aligned}$$

Suppose $|\psi_A(t)\rangle$ is expressed in the basis B_A, as $|\psi_A(t)\rangle = \sum_{j=0}^{p(n)}\alpha_j(t)\,|u_j\rangle$. Clearly, the goal of the unitary evolution is to make $|\,\langle u_0|\psi_A(T)\rangle\,|^2 = |\alpha_0(T)|^2$ of constant order. The

Schrödinger equation implies

$$\begin{aligned}\frac{\partial}{\partial t}|\psi_A\rangle &= -iH_A|\psi_A\rangle \\ &= -i\left(\sum_{l,j=0}^{p(n)}\langle u_l|H_A|u_j\rangle|u_l\rangle\langle u_j|\right)\sum_{k=0}^{p(n)}\alpha_k|u_k\rangle \\ &= -i\sum_{k=0}^{p(n)}\left(\sum_{j=0}^{p(n)}\langle u_k|H_A|u_j\rangle\alpha_j\right)|u_k\rangle,\end{aligned}$$

where the time-dependence notation has been dropped. A straightforward calculation gives the time derivative of $|\alpha_0(t)|^2$ as

$$\begin{aligned}&\frac{\partial}{\partial t}|\alpha_0(t)|^2 \\ =\ & \langle u_0|\frac{\partial}{\partial t}|\psi_A\rangle\langle u_0|\psi_A\rangle^* + \langle u_0|\psi_A\rangle\left[\left(\langle u_0|\frac{\partial}{\partial t}|\psi_A\rangle\right)^*\right] \\ =\ & 2\mathrm{Re}(\langle u_0|\frac{\partial}{\partial t}|\psi_A\rangle\cdot\langle u_0|\psi_A\rangle^*) \\ =\ & 2\mathrm{Re}\left[\langle u_0|\left(-i\sum_{k=0}^{p(n)}\left(\sum_{j=0}^{p(n)}\langle u_k|H_A|u_j\rangle\alpha_j\right)|u_k\rangle\right)\cdot\left(\langle u_0|\sum_{k=0}^{p(n)}\alpha_k|u_k\rangle\right)^*\right] \\ =\ & 2\mathrm{Re}\left[-i\sum_{j=0}^{p(n)}\langle u_0|H_A|u_j\rangle\alpha_j\cdot\alpha_0^*\right] \\ =\ & 2\mathrm{Im}\left[\sum_{j=0}^{p(n)}\langle u_0|H_A|u_j\rangle\alpha_j\alpha_0^*\right] \\ =\ & 2\sum_{j=0}^{p(n)}\langle u_0|H_A|u_j\rangle\,\mathrm{Im}\left(\alpha_0\alpha_j^*\right) \qquad [\text{since } \langle u_0|H_A|u_j\rangle\in\mathbb{R}] \\ =\ & 2\sum_{j=1}^{p(n)}\langle u_0|H_A|u_j\rangle\,\mathrm{Im}\left(\alpha_0\alpha_j^*\right) \qquad [\text{since } \alpha\alpha^*\in\mathbb{R}, \text{ for } \alpha\in\mathbb{C}] \\ =\ & -2(1-s(t))F_1\frac{\sqrt{|P_0|}}{N}\sum_{j=1}^{p(n)}\sqrt{|P_j|}\mathrm{Im}\left(\alpha_1\alpha_j^*\right) \qquad [\text{by expression 6.2}].\end{aligned}$$

Viewing $\sum_{j=1}^{p(n)}\sqrt{|P_j|}\mathrm{Im}\left(\alpha_1\alpha_j^*\right)$ as the dot-product of the real vectors $v_1 = (\sqrt{|P_1|},\sqrt{|P_2|},\ldots,\sqrt{|P_{p(n)}|})$ and $v_2 = (\mathrm{Im}(\alpha_0\alpha_1^*),\mathrm{Im}(\alpha_0\alpha_2^*),\ldots,\mathrm{Im}(\alpha_0\alpha_{p(n)}^*))$ and not-

ing that $|v_1| \leq \sqrt{N}$ and $|v_2| \leq |\alpha_0| \leq 1$, we can employ the Cauchy-Schwarz inequality $|v_1 \bullet v_2| \leq \|v_1\| \|v_2\|$ to get a bound on the absolute value of the above derivative:

$$\left| \frac{\partial}{\partial t} |\alpha_0(t)|^2 \right| \leq 2F_1 \sqrt{\frac{|P_0|}{N}}, \quad 0 \leq t \leq T.$$

Noting that $\alpha_0(0) = \frac{1}{\sqrt{N}}$ and using

$$|\alpha_0(T)|^2 - |\alpha_0(0)|^2 = \int_0^T \frac{\partial |\alpha_0(t)|^2}{\partial t} dt \leq \int_0^T \left| \frac{\partial |\alpha_0(t)|^2}{\partial t} \right| dt$$

gives

$$|\alpha_0(T)|^2 \leq 2F_1 \sqrt{\frac{|P_0|}{N}} T + \frac{1}{N}.$$

If T is required to be such that $|\alpha_0(T)|^2 \geq c$ for some constant c, then

$$T \geq \frac{c\sqrt{2^n} - 1/\sqrt{2^n}}{2F_1 \sqrt{|P_0|}}.$$

Thus if $\frac{N}{|P_0|} \in \omega(\text{poly}(n))$, then the algorithm $\mathcal{A}_A$ requires an exponentially large pure running time T and hence the best known discrete-time simulation of $\mathcal{A}_A$ requires exponentially many resources to implement. Suppose rather that $\frac{N}{|P_0|} \notin \omega(\text{poly}(n))$. Then $\frac{N}{|P_0|} \leq n^c$ for some constant c, and so $\frac{|P_0|}{N} \geq \frac{1}{n^c}$. In this case, there are enough solutions $w' \in P_0$ so that a good classical randomized algorithm for the problem is to iterate the following:

- pick an element $z \in \{0,1\}^n$ at random;
- compute $E(z)$ and check whether $E(z) = 0 \Leftrightarrow z \in P_0$.

It is clear that the expected number of iterations is $O(\text{poly}(n))$. Thus unless the problem instance is classically tractable the adiabatic algorithm $\mathcal{A}_A$ requires impractically many resources to implement. Note that if the energy eigenvalue F_1 were very large, then the bound on T shrinks, suggesting that using Hamiltonians with large $(\omega(\text{poly}(n)))$ energy differences might speed up the algorithm $\mathcal{A}_A$. Recall that using such Hamiltonians would result in an inefficient discrete-time simulation of $\mathcal{A}_A$.

Intuitively, the above result says that any adiabatic 3SAT algorithm whose final Hamiltonian encodes the function $v(z)$ must have a "more complex" driving Hamiltonian than H'_u of the adiabatic search algorithm – if it hopes to use only polynomially many resources (we define "more complex" in the next section).

The result also has an interpretation in the context of Farhi and Gutmann's search algorithm from section 3.2. First note that all of the results proved in section 3.2 still hold for H_u and H_w replaced with H'_u and H'_w respectively. Suppose that $f(z)$ is the binary function corresponding to the 3-CNF formula f that defines a 3SAT instance, and that $v(z)$ is the number of clauses of f violated by assignment z. We can use Farhi and Gutmann's search algorithm to solve the 3SAT instance in time $\Theta(\sqrt{N})$. If we replace $H'_w = \sum_z f(z)|z\rangle\langle z|$ with $\sum_z v(z)|z\rangle\langle z|$ in the search algorithm, can we solve the 3SAT instance in time $O(\text{poly}(n))$? Our result in this section implies that we cannot. To see this, note that $H'_u + \sum_z v(z)|z\rangle\langle z|$ has the same form as $2H_A(\frac{1}{2})$ and we can replace $H_A(s)$ with $2H_A(\frac{1}{2})$ in the above analysis without loss.

6.3 Allowing a More General Driving Hamiltonian

What result can we get using the above proof technique if we remove the restrictions $|Q_0| = 1$ and $q(n) = 1$? Let $|\psi_{\text{GEN}}(t)\rangle$ be the state evolving under $H_{\text{GEN}}(t)$, that is, we define the initial value problem

$$\begin{cases} \frac{\partial}{\partial t}|\psi_{\text{GEN}}(t)\rangle & = -\mathrm{i}H_{\text{GEN}}(t)\,|\psi_{\text{GEN}}(t)\rangle\,, \quad 0 \le t \le T \\ |\psi_{\text{GEN}}(0)\rangle & = |u\rangle\,. \end{cases}$$

In this case, we proceed with an analysis in the computational basis $\{|z\rangle : z \in \{0,1\}^n\}$, since it is not clear that the state $|\psi_{\text{GEN}}(t)\rangle$ remains in a subspace of dimension smaller than 2^n. Let K be the index of the largest set Q_k:

$$|Q_K| \ge |Q_k| \quad \text{for all } k = 0, 1, \ldots, q(n).$$

As before, we rewrite the Hamiltonian like this:

$$H_{\mathrm{GEN}}(t) = (1-s(t))\left(F_K\mathbb{I} - \sum_{k=0}^{q(n)}(F_K - F_k)\,\hat{Q}_k\right) + s(t)\sum_z E(z)|z\rangle\langle z|.$$

We can easily derive an expression for H_{GEN} in outer-product form: noting that

$$|\overline{z}\rangle \equiv W\,|z\rangle = \frac{1}{\sqrt{N}}\sum_x\sum_y(-1)^{x\bullet y}|x\rangle\langle y|\,|z\rangle = \frac{1}{\sqrt{N}}\sum_x(-1)^{x\bullet z}\,|x\rangle\,,$$

we have, dropping the time-dependence notation,

$$\begin{aligned} & H_{\mathrm{GEN}} \\ = \;& (1-s)\left(F_K\mathbb{I} - \frac{1}{N}\sum_x\sum_y\left(\sum_{k=0}^{q(n)}(F_K-F_k)\sum_{z\in Q_k}(-1)^{z\bullet(x\oplus y)}\right)|x\rangle\langle y|\right) \\ & +s\sum_z E(z)|z\rangle\langle z|. \end{aligned}$$

Suppose $|\psi_{\mathrm{GEN}}(t)\rangle$ is expressed in the computational basis as $|\psi_{\mathrm{GEN}}(t)\rangle = \sum_{z=0}^{N-1}\gamma_z(t)\,|z\rangle$. Clearly, the goal of the unitary evolution is to make $\sum_{w'\in P_0}|\,\langle w'|\psi_{\mathrm{GEN}}(T)\rangle\,|^2 = \sum_{w'\in P_0}|\gamma_{w'}(T)|^2$ of constant order so that we have a good probability of discovering one of the solutions w' by performing a measurement in the computational basis on the final state $|\psi_{\mathrm{GEN}}(T)\rangle$. The Schrödinger equation implies

$$\frac{\partial}{\partial t}\,|\psi_{\mathrm{GEN}}\rangle \;=\; -i\sum_x\left(\sum_y\langle x|\,H_{\mathrm{GEN}}\,|y\rangle\,\gamma_y\right)|x\rangle\,.$$

For each solution $w'\in P_0$,

$$\begin{aligned} & \frac{\partial}{\partial t}|\gamma_{w'}|^2 \\ = \;& 2\mathrm{Re}(\langle w'|\,\frac{\partial}{\partial t}\,|\psi_{\mathrm{GEN}}\rangle\cdot\langle w'|\psi_{\mathrm{GEN}}\rangle^*) \\ = \;& 2\sum_y\langle w'|\,H_{\mathrm{GEN}}\,|y\rangle\,\mathrm{Im}\left(\gamma_{w'}\gamma_y^*\right) \\ = \;& -2(1-s)\frac{1}{N}\sum_{y\neq w'}\sum_{k=0}^{q(n)}(F_K-F_k)\left(\sum_{z\in Q_k}(-1)^{z\bullet(w'\oplus y)}\right)\mathrm{Im}\left(\gamma_{w'}\gamma_y^*\right) \\ = \;& -2(1-s)\frac{1}{N}\sum_y\sum_{k=0}^{q(n)}(F_K-F_k)\left(\sum_{z\in Q_k}(-1)^{z\bullet(w'\oplus y)}\right)\mathrm{Im}\left(\gamma_{w'}\gamma_y^*\right), \end{aligned}$$

where the last two lines hold because $\mathrm{Im}(\gamma_{w'}\gamma^*_{w'}) = 0$. If the strings $z \in Q_k$ were somehow "random enough", we would expect the sums $\left(\sum_{z\in Q_k}(-1)^{z\bullet(w'\oplus y)}\right)$ to be close to 0. This would suggest that the derivative is small for an average problem instance. Without going into statistics about typical problem instances, we can only bound the size of these sums by $|Q_k|$. Using this bound for every $w' \in P_0$ and using $\left|\sum_y \mathrm{Im}(\alpha_{w'}\alpha^*_y)\right| \leq \sqrt{N}$ we get

$$\left|\sum_{w'\in P_0}\frac{\partial}{\partial t}|\gamma_{w'}|^2\right| \leq 2\frac{|P_0|}{\sqrt{N}}\cdot\max_k F_k\cdot\dim\left(\mathrm{span}\{|\overline{z}\rangle : z\in Q_K\}^\perp\right).$$

Integrating from $t = 0$ to $t = T$ as before, in order to lower-bound by a constant c the total probability of finding a solution, we require

$$T \geq \frac{c\sqrt{N} - |P_0|/\sqrt{N}}{2|P_0|\cdot\max_k F_k\cdot\dim\left(\mathrm{span}\{|\overline{z}\rangle : z\in Q_K\}^\perp\right)}.$$

Assume that the number of solutions $|P_0|$ is small. Thus if the dimension of the space orthogonal to the largest eigenspace of H_0 is small, then the algorithm $\{(|u\rangle, H_{\mathrm{GEN}}(t), T)\}$ requires exponential time T. In other words, the driving Hamiltonian must have at least two large, mutually-orthogonal subspaces in order for the algorithm to work quickly. In general, and in the case of H_{SAT}, the driving Hamiltonian indeed has two large mutually-orthogonal subspaces.

6.4 Searching with a More Complex Driving Hamiltonian

Consider the following problem: "What is the minimum time T needed such that the Hamiltonian

$$\begin{aligned} H_B(t) &\equiv H_{\mathrm{GEN}}\left(p(n) := 1, |P_0| := 1; t\right) \\ &= (1-s(t))\sum_z F(z)|\overline{z}\rangle\langle\overline{z}| + s(t)E_1\left(\mathbb{I} - |w\rangle\langle w|\right), \qquad 0 \leq t \leq T \end{aligned}$$

(adiabatically) evolves the start state $|\psi(0)\rangle = |u\rangle$ to a final state $|\psi(T)\rangle$ that is close to $|w\rangle$?" Thus we investigate the power of the driving Hamiltonian H_0 when H_1 is restricted

to being H'_w from our adiabatic search algorithm. The result we get is analogous to the result of section 6.2. As a corollary to the result, we reproduce the proof of optimality of Farhi and Gutmann's search algorithm in the present context.

We perform the analysis in the Hadamard basis $\{|\overline{z}\rangle : z \in \{0,1\}^n\}$. Noting that

$$|w\rangle\langle w| = \frac{1}{N}\sum_x\sum_y (-1)^{w\bullet(x\oplus y)}|\overline{x}\rangle\langle\overline{y}|,$$

we have

$$H_B = (1-s)\sum_z F(z)|\overline{z}\rangle\langle\overline{z}| + sE_1\left(\mathbb{I} - \frac{1}{N}\sum_x\sum_y (-1)^{w\bullet(x\oplus y)}|\overline{x}\rangle\langle\overline{y}|\right).$$

Letting $|\psi(t)\rangle = \sum_z \beta_z(t)\,|\overline{z}\rangle$, the Schrödinger equation implies that

$$\frac{\partial}{\partial t}|\psi\rangle \;\;=\;\; -iH_B\,|\psi\rangle = -\mathrm{i}\sum_x\sum_y \langle\overline{x}|\,H_B\,|\overline{y}\rangle\,\beta_y\,|\overline{x}\rangle\,.$$

Instead of bounding the rate at which probability amplitude flows into the solution state $|w\rangle$, we bound the rate at which it flows out of the start state $|u\rangle$. Noting that $\langle u|\psi(t)\rangle =$

$\beta_0(t)$,

$$\begin{aligned}
\frac{\partial}{\partial t}|\beta_0(t)|^2 &= \langle u|\frac{\partial}{\partial t}|\psi\rangle\langle u|\psi\rangle^* + \langle u|\psi\rangle\left[\left(\langle u|\frac{\partial}{\partial t}|\psi\rangle\right)^*\right] = 2\mathrm{Re}(\langle u|\frac{\partial}{\partial t}|\psi\rangle\cdot\langle u|\psi\rangle^*) \\
&= 2\mathrm{Re}\left[\langle u|\left(-\mathrm{i}\sum_x\sum_y\langle\overline{x}|H_B|\overline{y}\rangle\beta_y|\overline{x}\rangle\right)\cdot\left(\langle u|\sum_z\beta_z|\overline{z}\rangle\right)^*\right] \\
&= 2\mathrm{Re}\left[\left(-\mathrm{i}\sum_y\langle\overline{0^n}|H_B|\overline{y}\rangle\beta_y\right)\cdot\beta_0^*\right] \quad [\text{since } |u\rangle = |\overline{0^n}\rangle] \\
&= 2\mathrm{Im}\left[\sum_y\langle\overline{0^n}|H_B|\overline{y}\rangle\beta_y\beta_0^*\right] \\
&= 2\sum_y\langle\overline{0^n}|H_B|\overline{y}\rangle\,\mathrm{Im}(\beta_y\beta_0^*) \quad [\text{since } \langle\overline{0^n}|H_B|\overline{y}\rangle\in\mathbb{R}] \\
&= 2\sum_{y\neq 0}\langle\overline{0^n}|H_B|\overline{y}\rangle\,\mathrm{Im}(\beta_y\beta_0^*) \quad [\text{since } \mathrm{Im}(\beta_0\beta_0^*)=0] \\
&= 2\sum_{y\neq 0}\left(-\frac{sE_1}{N}(-1)^{w\bullet(0^n\oplus y)}\right)\mathrm{Im}(\beta_y\beta_0^*) \\
&= -\frac{2sE_1}{N}\sum_{y\neq 0}(-1)^{w\bullet y}\mathrm{Im}(\beta_y\beta_0^*) \\
&= -\frac{2sE_1}{N}\sum_{y}(-1)^{w\bullet y}\mathrm{Im}(\beta_y\beta_0^*) \quad [\text{since } \mathrm{Im}(\beta_0\beta_0^*)=0].
\end{aligned}$$

Using the Cauchy-Schwarz inequality we get the bound

$$\left|\frac{\partial}{\partial t}|\beta_0(t)|^2\right| \leq \frac{2E_1}{\sqrt{N}}, \quad 0\leq t\leq T.$$

Noting that $\beta_0(0)=1$ and using

$$|\beta_0(T)|^2 - |\beta_0(0)|^2 = \int_0^T\frac{\partial|\beta_0(t)|^2}{\partial t}dt \geq -\int_0^T\left|\frac{\partial|\beta_0(t)|^2}{\partial t}\right|dt$$

gives

$$|\beta_0(T)|^2 \geq 1 - \frac{2E_1}{\sqrt{N}}T.$$

In section 8.4 of the Appendix, we prove that the following inequality holds:

$$|\langle w|\psi_B(T)\rangle|^2 + |\langle u|\psi_B(T)\rangle|^2 \leq 1 + |\langle u|w\rangle|. \tag{6.3}$$

This implies

$$|\langle w|\psi_B(T)\rangle|^2 \leq \frac{2E_1}{\sqrt{N}}T + \frac{1}{\sqrt{N}},$$

which implies that the algorithm $\{(|u\rangle, H_B(t), T)\}$ requires exponentially large pure running time T in order to compute w with sufficiently high probability. In our generalized adiabatic setting, the result effectively bounds the power of any driving Hamiltonian that is diagonal in the Hadamard basis, in the presence of a final Hamiltonian that encodes the binary function $f(z)$ that defines the generalized search problem having a unique solution $w \in \{0,1\}^n$, i.e. $f(z) = 0$ if and only if $z = w$.

As in the last paragraph of section 6.2 we can interpret our result in the context of Farhi and Gutmann's search algorithm. Note that the optimality result proved in section 3.2 still holds if we replace H_u and H_w by H'_u and H'_w respectively. Thus we have a result saying that no (unrestricted) driving Hamiltonian can work faster than H'_u if it must do so on *most* problem instances. How is this result affected if we restrict H_{D} to be diagonal in the Hadamard basis? To answer this question we need to examine the Hamiltonian $H_0 + H'_w$. But note that $H_0 + H'_w$ is of the form $2H_B(\frac{1}{2})$ and we can replace $H_B(s)$ with $2H_B(\frac{1}{2})$ in the above analysis without loss. Thus we get the stronger result that no (restricted) driving Hamiltonian can outperform H_u on even one problem instance.

In sections 6.2 and 6.4, we saw two proposed adiabatic algorithms that provably require large pure running times. We also gave interpretations of these results in the context of Farhi and Gutmann's search algorithm. In section 6.3, we showed that if an adiabatic algorithm is to require a pure running time T in $O(\mathrm{poly}(n))$ then its driving Hamiltonian must have at least two large, mutually-orthogonal eigenspaces. Note that combining techniques of section 6.3 and section 6.4, we can get the result (analogous to the result in section 6.3) which says that if an adiabatic algorithm is to require only $T \in O(\mathrm{poly}(n))$, then its *final* Hamiltonian must have at least two large, mutually-

orthogonal eigenspaces. Combining the two results, we have that if an adiabatic algorithm is to require only $T \in O(\text{poly}(n))$, then *both* its driving and final Hamiltonians must have at least two large, mutually-orthogonal eigenspaces.

Chapter 7

Conclusion

In this work we have presented FGGS's proposed adiabatic 3SAT algorithm [FGGS00] as a natural extension of Farhi and Gutmann's search algorithm [FG98]. Although we were unable to determine whether the 3SAT algorithm is efficient in general, we did prove some relevant theorems about the behaviour of adiabatic algorithms by using a simple derivative-bounding techinique. We close with some known lower bounds and possible future areas of research.

7.1 Known Lower Bounds

The derivative calculation in section 6.3 certainly allows for the possibility that there exists functions $E(z)$ and $F(z)$ such that the adiabatic algorithm $\{(|u\rangle, H_{\mathrm{GEN}}(t), T)\}$ solves the generalized search problem in O(poly(n)) time. In fact, it has been shown [vDMV01] that when $E(z) = F(z) = |z| \equiv \sum_{i=1}^{n} z_i$ for all $z_1 z_2 \cdots z_n \in \{0,1\}^n$, the distance between the smallest two eigenvalues of $H_{\mathrm{GEN}}(t)$ reaches a minimum of $\frac{1}{\sqrt{2}}$ at $s(t) = \frac{1}{2}$. Thus, in that case, the algorithm succeeds with high probability if $T \in \Theta(1)$. When $E(z) = |z| = F(z)$ for all z, the total Hamiltonian can be written as a sum of n one-qubit Hamiltonians. This special structure makes the minimum distance between the

smallest two eigenvalues of constant order. If merely $E(z) = F(z)$ for all z, the minimum distance between the smallest two eigenvalues can be exponentially small. An example of this is the adiabatic search algorithm with solution $w = 0^n$; the result in section 6.4 implies that regardless of the solution w the algorithm $\{(|u\rangle, H_{\mathrm{AS}'}(t), T)\}$ requires $T \in \Omega(\sqrt{N})$, hence by the adiabatic approximation the minimum distance between the two smallest eigenvalues must be $O(\frac{1}{N^{1/4}})$.

The restriction $\mathcal{P} = \mathcal{Q}$ makes the generalized search problem classically tractable: the solution is always 0^n. With regard to 3SAT, $\mathcal{P} = \mathcal{Q}$ holds only if the given Boolean formula is satisfied when $z_i := 0$ for all $i = 1, 2, \ldots, n$. It is not so surprising that a classically tractable problem is quickly solvable by the adiabatic approach. However, it was also shown [vDMV01] that there exists a classically tractable generalized search problem with $p(n) = n = q(n)$ where the minimum distance between the smallest two eigenvalues of $H_{\mathrm{GEN}}(t)$ is exponentially small. Moreover, using a similar technique, it was shown [vDV02] that there exists a family of 3SAT instances such that the minimum distance between the smallest two eigenvalues of $H_{\mathrm{3SAT}}(t)$ is exponentially small. This particular family of 3SAT instances is classically tractable.

7.2 Future Directions

The following problems may suggest interesting avenues of investigation.

- If it exists, find a classically intractable family of 3SAT instances for which FGGS's proposed adiabatic 3SAT algorithm requires exponentially large pure running time.
- Do the results of chapter 6 hold for operators other than the Hadamard operator W?
- Derive the adiabatic approximation theory in the special case $H(\frac{t}{T}) = (1 - \frac{t}{T})WD_1W + \frac{t}{T}D_2$ with D_1 and D_2 diagonal. Does this give any further insight into the behaviour of FGGS's proposed 3SAT algorithm?

- Our general adiabatic scheme uses Hamiltonians of the form $\tilde{H}(s) = (1-s)H_0 + sH_1$. Consider other paths from H_0 to H_1 that may be easier to analyze.

Chapter 8

Appendix

8.1 Asymptotic Notation

Let $n \in \{0, 1, \ldots\}$, and let $f(n)$ and $g(n)$ be asymptotically nonnegative, real functions of n i.e. there exists $n_0 \geq 0$ such that $n > n_0$ implies that $f(n) \geq 0$, and similarly for $g(n)$. We have the following definitions:

$\Theta(g(n)) \equiv \{f(n) :$ there exist positive constants c_1, c_2, and n_0 such that
$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0\}$

$O(g(n)) \equiv \{f(n) :$ there exist positive constants c and n_0 such that
$0 \leq f(n) \leq cg(n)$ for all $n \geq n_0\}$

$o(g(n)) \equiv \{f(n) :$ for any positive constant $c > 0$, there exists a constant $n_0 > 0$
such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0\}$

$\Omega(g(n)) \equiv \{f(n) :$ there exist positive constants c and n_0 such that
$0 \leq cg(n) \leq f(n)$ for all $n \geq n_0\}$

$\omega(g(n)) \equiv \{f(n) :$ for any positive constant $c > 0$, there exists a constant $n_0 > 0$ such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0\}$.

We also use the following abbreviation:

$$\text{poly}(n) \equiv n^k \text{ for some constant } k.$$

We use jargon like "...the algorithm uses $O(\text{poly}(n))$ time..." to mean that the time is a function of n, say, $T(n)$, such that $T(n) \in O(\text{poly}(n))$. When using $\epsilon > 0$ which is intended to be considered in the limit as ϵ approaches 0, we have the following definitions (for example):

$o(g(\epsilon)) \equiv \{f(\epsilon) :$ for any positive constant $c > 0$, there exists a constant $\epsilon_0 > 0$ such that $0 \leq f(\epsilon) < cg(\epsilon)$ for all $\epsilon \leq \epsilon_0\}$,

$O\left(g\left(\frac{1}{\epsilon}\right)\right) \equiv \{f\left(\frac{1}{\epsilon}\right) :$ there exist positive constants c and ϵ_0 such that $0 \leq f\left(\frac{1}{\epsilon}\right) \leq cg\left(\frac{1}{\epsilon}\right)$ for all $\epsilon \leq \epsilon_0\}$.

Combining and extending the above definitions, we use the following definitions when g_1 and g_2 are asymptotically nonnegative functions of n:

$O(\text{poly}(g_1(n), g_2(n), \frac{1}{\epsilon})) \equiv \{f(n) :$ there exist positive constants c, k, k', k'', ϵ_0, and n_0 such that $0 \leq f(n) \leq c\frac{g_1^k(n)g_2^{k'}(n)}{\epsilon^{k''}}$ for all $n \geq n_0$ and all $\epsilon < \epsilon_0$ $\}$.

8.2 Proof of Equivalence of 22, 23, and 24

The following are equivalent:

- $\prod_{j=1}^{J} e^{-\mathrm{i}H_{\tau(j)}t} = \prod_{j=1}^{J} e^{-\mathrm{i}H_{\tau'(j)}t}$ for all $t \geq 0$ and any permutations τ and τ' of $\{1, 2, \ldots, J\}$ (22)

- H_i and H_j commute for all i and j (23)
- $e^{-\mathrm{i}(H_1+H_2+\cdots+H_J)t} = e^{-\mathrm{i}H_1 t}e^{-\mathrm{i}H_2 t}\cdots e^{-\mathrm{i}H_J t}$ for all $t \geq 0$. (24)

That 23 implies 24 is stated in exercise 4.47 in [NC00] and is straightforward to show.

That 24 implies 22 is clear by commutativity of operator addition.

That 22 implies 23 can be shown as follows. For J=2, we have that

$$\mathbb{I} + (-\mathrm{i}t)(H_1+H_2) + (-\mathrm{i}t)^2(H_1^2+H_2^2+H_1H_2) + \sum_{n=3}^{\infty} B(n)t^n$$
$$= \mathbb{I} + (-\mathrm{i}t)(H_1+H_2) + (-\mathrm{i}t)^2(H_1^2+H_2^2+H_2H_1) + \sum_{n=3}^{\infty} C(n)t^n$$

for all t, for operators $B(n)$ and $C(n)$. Differentiating twice with respect to t gives

$$H_1H_2 - H_2H_1 = A(t)$$

for all t where $A(t) \equiv \frac{1}{2}\sum_{n=3}^{\infty}[B(n)-C(n)]n(n-1)t^{n-2}$. Since the left side of the above equation is independent of t, we must have $A(t) = c$ for all t where c is independent of t. Since $A(0) = 0$, we have $A(t) = 0$ for all t. Thus H_1 and H_2 commute. Suppose that $H_1, H_2, \ldots, H_{J-1}$ commute. If 22 holds, then we can show that H_J commutes with H_k for $k = 1, 2, \ldots, J-1$ as follows. In one instance of 22, we certainly have that

$$\left(\prod_{j=1;j\neq k}^{J-1} e^{-\mathrm{i}H_j t}\right) e^{-\mathrm{i}H_k t}e^{-\mathrm{i}H_J t} = \left(\prod_{j=1;j\neq k}^{J-1} e^{-\mathrm{i}H_j t}\right) e^{-\mathrm{i}H_J t}e^{-\mathrm{i}H_k t}$$

for all t. Letting $D_k(t)$ denote the operator in parentheses in the above equation, we can apply $D_k^{-1}(t)$ on both sides to get

$$e^{-\mathrm{i}H_k t}e^{-\mathrm{i}H_J t} = e^{-\mathrm{i}H_J t}e^{-\mathrm{i}H_k t}$$

for all t. It follows that H_k and H_J commute by the method used above to show that H_1 and H_2 commute. Thus H_k and H_J commute for all $k = 1, 2, \ldots, J-1$. Thus, by mathematical induction, 22 implies 23.

8.3 How to Check Polynomially Many Elements Against the Search Oracle by Controlling the Driving Hamiltonian

We give the discrete-time quantum algorithm first, and then explain how it could be implemented by merely controlling the driving Hamiltonian $H_D(t)$ in the presence of the persistent Hamiltonian $H_w = E|w\rangle\langle w|$.

Define the unitary operator U_i for all $i \in \{0,1\}^n$ by its action on the computational basis states:

$$\begin{aligned}
\text{for } i \neq 0^n: \quad U_i : |0^n\rangle &\mapsto |0^n\rangle + |i\rangle \\
|i\rangle &\mapsto |0^n\rangle - |i\rangle \\
|z\rangle &\mapsto |z\rangle, \quad \text{for all } z \neq 0^n, i, \\
\text{for } i = 0: \quad U_0 : |z\rangle &\mapsto |z\rangle, \quad \text{for all } z \in \{0,1\}^n.
\end{aligned}$$

where we have ignored normalization factors. Note that U_i can be decomposed as

$$U_i = S_1 S_2 \cdots S_{|i|} \cdot V_{|i|} \cdot (S_1 S_2 \cdots S_{|i|})^\dagger \tag{8.1}$$

where $|i|$ denotes the Hamming weight of i, the S_j are SWAP gates, and V_k acts on n qubits like this:

$$\begin{aligned}
V_k : |0^n\rangle &\mapsto |0^n\rangle + \left|1^k 0^{n-k}\right\rangle \\
|1^n\rangle &\mapsto |0^n\rangle - \left|1^k 0^{n-k}\right\rangle \\
|z\rangle &\mapsto |z\rangle, \quad \text{for all } z \in \{0,1\}^n \text{ such that } z \neq 0^k y, 1^k y \text{ for any } (n-k)\text{-bit string } y.
\end{aligned}$$

The operator V_k can be carried out with the following steps [Z02]:

- controlled-NOT on each of the leftmost $k-1$ qubits conditioned on the kth qubit being 0 and the rightmost $n-k$ qubits being 0

- controlled-Hadamard gate on the kth qubit condtioned on the leftmost $k-1$ qubits being 1 and the rightmost $n-k$ qubits being 0
- repeat first step (it is its own inverse).

From exercises 4.29 and 4.30 in [NC00], it follows that the above steps may be performed *without* any ancillary qubits and with $O(n^2)$ one- and two-qubit gates. Since, the SWAP gates are each simple two-qubit unitary operations, U_i may be carried out by a $O(\text{poly}(n))$-sized sequence of one- and two-qubit operations without any ancillary qubits.

Let D_i be the gate which maps $|0^n\rangle$ to $|i\rangle$, and $|i\rangle$ to $|0^n\rangle$, and leaves other basis states unchanged. The operator D_i can be implemented with no ancillary qubits using the same "swapping" approach that we used to implement V_k.

Define the operator O_f as in section 3.3,

$$O_f : |z\rangle \mapsto (-1)^{f(z)} |z\rangle , \tag{8.2}$$

where f defines the search problem instance

$$f(z) = \begin{cases} 1 \text{ if } z = w \\ 0 \text{ otherwise.} \end{cases} \tag{8.3}$$

The following is the required algorithm:

- start with state $|0^n\rangle$
- pick any $i \in \{0,1\}^n$, $i \neq 0^n$, and apply $U_i^\dagger O_f U_i$
- perform a measurement in the computational basis
- if result of measurement is i (then either $w = i$ or $w = 0^n$)

  apply D_i (to reset register to $|0^n\rangle$)

  pick any $j \in \{0,1\}^n$, $j \neq 0^n$, $j \neq i$, and apply $U_j^\dagger O_f U_j$

perform a measurement in the computational basis

if result of measurement is j then RETURN $w = 0^n$, else RETURN $w = i$

- else (now we know $w \neq 0^n$)

  choose a random subset $\{g(i) : i = 1, 2, \ldots, M\}$ of $\{0,1\}^n$ with $M \in O(\text{poly}(n))$

  apply the operator $U^\dagger_{g(M)} O_f U_{g(M)} \cdot U^\dagger_{g(M-1)} O_f U_{g(M-1)} \cdots U^\dagger_{g(1)} O_f U_{g(1)}$

  perform a measurement in the computational basis

  if the result of the measurement is $z \neq 0^n$ then RETURN $w = z$, else FAIL.

It is clear that the above algorithm requires only $O(\text{poly}(n))$ resources. The first part of the algorithm, which checks to see if $w = 0^n$, is straightforward to verify. Suppose $w \neq 0^n$. If $g(i) \neq w$, then $U^\dagger_{g(i)} O_f U_{g(i)} |0^n\rangle = |0^n\rangle$. But if $g(i) = w$, then $U^\dagger_{g(i)} O_f U_{g(i)} |0^n\rangle = |w\rangle$. Also if $g(i) \neq j \neq 0^n$, then $U^\dagger_{g(i)} O_f U_{g(i)} |j\rangle = |j\rangle$, so that if we find the solution along the way, we keep it right up until the last measurement is performed. This proves the correctness of the algorithm.

It remains to argue that the above algorithm can be implemented by controlling the driving Hamiltonian $H_D(t)$. From the discussion in section 3.2, the operation O_f can be effected by setting $H_D(t) = 0$ for a period of π/E seconds. To perform U_i, we assume that we know the sequence of natural Hamiltonians (suitable for the given implementation scheme) implementing U_i. The problem is that $H_w = E|w\rangle\langle w|$ is present. Recall that we know the energy value E which is also the energy difference $\|\Delta H_w\|$. From the discussion in section 2.4.2, $\|\Delta H_w\|$ can be thought of as the rate of the rotation induced by H_w. We can effectively "drown out" this rotation by ensuring that $\|\Delta H_D(t)\| \gg E$ when $H_D(t)$ is not zero. We take the sequence of natural Hamiltonians implementing U_i and increase the energy differences sufficiently in order to perform each desired rotation in a shorter time interval such that the rotation induced by H_w in that same time interval is negligible. Note that this viewpoint clarifies the dependence on E of the lower bound 3.2: the only information we have about the solution w comes from H_w at a fixed "rate" E.

8.4 Proof of Inequality 6.3

Let $|u\rangle$, $|w\rangle$, and $|\psi\rangle$ be any three unit vectors in a complex inner-product space. Let

$$|\psi\rangle = a\,|u\rangle + b\,|w\rangle + c\,|v\rangle\,, \qquad \text{where } |v\rangle \in \text{span}\{|u\rangle\,, |v\rangle\}^{\perp},$$

and $\langle v|v\rangle = 1$. Suppose $\langle u|w\rangle = re^{\mathrm{i}\theta}$ for $0 \leq r \leq 1$ and let

$$\left|u'\right\rangle = e^{\mathrm{i}\theta}\,|u\rangle \quad \text{and} \quad \left|w'\right\rangle = e^{-\mathrm{i}\theta}\,|w\rangle$$

so that

$$0 \leq \left\langle u|w'\right\rangle = \left\langle u'|w\right\rangle = r \leq 1.$$

So we get the following inequality:

$$\begin{aligned}
1 &= \langle\psi|\psi\rangle \\
&= |a|^2 + |b|^2 + |c|^2 + 2\text{Re}(\langle u|w\rangle\, a^*b) \\
&= |a|^2 + |b|^2 + |c|^2 + 2\text{Re}(\left\langle u|w'\right\rangle a^*be^{\mathrm{i}\theta}) \\
&= |a|^2 + |b|^2 + |c|^2 + 2r\text{Re}(a^*be^{\mathrm{i}\theta}) \\
&= |a|^2 + |b|^2 + |c|^2 + 2r\text{Re}(ab^*e^{-\mathrm{i}\theta}) \\
&\geq |a|^2 + |b|^2 + 2r\text{Re}(ab^*e^{-\mathrm{i}\theta}). \qquad (8.4)
\end{aligned}$$

As well, since $|a \pm be^{\mathrm{i}\theta}|^2 \geq 0$, we have another inequality:

$$\begin{aligned}
\mp 2\text{Re}(ab^*e^{-\mathrm{i}\theta}) &\leq |a|^2 + |b|^2 \\
\Leftrightarrow |2\text{Re}(ab^*e^{-\mathrm{i}\theta})| &\leq |a|^2 + |b|^2. \qquad (8.5)
\end{aligned}$$

Thus

$$\begin{aligned}
|\langle w|\psi\rangle|^2 + |\langle u|\psi\rangle|^2 &= |\langle w'|\psi\rangle|^2 + |\langle u'|\psi\rangle|^2 \\
&= |a\langle w'|u\rangle + be^{\mathrm{i}\theta}|^2 + |ae^{-\mathrm{i}\theta} + b\langle u'|w\rangle|^2 \\
&= (ar + be^{\mathrm{i}\theta})(a^*r + b^*e^{-\mathrm{i}\theta}) + (ae^{-\mathrm{i}\theta} + br)(a^*e^{\mathrm{i}\theta} + b^*r) \\
&= (|a|^2 + |b|^2 + 2r\mathrm{Re}(ab^*e^{-\mathrm{i}\theta})) + r((|a|^2 + |b|^2)r + 2\mathrm{Re}(ab^*e^{-\mathrm{i}\theta})) \\
&\leq 1 + r((|a|^2 + |b|^2)r + 2\mathrm{Re}(ab^*e^{-\mathrm{i}\theta})) \quad \text{[by inequality 8.4]} \\
&\leq 1 + r((|a|^2 + |b|^2) + 2r\mathrm{Re}(ab^*e^{-\mathrm{i}\theta})) \quad \text{[by inequality 8.5]} \\
&\leq 1 + r \quad \text{[by inequality 8.4]} \\
&= 1 + |\langle u|w\rangle|
\end{aligned}$$

as required.

Bibliography

[BBBV97] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. "Strengths and Weaknesses of Quantum Computing", *SIAM J. Comput.*, 26(5):1510-1523, 1997.

[CEMM98] R. Cleve, A. Ekert, C. Macchiavello, M. Mosca. "Quantum Algorithms Revisited", *Proceedings of the Royal Society of London A*, 454:339-354, 1998.

[vD02] W. van Dam. Private communication, 2002.

[vDMV01] W. van Dam, M. Mosca and U. Vazirani, "How Powerful is Adiabatic Quantum Computation?", *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS01)*, 279-287, Las Vegas, USA.

[vDV02] W. van Dam, U. Vazirani. Presented at QIP 2002, Yorktown Heights, NY, 2002.

[D58] P.A.M. Dirac. *Principles of Quantum Mechanics*, 4th Edn. Clarendon Press, Oxford, 1958.

[F00] S. Fenner. "An intuitive Hamiltonian for quantum search", University of South Carolina Department of Computer Science and Engineering Tech Report CSE-TR-2000-1, April 2000.

[FG98] E. Farhi, S. Gutmann. "An Analog Analogue of a Digital Quantum Computation", *Physical Review A*, 57, 2403, 1998.

[FGG00] E. Farhi, J. Goldstone, S. Gutmann. "A Numerical Study of the Performance of a Quantum Adiabatic Evolution Algorithm for Satisfiability", http://xxx.lanl.gov/abs/quant-ph/0007071, 2000.

[FGG+01] E. Farhi, J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren, D. Prada. "A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem", *Science*, Vol. 292, 2001.

[FGGS00] E. Farhi, J. Goldstone, S. Gutmann, M. Sipser. "Quantum Computation by Adiabatic Evolution", http://xxx.lanl.gov/abs/quant-ph/0001106, 2000.

[G96] L. Grover, *Proc. 28th Annual ACM Symposium on the Theory of Computation*, pp. 212-219, ACM Press, New York, 1996.

[GJ79] M. R. Garey, D. S. Johnson. *Computers and Intractability (A Guide to the Theory of NP-Completeness)*. W.H. Freeman and Company, New York, 1979.

[H00] T. Hogg. "Quantum Search Heuristics", *Physical Review A*, Vol. 61, 052311, 2000.

[K50] T. Kato. "On the Adiabatic Theorem of Quantum Mechanics", *Journ. Phys. Soc. Jap.*, 5, 435, 1950.

[K97] A. Kitaev. "Fault-tolerant Quantum Computation and Error Correction", *Russ. Math. Surv.*, 52(6):1191-1249, 1997.

[M98] M. Mosca. "Quantum Searching and Counting by Eigenvector Analysis", *Proceedings of Randomized Algorithms, Workshop of MFCS98*, Brno, Czech Republic. 1998.

[M76] A. Messiah. *Quantum Mechanics*, Vol. II, Amsterdam: North Holland, New York: Wiley, 1976.

[NC00] M. Nielsen and I. Chuang. *Quantum Computation and Quantum Information*, Cambridge University Press, Cambridge, 2000.

[S85] J.J. Sakurai. *Modern Quantum Mechanics*, The Benjamin/Cummins Publishing Company, Inc., Menlo Park, California, 1985.

[SS99] A.T. Sornborger and E.D. Stewart. "Higher Order Methods for Simulations on Quantum Computers", *Phys. Rev. A*, 60(3):1956-1965, 1999.

[Z98] Christof Zalka. "Simulating Quantum Systems on a Quantum Computer", *Proc. R. Soc. London A*, 454(1969):313-322, 1998.

[Z02] Christof Zalka. Private communication, June, 2002.