PROBLEMS OF ADIABATIC QUANTUM PROGRAM DESIGN

by

Evgeniya Khusnitdinova

B.S., Applied Mathematics and Informatics, Novosibirsk State University, 2003

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering
Boğaziçi University

2006

ACKNOWLEDGEMENTS

I would like to thank my supervisor Prof. A.C. Cem Say for his guidance and help throughout my research for this thesis, for providing me with ideas and food for thoughts, and for his incredible patience. I am also grateful for the helpful advices of my jury members, Assist. Prof. Tunga Güngör and Assist. Prof. Muhittin Mungan. I would also like to thank my family which is so far in Siberia, my friends and my colleagues for their endless support and my dear husband for being so close and so patient.

ABSTRACT

PROBLEMS OF ADIABATIC QUANTUM PROGRAM DESIGN

Although several quantum programming languages have already been proposed, none of these are based on the newly discovered adiabatic evolution approach. We acknowledge the main problem in adiabatic quantum computation to be the lack of insight that common programmers have about the design of Hamiltonians.

In the present work we provide necessary background in physics and algebra and the basics of adiabatic quantum computation with comprehensive discussion of both examples seen in literature and new ones. We examine some flow control constructs like loops and branching from the adiabatic quantum perspective to illustrate the main design problems as a first step towards the development of an adiabatic quantum programming infrastructure.

ÖZET

ADYABATİK KUANTUM PROGRAM TASARIM SORUNLARI

Şimdiye dek önerilen kuantum programlama dillerinin hiç biri yeni bulunmuş olan adyabatik evrim yaklaşımına dayanmamaktadır. Adyabatik kuantum hesaplamanın ana sorunun sıradan programcıların Hamiltonyen'lerin tasarımı hakkındaki içgörü eksikliğidir.

Bu çalışmada fizik ve cebir hakkındaki gerekli ön bilgileri verdikten sonra adyabatik kuantum hesaplama konusunun esaslarını hem literatürde bilinen, hem de bizce geliştirilmiş yeni örnekleri etraflıca tartışarak ortaya koyulmuştur. Bir adyabatik kuantum programlama altyapısının geliştirilmesi yönünde bir ilk adım olarak ana tasarım sorumlarını göstermek için döngü ve dallanma gibi akış denetimi komutlarını adyabatik kuantum açısından incelenmiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

In 1982, the physicist Richard Feynman pointed out that quantum computers may perform certain tasks more efficiently than classical computers [1]. Since then, the young field of quantum computation has enjoyed a rapid growth and excited a lot of interest. Many quantum algorithms have been discovered, the most notable being Shor's famous algorithm [2] for polynomial-time factorization. However, there does not yet seem to be a unifying set of principles by which quantum algorithms are developed; this makes it hard to discover new quantum algorithms.

In 2000, a new paradigm for designing quantum algorithms, namely, the model of quantum computation by adiabatic evolution, was discovered by Farhi et al. [3]. Adiabatic quantum computation is done by tracking the evolution of a Hamiltonian from the known ground state to a ground state which encodes the desired result by a quantum computer. It was established that this model is polynomially equivalent to the standard model of quantum circuits [4], nevertheless, it provides a completely different way of constructing quantum algorithms and reasoning about them. Shor in his review on progress in quantum algorithms [5] mentioned this model as a promising one for inventing new algorithms, although this approach has not yet been shown to yield a fast algorithm for an interesting problem.

Meanwhile, adiabatic quantum computation is a hot topic in area of quantum computation. Adiabatic quantum algorithms for many well-known problems, such as Grover's problem, Hilbert's tenth problem, various NP-complete problems (SAT, 3SAT, Exact cover, MaxCut etc.), were proposed so far. Experimental implementation of the adiabatic quantum algorithm was realized on a nuclear magnetic resonance computer with three quantum bits [6].

In parallel to the work on quantum algorithms, efforts were started to develop quantum programming languages. Several imperative quantum programming languages were introduced, such as the earliest proposal by Knill [7] in 1996, followed by QCL [8], qGCL [9], and a quantum extension of C++ by Bettelli [10]. Some studies were done in the

direction of functional quantum programming languages, such as QFC [11], QML [12], and linear lambda calculus for quantum computation [13].

Till the present moment, none of the research on quantum programming languages was done within the framework of adiabatic quantum computation. We acknowledge the main problem in adiabatic quantum computation to be the lack of insight that common programmers have about the design of Hamiltonians. Compared to the others, this approach requires more knowledge and deep understanding of the physical and mathematical background.

In the present work we provide necessary background and basics of adiabatic quantum computation with comprehensive discussion of examples seen in the literature [3, 14] and new ones. Besides, we examine some flow control constructs like loops and branching from an adiabatic quantum perspective to illustrate the main design issues, as a first step towards the development of an adiabatic quantum programming infrastructure. It is hoped that looking at the topic from this “programming” perspective could also provide a better understanding of adiabatic quantum computation.

2. PRELIMINARIES

In the following three subsections, we will give the definitions and properties of different mathematical, physical and quantum computational objects that are used throughout the present work.

2.1. Some Algebra

Definition 2.1 Vector Space: A vector space over a field F (ex. field of real or of complex numbers) is a set V together with two operations:

- vector addition: defined on the Cartesian product $V \times V$ with values in V and denoted $\mathbf{v} + \mathbf{w}$, where $\mathbf{v}, \mathbf{w} \in V$,
- scalar multiplication: defined on the Cartesian product $F \times V$ with values in V and denoted $a\mathbf{v}$, where $a \in F$ and $\mathbf{v} \in V$,

which satisfy the following axioms (for all $a, b \in F$ and $\mathbf{u}, \mathbf{v}, \mathbf{w} \in V$):

1. Vector addition is associative: $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$.
2. Vector addition is commutative: $\mathbf{v} + \mathbf{w} = \mathbf{w} + \mathbf{v}$.
3. There exists an additive identity element $\mathbf{0}$ in V, such that for all elements $\mathbf{v}$ in V, $\mathbf{v} + \mathbf{0} = \mathbf{v}$.
4. For all $\mathbf{v}$ in V, there exists an element $\mathbf{w}$ in V, such that $\mathbf{v} + \mathbf{w} = \mathbf{0}$.
5. Scalar multiplication is associative: $a(b\mathbf{v}) = (ab)\mathbf{v}$.
6. $1\ \mathbf{v} = \mathbf{v}$, where 1 denotes the multiplicative identity in F.
7. Scalar multiplication distributes over vector addition: $a(\mathbf{v} + \mathbf{w}) = a\ \mathbf{v} + a\ \mathbf{w}$.
8. Scalar multiplication distributes over scalar addition: $(a + b)\mathbf{v} = a\ \mathbf{v} + b\ \mathbf{v}$.

The elements of V are called vectors and the elements of F are called scalars. In most applications the scalars are the real or complex numbers.

Definition 2.2 Subspace: Given a vector space V, any nonempty subset W of V which is closed under addition and scalar multiplication is called a subspace of V. Subspaces of V are vector spaces (over the same field) in their own right.

Definition 2.3 Span: The intersection of all subspaces containing a given set of vectors is called their span; if no vector can be removed without diminishing the span, the set is described as being linearly independent. A linearly independent set whose span is the whole space is called a basis.

Definition 2.4 Norm: Given a vector space V over a subfield F of the complex numbers (complex numbers themselves or the real or rational numbers), a norm on V is a function $\| \ \| : V \to \mathbf{R}$; $x \to \|x\|$ with the following properties:

For all a in F and all **u** and **v** in V,

1. $\|\mathbf{v}\| \geq 0$ (*positivity*)
2. $\|\mathbf{v}\| = 0$ if and only if **v** is the zero vector (*positive definiteness*)
3. $\|a\mathbf{v}\| = |a|\|\mathbf{v}\|$ (*positive scalability*)
4. $\|\mathbf{v} + \mathbf{w}\| \leq \|\mathbf{v}\| + \|\mathbf{w}\|$ (*triangle inequality*).

Definition 2.5 Metric: Given a normed vector space $(V, \| \ \|)$, a metric (also called the distance function or simply distance) on V can be defined by $d(x, y) = \|x - y\|$. This metric d is said to be induced by $\| \ \|$.

Definition 2.6 Hilbert Space: A Hilbert space is a vector space over the complex numbers **C**. In quantum theory vectors are denoted $|\psi\rangle$. A Hilbert space:

1. Has an inner product of two vectors $|\psi\rangle$ and $|\varphi\rangle$ denoted $\langle\psi|\varphi\rangle$. Inner product maps a pair of vectors to **C** and has the following properties:
   (i) Positivity: $\langle\psi|\psi\rangle > 0$; $\langle\psi|\psi\rangle = 0$ if and only if $|\psi\rangle = 0$
   (ii) Linearity: $\langle\varphi|\left(a|\psi_1\rangle + b|\psi_2\rangle\right)\rangle = a\langle\varphi|\psi_1\rangle + b\langle\varphi|\psi_2\rangle$, where $a, b \in C$
   (iii) Skew symmetry: $\langle\varphi|\psi\rangle = \langle\psi|\varphi\rangle^*$ (symbol * denotes complex conjugation)
2. Is complete in the norm $\|\psi\| = \sqrt{\langle\psi|\psi\rangle}$.

Definition 2.7 Linear Operator: A linear operator is a function taking vectors from a vector space V to vectors from a vector space W and preserving linear combinations:

1. $\mathbf{A}:|\psi\rangle \in V \rightarrow \mathbf{A}|\psi\rangle \in W$
2. $\mathbf{A}\left(a|\psi\rangle + b|\psi\rangle\right) = a\mathbf{A}|\psi\rangle + b\mathbf{A}|\psi\rangle$

In the chosen bases of the vector spaces V and W, every linear operator $\mathbf{A}: V \rightarrow W$ can be represented as a matrix. Conversely, matrices yield examples of linear transformations: if B is a real m-by-n matrix, then the rule $f(x) = Bx$ describes a linear transformation.

Definition 2.8 Operator Norm: The operator norm of a linear operator $\mathbf{A}: V \rightarrow W$, where V and W are normed vector spaces, is the largest value by which $\mathbf{A}$ stretches an element of V: $\|\mathbf{A}\| = \sup_{\||\psi\rangle\|=1} \|\mathbf{A}|\psi\rangle\|$. One can show that the following definitions are all equivalent:

$$\begin{aligned}\|\mathbf{A}\| &= \inf\left\{c : \|\mathrm{A}\psi\| \le c\|\psi\| \text{ for all } |\psi\rangle \in V\right\} = \\ &= \sup\left\{\|\mathrm{A}\psi\| : |\psi\rangle \in V \text{ with } \|\psi\| \le 1\right\} \\ &= \sup\left\{\|\mathrm{A}\psi\| : |\psi\rangle \in V \text{ with } \|\psi\| = 1\right\} \\ &= \sup\left\{\frac{\|\mathrm{A}\psi\|}{\|\psi\|} : |\psi\rangle \in V \text{ with } \|\psi\| \neq 0\right\}\end{aligned}$$

Properties of the operator norm:

1. $\|\mathbf{A}\| \ge 0$, and $\|\mathbf{A}\| = 0$ if and only if $\mathbf{A} = \mathbf{0}$
2. $\|a\mathbf{A}\| = |a|\|\mathbf{A}\|$ for every scalar a
3. $\|\mathbf{A} + \mathbf{B}\| \le \|\mathbf{A}\| + \|\mathbf{B}\|$
4. $\|\mathbf{A}\psi\| \le \|\mathbf{A}\|\|\psi\|$ for every $\psi \in V$
5. If V, W, X are three normed vector spaces over the same base field, and $\mathbf{A}: V \rightarrow W$, $\mathbf{B}: W \rightarrow X$ are two linear operators, then $\|\mathbf{BA}\| \le \|\mathbf{B}\| \cdot \|\mathbf{A}\|$.

Definition 2.9 Adjoint Operator: Adjoint operator of $\mathbf{A}$, denoted $\mathbf{A}^*$, corresponds to the conjugate transpose of the matrix of $\mathbf{A}$. Norm of the adjoint operator: $\|\mathbf{A}\| = \|\mathbf{A}^*\|$ and $\|\mathbf{A}^*\mathbf{A}\| = \|\mathbf{A}\|^2$.

Definition 2.10 Hermitian Operator: If $\mathbf{A} = \mathbf{A}^*$ then **A** is called a Hermitian operator.

Definition 2.11 Unitary Operator: If $\mathbf{A}^{-1} = \mathbf{A}^*$ then **A** is called a unitary operator.

Definition 2.12 Positive Definite Matrix: A positive definite matrix is a Hermitian matrix A such that $v^* A v > 0$ for all vectors v.

Definition 2.13 Positive Semidefinite Matrix: A positive semidefinite matrix is a Hermitian matrix A such that $v^* A v \geq 0$ for all vectors v.

Definition 2.14 Eigenvectors and Eigenvalues: Informally, an eigenvector of a transformation is a vector whose direction is unchanged by that transformation. The factor by which the magnitude is scaled is called the eigenvalue of that vector. Mathematically, $|\varphi\rangle_\lambda$ is an eigenvector and λ the corresponding eigenvalue of an operator **A** if the equation $\mathbf{A}|\varphi\rangle = \lambda|\varphi\rangle$ is true, where $\mathbf{A}|\varphi\rangle$ is the vector obtained when applying the transformation **A** to $|\varphi\rangle_\lambda$.

Eigenvalues of square matrices can be found using the characteristic polynomial: saying that λ is an eigenvalue of the n-by-n matrix A is equivalent to stating that the system of linear equations $(A - \lambda I)v = 0$ (where I is the identity matrix) has a non-zero solution v (an eigenvector), and so it is equivalent to the determinant $\det(A - \lambda I) = 0$.

The function $p(\lambda) = \det(A - \lambda I)$ is a polynomial in λ of the n^{th} degree. This is the characteristic polynomial of A: the eigenvalues of a matrix are the zeros of its characteristic polynomial. The fundamental theorem of algebra says that this equation has exactly n roots (zeroes), counted with multiplicity. Once the eigenvalues λ are known, the eigenvectors can then be found by solving: $(A - \lambda I)v = 0$.

Definition 2.15 Spectrum: The spectrum of an operator (matrix) is a set of all its eigenvalues.

Properties of Eigenvalues:

1. The spectrum is invariant under similarity transformations: the matrices A and $P^{-1}AP$ have the same eigenvalues for any matrix A and any invertible matrix P.
2. The spectrum is also invariant under transposition: the matrices A and A^{T} have the same eigenvalues.
3. All eigenvalues of a Hermitian matrix ($A = A^{*}$) are real.
4. Eigenvectors of a Hermitian matrix form a complete orthonormal basis.
5. All eigenvalues of a positive definite matrix are positive.
6. All eigenvalues of a positive semidefinite matrix are non-negative.
7. All eigenvalues of a unitary matrix have absolute value one.
8. The trace (the sum of the elements on the main diagonal of a matrix) equals the sum of the eigenvalues: $tr(A) = \sum_i \lambda_i$.
9. The determinant equals the product of the eigenvalues (counted according to algebraic multiplicity) $\det(A) = \prod_i \lambda_i$.

Definition 2.16 Tensor product: Let A be an mxn matrix and let B be a pxq matrix. Then the tensor (Kronecker, direct) product of matrices A and B is an (mp)x(nq) matrix denoted by $A \otimes B$ and is obtained as follows:

$$A \otimes B := \begin{pmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \cdots & \cdots & \cdots & \cdots \\ a_{n1}B & a_{n2}B & \cdots & a_{nn}B \end{pmatrix}.$$

Properties of Tensor Product:

1. Bilinearity : $A \otimes (B + C) = A \otimes B + A \otimes C$ if B and C have the same size;
   $(A + B) \otimes C = A \otimes C + B \otimes C$ if A and B have the same size;
   $(kA) \otimes B = A \otimes (kB) = k(A \otimes B)$.
2. Associativity: $(A \otimes B) \otimes C = A \otimes (B \otimes C)$.
3. $A \otimes B = P(B \otimes A)Q$, where P and Q are permutation matrices. In general, the tensor product of matrices is not commutative but $A \otimes B$ and $B \otimes A$ are permutation equivalent.

4. The mixed-product property: if A, B, C and D are matrices of such size that one can form the matrix products AC and BD, then $(A \otimes B)(C \otimes D) = AC \otimes BD$
5. Inverse: if A and B are invertible, then $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$
6. Transposition: $(A \otimes B)^T = A^T \otimes B^T$
7. Spectrum: suppose that A and B are square matrices of size n and q respectively. Let $\lambda_1,...,\lambda_n$ be the eigenvalues of A and $\mu_1,...,\mu_q$ be those of B. Then the eigenvalues of $A \otimes B$ are $\lambda_i \mu_j$, $i = 1,...,n$; $j = 1,...,q$.
8. Trace: $tr(A \otimes B) = trAtrB$
9. Determinant: $\det(A \otimes B) = (\det A)^q (\det B)^n$.

Definition 2.17 Matrix Exponential: Matrix exponential is a function on square matrices analogous to the ordinary exponential function. Let *X* be n-by-n real or complex matrix. The exponential of *X*, denoted by e^X or exp(*X*), is the n-by-n matrix given by the power series:

$$e^X = \sum_{k=0}^{\infty} \frac{X^k}{k!}.$$

Properties of Matrix Exponential:

1. $e^0 = I$
2. $e^{aX} e^{bX} = e^{(a+b)X}$
3. $e^X e^{-X} = I$
4. If *Y* is invertible then $e^{YXY^{-1}} = Ye^X Y^{-1}$
5. $\det(e^X) = e^{tr(X)}$
6. $\exp(X^\dagger) = \left(e^X\right)^\dagger$, where $X^\dagger$ denotes the conjugate transpose of *X*.
7. Corollary from (6): if *X* is Hermitian then e^X is also Hermitian.
8. If a matrix is diagonal: $A = diag(\lambda_1,...,\lambda_n)$ then its exponential can be obtained by just exponentiating every entry on the main diagonal: $e^A = diag(e^{\lambda_1},...,e^{\lambda_n})$.
9. Corollary from (8): Exponential of diagonalizable matrices: if $A = UDU^{-1}$ and *D* is diagonal, then $e^A = Ue^D U^{-1}$.

10. The exponential of sums: if the matrices X and Y commute (XY=YX), then $e^{X+Y} = e^X e^Y$.

Matrix exponential can be used to solve systems of linear ordinary differential equations. The solution of $\frac{d}{dt} y(t) = Ay(t),\ y(0) = y_0$, where A is a matrix, is given by $y(t) = e^{At} y_0$.

Definition 2.18 Markov Chains: A Markov chain is a discrete-time stochastic process with the Markov property. In such a process, the past is irrelevant for predicting the future given knowledge of the present. A Markov chain is a sequence X_1, X_2, X_3, ... of random variables; the value of X_n is the state of the process at time n. If the conditional probability distribution of X_{n+1} on past states is a function of X_n alone, then:

$$P(X_{n+1} = x | X_0, X_1, X_2, ..., X_n) = P(X_{n+1} = x | X_n),$$

where x is some state of the process. The identity above identifies the Markov property.

A simple way to visualize a specific type of Markov chain is through a finite state machine. In fact, a Markov chain is a finite state machine with probabilities for each transition, that is, a probability that the next state is s_j given that the current state is s_i.

The construction of a Markov chain requires two basic ingredients, namely an initial distribution and a transition matrix.

Definition 2.19 Initial Distribution: Initial distribution is given by a distribution vector $(\lambda_1, \lambda_2, ..., \lambda_n) \in \mathbf{R}^n$ where $\lambda_i = P(X_1 = i)$ - probabilities associated with each value of variables.

Definition 2.20 Transition Matrix: Transition matrix is a square matrix whose *(i,j)* element is given by $p_{ij} = P(X_{n+1} = i | X_n = j)$ - transition probabilities. Clearly, $p_{ij} > 0$ and $\sum_{j=1}^{n} p_{ij} = 1$.

Definition 2.21 Random Walks on Graphs: Given a graph *G* and a starting point, we select a neighbor of it at random and move to this neighbor; then we select a neighbor of this point at random and move to it, etc. The random sequence of points selected in this way is a random walk on the graph. A random walk is a finite Markov chain.

The notion of a transition probability matrix allows us to examine distributions on *G* after some number of steps on our walk: if $\lambda_0 = \lambda$ is initial distribution then $\lambda_1 = \lambda P$ is the distribution after one step, and $\lambda_t = \lambda P^t$ is the distribution after t steps.

Definition 2.22 Stationary Distribution: Distribution π is stationary or stable if $\pi P = \pi$ (in other words, the stationary distribution of the Markov process is the non-negative left eigenvector π of the matrix *P* corresponding to the eigenvalue 1), in which case $\pi P^t = \pi$.

Definition 2.23 Limiting Distribution: Limiting distribution is $\lambda_\infty = \lim_{t\to\infty} \lambda P^t$. It can be shown that for every Markov process with initial distribution λ and transition matrix *P* its limiting distribution approaches to the stationary distribution π.

Definition 2.24 Stochastic Matrix: Stochastic matrix *P* is a square matrix whose columns are probability vectors, i.e., the entries in each column are nonnegative real numbers whose sum is equal to 1. It is the same thing as the matrix of transition probabilities of a finite Markov chain.

Given a stochastic matrix *P* with limiting distribution π and a subset $B \subseteq \{0,...,L\}$:

1. The flow from *B* is by definition $F(B) = \sum_{i\in B,\, j\notin B} \pi_i P_{ij}$.
2. π-weight of B is $\pi(B) = \sum_{i\in B} \pi_i$.
3. The conductance of *P* is defined by $\varphi(P) = \min_B \frac{F(B)}{\pi(B)}$, where we minimize over all non-empty subsets $B \subseteq \{0,...,L\}$ with $\pi(B) \le 1/2$.

Theorem 2.1 Conductance bound of *P*: The eigenvalue gap of *P* is at least $\frac{1}{2}\varphi(P)^2$.

2.2. Some Physics

Definition 2.25 Kets and Bras: In quantum mechanics, the state of a physical system is identified with a vector in a complex Hilbert space *H*. Each vector is called a "ket" and denoted as $|\psi\rangle$. Every ket $|\psi\rangle$ has a dual bra, denoted as $\langle\psi|$. The bra is simply the conjugate transpose of the ket and vice versa.

Applying the bra $\langle\phi|$ to the ket $|\psi\rangle$ gives the inner product of vectors ϕ and ψ and results in a complex number (called a "bra-ket" or "bracket"), which is written as $\langle\phi|\psi\rangle$. In quantum mechanics, this is the probability amplitude for the state ψ to collapse into the state φ.

Properties of bras and kets:

1. $\langle\phi|\left(c_1|\psi_1\rangle + c_2|\psi_2\rangle\right) = c_1\langle\phi|\psi_1\rangle + c_2\langle\phi|\psi_2\rangle$, where $c_1, c_2 \in C$ - complex numbers
2. $\left(c_1\langle\phi_1| + c_2\langle\phi_2|\right)|\psi\rangle = c_1\langle\phi_1|\psi\rangle + c_2\langle\phi_2|\psi\rangle$, where $c_1, c_2 \in C$ - complex numbers
3. $c_1|\psi_1\rangle + c_2|\psi_2\rangle$ is dual to $c_1{}^*\langle\psi_1| + c_2{}^*\langle\psi_2|$
4. $\langle\phi|\psi\rangle = \langle\psi|\phi\rangle^*$
5. If **A** is a linear operator, it can be applied to the ket $|\psi\rangle$ to obtain the ket $\mathbf{A}|\psi\rangle$, or it can be applied to the bra $\langle\phi|$ from the right hand side to obtain the bra $\langle\phi|\mathbf{A}$:
$$\left(\langle\phi|\mathbf{A}\right)|\psi\rangle = \langle\phi|\left(\mathbf{A}|\psi\rangle\right) = \langle\phi|\mathbf{A}|\psi\rangle$$
6. The outer product $|\phi\rangle\langle\psi|$ denotes the operator that maps the ket $|\rho\rangle$ to the ket $|\phi\rangle\langle\psi|\rho\rangle$ (where $\langle\psi|\rho\rangle$ is a scalar multiplying the vector $|\phi\rangle$)
7. Given a ket $|\psi\rangle$ of the norm 1, the orthogonal projection onto the subspace spanned by $|\psi\rangle$ is $|\psi\rangle\langle\psi|$

8. If $|\psi\rangle$ is a ket in the Hilbert space *V* and $|\phi\rangle$ is a ket in the Hilbert space *W*, the tensor product of the two kets is a ket in $V \otimes W$ denotes as:
$$|\psi\rangle|\phi\rangle \equiv |\psi\rangle \otimes |\phi\rangle \equiv |\psi\phi\rangle .$$

Definition 2.26 Hamiltonian of the Quantum System: As explained above, the physical state of a system may be characterized as a vector in a Hilbert space. Physically observable quantities are described by Hermitian operators acting on these vectors. The Hamiltonian H is the observable corresponding to the total energy of the system. Mathematically speaking, it is a Hermitian operator.

The eigenvectors of *H*, denoted $|\varphi\rangle$, provide an orthonormal basis for the Hilbert space. The spectrum of allowed energy levels of the system is given by the set of eigenvalues, denoted $\{E_\varphi\}$, solving the equation:
$$H|\varphi\rangle = E_\varphi|\varphi\rangle .$$
Since *H* is a Hermitian operator, the energy is always a real number.

The Hamiltonian generates the time evolution of quantum states.

Definition 2.27 Ground State: The lowest-energy state of the quantum system is called the ground state. Any state with energy greater than the ground state is called excited state. If more that one ground state exists, they are said to be degenerate. It turns out that degeneracy occurs whenever a nontrivial unitary operator commutes with the Hamiltonian of the system.

Definition 2.28 Local Hamiltonian: Local Hamiltonian is a Hamiltonian of the form $H = \sum_j H_j$ where each term H_j acts on a small number of particles of a quantum system (e.g., no more than some constant number, independent of the number of particles etc.). Such a Hamiltonian is realistic to construct as the Hamiltonian of an actual physical system.

Definition 2.29 Schrödinger Equation: The state of the quantum system is described by the unit vector $|\psi\rangle$ in a Hilbert space. The evolution of the system in time is defined by the Second Postulate of Quantum Mechanics: The time evolution of the state of a quantum system is described by the Schrödinger Equation:

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = H(t)|\psi(t)\rangle,$$

where $\hbar$ is a constant (Planck's constant), and H is a Hermitian operator.

Definition 2.30 Adiabatic Theorem: The adiabatic theorem tells how to follow the evolution of the quantum system described by the Schrödinger equation under certain conditions. Let Hamiltonian $H(s)$ be a smooth operator in the Hilbert space of the dimension N. Define the instantaneous eigenvectors and eigenvalues of $H(s)$ by:

$$H(s)|\varphi_i(H(s))\rangle = E_i(s)|\varphi_i(H(s))\rangle,$$

with $E_0(s) \le E_1(s) \le ... \le E_{N-1}(s)$. Define $g_{\min} = \min_{0 \le s \le 1}(E_1(s) - E_0(s))$, it is called a spectral gap of $H(s)$. Take $H(s) = H(t/T)$ where t is a real time variable ($0 \le t \le T$), s is re-scaled time variable ($0 \le s \le 1$), and so that T controls the rate at which $H(s)$ varies.

Theorem 2.2 The Adiabatic Theorem: Let $H(s)$ be a function from [0, 1] to the vector space of Hamiltonians on n qubits. Assume $H(s)$ is continuous, has a unique ground state, for every $s \in [0,1)$, and is differentiable in all but possibly finitely many points. Let $\varepsilon > 0$ and assume that the following adiabatic condition holds for all points $s \in (0,1)$ in which the derivative is defined:

$$T\varepsilon \ge \frac{\left\| \frac{d}{ds} H(s) \right\|}{(g_{\min})^2}$$

Then, a quantum system that is initialized at time 0 with the ground state of $H(0)$: $|\psi(0)\rangle = |\varphi_g(H(0))\rangle$, and evolves according to the dynamics of the Hamiltonians $H(s)$, ends up at re-scaled time 1 at a state $|\psi(1)\rangle$ that is within ε^c distance from $|\varphi_g(H(1))\rangle$ for some constant $c > 0$.

Informally, the adiabatic theorem affirms that for large T the final state of the system is very close to the ground state of $H(1)$. Just how large T should be for this to happen is determined by the spectral gap of the Hamiltonians $H(s)$.

Definition 2.31 Pauli Matrices: The Pauli matrices are a set of 2×2 complex Hermitian matrices, which proved to be very useful in the study of quantum physics.

$$\sigma_0 \equiv I \equiv \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \qquad \sigma_1 \equiv \sigma_x \equiv X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$\sigma_2 \equiv \sigma_y \equiv Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 \equiv \sigma_z \equiv Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

2.3. Some Quantum Computation

Definition 2.32 Qubit: A quantum bit, or qubit, is a unit of quantum information. That information is described by a state in a two-level quantum mechanical system which is formally equivalent to a two-dimensional Hilbert space. The two basis states are conventionally written as $|0\rangle$ and $|1\rangle$.

Definition 2.33 Qubit State: A pure qubit state is a linear superposition of those two states, and each qubit can be represented as a linear combination of $|0\rangle$ and $|1\rangle$:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where α and β are complex probability amplitudes; α and β are constrained by the equation $|\alpha|^2 + |\beta|^2 = 1$. The probability that the qubit will be measured in the state $|0\rangle$ is $|\alpha|^2$ and the probability that it will be measured in the state $|1\rangle$ is $|\beta|^2$. This is significantly different from the state of a classical bit, which can only take the value 0 or 1.

In general, we may consider a quantum system of *n* qubits. The computational basis states of this system are of the form $|x_1 x_2 ... x_n\rangle$ where $x_i \in \{0,1\}$, and so a quantum state of such a system is specified by 2^n amplitudes.

Changes occurring to a quantum state can be described using the language of quantum computation. The basic principle of quantum computation is that the quantum properties of particles can be used to represent and structure data, and that devised quantum mechanisms can be used to perform operations with this data.

Definition 2.33 Quantum Computer: A quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information. The outputs of some of the gates are connected by wires to the input of others.

Definition 2.34 Quantum Computation: Quantum computation is defined as a unitary evolution of the circuit which takes its initial state, input, into some final state, output. The entire quantum computation is thus a unitary transformation where a measurement is performed at the end to extract the result. However, a unitary transformation is itself reversible; therefore, we have to use reversible gates to be able to implement quantum gates.

The size of the circuit is governed by its number of gates. The size of the input of the circuit is governed by its number of input gates, i.e. the qubits that are prepared appropriately at the beginning of each computation performed by the circuit. Inputs are encoded in binary form in the computational basis of selected qubits often called a quantum register.

3. ADIABATIC QUANTUM PROGRAMMING

3.1. Main Rules of Adiabatic Quantum Programming

Suppose we have a quantum system of n qubits with a time-dependent Hamiltonian $H(t)$. The rules of Quantum Mechanics say (see definition 2.29) that the state of our quantum system $|\psi(t)\rangle$ changes according to the Schrödinger equation:

$$i\hbar \frac{d|\psi(t)\rangle}{dt} = H(t)|\psi(t)\rangle \tag{3.1}$$

A quantum computer program can be viewed as a specification of the time-dependent Hamiltonian $H(t)$ and an initial state of the quantum system $|\psi(0)\rangle$, both chosen so that at a time T a state $|\psi(T)\rangle$ gives the answer to our problem.

In designing our quantum computer program we can take advantage of the Quantum Adiabatic Theorem (see Theorem 2.2), which claims that a quantum system will stay near its instantaneous ground state if the Hamiltonian that governs its evolution varies slowly enough.

To follow this, first of all we have to initialize our quantum system so that the initial state of the system is the ground state of some Hamiltonian H_{init}. For this purpose it is very convenient to use easy-to-prepare states, an example of which we will see in Section 3.2.

Next, for each specific problem that we want to solve by an adiabatic quantum algorithm, we have to encode the answer to that problem in a ground state of some other Hamiltonian, called the final Hamiltonian H_{final}. Although we may be able to construct H_{final} easily for a specific problem, it may be computationally difficult to find its ground state, i.e. the answer to our problem.

Finally, to specify our algorithm we must have a time-dependent Hamiltonian $H(t)$ for $0 \le t \le T$ (T is the running time of the algorithm). We choose $H(t)$ so that $H(0) = H_{init}$ and $H(T) = H_{final}$. For any other intermediate time t, $H(t)$ smoothly interpolates between $H(0)$ and $H(T)$. This can be done, for example, by linear interpolation $H(t) = (1 - t/T)H(0) + (t/T)H(T)$, however there are no reasons not to use any other interpolation.

Therefore, if our quantum system starts at $t = 0$ in the ground state of $H(0)$, that is $|\psi_g(0)\rangle$, and if $H(t)$ varies slowly, then the evolving state vector $|\psi(t)\rangle$ will remain close to the instantaneous ground state $|\psi_g(t)\rangle$ of $H(t)$, and in particular, at time $t = T$, $|\psi(T)\rangle$ will be close to a solution to our problem.

Note that a "speed" of such a Hamiltonian $H(t)$ is $\frac{dH(t)}{dt} = \frac{H_{final} - H_{init}}{T}$ and therefore is in inverse dependence of *T*: if *T* is bigger then $H(t)$ varies slower. The correctness of an adiabatic quantum algorithm strongly depends on the choice of *T*: the slower the Hamiltonian varies, the more correctly the algorithm works. How slow is slow enough will be discussed in detail in Section 3.3.1.

Definition 3.1 Adiabatic Quantum Computation Algorithm: [3] The general adiabatic quantum computation algorithm is designed and used as follows:

1. A time-dependent Hamiltonian $H(t)$ is constructed so that the initial state of the system is the given ground state of $H(0)$ and a ground state of $H(T)$ encodes the solution of the particular problem;
2. *T* is selected big enough;
3. The system undergoes Schrödinger evolution for time *T*; at the end of which the final state of the system $|\psi(T)\rangle$ will be very close to the solution;
4. The qubits of state $|\psi(T)\rangle$ are measured.

3.2. Simple Example of an Adiabatic Quantum Program

As an example, we apply the adiabatic quantum computation algorithm (see definition 3.1) to the 3-SAT problem [3].

Example 3.1. An *n*-bit instance of 3-SAT is a Boolean formula $C_1 \wedge C_2 \wedge ... \wedge C_M$ that is specified by a collection of *M* Boolean clauses, each of which involves 3 bits. Each bit z_i, $1 \le i \le n$, can be 0 or 1, and each clause *C* is associated with 3 bits i_C, j_C, k_C.

Final Hamiltonian. For each *C* the *energy function* is defined as follows:

$$h_C(z_{i_C}, z_{j_C} z_{k_C}) = \begin{cases} 0, \text{if } (z_{i_C}, z_{j_C} z_{k_C}) \text{ satisfies clause } C \\ 1, \text{if } (z_{i_C}, z_{j_C} z_{k_C}) \text{ violates clause } C \end{cases} \quad (3.2)$$

The total energy function is $h = \sum_C h_C$, which is clearly nonnegative and equal to zero if and only if $(z_1, z_2, ..., z_n)$ satisfies all of the clauses.

In quantum computation instead of bits we have qubits $|z_i\rangle$, where $z_i = 0,1$, and $|z_1\rangle|z_2\rangle...|z_n\rangle$ are 2^n basis vectors of the Hilbert space. We turn our energy function to a quantum operator which acts on those basis vectors as follows:

1. For each clause *C*: $H_{final,C}(|z_1\rangle|z_2\rangle...|z_n\rangle) = h_C(z_{i_C}, z_{j_C}, z_{k_C})|z_1\rangle|z_2\rangle...|z_n\rangle$;
2. For all clauses the consolidated Hamiltonian is $H_{final} = \sum_C H_{final,C}$.

So, basis vectors $|z_1\rangle|z_2\rangle...|z_n\rangle$ are eigenstates of H_{final}, and H_{final} is a diagonal operator. Its eigenvalues are $h = \sum_C h_C$. The smallest eigenvalue possible is 0, and the corresponding eigenstate is a ground state of H_{final} which describes the satisfying assignment of the given Boolean formula (or, in the case of existence of multiple satisfying assignments, the uniform superposition of satisfying assignments). In this context, solving a 3-SAT problem is equivalent to finding the ground state of a Hamiltonian. H_{final} is a final Hamiltonian.

Initial Hamiltonian. Let us construct the initial Hamiltonian H_{init} whose ground state is easy to find. Define 1-bit Hamiltonian acting on *(i)*th qubit as follows:

$$H_{init}^{(i)} = \frac{1}{2}(1-\sigma_x^{(i)}), \tag{3.3}$$

where $\sigma_x^{(i)} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, so explicitly $H_{init}^{(i)} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$. Its eigenstates are $|x_i^0\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}}\sum_{z_i}|z_i\rangle$ (corresponding to eigenvalue equal to 0) and $|x_i^1\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ -1 \end{pmatrix}$ (corresponding to eigenvalue equal to 1).

For each clause C associated with bits i_C, j_C, k_C, define the Hamiltonian $H_{init,C} = H_{init}^{i_C} + H_{init}^{j_C} + H_{init}^{k_C}$ and for a whole Boolean formula $H_{init} = \sum_C H_{init,C}$. The ground state of H_{init} is:

$$|x_1^0\rangle|x_2^0\rangle...|x_n^0\rangle = \frac{1}{2^{n/2}}\sum_{z_1}\sum_{z_2}...\sum_{z_n}|z_1\rangle|z_2\rangle...|z_n\rangle. \tag{3.4}$$

This means that the starting state of the algorithm is a uniform superposition of states corresponding to all possible assignments of bit values.

Time-dependent Hamiltonian. To construct $H(t)$ we use linear interpolation $H(t) = (1-t/T)H(0) + (t/T)H(T)$. Taking $s = t/T$ we obtain the following:

$$H(s) = (1-s)H_{init} + sH_{final} = \sum_C((1-s)H_{init,C} + sH_{final,C}), \tag{3.5}$$

i.e. explicit form of $H(t)$ as a sum of Hamiltonians associated with individual clauses.

Example 3.2. <u>One-bit example</u> of the adiabatic quantum algorithm for 3-SAT problem [3]: We apply the general idea of the algorithm presented in the example 3.1 above to a concrete simple case.

Consider a one-bit problem where the single clause *C* is satisfied if and only if $z_1 = 1$. Energy function (see (3.2)) in this case is:

$$h_C = \begin{cases} 0, \text{if } z_1 = 1 \, (C \text{ is satisfied}) \\ 1, \text{if } z_1 = 0 \, (C \text{ is violated}). \end{cases}$$

The final Hamiltonian in this case is $H_{final}|z_1\rangle = h_C|z_1\rangle$, or, in matrix form $H_{final} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \frac{1}{2}(I + \sigma_Z)$, where $\sigma_Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$. Ground state in this case is $|z_1\rangle = |1\rangle$.

The initial Hamiltonian is $H_{init} = \frac{1}{2}\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} = \frac{1}{2}(I - \sigma_X)$, where $\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

Time-dependent Hamiltonian now is:

$$H(s) = (1-s)H_{init} + sH_{final} = \frac{1}{2}\left(\begin{pmatrix} 1-s & s-1 \\ s-1 & 1-s \end{pmatrix} + \begin{pmatrix} 2s & 0 \\ 0 & 0 \end{pmatrix}\right) = \frac{1}{2}\begin{pmatrix} 1+s & s-1 \\ s-1 & 1-s \end{pmatrix}.$$

From the characteristic equation for this matrix $\det(H(s) - \lambda I) = 0$ we find eigenvalues of $H(s)$: $\lambda_{1,2} = \frac{1}{2}(1 \pm \sqrt{1 - 2s + 2s^2})$. In figure 3.1 those eigenvalues are plotted; the spectral gap $g_{\min}$ between them is positive so we can adiabatically evolve from the ground state of H_{init} to $|z_1\rangle = |1\rangle$.

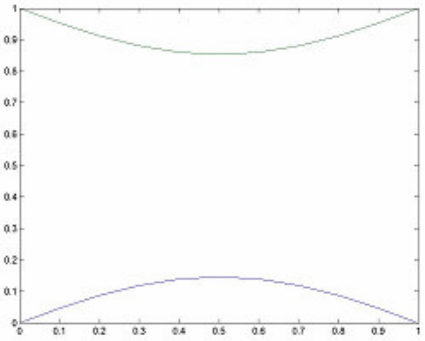


Figure 3.1. Eigenvalues of the Hamiltonian for one-bit 3-SAT example of the adiabatic quantum computation algorithm.

Example 3.3 Three-bit example for the adiabatic quantum algorithm for the SAT problem [3]: Consider three bits and a Boolean formula that consists of clauses each of which acts on two bits only: $\Phi = C_{imply}^{12} \wedge C_{disagree}^{13} \wedge C_{agree}^{23}$, where:

- C_{imply}^{12} acts on 1st and 2nd bits and satisfying assignments are 00, 01 and 11 ($i.e.\ x_1 \rightarrow x_2 \equiv \bar{x}_1 \vee x_2$)
- $C_{disagree}^{13}$ acts on 1st and 3rd bits and satisfying assignments are 01 and 10 ($i.e.\ x_1 \oplus x_3 \equiv (x_1 \vee x_3) \wedge (\bar{x}_1 \vee \bar{x}_3)$).
- C_{agree}^{23} acts on 2nd and 3rd bits and satisfying assignments are 00 and 11 ($i.e.\ \bar{x}_2 \oplus x_3 \equiv (\bar{x}_2 \vee x_3) \wedge (x_2 \vee \bar{x}_3)$).

Although each two bit clause has more than one satisfying assignment, the full problem has the unique satisfying assignment 011.

Final Hamiltonian: $H_P = H_P^{imply} + H_P^{disagree} + H_P^{agree} =$

$$= \begin{pmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 0 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{pmatrix} + \begin{pmatrix} 1 & & & & & & & \\ & 0 & & & & & & \\ & & 1 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 1 & & \\ & & & & & & 0 & \\ & & & & & & & 1 \end{pmatrix} + \begin{pmatrix} 0 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 0 \end{pmatrix} = \begin{pmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 2 & & & & & \\ & & & 0 & & & & \\ & & & & 1 & & & \\ & & & & & 3 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{pmatrix}.$$

Initial Hamiltonian:

$$H_B = H_B^{imply} + H_B^{disagree} + H_B^{agree} = (H_B^{(1)} + H_B^{(2)}) + (H_B^{(1)} + H_B^{(3)}) + (H_B^{(2)} + H_B^{(3)}) =$$

$$= 2(H_B^{(1)} + H_B^{(2)} + H_B^{(3)}) = \begin{pmatrix} 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix} +$$

$$+ \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & -1 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 3 & 0 & -1 & 0 & -1 & 0 & 0 \\ -1 & 0 & 3 & -1 & 0 & 0 & -1 & 0 \\ 0 & -1 & -1 & 3 & 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 & 3 & -1 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 3 & 0 & -1 \\ 0 & 0 & -1 & 0 & -1 & 0 & 3 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & -1 & 3 \end{pmatrix}.$$

Time dependent Hamiltonian:

$$H(s) = (1-s)H_B + sH_P = \begin{pmatrix} 3-2s & s-1 & s-1 & 0 & s-1 & 0 & 0 & 0 \\ s-1 & 3-2s & 0 & s-1 & 0 & s-1 & 0 & 0 \\ s-1 & 0 & 3-s & s-1 & 0 & 0 & s-1 & 0 \\ 0 & s-1 & s-1 & 3(1-s) & 0 & 0 & 0 & s-1 \\ s-1 & 0 & 0 & 0 & 3-2s & s-1 & s-1 & 0 \\ 0 & s-1 & 0 & 0 & s-1 & 3 & 0 & s-1 \\ 0 & 0 & s-1 & 0 & s-1 & 0 & 3-2s & s-1 \\ 0 & 0 & 0 & s-1 & 0 & s-1 & s-1 & 3-2s \end{pmatrix}.$$

Characteristic equation for this matrix:

$$\det \begin{pmatrix} 3-2s-\lambda & s-1 & s-1 & 0 & s-1 & 0 & 0 & 0 \\ s-1 & 3-2s-\lambda & 0 & s-1 & 0 & s-1 & 0 & 0 \\ s-1 & 0 & 3-s-\lambda & s-1 & 0 & 0 & s-1 & 0 \\ 0 & s-1 & s-1 & 3(1-s)-\lambda & 0 & 0 & 0 & s-1 \\ s-1 & 0 & 0 & 0 & 3-2s-\lambda & s-1 & s-1 & 0 \\ 0 & s-1 & 0 & 0 & s-1 & 3-\lambda & 0 & s-1 \\ 0 & 0 & s-1 & 0 & s-1 & 0 & 3-2s-\lambda & s-1 \\ 0 & 0 & 0 & s-1 & 0 & s-1 & s-1 & 3-2s-\lambda \end{pmatrix} = 0$$

The eigenvalues $\lambda_i(s),\ i = 1..8,$ of $H(s)$ are shown below in the Figure 3.2 (taken from [3]) and it is obvious that $g_{\min}$ is not zero. This example illustrates how the adiabatic quantum algorithm evolves to the unique satisfying assignment of several overlapping clauses even when each separate clause has more than one satisfying assignment.

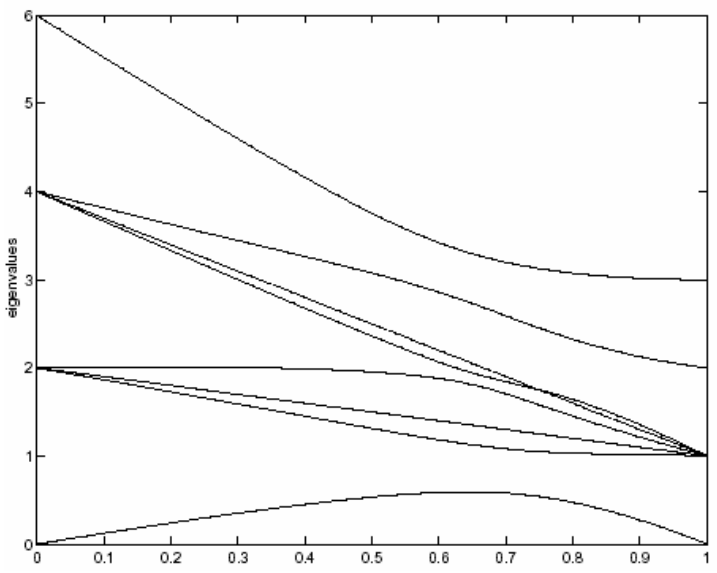


Figure 3. 2. Eigenvalues of the Hamiltonian for three-bit 3-SAT example of the adiabatic quantum computation algorithm.

3.3. Understanding the Idea of Adiabatic Quantum Computation

3.3.1. Running Time and Small Gap Problem

The correctness of the adiabatic quantum algorithm depends on how slow a time-dependent Hamiltonian *H(t)* varies. The Hamiltonian varies slower as *T* gets larger. When *T* goes to infinity, our algorithm becomes 100% correct, i.e. finds a solution to our problem with 100% probability, if solution exists.

Since in our physical world we cannot afford infinite time algorithms, we have to accept a probability of finding solution that is less than 100%, thus running an algorithm for a finite time *T*. The question is how large *T* we should choose so that the probability of finding a solution is acceptable.

According to the Quantum Adiabatic Theorem (Theorem 2.2), if $T\varepsilon \geq (g_{\min})^{-2}\left\|\frac{d}{ds}H(s)\right\|$, where $s = t/T$ - re-scaled time, then the result state will be within a distance of ε^c from a solution. Since, in particular for the linear interpolation,

$$\left\|\frac{d}{ds}H(s)\right\| = \left\|\frac{d}{ds}((1-s)H_{init} + sH_{final})\right\| = \left\|H_{final} - H_{init}\right\| \tag{3.6}$$

is no larger than the maximum eigenvalue of $\left(H_{final} - H_{init}\right)$ (see definition 2.8), which usually grows like a polynomial in *n* (*n* – length of input, number of qubits), the time of the algorithm, *T*, is governed by $g_{\min}^{-2}$.

Typically $g_{\min}$ is not zero unless the Hamiltonian has special symmetry properties. Spectral gap is zero means that there is some value $s = s'$ for which $E_0(s') = E_1(s')$. We can consider a general 2x2 Hamiltonian - a hermitian matrix with elements which are functions of *s:*

$$\begin{pmatrix} a(s) & c(s)+id(s) \\ c(s)-id(s) & b(s) \end{pmatrix}. \tag{3.7}$$

Since it is a hermitian matrix, *a*, *b*, *c*, *d* are real. Eigenvalues of this matrix are roots of the characteristic equation:

$$(a-\lambda)(b-\lambda)-(c^2+d^2)=0\,, \tag{3.8}$$

$$\lambda_{1,2}=\frac{a+b\pm\sqrt{(a+b)^2-4(ab-c^2-d^2)}}{2}. \tag{3.9}$$

Eigenvalues are equal if and only if:

$$\sqrt{(a+b)^2-4(ab-c^2-d^2)}=0;$$

$$(a-b)^2+4c^2+4d^2=0$$

if and only if

$$a(s)=b(s) \text{ and } c(s)=d(s), \tag{3.10}$$

at some point *s*. This is in general not the case and possible if Hamiltonian has some special symmetry properties.

However, for the efficiency of the algorithm it is not enough for $g_{\min}$ to be nonzero. It is also required to be not so small that the time *T* of the algorithm becomes very large. In other words, for an input of the length *n* a spectral gap $g_{\min}$ has to be not less than $\frac{1}{poly(n)}$. In this case, adiabatic quantum algorithm can solve the problem in time *T* which is less than *poly(n)*.

Therefore, in order to analyze the time complexity of an algorithm we should analyze the spectral gap of the corresponding Hamiltonian. In general, this is not always easy to manage, and often a numerical simulation is used to approximate a spectral gap [14].

3.3.2. Different Paths

In designing a quantum adiabatic algorithm, we are faced with the task of constructing a time dependent Hamiltonian from given H_{init} and H_{final}. A particular *H(t)* such that $H(0)=H_{init}$ and $H(T)=H_{final}$ is called a path from H_{init} to H_{final} in the

Hamiltonian space. The adiabatic quantum algorithm will work with any such a path as long as it is smooth and the corresponding time T is much larger than $O\left(g_{\min}^{-2}\right)$.

The simplest path which satisfies those conditions is a "straight line":

$$H(t) = (1 - t/T)H_{init} + (t/T)H_{final}. \tag{3.11}$$

But there is no reason to restrict attention only to that one. For example, one can consider paths of the form

$$H(t) = (1 - \tfrac{t}{T})H_{init} + (\tfrac{t}{T})H_{final} + (\tfrac{t}{T})(1 - \tfrac{t}{T})H_{extra}. \tag{3.12}$$

We can imagine that H_{extra} is an extra piece of the Hamiltonian that is turned off at the beginning and at the end of the evolution [15].

Since the adiabatic quantum algorithm is guaranteed to give 100% probability of success only for infinite running time, a finite running time requires repetition. Repetition can be done with different paths [15]. For example, by using different H_{extra} one can design a family of possibly random adiabatic paths. In general, different paths correspond to different spectral gaps, and varying H_{extra} may help to avoid a region in a Hamiltonian space where the gap is small and be computationally advantageous.

3.3.3. Final Hamiltonian with more than one Ground State

We design the adiabatic quantum algorithm so that the final state is the state with the minimum energy level. Therefore we have to ensure that the solution to our problem will have the minimum energy level.

What if our problem has several solutions? Since each solution corresponds to a ground state of H_{final}, the system gets to a uniform superposition of all these states. If there is no priority among solutions, there is an equal probability to measure one of those ground states. As an example consider the following simple example involving several ground states.

Example 3.4 Two-bit example for the adiabatic quantum algorithm for the SAT problem: Consider a two-bit Boolean formula

$$x_1 XOR x_2 = (x_1 \vee x_1 \vee x_2) \wedge (\bar{x}_1 \vee \bar{x}_1 \vee \bar{x}_2) = C_1 \wedge C_2 .$$

Energy functions for each of the clauses C_1 and C_2 in this case are:

$$h_{C_1} = \begin{cases} 0, \text{if } |z_1 z_2\rangle = |01\rangle, |10\rangle \text{ or } |11\rangle \ (C_1 \text{ is satisfied}) \\ 1, \text{if } |z_1 z_2\rangle = |00\rangle \ (C_1 \text{ is violated}) \end{cases}$$

$$h_{C_2} = \begin{cases} 0, \text{if } |z_1 z_2\rangle = |00\rangle, |01\rangle \text{ or } |10\rangle \ (C_2 \text{ is satisfied}) \\ 1, \text{if } |z_1 z_2\rangle = |11\rangle \ (C_2 \text{ is violated}) \end{cases}$$

Final Hamiltonian is 4x4 matrix:

$$H_{final} = H_{final}^{C_1} + H_{final}^{C_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Initial Hamiltonian is $H_{init} = H_{init}^{C_1} + H_{init}^{C_2}$.

Clauses C_1 and C_2 both act on same two bits thus their Hamiltonians are equal:

$$H_{init}^{C_1} = H_{init}^{C_2} = H_{init}{}^{1} \otimes I + I \otimes H_{init}{}^{2} = \frac{1}{2}\begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}.$$

So the initial Hamiltonian is:

$$H_{init} = 2H_{init}^{C_1} = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix}.$$

Time-dependent Hamiltonian:

$$H(s) = \begin{pmatrix} 2-s & s-1 & s-1 & 0 \\ s-1 & 2(1-s) & 0 & s-1 \\ s-1 & 0 & 2(1-s) & s-1 \\ 0 & s-1 & s-1 & 2-s \end{pmatrix}.$$

.

Characteristic equation for this matrix:

$$\det\begin{pmatrix} 2-s-\lambda & s-1 & s-1 & 0 \\ s-1 & 2(1-s)-\lambda & 0 & s-1 \\ s-1 & 0 & 2(1-s)-\lambda & s-1 \\ 0 & s-1 & s-1 & 2-s-\lambda \end{pmatrix} = (2-s-\lambda)(2(1-s)-\lambda)\left(\lambda^2 - (4-3s)\lambda + 2s(1-s)\right) = 0 .$$

Eigenvalues are as follows and showed in figure 3.3:

$$\lambda_1 = \frac{(4-3s)-\sqrt{17s^2-32s+16}}{2}, \lambda_2 = 2(1-s), \lambda_3 = 2-s, \lambda_4 = \frac{(4-3s)+\sqrt{17s^2-32s+16}}{2} .$$

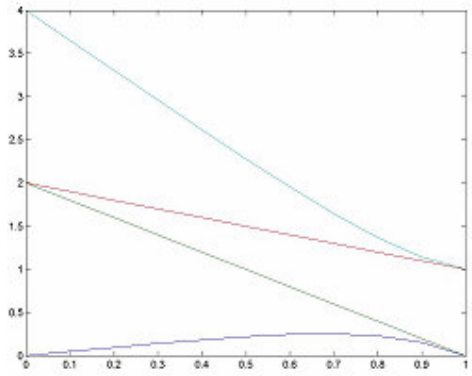


Figure 3. 3. Eigenvalues of the Hamiltonian for two-bit 3-SAT example of the adiabatic quantum computation algorithm.

There are two ground states of H_{final}, $|z_1 = 0\rangle|z_2 = 1\rangle$ and $|z_1 = 1\rangle|z_2 = 0\rangle$. The starting state $|\psi_g(0)\rangle$, which is the ground state of H_{init}, is $|x_1^0\rangle|x_2^0\rangle = \frac{1}{2^{n/2}}\sum_{z_1}\sum_{z_2}|z_1\rangle|z_2\rangle$.

There is a bit-exchange operation $X : |z_1\rangle|z_2\rangle \to |z_2\rangle|z_1\rangle$ that commutes with *H(s)*.

For any state $|\varphi\rangle$ Hamiltonian transforms it to $H|\varphi\rangle = |\phi\rangle$. If $|\varphi\rangle$ is symmetric under a bit-exchange operation, i.e. $X|\varphi\rangle = |\varphi\rangle$, so is $|\phi\rangle$, since $XH = HX$ and $X|\phi\rangle = XH|\varphi\rangle = H\underline{X|\varphi\rangle} = H|\varphi\rangle = |\phi\rangle$.

Since the starting state $|\psi_g(0)\rangle$ includes all possible combinations of two bits, it is invariant under this bit-exchange operation, and the state corresponding to $s = 1$ end of the lowest eigenvalue level $E_0(s)$ is the symmetric state $\frac{1}{\sqrt{2}}\left(|z_1 = 0\rangle|z_2 = 1\rangle + |z_1 = 1\rangle|z_2 = 0\rangle\right)$ (since it is the only possible symmetric combination of two eigenvectors). The next eigenvalue level, $E_1(s)$, begins at the antisymmetric state $\frac{1}{\sqrt{2}}\left(|x_1^0\rangle|x_2^1\rangle - |x_1^1\rangle|x_2^0\rangle\right)$ and ends at the antisymmetric state $\frac{1}{\sqrt{2}}\left(|z_1 = 0\rangle|z_2 = 1\rangle - |z_1 = 1\rangle|z_2 = 0\rangle\right)$. Since $H(s)$ commutes

with the bit-exchange operation, there can be no transitions from the symmetric to the antisymmetric states.

Therefore, $E_1(s)$ eigenvalue level is irrelevant to the adiabatic evolution of the ground state and the relevant gap is $E_2(s)$-$E_0(s)$. As it can be seen from figure 3.3, this gap is not negligibly small.

3.3.4. Termination of the Algorithm Earlier

By the adiabatic theorem the probability of finding the solution of a particular problem goes to 1 as the running time goes to infinity. We are, of course, forced to run an adiabatic quantum algorithm for a finite time and probability less than 1.

What happens with the state of the system if we terminate the algorithm earlier? How early we can stop the evolution and get a sensible probability of finding the solution?

In the efficient adiabatic quantum algorithm the spectral gap is not small and the state of the system at any time $t = t_1$ is expected to stay in the ground state of the instant Hamiltonian $H(t_1)$. In the following two examples we track the ground state change in rescaled, unitless "time" s.

Example 3.5 Ground state change in adiabatic quantum algorithm for one-bit SAT: In the one-bit example 3.2 the time-dependent Hamiltonian is $H(s) = \frac{1}{2}\begin{bmatrix} 1+s & s-1 \\ s-1 & 1-s \end{bmatrix}$ and its eigenvalues are $\lambda_{1,2} = \frac{1}{2}(1 \pm \sqrt{1-2s+2s^2})$. Their plots were presented in Figure 3.1.

Clearly, the ground state corresponds to the lowest eigenvalue $\lambda = \frac{1}{2}(1-\sqrt{1-2s+2s^2})$. When s goes from 0 to 1, the ground state changes as in Table 3.1:

Table 3. 1. Ground state change in adiabatic quantum algorithm for one-bit SAT.

Time	State	Probability to measure $\|1\rangle$
s=0	$0.7071*\|0\rangle+0.7071*\|1\rangle$	0.5
s=0.1	$0.6669*\|0\rangle+0.7451*\|1\rangle$	0.5552
s=0.2	$0.6154*\|0\rangle+0.7882*\|1\rangle$	0.6213
s=0.3	$0.5505*\|0\rangle+0.8348*\|1\rangle$	0.6969
s=0.4	$0.4719*\|0\rangle+0.8817*\|1\rangle$	0.7774
s=0.5	$0.3827*\|0\rangle+0.9239*\|1\rangle$	0.8536
s=0.6	$0.2898*\|0\rangle+0.9571*\|1\rangle$	0.9160
s=0.7	$0.2011*\|0\rangle+0.9796*\|1\rangle$	0.9596
s=0.8	$0.1222*\|0\rangle+0.9925*\|1\rangle$	0.9851
s=0.9	$0.0553*\|0\rangle+0.9985*\|1\rangle$	0.9970
s=1	$0*\|0\rangle+1*\|1\rangle$	1

Therefore, starting at $\frac{1}{\sqrt{2}}\left(|0\rangle+|1\rangle\right)$, the ground state vector rotates slowly, and as s gets closer to 1, ground state vector gets closer to $|1\rangle$. Note that already at s=0.3 the probability to find solution is higher than 2/3.

Example 3.6 Ground state change in adiabatic quantum algorithm for two-bit SAT: Let us consider the following Boolean formula which involves two bits: $x_1 \wedge x_2$. It has the unique satisfying assignment 11. The time-dependent Hamiltonian for this problem is:

$$H(s) = (1-s)H_{init} + sH_{final} = (1-s)[H_{init}^{(1)} + H_{init}^{(2)}] + s[H_{final}^{(1)} + H_{final}^{(2)}] =$$

$$= \frac{1}{2}(1-s)\begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix} + s\begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 2(1+s) & s-1 & s-1 & 0 \\ s-1 & 2 & 0 & s-1 \\ s-1 & 0 & 2 & s-1 \\ 0 & s-1 & s-1 & 2(1-s) \end{pmatrix}$$

Ground state changes as in Table 3.2 below:

Table 3. 2. Ground state change in adiabatic quantum algorithm for two-bit SAT.

Time	State	Probability to measure $\left\|11\right\rangle$
s=0	$0.5000*\left\|00\right\rangle+0.5000*\left\|01\right\rangle+0.5000*\left\|10\right\rangle+0.5000*\left\|11\right\rangle$	0.25
s=0.1	$0.4448*\left\|00\right\rangle+0.4969*\left\|01\right\rangle+0.4969*\left\|10\right\rangle+0.5552*\left\|11\right\rangle$	0.3082
s=0.2	$0.3787*\left\|00\right\rangle+0.4851*\left\|01\right\rangle+0.4851*\left\|10\right\rangle+0.6213*\left\|11\right\rangle$	0.3860
s=0.3	$0.3030*\left\|00\right\rangle+0.4596*\left\|01\right\rangle+0.4596*\left\|10\right\rangle+0.6970*\left\|11\right\rangle$	0.4858
s=0.4	$0.2226*\left\|00\right\rangle+0.4160*\left\|01\right\rangle+0.4160*\left\|10\right\rangle+0.7774*\left\|11\right\rangle$	0.6043
s=0.5	$0.1464*\left\|00\right\rangle+0.3536*\left\|01\right\rangle+0.3536*\left\|10\right\rangle+0.8536*\left\|11\right\rangle$	0.7286
s=0.6	$0.0840*\left\|00\right\rangle+0.2774*\left\|01\right\rangle+0.2774*\left\|10\right\rangle+0.9160*\left\|11\right\rangle$	0.8390
s=0.7	$0.0404*\left\|00\right\rangle+0.1970*\left\|01\right\rangle+0.1970*\left\|10\right\rangle+0.9596*\left\|11\right\rangle$	0.9208
s=0.8	$0.0149*\left\|00\right\rangle+0.1213*\left\|01\right\rangle+0.1213*\left\|10\right\rangle+0.9851*\left\|11\right\rangle$	0.9704
s=0.9	$0.0031*\left\|00\right\rangle+0.0552*\left\|01\right\rangle+0.0552*\left\|10\right\rangle+0.9969*\left\|11\right\rangle$	0.9938
s=1	$0*\left\|00\right\rangle+0*\left\|01\right\rangle+0*\left\|10\right\rangle+1*\left\|11\right\rangle$	1

Here again we observe that the ground state gets closer to $\left|11\right\rangle$ as s changes from 0 to 1. At *s*=0.5 the probability to find the solution is higher than 2/3.

3.4. Adiabatic Quantum Algorithms for Instances of NP-Complete Problems

In this section six examples of adiabatic quantum algorithms for NP-complete problems (or their instances) are given and discussed in detail. Although we give the Hamiltonians explicitly, this serves only pedagogical purposes. In reality, it is not possible (nor required) when a sizable number of bits constitute a system. The program is supposed to be describable implicitly, and doing this without knowing the final state is what requires ingenuity in this form of computation.

The reader should note that most of the examples presented (except Examples 3.9, 3.12, 3.13) are easily seen to be classically solvable in polynomial time, nevertheless, these algorithms operate in an entirely different way from classical ones and, hopefully, may bring an intuition for inventing other algorithms, more interesting and advantageous.

Example 3.7 Adiabatic quantum algorithm for 3-SAT with an arbitrary number of bits: agree and disagree clauses: [3] Consider *n* clauses, each of them acts only on adjacent bits *j* and *j+1*, where *j*=1...*n* and *n*+1 is identified with bit 1. Let each clause be either 'agree' (satisfying assignments are 00 and 11) or 'disagree' (satisfying assignments are 01 and 10). The satisfying assignment of such Boolean formula exists if the number of 'disagree' clauses is even. Suppose it is so.

Note that there are two satisfying assignments: if $w_1, w_2, ..., w_n$ is a satisfying assignment, so is $\overline{w}_1, \overline{w}_2, ..., \overline{w}_n$.

The final Hamiltonian can be written as following:

$$H_{final} = H_{C_1}^{12} + H_{C_2}^{23} + ... + H_{C_n}^{nn+1}, \tag{3.13}$$

where each C_j is either 'agree' or 'disagree', and its ground states are $|w_1\rangle|w_2\rangle...|w_n\rangle$ and $|\overline{w}_1\rangle|\overline{w}_2\rangle...|\overline{w}_n\rangle$.

Let *K* be the following unitary transformation:

$$K|z_1\rangle|z_2\rangle...|z_n\rangle = |z'_1\rangle|z'_2\rangle...|z'_n\rangle, \text{ where } z'_j = \begin{cases} \overline{z}_j, & if\ w_j = 1 \\ z_j, & if\ w_j = 0 \end{cases}. \tag{3.14}$$

Note that $H'_{final} = KH_{final} = H_{agree}^{12} + H_{agree}^{23} + ... + H_{agree}^{nn+1}$ with a symmetric ground state $|w\rangle = \frac{1}{\sqrt{2}}(|0\rangle|0\rangle...|0\rangle + |1\rangle|1\rangle...|1\rangle)$.

Note also that $KH_{init} = H_{init}$ and

$$H'(s) = (1-s)H_B + sH'_P = K((1-s)H_B + sH_P) = KH(s). \tag{3.15}$$

So the eigenvalues of $H(s)$ and $H'(s)$ are the same and we can analyze $g_{\min}$ using final Hamiltonian H'_{final}.

Recall that

$$H_{init} = \sum_{j=1\ldots n} H_{init}^{C_n} = \sum_{j=1\ldots n} 2 \cdot \tfrac{1}{2}(1-\sigma_x^{(j)}) = \sum_{j=1}^{n}(1-\sigma_x^{(j)}) . \tag{3.16}$$

For two bits

$$H_{agree}^{12} = \tfrac{1}{2}(1-\sigma_z^{(1)}\sigma_z^{(2)}) = \frac{1}{2}\left(1-\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\otimes\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}\right) = \begin{pmatrix}0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0\end{pmatrix}, \tag{3.17}$$

so

$$H'_{final} = \sum_{j=1}^{n} \tfrac{1}{2}(1-\sigma_z^{(j)}\sigma_z^{(j+1)}) \tag{3.18}$$

and the time-dependent Hamiltonian is:

$$H'(s) = (1-s)\sum_{j=1}^{n}(1-\sigma_x^{(j)}) + s\sum_{j=1}^{n}\tfrac{1}{2}(1-\sigma_z^{(j)}\sigma_z^{(j+1)}) . \tag{3.19}$$

Let G be the operator of negation:

$$G\left|z_1\right\rangle\left|z_2\right\rangle\ldots\left|z_n\right\rangle = \left|\bar{z}_1\right\rangle\left|\bar{z}_2\right\rangle\ldots\left|\bar{z}_n\right\rangle, \tag{3.20}$$

which can be written explicitly as:

$$G = \sigma_x^{(1)}\sigma_x^{(2)}\ldots\sigma_x^{(n)} = \prod_{j=1}^{n}\sigma_x^{(j)} . \tag{3.21}$$

Note that the ground state of the Hamiltonian at s=0 is invariant under G and $GH' = H'G$, so we may consider only states that are invariant under G, such as $\left|w\right\rangle$.

The following transformations will help us to write our Hamiltonian as a sum of n/2 commuting 2x2 Hamiltonians that can be diagonalized.

Define

$$b_j = \sigma_x^{(1)}\sigma_x^{(2)}\ldots\sigma_x^{(j-1)}\sigma_-^{(j)}1^{(j+1)}\ldots1^{(n)} \tag{3.22}$$

and

$$b_j^T = \sigma_x^{(1)}\sigma_x^{(2)}\ldots\sigma_x^{(j-1)}\sigma_+^{(j)}1^{(j+1)}\ldots1^{(n)} \tag{3.23}$$

where

$$\sigma_- = \tfrac{1}{2}\begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix},\ \sigma_+ = \tfrac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & -1 \end{pmatrix}. \tag{3.24}$$

Note that

$$\{b_j, b_k\} = 0 \qquad \text{(because } \sigma_-^{(j)}\sigma_x^{(j)} = -\sigma_x^{(j)}\sigma_-^{(j)}\text{)} \tag{3.25}$$

$$\{b_j, b_k^T\} = \delta_{jk} \qquad \text{(because } \sigma_-^{(j)}\sigma_+^{(j)} + \sigma_+^{(j)}\sigma_-^{(j)} = 1\text{).} \tag{3.26}$$

where operation $\{A, B\} = AB + BA$.

$$b_j^T b_j = \tfrac{1}{2}(1 - \sigma_x^{(j)}) \qquad j=1\ldots n \tag{3.27}$$

$$(b_j^T - b_j)(b_{j+1}^T + b_{j+1}) = \sigma_z^{(j)}\sigma_z^{(j+1)} \qquad j=1...n\text{-}1 \tag{3.28}$$

$$(b_n^T - b_n)(b_1^T + b_1) = -G\sigma_z^{(n)}\sigma_z^{(1)} \qquad j=n \tag{3.29}$$

Since we are restricted to the $G = I$ sector (i.e. only considering symmetric states), equations (3.28) and (3.29) are only consistent if b_{n+1}= -b_1, so we take this as the definition of b_{n+1}.

We can now express $H'(s)$ in terms of the b's:

$$H'(s) = \sum_{j=1}^{n}\{2(1-s)b_j^T b_j + \frac{s}{2}(1 - (b_j^T - b_j)(b_{j+1}^T + b_{j+1}))\}. \tag{3.30}$$

Because (3.30) is invariant under the translation $b_j \rightarrow b_{j+1}$, and is quadratic in the b_j and b_j^T, a following transformation will achieve the desired reduction of $H'(s)$. Let

$$\beta_p = \frac{1}{\sqrt{n}}\sum_{j=1}^{n} e^{i\pi pj/n} b_j \qquad \text{for } p = \pm1, \pm3, \ldots, \pm(n\text{-}1) \tag{3.31}$$

which is equivalent to

$$b_j = \frac{1}{\sqrt{n}} \sum_{p=\pm1,\pm3\ldots\pm(n-1)} e^{-i\pi pj/n}\beta_p \tag{3.32}$$

and is consistent with b_{n+1}= $-b_1$ (we assume for simplicity that n is even). Furthermore, $\{\beta_p, \beta_q\} = 0$ and $\{\beta_p, \beta_q^T\} = \delta_{pq}$, as follows from (3.25) and (3.26).

Substituting β's into $H'(s)$ gives $H'(s) = \sum_{p=1,3,\ldots(n-1)} A_p(s)$ where

$$A_p(s) = 2(1-s)[\beta_p^T\beta_p + \beta_{-p}^T\beta_{-p}] + s\left\{1 - \cos\tfrac{\pi p}{n}[\beta_p^T\beta_p - \beta_{-p}\beta_{-p}^T] + i\sin\tfrac{\pi p}{n}[\beta_{-p}^T\beta_p^T - \beta_p\beta_{-p}]\right\}.$$

Properties of β s:

$$\{\beta_p, \beta_q\} = 0, \tag{3.33}$$

$$\{\beta^T{}_p, \beta^T{}_q\} = 0, \tag{3.34}$$

$$\beta_p \beta_p = 0, \tag{3.35}$$

$$\{\beta_p, \beta_q^T\} = \delta_{pq}. \tag{3.36}$$

The A_p's commute for different values of p so we can diagonalize each A_p separately.

For each $p > 0$ let $|\Omega_p\rangle$ be such that $\beta_p|\Omega_p\rangle = \beta_{-p}|\Omega_p\rangle = 0$. When $s = 0$, $|\Omega_p\rangle$ is the ground state of A_p.

$$A_p(s)|\Omega_p\rangle = 2(1-s)[\beta_p^T\beta_p|\Omega_p\rangle + \beta_{-p}^T\beta_{-p}|\Omega_p\rangle] + s\{|\Omega_p\rangle - \cos\tfrac{\pi p}{n}[\beta_p^T\beta_p|\Omega_p\rangle -$$
$$-\beta_{-p}\beta_{-p}^T|\Omega_p\rangle] + i\sin\tfrac{\pi p}{n}[\beta_{-p}^T\beta_p^T|\Omega_p\rangle - \beta_p\beta_{-p}|\Omega_p\rangle]\} = s\{|\Omega_p\rangle + \cos\tfrac{\pi p}{n}\beta_{-p}\beta_{-p}^T|\Omega_p\rangle +$$
$$+ i\sin\tfrac{\pi p}{n}\beta_{-p}^T\beta_p^T|\Omega_p\rangle\} = s\{|\Omega_p\rangle + \cos\tfrac{\pi p}{n}(1-\beta_{-p}^T\beta_{-p})|\Omega_p\rangle + i\sin\tfrac{\pi p}{n}\beta_{-p}^T\beta_p^T|\Omega_p\rangle\} =$$
$$= s\{|\Omega_p\rangle + \cos\tfrac{\pi p}{n}|\Omega_p\rangle + i\sin\tfrac{\pi p}{n}\underbrace{\beta_{-p}^T\beta_p^T|\Omega_p\rangle}_{|\Sigma_p\rangle}\}$$

Now $A_p(s)$ only connects $|\Omega_p\rangle$ to $|\Sigma_p\rangle = \beta_{-p}^T\beta_p^T|\Omega_p\rangle$. Note that in the $\{|\Omega_p\rangle, |\Sigma_p\rangle\}$ basis $A_p(s)$ is a 2×2 matrix. To calculate its elements, see how it acts on basis vectors:

$$A_p(s)|\Omega_p\rangle = (s + s\cos\tfrac{\pi p}{n})|\Omega_p\rangle + i\sin\tfrac{\pi p}{n}|\Sigma_p\rangle \tag{3.37}$$

$$A_p(s)|\Sigma_p\rangle = 2(1-s)[\underline{\beta_p^T\beta_p|\Sigma_p\rangle} + \underline{\beta_{-p}^T\beta_{-p}|\Sigma_p\rangle}] + s\{|\Sigma_p\rangle - \cos\tfrac{\pi p}{n}[\beta_p^T\beta_p|\Sigma_p\rangle - \tag{3.38}$$
$$\underline{-\beta_{-p}\beta_{-p}^T|\Sigma_p\rangle}] + i\sin\tfrac{\pi p}{n}[\underline{\beta_{-p}^T\beta_p^T|\Sigma_p\rangle} - \underline{\beta_p\beta_{-p}|\Sigma_p\rangle}]\}.$$

Consider each element in the sum above, separately, using formulas (3.33)-(3.36). Since

$$\beta_p^T\beta_p|\Sigma_p\rangle = \beta_p^T\underline{\beta_p\beta_{-p}^T}\beta_p^T|\Omega_p\rangle = -\underline{\beta_p^T\beta_{-p}^T}\beta_p\beta_p^T|\Omega_p\rangle = \beta_{-p}^T\beta_p^T\underline{\beta_p\beta_p^T}|\Omega_p\rangle =$$
$$= \beta_{-p}^T\beta_p^T(1-\beta_p^T\underline{\beta_p})|\Omega_p\rangle = |\Sigma_p\rangle,$$

therefore

$$\beta_p^T \beta_p \left|\Sigma_p\right\rangle = \left|\Sigma_p\right\rangle . \tag{3.39}$$

Since

$$\beta_{-p}^T \beta_{-p} \left|\Sigma_p\right\rangle = \beta_{-p}^T \underline{\beta_{-p} \beta_{-p}^T} \beta_p^T \left|\Omega_p\right\rangle = \beta_{-p}^T \underline{(1 - \beta_{-p}^T \beta_{-p})} \beta_p^T \left|\Omega_p\right\rangle = \beta_{-p}^T \beta_p^T \left|\Omega_p\right\rangle -$$
$$- \beta_{-p}^T \beta_{-p}^T \underline{\beta_{-p} \beta_p^T} \left|\Omega_p\right\rangle = \left|\Sigma_p\right\rangle + \beta_{-p}^T \beta_{-p}^T \beta_p^T \underline{\beta_{-p} \left|\Omega_p\right\rangle} = \left|\Sigma_p\right\rangle,$$

therefore

$$\beta_{-p}^T \beta_{-p} \left|\Sigma_p\right\rangle = \left|\Sigma_p\right\rangle . \tag{3.40}$$

Two next elements in (3.38) are zero since

$$\beta_{-p} \beta_{-p}^T \left|\Sigma_p\right\rangle = (1 - \beta_{-p}^T \beta_{-p}) \left|\Sigma_p\right\rangle = \left|\Sigma_p\right\rangle - \beta_{-p}^T \beta_{-p} \left|\Sigma_p\right\rangle = \left|\Sigma_p\right\rangle - \left|\Sigma_p\right\rangle = 0 \tag{3.41}$$

and

$$\beta_{-p}^T \beta_p^T \left|\Sigma_p\right\rangle = \beta_{-p}^T \underline{\beta_p^T \beta_{-p}^T} \beta_p^T \left|\Omega_p\right\rangle = \beta_{-p}^T \beta_{-p}^T \underline{\beta_p^T \beta_p^T} \left|\Omega_p\right\rangle = 0 . \tag{3.42}$$

And the last element can be transformed into the following:

$$\beta_p \beta_{-p} \left|\Sigma_p\right\rangle = \beta_p \underline{\beta_{-p} \beta_{-p}^T} \beta_p^T \left|\Omega_p\right\rangle = \beta_p (1 - \beta_{-p}^T \beta_{-p}) \beta_p^T \left|\Omega_p\right\rangle = \underline{\beta_p \beta_p^T} \left|\Omega_p\right\rangle -$$
$$- \beta_p \beta_{-p}^T \underline{\beta_{-p} \beta_p^T} \left|\Omega_p\right\rangle = (1 - \beta_p^T \underline{\beta_p}) \left|\Omega_p\right\rangle - \beta_p \beta_{-p}^T \beta_p^T \underline{\beta_{-p} \left|\Omega_p\right\rangle} = \left|\Omega_p\right\rangle,$$

i.e.

$$\beta_p \beta_{-p} \left|\Sigma_p\right\rangle = \left|\Omega_p\right\rangle . \tag{3.43}$$

Finally, (3.38) can be rewritten as follows, using (3.39)-(3.43):

$$A_p(s) \left|\Sigma_p\right\rangle = 2(1-s)[\left|\Sigma_p\right\rangle + \left|\Sigma_p\right\rangle] + s\{\left|\Sigma_p\right\rangle - \cos\tfrac{\pi p}{n} \left|\Sigma_p\right\rangle + i \sin\tfrac{\pi p}{n} [-\left|\Omega_p\right\rangle]\} =$$
$$= (2.2(1-s) + s - s\cos\tfrac{\pi p}{n}) \left|\Sigma_p\right\rangle - i \sin\tfrac{\pi p}{n} \left|\Omega_p\right\rangle = (4 - 3s - s\cos\tfrac{\pi p}{n}) \left|\Sigma_p\right\rangle - i \sin\tfrac{\pi p}{n} \left|\Omega_p\right\rangle,$$

and the matrix $A_p(s)$ in $\left\{\left|\Omega_p\right\rangle, \left|\Sigma_p\right\rangle\right\}$ basis yields to:

$$A_p(s) = \begin{bmatrix} s + s\cos \pi p/n & is(\sin \pi p/n) \\ -is(\sin \pi p/n) & 4 - 3s - s\cos \pi p/n \end{bmatrix}. \tag{3.44}$$

For each p eigenvalues E_p of $A_p(s)$ are roots of the characteristic equation:

$$\det \begin{bmatrix} s + s\cos \pi p/n - E_p & is(\sin \pi p/n) \\ -is(\sin \pi p/n) & 4 - 3s - s\cos \pi p/n - E_p \end{bmatrix} =$$
$$= {E_p}^2 - 2(2-s)E_p + 4s(1-s)(1+\cos \pi p/n) = 0;$$

therefore,

$$E_p^{\pm}(s) = 2 - s \pm \{(2-s)^2 - 4s(1-s)(1+\cos \pi p / n)\}^{1/2}.$$

The ground state corresponds to the lowest energy level $\sum_p E_p^-(s)$. The next energy level is $E_1^+(s) + \sum_{p=3..} E_p^-(s)$. Therefore, the spectral gap is:

$$g(s) = E_1^+(s) - E_1^-(s) = 2\{(2-s)^2 - 4s(1-s)(1+\cos \pi p / n)\}^{1/2}. \qquad (3.45)$$

In order to find minimum spectral gap, we have to consider its first derivative:

$$\left((2-s)^2 - 4s(1-s)(1+\cos \pi p / n)\right)' = -2(2-s) - 4(1+\cos \pi p / n)(1-2s) \approx -12 + 18s\big|_{s=2/3} = 0.$$

So the minimum gap occurs very close to $s = \frac{2}{3}$ and is:

$$g_{\min} \approx E_1^+(\frac{2}{3}) - E_1^-(\frac{2}{3}) = 2\{\frac{16}{9} - \frac{8}{9}(1+\cos\frac{\pi}{n})\}^{1/2} = 2\{\frac{16}{9} - \frac{8}{9}(1+(1-2\sin^2\frac{\pi}{2n}))\}^{1/2} =$$

$$= 2\{\frac{16}{9} - \frac{16}{9}(1-\sin^2\frac{\pi}{2n})\}^{1/2} = 2\frac{4}{3}\sin\frac{\pi}{2n} \approx \frac{4\pi}{3n} > 0$$

for large *n*. We conclude, that the Hamiltonian (3.19) which describes an adiabatic quantum algorithm for 3-SAT with an arbitrary number of bits and agree and disagree clauses, has non-negligible spectral gap, thus the algorithm is efficient.

Example 3.8 Two-bit example of an adiabatic quantum algorithm for 3-SAT with agree and disagree clauses: We apply the algorithm described in the Example 3.7 for a certain case of two bits. A Boolean formula in this case is $C = C^{12}_{disagree} \wedge C^{21}_{disagree}$ which has two satisfying assignments $|w\rangle = |0\rangle|1\rangle$ and $|\overline{w}\rangle = |1\rangle|0\rangle$. Transformation *K,* according to (3.14), is $K|z_1\rangle|z_2\rangle = |z_1\rangle|\overline{z_2}\rangle$. Under this transformation,

$$KH^{12}_{disagree} = H^{12}_{agree},$$

$$KH^{31}_{disagree} = H^{21}_{agree}.$$

According to (3.16),

$$H_{init} = (1-\sigma_x^{(1)}) + (1-\sigma_x^{(2)}) = \begin{pmatrix} 2 & -1 & -1 & 0 \\ -1 & 2 & 0 & -1 \\ -1 & 0 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{pmatrix},$$

and according to (3.18),

$$H'_{final} = \tfrac{1}{2}(1-\sigma_z^{(1)}\sigma_z^{(2)}) + \tfrac{1}{2}(1-\sigma_z^{(2)}\sigma_z^{(1)}) = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

Therefore,

$$H'(s) = (1-s)\sum_{j=1}^{2}(1-\sigma_x^{(j)}) + s\sum_{j=1}^{2}\tfrac{1}{2}(1-\sigma_z^{(j)}\sigma_z^{(j+1)}) = \begin{pmatrix} 2(1-s) & s-1 & s-1 & 0 \\ s-1 & 2-s & 0 & s-1 \\ s-1 & 0 & 2-s & s-1 \\ 0 & s-1 & s-1 & 2(1-s) \end{pmatrix}.$$

In order to write $H'(s)$ in terms of *b*'s, we apply (3.22) and (3.23) to our case:

$$b_1 = \sigma_-^{(1)}1^{(2)} = \frac{1}{2}\begin{pmatrix} 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \end{pmatrix}, \quad b_1^T = \sigma_+^{(1)}1^{(2)},$$

$$b_2 = \sigma_x^{(1)}\sigma_-^{(2)} = \frac{1}{2}\begin{pmatrix} 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \\ 1 & -1 & 0 & 0 \end{pmatrix}, \quad b_2^T = \sigma_x^{(1)}\sigma_+^{(2)}.$$

Therefore, according to (3.30),

$$H'(s) = \sum_{j=1}^{2}\{2(1-s)b_j^T b_j + \frac{s}{2}(1-(b_j^T - b_j)(b_{j+1}^T + b_{j+1}))\} = \begin{pmatrix} 2(1-s) & s-1 & s-1 & 0 \\ s-1 & 2-s & 0 & s-1 \\ s-1 & 0 & 2-s & s-1 \\ 0 & s-1 & s-1 & 2(1-s) \end{pmatrix}.$$

In order to write $H'(s)$ in terms of β's, we apply (3.31):

$$\beta_1 = \frac{1}{\sqrt{2}}(e^{i\pi/2}b_1 + e^{i\pi}b_2) = \frac{1}{\sqrt{2}}(ib_1 - b_2),$$

$$\beta_{-1} = \frac{1}{\sqrt{2}}(e^{-i\pi/2}b_1 + e^{-i\pi}b_2) = \frac{1}{\sqrt{2}}(-ib_1 - b_2).$$

And finally, $H'(s)$ yields:

$$H'(s) = A_1(s) = 2(1-s)[\beta_1^T\beta_1 + \beta_{-1}^T\beta_{-1}] + s\left\{1 - \cos\tfrac{\pi}{2}[\beta_1^T\beta_1 - \beta_{-1}\beta_{-1}^T] + i\sin\tfrac{\pi}{2}[\beta_{-1}^T\beta_1^T - \beta_1\beta_{-1}]\right\}.$$

Condition $\beta_1|\Omega\rangle = \beta_{-1}|\Omega\rangle = 0$ gives us $|\Omega\rangle = \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$, and, consequently,

$$|\Sigma\rangle = \beta_{-1}^T\beta_1^T|\Omega\rangle = \frac{1}{8}\begin{pmatrix} -i & -i & -1 & -1 \\ i & i & 1 & 1 \\ -1 & -1 & -i & -i \\ 1 & 1 & i & i \end{pmatrix}\begin{pmatrix} i & i & -1 & -1 \\ -i & -i & 1 & 1 \\ -1 & -1 & i & i \\ 1 & 1 & -i & -i \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

Therefore, matrix $A_1(s)$ in $\{|\Omega\rangle, |\Sigma\rangle\}$ basis is:

$$A_1(s)=\begin{bmatrix} s & is \\ -is & 4-3s \end{bmatrix},$$

and its eigenvalues are $E_1^{\pm}(s)=2-s\pm\{(2-s)^2-4s(1-s)\}^{1/2}$. Finally, we can estimate the spectral gap of our algorithm:

$$g_{\min}=\min(E_1^{+}(s)-E_1^{-}(s))=2\{\min((2-s)^2-4s(1-s))\}^{1/2}=$$
$$=2\{(2-s)^2-4s(1-s))^{1/2}\Big|_{s=4/5}=\frac{4}{\sqrt{5}},$$

which is not near zero and ensures the efficiency of the algorithm.

Example 3.9 Adiabatic quantum algorithm for Grover Problem [3]: For the Grover problem we have a single generalized clause *G*, which depends on *n* bits with a unique satisfying assignment $w=w_1w_2...w_n$. Energy function is:

$$h_G(z)=\begin{cases}1, & z\neq w\\ 0, & z=w\end{cases}, \tag{3.46}$$

and corresponding final Hamiltonian is:

$$H_{final}\left|z\right\rangle=h_G(z)\left|z\right\rangle=\begin{cases}\left|z\right\rangle, & z\neq w\\ 0, & z=w\end{cases}=1-\left|w\right\rangle\left\langle w\right|. \tag{3.47}$$

Time dependent Hamiltonian is as follows:

$$H(s)=(1-s)H_{init}+sH_{final}=(1-s)\sum_{j=1}^{n}\tfrac{1}{2}(1-\sigma_x^{(j)})+s(1-\left|w\right\rangle\left\langle w\right|). \tag{3.48}$$

Let *K* be the following unitary transformation:

$$K\left|z_1\right\rangle\left|z_2\right\rangle...\left|z_n\right\rangle=\left|z'_1\right\rangle\left|z'_2\right\rangle...\left|z'_n\right\rangle, \text{ where } z'_j=\begin{cases}\bar{z}_j, & if\ \ w_j=1\\ z_j, & if\ \ w_j=0\end{cases}.. \tag{3.49}$$

Under this transformation *H*(*s*) becomes:

$$H'(s)=KH(s)=(1-s)\sum_{j=1}^{n}\tfrac{1}{2}(1-\sigma_x^{(j)})+s(1-\left|0\right\rangle\left\langle 0\right|). \tag{3.50}$$

H(*s*) and $H'(s)$ are unitarily equivalent thus have the same eigenvalues. Therefore, we can study $g_{\min}$ for $H'(s)$.

Note that the ground state of $H'(0) = H_{init}$ is symmetric under interchange of any two bits, and also $H'(s)$ is symmetric under interchange of any two bits. So, instead of working in 2^n dimensional space, we can work in the (n+1)-dimensional subspace of symmetrized states $\binom{n}{k}^{-1/2} \sum_{z_1+...+z_n=k} |z_1\rangle |z_2\rangle ... |z_n\rangle$. These states can be characterized as eigenvectors of $S_z = \frac{1}{2}\sum_{j=1}^{n} \sigma_z^{(j)}$ or $S_x = \frac{1}{2}\sum_{j=1}^{n} \sigma_x^{(j)}$ - total spin:

$$S_z |m_z = m\rangle = m |m_z = m\rangle, \ m = -\tfrac{n}{2}, \ -\tfrac{n}{2}+1, \ ..., \tfrac{n}{2}; \tag{3.51}$$

$$S_x |m_x = m\rangle = m |m_x = m\rangle, \ m = -\tfrac{n}{2}, \ -\tfrac{n}{2}+1, \ ..., \tfrac{n}{2}; \tag{3.52}$$

(because each $\sigma_z^{(j)}$ or $\sigma_x^{(j)}$ has two eigenvalues: -1 and 1)

$$\left|m_z = \tfrac{n}{2} - k\right\rangle = \binom{n}{k}^{-1/2} \sum_{z_1+...+z_n=k} |z_1\rangle |z_2\rangle ... |z_n\rangle, \tag{3.53}$$

$$\left|m_x = \tfrac{n}{2} - k\right\rangle = \binom{n}{k}^{-1/2} \sum_{x_1+...+x_n=k} |x_1\rangle |x_2\rangle ... |x_n\rangle. \tag{3.54}$$

Note that $\left|m_z = \tfrac{n}{2}\right\rangle = |z = 0\rangle$, and so we can rewrite $H'(s)$ as (n+1)-dimensional matrix:

$$H'(s) = (1-s)[\tfrac{n}{2} - S_x] + s[1 - \left|m_z = \tfrac{n}{2}\right\rangle\left\langle m_z = \tfrac{n}{2}\right|]. \tag{3.55}$$

Eigenvalues of $H'(s)$ are Es such that

$$H'(s)|\psi\rangle = E|\psi\rangle. \tag{3.56}$$

If we multiply left and right sides of (3.56) by $\left\langle m_x = \tfrac{n}{2} - r\right|$, we get the following:

$$(1-s)\left\langle m_x = \tfrac{n}{2} - r\right|[\tfrac{n}{2} - S_x]|\psi\rangle + s\left\langle m_x = \tfrac{n}{2} - r\right|[1 - \left|m_z = \tfrac{n}{2}\right\rangle\left\langle m_z = \tfrac{n}{2}\right|]|\psi\rangle = E\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle,$$

$$(1-s)r\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle + s\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle - s\left\langle m_x = \tfrac{n}{2} - r\right|m_z = \tfrac{n}{2}\rangle\left\langle m_z = \tfrac{n}{2}\right|\psi\rangle =$$

$$= E\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle,$$

$$[s + (1-s)r]\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle - s\left\langle m_x = \tfrac{n}{2} - r\right|m_z = \tfrac{n}{2}\rangle\left\langle m_z = \tfrac{n}{2}\right|\psi\rangle = E\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle.$$

Let $E = s + (1-s)\lambda$. Then, the above transforms to the following:

$$[s + (1-s)r]\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle - s\left\langle m_x = \tfrac{n}{2} - r\right|m_z = \tfrac{n}{2}\rangle\left\langle m_z = \tfrac{n}{2}\right|\psi\rangle = [s + (1-s)\lambda]\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle,$$

$$(1-s)(r-\lambda)\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle = s\left\langle m_x = \tfrac{n}{2} - r\right|m_z = \tfrac{n}{2}\rangle\left\langle m_z = \tfrac{n}{2}\right|\psi\rangle,$$

$$\frac{1-s}{s}\left\langle m_x = \tfrac{n}{2} - r\right|\psi\rangle = \frac{1}{r-\lambda}\left\langle m_x = \tfrac{n}{2} - r\right|m_z = \tfrac{n}{2}\rangle\left\langle m_z = \tfrac{n}{2}\right|\psi\rangle.$$

Multiply the last equation by $\left\langle m_z = \frac{n}{2} \middle| m_x = \frac{n}{2} - r \right\rangle$:

$$\frac{1-s}{s}\left\langle m_z = \tfrac{n}{2} \middle| m_x = \tfrac{n}{2} - r \right\rangle \left\langle m_x = \tfrac{n}{2} - r \middle| \psi \right\rangle =$$

$$= \frac{1}{r-\lambda}\left\langle m_z = \tfrac{n}{2} \middle| m_x = \tfrac{n}{2} - r \right\rangle \left\langle m_x = \tfrac{n}{2} - r \middle| m_z = \tfrac{n}{2} \right\rangle \left\langle m_z = \tfrac{n}{2} \middle| \psi \right\rangle$$

and sum over r:

$$\frac{1-s}{s} = \sum_{r=0}^{n} \frac{1}{r-\lambda} \underbrace{\left| \left\langle m_z = \tfrac{n}{2} \middle| m_x = \tfrac{n}{2} - r \right\rangle \right|^2}_{P_r}, \tag{3.57}$$

where

$$P_r = \left| \left\langle m_z = \tfrac{n}{2} \middle| m_x = \tfrac{n}{2} - r \right\rangle \right|^2 = \left| \binom{n}{r}^{-1/2} \sum_{x_1 + \ldots + x_n = r} \left\langle z = 0 \middle| x_1 .. x_n \right\rangle \right|^2 = \left| \binom{n}{r}^{-1/2} \binom{n}{r} \frac{1}{2^{n/2}} \right|^2 = \frac{1}{2^n}\binom{n}{r}$$

(because each $\left| x_j \right\rangle = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 \\ \pm 1 \end{pmatrix}$ and $\left\langle 0 \middle| x_j \right\rangle = \frac{1}{\sqrt{2}}$). So, the eigenvalue equation (3.57) can be written as:

$$\frac{1-s}{s} = \frac{1}{2^n} \sum_{r=0}^{n} \binom{n}{r} \frac{1}{r-\lambda}, \tag{3.58}$$

and its power is (n+1).

Take $s = s^*$ such that $\frac{1-s^*}{s^*} = \sum_{r=1}^{n} \frac{P_r}{r}$. At $s = s^*$ the eigenvalue equation (3.58) becomes:

$$\sum_{r=1}^{n} \frac{P_r}{r} = \sum_{r=0}^{n} \frac{P_r}{r-\lambda} \Rightarrow \frac{P_0}{\lambda} = \sum_{r=1}^{n} \left[\frac{P_r}{r-\lambda} - \frac{P_r}{r} \right] = \sum_{r=1}^{n} \frac{\lambda P_r}{r(r-\lambda)}. \tag{3.59}$$

Note that $P_0 = 2^{-n}$. Let $\lambda = 2^{-n/2} u$:

$$\frac{1}{u} = \sum_{r=1}^{n} \frac{u P_r}{r(r - 2^{-n/2} u)} \approx \sum_{r=1}^{n} \frac{u P_r}{r^2},$$

$$\frac{1}{u^2} \approx \sum_{r=1}^{n} \frac{P_r}{r^2},$$

$$\lambda = 2^{-n/2} u \approx \pm 2^{-n/2} \left(\sum_{r=1}^{n} \frac{P_r}{r^2} \right)^{-1/2}$$

$$g_{\min} = E_1 - E_0 \approx 2(1-s^*)2^{-n/2}(\sum_{r=1}^{n}\frac{P_r}{r^2})^{-1/2}.$$

Note that $\sum_{r=1}^{n}\frac{P_r}{r} = \frac{2}{n} + O(\frac{1}{n^2})$ and $\sum_{r=1}^{n}\frac{P_r}{r^2} = \frac{4}{n^2} + O(\frac{1}{n^3})$. Finally, $g_{\min} \approx 2\cdot 2^{-n/2}$. This means that the required time for finding the satisfying assignment grows like 2^n and quantum adiabatic evolution is not exponentially better than the classical algorithm which checks all 2^n variable assignments.

Example 3.10 Quantum adiabatic algorithm for "bush of implications" [3]: Consider the following Boolean formula:

$$z_0 \wedge (z_0 \to z_1) \wedge (z_0 \to z_2) \wedge ... \wedge (z_0 \to z_n), \tag{3.60}$$

i.e. we have *n* 2-bit imply clauses, each of them is satisfied by bit values 00, 01, 11 but not 10, and one 1-bit clause which is satisfied if and only if $z_0 = 1$. Therefore there is the unique satisfying assignment $z_0 = 1, z_1 = 1, z_2 = 1, ..., z_n = 1$.

According to general example 3.1, a final Hamiltonian is a sum of final Hamiltonians for each clause:

$$C_0 = z_0 \ : \ H_{final}^{C_0} = \frac{1}{2}(1+\sigma_z^{(0)}),$$

$$C_j = z_0 \to z_j, j \in \{1,..,n\}: \ H_{final}^{C_j} = \frac{1}{4}(1-\sigma_z^{(0)})(1+\sigma_z^{(j)}),$$

$$H_{final} = \frac{1}{2}(1+\sigma_z^{(0)}) + \frac{1}{4}\sum(1-\sigma_z^{(0)})(1+\sigma_z^{(j)}). \tag{3.61}$$

Similarly, an initial Hamiltonian is a sum of initial Hamiltonians for each clause. Note that bit z_0 is involved in (*n*+1) clauses.

$$H_{init} = (n+1)\frac{1}{2}(1-\sigma_x^{(0)}) + \sum_{j=1}^{n}\frac{1}{2}(1-\sigma_x^{(j)}). \tag{3.62}$$

Therefore, the time-dependent Hamiltonian is:

$$H(s) = (1-s)H_{init} + sH_{final} =$$

$$= (1-s)[\frac{n+1}{2}(1-\sigma_x^{(0)}) + \frac{n}{2} - S_x] + s[\frac{1}{2}(1+\sigma_z^{(0)}) + \frac{1}{2}(1-\sigma_z^{(0)})(\frac{n}{2} + S_z)]$$

States that are symmetrized in the bits 1 to n are eigenvectors of S_x or S_z (take $|m_z\rangle$s) therefore $H(s)$ acts different for different $|m_z\rangle$s and acts the same for states which are equivalent, up to a bit permutation. Adding z_0 gives us states like $|z_0\rangle|m_z\rangle$, thus we can work in $2(n+1)$ dimensional space. The matrix of $H(s)$ in this space has elements $\left(\langle z_0'|\langle m_z'|\right)H(s)\left(|z_0\rangle|m_z\rangle\right)$.

Let us calculate matrix element of S_x in $|m_z\rangle$ basis:

$$|m_z\rangle = \binom{n}{\frac{n}{2}-m_z}^{-1/2} \sum_{z_1+\ldots+z_n=\frac{n}{2}-m_z} |z_1\rangle\ldots|z_n\rangle;$$

$$\langle m_z'|S_x|m_z\rangle = \frac{1}{2}\sum_{j=1}^{n}\langle m_z'|\sigma_x^{(j)}|m_z\rangle =$$

$$= \frac{1}{2}\left[\left(\frac{n}{2}(\frac{n}{2}+1)-m_z{}^2-m_z\right)\delta_{m_z',m_z+1}+\left(\frac{n}{2}(\frac{n}{2}+1)-m_z{}'^2-m_z'\right)\delta_{m_z',m_z-1}\right]$$

In [3] eigenvalues of matrix $H(s)$ were found numerically for different n in the range from 20 to 120, which then were plotted and it was seen that the spectral gap is not small and is of the order $g_{\min} \sim n^{-p}$, $p \approx \frac{3}{8}$.

Example 3.11 Quantum adiabatic algorithm for "overconstrained 2-SAT" [3]: Suppose we have an n-bit Boolean formula consisting of $\binom{n}{2}$ 2-bit clauses, each of which is either agree or disagree clause. Under the unitary transformation K defined as follows:

$$K|z_1\rangle|z_2\rangle\ldots|z_n\rangle = |z'_1\rangle|z'_2\rangle\ldots|z'_n\rangle, \text{ where } z'_j = \begin{cases}\bar{z}_j, & if\ \ w_j = 1\\ z_j, & if\ \ w_j = 0\end{cases}, \tag{3.63}$$

all clauses become agree clauses, and we can analyze our problem for the final clause $H_{final} = \sum_{j<k} H_{agree}^{jk}$.

Time-dependent Hamiltonian is:

$$H(s) = (1-s)(n-1)\sum_{j=1}^{n}\frac{1}{2}(1-\sigma_x^{(j)}) + s\sum_{j<k}\frac{1}{2}(1-\sigma_z^{(j)}\sigma_z^{(k)}) = \\ = (1-s)(n-1)[\frac{n}{2} - S_x] + s[\frac{n^2}{4} - S_z S_z] \tag{3.64}$$

Here we again consider only symmetrized states $|m_z\rangle$ (see Example 3.10). Moreover, we note that Hamiltonian is invariant under the operation of negating all bits in z basis, and in particular, final Hamiltonian has two ground states $\left|m_z = \frac{n}{2}\right\rangle$ (all bits are 0) and $\left|m_z = -\frac{n}{2}\right\rangle$ (all bits are 1) which are clearly invariant. So, restricting attention only to those states, eigenvalues of (n+1) dimensional matrix can be found numerically and plotted. In [3] it is done for different n and is observed that $g_{\min} \sim n^{-p}$, $p \approx 0{,}7$.

Remember that evolution time for an adiabatic algorithm is $T \geq \frac{\varepsilon}{g^2_{\min}}$, where ε is of order a typical eigenvalue of H. We showed that for overconstrained 2SAT $g_{\min} \approx n^p$, $p \approx 0.7$. Also, maximum eigenvalues for H_{init} and H_{final} are of order n^2, so time required is $n^{2-2p} > n^1$.

Example 3.12 Adiabatic quantum algorithm for DOUBLE-SAT problem: DOUBLE-SAT is an NP-complete problem and is defined as follows [16]:

$$DOUBLE - SAT = \{|\phi\rangle | \phi \text{ has at least two satisfying assignments}\} \tag{3.65}$$

Suppose we have a Boolean formula. In order to learn whether this formula has at least two satisfying assignments, we apply the following algorithm:

1. Convert this formula to 3CNF.
2. Apply Adiabatic Quantum Algorithm for 3SAT, measure the final state.
   If there is more that one satisfying assignment, we have an equal probability to measure each of existing satisfying assignments.
3. Repeat step (2) m times.

If there are k ($k \geq 2$) satisfying assignments, then, during step (3) the probability not to measure two (or more) different final states is $\frac{1}{k^m} \leq \frac{1}{2^m}$. By choosing sufficient m, that probability can be decreased to a desired small value.

4. If, during the step 3, only one final state was observed, then (up to chosen probability) the given Boolean formula has a unique satisfying assignment. Otherwise, it has at least two satisfying assignments.

Example 3.13 Adiabatic quantum algorithm for $\neq SAT$ problem: $\neq SAT$ is a set of 3CNF Boolean formulas which have $\neq$ assignment [16] (assignment without assigning three true literals in any clause). $\neq SAT$ is NP-Complete. Adiabatic quantum algorithm for this problem can be proposed as follows.

Initial Hamiltonian H_{init} is the Hamiltonian from Example 3.1 with easy to prepare ground state. Final Hamiltonian is defined through the energy function. For each C the energy function is defined as follows:

$$h_C(z_{i_C}, z_{j_C} z_{k_C}) = \begin{cases} 0, \text{if } (z_{i_C}, z_{j_C} z_{k_C}) \text{ is an } \neq -\text{assignment for clause } C \\ 1, \text{if } (z_{i_C}, z_{j_C} z_{k_C}) \text{ is not an } \neq -\text{assignment for clause } C \end{cases}. \quad (3.66)$$

The total energy function is $h = \sum_C h_C$, which is clearly nonnegative and equal to zero if and only if $(z_1, z_2, ..., z_n)$ is $\neq$ assignment of the whole Boolean formula.

We turn our energy function to a quantum operator which acts on those basis vectors as follows:

- for each clause C: $H_{final,C}(|z_1\rangle|z_2\rangle...|z_n\rangle) = h_C(z_{i_C}, z_{j_C}, z_{k_C})|z_1\rangle|z_2\rangle...|z_n\rangle$;
- for all clauses the consolidated Hamiltonian is $H_{final} = \sum_C H_{final,C}$.

To construct $H(t)$ we use linear interpolation $H(t) = (1 - t/T)H(0) + (t/T)H(T)$. Taking $s = t/T$ we obtain the following:

$$H(s) = (1-s)H_{init} + sH_{final} = \sum_C ((1-s)H_{init,C} + sH_{final,C}), \quad (3.67)$$

i.e. explicit form of $H(t)$ as a sum of Hamiltonians associated with individual clauses.

4. ADIABATIC QUANTUM COMPUTATION AND CLASSICAL QUANTUM CIRCUITS

4.1. Simulation of the Adiabatic Quantum Algorithm by a Quantum Circuit

The adiabatic quantum algorithm can be recast within the conventional quantum computing paradigm [3]. Schrödinger's equation, which describes the time evolution of the state of the quantum system, is:

$$i\frac{d|\psi(t)\rangle}{dt} = H(t)|\psi(t)\rangle .$$

If $|\psi(t)\rangle = U(t,t_0)|\psi(t_0)\rangle$, where $U(t,t_0)$ is a unitary operator, then Schrödinger's equation can be rewritten for the unitary time evolution operator U:

$$i\frac{d}{dt}U(t,t_0) = H(t)U(t,t_0) . \tag{4.1}$$

$U(t,t_0)$ is called a unitary transformation induced by the Hamiltonian $H(t)$. Then

$$|\psi(T)\rangle = U(T,0)|\psi(0)\rangle ;$$

$$|\psi(T)\rangle = U(T,T-\Delta)|\psi(T-\Delta)\rangle = U(T,T-\Delta)U(T-\Delta,T-2\Delta)|\psi(T-2\Delta)\rangle = ... =$$
$$= U(T,T-\Delta)U(T-\Delta,T-2\Delta)...U(\Delta,0)|\psi(0)\rangle .$$

So, the unitary operator $U(T,0)$ can be written as a product of M factors:

$$U(T,0) = U(T,T-\Delta)U(T-\Delta,T-2\Delta)...U(\Delta,0) , \tag{4.2}$$

where $\Delta = T/M$.

If in a time interval $[l\Delta,(l+1)\Delta]$ for all $t_1,t_2 \in [l\Delta,(l+1)\Delta]$ the following is fulfilled:

$$\|\Delta H(t_1) - \Delta H(t_2)\| << \frac{1}{M} , \tag{4.3}$$

then we can consider H as a constant in a time interval $[l\Delta,(l+1)\Delta]$ and we can use the solution of the time-independent Schrödinger equation at $[l\Delta,(l+1)\Delta]$ as an approximation of the unitary time evolution operator:

$$U((l+1)\Delta,l\Delta) \approx e^{-i\Delta H(l\Delta)} . \tag{4.4}$$

Remember that $H(t) = (1-\frac{t}{T})H_{init} + \frac{t}{T}H_{final}$. Then,

$$\left\|\Delta H(t_1) - \Delta H(t_2)\right\| = \left\|\Delta\left((1-\frac{t_1}{T})H_{init} + \frac{t_1}{T}H_{final} - (1-\frac{t_2}{T})H_{init} - \frac{t_2}{T}H_{final}\right)\right\| = \tag{4.5}$$

$$= \left\|\Delta\left(\frac{t_2 - t_1}{T}H_{init} + \frac{t_1 - t_2}{T}H_{final}\right)\right\| < \left\|\Delta\left(\frac{\Delta}{T}H_{init} - \frac{\Delta}{T}H_{final}\right)\right\| =$$

$$= \frac{\Delta}{M}\left\|H_{init} - H_{final}\right\| << \frac{1}{M},$$

if

$$\Delta\left\|H_{init} - H_{final}\right\| << 1. \tag{4.6}$$

Since $\left\|H_{init} - H_{final}\right\| = poly(n)$, the number of factors

$$M = T/\Delta = T \times poly(n). \tag{4.7}$$

Each term in a product representation of U can be approximated in the same way. Taking into consideration that

$$H(l\Delta) = (1-\frac{l\Delta}{T})H_{init} + \frac{l\Delta}{T}H_{final} = uH_{init} + vH_{final}, \tag{4.8}$$

we apply the Trotter formula:

$$e^{-i\Delta H(l\Delta)} \approx (e^{-i\Delta u H_{init}/K} e^{-i\Delta v H_{final}/K})^K, \tag{4.8}$$

requiring

$$K >> M\left(1 + \Delta\left\|H_{init}\right\| + \Delta\left\|H_{final}\right\|\right)^2 = T * poly(n). \tag{4.9}$$

Remember that H_{init} is a sum of n commuting one-bit operators, so $e^{-i\Delta u H_{init}/K}$ can be written exactly as a product of n one-qubit unitary operators (because if $AB=BA$ then A and B are simultaneously diagonalizable and $e^{A+B} = e^A e^B$). Similarly, H_{final} is a sum of commuting operators and therefore can be written exactly as a product of $poly(n)$ unitary operators, each of which acts only on the qubits involved in the clause.

Finally, U can be approximated as a product of $M \times 2K \times poly(n) = T^2 poly(n)$ unitary operators each of which acts on a few qubits.

4.2. Adiabatic Quantum Algorithm for a Quantum Circuit

Given an arbitrary quantum circuit it is possible to design an adiabatic computation whose output is the same as that of the quantum circuit [4].

W.l.o.g assume that the quantum circuit consists on n qubits, and all of them are initialized to $|0\rangle$ (otherwise, we can add at most n flip-gates to get the desired input). Assume also that the circuit is a sequence of L unitary gates $U_1,...,U_L$, each of which acts on two qubits only. This sequence of gates is applied to the initial state $|0\rangle^{\otimes n}$, and the state of n qubits after application of l gates is denoted by $|\alpha(l)\rangle$. Therefore, the final state of the system is $|\alpha(L)\rangle$.

Under these conditions we would like to design an adiabatic quantum algorithm. We have to decide on the initial and final Hamiltonians.

Final Hamiltonian. We cannot define the final Hamiltonian so that its ground state is $|\alpha(L)\rangle$, since this state is the output of the quantum circuit which is not known and thus it is impossible to explicitly specify such a Hamiltonian.

Another possibility for the ground state of the final Hamiltonian is such a state that has a non-negligible inner product with $|\alpha(L)\rangle$ [4].

Consider the history state – a state that includes the entire history of the quantum computation by the quantum circuit, in superposition:

$$|\eta\rangle = \frac{1}{\sqrt{L+1}} \sum_{l=o}^{L} |\alpha(l)\rangle \otimes \left|1^l 0^{L-l}\right\rangle^c . \tag{4.10}$$

The right L qubit register (with the superscript c) serves as a clock that counts steps, starting from all $|0\rangle$'s and adding 1s from left to right after each application of the unitary gate. The best thing about the clock is that it enables a local verification of correct

propagation of the computation by quantum circuit from one step (gate) to another, which, clearly, cannot be done without including all intermediate computational steps.

Denote

$$\left|\gamma_l\right\rangle = \left|\alpha(l)\right\rangle \otimes \left|1^l 0^{L-l}\right\rangle^c, \tag{4.11}$$

then the history state's definition gets shorter:

$$\left|\eta\right\rangle = \frac{1}{\sqrt{L+1}} \sum_{l=o}^{L} \left|\gamma_l\right\rangle. \tag{4.12}$$

We would like to define a local Hamiltonian H_{final} with the ground state $\left|\eta\right\rangle$. To do this, we write H_{final} as a sum of terms:

$$H_{final} = \frac{1}{2} \sum_{l=1}^{L} H_l + H_{input} + H_{clock}. \tag{4.13}$$

The terms in H_{final} are defined such that the only state whose energy (i.e. eigenvalue) is 0 is the desired ground state. This is done by assigning an energy penalty to any other state that does not satisfy the required properties of the ground state.

H_{clock} checks that the clock's state is of the form $\left|1^l 0^{L-l}\right\rangle$ for some $0 \le l \le L$. This is achieved by assigning an energy penalty to any basis state on the clock qubits that contains the sequence $\left|01\right\rangle$:

$$H_{clock} = \sum_{l=1}^{L-1} \left|01\right\rangle\left\langle 01\right|_{l,l+1}^{c}. \tag{4.14}$$

Subscript indicates which clock qubits the projection operates on. Note that illegal clock states are eigenvectors of H_{clock} with eigenvalue at least 1, and legal clock states have eigenvalue 0.

H_{input} checks that if the clock is $\left|0^L\right\rangle^c$, the computation qubits must be in the state $\left|0^n\right\rangle$:

$$H_{input} = \sum_{i=1}^{n} \left|1\right\rangle\left\langle 1\right|_i \otimes \left|0\right\rangle\left\langle 0\right|_1^c. \tag{4.15}$$

Again, note that any state which has the clock in the initial state $\left|0^L\right\rangle^c$, is the eigenvector of the H_{input} with the eigenvalue at least 1, unless the computation qubits are in state $\left|0^n\right\rangle$, which in this case is the ground state with the eigenvalue 0.

We now proceed to the first term of the H_{final}. Each Hamiltonian H_l checks that the propagation from step $l-1$ to l is correct and corresponds to the application of the gate U_l. For $1 < l < L$, it is defined as

$$H_l = I \otimes |100\rangle\langle 100|^c_{l-1,l,l+1} - U_l \otimes |110\rangle\langle 100|^c_{l-1,l,l+1} - \\ - U_l^{\perp} \otimes |100\rangle\langle 110|^c_{l-1,l,l+1} + I \otimes |110\rangle\langle 110|^c_{l-1,l,l+1}. \tag{4.16}$$

Three-qubit clock terms move the clock state one step forward, one step backward or leave unchanged, and the corresponding matrices $U_l, U_l^{\perp}$ describe the associated time evolution.

Due to the unitary transformations regarding the clock qubits state, the Hamiltonian H_l acts non-trivially only on intermediate computation states $\left|\gamma_{l-1}\right\rangle$ and $\left|\gamma_l\right\rangle$. Since

$$\left|\gamma_{l-1}\right\rangle = \left|\alpha(l-1)\right\rangle \otimes \left|100\right\rangle^c_{l-1,l,l+1} \text{ and } \left|\gamma_l\right\rangle = \left|\alpha(l)\right\rangle \otimes \left|110\right\rangle^c_{l-1,l,l+1}, \tag{4.17}$$

$$\left|\alpha(l)\right\rangle = U_l \left|\alpha(l-1)\right\rangle \text{ and } \left|\alpha(l-1)\right\rangle = U_l^{\perp} \left|\alpha(l)\right\rangle, \tag{4.18}$$

we have

$$H_l \left|\gamma_{l-1}\right\rangle = \left|\gamma_{l-1}\right\rangle - U_l \left|\alpha(l-1)\right\rangle \otimes |110\rangle\langle 100\|100\rangle^c_{l-1,l,l+1} - 0 + 0 = \left|\gamma_{l-1}\right\rangle - \left|\gamma_l\right\rangle, \tag{4.19}$$

$$H_l \left|\gamma_l\right\rangle = 0 - 0 - U_l^{\perp} \left|\alpha(l)\right\rangle \otimes |100\rangle\langle 110\|110\rangle^c_{l-1,l,l+1} + \left|\gamma_l\right\rangle = -\left|\gamma_{l-1}\right\rangle + \left|\gamma_l\right\rangle. \tag{4.20}$$

For the boundary cases $l = 1, L$

$$H_1 = I \otimes |00\rangle\langle 00|^c_{1,2} - U_1 \otimes |01\rangle\langle 00|^c_{1,2} - U_1^{\perp} \otimes |00\rangle\langle 10|^c_{1,2} + I \otimes |10\rangle\langle 10|^c_{1,2} \tag{4.21}$$

$$H_L = I \otimes |10\rangle\langle 10|^c_{L-1,L} - U_L \otimes |11\rangle\langle 10|^c_{L-1,L} - U_L^{\perp} \otimes |10\rangle\langle 11|^c_{L-1,L} + I \otimes |11\rangle\langle 11|^c_{L-1,L} \tag{4.22}$$

It is easy to check that $\left|\eta\right\rangle$ is the ground state of the H_{final}:

$$H_{final}|\eta\rangle = \left(H_{clock} + H_{input} + \frac{1}{2}\sum_{l=1}^{L} H_l \right)|\eta\rangle = H_{clock}|\eta\rangle + H_{input}|\eta\rangle + \frac{1}{2}\sum_{l=1}^{L} H_l|\eta\rangle .$$

$$H_{clock}|\eta\rangle = \frac{1}{\sqrt{L+1}}\sum_{l=0}^{L} H_{clock}|\gamma_l\rangle = 0 ,$$

since each $|\gamma_l\rangle$ contains the legal clock state thus isn't penalized.

$$H_{init}|\eta\rangle = \frac{1}{\sqrt{L+1}}\sum_{l=0}^{L} H_{init}|\gamma_l\rangle = 0 ,$$

since the only $|\gamma_l\rangle$ which correspond to clock state $|0^L\rangle^c$ is $|\gamma_0\rangle = |0^n\rangle \otimes |0^L\rangle^c$ and thus isn't penalized either.

$$\frac{1}{2}\sum_{l=1}^{L} H_l|\eta\rangle = \frac{1}{2}\sum_{l=1}^{L} H_l\left(\frac{1}{\sqrt{L+1}}\sum_{l=0..L}|\gamma_l\rangle \right) = \frac{1}{2\sqrt{L+1}}\sum_{l=1}^{L}\left(H_l|\gamma_{l-1}\rangle + H_l|\gamma_l\rangle \right) =$$

$$= \frac{1}{2\sqrt{L+1}}\sum_{l=1}^{L}\left(|\gamma_{l-1}\rangle - |\gamma_l\rangle - |\gamma_{l-1}\rangle + |\gamma_l\rangle \right) = 0$$

Finally, $H_{final}|\eta\rangle = 0$.

Initial Hamiltonian. Having in mind all the above, we would like to define a local Hamiltonian H_{init} with the ground state $|\gamma_0\rangle = |0^n\rangle \otimes |0^L\rangle^c$:

$$H_{init} = H_{clockinit} + H_{input} + H_{clock} . \tag{4.23}$$

H_{input} is responsible for the desired input, H_{clock} is responsible for the correctness of the clock qubits state.

$$H_{clockinit} = |1\rangle\langle 1|_1^c , \tag{4.24}$$

i.e. penalizes states whose first clock qubit is 1.

Summarizing the effects of the three terms in H_{init}, the initial Hamiltonian has the ground state $|\gamma_0\rangle = |0^n\rangle \otimes |0^L\rangle^c$ corresponding to the 0 eigenvalue:

$$H_{init}|\gamma_0\rangle = H_{clockinit}|\gamma_0\rangle + H_{input}|\gamma_0\rangle + H_{clock}|\gamma_0\rangle = 0 + 0 + 0 = 0 .$$

Time-dependent Hamiltonian. The time-dependent Hamiltonian is constructed from H_{init} and H_{final} by linear interpolation:

$$H(s) = (1-s)H_{init} + sH_{final}, \text{ where } s = \frac{t}{T}.$$

Analysis of the spectral gap of the time-dependent Hamiltonian. First of all, it is sensible to consider only states from the subspace S_0, which is spanned by $|\gamma_0\rangle,...,|\gamma_L\rangle$ [4].

Note that S_0 is invariant under $H(s)$. To check this, take an arbitrary linear combination of states from S_0, $\sum_{l=0}^{L} c_l |\gamma_l\rangle$:

$$H(s)\sum_{l=0}^{L} c_l |\gamma_l\rangle = \sum_{l=0}^{L} c_l H(s)|\gamma_l\rangle = s\sum_{l=0}^{L} c_l H_{init}|\gamma_l\rangle + (1-s)\sum_{l=0}^{L} c_l H_{final}|\gamma_l\rangle =$$
$$= s\sum_{l=0}^{L} a_l |\gamma_l\rangle + (1-s)\sum_{l=0}^{L} b_l |\gamma_l\rangle$$

since terms in H_{init} and H_{final} have $|\gamma_l\rangle$'s either as eigenvectors (Hamiltonians $H_{input}, H_{clock}, H_{clockinit}$) or transfer $|\gamma_l\rangle$'s to the linear combination of $|\gamma_l\rangle$'s (Hamiltonians H_l).

Since we assume that our initial state is $|\gamma_0\rangle \in S_0$ and $H(s)S_0 \subseteq S_0$, then all possible states during the evolution are within S_0 and it is sufficient to bound the spectral gap of $H(s)$ in the subspace S_0. In the analysis below $H(s)$ means the restriction of $H(s)$ to S_0.

Let us take $|\gamma_0\rangle,...,|\gamma_L\rangle$ as a basis of S_0 and write H_{init} and H_{final} in this basis. Initial Hamiltonian is

$$H_{init} = \begin{pmatrix} 0 & 0 & \dots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix},$$

and its spectral gap is, obviously, equal to 1. Final Hamiltonian is

$$H_{final} = \tfrac{1}{2}|\gamma_0\rangle\langle\gamma_0| - \tfrac{1}{2}|\gamma_0\rangle\langle\gamma_1| - \tfrac{1}{2}|\gamma_L\rangle\langle\gamma_{L-1}| + \tfrac{1}{2}|\gamma_L\rangle\langle\gamma_L| +$$
$$+ \sum_{l=1}^{L-1}(-\tfrac{1}{2}|\gamma_l\rangle\langle\gamma_{l-1}| + |\gamma_l\rangle\langle\gamma_l| - \tfrac{1}{2}|\gamma_l\rangle\langle\gamma_{l+1}|) =$$
$$= \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} & 0 & & \cdots & & 0 \\ -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & & \ddots & \\ 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 & & \vdots \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ \vdots & & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} & 0 \\ & \ddots & & 0 & -\frac{1}{2} & 1 & -\frac{1}{2} \\ 0 & & \cdots & & 0 & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

Our task is to lower bound the spectral gap $g_{\min}$ of the Hamiltonian $H(s) = (1-s)H_{init} + sH_{final}$. In the analysis below we consider two cases, $s < 1/3$ and $s \geq 1/3$.

Case 1: $\mathbf{s < \frac{1}{3}}$. $H(s)$ is sufficiently close to H_{init}, whose spectral gap is 1. We can use the following lemma:

Lemma 4.1 Gerschgorin's Circle Theorem [4]: Let *A* be any matrix with entries a_{ij}. Consider the discs D_i in the complex plane given by

$$D_i = \left\{ z \mid |z - a_{ii}| \leq \sum_{j \neq i} |a_{ij}| \right\}, 1 \leq i \leq n.$$

Then the eigenvalues of *A* are contained in $\cup D_i$ and any connected component of $\cup D_i$ contains as many eigenvalues of *A* as the number of discs that form this component.

For $s < 1/3$,

$$H(s)_{1,1} = (1-s)*0 + s*1/2 < 1/6$$

and

$$\sum_{j \neq 1} |H(s)_{1,j}| = |(1-s)*0 - s*1/2| < 1/6.$$

Therefore, according to the Theorem above, there is one eigenvalue in

$$D_1 = \left\{ z \mid |z - H(s)_{1,1}| \leq \sum_{j \neq 1} |H(s)_{1,j}| \right\} \subset \{ z \mid |z - 1/6| < 1/6 \} = \{ z \mid 0 < z < 1/3 \},$$

i.e. there is one eigenvalue smaller than 1/3.

Moreover, for any $i \neq 1$,

$$H(s)_{i,i} \geq (1-s)*1 + s*1/2 > 2/3 + 1/6 = 5/6$$

and

$$\sum_{j \neq i} \left| H(s)_{i,j} \right| \leq \left| (1-s)*0 - s*1/2 \right| < 1/6 .$$

Therefore, for any $i \neq 1$,

$$D_i = \left\{ z \mid \left| z - H(s)_{i,i} \right| \leq \sum_{j \neq i} \left| H(s)_{i,j} \right| \right\} \subset \left\{ z \mid \left| z - 5/6 \right| < 1/6 \right\} = \left\{ z \mid 2/3 < z < 1 \right\},$$

i.e. all other eigenvalues are larger than 2/3. Hence, the spectral gap is at least 1/3.

Case 2: $s \geq \frac{1}{3}$. We note that $H_{final}(s)$ is the Laplacian of the simple random walk [4] (i.e. matrix of the form $\Lambda = I - M$, where M is stochastic symmetric matrix related to the particular Markov process) of a particle on a line of length L+1. A standard result in Markov chain theory implies that the spectral gap of $H_{final}(s)$ is $\Omega(1/L^2)$. For $s \geq 1/3$, H(s) is sufficiently close to $H_{final}(s)$ to apply Markov chain techniques, as follows.

Let $(\alpha_0, ..., \alpha_L)^\dagger$ be the ground state of H(s) with eigenvalue λ. Define the Hermitian matrix $G(s) = I - H(s)$. It is easy to see that G(s) satisfies the condition of the following fact for all s>0.

Fact 4.2 [4] Let G be a Hermitian matrix with real non-negative entries. If there exists a finite k such that all entries of G^k are positive, then G's largest eigenvalue is positive, and all other eigenvalues are strictly smaller in absolute value. Moreover, the corresponding eigenvector is unique, and all its entries are positive.

We obtain that the largest eigenvalue $\mu = 1 - \lambda$ of $G(s)$ is positive and non-degenerate and the corresponding eigenvector $(\alpha_0, ..., \alpha_L)^\dagger$ has positive entries. We can now map the matrix $G(s)$ to a stochastic matrix $P(s)$ as follows:

$$P_{ij} = \frac{\alpha_j}{\mu\alpha_i} G_{ij} . \tag{4.25}$$

.

The matrix P is well defined and is stochastic because all its entries are non-negative and each of its rows sums up to one:

$$\sum_j P_{ij} = \frac{\sum_j \alpha_j G_{ij}}{\mu\alpha_i} = \frac{\mu\alpha_i}{\mu\alpha_i} = 1 .$$

The transition matrix $P(s)$ describes a random walk on the line of L+1 sites. We make use of the following fact:

Fact 4.3 [4] The vector $(v_0,...,v_L)$ is an eigenvector of G with eigenvalue δ if and only if $(\alpha_0 v_0,...,\alpha_L v_L)$ is a left eigenvector of P with eigenvalue δ/μ.

Proof of Fact 4.3 $\boxed{\Rightarrow}$ Let $G(v_0,...,v_L) = \delta(v_0,...,v_L)$. Check the conclusion of the fact.

$$(\alpha_0 v_0,...,\alpha_L v_L)P = (\sum_{i=1}^{L} \alpha_i v_i P_{i0}, \sum_{i=1}^{L} \alpha_i v_i P_{i1},...)$$

$$\sum_{i=1}^{L} \alpha_i v_i P_{ij} = \sum_{i=1}^{L} \alpha_i v_i \frac{\alpha_j}{\mu\alpha_i} G_{ij} = \sum_{i=1}^{L} v_i \frac{\alpha_j}{\mu} G_{ij} = \frac{\alpha_j}{\mu} \sum_{i=1}^{L} v_i G_{ij} = \frac{\alpha_j}{\mu} * \delta v_j = \frac{\delta}{\mu} \alpha_j v_j$$

Therefore, $(\alpha_0 v_0,...,\alpha_L v_L)P = \frac{\delta}{\mu}(\alpha_0 v_0,...,\alpha_L v_L)$.

$\boxed{\Leftarrow}$ Let $(\alpha_0 v_0,...,\alpha_L v_L)P = \frac{\delta}{\mu}(\alpha_0 v_0,...,\alpha_L v_L)$. Following the same sequence of equivalences as above, we get:

$$\frac{\delta}{\mu}\alpha_j v_j = \sum_{i=1}^{L} \alpha_i v_i P_{ij} = \sum_{i=1}^{L} \alpha_i v_i \frac{\alpha_j}{\mu\alpha_i} G_{ij} = \frac{\alpha_j}{\mu} \sum_{i=1}^{L} v_i G_{ij} ,$$

and therefore

$$\sum_{i=1}^{L} v_i G_{ij} = \delta v_j ,$$

$$G(v_0,...,v_L) = (\sum_{i=1}^{L} v_i G_{i1},...,\sum_{i=1}^{L} v_i G_{iL}) = \delta(v_0,...,v_L).$$

Since we have taken $G(s) = I - H(s)$, which has the largest eigenvector $(\alpha_0,...,\alpha_L)^\dagger$ with eigenvalue $\mu = 1-\lambda$, and this eigenvector is at the same time the ground state of $H(s)$ with eigenvalue λ, then, according to the statements above, the matrix P has the left eigenvector $(\alpha_0{}^2,...,\alpha_L{}^2)^\dagger$ with eigenvalue $\frac{1-\lambda}{1-\lambda} = 1$. Therefore, the limiting distribution of the stochastic matrix P is $\pi = \frac{1}{\sum \alpha_i{}^2}(\alpha_0{}^2,...,\alpha_L{}^2)$.

It is useful to note the relation between the spectral gap of $H(s)$ and eigenvalues of $P(s)$. If $g_{\min} = E_1 - E_0$, where E_0, E_1 are the lowest and the second lowest eigenvalues of $H(s)$ (using the above notation $E_0 = \lambda$), then the difference between the largest and second largest eigenvalues of $G(s)$ is $(1-E_0)-(1-E_1) = E_1 - E_0 = g_{\min}$, and the difference between the largest and second largest eigenvalues of $P(s)$ is $\frac{1-E_0}{1-\lambda} - \frac{1-E_1}{1-\lambda} = \frac{g_{\min}}{1-\lambda}$.

We bound the eigenvalue gap of $P(s)$ using the conductance bound (for details, check Section 2.1). The following claim has as corollary the monotonicity of π.

Claim 4.4 [4] For all $0 \le s \le 1$ the ground state of $H(s)$ is monotone, namely $\alpha_0 \ge \alpha_1 \ge ... \ge \alpha_L \ge 0$.

Proof of Claim 4.4 The case s=0 is obvious. Assume that s>0. We first claim that the ground state $(\alpha_0,...,\alpha_L)^\dagger$ can be written in the following form:

$$(\alpha_0,...,\alpha_L)^\dagger = \frac{1}{c_0}\lim_{l\to\infty}(G(s)/\mu)^l(1,...,1)^\dagger, \tag{4.26}$$

where c_0 is some positive constant. To see this, let $|v_0\rangle, |v_1\rangle, ..., |v_L\rangle$ be an orthonormal set of eigenvectors of $G(s)$, with corresponding eigenvalues $\mu_0 \ge \mu_1 \ge ... \ge \mu_L$. By fact 4.2, the largest eigenvalue of $G(s)$ corresponds to the unique eigenvector, and hence we have $|v_0\rangle = (\alpha_0,...,\alpha_L)^\dagger$ and $\mu_0 = \mu$.

The set of eigenvectors $|v_i\rangle$ forms an orthonormal basis, and we can write $(1,...,1)^\dagger$ in terms of this basis:

$$(1,...,1)^\dagger = \sum_i c_i |v_i\rangle .$$

Now we have that:

$$(G(s)/\mu)^l (1,...,1)^\dagger = (G(s)/\mu)^l \sum_i c_i |v_i\rangle = \sum_i \frac{c_i}{\mu^l} G(s)^l |v_i\rangle = \sum_i \frac{c_i}{\mu^l} \mu_i{}^l |v_i\rangle = \sum_i c_i \left(\frac{\mu_i}{\mu^l}\right)^l |v_i\rangle$$

By fact 4.2, we have $|\mu_i| < \mu$ for all $i \neq 0$, and $\mu > 0$. We thus have that

$$\lim_{l\to\infty}(G(s)/\mu)^l (1,...,1)^\dagger = \lim_{l\to\infty} \sum_i c_i \left(\frac{\mu_i}{\mu^l}\right)^l |v_i\rangle = \lim_{l\to\infty}\left(c_0 |v_0\rangle + \sum_{i>0} c_i \left(\frac{\mu_i}{\mu^l}\right)^l |v_i\rangle \right) =$$
$$= c_0 |v_0\rangle = c_0 (\alpha_0,...,\alpha_L)^\dagger .$$

It is easy to check that $G(s)$ preserves monotonicity, namely, if $G(s)$ is applied to a monotone vector, the result is a monotone vector. Hence, when $G(s)/\mu$ is applied to a monotone vector $(1,...,1)^\dagger$, the result is a monotone vector. Thus, $c_0|v_0\rangle$is monotone. Finally, we observe that $c_0 = (1,...,1)\cdot|v_0\rangle$ - the inner product of two vectors, entries of $|v_0\rangle$ are all positive (by fact 4.2), therefore $c_0 > 0$. This implies that $|v_0\rangle = (\alpha_0,...,\alpha_L)^\dagger$ is monotone, as desired.

Corollary 4.5 [4] The limiting distribution of $P(s)$, $\pi = \frac{1}{\sum \alpha_i{}^2}(\alpha_0{}^2,...,\alpha_L{}^2)$, is monotone.

Proof of Corollary 4.5 Follows immediately from the monotonicity of $(\alpha_0,...,\alpha_L)^\dagger$.

Recall that the conductance of P is defined by $\varphi(P) = \min_B \frac{F(B)}{\pi(B)}$.

Claim 4.6 [4] For all $1/3 \le s \le 1$, $\varphi(P(s)) \ge \frac{1}{6L}$.

Proof of Claim 4.6 We show that for any nonempty subset $B \subseteq \{0,...,L\}$, $\frac{F(B)}{\pi(B)} \geq \frac{1}{6L}$. We consider two cases.

Case 1. First, assume that $0 \in B$. Let k be the smallest such that $k \in B$ but $k+1 \notin B$. Then,

$$F(B) \overset{def}{=} \sum_{i \in B, j \notin B} \pi_i P_{ij} \geq \pi_k P(s)_{k,k+1} \overset{\alpha_k = \sqrt{\pi_k}}{\underset{\alpha_{k+1} = \sqrt{\pi_{k+1}}}{=}} \pi_k \cdot \frac{\sqrt{\pi_{k+1}}}{\mu\sqrt{\pi_k}} G(s)_{k,k+1} = \frac{\sqrt{\pi_k \pi_{k+1}}}{1-\lambda} G(s)_{k,k+1} \geq \frac{\pi_{k+1}}{1-\lambda} G(s)_{k,k+1}$$

where the last inequality follows from the monotonicity of π. Using the definition of G and the assumption that $s \geq 1/3$ we get that

$$G(s)_{k,k+1} = I_{k,k+1} - H(s)_{k,k+1} = 0 - \left((1-s)\left(H_{init}\right)_{k,k+1} + s\left(H_{final}\right)_{k,k+1}\right) = -\left(0 + s(-\frac{1}{2})\right) \geq 1/6 .$$

We also have $0 < 1-\lambda \leq 1$, where the second inequality follows from the fact that $H(s)$ is positive semidefinite, and the first follows from $\mu > 0$ which is previously deduced from the fact 4.2. Hence,

$$\frac{F(B)}{\pi(B)} \geq \frac{\pi_{k+1}}{6\pi(B)} .$$

By $\pi(B) \leq 1/2$ we have $\pi(\{k+1,...,L\}) \geq 1/2$. Together with $\pi(\{k+1,...,L\}) \leq L\pi_{k+1}$ we obtain $\pi_{k+1} \geq 1/(2L)$. This yields the desired bound $\frac{F(B)}{\pi(B)} \geq \frac{1}{6L}$.

Case 2. Now assume that $0 \notin B$ and let k be the smallest such that $k \notin B$ and $k+1 \in B$. It is easy to see that

$$\pi_k P(s)_{k,k+1} = \pi_{k+1} P(s)_{k+1,k}$$

since

$$\pi_k P(s)_{k,k+1} = \pi_k \cdot \frac{\sqrt{\pi_{k+1}}}{\mu\sqrt{\pi_k}} G(s)_{k,k+1} = \frac{\sqrt{\pi_k \pi_{k+1}}}{1-\lambda} G(s)_{k,k+1}$$

and

$$\pi_{k+1}P(s)_{k+1,k} = \pi_{k+1}\cdot\frac{\sqrt{\pi_k}}{\mu\sqrt{\pi_{k+1}}}G(s)_{k+1,k} = \frac{\sqrt{\pi_k\pi_{k+1}}}{1-\lambda}G(s)_{k+1,k}$$

and $G(s)$ is Hermitian.

Hence, using the same arguments as before we can see that in this case $\frac{F(B)}{\pi(B)} \geq \frac{\pi_{k+1}}{6\pi(B)}$ also takes place. Since $B \subseteq \{k+1,...,L\}$, we have $\pi(\{k+1,...,L\}) \geq \pi(B)$. Hence, $\pi_{k+1} \geq \pi(B)/L$. Again, this yields the bound $\frac{F(B)}{\pi(B)} \geq \frac{1}{6L}$.

By theorem 2.1, the eigenvalue gap of P is:

$$P \geq \frac{1}{2}\varphi(P)^2 = \frac{1}{2}\left(\min_B \frac{F(B)}{\pi(B)}\right)^2 \geq \frac{1}{2}\cdot\left(\frac{1}{6L}\right)^2 = \frac{1}{72L^2}.$$

Recalling the relation between spectral gap of $H(s)$ and the eigenvalue gap of $P(s)$,

$$g_{\min} \geq (1-\lambda)\frac{1}{72L^2}.$$

Finally, we notice that $\lambda \leq \langle\gamma_0|H(s)|\gamma_0\rangle = \frac{s}{2} \leq \frac{1}{2}$, and thus

$$g_{\min} \geq \frac{1}{144L^2}.$$

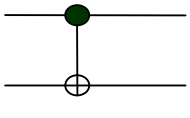
We conclude that the spectral gap of the time-dependent Hamiltonian $H(s)$ (precisely, $H(s)$ restricted to S_0), which describes the adiabatic quantum algorithm corresponding to the quantum circuit, is the inverse polynomial in terms of L – number of quantum gates.

4.3. Converting Quantum Circuit to Quantum Adiabatic Algorithm

Using the technique presented in Section 4.2, let us apply it to a simplest case. Consider a two-qubit, one-gate circuit:

CNOT: $|1\rangle$ $|1\rangle$

$|0\rangle$ $|1\rangle$



$$U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Recall that $H_{init} = H_{clockinit} + H_{input} + H_{clock}$ and $H_{final} = \frac{1}{2}\sum_{l=1}^{L} H_l + H_{input} + H_{clock}$, where L is the number of gates, and

$$H_{clockinit} = |1\rangle\langle 1|_1^c, \quad H_{clock} = \sum_{l=1}^{L} |01\rangle\langle 01|_{l,l+1}^c, \quad H_{input} = \sum_{i=1}^{n} |1\rangle\langle 1|_i \otimes |0\rangle\langle 0|_1^c$$

$$H_l = I \otimes |100\rangle\langle 100|_{l-1,l,l+1}^c - U_l \otimes |110\rangle\langle 100|_{l-1,l,l+1}^c - U_l^+ \otimes |100\rangle\langle 110|_{l-1,l,l+1}^c + I \otimes |110\rangle\langle 110|_{l-1,l,l+1}^c$$

In our case L=1, thus we need only 1 clock qubit. Therefore, there is no need in H_{clock} term. Also, although H_{input} refers to the case when input is $|0\rangle$, we can make it correct for our particular input $|10\rangle$: $H_{input} = (|00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 11|) \otimes |0\rangle\langle 0|^c$.

Other terms are:

$$H_{clockinit} = |1\rangle\langle 1|^c,$$

$$H_1 = I \otimes |0\rangle\langle 0|^c - U \otimes |1\rangle\langle 0|^c - U_l^+ \otimes |0\rangle\langle 1|^c + I \otimes |1\rangle\langle 1|^c.$$

Explicitly,

$$H_{init} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 0 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 1 \end{bmatrix}.$$

$$H_{final} = \frac{1}{2}H_1 + H_{input} = \frac{1}{2}\begin{bmatrix} 1 & -1 & & & & & & \\ -1 & 1 & & & & & & \\ & & 1 & -1 & & & & \\ & & -1 & 1 & & & & \\ & & & & 1 & & & -1 \\ & & & & & 1 & -1 & \\ & & & & & -1 & 1 & \\ & & & & -1 & & & 1 \end{bmatrix} + \begin{bmatrix} 1 & & & & & & & \\ & 0 & & & & & & \\ & & 1 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 1 & \\ & & & & & & & 0 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} 3 & -1 & & & & & & \\ -1 & 1 & & & & & & \\ & & 3 & -1 & & & & \\ & & -1 & 1 & & & & \\ & & & & 1 & & & -1 \\ & & & & & 1 & -1 & \\ & & & & & -1 & 3 & \\ & & & & -1 & & & 1 \end{bmatrix}$$

It is obvious that the ground state of H_{init} is $|10\rangle \otimes |0\rangle^c$. The ground state of H_{final} has to be a history state, i.e. $|\eta\rangle = \frac{1}{\sqrt{2}}\left(|10\rangle \otimes |0\rangle^c + |11\rangle \otimes |1\rangle^c\right)$. To check it, we do the following:

$$H_{final}|\eta\rangle = \frac{1}{2\sqrt{2}}(1 - 1 + (-1) + 1) = 0.$$

4.4. Speedup by a Factor

As we have seen above, the correctness of the adiabatic quantum algorithm depends on the size of the spectral gap of the Hamiltonian: the larger it is the faster algorithm is. Therefore, one might be interested in how this spectral gap can be enlarged.

One such technique was mentioned in Section 3.3.2, suggesting to use different paths like (3.12) to avoid regions where the gap is too small.

Another idea, a more intuitive one, is to multiply the Hamiltonian by a factor, thus increasing the spectral gap by that factor. The following explains our idea in more detail.

Suppose we have the initial Hamiltonian H_{init} that is easy to construct, and the final Hamiltonian H_{final}, whose ground state we are aimed to find. We construct the time-dependent Hamiltonian using linear interpolation:

$$H(s) = sH_{init} + (1-s)H_{final}, \text{ where } s = \frac{t}{T}.$$

T is the running time of the experiment and is determined by the spectral gap $g_{\min}$ of the $H(s)$:

$$T = O\left(\frac{\|H'(s)\|}{g_{\min}^2}\right).$$

If we multiply $H(s)$ by some constant c, then we will have an experiment with the initial Hamiltonian cH_{init} and final Hamiltonian cH_{final}:

$$\tilde{H}(s) = cH(s) = s(cH_{init}) + (1-s)(cH_{final}).$$

Note that the eigenstates of $\tilde{H}(s)$ and $H(s)$ are the same:

- if $H(s)\varphi(s) = \lambda\varphi(s)$ then $\tilde{H}(s)\varphi(s) = cH(s)\varphi(s) = c\lambda\varphi(s) = \tilde{\lambda}\varphi(s)$ and
- if $\tilde{H}(s)\varphi(s) = \tilde{\lambda}\varphi(s)$ then $\tilde{H}(s)\varphi(s) = cH(s)\varphi(s) = \tilde{\lambda}\varphi(s) \Rightarrow H(s)\varphi(s) = \frac{\tilde{\lambda}}{c}\varphi(s)$.

Therefore the ground states are the same. But, as it can be seen from the equations above, the eigenvalues differ by the factor c. Therefore, the spectral gap $\tilde{g}_{\min}$ of $\tilde{H}(s) = cH(s)$ will be c times that of $H(s)$. Plugging $\tilde{H}(s)$ and $\tilde{g}_{\min}$ to the equation for the time of the

experiment we end up with $\tilde{T} = T / c$. Therefore, the evolution of the system with the Hamiltonian $\tilde{H}(s) = cH(s) = cH(\frac{t}{T})$ will end up with the same result (the same ground state) but requiring time which is *c* times shorter.

So, by multiplying any Hamiltonian by a constant we may increase the spectral gap, thus decreasing the running time of the experiment.

Will this technique give any advantages when simulating the new "fast" system?

As it was explained in Section 4.1, simulation of the quantum system with the time-dependent Hamiltonian is performed by approximating the unitary transformation induced by that Hamiltonian by a polynomial (in terms of the size of the input) number of quantum gates - unitary transformations - each of which acts on a few qubits [3]. Each of such "small" unitary transformations in fact corresponds to a certain time slice of the overall evolution of the system. If the number of time slices (and therefore the number of quantum gates is) is M, and the evolution time is T, then the duration of each time slice is $\Delta = T / M$. The approximation will simulate time evolution if Δ satisfies (4.6):

$$\Delta \left\| H_{init} - H_{final} \right\| << 1 .$$

Therefore, the size of the time interval Δ is in inverse dependence to the "distance" between initial and final Hamiltonians of the adiabatic quantum algorithm. In the "fast" system, the "distance" between the initial and final Hamiltonians is *c* times bigger: $\left\| \tilde{H}_{init} - \tilde{H}_{final} \right\| = \left\| cH_{init} - cH_{final} \right\| = c \left\| H_{init} - H_{final} \right\|$, and it follows that size of the time interval $\tilde{\Delta}$, which is necessary to define for the simulation of the "fast" system, has to be *c* times shorter: $\tilde{\Delta} = \Delta / c$.

Now we can calculate number of simulation steps of the "fast" system:

$$\tilde{M} = \tilde{T} / \tilde{\Delta} = \frac{T}{c} / \frac{\Delta}{c} = T / \Delta = M .$$

We conclude that since the same number of simulation steps is required for simulation of our system and its "fast" version, there is no advantage of the "fast" system in this sense.

5. FLOW CONTROL MANIFESTATIONS IN ADIABATIC QUANTUM PROGRAMS

After comprehensive discussion of adiabatic quantum computation and adiabatic quantum algorithms, we wonder how this new paradigm can be used in developing a new programming technique, namely, adiabatic quantum programming (AQP). In this section, we discuss two familiar programming structures – loops and conditional branching – from the perspective of adiabatic quantum programming, to illustrate the fundamentally different "feel" of the framework. As will be elaborated in the discussion, this is basically an introduction about what *not* to do during adiabatic program design.

5.1. Loops in AQP

Let us consider how loop structures look in the framework of AQP. The simplest loop format is:

```
DO n times {loop_body}
```

In a standard quantum program, this kind of loop would be implemented as a quantum circuit containing the sub-circuit which corresponds to {loop_body} *n* consecutive times. In an AQP, this will just correspond to a Hamiltonian. In the examples below, Hamiltonians of the body and its loop are constructed and compared.

Example 5.1 Consider the circuit shown in Figure 5.1, and convert it to an AQP. The matrix representation of the π/2-rotation around the *y*-axis is:

$$R_y(\pi/2) = \cos(\pi/4) - i\sigma_y \sin(\pi/4) = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix} \tag{5.1}$$

According to Section 4.2,

$$H_{clockinit} = |1\rangle\langle 1|^c,$$

$$H_{input} = |1\rangle\langle 1| \otimes |0\rangle\langle 0|^c,$$

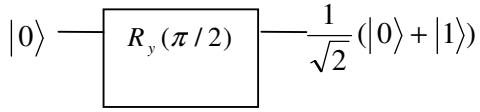


Figure 5. 1 Simple quantum circuit involving one qubit and one quantum gate corresponding to the π/2-rotation around the y-axis.

H_{clock} may be ignored and

$$H_1 = I \otimes |0\rangle\langle 0|^c - R_y(\pi/2) \otimes |1\rangle\langle 0|^c - R_y^+(\pi/2) \otimes |0\rangle\langle 1|^c + I \otimes |1\rangle\langle 1|^c .$$

Therefore, $H_{init} = \begin{pmatrix} 0&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{pmatrix}$, $H_{final} = \frac{1}{2\sqrt{2}} \begin{pmatrix} \sqrt{2}&-1&0&-1 \\ -1&\sqrt{2}&1&0 \\ 0&1&3\sqrt{2}&-1 \\ -1&0&-1&\sqrt{2} \end{pmatrix}$ and

$$H(s) = (1-s)H_{init} + sH_{final} = \frac{1}{4} \begin{pmatrix} 2s & -\sqrt{2}s & 0 & -\sqrt{2}s \\ -\sqrt{2}s & 4-2s & \sqrt{2}s & 0 \\ 0 & \sqrt{2}s & 4+2s & -\sqrt{2}s \\ -\sqrt{2}s & 0 & -\sqrt{2}s & 4-2s \end{pmatrix} \tag{5.2}$$

Example 5.2. Consider the "loop" in Figure 5.2, and convert it to an AQP. Again,

$$H_{clockinit} = |1\rangle\langle 1|_1^c ,$$

$$H_{input} = |1\rangle\langle 1| \otimes |0\rangle\langle 0|_1^c ,$$

$$H_{clock} = |01\rangle\langle 01|^c ,$$

$$H_1 = I \otimes |00\rangle\langle 00|^c - R_y(\pi/2) \otimes |10\rangle\langle 00|^c - R_y^+(\pi/2) \otimes |00\rangle\langle 10|^c + I \otimes |10\rangle\langle 10|^c ,$$

$$H_2 = I \otimes |10\rangle\langle 10|^c - R_y(\pi/2) \otimes |11\rangle\langle 10|^c - R_y^+(\pi/2) \otimes |10\rangle\langle 11|^c + I \otimes |11\rangle\langle 11|^c .$$

Therefore,

$$H_{init} = \begin{pmatrix} 0 & \cdots & & & & & & 0 \\ & 1 & 0 & \cdots & & & & 0 \\ \vdots & 0 & 1 & 0 & \cdots & & & 0 \\ & \vdots & 0 & 1 & 0 & \cdots & & 0 \\ & & \vdots & 0 & 1 & 0 & \cdots & 0 \\ & & & \vdots & 0 & 2 & 0 & 0 \\ & & & & \vdots & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \quad H_{final} = \frac{1}{2\sqrt{2}} \begin{pmatrix} \sqrt{2}&0&-1&0&0&0&-1&0 \\ 0&2\sqrt{2}&0&0&0&0&0&0 \\ -1&0&2\sqrt{2}&-1&1&0&0&-1 \\ 0&0&-1&\sqrt{2}&0&0&1&0 \\ 0&0&1&0&3\sqrt{2}&0&-1&0 \\ 0&0&0&0&0&4\sqrt{2}&0&0 \\ -1&0&0&1&-1&0&2\sqrt{2}&-1 \\ 0&0&-1&0&0&0&-1&\sqrt{2} \end{pmatrix} \text{ and}$$

$$H_{loop}(s) = \frac{1}{4} \begin{pmatrix} 2s & 0 & -\sqrt{2}s & 0 & 0 & 0 & -\sqrt{2}s & 0 \\ 0 & 4 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\sqrt{2}s & 0 & 4 & -\sqrt{2}s & \sqrt{2}s & 0 & 0 & -\sqrt{2}s \\ 0 & 0 & -\sqrt{2}s & 4-2s & 0 & 0 & \sqrt{2}s & 0 \\ 0 & 0 & \sqrt{2}s & 0 & 4+2s & 0 & -\sqrt{2}s & 0 \\ 0 & 0 & 0 & 0 & 0 & 4+4s & 0 & 0 \\ -\sqrt{2}s & 0 & 0 & \sqrt{2}s & -\sqrt{2}s & 0 & 4 & -\sqrt{2}s \\ 0 & 0 & -\sqrt{2}s & 0 & 0 & 0 & -\sqrt{2}s & 4-2s \end{pmatrix}. \tag{5.3}$$

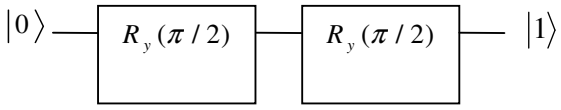


Figure 5. 2. Simple quantum "loop" circuit involving one qubit and two consecutive quantum gates each corresponding to the π/2-rotation around the y-axis.

Compared to (5.2), (5.3) has a larger dimension, since more gates are involved in the computation. But what these examples really show is that one really should not be tempted to automatically build one's AQPs by conversion from the circuit model. The obvious equality of a single $R_y(\pi/4)$ (i.e. "not") gate to Figure 5.2 makes this point clear: Like in all other kinds of programming, we should develop an insight about the "natural" way to write good programs in this new language. As mentioned in Introduction, this will probably require future programming students to learn linear algebra and related mathematical topics as thoroughly as today's physics students as a prerequisite.

Consider the following "hybrid" classical/quantum loop:

```
DO n times {loop_body; measure; check}
```

This loop describes the act of running of an AQP *n* times consecutively, checking the output for correctness after each run. (This can clearly be deemed efficient only for problems in NP). In practice, due to the probabilistic nature of AQPs, they have to be run several times in this fashion to amplify the probability of success. This causes an *n*-fold increase in the running time of the originally designed AQP. However, assuming that there are resources available to increase the number of qubits participating in the computation *n* times, the *n* copies of the AQP can be run simultaneously, keeping the original running time. The above-mentioned loop can be realized, therefore, by an AQP with the new Hamiltonian obtained by taking the tensor product of the original Hamiltonian by itself *n* times:

$$\text{AQPPll}\left(\left|\psi_{init}\right\rangle^{\otimes poly(n)}, H^{\otimes poly(n)}, T\right) : \left|\psi_{init}\right\rangle^{\otimes poly(n)} \rightarrow \{0,1\}^{n \cdot poly(n)} \qquad (5.4)$$

5.2. Conditional Branching

Classical programs allow the conditional execution of code in dependence on the content of a Boolean variable, i.e. conditional branching:

```
IF condition THEN branch1 ELSE branch2
```

In standard quantum computation, pure quantum conditional branching can be realized using controlled gates. For instance, consider the conditional π/2-rotation shown in Figure 5.3. The unitary matrix of that quantum circuit is:

$$U = \begin{pmatrix} I & 0 \\ 0 & R_y(\frac{\pi}{2}) \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} \sqrt{2} & 0 & 0 & 0 \\ 0 & \sqrt{2} & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

Following the conversion procedure from Section 4.2, for this particular circuit,

$$|\psi_{init}\rangle = |100\rangle,$$

$$H_{input} = (|00\rangle\langle 00| + |01\rangle\langle 01| + |11\rangle\langle 11|) \otimes |0\rangle\langle 0|^c,$$

$$H_{clockinit} = |1\rangle\langle 1|^c,$$

$$H_1 = I \otimes |0\rangle\langle 0|^c - U \otimes |1\rangle\langle 0|^c - U^+ \otimes |0\rangle\langle 1|^c + I \otimes |1\rangle\langle 1|^c$$

and we ignore H_{clock}. Explicitly,

$$H_{init} = \begin{pmatrix} 1&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1 \end{pmatrix}, \; H_{final} = \frac{1}{4} \begin{pmatrix} 6&-2&0&0&0&0&0&0 \\ -2&2&0&0&0&0&0&0 \\ 0&0&6&-2&0&0&0&0 \\ 0&0&-2&2&0&0&0&0 \\ 0&0&0&0&2&-\sqrt{2}&0&-\sqrt{2} \\ 0&0&0&0&-\sqrt{2}&2&\sqrt{2}&0 \\ 0&0&0&0&0&\sqrt{2}&6&-\sqrt{2} \\ 0&0&0&0&-\sqrt{2}&0&-\sqrt{2}&2 \end{pmatrix}$$

and

$$H_{cond}(s) = \frac{1}{4} \begin{pmatrix} 4+2s&-2s&0&0&0&0&0&0 \\ -2s&4-2s&0&0&0&0&0&0 \\ 0&0&4+2s&-2s&0&0&0&0 \\ 0&0&-2s&4-2s&0&0&0&0 \\ 0&0&0&0&2s&-\sqrt{2}s&0&-\sqrt{2}s \\ 0&0&0&0&-\sqrt{2}s&4-2s&\sqrt{2}s&0 \\ 0&0&0&0&0&\sqrt{2}s&4+2s&-\sqrt{2}s \\ 0&0&0&0&-\sqrt{2}s&0&-\sqrt{2}s&4-2s \end{pmatrix}. \tag{5.5}$$

Two remarks are in order: First, note that, although we give the Hamiltonians explicitly for pedagogical purposes, this is clearly not possible (nor required) when a sizable number of bits constitute our system. The program is supposed to be describable implicitly, and doing this without knowing the final ground state is what requires ingenuity

in this form of programming Second, we see once again that automatic conversion from a circuit model does not yield obviously identifiable patterns corresponding to our intuitions about an if statement in the Hamiltonian. Once again, a “native” design approach is called for.

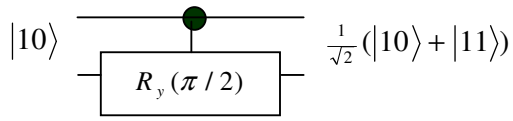


Figure 5. 3. Simple quantum conditional circuit involving one control and one target qubit and one controlled gate corresponding to the controlled π/2-rotation around the y-axis.

6. CONCLUSIONS

In this work we aimed to develop an understanding of relatively new paradigm for design of quantum algorithms – the model of adiabatic quantum programming. We have reviewed the continuous-time adiabatic quantum algorithm and comprehensively examined examples of application of the algorithm to several problems, demonstrating a different way of thinking about algorithms and of developing programs.

We also discussed the equivalency of standard quantum computation in terms of quantum gates and adiabatic quantum computation in terms of Hamiltonians. Nevertheless, there are certain qualitative differences between two models. The major one is that standard quantum computation in terms of gates constitutes a sequential process of applying one gate after the other. Time is measured as the number of operations performed during the computation. On the contrary, adiabatic quantum computation is a single application of a Hamiltonian, which is continuous in real time. Evolution time plays a very important role, since the result of the adiabatic quantum program depends on the duration of the program.

Since adiabatic quantum computation was shown to have universal computational power, it is only natural to ask how a programming language suitable for this paradigm would look like. We have started to examine this issue by "adiabaticizing" some well-known flow control structures using the technique presented in [4]. As expected, such a direct conversion from programs written in a different paradigm neither yields efficient adiabatic programs, nor seems to be helpful about developing an insight about how to design them. The correct approach would indeed probably rely on the mathematical vocabulary and tools that have already been developed for dealing with these kinds of matrices, [4] and the educational requirements that this imposes on prospective adiabatic programmers are likely to have important consequences for the design of future computer science curricula.

7. REFERENCES

1. Feynman, R., *Simulating Physics with Computers*, International Journal of Theoretical Physics 21: 467–488, 1982.

2. Shor, P., *Algorithms for Quantum Computation: Discrete Logarithms and Factoring,* arXiv: quant-ph/9508027, 1994.

3. Farhi, E., J. Goldstone, S. Gutmann, and M. Sipser, *Quantum Computation by Adiabatic Evolution*, arXive: quant-ph/0001106, 2000.

4. Aharonov, D., W. Van Dam, J. Kempe, Z. Landau, S. Lloyd and O. Regev, *Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation*, arXiv: quant-ph/0405098, 2004.

5. Shor, P., *Progress in Quantum Algorithms*, Quantum Information Processing, Vol. 3, pp. 5-13, 2004.

6. Steffen, M., W. van Dam, T. Hogg, G. Breyta and I. Chuang, *Experimental Implementation of an Adiabatic Quantum Optimization Algorithm*, arXiv: quant-ph/0302057, 2003.

7. Knill, E., *Conventions for Quantum Pseudocode,* LANL report LAUR-96-2724, 1996.

8. Ömer, B., *A Procedural Formalism for Quantum Computing*, Master thesis, Technical University of Vienna, 1998.

9. Zuliani, P., *Quantum Programming,* PhD Thesis, University of Oxford, 2001.

10. Bettelli, S., *Toward an Architecture for Quantum Programming*, PhD Thesis, Universita di Trento, 2002.

11. Altenkirch, T., *A Functional Quantum Programming Language*, arXiv: quant-ph/0409065, 2005.

12. Selinger, P., *Towards a Quantum Programming Language*, Mathematical Structures in Computer Science 14(4): 527-586, 2004.

13. Van Tonder, A., *A Lambda Calculus for Quantum Computation*, arXiv: quant-ph/0307150, 2004.

14. Farhi, E., J. Goldstone, S. Gutmann, J. Lapan, A. Lundgren and D. Preda, *A Quantum Adiabatic Evolution Algorithm Applied to Random Instances of an NP-Complete Problem*, arXiv: quant-ph/0104129, 2001.

15. Farhi, E., J. Goldstone and S. Gutmann, *Quantum Adiabatic Evolution Algorithms with Different Paths*, arXiv:quant-ph/0208125 v1, 2002.

16. Sipser, M., *Introduction to the Theory of Computation*, PWS Publishing Company, Boston, 1997.

17. Ambainis, A. and O. Regev, *An Elementary Proof of the Quantum Adiabatic Theorem*, arXive: quant-ph/0411152, 2004.

18. Selinger, P., *A Brief Survey of Quantum Programming Languages*, Proceedings of the 7th International Symposium on Functional and Logic Programming, Nara, Japan, Springer LNCS 2998, pp. 1-6, 2004.

19. Ruskai, M. B., *Comments on Adiabatic Quantum Algorithms*, arXiv: quant-ph/0203127 v3, 2002.

20. Nielsen, M. A. and I. L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2000.

21. Kempe, J., *Quantum Random Walks: an Introductory Overview*, Contemporary Physics, vol. 44, no. 4, pp. 307 – 327, 2003.

22. Lovasz, L., *Random Walks on Graphs: A Survey*, Combinatorics, Paul Erdos is eighty, Vol. 2, Keszthely, 1993.