Title
On Quantum Search, Experts and Geometry

Author
Drezgich, Milosh

Publication Date
2010

Peer reviewed|Thesis/dissertation

On Quantum Search, Experts and Geometry

by

Milosh Drezgich

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Engineering - Electrical Engineering and Computer Science

in the

GRADUATE DIVISION
of the
UNIVERSITY OF CALIFORNIA, BERKELEY

Committee in charge:
Professor Shankar Sastry, Chair
Professor Umesh Vazirani
Professor Michael Crommie

Spring 2011

On Quantum Search, Experts and Geometry

Abstract

On Quantum Search, Experts and Geometry

by

Milosh Drezgich
Doctor of Philosophy in Engineering - Electrical Engineering and Computer Science

University of California, Berkeley

Professor Shankar Sastry, Chair

The problem of unstructured search plays the central role in our current understanding of the computational power of quantum computers. Improvement in the efficiency of solving unstructured search problem has an immediate consequence in the improvement in solving NP-complete problems. We introduce the new framework of natural continuous time quantum search algorithms, that in contrast to the adiabatic quantum algorithms, require neither the ground state initialization nor the adiabatic change of the Hamiltonian parameters. We moreover provide the concrete examples of transliteration from the unique marked element unstructured search problem into the particular class of quantum Hamiltonians, that facilitate the search in quantum continuous constant time. Since it is not clear how to implement that class of Hamiltonians one can either consider this result as a step toward proving that NP is a subset of BQP or as an indication that, that class of Hamiltonians is NP-hard to implement.

Multiplicative weights update rule has been used in a few different fields as the underlying algorithmic structure. In its two different forms, vector and matrix, multiplicative update method provided a surprising simplicity and promised a small performance regret. We derive a slightly more general bound for the cumulative matrix multiplicative weights algorithm and introduce the iterative (streaming) matrix multiplicative weights algorithm with the same computational complexity and regret bound. In particular we also define the iterative Hadamard updates, matrix multiplicative updates algorithm, with the improved computational complexity for nonnegative games, from $O(n^3)$ to $O(n^2)$, and the same regret bound.

Furthermore, we address the following question:"What is the minimal size

quantum circuit required to exactly implement a specified $n$qubit unitary operation U, without the use of ancilla qubits?" Nielsen proved that a lower bound on the minimal size circuit is provided by the length of the geodesic between the identity I and U, where the length is defined by a suitable Finsler metric on $SU(2^n)$. We prove that the minimum circuit size that simulates U is in linear relation with the geodesic length and simulation parameters, for the given Finsler structure F. As a corollary we prove the highest lower bound and the lowest upper bound for the standard simulation technique, that show that by standard simulation one can not expect a better then n^2 times improvement in the upper bound over the result from Nielsen, Dowling, Gu and Doherty [4]. Moreover, our equivalence result can be applied to the arbitrary path on the manifold including the one that is generated adiabatically.

Finally we investigate the n-dimensional hypercube quantum random walk (QRW) as a particularly appealing example of a quantum walk because it has a natural implementation on a register on n qubits. However, any real implementation will encounter decoherence effects due to interactions with uncontrollable degrees of freedom. We present a complete characterization of the mixing properties of the hypercube QRW under a physically relevant Markovian decoherence model. In the local decoherence model considered the non-unitary dynamics are modeled as a sum of projections on individual qubits to an arbitrary direction on the Bloch sphere. We prove that there is always a classical asymptotic mixing in this model and specify the conditions under which instantaneous mixing *always* exists. We show that the latter mixing property, as well as the classical mixing time, depend heavily on the exact environmental interaction and its strength. Therefore, algorithmic applications of the QRW on the hypercube, if they intend to employ mixing properties, need to consider both the walk dynamics and the precise decoherence model.

To my teachers in life and science,
past and present,
some of who are in the Heaven,
some of who are on the Earth...

Contents

List of Figures

List of Tables

Acknowledgments

I can responsibly claim that this dissertation would not see the light of the day without professor Shankar Sastry and professor Umesh Vazirani.

I owe a grand gratitude to Shankar Sastry for his constant encouragement, support, advices, fine mathematical lessons and enlightenments and for the freedom and the privilege to pursue the ideas that I found very important and interesting and his friendship.

I owe a grand gratitude to Umesh Vazirani for his advices in countless meetings, vibrant discussions, careful nurturing, development of intuition and sense for choosing research topics, explanation of concepts, "how to solve", "the way to think about" lessons and friendship.

I am grateful to Christos Papadimitriou, for the early encouragement, positive attitude toward my research and insightful comments; to Luca Trevisan, Alistair Sinclair, Satish Rao, Michael Crommie, Alex Pines, Michael Hutchings and Nikolai Reshetikhin for the great explanations, hospitality and out of schedule meetings.

I am also grateful to Yasmina Vuyich for her constant enthusiasm, encouragement, support and the help I would be missing time to explain here.

I owe a grandiose gratitude to my Maria for her love, patience and support.

Thanks to my family for their constant encouragement.

Thanks also to the fellow researchers for the valuable remarks even though they work in the different field Dushan Stipanovich an Vukman Chovich.

I also wish to thank dear colleagues for the important and interesting discussions and also for making the life in Berkeley a special event: Zeph Landau, Thomas Vidick, Lorenzo Orecchia, James Cook, Anand Bhaskar, George Pierrakos, Falk Unger, Robert Spalek, Ben Reichardt, Dorit Aharonov and Julia Kempe. Thanks to Berkeley staff for their kindness and support in the administrative conundrums: Ruth Gjerde, La Shana Polaris, Dana Jantz, Delia Umel, Cassandra Hill, sMaria Jauregui, Jessica Gamble, Gladys Khoury and Tracey Richards.

Finally I wish to thank NSF for the support under the ITR Grant No. EIA-0205641

Chapter 1

On NP vs. BQP

1.1 Introduction

It would be hard to find contemporary scientific discipline that would not take advantage if the current programmable machines were able to solve the generalized search problem more efficiently. Perhaps due to the fact that finding a single marked element in an array of N elements, was, and still is, one of the central problems of the modern theoretical and practical computer science, albeit for different reason.

The model of computation that is governed by the quantum mechanical process has been heavily studied in the last decade. Most computer scientists today do believe that if the information is represented by the vector in two dimensional complex plane $\mathbb{C}^2$ instead of the classical binary value, computational power of such computer would be substantially different. Moreover, most of the evidence that we have today, tell us that the quantum computers typically offer a quadratic speed up over the classical counterparts. However, there are also examples of computational problems that can be solved exponentially faster on the quantum computer and unfortunately there are problems for which the quantum computers do not offer any speed up at all, leaving the question on the power of quantum computation with the inconclusive answer.

The generalized search problem that we study has a simple structure and a fundamental importance in any study that aims to determine the relation between the most fundamental complexity classes NP and BQP. This problem is NP-hard and it is a generalization of a $3SAT$ that is a NP-complete problem, for which the existence of a subexponential algorithm is currently not known. The problem is as follows.

The generalized search problem is the following. In the list of N one bit Boolean elements only one element, called the marked element, has the value zero while all other elements have the value one. The problem is to find the index of the marked element in the list.

For an array of N identical items out of which only one is marked, best classical algorithms would spend $\mathcal{O}(N)$ time to find the marked item. Considerations of the computational model that is very different than classical, namely quantum computation model, changed this picture considerably providing the quadratic speed up. In 1994 Bennett, Bernstein, Brassard and Vazirani, proved that, relative to a random oracle, generalized search problem can not be solved faster then $\Omega\left(\sqrt{N}\right)$, and with respect to that it is unlikely that NP is in BQP, however this question as of 2010 is yet unresolved. Numerous discrete time quantum algorithm have all proven to obey this bound. The similar situation is with the continuous quantum algorithms, that have not been able to facilitate a miraculous solution of this problem that seem to be extremely hard for the classical computers. In recent years the main effort have been dedicated to the study of the adiabatic algorithms and the difference in its efficiency with the respect to its initial condition.

Interestingly, introducing small amount of nonlinearity in quantum mechanics, possibility of closed timelike curves, or superluminal signaling all open the possibility of solving NP complete problems in polynomial time, and thus generalized search problem as well, however such considerations are not generally considered as viable.

In this chapter we aim to provide the additional framework for the study of the relation between NP and BQP. We introduce the notion of the *natural continuous quantum algorithms* that does not follow into the framework of the adiabatic algorithms.

Natural continuous quantum algorithms that we introduce rely on the following assumptions:

- that there is no changes to the any of the quantum mechanical postulates;
- there is the usual transliteration from the classical problem to the quantum Hamiltonian problem, i.e. the transliteration from classical generalized search to quantum Hamiltonian does not introduce the additional structure to the problem;

The two main properties of natural continuous quantum algorithms, that distinguish them from the adiabatic algorithm are that they either:

- do not require the study of the algorithm with respect to the finite set of initial conditions, as the success probability of the algorithm is remains high invariably with respect to the initial condition;
- do not require adiabatic, i.e. adaptive change, of the parameters, that regulate the norm of the system initial and final Hamiltonian, with respect to the smallest eigenvalue gap of the instantaneous system Hamiltonian.

Unfortunately, the only speed up result that follows, in the framework of natural algorithms must assume the quantum computer model with the ability to implement a simply describable Hamiltonian, in the form $|w\rangle\langle w| + |s\rangle\langle s|$, where $|s\rangle$ is an implicitely defined state and $|w\rangle$ is the all uniform state in the computational basis. We do not know how to implement this Hamiltonian and therefore either one can consider this as a step toward proving that NP is a subset of BQP or that the class of Hamiltonians in that is NP-hard to implement. Accordingly, we claim *no* asymptotic speed up for our algorithms and leave the definite answer about the relation between NP and BQP open for the future study. Nevertheless, we believe that the above mentioned properties of our algorithms might prove themselves as valuable theoretical grains, the future researchers of this topic might want to know about.

1.2 The Quantum Information Preliminaries

A state of quantum bit, i.e. qubit, is essentially a vector in the two dimensional complex space $\mathbb{C}^2$, rather then the element from the set $\mathbb{Z}_2 = \{0, 1\}$. Most commonly chosen orthonormal basis for $\mathbb{C}^2$ is, so called computational basis, or z-basis, is denoted as $|0\rangle_z = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle_z = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. In general, the state of the qubit can be a unitary vector that points in any direction in $\mathbb{C}^2$ space, and in that case its state is denoted as:

$$|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1.1}$$

where normalization condition is $\alpha^2 + \beta^2 = 1$, and $\alpha, \beta \in \mathbb{C}$.

The postulates of quantum mechanics determine, that the state of the quantum system with n qubits, that is in general a vector $|\Psi(t)\rangle \in \mathbb{C}^{2^n}$. Time

changes of $|\Psi(t)\rangle$ are determined by a unitary transformation, i.e. a unitary matrix, that is the solution of the Schroedinger equation:

$$i\frac{d}{dt}|\Psi(t)\rangle = H(t)|\Psi(t)\rangle \tag{1.2}$$

with the initial state $|\Psi(0)\rangle$.

It is important to note that the composite state of the system, i.e. the state of the n–qubit register is a tensor product of individual qubits and therefore it belongs to space $\mathbb{C} \otimes \mathbb{C} \otimes ... \otimes \mathbb{C} \equiv \mathbb{C}^{2^n}$. The usual notation is as follows; the state of n–qubit register can be initialized to one of the 2^n basis states, i.e. zero state denoted as $|0\rangle^{\otimes n} \equiv |00...0\rangle \equiv |0\rangle \otimes |0\rangle \otimes ... \otimes |0\rangle$, simply as $|0\rangle \in \mathbb{C}^{2^n}$, or in some of the other $2^n - 1$ basis states of the register, for example, $|3\rangle \equiv |0...011\rangle$, and so on. Therefore, following description (1.1) a state at arbitrary time t is $|\Psi(t)\rangle \in \mathbb{C}^{2^n}$, is determined with 2^n time varying complex coefficients, i.e. one for each vector in the orthonormal basis $|0\rangle, |1\rangle, |2\rangle..|2^n\rangle \equiv |N-1\rangle$.

In the equation (1.2) $H(t)$ is $2^n \times 2^n$ Hermitian matrix, is called the system Hamiltonian since it determines the energy of state basis vectors. This is an important observable as the spectrum of the Hamiltonian is the set of possible outcomes when one measures the total energy of the system. For example, as we will see below, we will be particularly interested in measuring the lowest eigenvalue eigenvector, called the ground state, of the system Hamiltonian. That measurement will reveal the Hamiltonian eigenvector that encodes with high probability the solution of the hard computational problem.

In the case when the system Hamiltonian is time invariant $H(t) \equiv H$, the solution of the Schroedinger equation is $|\Psi(t)\rangle = exp(-iHt)|\Psi(0)\rangle = U(t)|\Psi(0)\rangle$. Here $U(t)$ is a unitary matrix, or a unitary rotation that rotates initial state $|\Psi(0)\rangle \in \mathbb{C}^{2^n}$ to the final state at time t, $|\Psi(t)\rangle \in \mathbb{C}^{2^n}$.

The "gap". By the *gap* we will always refer to the gap between the lowest and the second lowest eigenvalue of the matrix. The notation for the matrix A that $A \geq a$ should be understood as $A - aI \geq 0$. Clearly if $A \geq 0$ all eigenvalues of A are greater or equal then 0.

1.3 The Quantum Adiabatic Theorem Preliminaries

The Hamiltonian $H(t)$ of a physical system gives a complete specification of the time evolution of the system state $|\psi(t)\rangle$. The differential equation that

describes the time evolution of the isolated quantum system is the well-known Schroedinger equation:

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

Again, a Hamiltonian is described by a Hermitian matrix, whose eigenvectors represent the eigenstates of the system. The corresponding eigenvalues refer to the different energies of the eigenstates. The state (eigenvector) with the lowest energy (eigenvalue) is called the "ground state" of the system. Moreover, the Schroedinger equation can also be described with reference to the unitary transformation U that is defined by the Hamiltonian $H(t)$. Substituting the solution of (1.2) back to the equation (1.2) we get:

$$\frac{d}{dt}U(t) = -iH(t)U(t),$$

with the initial condition $U(0) = I$.

We usually say that the Hamiltonian evolution from $H(0)$ to $H(T)$ *induces* the unitary transformation $U(T)$. Clearly, only the evolution of a system with a time-independent Hamiltonian H is easy to express by the exponential $U(T) = \mathrm{e}^{-iTH}$.

Finding the approximate solutions for Hamiltonians that vary in time is one of the core tasks in quantum physics. One of the most important cases of such approximation of a time-dependent case is described by the adiabatic evolution of an isolated quantum mechanical system.

The quantum adiabatic theorem states that a physical system that is initially in its ground state, tends to stay in this lowest energy state, provided that the Hamiltonian of the system is changed at the rate that sufficiently slow so that it satisfies the condition stated below [27].

The quantitative version of the adiabatic theorem gives the following specific upper bound on the running time that is required so that the ground state eigenvector is in fact very closely approximating the true ground state that would exist without the adiabatic approximation. It is convenient to parameterize the time-dependent Hamiltonian by $H(s)$ for $0 \leq s \leq 1$ and its ground state by $\phi(s)$. Our goal here is to gradually transform the applied Hamiltonian from $H(0)$ to $H(1)$ such that the initial state $\psi(0) = \phi(0)$ evolves to a close approximation $\psi(1) \approx \phi(1)$ of the ground state of $H(1)$. For that purpose the evolution time $\tau(s)$ is usually introduced, which determines the rate at which the Hamiltonian is modified as a function of s. Now the

Schroedinger equation in s equals:

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

The crucial quantity for this transformation is the gap between the two lowest eigenvalues of $H(s)$, which is denoted by $g(s)$. It can be shown that the evolution time τ such that:

$$\tau(s) \gg \frac{\max_{0\leq s\leq 1}\left|\left\langle\xi_1\right|\frac{d}{ds}H\left(s\right)\left|\xi_0\right\rangle\right|}{g(s)^2} \tag{1.3}$$

is "sufficiently slow" for the quantum system to adiabatically evolve from $\phi(0)$ to $\phi(1)$. In equation (1.3) $|\xi_0\rangle$, $|\xi_1\rangle$ are the ground state and the first excited state of $H\left(s\right)$.

As a result, the total evolution time (delay) of this process will be of the order $\int_{s=0}^{1}\tau(s)ds$. For most Hamiltonians it is too difficult to determine the gap $g(s)$ for every s. Therefore the common procedure is to look at the *minimum gap* $g_{\min} := \min_s g(s)$ and the maximum $\Delta_{\max} := \max_s\left|\left\langle\xi_1\right|\frac{d}{ds}H\left(s\right)\left|\xi_0\right\rangle\right|$, to obtain the adiabatic evolution with the constant delay factor that is: $\tau(s) = \tau_c \in O(\Delta_{\max}/g_{\min}^{-2})$.

1.4 Exact Solution of the Schroedinger Equation

The general proof of the adiabatic theorem relies on the bounds on the transition amplitudes away from the ground state. In this section we look at the transition amplitudes that determine the instantaneous ground state of the system that evolves under the time varying Hamiltonian under the Schroedinger equation.

We can write the state of the instantaneous state of the quantum computer as:

$$|\psi(t)\rangle = \sum_n c_n\left(t\right)e^{-i\int_0^t d\tau E_n(\tau)}|\xi_n(t)\rangle. \tag{1.4}$$

Since the system is evolving under the time dependent Hamiltonian $H\left(t\right)$ amplitudes $c_n\left(t\right)$ are clearly time dependent, and their value can be calculated

by substituting the above expression to the original Schroedinger equation:

$$\dot{c}_n(t) = -\sum_{m,n\neq m} c_m(t) \left\langle \xi_n(t) \left| \dot{H}(t) \right| \xi_m(t) \right\rangle e^{-i\int_0^t d\tau\, E_{mn}(\tau)}. \tag{1.5}$$

what after the integration brings:

$$c_n(T) = c_n(0) - \sum_{m,n\neq m} \int_0^T dt\, c_m(t) \left\langle \xi_n(t) \left| \dot{H}(t) \right| \xi_m(t) \right\rangle e^{-i\int_0^t d\tau\, E_{mn}(\tau)}. \tag{1.6}$$

In the adiabatic evolution the sum on the right hand side of the above equation should be small compared to $c_n(0)$. Since the term $e^{-i\int_0^t d\tau\, E_{mn}(\tau)}$ is a fast oscillating its integral will be small or zero if the term $c_m(t) \left\langle \xi_n(t) \left| \dot{H}(t) \right| \xi_m(t) \right\rangle$ is slowly oscillating or zero.

1.5 Adiabatic Quantum Computation Preliminaries

Adiabatic theorem in the form explained above is used in the great number of applications, but one of the most interesting applications of this theorem for us are quantum state preparation and quantum adiabatic computation. The latter, as proposed by Farhi, Goldstone, Gutmann, and Sipser [18], works as follows. Described below are the steps of the quantum adiabatic algorithm for any optimization problem.

- At time $t = 0$, the quantum mechanical system is described by an initial Hamiltonian W, whose ground state is easy to prepare.
- Next, this system is slowly transformed to its final Hamiltonian F, for which the ground state is the solution to a specific minimization problem $f(z)$. We do this is by letting the eigenvalues λ_z of the eigenvectors $|z\rangle$ of F correspond to the function that we try to minimize. Hence, if this function $f(z)$ has domain $\{0,1\}^n$, then the final Hamiltonian would be defined by:

$$F \equiv \sum_{z\in\{0,1\}^n} f(z) \cdot |z\rangle\langle z|_z.$$

Here the outer product $|z\rangle\langle z|_z$ denotes a $2^n \times 2^n$ matrix that is zero everywhere except at z-th diagonal position where it has value 1. Subscript z, again, denotes that the final Hamiltonian is diagonal in the computational basis.

- Finally the measurement of the quantum system in the computational basis reveals the solution of the problem with the high probability.

The choice of the initial Hamiltonian W is independent of the solution of the problem, and will be such that W is not diagonal in the computational z-basis. This is because initial and the final Hamiltonian diagonal in the same basis would, unless a special case, have eigenvalue crossings, so that the quantum system at the end of evolution would end up in an excited state rather than the ground state. In particular, we consider the "Hadamard H-basis" that is 45 degree rotation of computational z basis, with the bit values:

$$|\hat{0}\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \quad \text{and} \quad |\hat{1}\rangle \equiv \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

For a binary string $z \in \{0,1\}^n$, with $|\hat{z}\rangle$ is usually denoted the state which would be written as in $|\hat{0}\rangle$ and $|\hat{1}\rangle$ basis. This basis looks random in the computational basis as the measurement reveals the state $|0\rangle$ and the state $|1\rangle$ with the equal probability. The unitary mapping between these two representations is provided by the n-fold Hadamard matrix: $H^{\otimes n}|z\rangle = |\hat{z}\rangle$ and $H^{\otimes n}|\hat{z}\rangle = |z\rangle$, where $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$.

A simple starting Hamiltonian that fulfills the above requirements is

$$W \equiv \sum_{\hat{z} \in \{0,1\}^n} h(\hat{z}) \cdot |\hat{z}\rangle\langle\hat{z}|,$$

with $h(\hat{0}^n) = 0$ and $h(\hat{z}) \geq 1$ for all other $\hat{z} \neq 0^n$. In this case the ground state with zero energy of W is the uniform superposition $|\hat{0}\cdots\hat{0}\rangle = \frac{1}{\sqrt{2^n}} \sum_z |z\rangle$. This is the most common set up for the initial and the final Hamiltonian of the system. The time-evolution of the Hamiltonian is as follows.

Following the proposal by Farhi et al. in [18, 17], the time dependent Hamiltonian $H(t)$ is the convex combination of the starting and the final Hamiltonian:

$$H(t) := \left(1 - \frac{t}{T}\right) W + \frac{t}{T} F, \tag{1.7}$$

with $0 \leq t \leq T$, and T is the crucial evolution time(delay factor) of the $H_i \rightarrow H_f$ transition. For the purpose of easing the notation the ratio $\frac{t}{T}$ is denoted as s in most cases so that $s \in [0,1]$.

The adiabatic theorem assures that the system with this Hamiltonian will map the initial ground state $|\psi(0)\rangle = |\hat{0}^n\rangle$ to the global minimum of the function $f(z)$, provided that we pick T to be large enough. In the previous section we mentioned that $T \in O(\Delta_{\max} g_{\min}^{-2})$ is a sufficient upper bound on this delay. Without any further knowledge about the specific Hamiltonian $H(t)$ — which involves detailed knowledge about the function $f(z)$, this is also a lower bound for a reliable adiabatic evolution from W to F. Since $\left|\langle \xi_1 | \frac{d}{ds} H(s) | \xi_0 \rangle\right|$, $s = \frac{t}{T}$ is polynomial in n, as long as $f \in \text{poly}(n)$, the dominant factor is only $g_{\min}^{-2}$ and this is why one needs to choose $T \gg g_{\min}^{-2}$ as a requirement for the necessary running time of the adiabatic quantum algorithm.

Hamiltonian locality. One of the important properties of W is its locality. The locality of the initial Hamiltonian W, refers to the number of non identity 2×2 matrices in its representation as a tensor product of Pauli matrices. For example: $W = I \otimes X \otimes I ... I \otimes Y$ is the two local Hamiltonian. This property is important to observe due to the fact that in general only all k-local Hamiltonians are physically realizable, where k is a small fixed number, say 5.

The locality of the Hamiltonians employed by the quantum adiabatic algorithm is also important from computational complexity side. Unless the Hamiltonians are local, even if there exists a polynomial time quantum adiabatic algorithm, the adiabatic algorithm would not render the problem of interest in the bounded error quantum polynomial time complexity class - BQP that is defined in terms of quantum circuits. The model of quantum computation with the circuits is on the other side equivalent to the model of continuous quantum computation with the local Hamiltonians. We will elaborate on this issue latter on.

The aim in adiabatic algorithm design. The aim in the design of an adiabatic algorithm is the following. One would like to design the algorithm, for a problem that is believed to be NP-hard classically by the prescription of the initial and final Hamiltonian with the promise that the lowest eigenvalue gap in the instantaneous Hamiltonian is large enough. Large enough so that the rate at which we make transition from the initial to the final Hamiltonian is not exponentially proportional to the size of the instance and at the same time that the rate of Hamiltonian change is low enough to ensure that system stays in the ground state.

The black box flavour. Adiabatic algorithms are most often regarded as inherently the black box algorithms in some sense. This is due to the fact that eigenstates of the final Hamiltonian in the adiabatic algorithms are prescribed implicitely by the direct transliteration from an implicitely defined problem, say *3SAT*, i.e. in the black box manner.

1.6 The Lower Bound Methodology for the Unique Marked Element Unstructured Search

The purpose of this section is to set the approach and clarify the methodology that has to be used to determine the true power of the continuous quantum computation regardless of whether it is adiabatic or not. To ease the description of the protocol between classical and quantum world, we will refer to the common two party set up, represented by Alice and Bob. A classical adversary Alice will define the problem, and the quantum computation master Bob, will have the task to solve the problem that Alice defined.

Transliteration process. Any protocol that classical Alice and quantum Bob follow, that aims to establish the relation between complexity class NP and BQP, must include the process of *transliteration*. The process of transliteration is the rewriting of the description of the classical problem using classical literals, i.e. bits, graphs, logical formulas etc., into the same problem, i.e. without adding the extra structure or symmetry, described by quantum mechanical alphabet that in our case is going to be the problem description using tensor products of Hermitian matrices. Those Hermitian matrices are in fact going to define the system Hamiltonian that will govern the process of quantum mechanical continuous time computation.

There exist a stark difference in the transliteration process if the classical problem, that we are going to solve with the quantum computer, is described *implicitely* or *explicitely*. Clearly, if the classical problem is described implicitely by a $3SAT$ logical formula Bob can take a look at the formula, since the process of looking, by itself, does not reveal the satisfying assignment of the formula.

The process of transliteration is more subtle and less natural if the classical problem is defined explicitely. Hence, if the classical problem that we are trying to solve is described *explicitely*, like for instance unique marked element unstructured search problem defined below, than the mere fact that Bob looks at the problem instance reveals the problem solution, i.e. the marked element.

Therefore, if the classical problem is defined explicitely we must rely on the fact that somehow the process of transliteration from, say bit string to system Hamiltonian, is done so that Bob does not get to see what was the hidden index of the element Alice marked. One may impose various assumptions to assure that hiding property. However, as we will see in one of the next sections, hiding the index of the marked element from Bob by itself seem not to be enough.

The structure invariance. Alice must also request that the transliteration of her classical problem into the quantum problem, must be done with a transliteration that does not change the structure of the problem. In particular unstructured search transliteration must be done with the Hamiltonian whose all, but one, eigenvalues are the same. That being provided, as we will see further on, Bob has a freedom to choose in which basis, i.e. alphabet, he wants to encode the problem.

Provided the structure invariance, without the loss of generality, we will can assume that Alice writes her hidden index into "the register", that Bob can not read, but the mere process of Alice's writing of hidden index into the register initializes the system Hamiltonian to be just the particular one in the set of exponentially many Hamiltonians, that Bob decided he would like to use to solve the Alice's problem.

Informally the unique marked element unstructured search ($UMEUS$) is the following. In the $N = 2^n$ dimensional space classical adversary Alice chooses a single element in the set $\{0,1\}^n$ and marks it as the solution. Formally, Alice defines the function $f(i)$, for $i \in \{0,1\}^n$ so that:

$$f(i) = \begin{cases} 1, & i = x \\ 0, & \forall i \neq x \end{cases} .$$

Clearly, the aim of the Bob's algorithm is to determine x, the unknown index of the marked element. To complete the task Bob comes up with his favorite Hamiltonian $H_f(b)$ not initialized, i.e. it becomes initialized only after Alice initializes $H_f(b)$ by inputing her x into it, making it to become uniquely defined $H_f(x)$. This is what we refer to as the process of transliteration or hardwireing x into $H_f(b)$.Therefore we should keep in mind what is the protocol between Alice and Bob for the search problem. The protocol described bellow is simply the direct analogy of a clearly and naturally defined transliteration of the classical problem, say *Unique-3SAT* logical formula, into the Hamiltonian *Unique-3SAT* with the additional assumption with respect to the basis. The search protocol is as follows.

The Unique Marked Element Unstructured Search Problem Protocol

1. Bob decides which un-initialized problem Hamiltonian $H_f(b)$ he wants to use (b stands for blank) in the computational basis.

2. Alice defines the marked element x, and initializes the Hamiltonian to be the $H_f(x)$;

3. Bob having no idea about x, prescribes the instantaneous system Hamiltonian $H(s)$ to be:

$$H(s) = -(1-s)H_i - sH_f(x)$$

4. Bob lets the system evolve by changing parameter s in $H(s)$, i.e. turning the "knob" of parameter s that defines the norm of both H_i and $H_f(x)$, to increase from 0 to 1;

5. Bob measures the state of the system when the parameter $s = 1$ and as the result of his measurement he gets as the result the classical string r.

6. Bob asks Alice is it correct that the marked element is r?

7. Alice sees that $r \neq x$, and tells Bob that his solution is wrong.

8. Bob repeats the steps 2.-6. until he gets the positive reply.

The protocol consistency. In this protocol, we have formally allowed Bob the freedom to choose the basis of the un-initialized problem Hamiltonian $H_f(a)$, to ensure the consistency with the protocol that Alice and Bob would follow for the *NP*-complete *3SAT* problem. We use the *3SAT* as a reference problem for our protocol since it is implicitely defined and thus more natural for the transliteration. Just by looking at the logical *Unique*-3SAT formula Bob can simply transliterate between *3* literal clauses and local Hamiltonians. For instance, a clause $(x_1 \vee x_2 \vee \bar{x}_5)$ Bob may transliterate into the Hamiltonian term

$$(I + (-1)^{x_1} Z_1) \otimes (I + (-1)^{x_2} Z_2) \otimes (I + (-1)^{x_5} Z_5),$$

or into some other Hamiltonian that he chooses, and believes that might help him in the computation. He has the complete freedom over the Hamiltonian alphabet in which he transliterates, i.e. the basis in which the terms are

diagonal, the magnitude of each term – that is physically permissible for his apparatus and-or the total number of terms in the problem Hamiltonian – as long as there are polynomially many terms.

Clearly Bob will choose the Hamiltonian alphabet for the time varying local Hamiltonian, that maximizes his chances to find the unique ground state of the problem Hamiltonian, i.e. the unique satisfying assignment of the *Unique-3SAT*. This is what we refer to as the *implicit* problem definition, and that is in sharp contrast to the *explicit* problem definition that we have in the case of the $UMEUS$ problem. Needless to say a priori we have no idea which assignment is to the logical variables $(x_1, x_2, .., x_n) \in \{0,1\}^n$ is a unique solution, i.e. will be the ground state of the system Hamiltonian.

Clearly the two edges of this approach is in the implicit definition of the marked element through the system Hamiltonian. We may end up with the system Hamiltonian that yields the algorithm with the standard quantum speed up, i.e. quadratically faster than the classical algorithm, of if NP is in BQP with even faster algorithm.

The common transliteration. The usual model that have been used in last ten years in the literature is the following. Suppose Alice chooses as the marked element a string $x \in \{0,1\}^n$. Since the Schroedinger equation will not change amplitudes of the initial system state vector unless the off diagonal entries in the system Hamiltonian are non zero, the initial Hamiltonian is usually diagonal in X basis while the final Hamiltonian is diagonal in the computational Z basis. Therefore the time dependent Hamiltonian most frequently advised is the following:

$$\begin{aligned} H(s) &= -(1-s)\bigotimes_{i=1}^{n} XP_i - s\bigotimes_{i=1}^{n}(I + (-1)^{x_i} Z) \\ &= -(1-s)\,|w\rangle\langle w|_x - s|x\rangle\langle x|_z \ , \end{aligned}$$

where I denotes the identity matrix and

$$XP = \frac{1}{2}\,(I + X)\,,$$

and $|w\rangle \equiv \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle$. Matrices $-|w\rangle\langle w|_x, -|x\rangle\langle x|_z$ both have minimum eigenvalue is -1 and are diagonal in in the X and Z basis respectively.

Precisely this model and the Hamiltonian alphabet, i.e. encoding of the *UMEUS* problem, has served to show that the running time of the adiabatic

search algorithm with this is $\mathcal{O}(\sqrt{2^n})$.The usual rationale behind this encoding of the problem into the system Hamiltonian has been the following. Since we have no idea which string $\sigma$the adversary chooses as the solution, giving equal opportunity to each possibility in the set $\{0,1\}^n$ seams a natural choice.

On the Absence of Structure

The core argument in the oracular black box proof that unique market element unstructured search problem have no algorithms more efficient then $\Omega\left(\sqrt{2^n}\right)$ is due to the fact that the problem have absolutely no structure.

Any attempt to transliterate the *UMEUS* problem from classical into the quantum alphabet must employ n-local Hamiltonians, since only the n-local Hamiltonians have all but one eigenvalue that is the same. For example, the n-local Hamiltonian in computational Z basis:

$$\bigotimes_{i=1}^{n-1}(I+(-1)^0 Z)\otimes(I+(-1)^1 Z)$$

has all eigenvalues zero, apart from one eigenvalue -1 for the basis vector $|0^{n-1}1\rangle$. Hence this Hamiltonian is the transliteration of a *UMEUS* problem into the Hamiltonian alphabet in the computational Z basis. The attentive reader may notice that without adding the identity I to each term the the above tensor product on n terms we would have half of the eigenvalues equal 1 and the other half of the eigenvalues equal -1. Therefore such n-local Hamiltonian can not be used to transliterate no structure of *UMEUS* problem. As we will explain further on, n-local Hamiltonians are not considered as viable.

The situation is similar for the *NP*-complete problems that have a lot of structure that we have no clue how to use. As we saw earlier *3SAT* problem can be transliterated into the sum of 3-local Hamiltonians. Hence we would regard a quantum algorithm that employs such Hamiltonians as realizable. In general any k-local Hamiltonian for a fixed k is regarded as realizable.

Infeasibility of n-local Hamiltonians

A few remarks are due here regarding the nature of the n-local Hamiltonians that we consider in this chapter. One can easily think of at least the two classes of Hamiltonians.

The first class. The Hamiltonians that we consider further on is the one that is a tensor product of one qubit terms, so that the Hamiltonian at the same time produces the interaction between all qubits in the register. Since, each term in the tensor product of the Hamiltonian can be diagonalized one can easily find its eigenvalues and in that basis, in which the Hamiltonian is diagonal, it only applies a product to each basis state. For instance, consider the Hamiltonian:

$$H_{1,n} = \bigotimes_{i=1}^{n-1} Z_i \otimes X$$

it has a norm one, but the coupling between the registers qubits raises as we increase n. Generally the current technology can only provide the coupling between the fixed number of qubits. Similarly, one has to take into the account the norm of the Hamiltonian. One Hamiltonian that we already saw in the algorithm presented thus far was:

$$H_{2,n} = \bigotimes_{i=1}^{n} (I + X)_i = N|w\rangle\langle w|$$

that on top of the fact that it produces the global coupling has norm N, that is exponentially large, and therefore unphysical and unscalable. We certainly are not hoping to build a quantum computer that doubles the amount of energy required for operation each time we add one qubit. Therefore this Hamiltonian appears in the literature always scaled by N, that has the unitary unitary norm and is diagonal in the Fourier basis. These Hamiltonians are widely used in the literature to determine the lower bounds for the running time of the continuous time quantum algorithms. The justification for this lies in the fact that quantum computers are able to efficiently perform the Fourier transformation.

One more remark should be made in the defence of the consideration of the n-local Hamiltonians in the algorithmic setup. The absence of structure in the n-local Hamiltonians allows us to solve the Schroedinger equation exactly analytically without the need to rely on the adiabatic approximation. The analytical solution of the Schroedinger equation that we obtained for the n-local Hamiltonians may be the same one that arises in the subspace of the 2 or 3 qubits, once we redesign the algorithm to use 2 or 3-local Hamiltonians. Future studies will show whether this approach that we propose is useful or not. To the best of our knowledge it is new and unexplored for the time being.

The second class. Consider the Hamiltonian in the following form:

$$H_{3,n} = |w\rangle\langle w| + |s\rangle\langle s|$$

where s is a string implicitely defined through, say, a global minimum of function that encodes number of clauses violated in *Unigue-3SAT*. It is important to notice that even though we can implement the Hamiltonian $|s\rangle\langle s|$, as we know the eigenvalues of that Hamiltonian, the Hamiltonian $|w\rangle\langle w| + |s\rangle\langle s|$ should be as hard for the implementation as it is to find its eigenvalues, namely NP-hard.

1.7 Natural Continuous Algorithm with the Time Independent Hamiltonians

1.7.1 $\Theta\left(\sqrt{N}\right)$ Time Algorithm for the Unique Marked Element Detection with the Hamiltonian Oracle in Fourier basis

Bennett, Bernstein, Brassard and Vazirani in [26] showed that any algorithm that can distinguish between the following two cases in the time that is subexponential in the number of qubits can not be a query based black box algorithm. The case (**a**) when in the array of N elements there is no marked element and the case (**b**) where among N unmarked elements of the array there is only one marked, that we sometimes refer to as a solution. This problem is believed to be notoriously hard for the classical computer. Quantum computers thus far have not been known to be able to solve it either.

However, the article by van Dam and Vazirani showed that since the adiabatic algorithm recovers the minimum number of unsatisfied clauses one can recover the actual satisfying assignment that is being solved and therefore that particular instance of adiabatic algorithm can not be regarded as a black box algorithm. It seems also that many researchers today believe that even though the adiabatic algorithm is not a black box algorithm, it would have close to zero chances of recovering solution clusters that arise in the hard instances of SAT formulas in subexponential time.

In this section we step away from the adiabatic formalism and present the natural algorithm with the time independent Hamiltonian for the unique solution detection. Natural continuous time quantum algorithm that we present has the usual $\mathcal{O}\left(\sqrt{N}\right)$ scaling, it is arguably simpler and offers the following three advantages:

- there is no need to initialize the quantum computer in the ground state of the system Hamiltonian;
- determine the running time it suffices to find the lowest eigenvalue gap for the time independent Hamiltonian;
- unlike the adiabatic algorithm the provides the lower bound for the running time of the algorithm our algorithm determines exactly the scaling constant that is of potential importance in the experimental setup.

Moreover, let

$$|w\rangle \equiv \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \ .$$

without loss of generality we could have chosen any other state in the Hadamard basis, the results that follow would stay the same.

Algorithm 1 *Unique Solution Detection:*

1. Prepare the quantum computer in the state $|w\rangle$*;*
2. Apply one the following time independent Hamiltonians:
if there in no marked element the Hamiltonian is:

$$H_0 = -|w\rangle\langle w|$$

if there exists one marked element the Hamiltonian is:

$$H_1 = -|0\rangle\langle 0| - |w\rangle\langle w| \equiv -|0\rangle\langle 0| - S_F|0\rangle\langle 0|S_F^{-1}.$$

3. Measure in the Fourier basis at $T_{meas} \in Unif\,(0,1)$
4. Repeat the evolution with the chosen $H_{0/1}$ *(step 1-2) and measurement (step 3)* k *times*
5. Unless outcome of all k *measurements is* $|w\rangle$ *there is a marked element in the array*

In this setup the adversary is supposed to choose in advance, before we run the algorithm, whether the $H_{0/1}$ is $-|0\rangle\langle 0|$ or $-|0\rangle\langle 0| - |w\rangle\langle w|$ and is not allowed to change her choice afterwards. Matrix S_F denotes transformation to Fourier basis.

We will further make an important assumption that the $\hbar$ is Plank's constant normalized with both 2π and energy units of the Hamiltonian applied to the system, and thus have in our setup dimension of time $[s]$

Theorem 2 *Natural continuous time quantum algorithm which employs time independent Hamiltonian H_1 detects on the existence of the marked with average probability $\frac{1}{2}$ in time $T_{meas} = \frac{\pi\hbar\sqrt{N}}{2}$.*

Proof. If there is no marked element system will stay in the initial state since the Hamiltonian H_0 is diagonal in computational basis and contributes only to the accumulation of the phase. As a result the outcome of the measurement will be with the certainty the state $|w\rangle$.

If there exist marked element the average probability of measuring the initial state $|w\rangle$ will be $\frac{1}{2}$. This comes from the exact solution of the Schroedinger equation that reveals the time dependent probability of measuring $|w\rangle$. When the system Hamiltonian is H_1 the Schroedinger equation becomes:

$$|\dot{\Psi}(t)\rangle = \frac{i}{\hbar}\left(|0\rangle\langle 0| + |w\rangle\langle w|\right)|\Psi(t)\rangle.$$

Since the Hilbert space can be divided in the marked and unmarked subspace this equation can be reduced down to a two dimensional system:

$$|\dot{\Psi}(t)\rangle = \begin{bmatrix} \dot{c}_0(t) \\ \dot{c}_1(t) \end{bmatrix} = \frac{i}{\hbar N}\begin{bmatrix} 1+N & N-1 \\ 1 & N-1 \end{bmatrix}\begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} \tag{1.8}$$

what is equivalent to solving:

$$\begin{bmatrix} \dot{c}_0(t) \\ \dot{c}_1(t) \end{bmatrix} = \frac{i}{\hbar N}\begin{bmatrix} 2 & N-1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} \tag{1.9}$$

since the part of the Hamiltonian proportional to the identity matrix con-

tributes only to the global phase. The solution of this equation is:

$$\begin{aligned}
\begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} &= A(t) \begin{bmatrix} c_0(0) \\ c_1(0) \end{bmatrix} = e^{\frac{i}{\hbar N}\begin{bmatrix} 2t & (N-1)t \\ t & 0 \end{bmatrix}} \begin{bmatrix} c_0(0) \\ c_1(0) \end{bmatrix} \\
&= \begin{bmatrix} 1-\sqrt{N} & 1+\sqrt{N} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} e^{\frac{i\left(1-\sqrt{N}\right)}{\hbar N}} & 0 \\ 0 & e^{\frac{i\left(1+\sqrt{N}\right)}{\hbar N}} \end{bmatrix} \\
&\quad \begin{bmatrix} 1-\sqrt{N} & 1+\sqrt{N} \\ 1 & 1 \end{bmatrix}^{-1} \begin{bmatrix} c_0(0) \\ c_1(0) \end{bmatrix} \\
&= \begin{bmatrix} 1-\sqrt{N} & 1+\sqrt{N} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} e^{\frac{i\left(1-\sqrt{N}\right)}{\hbar N}} & 0 \\ 0 & e^{\frac{i\left(1+\sqrt{N}\right)}{\hbar N}} \end{bmatrix} \\
&\quad \begin{bmatrix} -\frac{1}{2\sqrt{N}} & -\frac{-1-\sqrt{N}}{2\sqrt{N}} \\ \frac{1}{2\sqrt{N}} & -\frac{1-\sqrt{N}}{2\sqrt{N}} \end{bmatrix} \begin{bmatrix} c_0(0) \\ c_1(0) \end{bmatrix}
\end{aligned}$$

where

$$A_{11}(t) = \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2\sqrt{N}} \left(\left(e^{\frac{2i\sqrt{N}t}{\hbar N}} \left(1+\sqrt{N}\right) - 1 + \sqrt{N} \right) \right)$$

$$A_{12}(t) = \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2\sqrt{N}} \left(-1 + e^{\frac{2i\sqrt{N}t}{\hbar N}} \right) \left(-1 + \sqrt{N} \right)$$

$$A_{21}(t) = \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2\sqrt{N}} \left(-1 + e^{\frac{2i\sqrt{N}t}{\hbar N}} \right)$$

$$A_{22}(t) = \frac{1}{2\sqrt{N}} \left(e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}} \left(e^{\frac{2i\sqrt{N}t}{\hbar N}} \left(-1+\sqrt{N} \right) + 1 + \sqrt{N} \right) \right)$$

and $R = \sqrt{iN}$.

The probability of measuring $|0\rangle$ is $P_0(t) = |c_0(t)|^2 = c_0(t)\, c_0^*(t)$ and since $c_0(0) = \frac{1}{\sqrt{N}}$ and $c_1(0) = \frac{1}{\sqrt{N}}$ we have:

$$P_0(t) = \frac{1}{N} \left(A_{11}(t) + A_{12}(t) \right) \left(A_{11}(t) + A_{12}(t) \right)^*$$

where

$$\begin{aligned}
& A_{11}(t) + A_{12}(t) \\
= & \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2\sqrt{N}}\left(\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}\left(1+\sqrt{N}\right)-1+\sqrt{N}\right)+\left(-1+e^{\frac{2i\sqrt{N}t}{\hbar N}}\right)(-1+N)\right) \\
= & \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2\sqrt{N}}\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}\left(1+\sqrt{N}\right)-1+\sqrt{N}+1-N+e^{\frac{2i\sqrt{N}t}{\hbar N}}(-1+N)\right) \\
= & \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2\sqrt{N}}\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}\sqrt{N}+\sqrt{N}-N+e^{\frac{2i\sqrt{N}t}{\hbar N}}(N)\right) \\
= & \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2}\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}+1-\sqrt{N}+e^{\frac{2i\sqrt{N}t}{\hbar N}}\sqrt{N}\right) \\
= & \frac{e^{-\frac{i\left(-1+\sqrt{N}\right)t}{\hbar N}}}{2}\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}\left(1+\sqrt{N}\right)+1-\sqrt{N}\right)
\end{aligned}$$

Finally, success probability is:

$$\begin{aligned}
P_0(t) &= \frac{1}{4N}\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}\left(1+\sqrt{N}\right)+1-\sqrt{N}\right)\left(e^{-\frac{2i\sqrt{N}t}{\hbar N}}\left(1+\sqrt{N}\right)+1-\sqrt{N}\right) \\
&= \frac{1}{4N}\left(\begin{array}{c}\left(1+\sqrt{N}\right)^2+e^{\frac{2i\sqrt{N}t}{\hbar N}}(1-N) \\ +e^{-\frac{2i\sqrt{N}t}{\hbar N}}(1-N)+\left(1-\sqrt{N}\right)^2\end{array}\right) \\
&= \frac{1}{4N}\left(2+2N+(1-N)\left(e^{\frac{2i\sqrt{N}t}{\hbar N}}+e^{-\frac{2i\sqrt{N}t}{\hbar N}}\right)\right) \\
&= \frac{1}{4N}\left(2+2N+2(1-N)\cos\left(\frac{2i\sqrt{N}t}{\hbar N}\right)\right) \\
&= \frac{1}{2}+\frac{1}{2N}+\left(-\frac{1}{2}+\frac{1}{2N}\right)\cos\left(\frac{2\sqrt{N}t}{\hbar N}\right) \\
&\approx \frac{1}{2}\left(1-\cos\left(\frac{2t}{\hbar\sqrt{N}}\right)\right) \\
&= \sin^2\left(\frac{t}{\hbar\sqrt{N}}\right)
\end{aligned}$$

To get the average probability of detection $|0\rangle$ of equal $\frac{1}{2}$ we must wait until $T_{meas} = \frac{\pi\hbar\sqrt{N}}{2}$, as stated.

■

In contrast to the adiabatic algorithm here we are able to determine the scaling constant that is irrelevant with respect to the asymptotic behavior, but in an (un)realistic setup in which a system Hamiltonian eigenvalues are in Joules a measurement in order of seconds would reveal the existence of the marked element up to about 226 qubits.

1.7.2 Constant Time Search Algorithm with the Hamiltonian Oracle in $\frac{\pi}{8}$ Basis

In this subsection we consider the following simple Hamiltonian $|0\rangle\langle 0|_z + |\psi\rangle\langle\psi|$, where $|0\rangle\langle 0|_z$ is a projection onto zero vector in computational basis and $|\psi\rangle\langle\psi|$ is projection onto the $|0\rangle + |w\rangle$. We show that if one can efficiently implement this Hamiltonian on the quantum computer then one can solve unique marked element unstructured search in quantum continuous constant time. Unfortunately, we do not know how to implement this Hamiltonian. Depending of the viewpoint, one can view this as either providing a step towards a solving of unstructured search or showing that this Hamiltonian is NP-hard to implement.

Formally, let us assume the following notation:

$$\begin{aligned} |w\rangle &\equiv \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \\ |\psi\rangle &= \frac{1}{\sqrt{2}} \left(|0\rangle + |w\rangle\right). \end{aligned}$$

and that $|w\rangle$ is the unique marked element in the Fourier basis.

Algorithm 3 ***Unique Marked Element Unstructured Search with the Hamiltonian oracle in $\frac{\pi}{8}$ Basis:***

1. Prepare the quantum computer in the state $|0\rangle_z$;

2. Let Apply the following time independent Hamiltonian:

$$\begin{aligned} H_s(t) &= -|0\rangle\langle 0|_z - |\psi\rangle\langle\psi| \\ &= -|0\rangle\langle 0|_z - S_{\frac{\pi}{8}} |0\rangle\langle 0|_z S_{\frac{\pi}{8}}^{-1} \end{aligned}$$

3. Measure in the Fourier basis.

Claim 4 *Natural algorithm with time independent Hamiltonian H_s reveals with the unique marked element $|w\rangle$ after continuous constant time with the average probability $\frac{1}{4}$.*

We first prove the following general lemma.

Lemma 5 *The gap between the lowest and second lowest eigenvalue of the following Hamiltonian*

$$H(t) = g(t)\,|i\rangle\,\langle i| + f\,(t)\,|\psi\rangle\,\langle\psi|\,, \tag{1.10}$$

is

$$Gap = \sqrt{\left(g(t) - f\,(t)\right)^2 + 4g(t)f\,(t)\,\alpha^2}, \tag{1.11}$$

where $\alpha = \langle\psi|i\rangle = \langle i|\psi\rangle$

Proof. Let $|\xi\rangle$ be one of N instantaneous eigenstates of the Hamiltonian, i.e. $H(t)\,|\xi\rangle = h\,|\xi\rangle$ Projection of the above equation on $\langle i|\cdot|\xi\rangle$ and on the $\langle\psi|\cdot|\xi\rangle$, gives:

$$-H_s(t) + g\,|i\rangle\,\langle i| + f(|\psi\rangle\,\langle\psi|) = 0\,. \tag{1.12}$$

$$\begin{aligned} -h\,\langle i|\xi\rangle + g\,\langle i|\xi\rangle + f\,[\langle i|\psi\rangle\,\langle\psi|\xi\rangle] &= 0 & (1.13)\\ -h\,\langle i|\xi\rangle + g\,\langle\psi|i\rangle\,\langle i|\xi\rangle + f\,\langle\psi|\xi\rangle &= 0, & (1.14)\end{aligned}$$

This homogeneous system has nontrivial solution only if the following determinant is zero,

$$Det\begin{vmatrix} -h+g & f\alpha \\ g\alpha & -h+f \end{vmatrix} = 0, \tag{1.15}$$

$$h^2 - (g+f)\,h + gf - gf\alpha^2 = 0 \tag{1.16}$$

$$\begin{aligned} h_{1/2} &= \frac{(g+f)}{2} \pm \frac{1}{2}\sqrt{(g+f)^2 - 4gf + 4gf\alpha^2} & (1.17)\\ &= \frac{(g+f)}{2} \pm \frac{1}{2}\sqrt{(g-f)^2 + 4gf\alpha^2}. & (1.18)\end{aligned}$$

The difference between this lowest and second lowest eigenvalue gives the desired result. ■

Since in our case $g = f = 1$ the actual gap is of the above algorithm is

$$\begin{aligned} Gap &= \sqrt{(-1+1)^2 + 4\left(\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2N}}\right)^2} & (1.19)\\ &= \sqrt{2} + \sqrt{\frac{2}{N}}. & (1.20)\end{aligned}$$

Therefore, on average, in constant time we will measure with the probability half state $|0\rangle$ and as well with the probability half the state $|\psi\rangle$, within which $|w\rangle$ will appear only with probability half, bringing the overall probability of measuring the solution $|w\rangle$ to be one $\frac{1}{4}$.

However as already stated Hamiltonian $H_s = -|0\rangle\langle 0|_z - S_{\frac{\pi}{8}}|0\rangle\langle 0|_z S_{\frac{\pi}{8}}^{-1}$ should be *NP*-hard to implement, unless quantum computers could search in subexponential time.

1.8 Natural Algorithms with Time Dependent Hamiltonian for NP Problems

This section is dedicated to the second version of the natural algorithms, that unlike the previous version employ the time dependent Hamiltonian, but that do not depend on the second fundamental property of the adiabatic algorithms: the adiabatic change in the Hamiltonian parameters by the speed proportional to the inverse square of the gap of the instantaneous system Hamiltonian. The algorithm that we present here do employ the time varying Hamiltonians, do initialize the quantum computer in the ground state of the initial Hamiltonian but have the following advantage over the adiabatic formalism: (a) they provide as well exact scaling of the running time $\Theta\left(\sqrt{N}\right)$; (b) they do not require the evolution in the ground state of the initial Hamiltonian yet they allow for constant speed of change of the Hamiltonian parameters and provide the constant probability of measuring the lowest eigenvalue of the system Hamiltonian (c) probability of measuring the solution is a fixed probability constant close to one, rather than the average probability constant that we had with the time independent natural continuous algorithms in the previous section.

We will first introduce these algorithms for the in the *Unique-3SAT* problem and then provide the exact proof for the generalized unstructured search problem.

1.8.1 *Unique-3SAT* Problem and its Relaxation

Following our trend to transliterate classical problems into the Hamiltonians diagonal in computational basis, here we make the difference be-

tween the problem Hamiltonian F that arise in unique generalized search and $Unique\ 3-SAT$ problem. These two problems as we will see will have very similar transliteration into the problem Hamiltonian. Nevertheless one additional remark is due here.

Valiant and Vazirani have proved that $UNIQUE-SAT$ is an NP-complete promise problem that decides whether a given Boolean formula is unsatisfiable or has exactly one satisfying assignment. The same polynomial reduction of Valiant and Vazirani implies that $Unique-3SAT$ is an NP-complete problem. We, however, restrict our analysis to the search version of the $Unique-3SAT$ in which instead of just detecting whether there exists a unique satisfying assignment we in fact output such assignment. We will assume that the given formula F is satisfiable and that it has exactly one satisfying assignment. It will be trivial to generalize our search problem to a optimization problem in which our algorithm outputs the assignment that violates the minimum number of clauses in the given logical formula F.

For the unique generalized search we can make transliteration

$$F = diag\left\{f_0, f_1, .., f_{N-1}\right\}$$

basically identical to zero matrix, $f_{i,i\neq m} = 0$ apart from exactly one diagonal entry in the matrix, that we will call marked element $f_m = -1$.

For $Unique-3SAT$ problem $F = diag\left\{f_0, f_1, .., f_{N-1}\right\}$ will be a diagonal matrix were each entry of the matrix $f_i \in \mathbb{Z},\ f_i \leq M, M \in \mathbb{Z}$ apart from exactly one diagonal entry $f_x = -1$. Clearly this is easy to achieve as we can assign the penalty for every unsatisfied clause in the logical formula F to be arbitrary constant strictly greater then 1, say 3, and then subtract the identity Hamiltonian. That transliteration of the $Unique-3SAT$ formula, has the -1 diagonal entry instead of 0 on the diagonal index that corresponds to the satisfying assignment.

Algorithm 6 ***Time dependent natural algorithm for U3SAT with*** n*-local* ***Hamiltonian:***

1. Prepare the quantum computer in the all uniform state $|w\rangle_z$ *in computational basis* z*;*

2. Apply the following Hamiltonian decreasing in constant time $S(t) \in \mathbb{R}$ *from* $1 \rightarrow 0$ *while* $T(t) \in \mathbb{R}$ *increases from* $0 \rightarrow 1$*:*

$$H_I(t) = -S(t)\, XP^{\otimes n} - T(t)\, F$$

3. Measure in the computational basis to reveal the marked element.

For now we will assume that $S(t) = 1$ and $T(t) = t$.

The following will be difference between the transliteration of the *UMEUS* and *Unique-3SAT*.

Let us call the solution subspace the subspace of the Hilbert space that corresponds to the eigenvalue and the eigenvector of the marked element and non-solution subspace its complement.

The Schroedinger equation that determines the time evolution of the system will be:

$$|\dot{\Psi}(t)\rangle = \frac{i}{\hbar N}\begin{bmatrix} 1+Nf_0t & 1 & 1 & 1 & .. & 1 \\ 1 & 1-t+Nf_1t & 1 & 1 & .. & 1 \\ ... & ... & ... & ... & ... & .. \\ 1 & 1 & 1 & 1 & ... & 1+Nf_{N-1}t \end{bmatrix}|\Psi(t)\rangle$$

since the final Hamiltonian for *Unique-3SAT* problem is

$$F = diag\left\{f_0, f_1, .., f_{N-1}\right\}.$$

Under the assumption that

$$|\Psi(t)\rangle = \sum_{i=1}^{N} c_i\left(t\right)|\xi_i\left(t\right)\rangle$$

we will be interested in finding $\rho\left(t\right) = |\Psi^*(t)\rangle\langle\Psi(t)|$, since the modulo square of the system state whose amplitudes $\rho_{ii} = |c_i|^2 = c_i^* c_i$ will reveal probability of measurement for each of the basis vectors $|\xi_i\left(t\right)\rangle$. Clearly this Schroedinger equation falls under the general dynamic system formalism where:

$$|\dot{\Psi}(t)\rangle = iA\left(t\right)|\Psi(t)\rangle$$

where

$$A\left(t\right) = \frac{1}{\hbar N}\begin{bmatrix} 1+Nf_0t & 1 & 1 & 1 & .. & 1 \\ 1 & 1-t+Nf_1t & 1 & 1 & .. & 1 \\ ... & ... & ... & ... & ... & .. \\ 1 & 1 & 1 & 1 & ... & 1+Nf_{N-1}t \end{bmatrix}.$$

Since matrix $A(t)$ is time continuos solution of this differential equation exist and the that solution is unique. Further on we will solve this equation exactly.

The clear distinction between this problem and the generalized problem of the unique marked unstructured search is only in the diagonal elements that are greater then zero.

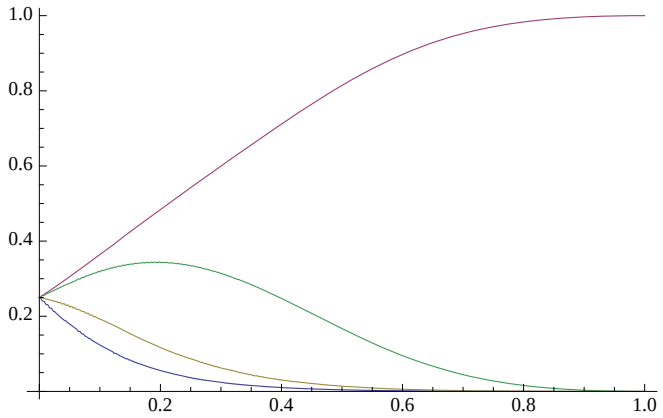


Figure 1.1: Monotonic amplitude build up, in the red color, of the marked element $|1\rangle$ for the two qubit, i.e. 4-dimensional, system.

Claim 7 *Modulo amplitudes squared for Unique-3SAT of the vectors in the non-solution subspace of the system state $|\Psi(t)\rangle$ can be upperbounded by the modulo amplitudes squared of the vectors in the non-solution subspace of UMEUS problem, i.e., for $i \neq m$:*

$$\left|c_i^{U3SAT}(t)\right|^2 \leq \left|c_i^{UMEUS}(t)\right|^2$$

We will leave this claim without the proof, but to illustrate the phenomena of the diagonal entries that are in Unique 3SAT problem $f_i \in \{-1\} \cup \mathbb{Z}_0^+$, rather then $f_i \in \{-1, 0\}$ like in the *UMEUS* problem we solve the toy two qubit example.

The two qubit example that reveals the nature of the solution can be easily simulated numerically:

$$|\dot{\Psi}(t)\rangle = -\frac{i}{4\hbar}\begin{bmatrix} 1+20t & 1 & 1 & 1 \\ 1 & 1-t & 1 & 1 \\ 1 & 1 & 1+10t & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}|\Psi(t)\rangle$$

for the artificially chosen optimization set $F = (f_0, f_1, f_2, f_3) = (20, -1, 10, 0)$. As the figure shows the amplitude modulo squared for $|1\rangle$ is increased to one as the amplitude modulo squared for $|0\rangle, |2\rangle, |3\rangle$ monotonically decrease to zero. The rate of the decrease $\frac{d||\xi_i(t)\rangle|^2}{dt}$ for $i \in \{0, 2, 3\}$ satisfy $\frac{d||\xi_0(t)\rangle|^2}{dt} > \frac{d||\xi_2(t)\rangle|^2}{dt} > \frac{d||\xi_3(t)\rangle|^2}{dt}$since $f_0 > f_2 > f_3$, i.e. $20 > 10 > 0$.

Since the value or the diagonal entries $f_i$of $A(t)$ only determine the different decrease rates for the respective modulo amplitudes we now turn to the general case, that we have in the *UMEUS* problem, in which all the diagonal entries of F are the same apart from one element, that is marked and whose $f_m = -1$.

1.8.2 Unique Marked Element Generalized Search Problem

As we already sad the logical formula that we consider in this section is $F = (f_0, f_1, f_2, .f_{N-1})$ where

$$f_i = \begin{cases} 1, i = m \\ 0, i \neq m \end{cases} .$$

Without loss of the generality we can assume for the easiness in notation that $m = 0$, for which we make the following claim. Similarly as in the previous section we will assume constant $\hbar$ is normalized with the characteristic units of the system Hamiltonian and the units of time characteristic to the time change of the system state probability amplitudes, hence being dimensionless. First, let us first concentrate to the case when. $(\hbar N)^{-1} \geq 1$

Claim 8 *Let the quantum system evolve the generalized n-local Hamiltonian*

$$H_{ST}(t) = -S(t)|w\rangle\langle w| - T(t)|0\rangle\langle 0| = -S(t)H_i - T(t)H_f,$$

such that the time derivative of the Hamiltonian norm parameters $S(t)$ and $T(t)$ is constant, i.e. $\dot{S}(t) = k_S, \dot{T}(t) = k_T \in \mathbb{R}$. There exist $S(t)$ and $T(t)$ such that $S(0) = 1$ and $T(0) = 0$ so that the outcome of the measurement at time $t_m = 2$ in the computational basis of the system's state is the marked state $|0\rangle$ width the probability at least $\frac{1}{8}$, provided that $(\hbar N)^{-1} \geq 1$.

We will now prove this claim first exactly solving the Schroedinger equation to find the system's state at $t_m = 2$ to determine that the measurement will reveal the marked state $|0\rangle$ with the high probability. We will prove this claim for the arguably simplest norm parameter schedule $S(t) = 1$ and $T(t) = t$, even though one can come up the other set of Hamiltonian norm parameters that would also satisfy the given constraints.

As earlier we call the solution subspace $L = \{|\xi_0(t)\rangle\}$ the component of the instantaneous system state $|\xi(t)\rangle$ that corresponds to the marked element $|0\rangle$.

The solution complement subspace is $\bar{L} = \mathbb{C}^n \backslash \{|\xi_0(t)\rangle\}$. Due to the symmetry in the solution complement subspace and the uniqueness of solution we have that $||\xi_1(t)\rangle|^2 = ||\xi_2(t)\rangle|^2 = ||\xi_3(t)\rangle|^2 = .. = \left||\xi_{N-1}(t)\rangle\right|^2$ and hence the $N \times N$ system reduces to the following two dimensional first order differential system

$$\begin{bmatrix} \dot{c}_0(t) \\ \dot{c}_1(t) \end{bmatrix} = \frac{i}{\hbar N} \begin{bmatrix} S(t) + NT(t) & (N-1)S(t) \\ S(t) & (N-1)S(t) \end{bmatrix} \begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} \tag{1.21}$$

Since we can neglect the part of the Hamiltonian proportional the the identity matrix that only contributes to the systems phase without the influence to the amplitudes of the vector components $|\xi_i(t)\rangle$ of the system's state, the above equation reduces to:

$$\begin{bmatrix} \dot{c}_0(t) \\ \dot{c}_1(t) \end{bmatrix} = \frac{i}{\hbar}\frac{1}{N} \begin{bmatrix} 2 - N + NT(t) & N-1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} c_0(t) \\ c_1(t) \end{bmatrix} \tag{1.22}$$

Differentiation of the second equation and the substitution of the first equation reduces this system to:

$$\begin{aligned} \ddot{c}_1(t) &= \frac{i}{\hbar N}\dot{c}_0(t) && (1.23)\\ &= \frac{i}{\hbar N}\left(\frac{i}{\hbar N}(2 - N + NT(t))\, c_0(t) + \frac{i}{\hbar N}(N-1)\, c_1(t)\right) && (1.24)\\ &= \frac{i}{\hbar N}\left(\frac{i}{\hbar N}(2 - N + NT(t))\frac{\hbar N}{i}\dot{c}_1(t) + \frac{i}{\hbar N}(N-1)\, c_1(t)\right) && (1.25) \end{aligned}$$

$$\ddot{c}_1(t) = \frac{i}{\hbar N}(2 - N + NT(t))\,\dot{c}_1(t) - \frac{1}{\hbar^2 N^2}(N-1)\, c_1(t) \tag{1.26}$$

This differential equation can be rewritten in the form:

$$\ddot{c}_1(t) + \frac{i}{\hbar N}(-2 + N - NT(t))\,\dot{c}_1(t) + \frac{1}{\hbar^2 N^2}(N-1)\, c_1(t) = 0 \tag{1.27}$$

$$\ddot{c}_1(t) + i\omega(t)\,\dot{c}_1(t) + \Omega c_1(t) = 0 \tag{1.28}$$

denoting with

$$\begin{aligned} \omega(t) &= \frac{(-2 + N - NT(t))}{\hbar N} \equiv k_N(-2 + N - Nt) \equiv \alpha t + \beta && (1.29)\\ \Omega &= \frac{N-1}{\hbar^2 N^2} \equiv k_N^2(N-1) && (1.30)\\ \alpha &= -k_N N = -\hbar^{-1} && (1.31)\\ \beta &= k_N(N-2) \approx \hbar^{-1} && (1.32) \end{aligned}$$

and $k_N = (\hbar N)^{-1}$.

This equation can be simplified with the following substitution into the Weber differential equation:

$$c_1 = e^{-\frac{i}{4}\left(\alpha t^2 + 2\beta t\right)} u \tag{1.33}$$

$$\ddot{u}(\tau) + \left(m + \frac{1}{2} - \frac{\tau^2}{4}\right) u(\tau) = 0 \tag{1.34}$$

where

$$\tau = \sqrt{\alpha}\left(t + \frac{\beta}{\alpha}\right) e^{-\frac{i\pi}{4}} = i\sqrt{\hbar^{-1}}\left(t - \frac{N-2}{N}\right) e^{-\frac{i\pi}{4}} \tag{1.35}$$

$$m = i\frac{\Omega}{\alpha} = -ik_N \frac{N-1}{N} \approx -i(\hbar N)^{-1} \tag{1.36}$$

Since in our case $\alpha < 0$ that we consider when the Weber differential equation has the following solution in terms of the parabolic cylinder functions $D_m(\tau)$:

$$u(\tau) = \frac{D_{-m-1}(-i\tau)}{D_{-m-1}(-i\tau_0)} \tag{1.37}$$

where

$$\tau_0 = \sqrt{\alpha}\left(\frac{\beta}{\alpha}\right) e^{-\frac{i\pi}{4}} = i\sqrt{\hbar^{-1}}\left(\frac{N-2}{N}\right) e^{-\frac{i\pi}{4}} \approx \sqrt{\hbar^{-1}} e^{-\frac{i\pi}{4}} \tag{1.38}$$

Finally, our aim is to approximate probability of measuring anything in the solution complement subspace:

$$P_1(t) = c_1^*(t)\, c_1(t) = |c_1(t)|^2 = 1 - |c_0(t)|^2 \tag{1.39}$$

$$= |u(t)|^2 = |u(\tau)|^2 = 1 - P_1(t)^2 \tag{1.40}$$

in the neighborhood of $t = 2$, to show that the measurement in the computational basis has the outcome $|0\rangle$ with the probability $P_0(2) = 1 - P_1(2) > \frac{1}{3}$.To do that we substitute all variables into the equation (1.37)and proximate the parabolic cylinder function for the appropriate asymptotic values:

$$u(t) = \frac{D_{i(\hbar N)^{-1}-1}\left(-i^2\sqrt{\hbar^{-1}}\left(t - \frac{N-2}{N}\right) e^{-\frac{i\pi}{4}}\right)}{D_{i(\hbar N)^{-1}-1}\left(-i^2\sqrt{\hbar^{-1}}\left(\frac{N-2}{N}\right) e^{-\frac{i\pi}{4}}\right)} \tag{1.41}$$

$$\approx \frac{D_{i(\hbar N)^{-1}-1}\left(\sqrt{\hbar^{-1}}(t-1) e^{-\frac{i\pi}{4}}\right)}{D_{i(\hbar N)^{-1}-1}\left(-\sqrt{\hbar^{-1}} e^{-\frac{i\pi}{4}}\right)}. \tag{1.42}$$

When $t_m = 2$ we have:

$$
\begin{aligned}
u(2) &\approx \frac{D_{i(\hbar N)^{-1}-1}\left(\sqrt{\hbar^{-1}}e^{-\frac{i\pi}{4}}\right)}{D_{i(\hbar N)^{-1}-1}\left(-\sqrt{\hbar^{-1}}e^{-\frac{i\pi}{4}}\right)} && (1.43)\\
&= \frac{D_{i(\hbar N)^{-1}-1}\left(i\left(-i\sqrt{\hbar^{-1}}e^{-\frac{i\pi}{4}}\right)\right)}{D_{i(\hbar N)^{-1}-1}\left(i\left(i\sqrt{\hbar^{-1}}e^{-\frac{i\pi}{4}}\right)\right)} && (1.44)\\
&= .\frac{D_{i(\hbar N)^{-1}-1}\left(i\sqrt{\hbar^{-1}}e^{-i\frac{3\pi}{4}}\right)}{D_{i(\hbar N)^{-1}-1}\left(i\left(\sqrt{\hbar^{-1}}e^{i\frac{\pi}{4}}\right)\right)} && (1.45)\\
&\approx \frac{e^{i\frac{\pi}{4}\left(1-i(\hbar N)^{-1}\right)}e^{i\frac{\hbar^{-1}}{4}}\left(\sqrt{\hbar^{-1}}\right)^{\left(-1+i(\hbar N)^{-1}\right)}}{e^{-i\frac{3}{4}\pi\left(1-i(\hbar N)^{-1}\right)}e^{i\frac{\hbar^{-1}}{4}}\left(\sqrt{\hbar^{-1}}\right)^{\left(-1+i(\hbar N)^{-1}\right)}} && (1.46)\\
&\quad +\frac{\sqrt{2\pi}}{\Gamma\left(1-i(\hbar N)^{-1}\right)}e^{-i\frac{\pi}{4}(\hbar N)^{-1}}e^{-i\frac{\hbar^{-1}}{4}}\left(\sqrt{\hbar^{-1}}\right)^{-i(\hbar N)^{-1}}
\end{aligned}
$$

Hence the probability of measurement for each of the eigenvectors in the solution complement subspace is:

$$
\begin{aligned}
P_1(2) &= u(2)\,u^*(2) = |u(2)|^2\\
&= \frac{e^{\frac{\pi}{4}(\hbar N)^{-1}}\sqrt{\hbar^{-1}}}{\frac{2\pi e^{\frac{\pi}{2}(\hbar N)^{-1}}}{\Gamma\left(1-i(\hbar N)^{-1}\right)\Gamma\left(1+i(\hbar N)^{-1}\right)}}\\
&= \frac{\Gamma\left(1-i\left(\hbar N\right)^{-1}\right)\Gamma\left(1+i\left(\hbar N\right)^{-1}\right)}{2\pi}e^{\frac{\pi}{2}(\hbar N)^{-1}}\sqrt{\hbar^{-1}}
\end{aligned}
$$

it is easy to check that this expression is much less then $\frac{1}{8}$.

Finally, as we increase the size of the problem N we will eventually run into the regime in which $(\hbar N)^{-1} < 1$. Since the $(\hbar N)^{-1}$ defines the order of the parabolic cylinder function probability amplitude of measuring any state in the solution complement subspace will no longer be negligible. Therefore at $t = 1$ one must stop increasing t, that changes the norm of the Hamiltonian $|0\rangle\langle 0|$, and insert the "idle interval" for time $\tau = \frac{\pi\hbar\sqrt{N}}{2}$, in which the system goes true at least a half period of the probability amplitude oscillation that we have seen in the previous section with the time independent natural continuous algorithm. After that, one may continue with the increase of from $t = 1$ to 2, with constant speed, regardless of the size of the instance. This process of

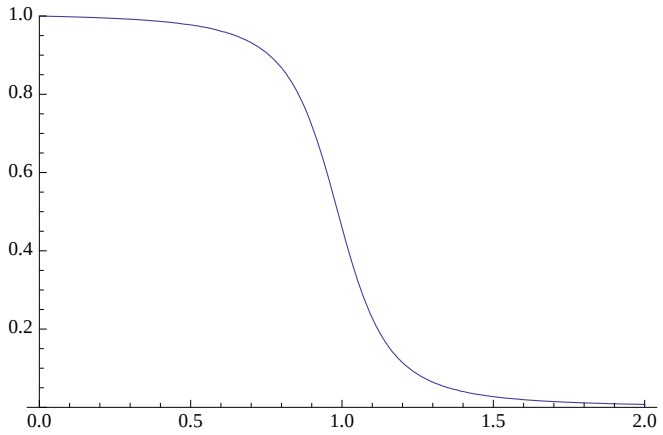


Figure 1.2: Modulo probability amplitude time dependence $|c_1(t)|^2$ for $n = 7$ qubits. The probability amplitudes rendered without the normalization of $\frac{1}{\sqrt{N}}$.

insertion of the "idle interval" is the root of the $\Theta\left(\sqrt{N}\right)$ asymptotic scaling of the time dependent natural continuous algorithm.

1.9 Relevance to The Previously Known Results

Oracular bounds have been used in complexity theory for decades to facilitate our understanding of complexity classes. Even though very powerful in their nature oracular proofs and separations can be misleading, as it was in the case of the oracular approach was the proof that: *IP=PSPACE.*

Still the oracular proofs are somewhat information theoretic in their nature and model, roughly saying, to power of the brute force search. Therefore their acceptance in the literature vary from over emphasizing to underestimating their true power, as Aaronson rightfully notices in his thesis [5]

Some of the results in both discrete and continuous oracular model that address the issue of solving *3SAT* on quantum computer are as follows.

Van Dam, Mosca and Vazirani ([25]) provided the proof that adiabatic search algorithm takes exponential time and it is moreover not a falling into the oracular black box formalism.

Aaronson [5] shows that relative to an oracle, quantum computers could not solve NP-complete problems in polynomial time, even with the help of nonuniform "quantum advice states"; and moreover, that any quantum algorithm needs $\Omega\left(2^{n/4}\right)$ discrete quantum queries to find a local minimum of a black-box function on the n-dimensional hypercube.

Hamiltonian oracles are the continuous counter part of the discrete oracular approach in determining the algorithm's optimal running time.

Mochon [13] managed to prove $\mathcal{O}\left(\sqrt{N}\right)$ lower bound for *3SAT* by using the geodesic equation to find the shortest path on the manifold for the generalized search problem in the quantum computation model that he defined - that is somewhat different from quantum adiabatic computation. In the same paper Mochon raised the question of the existence of the *equivalence* result that would establish the discrete query lower bound once the continuous query lower bound has been determined, and vice versa. Nielsen [11] initiated the question by using algebraic geometry and geodesic approach in finding the shortest path on the manifold. This approach was in the flavor of Mulmuley's [12] approach to avoid the Razborov and Rudic framework of natural proofs.

In the last decade years any time a continuous quantum algorithm was discovered, shortly after the discrete version followed. Even the opposite was partially true. Discovery of the continuous time Grover search and Deutch-Josza algorithm was followed by the discrete version of the algorithm. Most notably the most efficient algorithm for the evaluation of the NAND trees was first discovered in its continuous version.

Farhi, Childs and Goldstone [6] have also used similar continuous approach to the shortest path on the manifold to prove $\mathcal{O}\left(\sqrt{N}\right)$ lower bound for the generalized search problem.

Moreover, Reichardt [9] used exactly the same formulation of adiabatic quantum computation to show that for SAT instance with two qubit agree/disagree clauses adiabatic algorithm has the $\Omega(2^n)$ running time as the minimal instantaneous gap of the system Hamiltonian becomes at some point in the evolution exponentially small.

Good review of the complexity of the local Hamiltonian in 1D and 2D setup is given the chapter of Brandao [6] thesis that gives a good overview of the recent results in the field.

1.10 Strong Hamiltonian Oracles

The purpose of this section is to show that for the adiabatic model of computation, if we assume certain Hamiltonian oracles, that we call strong Hamiltonian oracles we can get the misleading speedup results, unless NP is in BQP, what currently seems to be unlikely. One can also view this result as a framework which shows linear hitting and mixing times for the continuous quantum random walk on the hypercube, but we will not elaborate here on that aspect.

1.10.1 Linear Time Search Algorithm with the Skewed Basis Final Hamiltonian

As explained in the methodology section, unique marked element unstructured search (*UMEUS*) problem is an *explicitely* defined classical problem. The explicitness of the problem definition simply means that by marking the unique element Alice chooses one final system Hamiltonian out of the set exponentially many Hamiltonians that Bob decided to be his favorite set of Hamiltonians for quantum search algorithm. Before Alice's choice Bob has no idea which Hamiltonian Alice is going to pick out of his set. Remember this protocol between Alice and Bob came as an exact analogy to the protocol that is followed by Alice and Bob in the case of implicitly defined hard problem like $3SAT$.

In this section we prove that without the additional assumption that Bob encodes into the basis that is orthogonal to his initial basis vector, Bob can always cheat by transliterating into the skewed basis that allows him to learn one bit of the solution at the time, leading to the linear time quantum algorithm for the unstructured search. This result is certainly misleading as Bob is by transliterating into the skewed basis having access to the oracle that is too powerful.

Therefore the following transliteration, or hard-wiring, between classical and quantum definition of the search problem seems to be the legitimate one.. Alice's marks the element whose index is classical string $\sigma \equiv \sigma_1\sigma_2...\sigma_n$, and, as we explained, writes it into the hidden register that Bob can not read. As soon as she writes σ into the hidden register she effectively choose one out of

2^n Hamiltonians with the following form $\bigotimes_{i=1}^{n}(I \pm Z)$ that becomes the system final Hamiltonian in the computational Z basis:

$$H_f(t) = \bigotimes_{i=1}^{n}(I + (-1)^{\sigma_i} Z) \ = |\sigma\rangle\langle\sigma|_z \ .$$

If we however let the freedom of choice for the final Hamiltonian as a Bob's call, he can choose to cheat by doing the following. The Alice's problem of unique marked element unstructured search has no structure so Bob must use the transliteration into the set of Hamiltonians with no structure as well, but nevertheless over the Hamiltonian alphabet of his choice, preconditioned with his knowledge that he is solving a $UMEUS$ problem. As the following algorithm shows Bob can also choose to transliterate (hard-wire) Alice's problem not in one but rather into the set of n Hamiltonians. This can allow him to run n iterations of the adiabatic algorithm that each resolve one bit of the marked element, and therefore to perform the search in the linear time.

Algorithm 9 ***Adiabatic Unique Marked Element Unstructured Search***

1. Prepare the quantum computer in the state $|w\rangle_x$;
2. Apply the following Hamiltonian increasing $t \in (0, n)$ from 0 to n:

$$H_s(t) = g(t)XP^{\otimes n} + H_f(t)$$

where:

$$\begin{aligned} H_f(t) \ = \ & f_1(t)(I + (-1)^{\sigma_1} Z) \otimes XP^{\otimes n-1} \\ & + f_2(t)\,(I + (-1)^{\sigma_1} Z) \otimes (I + (-1)^{\sigma_2} Z) \otimes XP^{\otimes n-2} ... \\ & + f_{n-1}(t) \bigotimes_{i=1}^{n-1}(I + (-1)^{\sigma_i} Z) \otimes XP \\ & + \ f_n(t) \bigotimes_{i=1}^{n}(I + (-1)^{\sigma_i} Z)\,, \end{aligned}$$

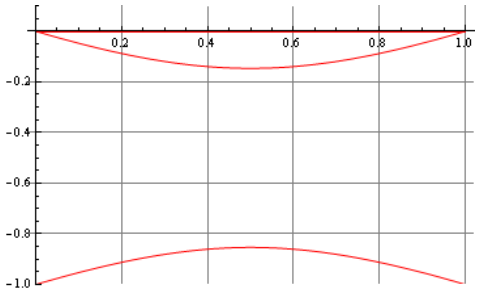


Figure 1.3: Instantaneous Hamiltonian eigenvalues are this image tailed to itself n times.

with the following schedule in n phases:

$$g(t) = \begin{cases} -1+t, \text{ for } t \in [0,1] \\ 0, \text{ otherwise} \end{cases}$$

$$f_i(t) = \begin{cases} -t, \text{ for } t \in [i-1,i] \\ -1+t, \text{ for } t \in [i,i+1] \\ 0, \text{ otherwise} \end{cases} \quad \forall i \in \{1, n-1\}$$

$$f_n(t) = \begin{cases} -t, \text{ for } t \in [n-1,n] \\ 0, \text{ otherwise} \end{cases} .$$

3. Measure in the computational basis to reveal the marked element.

For each of n phases in the parameter space we will have the following eigenvalues of the instantaneous Hamiltonian.

To prove that this Hamiltonian norm change schedule assures that at no time instant the gap gets smaller then $\frac{\sqrt{2}}{2}$, and the overall running time is n we need to find the instantaneous ground state of the system Hamiltonian. Without loss of generality we can assume that the marked state is $ZP^{\otimes n}$ otherwise the calculation that follows will only defer in the appropriate shift in the coefficients.

The lowest two unnormalized eigenvector eigenvalue pair of the instantaneous Hamiltonian in the $i-th$ phase, $i \in \{1, n-1\}$, of the algorithm are, assuming that in each phase $t \in (0,1)$:

$$|\xi_{0,i}(t)\rangle = -\frac{1-t+2t^2+(1+t)\sqrt{1-2t+2t^2}}{(-1+t)\left(1+\sqrt{1-2t+2t^2}\right)} \sum_{k=1}^{2^{n-i}-1} |k\rangle + \sum_{l=2^{n-i}}^{2^{n-i+1}} |l\rangle, \quad (1.47)$$

$$|\xi_{1,i}(t)\rangle = -\frac{-1+t-2t^2+(1+t)\sqrt{1-2t+2t^2}}{(-1+t)\left(-1+\sqrt{1-2t+2t^2}\right)}\sum_{k=1}^{2^{n-i}-1}|k\rangle + \sum_{l=2^{n-i}}^{2^{n-i+1}}|l\rangle. \quad (1.48)$$

While for $i = n$,

$$|\xi_{0,n}(t)\rangle = \frac{t+\sqrt{1-2t+2t^2}}{(-1+t)}|1\rangle + |2\rangle, \quad (1.49)$$

$$|\xi_{1,n}(t)\rangle = \frac{-t+\sqrt{1-2t+2t^2}}{(-1+t)}|1\rangle + |2\rangle. \quad (1.50)$$

with the corresponding eigenvalues for all i's are:

$$h_0^{\downarrow}(t) = -\frac{1}{2}\left(1+\sqrt{1-2t+2t^2}\right), \quad (1.51)$$

$$h_1^{\downarrow}(t) = -\frac{1}{2}\left(1-\sqrt{1-2t+2t^2}\right). \quad (1.52)$$

It follows that the instantaneous gap in the $i-th$ phase of the algorithm is

$$Gap_i = \sqrt{1-2t+2t^2}, \quad (1.53)$$

with the minimum at $t = \frac{1}{2}$, regardless of the size of the input,

$$Gap_{\min}(t) = \frac{\sqrt{2}}{2}. \quad (1.54)$$

The overall running time of this algorithm is $\mathcal{O}(n)$, since there are in total n phases of the algorithm each demanding a constant time.

Bibliography

[1] Leslie Valiant, Vijay Vazirani (1986). "*NP is as easy as detecting unique solutions*". Theoretical Computer Science (North-Holland) 47: 85–93.

[2] Edward Farhi, Jeffrey Goldstone, David Gosset, Sam Gutmann, Peter Shor, *Unstructured Randomness, Small Gaps and Localization,* arXiv:1010.0009

[3] Ramis Movassagh, Edward Farhi, Jeffrey Goldstone, Daniel Nagaj, Tobias J. Osborne, Peter W. Shor, *Unfrustrated Qudit Chains and their Ground States*, arXiv:1001.1006

[4] Edward Farhi, Jeffrey Goldstone, David Gosset, Harvey B. Meyer, *A Quantum Monte Carlo Method at Fixed Energy,* arXiv:0912.4271

[5] S. Aaronson, *Limits on Efficient Computation in the Physical World*, quantu-ph/0412143;

[6] F. Brandao, *Entanglement Theory and the Quantum Simulation of Many Body Physics*, quant-ph/0810.0026v1

[7] I. Chuang, M. Nielsen, *Quantum Computation and Quantum Information*, Cambridge, 2000;

[8] A. Kitaev, A. Shen, M. Vyalyi, *Classical and Quantum Computation*, AMS, 2002;

[9] B. Reichardt, *Quantum Adiabatic Optimization Algorithm and the Local Minima,* STOC 2004;

[10] T. Kieu, *Quantum Adiabatic Computation and Travelling Salesman Problem*, quant-ph/0601151;

[11] Nielsen M., *A geometric approach to circuit lower bounds*, quant-ph/0502070;

[12] Mulmuley K., Sohoni M., *Geometric complexity theory, P vs. NP and explicit obstructions*, Advances in Algebra and Geometry, 2001;

[13] Mochon C., *Hamiltonian Oracles*, quant-ph/0602032;

[14] Razborov A., Rudic S., *Natural Proofs*, STOC 94;

[15] A. Childs, et *al., Exponential algorithmic speedup by quantum walk,* Proceedings of the 35th ACM Symposium on Theory of Computing, pp.59-68, 2003.

[16] A. Childs, E. Farhi, S. Gutmann, *An Example of the difference between quantum and classical random walks,* Quantum Information Processing 1, 35 (2002), quant-ph/0103020;

[17] E. Farhi, 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*, *Science,* Volume 292, April, pp. 472–476 (2001);

[18] E. Farhi, J. Goldstone, S. Gutmann and M. Sipser, "*Quantum Computation by Adiabatic Evolution*", quant-ph report no. 0001106 (2000);

[19] E. Farhi, J. Goldstone, S. Gutmann and D. Nagaj, *How to make the quantum adiabatic algorithm fail,* quant-ph/0512159;

[20] E. Farhi and S. Gutmann, *An Analog Analogue of a Digital Quantum Computation,* quant-ph/9612026;

[21] E. Farhi, J. Goldstone, S. Gutmann, *A Numerical Study of the Performance of a Quantum Adiabatic Algorithm for the Satisfiability,* quant-ph/0007071;

[22] E. Farhi, J. Goldstone, S. Gutmann, *Quantum Adiabatic Evolution Algorithm with Different Paths*, quant-ph/0208135;

[23] A. Childs, E. Farhi, J. Goldstone and Sam Gutmann, *Finding Cliques with the Quantum Adiabatic Evolution,* quant-ph/0012104;

[24] Magniez F. *et al.*, *Search via Quantum Walk*, quant-ph/0606016;

[25] W. van Dam, M. Mosca, U. Vazirani, *How Powerful is Adiabatic Computation,* quant-ph/0206003;

[26] Bennett C., Brassard G., Bernstein E., Vazirani U., *Strengths and Weaknesses of Quantum Computation*, SIAM 1997;

[27] M. Born and V. Fock, "Beweis des Adiabatensatzes", *Zeitschrift für Physik,* Volume 51, pp. 165–180 (1928);

[28] M. B. Ruskai, *Comments on Adiabatic Algorithms,* quant-ph/0203127;

[29] R. Bhatia, *Matrix Analysis*, Springer-Verlag, 1997.

[30] M. S. Siu, *From Quantum Circuits to Adiabatic Algorithms*, quant-ph/0409024v7;

[31] D. Berry, G. Ahokas, R. Cleve and B.Sanders, *Efficient quantum algorithms for simulating sparse Hamiltonians,* Communications in Mathematical Physics, Volume 270, Number 2 / March, 2007

[32] Gupta A, Pathak A. *Modified Grover's search algorithm in* $O(M + logN)$ *steps*, quant-ph/0506093;

[33] Gupta A., Gupta M., Pathak A., *Modified Grover's search algorithm for the cases where the number of solutions is known*, quant-ph/0506105;

[34] C. Zener, *Non-Adiabatic Crossing of Energy Levels*, Proc. R. Soc. Lond. A 1932, 137 696-702

[35] A. Perez, A. Romanelli, *Non-Adiabatic search algorithms*, Physical Review A 76, 052318 (2007)

[36] S. Das, R. Kobes, G. Kunstatter, *Energy and Efficiency of Adiabatic Quantum Search Algorithms*, quant-ph/0204044;

[37] L. Ioannou, M. Mosca, *Limitations of Some Simple Adiabatic Algorithms*, quant-ph/0702241; International Journal of Quantum Information, 2008;

Chapter 2

Experts Algorithm and Non Zero Sum Games

This chapter concentrates on the experts or the matrix multiplicative weights algorithm, presents its improvements and reveals the origin of the close relationship to vector version of the multiplicative updates algorithm. We also present some evidence that the Nash equilibrium for the augmented Shapely's game, i.e. a non-zero sum game, can be found using matrix version of multiplicative weights updates algorithms. A more detailed abstraction of the results presented here follows in the introductory chapter.

2.1 Introduction

Matrix multiplicative weight algorithm or the experts algorithm has provided a very useful framework for many applications: approximate solutions to LPs and SDPs, flow problems, online learning, boosting, derandomization and Chernoff bounds, online convex optimization, computational geometry, metric embeddings, portfolio management, learning the Nash equilibrium of the Zero-sum games. There has been an extensive body of work for algorithms of this type corresponding to this plethora of applications.

2.1.1 Previous Results

There have been a large corpus of results in the literature with respect to gradient based algorithms. The same is true for the multiplicative weight

algorithm that is a direct descendent of a gradient based algorithms. However, our work is mostly inspired by results presented by Arora and Kale [1] in their study of a combinatorial primal-dual approach to SDPs. They consider a matrix version of the game that we will explain further on.

The ancestor of the multiplicative weight algorithm was the vector multiplicative weight algorithm that was extensively used in the area of prediction, learning and games. In particular, it is most commonly used for finding the Nash equilibrium for zero sum games. However, for non-zero sum games not much is known. Daskalakis, Frongillo, Papadimitriou, Pierrakos and Valiant [10] have shown that vector multiplicative weight algorithm is not converging in strategy to a Nash equilibrium for a non zero-sum game. Similar result is shown by Zinkevich [5], although in a different set up. We extend this study and show that a matrix multiplicative weights algorithm converges to the Nash equilibrium for non-zero sum games. In particular it shows a convergence in strategy to a Nash equilibrium of the augmented Shapley game, that we will define later on in the text. A definite, general answer to the question of convergence demands a further study and it is beyond of the scope of this presentation.

2.1.2 Motivation

Game theoretic model that we consider is the following. Two players play a zero-sum game. The row player, Alice, has no previous knowledge over what matrix she is going to see as a first event, namely a matrix M that the column player, Bob, is going to choose and reveal. As a result Alice following the matrix multiplicative weight algorithm is going to update her own state of knowledge, namely a matrix ρ. Alice and Bob continue to play this game for T rounds.

The problem that we consider is to find the algorithm such that after T rounds of game, the loss of the row player is not much larger than the loss of the hypothetical player who would play a single best strategy in all T rounds and by playing it encounter the smallest possible pure strategy loss.

The matrix multiplicative weights algorithm is a surprisingly simple algorithm with respect to quality of the promise that it achieves. Results presented in this here provide only a partial answer to the questions that naturally arise along the attempt to acquire a deeper understanding of this algorithm. Our aim was to establish a clear, and to the best of our knowledge, yet unknown relation, to the vector multiplicative weights algorithm. The motivation for us

was to reveal the true nature of the algorithm in both cumulative and iterative set up. This was somewhat important to the effort to clarify the apparent similarity between matrix multiplicative weight algorithm with gradient descent/ascent approach and possibly to the natural evolution in the quantum systems.

2.2 Matrix Multiplicative Weights Algorithm with Cumulative Updates

In the text that follows we will assume that the matrix M, that defines the loss for the row player A is a symmetric matrix, positive definite matrix, and that its eigenvalues satisfy $\lambda_1(M) \geq \lambda_2(M) \geq \cdot \cdot \cdot \geq \lambda_n(M)$. It is also useful to denote the inner product of the matrices, say, M and ρ with $M \cdot \rho \equiv \sum_{ij} M_{ij}\rho_{ji}$, and with "$\circ$" the Hadamard product $[M \circ \rho]_{ij} \equiv M_{ij}\rho_{ij}$ of the matrices that we will be using in the next chapter.

2.2.1 Game Theoretic Setup

In the matrix generalization of the usual 2-player zero-sum game the row player (Alice) chooses a unit vector $v \in \mathbb{S}^{n-1}$, and the column player (Bob, the external adversary) chooses a matrix M such that $0 \leq M \leq I$, and $\dim(M) = n$. Then the row player has to pay to the column player $v^T M v = M \cdot vv^T$. Since the row player chooses his strategy v vector from a distribution $\mathcal{D}$ over $\mathbb{S}^{n-1}$ in each round we are interested in the expected loss of the row player:

$$\mathbb{E}_{\mathcal{D}}\left[v^T M v\right] = M \cdot \mathbb{E}_{\mathcal{D}}\left[vv^T\right] \equiv M \cdot \rho$$

and $\rho \equiv \mathbb{E}_{\mathcal{D}}\left[vv^T\right]$ is a, so called, density matrix that is clearly symmetric, positive definite and has $Tr\left[\rho\right] = 1$.

We are interested here to consider repeated version of this game in which row player has to react to an observed event M_i, $i \in \{1, .., T\}$, as external adversary picks in each round a different payoff matrix M_i, such that $0 \preceq M_i \preceq 1$. The algorithms that we are going to consider will provide the row player with the sequence of density matrices, $\rho_1, \rho_2, ...\rho_T$, that are responses to the T observed events $M_1, M_2, ...M_T$, each for the T rounds of the game. This setup is sometimes called a prediction game. For now we will not assume that matrices M_i are non-negative, even though implicitly by definition matrices

M_i are loss matrices that have all entries non-negative, otherwise matrices M_i would define both losses and gains what is not usually not the case in game theoretic setup. This detail will eventually became important later on in the text, as we will see.

The arbitrary correlation. As it will be clear from the algorithm explained below, the repeated version of this game is deterministic and does not include any notion of distribution over vectors by which the row player responses to the observed event. With respect to that this approach is very powerful as it allows arbitrary correlation between adversaries moves.

The aim of the matrix multiplicative weights algorithm is to minimize the total loss of the row player (or the algorithm) A, after the T rounds of the game, and that is defined as:

$$Loss(A) = \sum_{t=1}^{T} M_t \cdot \rho_t \equiv \sum_{t=1}^{T} Tr\left(M_t \rho_t\right) = \sum_{t=1}^{T} Tr\left(\rho_t M_t\right)$$

As we will see, this total loss of the algorithm can be tightly related to the loss of an ideal clairvoyant row player, who would see all the adversary moves in advance, decide which strategy to play and play that fixed strategy, say v_{opt}, through all T rounds of the game. As a result ideal player would suffer the loss of over T rounds that is equal to

$$\lambda_{\min}\left(\sum_{t=1}^{T} M_i\right) = \min_{v} v^T \left(\sum_{t=1}^{T} M_i\right) v = v_{opt}^T \left(\sum_{t=1}^{T} M_i\right) v_{opt}.$$

In what follows we will see that if the row player follows the prescribed strategy its loss will be very close to $\lambda_{\min}\left(\sum_{t=1}^{T} M_i\right)$ in the case of loss matrices or $\lambda_{\max}\left(\sum_{t=1}^{T} M_i\right)$ in the dual picture of gain matrices.

2.2.2 The Cumulative Updates

The following algorithm will be our starting point in the study that follows *[1]*.

> *Matrix Multiplicative Weights Algorithm with Cumulative Updates*
>
> Fix an $\epsilon < \frac{1}{2}$ and denote with $\epsilon' = -\ln(1-\epsilon)$, and let $W_1 = \rho_1 = \frac{1}{n} I$. For $t = 1, 2, ..T$ do the following:

1. Compute $W_{t+1} = W_1 (1-\epsilon)^{\sum_{\tau=1}^{t} M_\tau} \triangleq W_1 \exp\left(-\epsilon' \left(\sum_{\tau=1}^{t} M_\tau\right)\right)$
2. Play $\rho_{t+1} = \frac{W_{t+1}}{Tr(W_{t+1})}$ and observe the column player next move M_t.

Theorem 10 *The Matrix Multiplicative Weights algorithm with cumulative updates generates density matrices $\rho_1, \rho_2, ..., \rho_T$ such that:*

$$\sum_{t=1}^{T} M_t \cdot \rho_t \leq \min\left\{(1+\epsilon)\,\lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) + \frac{\ln n}{\epsilon}, (1+\epsilon)\,\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right)\right\} \tag{2.1}$$

where $\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right)$ denotes the loss of hypothetical player, i.e. the player that would in the hindsight achieve the average loss among players.

Proof. We can partially follow the same methodology of the proof of the similar theorem that is given by Arora and Kale [1], to emphasize the difference that yields the final result 2.1. We can find the upper and lower bound for the loss in the $T+1$ round $Tr(W_{T+1})$, as sum of this quantity is convenient since it effectively encodes the payoffs that the row player will "suffer". To bound $Tr(W_{T+1})$ from above we have:

$$\begin{aligned}
Tr\left(W_{T+1}\right) &= Tr\left(W_1 \exp\left(-\epsilon' \sum_{\tau=1}^{T} M_\tau\right)\right) \\
&\leq Tr\left(W_1 \exp\left(-\epsilon' \sum_{\tau=1}^{T-1} M_\tau\right) \exp\left(-\epsilon' M_T\right)\right) \\
&= Tr\left(W_1 \exp\left(-\epsilon' \sum_{\tau=1}^{T-1} M_\tau\right) (1-\epsilon)^{M_T}\right) \\
&\leq Tr\left(W_T\left(1-\epsilon M_T\right)\right) \\
&= Tr\left(W_T\right)\left(1-\epsilon\frac{Tr\left(W_T M_T\right)}{Tr\left(W_T\right)}\right) \\
&= Tr\left(W_T\right)\left(1-\epsilon Tr\left(\rho_T M_T\right)\right) \\
&= Tr\left(W_T\right)\left(1-\epsilon \rho_T \cdot M_T\right) \\
&\leq Tr\left(W_T\right)\exp\left(-\epsilon M_T \cdot \rho_T\right)
\end{aligned} \tag{2.2}$$

The first inequality follows by Golden-Thompson inequality and by $(1-\epsilon)^A = \exp\left(-\epsilon' A\right) \preceq (I - \epsilon A)$ for $0 \preceq A \preceq 1$ being a matrix in second inequality or a number in the third inequality.

Since we can start from $W_1 = \frac{1}{n}I$ it follows $Tr\left(W_1\right) = 1$. Again $M_t \cdot \rho_t$ are just real numbers that commute therefore, iterative application of the inequality we have derived above yields:

$$Tr\left(W_{T+1}\right) \leq Tr\left(W_1\right) \exp\left(-\epsilon \sum_{t=1}^{T} M_t \cdot \rho_t\right).$$

Moreover it is even easier to bound from below:

$$\begin{aligned} Tr\left(W_{T+1}\right) &= Tr\left(W_1 \exp\left(-\epsilon' \sum_{t=1}^{T} M_t\right)\right) \\ &= \frac{1}{n} Tr\left(\exp\left(-\epsilon' \sum_{t=1}^{T} M_t\right)\right) \\ &= \frac{1}{n} \sum_{i=1}^{n} \exp\left(-\epsilon' \lambda_i \left(\sum_{t=1}^{T} M_t\right)\right) \\ &\geq \left\{ \begin{array}{c} \frac{1}{n} \exp\left(-\epsilon' \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right)\right) \\ \exp\left(-\epsilon' \frac{1}{n} \sum_{i=1}^{n} \lambda_i \left(\sum_{t=1}^{T} M_t\right)\right) \end{array} \right. . \end{aligned}$$

The last inequality follows since the arithmetic mean of positive real numbers is not smaller than the geometric mean of those numbers. Composing lower and upper bound and taking the logarithm of both sides we have:

$$\begin{aligned} \left. \begin{array}{c} \frac{1}{n} \exp\left(-\epsilon' \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right)\right) \\ \exp\left(-\epsilon' \lambda_{avg}\left(\sum_{t=1}^{T} M_t\right)\right) \end{array} \right\} &\leq \exp\left(-\epsilon \sum_{t=1}^{T} M_t \cdot \rho_t\right) \\ \left. \begin{array}{c} -\ln\left(n\right) - \epsilon' \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) \\ -\epsilon' \lambda_{avg}\left(\sum_{t=1}^{T} M_t\right) \end{array} \right\} &\leq -\epsilon \sum_{t=1}^{T} M_t \cdot \rho_t \\ \left. \begin{array}{c} \frac{\ln(n)}{\epsilon} - \frac{\ln(1-\epsilon)}{\epsilon} \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) \\ -\frac{\ln(1-\epsilon)}{\epsilon} \lambda_{avg}\left(\sum_{t=1}^{T} M_t\right) \end{array} \right\} &\geq \sum_{t=1}^{T} M_t \cdot \rho_t \\ \left. \begin{array}{c} \frac{\ln(n)}{\epsilon} - \frac{-\epsilon-\epsilon^2}{\epsilon} \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) \\ -\frac{-\epsilon-\epsilon^2}{\epsilon} \lambda_{avg}\left(\sum_{t=1}^{T} M_t\right) \end{array} \right\} &\geq \sum_{t=1}^{T} M_t \cdot \rho_t \\ \left. \begin{array}{c} \frac{\ln(n)}{\epsilon} + (1+\epsilon) \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) \\ (1+\epsilon) \lambda_{avg}\left(\sum_{t=1}^{T} M_t\right) \end{array} \right\} &\geq \sum_{t=1}^{T} M_t \cdot \rho_t \end{aligned}$$

as claimed. By Taylor expansion we can notice that $\log(1-\epsilon) = -\epsilon - \frac{\epsilon^2}{2} - \frac{\epsilon^3}{3} - O(\epsilon^4) \leq -\epsilon - \frac{\epsilon^2}{3}$ since for $0 < \epsilon < 1/2$, $\frac{\epsilon^2}{2} + \frac{\epsilon^3}{3} + O(\epsilon^4) = \frac{\epsilon^2}{2}\left(1 + \frac{2}{3}\epsilon + \frac{2}{4}\epsilon^2 + ..\right) \leq \frac{\epsilon^2}{2}\frac{1}{1-\epsilon} \leq \epsilon^2$ □ ■

Remark 11 *We can start from any of the following* $W_1 = \left\{\frac{1}{n}J, \frac{1}{n}I\right\}$, *i.e. from maximally mixed or maximally pure state, or any other positive definite matrix* W_1 *that has* $Tr(W_1) = 1$. *The only difference between the two cases will be on which side of inequality* $\log n$ *term will show up; eventually giving the same end result. The symbol* J *here denotes a matrix* $n \times n$ *that has all entries equal to one. Starting from maximally mixed state, that corresponds to the uniform distribution over all experts, was the choice in [1].*

Remark 12 *It is worthwhile noticing here that the updates are not iterative but rather cumulative, as the row player's next move is determined with the cumulative sum of the history of observed events* M_i *thus far, that are exponentiated. As we will see later the relaxation of this requirement, to the case where all* M_i *commute will immediately reduce to the vector version of the matrix multiplicative weight algorithm.*

Remark 13 *The above result includes minor generalization of the result derived in [1], by encompassing more adversary strategies and giving a better bound for some of them. For example, if adversary plays as a part of his strategy sequence of identity matrices then* $\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right) = \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right)$, *therefore* $(1+\epsilon)\,\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right)$ *gives* $\frac{\log n}{\epsilon}$ *better bound. Moreover, the same is true when* $T \leq \frac{\log n}{\epsilon(1+\epsilon)}$ *as there have been not enough rounds in the game to achieve:*

$$\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right) \geq \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) + \frac{\ln N}{\epsilon\,(1+\epsilon)}$$

Finally, it is good to keep the dual picture when losses are replaced with the gains, the analogous inequality can be written down

$$\sum_{t=1}^{T} M_t \cdot \rho_t \geq \max\left\{(1-\epsilon)\,\lambda_{\max}\left(\sum_{t=1}^{T} M_t\right) - \frac{\ln n}{\epsilon}, (1-\epsilon)\,\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right)\right\} \tag{2.3}$$

where the updates are obtained with the following algorithm:

Matrix Multiplicative Weights Algorithm with Cumulative Updates (Gains Instead of Losses)

Fix an $\epsilon < \frac{1}{2}$ and denote with $\epsilon' = \ln(1+\epsilon)$, and let $W_1 = \rho_1 = \frac{1}{n}I$.For $t = 1, 2, ..T$ do the following:

1. Compute $W_{t+1} = W_1 (1+\epsilon)^{\sum_{\tau=1}^{t} M_\tau} \triangleq W_1 \exp\left(\epsilon' \left(\sum_{\tau=1}^{t} M_\tau\right)\right)$

2. Play $\rho_{t+1} = \frac{W_{t+1}}{Tr(W_{t+1})}$ and observe the column player next move M_t.

Interestingly the upper bound that we presented above can not be improved even by $\mathcal{O}(\epsilon^2)$, what would yield the following result.

$$\sum_{t=1}^{T} M_t \cdot \rho_t \leq \lambda_{\min}\left(\sum_{t=1}^{T} M_t\right) + \frac{\ln n}{\epsilon} \tag{2.4}$$

The reason is the following. Since at each step of the matrix multiplicative weight algorithm we do the normalization we have the freedom over choosing whether $W_{t+1} = W_1 (1+\epsilon)^{\sum_{\tau=1}^{t} M_\tau} \triangleq W_1 \exp\left(\epsilon' \left(\sum_{\tau=1}^{t} M_\tau\right)\right)$ or $W_{t+1} = W_1 (1-\epsilon)^{\sum_{\tau=1}^{t} M_\tau} \triangleq W_1 \exp\left(-\epsilon' \left(\sum_{\tau=1}^{t} M_\tau\right)\right)$ depending on whether the matrices M_i are defined as losses or gains. Once we define the updates we defined whether we need to bound $Tr(W_{T+1})$ from above or below in order to get the interesting bound. For example in the case when matrices M_i are defined as losses, when $W_{t+1} = W_1 \exp\left(-\epsilon' \left(\sum_{\tau=1}^{t} M_\tau\right)\right)$, to get the nontrivial bound we need to bound $Tr(W_{T+1})$ from above and this is done by introducing the constant $-\ln(1-\epsilon)$ that is basically $\mathcal{O}(\epsilon^2)$ greater then ϵ'. Therefore, the bounds that we have seen above can not be improved for $\mathcal{O}(\epsilon^2)$ within this proof structure.

Theorem 14 *The following matrix update algorithm guaranties that the total gain will be:*

$$\sum_{t=1}^{T} M_t \cdot \rho_t \geq \max\left\{(1-\epsilon)\lambda_{\max}\left(\sum_{t=1}^{T} M_t\right) - \frac{\ln n}{\epsilon}, (1-\epsilon)\lambda_{avg}\left(\sum_{t=1}^{T} M_t\right)\right\}$$

Matrix Multiplicative Weights Algorithm with Iterative Updates (Matrices M_i are Gains)

Fix an $\epsilon < \frac{1}{2}$ and denote with $\epsilon' = \ln(1+\epsilon)$, and let $W_1 = \rho_1 = \frac{1}{n}I$

For $t = 1, 2, ..T$ do the following:

1. Compute $W_{t+1} = W_t + \epsilon\sqrt{M_t}W_t\sqrt{M_t}$
2. Play $\rho_{t+1} = \frac{W_{t+1}}{Tr(W_{t+1})}$ and observe the column player next move M_t.

We will leave the proof of this theorem until the end of the next chapter, until after we build the necessary formalism for the proof.

2.3 Matrix Multiplicative Update Algorithm with Hadamard Updates

2.3.1 Preliminaries: Hadamard Product

In this chapter we will define and prove some properties of the Hadamard product.

Definition 15 *Let A and B be $m \times n$ matrices with entries in $\mathbb{C}$. The Hadamard product of A and B is defined by $[AB]_{ij} = [A]_{ij}[B]_{ij}$ for all $1 \leq i \leq m, 1 \leq j \leq n$.*

Since the Hadamard product is simply entrywise multiplication, it inherits the same benefits and restrictions as the multiplication in $\mathbb{C}$. Clearly matrices need to be of the same size, but not necessarily square. To distinguish it from the regular matrix multiplication we will use symbol $\circ$ to denote the Hadamard product.

Lemma 16 *Let A and B be $m \times n$ matrices with entries in $\mathbb{C}$. The Hadamard product is commutative: $A \circ B = B \circ A$.*

Proof. The proof follows straight forwardly from definition $[A \circ B]_{ij} = [A]_{ij}[B]_{ij} = [B]_{ij}[A]_{ij} = [B \circ A]_{ij}$ ■

Lemma 17 *The identity matrix under the Hadamard product is $m \times n$ with all entries equal to one, that we will call J. Hence, $J_{ij} = 1$ for all $1 \leq i \leq m, 1 \leq j \leq n$.*

Proof. Follows from definition analogously to the previous lemma. ■

Lemma 18 *Let A be an $m \times n$ matrix. Then A has a Hadamard inverse, denoted $\hat{A}$, if and only if $[A]_{ij} \neq 0$ for all $1 \leq i \leq m, 1 \leq j \leq n$. Furthermore, $[\hat{A}]_{ij} = ([A]_{ij})^{-1}$*

Proof. Let A be an $m \times n$ matrix with Hadamard inverse $\hat{A}$. Then we know $A \circ \hat{A} = J$. That is, $[A \circ A]_{ij} = [A]_{ij}[\hat{A}]_{ij} = 1$. Multiplying by inverses in $\mathbb{C}$ we know that $[\hat{A}]_{ij} = (1)\left(([A]_{ij})^{-1}\right) = ([A]_{ij})^{-1}$ which is only possible when all entries of A are invertible (in $\mathbb{C}$), i.e. when, $[A]_{ij} \neq 0$ for all $1 \leq i \leq m, 1 \leq j \leq n$.

Inversely, for any $m \times n$ matrix A with entries in $\mathbb{C}$ such that $[A]_{ij} \neq 0$ there exists $([A]_{ij})^{-1}$ for all i, j. This implies $[A]_{ij}([A]ij)^{-1} = ([A]_{ij}^{-1})[A]_{ij} = 1$, and so A has an inverse $\hat{A}$ defined by $[\hat{A}]_{ij} = ([A]_{ij})^{-1}$ for all i, j. ■

Lemma 19 *(Linearity of Hadamard Product) Suppose $\alpha \in \mathbb{C}$, and that A, B and C are $m \times n$ matrices. Then $C \circ (A+B) = C \circ A + C \circ B$, and furthermore, $\alpha(A \circ B) = (\alpha A) \circ B = A \circ (\alpha B)$.*

Proof. Both equalities can be straightforwardly proved:

$$\begin{aligned} [C \circ (A+B)]_{ij} &= [C]_{ij}\,(A+B)_{ij} = [C]_{ij}[A+B]_{ij} \\ &= [C]_{ij}\,[A]_{ij} + [C]_{ij}\,[B]_{ij} = [C \circ A + C \circ B]_{ij} \end{aligned}$$

$$\begin{aligned} \alpha[A \circ B]_{ij} &= \alpha[A]_{ij}[B]_{ij} = [\alpha A]_{ij}[B]_{ij} \\ &= [\alpha A \circ B]_{ij} = [A]_{ij}\alpha[B]_{ij} \\ &= [A \circ \alpha B]_{ij} \end{aligned}$$

■

Now we provide the following useful result.

.

Theorem 20 ***(Shur's Product Theorem)*** *Suppose A and B are positive semidefinite matrices of size n. Then $A \circ B$ is also positive semidefinite.*

Proof of this theorem demands two other smaller results and we will not include it here. One version of the proof can be found for example in [19].

Theorem 21 ***(Oppenheim's Inequality)***. *Let A and B be positive semidefinite matrices of size n. Then $|A \circ B| \geq [A]_{11}...[A]_{nn}|B|$.*

The proof of Oppenheim's Inequality it is somewhat long and requires a further setup, and can be found on page 144 of Bapat [18].

Theorem 22 ***(Hadamard's Inequality)*** *Suppose A is positive semidefinite of size n. Then the determinant $|A| \leq [A]_{11}...[A]_{nn}$.*

Proof. Let A be any positive semidefinite matrix of size n. Note that I_n is a positive semidefinite matrix of size n. Now we have the following, due to Oppenheim's Inequality: $|A| = [I_n]_{11}...[I_n]_{nn}|A| \leq |I_n \circ A| = [A]_{11}...[A]_{nn}$ ■

Corollary 23 *Let A and B be positive semidefinite matrices of size n. Then $|A \circ B| > |AB|$.*

Proof.

$$|A \circ B| \geq [A]_{11}...[A]_{nn}|B| \geq |A||B| = |AB|$$

■

Finally the following lemma will be also useful for us:

Lemma 24 *Let A and B be nonnegative matrices, the following holds:*

$$Tr\,(A \circ B) \leq Tr(AB).$$

Proof. Inequality is immediate by definition. Since $a_{ij}, b_{ij} \geq 0$ and left hand side is $\sum_{i=1}^{n} a_{ii}b_{ii}$ while the right hand side is $\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}b_{ji}$. ■

2.3.2 Pretty Good Bound for $Tr\left(e^{\epsilon \sum_{i=1}^{T} H_i}\right)$

The purpose of this section is to prove the following theorem that naturally leads toward the algorithm in the section that follows. Ando [6] proved similar theorem for the unitarily invariant norm. The theorem that follows is in some sense weaker and a bit less involved.

Theorem 25 *For Hermitian Matrices H_i, $i \in \{1, ..m\}$ and $\alpha > 0$ holds that:*

$$Tr\left(\left(e^{\alpha H_1} \circ e^{\alpha H_2} ... \circ e^{\alpha H_m}\right)^{\frac{1}{\alpha}}\right) \geq Tr\left(e^{\sum_{j=1}^{m} H_j}\right)$$

Proof. Let $\mathbb{M}_n$ denote the space of $n \times n$ matrices. If tensor product of m, $n \times n$ matrices $\mathbb{M}_n$, is identified with $\mathbb{M}_{n^m}$ in a natural way, there is uniquely an unital positive linear map from $\mathbb{M}_{n^m}$ to $\mathbb{M}_n$ that satisfies:

$$\Phi\left(H_1 \otimes ... \otimes H_m\right) = H_1 \circ ... \circ H_m, \;\; s.t. \;\; H_i \in \mathbb{M}_n$$

where $H_1 \circ ... \circ H_m$ is the Hadamard i.e. entrywise product. of $H_1, ..., H_m$. By unital we mean that $\Phi\left(I_{n^m}\right) = I_n$, and the positive means that $\Phi\left(H\right) \geq 0$, whenever $H \geq 0$.

This is important since unital positive linear maps between C^*-algebras satisfy a nice property:

$$\Phi\left(H\right)^p \geq \Phi\left(H^p\right), \;\; for \;\; H > 0, \, 0 < p \leq 1,$$

and

$$\log \Phi\left(H\right) \geq \Phi\left(\log H\right), \;\; for \;\; H > 0.$$

For Hermitian matrices $H_1, ..., H_m \in \mathbb{M}_n$ we can consider the tensor product of $e^{H_1} \otimes e^{H_2} .. \otimes e^{H_m}$ for which we know the following:

$$e^{H_1} \otimes e^{H_2} .. \otimes e^{H_m} = \prod_{j=1}^{m} I \otimes ... \otimes I \otimes \underset{(j)}{e^{H_j}} \otimes I \otimes ... \otimes I.$$

Since products $I \otimes ... \otimes I \otimes \underset{(j)}{e^{H_j}} \otimes I \otimes ... \otimes I$ are mutually commuting we can take the logarithm of the both sides to get:

$$\log\left(e^{H_1} \otimes e^{H_2} .. \otimes e^{H_m}\right) = \sum_{j=1}^{m} \log\left(I \otimes ... \otimes I \otimes \underset{(j)}{e^{H_j}} \otimes I \otimes ... \otimes I\right),$$

$$\log\left(e^{H_1} \otimes e^{H_2} .. \otimes e^{H_m}\right) = \sum_{j=1}^{m} I \otimes ... \otimes I \otimes \underset{(j)}{H_j} \otimes I \otimes ... \otimes I \;.$$

Using the property of the positive unital map of Φ, we know that:

$$\Phi\left(I \otimes ... \otimes I \otimes \underset{(j)}{H_j} \otimes I \otimes ... \otimes I\right) = H_j \circ I, \quad for \;\; j = 1, 2, ...m.$$

from which it follows by convexity of the unital map that:

$$\log\left(e^{\alpha H_1} \circ e^{\alpha H_2} .. \circ e^{\alpha H_m}\right)^{\frac{1}{\alpha}} \geq \left(\sum_{j=1}^{m} H_j\right) \circ I \;\; , \alpha > 0.$$

If we choose $V \in \mathcal{U}$ which diagonalizes $\sum_{j=1}^{m} H_j$, than since

$$\left(\sum_{j=1}^{m} H_j\right)(V) = \left(\sum_{j=1}^{m} H_j\right)(V) \circ I = \left(\sum_{j=1}^{m} H_j(V)\right) \circ I$$

$$\log\left(e^{\alpha H_1} \circ e^{\alpha H_2} .. \circ e^{\alpha H_m}\right)^{\frac{1}{\alpha}} \geq \left(\sum_{j=1}^{m} H_j(V)\right) \circ I$$

and moreover for Hermitian matrices X and Y order relation

$$X \geq Y$$

implies:

$$\lambda_i(X) \geq \lambda_i(Y), \quad i = 1, 2, .., n.$$

where $\lambda_1(X) \geq \lambda_2(X) \geq ... \geq \lambda_n(X)$, are the eigenvalues of X, counted with multiplicities and arranged in non-increasing order, we have also that for any nonnegative non-decreasing function $g(t)$

$$\lambda_i(g(X)) = g(\lambda_i(X)) \geq g(\lambda_i(Y)) \geq \lambda_i(g(Y)) \geq 0.$$

Applying this to function $g(t) = e^t$ we derive:

$$\left(e^{\alpha H_1} \circ e^{\alpha H_2} ... \circ e^{\alpha H_m}\right)^{\frac{1}{\alpha}} \geq e^{\left(\sum_{j=1}^{m} H_j\right)(V)}$$

that gives the relaxation:

$$Tr\left(\left(e^{\alpha H_1} \circ e^{\alpha H_2} ... \circ e^{\alpha H_m}\right)^{\frac{1}{\alpha}}\right) \geq Tr\left(e^{\left(\sum_{j=1}^{m} H_j\right)(V)}\right) = Tr\left(e^{\sum_{j=1}^{m} H_j}\right)$$

■

To prove that the presented bound is a pretty good bound for the $Tr\left(e^{\sum_{j=1}^{m} H_j}\right)$ we would need to show the convergence

$$\lim_{\alpha \to 0} Tr\left(\left(e^{\alpha H_1} \circ e^{\alpha H_2} ... \circ e^{\alpha H_m}\right)^{\frac{1}{\alpha}}\right) = Tr\left(e^{\sum_{j=1}^{m} H_j}\right).$$

Proof of the similar fact can be found in *[6],* and we will leave out the rest of details here.

2.4 The Hadamard Updates

Unlike previous section where we have seen updates that depend on the cumulative history of the adversary's responses H_i, in this section we prove that there exists an algorithm such that: *(i)* expected overall loss is as small as with the cumulative updates, *(ii)* it is less computationally demanding, *(iii)* its iterative formula depends only on most recent response from the other party. Iterative updates come as a clear advantage as we do not need to follow the history of matrices H_i and to sum them before the exponentiation to get the state V_{i+1}. Let us denote with J a matrix whose all entries are equal to one, and in contrast to the cumulative updates we define matrices $0 \leq H_i \leq 1$ to be gains, rather then losses.

Matrix Multiplicative Weights Algorithm with Iterative Hadamard Updates

Fix an $\epsilon < \frac{1}{2}$ and denote with $\epsilon' = \ln(1+\epsilon)$, and define $\rho_1 = V_1 = \frac{1}{n}J$. For $t = 1, 2, ..T$ do the following:

1. Compute $V_{t+1} = V_t \circ e^{\epsilon' H_t} = V_t \circ (1+\epsilon)^{H_t}$
2. Play $\xi_{t+1} = \frac{V_{t+1}}{Tr(V_{t+1})}$ and observe the column player next move H_{t+1}.

Theorem 26 *The Matrix Multiplicative Weights algorithm with iterative updates generates density matrices $\rho_1, \rho_2, ..., \rho_T$ such that:*

$$\sum_{t=1}^{T} H_t \cdot \xi_t \geq \max\left\{(1-\epsilon)\,\lambda_{\max}\left(\sum_{t=1}^{T} H_t\right) - \frac{\ln n}{\epsilon},\ (1-\epsilon)\,\lambda_{avg}\left(\sum_{t=1}^{T} H_t\right)\right\} \tag{2.5}$$

where $\lambda_{avg}\left(\sum_{t=1}^{T} H_t\right)$ denotes the strategy of the row player that would achieve the average gain.

Proof. The proof that we present here is based on the theorem presented in the introductory section. At the round $T+1$ the expected loss $Tr(V_{T+1})$ can be bounded from above as follows:

$$
\begin{aligned}
Tr\left(V_{T+1}\right) &= Tr\left(V_T \circ \exp^{\epsilon' H_T}\right) \\
&= Tr\left(V_T \circ (1+\epsilon)^{H_T}\right) \\
&= Tr\left(V_T \circ \left(1 + \log(1+\epsilon) H_T + \frac{\log^2(1+\epsilon)}{2} H_T^2 + ..\right)\right) \\
&\leq Tr\left(V_T \circ (1+\epsilon H_T)\right) \\
&= Tr\left(V_T\right)\left(1 + \epsilon \frac{Tr\left(V_T \circ H_T\right)}{Tr\left(V_T\right)}\right) \\
&\leq Tr\left(V_T\right)\left(1 + \epsilon \frac{Tr\left(V_T H_T\right)}{Tr\left(V_T\right)}\right) \\
&= Tr\left(V_T\right)\left(1 + \epsilon Tr\left(\xi_T H_T\right)\right) \\
&= Tr\left(V_T\right)\left(1 + \epsilon \xi_T \cdot H_T\right) \\
&\leq Tr\left(V_T\right) \exp\left(\epsilon H_T \cdot \xi_T\right) \\
&\leq Tr\left(V_{T-1}\right) \exp\left(\epsilon H_T \cdot \xi_T + \epsilon H_{T-1} \cdot \xi_{T-1}\right) .. \\
&\leq \exp\left(\epsilon \sum_{i=1}^{T} H_i \cdot \xi_i\right)
\end{aligned}
$$

The first inequality follows from the fact that $(1+\epsilon)^A = \exp(\epsilon' A) \preceq (I + \epsilon A)$ for $0 \preceq A \preceq 1$, and the second inequality follows from Lemma 24

Since we can start from $V_1 = \frac{1}{n} J$ it follows $Tr\left(V_1\right) = 1$. Again, $H_t \cdot \xi_t$ are just real numbers that commute therefore, iterative application of the inequality we have derived above yields:

$$
Tr\left(V_{T+1}\right) \leq Tr\left(V_1\right) \exp\left(\epsilon \sum_{t=1}^{T} H_t \cdot \xi_t\right).
$$

Moreover according to the theorem 25:

$$
\begin{aligned}
Tr\left(V_{T+1}\right) &= Tr\left(V_T \circ e^{\epsilon H_T}\right) \\
&= \frac{1}{n} Tr\left(e^{\epsilon' H_1} \circ e^{\epsilon' H_2} ... \circ e^{\epsilon' H_T}\right) \\
&\geq \frac{1}{n} Tr\left(e^{\epsilon' \sum_{i=1}^{T} H_i}\right) \\
&\geq \begin{cases} \frac{1}{n} \exp\left(\epsilon' \lambda_{\max}\left(\sum_{t=1}^{T} H_t\right)\right) \\ \exp\left(\epsilon' \frac{1}{n} \sum_{i=1}^{n} \lambda_i \left(\sum_{t=1}^{T} H_t\right)\right) \end{cases} .
\end{aligned}
$$

Finally, the last inequality follows since the arithmetic mean of positive real numbers is not smaller than the geometric mean of those numbers. Composing lower and upper bound and taking the logarithm of both sides we have:

$$\left.\begin{array}{c} \frac{1}{n}\exp\left(\epsilon'\lambda_{\max}\left(\sum_{t=1}^{T}H_t\right)\right) \\ \exp\left(\epsilon'\lambda_{avg}\left(\sum_{t=1}^{T}H_t\right)\right) \end{array}\right\} \leq \exp\left(\epsilon\sum_{t=1}^{T}H_t\cdot\xi_t\right)$$

$$\left.\begin{array}{c} -\ln(n)+\epsilon'\lambda_{\max}\left(\sum_{t=1}^{T}H_t\right) \\ \epsilon'\lambda_{avg}\left(\sum_{t=1}^{T}H_t\right) \end{array}\right\} \leq \epsilon\sum_{t=1}^{T}H_t\cdot\xi_t$$

$$\left.\begin{array}{c} -\frac{\ln(n)}{\epsilon}+\frac{\ln(1+\epsilon)}{\epsilon}\lambda_{\max}\left(\sum_{t=1}^{T}H_t\right) \\ \frac{\ln(1+\epsilon)}{\epsilon}\lambda_{avg}\left(\sum_{t=1}^{T}H_t\right) \end{array}\right\} \leq \sum_{t=1}^{T}H_t\cdot\xi_t$$

$$\left.\begin{array}{c} -\frac{\ln(n)}{\epsilon}+(1-\epsilon)\lambda_{\max}\left(\sum_{t=1}^{T}H_t\right) \\ (1-\epsilon)\lambda_{avg}\left(\sum_{t=1}^{T}H_t\right) \end{array}\right\} \leq \sum_{t=1}^{T}H_t\cdot\xi_t$$

as claimed. ■

Interestingly, the above proof can not be reproduced when the observed events are losses, in which case we would have:

$$V_{t+1}=V_t\circ e^{-\epsilon' H_t}=V_t\circ(1-\epsilon)^{H_t}.$$

This comes from the fact that even though matrices H_i are non-negative by definition, the matrix $(1-\epsilon)^{H_i}\approx(I-\epsilon H_i)$ certainly is not, and therefore the following step in the above proof would fail:

$$Tr\left(V_T\right)\left(1+\epsilon\frac{Tr\left(V_T\circ H_T\right)}{Tr\left(V_T\right)}\right)\leq Tr\left(V_T\right)\left(1+\epsilon\frac{Tr\left(V_T H_T\right)}{Tr\left(V_T\right)}\right)$$

Finally we revert to the theorem 14 that we have not proved in the previous chapter that the following update rule

$$W_{t+1}=W_t+\epsilon\sqrt{M_t}W_t\sqrt{M_t}$$

provides the performance with the worst case gain:

$$\sum_{t=1}^{T}M_t\cdot\rho_t\geq\max\left\{(1-\epsilon)\lambda_{\max}\left(\sum_{t=1}^{T}M_t\right)-\frac{\ln n}{\epsilon},\,(1-\epsilon)\lambda_{avg}\left(\sum_{t=1}^{T}M_t\right)\right\} \tag{2.6}$$

Proof. (Of the Theorem 14) Now we prove the corollary for the matrix multiplicative weights algorithm with iterative updates. It is straightforward to see that the same bound that we had above would apply to this algorithm as well. This comes from the fact that

$$Tr\left(W_t + \epsilon\sqrt{M_t}W_t\sqrt{M_t}\right) = Tr\left(W_t + \epsilon M_t W_t\right) = Tr\left(W_t + \epsilon W_t M_t\right)$$

and the same line of reasoning before and beyond line 2.2 in the proof that we have presented above. For the sake of the completeness we present the whole argument.

$$\begin{aligned}
& Tr\left(W_T\left(1+\epsilon M_T\right)\right) \\
\geq\ & Tr\left(\left(W_{T-1}\left(1+\epsilon M_{T-1}\right)\right)\circ\left(1+\epsilon M_T\right)\right) && (2.7)\\
\geq\ & Tr\left(W_{T-1}\circ\left(1+\epsilon M_{T-1}\right)\circ\left(1+\epsilon M_T\right)\right).. && (2.8)\\
\geq\ & \frac{1}{n}Tr\left(\left(1+\epsilon M_1\right)\circ..\circ\left(1+\epsilon M_{T-1}\right)\circ\left(1+\epsilon M_T\right)\right) && (2.9)\\
\geq\ & \frac{1}{n}Tr\left(\left(1+\epsilon\right)^{M_1}\circ..\circ\left(1+\epsilon\right)^{M_{T-1}}\circ\left(1+\epsilon\right)^{M_T}\right) && (2.10)\\
=\ & \frac{1}{n}Tr\left(\exp\left(\epsilon' M_1\right)\circ..\circ\exp\left(\epsilon' M_{T-1}\right)\circ\exp\left(\epsilon' M_T\right)\right) && (2.11)\\
\geq\ & \frac{1}{n}Tr\left(\exp\left(\epsilon'\sum_{\tau=1}^{T}M_\tau\right)\right) && (2.12)\\
\geq\ & \frac{1}{n}\exp\left(\epsilon'\lambda_{\max}\left(\sum_{\tau=1}^{T}M_\tau\right)\right) && (2.13)
\end{aligned}$$

Moreover, we can find the upper bound for $Tr\left(W_T\left(1+\epsilon M_T\right)\right)$ as follows:

$$\begin{aligned}
& Tr\left(W_T\left(1+\epsilon M_T\right)\right) \\
= \;& Tr\left(W_T\right)\left(1+\epsilon\frac{Tr\left(W_T M_T\right)}{Tr\left(W_T\right)}\right) && (2.14)\\
= \;& Tr\left(W_T\right)\left(1+\epsilon Tr\left(\rho_T M_T\right)\right) \\
= \;& Tr\left(W_T\right)\left(1+\epsilon\rho_T\cdot M_T\right) \\
= \;& Tr\left(W_{T-1}\left(1+\epsilon M_{T-1}\right)\right)\exp\left(\epsilon M_T\cdot\rho_T\right) && (2.15)\\
= \;& Tr\left(W_{T-1}\right)\left(1+\epsilon\frac{Tr\left(W_{T-1}M_{T-1}\right)}{Tr\left(W_{T-1}\right)}\right)\exp\left(\epsilon M_T\cdot\rho_T\right) && (2.16)\\
= \;& Tr\left(W_{T-1}\right)\left(1+\epsilon\rho_{T-1}\cdot M_{T-1}\right)\exp\left(\epsilon M_T\cdot\rho_T\right) && (2.17)\\
\leq \;& Tr\left(W_{T-1}\right)\exp\left(\epsilon M_{T-1}\cdot\rho_{T-1}+\epsilon M_T\cdot\rho_T\right) && (2.18)\\
\leq \;& ..\leq Tr\left(W_1\right)\exp\left(\epsilon\sum_{t=1}^{T}M_t\cdot\rho_t\right) && (2.19)\\
= \;& \exp\left(\epsilon\sum_{t=1}^{T}M_t\cdot\rho_t\right) && (2.20)
\end{aligned}$$

Composing the upper and lower bound in the same way we have seen in the previous chapter yields the following result:

$$\sum_{t=1}^{T}M_t\cdot\rho_t\geq\max\left\{\left(1-\epsilon\right)\lambda_{\max}\left(\sum_{t=1}^{T}M_t\right)-\frac{\ln n}{\epsilon},\left(1-\epsilon\right)\lambda_{avg}\left(\sum_{t=1}^{T}M_t\right)\right\}.$$

■

Computational complexity for this updates $\tilde{O}\left(n^2\right)$ rather that $O\left(n^3\right)$ but this is of secondary importance as it is sufficient to generate approximate updates and that can be done in time that is linear in the sparsity of matrices in the updates, what is usually very efficient.

Remark 27 *It is important to note that it is necessary to provide the updates through unitarily invariant form $\sqrt{M_t}W_t\sqrt{M_t}$ rather than M_tW, as the later one has no warranties of being positive definite, what is the necessary condition with respect to ρ_t.*

Corollary 28 *The following matrix multiplicative weight algorithm with iterative Hadamard updates achieves worst case gain lower bound*

$$\sum_{t=1}^{T}H_t\cdot\xi_t\geq\max\left\{\left(1-\epsilon\right)\lambda_{\max}\left(\sum_{t=1}^{T}H_t\right)-\frac{\ln n}{\epsilon},\left(1-\epsilon\right)\lambda_{avg}\left(\sum_{t=1}^{T}H_t\right)\right\}.$$

Matrix Multiplicative Weights Algorithm with Iterative Hadamard Updates

Fix an $\epsilon < \frac{1}{2}$ and denote with $\epsilon' = \ln(1+\epsilon)$, and define $\rho_1 = V_1 = \frac{1}{n}J$

For $t = 1, 2, ..T$ do the following:

1. Compute $V_{t+1} = V_t \circ (I + \epsilon H_t)$

2. Play $\xi_{t+1} = \frac{V_{t+1}}{Tr(V_{t+1})}$ and observe the column player next move H_{t+1}.

Proof. Follows straightforwardly from the proof of the Theorem 14. Clearly every step in the update take $\mathcal{O}(n^2)$ time. ■

2.5 Quantum Discrepancy Against Multiplicative Weights Algorithm

In this section we elaborate the lack of compatibility between multiplicative weights algorithm and unitary quantum evolution. Motivation behind this discrepancy remark came from perhaps counter intuitive resemblance between multiplicative weights updates algorithm and the first order dynamical systems. The two formulas differ only by a imaginary constant, i, that multiplies the gradient term in the update term.

In the classical case vector weighted majority algorithm is the following.

At every step t, we have a weight $w_i(t)$ assigned to expert i.. Initially for all i, $w_i(0) = 1$. At a step $t+1$, for each i such that expert i was found to have predicted the stock value correctly, we set the update rule to be:

$$w_i(t+1) = w_i(t)(1+\epsilon_c) \tag{2.21}$$

and then renormalize all the weights with $\sum_{i=1}^{N} w_i(t+1)$. Once we update the weights on all experts our own prediction for the stock value for step $t+1$ is the opinion of a weighted majority of the experts. In other words, if the total weight of all experts predicting "up" is at least $\frac{1}{2}$ then we predict "up" as well and otherwise we predict "down." Here we denoted with $\epsilon_c << 1$ the epsilon for the classical algorithm.

We can rewrite the classical weighted majority algorithm update rule as the following vector equation, that represents current state of our knowledge:

$$w(t+1) = w(t) + M_c w(t) \approx e^{M_c} w(t) \tag{2.22}$$

where we denoted with w the N dimensional vector that describes our current state of the knowledge, and the *diagonal* matrix M_c is:

$$M_c = \begin{cases} m_{i,i} = \epsilon_c, \text{ if expert } i \text{ predicts correctly} \\ 1, \text{ otherwise} \end{cases} \tag{2.23}$$

Now we can define the same state of knowledge but rather through the quantum vector $|\xi(t+1)\rangle$ whose density matrix $\rho_\xi(t+1) = |\xi(t+1)\rangle\langle\xi(t+1)|$ corresponds to the first order approximation of classical state of knowledge vector $w(t+1)$:

$$w(t+1) \approx I \circ \rho_\xi(t+1) = I \circ |\xi(t+1)\rangle\langle\xi(t+1)| \tag{2.24}$$

As we saw earlier state of the system is changed according to the Schroedinger equation, for (backward) time corespondent to the small change ϵ:

$$|\xi(t+1)\rangle = |\xi(t)\rangle + \frac{i}{\hbar}\Delta_\epsilon H|\xi(t)\rangle = \left(1 + \frac{i}{\hbar}\Delta_\epsilon H\right)|\xi(t)\rangle \tag{2.25}$$

that yields a unitary update:

$$|\xi(t+1)\rangle = e^{+\frac{i}{\hbar}\Delta_\epsilon H}|\xi(t)\rangle \tag{2.26}$$

or equivalently

$$\rho_\xi(t+1) = e^{\frac{i}{\hbar}\Delta_\epsilon H}\rho_\xi(t)\, e^{-\frac{i}{\hbar}\Delta_\epsilon H^\dagger}. \tag{2.27}$$

Increase in the absolute value of one diagonal coordinate of $\rho_\xi(t)_{i,i}$ would imply a decrease in the some other coordinate $j \neq i$, as this is by definition a change of basis, no matter which nondiagonal Hamiltonian $H_{N\times N}$ we employ to do that. Therefore in this setup, without the ancillae qubits and apart from the first iteration in experts algorithm, it seems nontrivial to implement the basic mechanism of classical multiplicative weights algorithm. This is due to the fact that the basic mechanism of the experts algorithm is the following: increase the weight of the good experts while keeping the weight of every other expert the same. That is precisely the story, that the difference in the imaginary constant in the exponent, in the following two expressions, is trying to tell:

$$w(t+1) = e^{M_c} w(t) \tag{2.28}$$

$$|\xi(t+1)\rangle = e^{iM_q}|\xi(t)\rangle \tag{2.29}$$

where we denoted with $M_q = \frac{\Delta_\epsilon H}{\hbar}$.

2.6 Convergence to Nash Equilibrium in the Non-Zero Sum Games

Convergence to Nash equilibrium is one of the central issues in the evaluation of the prediction and learning algorithms. In the early days of the game theory and Robinson [21] showed that in two player zero-sum game with finite number of strategies, where two players play the best response in every round of a zero sum game, players will eventually converge to a distribution over the strategies. That distribution will probabilities that correspond to a Nash equilibrium of the underlying game. Since then there have been a great deal of progress in terms of improving the rate at which certain algorithms converge. A good overview of the historical bibliographic remarks can be found [13] and [14].

The situation with the convergence to a Nash distribution in non-zero sum games is different and much less studied. Singh, Kearns and Mansour [4] showed, for the most common type of algorithms that use gradient ascent or descent on the objective function, although the strategies of the players may not always converge, their average payoffs always do converge to the expected payoffs of some Nash equilibrium. Their study is however limited to two person, two action non-zero sum game. This result therefore implies that dynamics of gradient ascent ensure that the average payoffs to two players adopting this simple strategy is the same as the payoff they would achieve by adopting arbitrary complex strategies. Interestingly, this result is achieved by showing that gradient ascent algorithm of the players can be modeled as an affine dynamical system.

Even though the method of Singh, Kearns and Mansour seems generalizable, until recently nothing was known even for two player three strategies non-zero sum game. Daskalakis, Frongillo, Papadimitriou, Pierrakos and Valiant have proved that the learning algorithm, i.e. vector multiplicative weights algorithms do not converge to Nash equilibria in non-zero sum games. They have considered two player three strategy game non-zero sum game, i.e. a Shapley game that we will see in the next section.

The algorithms that we have explained here are external regret algorithms that compare the performance of an online algorithm, selecting among N actions, to the performance of the best of those actions in hindsight. There also exist the algorithms, that we have not studied here, that are called the internal regret algorithms since they compare the loss of an online algorithm to the loss of a modified online algorithm, which consistently replaces one action by

	P	R	S
p	0,0	0,1	1,0
r	1,0	0,0	0,1
s	0,1	1,0	0,0

Table 2.1: Payoffs for the Paper-Rock-Scissor zero-sum game.

another.

Consideration of non-zero sum games is interesting in various aspects. Namely, since in contrast to the zero-sum games where the min-max value is the only interesting value, in a non zero-sum game, there are at least three interesting values for the first agent.

1. The safe value, or min-max value, that a no-external regret (external regret vanishes over the time) algorithm is guaranteed regardless of whether it plays against another learning agent or arbitrary strategy.

2. The minimum correlated equilibrium value, or the lowest expected value for the first agent in any correlated equilibrium of the single-shot game, which a no-internal regret algorithm is guaranteed when it plays another no-internal regret algorithm.

3. The minimum Nash equilibrium value, or the lowest expected value for the first agent in any Nash equilibrium of the single-shot game.

Apart form this consideration in the section that follows we present the evidence that matrix multiplicative weights algorithm might have better convergence properties than the analogous vector multiplicative weights.

2.6.1 The Shapely's Game

The good example of the non-zero sum game, for which many learning and prediction algorithms show divergence away from Nash equilibrium in strategy, is a Shapley game. It is basically a non-zero sum analog of the paper-rock-scissors game. The actual payoff for Shapley's game are in the table 2.1.

This game has only one Nash equilibrium that is all uniform strategy.

One of early discovered algorithms for zero-sum version of this game (paper-rock-scissors game) is a fictitious play. That algorithm for Shapley game produces a play of strategies in the asymptotic cycles. It does not converge and number of iterations to get to the next cycle in strategies is increasing exponentially.

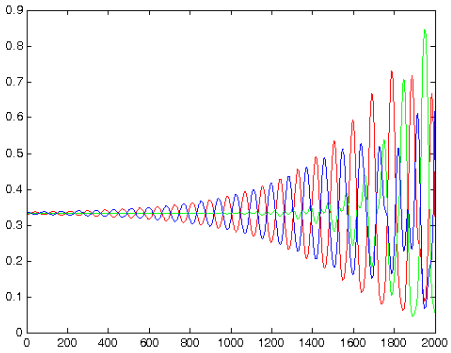


Figure 2.1: Vector multiplicative update algorithm diverges from the Nash strategy.

Similar, divergence effect is present with the vector multiplicative weight algorithm that gradually diverges from the strategy that is a Nash equilibrium, as shown on the figure 2.1.

The same does not hold in the case of the matrix multiplicative weight algorithm applied on the Shapley's game.

Weights in this case the same uniformly distributed between the strategies, rasing the question whether the eventual distribution over the strategies is dependent only on the eigenvalues of the game matrix. In Shapley game eigenvalues of the payoff matrix are all the same and equal one.

2.6.2 The Augmented Shapley Game

The example of the Shapley's game that we have seen in the previous section has a simple Nash equilibrium, for which the players need to figure out that there is a mixed strategy that provides the optimal payoff. The game that we consider in this section [[5]] has a bit different setup. The following loss matrix defines an augmented Shapley game.

Clearly in this case the players have a choice of playing the Shapley game or strategy g(or G). The difficulty in this non-zero sum game is in the fact that players have to learn that they are better off by playing g (or G) strategy since they will be paying more but also receiving as much. Therefore the only Nash

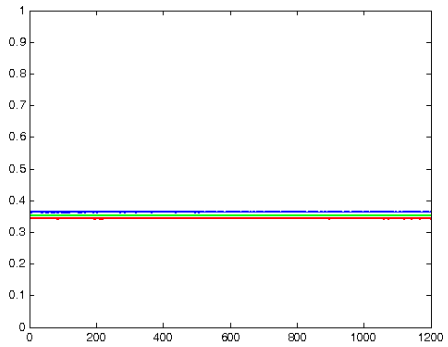
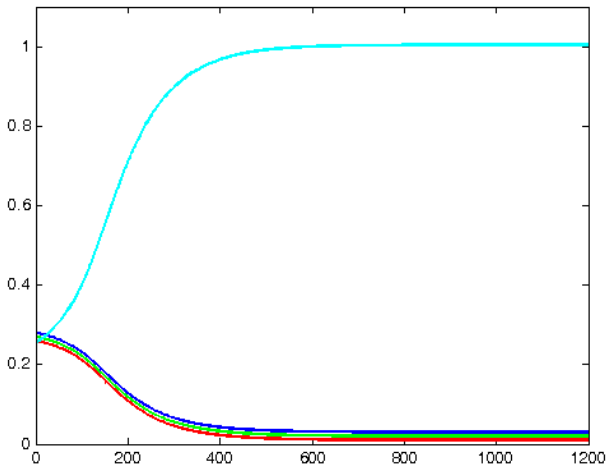


Figure 2.2: Matrix multiplicative algorithm does not diverge from the Nash strategy in the Shapley's game. Small distance in the probabilities is intentionally introduced to avoid overlapping.

	P	R	S	G
p	0, 0	0, 1	1, 0	2,0.4
r	1, 0	0,0	0, 1	2,0.4
s	0, 1	1, 0	0, 0	2,0.4
g	0.4, 2	0.4, 2	0.4, 2	3,3

Table 2.2: Payoffs for the Augmented Shapley's Game

equilibrium is in this case g-G strategy. Zinkevich [5] showed that any no-internal regret algorithm will necessarily fail to converge to Nash equilibrium. Moreover the vector multiplicative weights algorithms does not converge and their performance is nicely illustrated for the concrete examples, as well, in.[5].



Convergence of MMWA the Nash equilibrium of the augmented Shapley game.

In general, the min-max value is always less than or equal to the minimum correlated equilibrium value, which is always less than or equal to the minimum Nash equilibrium value. In the Augmented Shapley Game, the min-max value and the minimum correlated equilibrium value are both 0.4. The minimum Nash equilibrium value is 3. Thus, in a nonzero-sum game, getting the minimax value or minimum correlated equilibrium value is not always a very helpful guarantee. Nevertheless, we are here only interested in convergence to a Nash equilibrium value.

Perhaps surprisingly, matrix multiplicative weight algorithm exhibits a different behavior, as it in fact converges to a strategy that defines Nash equilibrium.

A definite answer to a natural question that arises after this results, whether a off-diagonal elements in the matrix multiplicative weight algorithm play a crucial roll in the convergence property, is yet to be determined. The result that we have seen here is the ramification of the fact that the matrix multiplicative weights, unlike vector updates, update the weights of the strategies with respect to the all possible opponent strategies at once. In contrast to the vector multiplicative updates, the exact move that the other party takes

is not considered with the matrix updates since eventually the eigenvalues, or diagonal entries, of the game matrix are going to determine the convergence to a distribution, that at least in our case turns to be the Nash equilibrium.

2.7 Concluding Remarks

In this chapter we have elaborated several results regarding the matrix multiplicative weights algorithm. In particular we have improved a generalization and minor improvement of the algorithms upper bound, showed that there exist iterative version of the matrix multiplicative weights update algorithm with the same performance promise and established the exact connection with the vector matrix multiplicative algorithm. This connection clarifies the origin of the same type of bound that is achieved by both matrix and vector multiplicative updates algorithm.

Even though it is computationally more demanding for the same performance bound, matrix multiplicative weight algorithm, unlike its vector version, showed a distinctive convergence properties to Nash equilibrium of the two player, three strategy, non-zero sum game. Further study of the convergence of this algorithms is necessary in order to provide a definitive answer to the convergence question in non-zero sum games. Currently it is not known whether matrix multiplicative updates algorithms converge in the multi-agent, multi-strategy non-zero sum games. Results by Sigh, Kearns and Monsour [4] seem to provide some hope for the positive results in further research and be directly applicable to the setup with more then two available strategies.

Another very interesting relation, that is in our opinion worth further study, with the respect to the above result, is to establish connection between SDPs and graph covering. This work might provide a meeting point for these to seemingly unrelated concepts. Namely, natural existence of the random projections are present in both multiplicative updates and the proof for the graph covering time [20].

Bibliography

[1] S. Arora and S. Kale, *A Combinatorial, Primal-Dual Approach to Semi-definite Programing*, STOC 2007;

[2] S. Arora, E. Hazan, S. Kale, *The Multiplicative Weights Update Method: a Meta Algorithm and Applications*, manuscript

[3] R. Jain and J. Watrous, *Parallel Approximation of Non-interactive Zero-sum Quantum Games*, quant-ph:0808.2775;

[4] S. Singh, M. Kearns, Y. Mansour, *Nash Convergence of Gradient Dynamics in General-Sum Games,* Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, 2000;

[5] M. Zinkevich, *Theoretical Guarantees for Algorithms in Multi-Agent Settings*, CMU Dissertation 2004;

[6] T. Ando, *Hadamard Products and Golden-Thompson Type Inequalities*, Linear Algebra and its Applications, 241-243: 105-112 (1996);

[7] Y. Seo, *Reverses of the Golden-Thompson Type Inequalities Due to Ando-Hiai-Petz*, Banach Journal of Mathematical Analysis 2, 2008, no. 2, 140-149;

[8] Y. Liu, *Gibbs States and the Consistency of Local Density Matrices*, quant-ph/0603012v1;

[9] Y. Liu, *Consistency of Local Density Matrices is QMA-complete*, quant-ph/0604166v3;

[10] C. Daskalakis, R. Frongillo, C. Papadimitriou, G. Pierrakos and G. Valiant, *On Learning Algorithms for Nash Equilibria*, personal communication;

[11] M. Nielsen and I. Chuang, *Quantum Computation*, Cambridge University Press, 2001

[12] H. Guo, J. Zhang and G. Koehler, *A survey of quantum games,* Decision Support Systems 46 (2008) 318–332

[13] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning and Games,* Cambridge University Press, 2006

[14] N. Nisan *at el, Algorithmic Game Theory*, Cambridge University Press, 2007

[15] R. Aumann and S. Hart, *Handbook of Game Theory*, North-Holland, 1992

[16] R. Bhatia, *Matrix Analysis,* Springer, 1996

[17] J. von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1928):295–320, 19

[18] R. Bapat, *Nonnegative Matrices and Applications,* Cambridge University Press, 1997;

[19] R. Beezer, *A First Course in Linear Algebra*, A free textbook, 2007

[20] J. Lee, *Complexity of Graph Coverings*, UC Berkeley theory lunch, April 2010;

[21] J. Robinson, *An Iterative Method of Solving a Game*, Annals of Mathematics 54, 296-301, 1951

Chapter 3

On the Quantum Circuit Complexity Equivalence by the Geometric Arguments

Nielsen [3] recently asked the following question: "What is the minimal size quantum circuit required to exactly implement a specified n-qubit unitary operation U, without the use of ancilla qubits?" Nielsen was able to prove that a lower bound on the minimal size circuit is provided by the length of the geodesic between the identity I and U, where the length is defined by a suitable Finsler metric on $SU(2^n)$. We prove that the minimum circuit size that simulates U is in linear relation with the geodesic length and simulation parameters, for the given Finsler structure F. As a corollary we prove the highest lower bound of $O(\frac{n^4}{p} d^2_{F_p}(I,U) L_{F_p}(I,\tilde{U}))$ and the lowest upper bound of $\Omega(n^4 d^3_{F_p}(I,U))$, for the standard simulation technique. Therefore, our results show that by standard simulation one can not expect a better then n^2 times improvement in the upper bound over the result from Nielsen, Dowling, Gu and Doherty [4]. Moreover, our equivalence result can be applied to the arbitrary path on the manifold including the one that is generated adiabatically.

3.1 Introduction

Quantum computation is inherently a process of continuous evolution of quantum states that has the potential to fundamentally change the notion of feasibly tractable computation. Only recently did researchers start to think how notions from the differential geometry [13] can be used to represent this

process. Instead, quantum circuits, as an inherently discrete notion of computation, have been most commonly used to represent this continuous process. Any quantum operation, a unitary matrix $U \in SU(2^n)$, is an element of a Lie group, and a point on the $\mathcal{U} \equiv SU(2^n)$ manifold, whose tangent bundle can be endowed with the Finsler structure F, that effectively provides a measure of length for any path on the Finsler manifold $(\mathcal{U}, F)$. In particular, the paths that we are interested in are geodesics. These are locally and, under certain conditions, globally minimal length paths between any two points on the manifold. They are of particular interest, because if closely simulated they can give the smallest circuit complexity for the given unitary U.

The aim in the approach that we take here is to tackle the question about the complexity of the circuit necessary for the simulation of an arbitrary unitary gate. As the length of the geodesic for the particular unitary is its intrinsic property, ideally one would succeed in finding the minimum number of circuits necessary to implement the unitary by simulating exactly its geodesic. Therefore, the hope here is to learn about the circuit lower bounds by basically transforming the hard combinatorial optimization problems over large sets to the problems in continuous domain that can be solved with tools of differential geometry and the calculus of variations.

One of the first results that had the flavor of this transform was introduced by Mochon [5], who proved that in the discrete model and the analogous continuous model, i.e. the Hamiltonian oracle model, Oracle interrogation, the problem of computing XOR and Grover search have the same complexity. Moreover, Nielsen [3] and subsequently Nielsen, Dowling, Gu and Doherty [4] proved that for particularly chosen metric there is a polynomial equivalence between the geodesic length and number of gates necessary for the simulation. The lower bound for the minimum number of gates necessary for the simulation has been determined for exact simulation and the upper bound has been determined for the arbitrary precision. The metric chosen in [4] penalizes all those directions on the manifold that are not easily simulated by local gates, so that coefficients for stabilizer elements of Hamming weight greater than two bear high cost, i.e. have longer paths.

In this paper we prove the stronger result and show the exact upper and lower bound that determine the equivalence between the minimal number of gates in the standard circuit simulation and the length of the geodesic. Both upper and lower bound are determined by the simulation parameters and, of course, the length of the geodesic.

We consider the two cases. First: the simulation of the geodesic with set

of gates $\mathcal{G}$ that is exactly universal, and the case with approximately universal set of gates. With the exactly universal set of gates for any point $x_0 \in \mathcal{U}$ there is a gate in a set $\mathcal{G}$ by which we can simulate exactly any point $x_1 \in \mathcal{U}$ in the ball of radius r centered around x_0; we denote this ball as $B^+_{x_0}(r)$. Under that assumption, we prove that the number of gates in the simulation of a geodesic may be upper and lower bounded by a linear factor in the length of geodesic and simulation parameters.

When the set of gates, $\mathcal{G}_\epsilon$, is approximately universal, a single gate from this set can simulate the points in $B^+_{x_0}(r)$ only with some finite precision ϵ, and that will necessarily mean that the circuit that simulates the geodesic is doing so along the path that is not shorter than the actual geodesic, for that very point.

Our aim here is not to elaborate on the algorithm for the geodesic simulation but rather to prove the bounds that optimal simulation can achieve. We say optimal, because the set of gates $\mathcal{G}$ that we first consider is much more powerful than any local and universal set of gates. Therefore the result that we present is the optimal result about complexity equivalence between discrete and continuous notions of computation. In particular, for the standard simulation model described in [4], we derive the highest lower bound and the lowest upper bound in the minimal circuit complexity that one can hope to achieve with the simulation of a geodesic.

3.2 Preliminaries

A quantum operation $U \in \mathcal{U}$ is a point on the manifold $\mathcal{U} \equiv SU(2^n)$ at some distance from identity $I \in \mathcal{U}$. The distance considered is the integral distance that is determined by the structure used on the manifold. In general that structure may be more general than Riemannian, i.e. it is called the Finsler structure $F(x, y)$. The restriction of a Finsler structure F to any specific tangent space $T_x\mathcal{U}$ with the origin at the point $x \in \mathcal{U}$ is called Minkowski norm on $T_x\mathcal{U}$. The second argument of the structure $F(x, y)$ is the velocity and its definition follows. Therefore a Finsler structure is basically a family of the smoothly varying Minkowski norms, one for each tangent space.

The defining properties of a non-negative real-valued structure $F(x, y)$ on $\mathbb{R}^{4^n-1}$ are as follows:

(1) it is C^∞ anywhere on $\mathbb{R}^{4^n-1}$ except at $y = 0$;

(2) it is positive homogeneous, i.e. $F(x, \lambda y) = \lambda F(x, y)$ for $\lambda > 0$;

(3) the $(4^n-1)\times(4^n-1)$ matrix $\frac{\partial}{\partial y_i}\frac{\partial}{\partial y_j}[\frac{1}{2}F^2]$ is positive definite unless $y=0$. As a consequence, one can derive positivity and triangular equality of Minkowski norms [6]. The structure $F(x,y)$ is usually denoted simply as $F(y)$.

For any $a,b\in\mathbb{R}_+$ we say that a map $\sigma:[a,b]\to\mathcal{U}$ is a piecewise C^∞ curve with velocity $y\equiv\frac{d\sigma}{dt}=\sum_i\frac{d\sigma_i}{dt}\frac{\partial}{\partial x_i}\in T_{\sigma(t)}\mathcal{U}$. The integral length of the curve σ, $L(\sigma)$, is defined as:

$$L_F(\sigma)=\int_a^b F(\sigma,\frac{d\sigma}{dt})dt\ . \tag{3.1}$$

Since we are usually interested in minimum length curves for $x_0,x_1\in\mathcal{U}$, we denote by $\Gamma(x_0,x_1)$ the collection of all piecewise C^∞ curves $\sigma:[a,b]\to\mathcal{U}$ such that $\sigma(a)=x_0$ and $\sigma(b)=x_1$. Similarly, the integral distance is defined as a map $d_F:\mathcal{U}\times\mathcal{U}\to[0,\infty)$:

$$d_F(x_0,x_1)=\inf_{\Gamma(x_0,x_1)}L_F(\sigma) \tag{3.2}$$

Using these definitions, one can show that the Finsler manifold $(\mathcal{U},d_F)$ satisfies the two axioms of a metric space: (1) positivity: $d_F(x_0,x_1)\geq 0$, where equality holds if and only if $x_0=x_1$ and (2) the triangular inequality: $d_F(x_0,x_2)\leq d_F(x_0,x_1)+d_F(x_1,x_2)$. In general, the symmetric property of a distance does not need to hold, and therefore $d_F(x_0,x_1)\neq d_F(x_1,x_0)$.

3.3 Distortion Lemma

To establish the equivalence result, we introduce in this section the main tool of our analysis. The intuitive idea on which we build our results relies on the relation between the distances on the manifold and the distances on the tangent space of the manifold. While the former are introduced by the unitary gates and their complexity, the latter are defined by the appropriately defined distances between the Hamiltonians of gates used in the simulation. This will be proven useful in the sections below.

The lemma that follows is a slightly stronger result of a well-known and very useful fact from the differential geometry. Again, it relates distances on the manifold with the minimum and maximum distortion of the Euclidian norm on the tangent space over the compact set. Interested reader are encouraged to consult [6], an excellent and very elaborate reference on this subject.

Lemma 29 (Distortion Lemma) *Let $(\mathcal{U}, F)$ be a Finsler manifold, and for any point $x \in \mathcal{U}$ let $\varphi : P_x \to \mathbb{R}^{4^n-1}$ be the local coordinate system diffeomorphism of a compact set P_x onto an open ball of $\mathbb{R}^{4^n-1}$, such that $\varphi(x) = 0$. Then for a given $x_0,\ x_1 \in P_x$, and any Finsler metric $F(x, y)$, there exist a constant minimum $\mathfrak{m} > 0$ and a constant maximum $\mathfrak{M} > 1$ such that the following relation is true:*

$$\mathfrak{m}|\varphi(x_1) - \varphi(x_0)| \leq L_F(x_0, x_1) \leq \mathfrak{M}\,|\varphi(x_1) - \varphi(x_0)|\ . \tag{3.3}$$

Here $|\varphi(x_1) - \varphi(x_0)|$ denotes the Euclidean length of the $4^n - 1$ dimensional vector in the tangent space.

Proof: We first note that a compact set P_x for which $\varphi(x) = 0$ always exists. This is true because, given a local coordinate system $\varphi : Q \to \mathbb{R}^{4^n-1}$ and $x \in Q$ for which $\varphi(x) = 0$, we can choose P_x to be a closure of the preimage of $\varphi^{-1}(B^{4^n-1}(r))$ for some $r > 0$. By $B^{4^n-1}(r) = \{v \in \mathbb{R}^{4^n-1} : |v| = \sqrt{\sum_i v_i^2} < r\}$ we denote the ball of radius r in the tangent space whose closure is a subset of $\varphi(Q)$.

Next we note that, for tangent vector $y = \sum_i y_i \frac{\partial}{\partial x_i} \equiv \frac{dx}{dt} \in T_x\mathcal{U}$, the ratio between Minkowski norm $F(x, y)$ and x-dependent Euclidean norm $|y| := \sqrt{\sum_i y_i^2}$ for the basis $\{\frac{\partial}{\partial x_i}\}$ is well defined for $y \neq 0$. Since both norms are positive continuous functions over the compact sets their quotient is also a positive continuous function. Therefore the quotient's minimum $\mathfrak{m}$ and maximum $\mathfrak{M}$ exist and are both positive: $0 < \mathfrak{m} \leq \frac{F(y)}{|y|} \leq \mathfrak{M}$. In other words, for all $y \in T_x\mathcal{U}$ and all $x \in P_x$:

$$\mathfrak{m}|y| \leq F(x, y) \leq \mathfrak{M}|y|. \tag{3.4}$$

Now we can prove the right hand side (RHS) of inequality (3.4) by choosing the path $\sigma \in P$ that maps under φ to a line segment. In that case we can write:

$$L_F(x_0, x_1) = \int_{t_0}^{t_1} F(\sigma')dt \leq \mathfrak{M} \int_{t_0}^{t_1} |\sigma'|dt = \mathfrak{M}\,|\varphi(x_1) - \varphi(x_0)|\ , \tag{3.5}$$

where $\sigma' = \frac{d\sigma}{dt}$ denotes the velocity field of a path σ.

To prove the left hand side of inequality (3.4) we first show that σ must be contained in P_x. The proof is by contradiction as follows.

Choose $r_0 < \frac{\mathfrak{m}}{\mathfrak{m}+3\mathfrak{M}} r$ and $\epsilon_0 = \mathfrak{M} r_0$ and $P_0 = \varphi^{-1}[B^n(r_0)] \subset P_x$. Let $\sigma : [t_0, t_1] \to \mathcal{U}$ be a piecewise C^∞ curve such that $\sigma(t_0) = x_0$ and $\sigma(t_1) = x_1$ for $x_0,\ x_1 \in P_0$. If $L_F(\sigma) \leq d_F(x_0, x_1) + \epsilon_0$ then the curve σ is certainly

contained in P_x, and since by equation (3.5) $d_F(x_0, x_1) \leq 2\mathfrak{M}r_0$ we have by assumption that $L_F(\sigma) \leq 3\mathfrak{M}r_0$. Now if we suppose that σ is not contained in P_x, and let $t_0 \leq t^* \leq t_1$ be the first instance where σ reaches the boundary ∂P_x, at the point $q \equiv \sigma(t^*)$, so that $|\varphi(q)| = r$, then:

$$L_F(\sigma) \geq L_F(\sigma_{[t_0,t^*]}) = \int_{t_0}^{t^*} F(\sigma')dt \geq \mathfrak{m}\int_{t_0}^{t^*} |\sigma'|dt \geq \mathfrak{m}|\varphi(q)-\varphi(x_0)| \geq \mathfrak{m}(r-r_0) \ . \tag{3.6}$$

But the length of this curve would in fact be longer then the maximum possible length of $3\mathfrak{M}r_0 < \mathfrak{m}(r - r_0)$, since by assumption we are assured that for $\mathfrak{m} > 0$ and $\mathfrak{M} > 1$ it is true that $r_0 < \frac{\mathfrak{m}}{\mathfrak{m}+3\mathfrak{M}}r$. Therefore σ must be contained in P_x.

The proof of the left hand side of inequality (3.3) follows by the same arguments as were used to prove (3.6)$\square$

Given the distortion lemma for the length of the path for any two points that belong to the compact set, we can easily derive a similar result that is valid for the shortest distances.

Corollary 30 *For a Finsler manifold $(\mathcal{U}, F)$, and any point $x \in \mathcal{U}$, let $\varphi : P_x \to \mathbb{R}^{4^n-1}$ be the local coordinate system diffeomorphism of a compact set P_x onto an open ball of $\mathbb{R}^{4^n-1}$, such that $\varphi(x) = 0$. Then, for a given $x_0,\ x_1 \in P_x$ and any Finsler metric $F(x, y)$ there exist a constant minimum $\mathfrak{m} > 0$ and a constant maximum $\mathfrak{M} > 1$ such that the following relation is true:*

$$\mathfrak{m}|\varphi(x_1) - \varphi(x_0)| \leq d_F(x_0, x_1) \leq \mathfrak{M}\,|\varphi(x_1) - \varphi(x_0)| \ . \tag{3.7}$$

Proof: We only need to verify the left hand side of inequality (3.3) is still true for minimal length curves. By definition of metric distance, for $0 \leq \epsilon \leq \epsilon_0$, two points $x_0, x_1 \in P_0$ can be joined by a piecewise C^∞ curve $\sigma : [t_0, t_1] \to \mathcal{U}$ with integral length:

$$L_F(\sigma) \leq d_F(x_0, x_1) + \epsilon \ . \tag{3.8}$$

By previous arguments, σ must lie in P_x, end by similar calculations we find that:

$$\mathfrak{m}|\varphi(x_1) - \varphi(x_0)| \leq L_F(\sigma) \leq d_F(x_0, x_1) + \epsilon \ , \tag{3.9}$$

Letting $\epsilon \to 0$ proves the desired result. $\square$

Lemma (29) and Corollary (30) allow us to bound the lengths on the manifold to the Euclidian lengths on the tangent space. For Euclidean coordinates

in our tangent space we will have the coefficients in the decomposition of the gate Hamiltonian matrix in terms of the generalized Pauli matrices, n times tensored two dimensional matrices from the set $\{I, X, Y, Z\}$.

For example, in the context of simulation, points x_0, $x_1 \in P_x$ and an open set P_x are chosen such that they correspond to the end points in the simulation by a single gate. Moreover, we can construct a *local coordinate system* on the Lie group $SU(2^n)$ which is a Lie algebra $\mathfrak{su}(2^n)$. For the *origin* $x_s \in \mathcal{U}$, define a pull back map $\varphi^{-1} : \mathbb{R}^{4^n-1} \to \mathcal{U}$, so that $x_{s+1} \equiv \exp^{-iy_{s+1}\cdot\sigma} x_s = \exp^{-i\varphi(x_{s+1})\cdot\sigma} x_s$, where σ denotes the coordinate basis, i.e. (4^n-1)-component vector whose entries are the generalized Pauli matrices.

For the particularly chosen metric, as in [4],

$$F_p(x_0, y) \equiv F_p(y) = \sqrt{\sum_{i=1}^{k} y_i^2 + p^2 \sum_{j\neq i} y_i^2},$$

where $k = \frac{9(n^2-n)}{2} + 3n$, which introduces a penalty p for the subset of Hamiltonian coordinates in the tangent space, so we have:

$$|y| \leq F_p(y) \leq p|y|.$$

Since this relation is true on any compact set, by Corollary (30) we have:

$$|\varphi(x_{s+1}) - \varphi(x_s)| \leq d_{F_p}(x_s, x_{s+1}) \leq p|\varphi(x_{s+1}) - \varphi(x_s)|\,. \tag{3.10}$$

It is important to note that in our analysis constants $\mathfrak{m}$ and $\mathfrak{M}$ do not depend on the compact set within which each gate is applied, and they are basically the property of the metric. This property might not be true in general for some other Finsler structures, but for our purposes here this assumption is very plausible.

3.4 Equivalence Result

For the sake of consistency and easier understanding, we follow the notation from [3] and denote with $m_{\mathcal{G}}$ the minimum number of gates, for a given set $\mathcal{G}$, needed to implement an arbitrary unitary $U \in \mathcal{U}$. Moreover, in this section we assume that the geodesic is simulated by sequential application of the gates from the set $\mathcal{G}$, and that by using a single gate from the set $\mathcal{G}$ we can simulate exactly any other point in the ball $B^+_{x_0}(\epsilon)$, i.e. which is the $\epsilon-$neighborhood around the initial condition at the point x_0. This is an unrealistic scenario,

since the set of gates would need to be infinite and non-local. Hence, we relax it in the next section. However, for the purpose of exact simulation of the geodesic it is an important tool. Clearly consideration of the set of gates defined in this way, as we shall see, is the best we can possibly hope for, and thus the bounds achieved by the set $\mathcal{G}$ are optimal, i.e. they determine the bounds achievable by any other set of gates that is less powerful.

For any gate used in a simulation we assign a gate index, so that eventually the index set is $s = \{0, 1, 2, ..., m_{\mathcal{G}} - 1\}$ for every gate in the simulation. Moreover, by $\sigma(t) : [0, m_{\mathcal{G}}] \to \mathcal{U}$ we denote a minimal geodesic between $\sigma(0) = I$, $\sigma(m_{\mathcal{G}}) = U$.

Note that, since $\mathcal{U}$ is a compact manifold with Finsler structure, all forward and backward Cauchy sequences with respect to d must converge on U. More precisely, compact Finsler spaces are automatically both forward complete and backward complete. This fact holds regardless of whether the Finsler structure is absolutely homogeneous or only positively homogeneous. Therefore, any two points on the manifold can be connected by a minimizing geodesic, as that property itself is a sufficient condition for the Hopf-Rinow theorem [6].

Theorem 31 *Let $d_F(I, U)$ denote a length of a geodesic between I and $U \in SU(2^n)$. For any simulation index set $s = \{0, 1, 2, ..., m_{\mathcal{G}} - 1\}$ let $P_{x_s} \in \mathcal{U}$ be an open set on the manifold that contains a segment of minimizing geodesic $\sigma_s(t) : [s, s + 1] \to \mathcal{U}$, that is simulated exactly by a single gate. Moreover, let P_{x_s} be mapped by φ diffeomorphically onto an open ball in $\mathbb{R}^{4^n-1}$, so that $\rho_s = |\varphi(x_{s+1}) - \varphi(x_s)|$ is the Euclidean length of the image of the geodesic segment σ_s. If we denote $\rho_{\sup} = \sup_s \rho_s$ and $\rho_{\inf} = \inf_s \rho_s$, then the following relation holds:*

$$\frac{d_F(I, U)}{\rho_{\sup}\mathfrak{M}} \leq m_{\mathcal{G}} \leq \frac{d_F(I, U)}{\rho_{\inf}\mathfrak{m}} \, . \tag{3.11}$$

Proof: For any segment gate index from set s, by the Corollary (30) we see that:

$$\mathfrak{m}\rho_s \leq d_F(x_s, x_{s+1}) \leq \mathfrak{M}\rho_s \, ,$$

Summing over all segments of minimizing geodesic $\sum_{s=0}^{m_g-1} d_F(x_s, x_{s+1}) = d_F(I, U)$, and taking into account that $\sum_{s=0}^{m_g-1} \beta_s \leq m_{\mathcal{G}}\beta$ and $m_{\mathcal{G}}\frac{\rho^2}{\beta} \leq \sum_{s=0}^{m_g-1} \frac{\rho_s^2}{\beta_s}$, it is easy to see that:

$$m_{\mathcal{G}}\mathfrak{m}\rho_{\inf} \leq d_F(I, U) = \sum_{s=0}^{m_g-1} d_F(x_s, x_{s+1}) \leq m_{\mathcal{G}}\mathfrak{M}\rho_{\sup} \, , \tag{3.12}$$

which gives the desired result by rearranging the variables.□

Note that the above theorem is derived in terms of bounds of the Euclidean distances in the tangent space. One may take a different path though, as for example Nielsen in [3], by deriving the result for the lower bound in terms of the lengths of the geodesic segments simulated by the single gate: $d_F(x_s, x_{s+1}) \leq \beta_{\sup}$. From the following theorem, one can reproduce the result derived by Nielsen as a special case when $\beta_{\sup} = 1$.

Theorem 32 *Let $d_F(I, U)$ denote a length of a geodesic between I and $U \in SU(2^n)$. For any simulation index set $s = \{0, 1, 2, ..., m_{\mathcal{G}} - 1\}$ let $P_{x_s} \in \mathcal{U}$ be an open set on the manifold that contains a segment of minimizing geodesic $\sigma_s(t) : [s,\, s+1] \rightarrow \mathcal{U}$, that is simulated exactly by a single gate. Moreover, let P_{x_s} be mapped by φ diffeomorphically onto an open ball in $\mathbb{R}^{4^n-1}$, so that $\rho_s = |\varphi(x_{s+1}) - \varphi(x_s)|$ is an image of the bounded length geodesic segment $d_F(x_s, x_{s+1}) \leq \beta_s$. If we denote $\beta_{\sup} = \sup_s \beta_s$ and $\beta_{\inf} = \inf_s \beta_s$, then the following relation holds:*

$$\frac{d_F(I,U)}{\beta_{\sup}} \leq m_{\mathcal{G}} \leq \frac{\mathfrak{M}}{\mathfrak{m}} \frac{d_F(I,U)}{\beta_{\inf}} \ . \tag{3.13}$$

Proof: Following along the lines of Theorem (31):

$$\mathfrak{m}\frac{\beta_s}{\mathfrak{M}} = \mathfrak{m}\rho_s \leq d_F(x_s, x_{s+1}) \leq \beta_s \equiv \mathfrak{M}\rho_s \,,$$

Summing over all segments of minimizing geodesic $\sum_{s=0}^{m_g-1} d_F(x_s, x_{s+1}) = d_F(I, U)$, and taking into account that $m_{\mathcal{G}}\beta_{\inf} \leq \sum_{s=0}^{m_g-1} \beta_s \leq m_{\mathcal{G}}\beta_{\sup}$:

$$\frac{\mathfrak{m}}{\mathfrak{M}} m_{\mathcal{G}} \beta_{\inf} \leq d_F(I,U) \leq m_{\mathcal{G}}\beta_{\sup} \ , \tag{3.14}$$

which gives the stated result. □

Equations (3.11) and (3.14) establish the tightest possible equivalence between the minimal number of gates in the circuit and geodesic length as a function of the simulation parameters. Again, the simulation parameters may be defined in terms of distances traversed with the single gate on the manifold or in terms of the Euclidean distances between the initial and final coefficients in the generalized Pauli expansion of the gate Hamiltonian. Even though the above results give no indication as to how to implement the simulation, they do provide us the best bounds we currently have and give us an estimate to the quality of the simulation provided that one knows the simulation parameters. However, the above results can be applied to the arbitrary paths on

the manifold including those that are generated adiabatically. In particular, it would be very interesting to compare the results for bounds of circuit size obtained by geometric techniques with the equivalence results obtained in [1].

3.5 Approximate simulation

In this section we reformulate the bounds for the standard circuit simulation procedure where the set of gates used consists solely of the single and two qubit gates, which are applied sequentially. Since the exact simulation of arbitrary unitary gate by single and two qubit gates demands an exponential number of gates, almost all unitaries simulated by the polynomial number of gates will be simulated approximately.

In particular, we consider two paths. Let the first be $d_{F_p}(I, \tilde{U})$, denoting the length of the geodesic simulated exactly with the set of gates from $\mathcal{G}$ with respect to the Finsler metric F_p , and let the second one $L_{F_p}(I, \tilde{U})$ be the minimum length path for the exact simulation of $\tilde{U}$ by the set of gates from $\mathcal{G}_2$. Here we denote by $\mathcal{G}_2$ the set of unitary gates whose time independent Hamiltonians have Hamming weight not greater than two.

Note that the length $L_{F_p}(I, \tilde{U})$ has nothing to do with $d_{F_p}(I, \tilde{U})$, as $L_{F_p}(I, \tilde{U})$ is completely determined by the simulation, and almost everywhere does not simulate the geodesic $d_{F_p}(I, \tilde{U})$.

Corollary 33 *Let $\tilde{U}$ be the approximation of the unitary operation U that is simulated by the one and two qubit gates. Then the lower bound on the minimum circuit size $\tilde{m}_{\mathcal{G}_2}$ is at most $O(\frac{n^4}{p} d^2_{F_p}(I, U) L_{F_p}(I, \tilde{U}))$, and the upper bound on $m_{\mathcal{G}_2}$ is at least $\Omega(n^4 d^3_{F_p}(I, U))$.*

Proof: The three step standard simulation of arbitrary $U = e^{-iH(t)t}$ is elaborated in detail by Nielsen, Dowling, Gu and Doherty in [4]. The procedure can be sketched as follows:

(1) the time variable Hamiltonian $H(t)$ is substituted by projected the Hamiltonian $H_P(t)$ that is formed by deleting all σ_i for $i > k$, i.e. all three- and more-body terms in the Pauli expansion of $H(t) = \sum_{i=1}^{k} y_i \sigma_i + \sum_{j \neq i} y_i \sigma_i$, where $k = \frac{9(n^2-n)}{2} + 3n$;

(2) the evolution due to $H_p(t)$ is broken up into many small intervals, each of length Δ, over which the time-dependent Hamiltonian $H_p(t)$ is accurately simulated by a constant mean Hamiltonian $\bar{H}_p^{\Delta}$;

(3) the mean Hamiltonian $\bar{H}_p^{\Delta}$ that has k terms in the Pauli expansion with coefficients $|y_i| \leq 1$ is simulated with a standard simulation technique [14] using one and two qubit gates.

The reader is encouraged to see [4] for full detail of the approximation result.

For the above procedure, since $SU(2^n)$ is compact and simply connected, there exists a path $L_{F_p}(I, \tilde{U})$ that is exactly synthesized with the gates in the simulation. By exactly simulated we mean that the end points of each gate in the simulation lie precisely on the path of length $L_{F_p}(I, \tilde{U})$. Clearly, the length of $L_{F_p}(I, \tilde{U}) \geq d_{F_p}(I, U)$.

Now we bound length of the path segments, $L_{F_p}(\tilde{x}_s, \tilde{x}_{s+1})$, for each of $\tilde{m}_{\mathcal{G}_2}$ gates in the simulation. Since there exists a compact set P_{x_s}, such that end points $\tilde{x}_s, \tilde{x}_{s+1} \in P_{x_s}$, that maps diffeomorphically to the local coordinate system, we can use Lemma (29) and its corollaries. Corroborating the arguments used to derive equation (3.10), over the compact set P_{x_s}, the Finsler structure, i.e. the Minkowski norm for the Pauli expansion of $H(t)$, is $F_p(x_s, y_s) = \sqrt{\sum_{i=1}^{k} y_i^2 + p^2 \sum_{j \neq i} y_i^2}$. Its minimum and maximum distortion over the compact set P_{x_s} are: $|y_s| \leq F_p(y_s) \leq p|y_s|$. Therefore, by the Lemma (29)

$|\varphi(\tilde{x}_{s+1}) - \varphi(\tilde{x}_s)| \leq L_{F_p}(\tilde{x}_s, \tilde{x}_{s+1}) \leq p|\varphi(\tilde{x}_{s+1}) - \varphi(\tilde{x}_s)|$.

The same is true for any other segment in the simulation, and hence:

$$m_{\mathcal{G}_2} \rho_{\text{inf}}^{\Delta} \leq L_{F_p}(I, \tilde{U}) = \sum_{s=0}^{m_g - 1} L_{F_p}(\tilde{x}_s, \tilde{x}_{s+1}) \leq m_{\mathcal{G}_2} p \rho_{\text{sup}}^{\Delta} \tag{3.15}$$

where $\rho_{\text{inf}}^{\Delta} = \inf_s |\varphi(\tilde{x}_{s+1}) - \varphi(\tilde{x}_s)|$, and $\rho_{\text{sup}}^{\Delta} = \sup_s |\varphi(\tilde{x}_{s+1}) - \varphi(\tilde{x}_s)|$. Note that we can always choose the s-th gate local coordinate system so that $\varphi(\tilde{x}_s) = 0$.

Finally, in the three-step simulation summarized above, gates at the third stage simulate the time invariant Hamiltonian $\bar{H}_p^{\Delta}$ for the segment Δ, with coordinates $|y_i| \leq 1$. More precisely, the s-th gate simulates the neighborhood around x_s: $x_{s+1} \equiv e^{-i\varphi(x_{s+1}) \cdot \sigma} x_s = e^{-iy_s \sigma_s \Delta^2} x_s$. Here $\sigma \in \mathcal{G}$ denotes stabilizer basis on n qubits, $\sigma_s \in \mathcal{G}_2$, and Δ^2 is the simulation time for every gate. If we choose $\Delta = \Theta((n^2 d_{F_p}(I, U))^{-1})$, as in [4], then for $|y_s| \leq 1$ we see that $\rho_s = |y_s \Delta^2| = \Theta((n^4 d_{F_p}^2(I, U))^{-1})$. Finally, using equation (3.15) we establish that the simulation with $\tilde{m}_{\mathcal{G}_2}$ gates has the upper bound

$$\Theta(n^4 d_{F_p}^2(I, U) L_{F_p}(I, \tilde{U})) \geq \Omega(n^4 d_{F_p}^3(I, U)). \tag{3.16}$$

By similar arguments, for the lower bound, we get

$$O(\frac{n^4}{p} d^2_{F_p}(I, U) L_{F_p}(I, \tilde{U})). \tag{3.17}$$

□

3.6 Conclusion

The Distortion Lemma and its corollary provide a general tool for relating distances on the manifold with distances on the tangent space. In this paper we have derived a generalized linear bounds for the exact simulation of any path on the manifold, in terms of the minimum circuit size and the simulation parameters.

The equivalence between the path on the manifold and circuit size still persists in the case of approximate simulation, provided that the simulation parameters have the appropriate scaling. However, one can not expect better than n^2 times improvement in the minimum circuit size upper bound over the result for standard circuit simulation derived by Nielsen, Dowling, Gu and Doherty [4].

Moreover, if one defines a metric on the manifold that penalizes the hard-to-simulate directions on the tangent space with high cost, that cost is, in effect, translated to the increased ratio between upper and lower bound in minimum circuit size.

Bibliography

[1] D. Aharonov, W. van Dam, J. Kempe, Z. Landau and Seth Lloyd, *Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation*, quant-ph/0405098;

[2] R. Oliveira, B. Terhal , *The complexity of quantum spin systems on a two-dimensional square lattice*, quant-ph/0504050;

[3] M. Nielsen, *A geometric approach to quantum circuit lower bounds*, quant-ph/0502070;

[4] M. Nielsen, M. Dowling, M. Gu and A. Doherty, *Quantum Computation as Geometry, Science Vol 311, Feb 2006;*

[5] C. Mochon, *Hamiltonian Oracles*, quant-ph/0602032

[6] D. Bao, S.-S. Chern and Z. Shen, *An Introduction to Riemann-Finsler Geometry*, Springer-Verlag 2000;

[7] J. Alvarez , C. Duran, *An Introduction to Finsler Geometry*, www.math.poly.edu/research/finsler;

[8] R. Montgomery, *A Tour of Subreimannian Geometries*, Their Geodesics and Applications, AMS 2000;

[9] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer-Verlag, 1989;

[10] H. Earp, J. Pachos, *A Constructive Algorithm for the Cartan Decomposition of* $SU(2^N)$*,* quant-ph/0505128;

[11] N. Khaneja, S. Glaser, *Cartan decomposition of* $SU(2^N)$*, constructive controllability of spin systems and universal quantum computing,* Chem. Physics 267, 11 (2001);

[12] V. Shende, S. Bullock, and I. Markov, *A Practical Top-down Approach to Quantum Circuit Synthesis*, quant-ph/0406176;

[13] F. Warner, *Foundations of Differentiable Manifolds and Lie Groups,* Springer-Verlag 1983;

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

Chapter 4

Complete Characterization of Mixing Time for the Continuous Quantum Walk on the Hypercube with Markovian Decoherence Model

The n-dimensional hypercube quantum random walk (QRW) is a particularly appealing example of a quantum walk because it has a natural implementation on a register on n qubits. However, any real implementation will encounter decoherence effects due to interactions with uncontrollable degrees of freedom. We present a complete characterization of the mixing properties of the hypercube QRW under a physically relevant Markovian decoherence model. In the local decoherence model considered the non-unitary dynamics are modeled as a sum of projections on individual qubits to an arbitrary direction on the Bloch sphere. We prove that there is always classical (asymptotic) mixing in this model and specify the conditions under which instantaneous mixing *always* exists. And we show that the latter mixing property, as well as the classical mixing time, depend heavily on the exact environmental interaction and its strength. Therefore, algorithmic applications of the QRW on the hypercube, if they intend to employ mixing properties, need to consider both the walk dynamics and the precise decoherence model.

4.1 Introduction, Previous Work, and Our Work

Quantum walks [1] play a prominent role in the design of quantum algorithms. Their distinction from classical random walks lies in their potentially faster mixing and hitting times. The underlying dynamics of quantum walks can be either continuous or discrete, and even though these two representations have some properties in common, analysis usually demands different techniques and tools. Despite their different origins, discrete and continuous time quantum walks can be precisely related to each another [41].

The theoretical properties of quantum walks on general graphs have been outlined in Ref. [10], and the remarkable result by Szegedy [9] explains a general framework for the proper quantization of any Markov chain algorithm. . The dynamics of quantum walks have been analyzed, for example, on the line [11], circle [12], hyperlattice [13] and hypercube [14]. Shenvi *et. al.* [7] proved that the discrete quantum walk can be used in a search algorithm and Ambainis used it for the best known algorithm for element distinctness [8]. The continuous-time quantum walk was originally proposed by Childs, Farhi and Guttman [4, 5] as an algorithmic primitive. Childs *et. al.* used a quantum walk to prove the best known results for the separation between quantum and classical query complexity [6], and a highly efficient algorithm for NAND formula evaluation [42].

Central to the algorithmic application of both classical and quantum random walks are their mixing characteristics. For a classical random walk that has a unique steady state, the mixing time characterizes the convergence of the walk to this steady state. In the quantum case, unitarity prevents the walk from reaching a steady state. This has led to alternative notions of mixing for quantum walks. One is to define an instantaneous mixing time, as the first instant the probability distribution of the walker's location on the graph is ϵ-close to the uniform distribution. Another sensible definition for the mixing time of a continuous quantum walk, although one that is depend on the initial state, is based on a limiting value of a time-averaged probability distribution [10].

Quantum systems are very susceptible to imperfections and interactions with their environment, both of which cause decoherence. Sufficient decoherence can remove any potential benefits from the quantum dynamics. Investigations to date have either used Markovian models for the environment, with the environment monitoring the walker position or state of the 'coin' driving the walk [16, 17, 18, 19, 20, 21, 20]; or imperfect evolution [22, 23, 24, 25, 26, 30],

such as broken edges.

While decoherence is nominally the nemesis of quantum information processing, it has been argued that decoherence can in fact be 'useful' in the context of quantum walks [16]. Decoherence can be used to force a quantum walk to mix to a uniform distribution, and in this paper we will illustrate this for the quantum walk on the hypercube. Similar results have been shown for quantum walks on the line [16] and N-cycle [28] using weak measurements of the walker's position.

In ref. [30], Marquezino *et al.* examined the discrete-time quantum walk on the hypercube, and derived the limiting time-averaged distribution in the coherent case (no decoherence). The mixing behavior, both to this distribution and the uniform distribution, were considered for a coherent walk as well as under the decohering effects of randomly breaking links in the network. In the decoherent case, the walk was shown to approach the uniform distribution. Interestingly there is an optimal decoherence rate which provides the fastest convergence. A similar effect was found for the N-cycle in [16]. Below we show that this can also be true for the continuous-time quantum walk on the hypercube, but not always.

Hitting times [1] and instantaneous mixing times for the continuous-time version of the hypercube quantum walk with decoherence were calculated recently by Alagic and Russell [21]. Analytical results were derived by exploiting the representation of the quantum walk on the hypercube as a set of non-interacting qubits; a simple example of how spin networks may be mapped to quantum walks [31]. The decoherence model was claimed to be the continuous-time analogue of weak position measurements in the discrete-time case, but was in fact an analogue of single-qubit, computational-basis measurements (for a discussion of this see [32]). We extend these results and provide a complete characterization of the mixing time in terms of the decoherence intensity and projection direction. In our case the projection direction of the decoherence operator can be in any arbitrary direction in contrast to the previous analysis which was restricted to decoherence in the computational basis $\{|0\rangle, |1\rangle\}$. Moreover, we show that randomizing the direction of decoherence leads to the depolarization channel that, regardless of the decoherence intensity, shows universal mixing behavior.

The paper is organized as follows: we first describe the quantum walk on the hypercube and its mapping to a register of qubits in section 4.2. Then in section 4.3 we introduce our decoherence model and explicitly show that it is

[1] The hitting time is defined as the first time a given vertex, or set of such, is reached.

the continuous-time analogue of the standard discrete-time projection model. The definitions of various mixing times are introduced formally in section 4.4. With our quantum walk and decoherence model the hypercube quantum walk is separable as a product over single qubits, and we need only consider single qubit dynamics, which we discuss in section 4.5. Analysis of simple channels, randomized projections and numerical results are presented as subsections of section 4.5.

4.2 Quantum Walk on the Hypercube

Continuous-time quantum walks [4] are defined over an undirected graph with $N \equiv 2^n$ nodes, each labelled by an integer $i \in [0, N-1]$. These walks can in general be described by the Hamiltonian

$$\begin{aligned} \hat{H}_s &= \sum_{[ij]} \Delta_{ij}(t) \left(\hat{c}_i^\dagger \hat{c}_j + \hat{c}_i \hat{c}_j^\dagger \right) + \sum_j \epsilon_j(t) \hat{c}_j^\dagger \hat{c}_j, \\ &\equiv \sum_{[ij]} \Delta_{ij}(t) \left(|i\rangle\langle j| + |j\rangle\langle i| \right) + \sum_j \epsilon_j(t) |j\rangle\langle j|, \end{aligned} \tag{4.1}$$

where each node i corresponds to the quantum state, $|i\rangle = \hat{c}_i^\dagger |0\rangle$, and $[ij]$ denotes connected nodes i and j. The state $|i\rangle$ thus corresponds to a 'particle' located at node i. The first term in (4.1) is a 'hopping' term with amplitude $\Delta_{ij}(t)$ between nodes i and j; the second describes 'on-site' node energies $\epsilon_j(t)$. Both these terms can depend on time. One can simplify (4.1) by dropping all the on-site energies, and by making all the internode hopping matrix elements the same, i.e., $\Delta_{ij} \to \Delta, \forall \{i, j\}$.

The structure of a hypercube is particularly appealing; it is an $n-$regular graph that is the underlying model for many computational problems. The nodes of this graph can be represented as basis vectors $|v\rangle \in \{|1\rangle, |0\rangle\}^{\otimes n}$ in $\mathbb{C}^{2^n}$ dimensional Hilbert space – the same Hilbert space describing n qubits. Each node is labelled by a binary string representing a multi-qubit state, i.e. $|\vec{z}\rangle \equiv |z_1 z_2 \dots z_n\rangle = |z_1\rangle \otimes |z_2\rangle \otimes \dots \otimes |z_n\rangle$, where the $z_i' s$ are 0 or 1; pairs of nodes with a Hamming distance of 1 (the number of bits that must be flipped to obtain one from the other) are connected to give the hypercube. In this way n qubits describe a quantum walk over a N-dimensional hypercube, which takes place in information space. This is described by the simple qubit Hamiltonian

$$H = \Delta \sum_{i=1}^{n} \hat{\sigma}_x^i, \tag{4.2}$$

representing a set of non-interacting qubits, each evolving under $\hat{\sigma}_x$.

In this case the unitary dynamics is trivially solvable; for the walker initialized at the $\vec{z} = \vec{0}$ corner, the probability of being at some site $\vec{z}$ is

$$P_{\vec{z}}(t) \equiv \langle \vec{z}|\varrho(t)|\vec{z}\rangle = \cos^{2n_0}(\Delta t)\sin^{2n_1}(\Delta t), \tag{4.3}$$

where n_0 is the number of 0's, and n_1 the number of 1's appearing in $\vec{z}$, and $\varrho(t) = e^{-iHt}|\vec{z}=0\rangle\langle\vec{z}=0|e^{iHt}$ is the density matrix of the qubit register at time t.

This mapping to a qubit model is not only useful conceptually, but it is highly suggestive of a potential physical implementation of the hypercube quantum walk. If the system is not closed and it is exposed to measurement or an environment that it interacts with, the quantum dynamics is in general far more complicated and we will examine this now.

4.3 Decoherence Model

Previous studies of decoherence in quantum walks have mainly focused upon discrete-time quantum walks where decoherence has been modelled as a sequence of weak measurements on the walker [16, 15, 17, 18]. When the result of the measurement is ignored (i.e. lost to the environment), the nonunitary process is described by:

$$\varrho_{n+1} = (1-p)\hat{U}\varrho_n\hat{U}^\dagger + p\sum_\alpha M_\alpha(\hat{U}\varrho_n\hat{U}^\dagger)M_\alpha^\dagger, \tag{4.4}$$

where the measurement is given by the POVM $\{M_\alpha\}$, such that $\sum_\alpha M_\alpha = 1$ and $M_\alpha \geq 0$, occurring with probability p at each time step; $\hat{U}$ describes the unitary evolution of the quantum walk. This model is equivalent to a memoryless environment, unperturbed by the system. We will consider the continuous-time analogue of this process, which we derive below.

Claim: The discrete-time weak measurement model of the system dynamics,

$$\varrho_{t+\tau} = (1-\gamma\tau)U_\tau\varrho_t U_\tau^\dagger + \gamma\tau\sum_\alpha M_\alpha[U_\tau\varrho_t U_\tau^\dagger]M_\alpha^\dagger\ , \tag{4.5}$$

is equivalent, in the limit $\tau \to 0$ to the master equation

$$\dot{\varrho}(t) = -i[H,\varrho(t)] + \gamma\sum_\alpha \mathcal{D}[M_\alpha]\varrho(t)\ , \tag{4.6}$$

if the measurement rate γ is such that for a time-step of duration τ, $p = \gamma\tau$, unitary evolution $\hat{U}_\tau = \exp(-iH\tau)$. Here the superoperator $\mathcal{D}[X]\varrho \equiv X\varrho(t)X^\dagger - \frac{1}{2}\left(X^\dagger X\varrho(t) + \varrho(t)X^\dagger X\right)$ for any operator X.

Proof: For small τ we expand the exponential to first-order such that Eq. (4.5) becomes

$$\begin{aligned}
\varrho_{t+\tau} &= (1-\gamma\tau)(\mathbf{1}-i\tau H)\varrho_t(\mathbf{1}+i\tau H) && (4.7)\\
&\quad +\gamma\tau\sum_\alpha M_\alpha(\mathbf{1}-i\tau H)\varrho_t(\mathbf{1}+i\tau H)M_\alpha^\dagger && (4.8)\\
&= \varrho_t + i\tau\varrho_t H - i\tau H\varrho_t - \gamma\tau\varrho_t + \gamma\tau\sum_\alpha M_\alpha\varrho_t M_\alpha^\dagger && (4.9)\\
&= \varrho_t + i\tau\varrho_t H - i\tau H\varrho_t && (4.10)\\
&\quad +\gamma\tau\sum_\alpha\left(M_\alpha\varrho_t M_\alpha^\dagger - \frac{1}{2}M_\alpha^\dagger M_\alpha\varrho_t - \frac{1}{2}\varrho_t M_\alpha^\dagger M_\alpha\right). && (4.11)
\end{aligned}$$

Dividing by τ we obtain,

$$\frac{\varrho_{t+\tau}-\varrho_t}{\tau} = -i[H,\varrho_t] + \gamma\sum_\alpha\left(M_\alpha\varrho_t M_\alpha^\dagger - \frac{1}{2}[M_\alpha^\dagger M_\alpha\varrho_t - \varrho_t M_\alpha^\dagger M_\alpha]\right), \quad (4.12)$$

then taking the limit $\tau \to 0$,

$$\begin{aligned}
\dot{\varrho}(t) &= -i[H,\varrho(t)] && (4.13)\\
&\quad +\gamma\sum_\alpha\left(M_\alpha\varrho(t)M_\alpha^\dagger - \frac{1}{2}[M_\alpha^\dagger M_\alpha\varrho(t) - \varrho(t)M_\alpha^\dagger M_\alpha]\right) && (4.14)\\
&= -i[H,\varrho(t)] + \gamma\sum_\alpha\mathcal{D}[M_\alpha]\varrho(t)\ , && (4.15)
\end{aligned}$$

as claimed.■

The interaction between the system and the environment can be represented in various ways, through the choice of the set of $\{M_\alpha\}$. In the quantum walk literature, the standard choice is the walker location, i.e. $M_\alpha = |i\rangle\langle i| = \mathbb{P}_i$, the projectors onto the graph node states. In their analysis of the quantum walk on the hypercube, this is what [21] claim to consider. However, when the quantum walk on the hypercube is implemented via a set of qubits, as we describe here, position measurements corresponds to a computational basis measurement of the state of every qubit simultaneously. This implies a physically unrealistic, multi-qubit measurement/interaction with the environment.

If the quantum walk were to be implemented using a qubit register, a more physically realistic decoherence process is described by single-qubit projective

measurements. The obvious choice, as an analogue to the location measurements, are measurements that are projections onto single qubit computational basis states; this is the choice considered in [21]. We can generalize this to single-qubit projective measurements onto arbitrary antipodal points on the Bloch sphere. We express this as:

$$\begin{aligned}
\mathbb{P}_0(\overrightarrow{\mathbf{r}}) &= \frac{\mathbb{I} + \overrightarrow{\mathbf{r}} \cdot \vec{\boldsymbol{\sigma}}}{2} \\
&\equiv \frac{\mathbb{I} + (\sin\theta\cos\varphi)\sigma_x + (\sin\theta\sin\varphi)\sigma_y + (\cos\theta)\sigma_z}{2} \in \mathbb{C}^{2\times 2} \quad (4.16) \\
\mathbb{P}_1(\overrightarrow{\mathbf{r}}) &= \frac{\mathbb{I} - \overrightarrow{\mathbf{r}} \cdot \vec{\boldsymbol{\sigma}}}{2} \quad (4.17) \\
&\equiv \frac{\mathbb{I} - (\sin\theta\cos\varphi)\sigma_x - (\sin\theta\sin\varphi)\sigma_y - (\cos\theta)\sigma_z}{2} \in \mathbb{C}^{2\times 2} \quad (4.18)
\end{aligned}$$

where $\mathbb{I}$ is the two-dimensional identity matrix,

$$\overrightarrow{\mathbf{r}} = (r_x, r_y, r_z) = (\sin\theta\cos\varphi, \sin\theta\sin\varphi, \cos\theta),$$

for $\theta \in [0, \pi]$, $\varphi \in [0, 2\pi]$, defines what we refer to as the "decoherence axis" or measurement projection direction. By $\vec{\boldsymbol{\sigma}} = (\sigma_x, \sigma_y, \sigma_z)$ we denote the three dimensional matrix vector composed of the three nontrivial Pauli matrices. This decoherence model is a generalization of what has been referred to as the *subspace projection* decoherence model [3].

The continuous quantum walk described by the master equation (4.6), with Hamiltonian (4.2) and

$$M_\alpha^k = \mathbb{I} \otimes \ldots \otimes \mathbb{P}_\alpha(\overrightarrow{\mathbf{r}}) \otimes \ldots \otimes \mathbb{I}, \quad (4.19)$$

where $\alpha = \{0, 1\}$ and the projector is on the k^{th} qubit. Note that $\sum_\alpha M_\alpha^k = \mathbb{I}^{\otimes n}$. The qubit register evolution equation can then be written as:

$$\dot{\varrho}(t) = \sum_{k=1}^{n} -i\Delta\left[\sigma_x^k, \varrho(t)\right] + \gamma \sum_{k=1}^{n}\sum_{\alpha=0}^{1} \mathcal{D}\left[M_\alpha^k\right]\varrho(t). \quad (4.20)$$

This can alternatively be written as:

$$\dot{\varrho}(t) = \sum_{k=1}^{n} -i\Delta\left[\sigma_x^k, \varrho(t)\right] + \frac{\gamma}{2}\sum_{k=1}^{n} \mathcal{D}\left[\overrightarrow{\mathbf{r}} \cdot \vec{\boldsymbol{\sigma}}_k\right]\varrho(t), \quad (4.21)$$

where the sum is over qubits. None of the qubits are interacting and therefore this master equation has a separability property that allows us to treat the dynamics as n single qubit density matrices undergoing the evolution:

$$\dot{\rho_k}(t) = -i\Delta\left[\sigma_x, \rho_k(t)\right] + \frac{\gamma}{2}\mathcal{D}\left[\overrightarrow{\mathbf{r}} \cdot \vec{\boldsymbol{\sigma}}\right]\rho_k(t), \quad (4.22)$$

and $\varrho = \bigotimes_{k=1}^{n} \rho_k$. See the addendum for an explicit derivation of this. This property allows one to analyze the full system dynamics by looking at single qubit dynamics, since the dynamics of the system is just the direct sum of the dynamics of individual, non-interacting qubits. Furthermore, the evolution equation for ρ_k is the same for all k, and so we will drop the subscript when referring to single qubit dynamics.

We will investigate how the changes in the single qubit dynamics affect properties of the quantum walk on the hypercube. After formally defining the mixing time of a quantum walk in the next section, we provide a complete characterization of the mixing time in terms of the quantum channel in Eq. (4.22), i.e. dependence on the physical rates and the direction of the decoherence axis, $\overrightarrow{\mathbf{r}}$ [2].

4.4 Mixing Time

To identify the physical quantities of interest, we return to a principal motivation for considering random walks (quantum or classical). In computer science, the most efficient solution to many problems is given by a probabilistic algorithm, where the correct answer is attained with high-probability if the space of solutions is sampled with a well-chosen sampling distribution. Generating the correct sampling distribution is often a matter of mapping the uniform distribution into the desired one, and therefore generating a truly uniform distribution is an important problem.

There are several different definitions in the literature that have been used as a measure of mixing time. For completeness will briefly list them all here. For our purposes the notions of instantaneous mixing and classical mixing will be the most important ones.

Instantaneous mixing is defined as the first time instant at which the probability distribution of the walker's position is sufficiently close to uniform distribution:

$$M_{inst,\varepsilon} = \min\{t \mid ||P(x,t) - P_u||_{tv} < \varepsilon\} \tag{4.23}$$

where $P(x,t)$ is the probability of obtaining element $x \in \mathcal{X}$ ($\mathcal{X}$ is the space of events we are sampling from, which in the case of random walks is the space of the walker location parameter) at time t, and P_u is the uniform distribution over $\mathcal{X}$. $||\cdot||_{tv}$ is the *total variation* distance over probability distributions (we

[2]Note that varying $\overrightarrow{\mathbf{r}}$ is equivalent to changing the basis in which node states are encoded while the decoherence axis is kept fixed.

will restrict our attention to finite sample spaces). This definition is mostly used in idealized continuous quantum random walks where no decoherence effects are present. Although formally present by this definition, mixing in continuous quantum random walks without decoherence is only an instantaneous phenomenon. The ability to harness this instantaneous mixing is still questionable.

Average mixing is based on the time-averaged probability distribution, that even for unitary quantum walks is shown to converge [10]. In the continuous-time case the time-averaged probability distribution for the state x is defined as:

$$\bar{P}(x,\tau) = \frac{1}{\tau}\int_0^\tau P(x,t)dt\,, \tag{4.24}$$

We can define the corresponding time-averaged mixing as:

$$M_{avg,\varepsilon} = \min\{T \mid \forall\tau > T : ||\bar{P}(x,\tau) - P_u||_{tv} < \varepsilon\}. \tag{4.25}$$

This time-averaged distribution can be sampled from by selecting t uniformly in $[0,T]$, running the quantum walk for time t, and measuring the walker position.

Classical (asymptotic) mixing is the quantity originally used in classical random walks and it defines mixing as the time after which the register's distribution is desirably close to uniform:

$$M_{class,\varepsilon} = \min\{T \mid \forall t > T : ||P(x,t) - P_u||_{tv} < \varepsilon\} \tag{4.26}$$

This definition characterizes the time it takes for the probability of finding the walker at a particular location to be distributed uniformly across the entire sample space, $\mathcal{X}$.

The mixing time is a well defined quantity for classical random walks because there exists a stationary distribution for classical random walks over any connected, non-bipartite graph [37], however for continuous quantum walks this is not necessarily the case; unitary dynamics means the probability distribution over the graph nodes oscillates, and therefore never converges to the uniform distribution. We shall return to this issue below. But first, let us examine the total variational distance in the context of a quantum walk on a hypercube.

For a hypercube quantum walk implemented using qubits, the walker location is encoded into the value of the qubit register in the computational basis. Therefore, the sample space in this case is the space of binary strings of length

n, and the probability of measuring any register value (walker "location"), $\vec{z}$ is:

$$P(\underline{x}, t) = (1 - p_0(t))^k p_0(t)^{n-k} \tag{4.27}$$

where k is the Hamming weight of the binary string $\vec{z}$ (i.e. number of ones in $\vec{z}$), and $p_0(t)$ is the probability of a qubit value being 0 at time t. Note that:

$$p_0(t) = \frac{1 + \langle \sigma_z(t) \rangle}{2}$$

where $\langle A\,(t) \rangle \equiv \mathrm{tr}(A\rho(t))$ for any operator A. The total variational distance in this case is:

$$\begin{aligned} ||P(\vec{z}, t) - P_u||_{tv} &= \sum_{\vec{z}} |P(\vec{z}, t) - \frac{1}{2^n}| \\ &= \sum_k \binom{n}{k} |(1 - p_0(t))^k p_0(t)^{n-k} - \frac{1}{2^n}| \end{aligned} \tag{4.28}$$

We can bound this variational distance using the *Hellinger distance* on distributions [21]. The Hellinger distance, for two distributions $\mathcal{P}(x)$ and $\mathcal{Q}(x)$ (both defined over the same sample space $\mathcal{X}$), is defined as:

$$H(\mathcal{P}, \mathcal{Q})^2 = \sum_x [\sqrt{\mathcal{P}(x)} - \sqrt{\mathcal{Q}(x)}]^2 = 1 - \sum_x \sqrt{\mathcal{P}(x)\mathcal{Q}(x)} \tag{4.29}$$

Its usefulness for us comes from its relation to the total variational distance:

$$||\mathcal{P} - \mathcal{Q}||_{tv} \leq 2H(\mathcal{P}, \mathcal{Q}) \leq 2||\mathcal{P} - \mathcal{Q}||_{tv}^{1/2} \tag{4.30}$$

Therefore in our case,

$$\begin{aligned} ||P(\vec{z}, t) - P_u||_{tv}^2 &\leq 4H(P(\vec{z}, t), P_u)^2 \\ &= 4 - 4 \sum_{\vec{z}} \sqrt{P(\vec{z}, t) \frac{1}{2^n}} \\ &= 4 - 4 \sum_k \binom{n}{k} \sqrt{\frac{(1 - p_0(t))^k p_0(t)^{n-k}}{2^n}} \qquad (4.31) \\ &= 4 - 4 \left(\sqrt{\frac{1 - p_0(t)}{2}} + \sqrt{\frac{p_0(t)}{2}} \right)^n \\ &= 4 \left[1 - \frac{1}{2^n} \left(\sqrt{1 - \langle \sigma_z(t) \rangle} + \sqrt{1 + \langle \sigma_z(t) \rangle} \right)^n \right] \qquad (4.32) \end{aligned}$$

Note that this is a positive quantity because we only take the positive branch of the square roots, and it attains its minimum value of zero when $\langle \sigma_z(t) \rangle = 0$.

Furthermore, from this bound we see that the variational distance for the distribution of the register of n qubits (from the uniform distribution) is small exactly when the variational distance for the distribution of a single qubit is small. In fact, when $\langle\sigma_z\rangle$ is small, we can expand the square roots to second order in this quantity to get:

$$||P(\vec{z},t) - P_u||_{tv}^2 \leq 4 - 4\left(1 - \frac{\langle\sigma_z(t)\rangle^2}{8}\right)^n \tag{4.33}$$

Given this fact, we will concentrate on the distribution for a single qubit in the following, and appeal to the fact that the variational distance for the entire register from the uniform distribution is small precisely when $\langle\sigma_z(t)\rangle^2$ is small for a single qubit.

4.5 Single-qubit dynamics

In order to solve for the single qubit dynamics we use the following parametrization of the single qubit density operator,

$$\rho(t) = \frac{1}{2}\begin{pmatrix} 1+\langle\sigma_z(t)\rangle & \langle\sigma_x(t)\rangle - i\langle\sigma_y(t)\rangle \\ \langle\sigma_x(t)\rangle + i\langle\sigma_y(t)\rangle & 1-\langle\sigma_z(t)\rangle \end{pmatrix}, \tag{4.34}$$

This operator is completely described by the Bloch vector

$$\langle\vec{\sigma}\rangle = (\langle\sigma_x\rangle, \langle\sigma_y\rangle, \langle\sigma_z\rangle) \equiv (x, y, z).$$

We can then derive the matrix equation for the Bloch vector of a single qubit when the evolution is given by Eq. (4.22), repeated here for clarity:

$$\dot{\rho}(t) = -i\Delta\left[\sigma_x, \rho(t)\right] - \frac{\gamma}{2}\rho(t) + \frac{\gamma}{2}\left(\mathbf{r}\cdot\boldsymbol{\sigma}\right)\rho(t)\left(\mathbf{r}\cdot\boldsymbol{\sigma}\right)^\dagger. \tag{4.35}$$

Multiplying this equation by each of the Pauli matrices and taking the trace we obtain the 3×3 parametrized linear system equation:

$$\langle\dot{\overrightarrow{\boldsymbol{\sigma}}}(t)\rangle \equiv \frac{d}{dt}\mathrm{tr}(\overrightarrow{\boldsymbol{\sigma}}\rho(t)) = \mathrm{tr}(\overrightarrow{\boldsymbol{\sigma}}\dot{\rho}(t)) = \mathbf{A}\langle\overrightarrow{\boldsymbol{\sigma}}(t)\rangle, \tag{4.36}$$

where,

$$\mathbf{A} = \begin{pmatrix} \gamma(r_x^2-1) & \gamma r_x r_y & \gamma r_x r_z \\ \gamma r_x r_y & \gamma(r_y^2-1) & \gamma r_y r_z - 2\Delta \\ \gamma r_x r_z & \gamma r_y r_z + 2\Delta & \gamma(r_z^2-1) \end{pmatrix}. \tag{4.37}$$

The solution of this system is $\langle \overrightarrow{\boldsymbol{\sigma}}(t) \rangle = \exp(\mathbf{A}t)\langle \overrightarrow{\boldsymbol{\sigma}}(0) \rangle$, where $\langle \overrightarrow{\boldsymbol{\sigma}}(0) \rangle$ is the Bloch vector of the initial state $|0\rangle$. The dynamics of the qubit, and in turn the entire register implementing the quantum walk, is completely determined by the properties the matrix $\mathbf{A}$. We now examine the key properties of this matrix.

Firstly, by the Routh-Hurwitz criterion [27], the matrix $\mathbf{A}$ has eigenvalues that lie in the left half of the complex plane for positive γ, Δ and all decoherence axes *except* for when $r_x = 1$. This singular case represents a channel where the Hamiltonian dynamics and the decoherence dynamics commute. This case is easy to solve for explicitly (we present the solution in 4.5.1) and the dynamics for it are fairly uninteresting. For all other parameter regimes, the Routh-Hurwitz criterion tells us that Eq. (4.36) is a strictly stable system.

The eigenvalues of A can be determined from its characteristic equation:

$$\lambda^3 + 2\gamma\lambda^2 + (\gamma^2 + 4\Delta^2)\lambda + 4\gamma\Delta^2(1 - r_x^2) = 0 \tag{4.38}$$

Interestingly the eigenvalues depend only on r_x, γ, Δ. The most convenient form for the solutions to this cubic equation can be found by mapping the equation into a third order Chebyshev polynomial and using the Chebyshev cube root. Then the solution can be written in closed form as:

$$\lambda_1 = 2\sqrt{\frac{\gamma^2 - 12\Delta^2}{9}}\cos(\frac{1}{3}\arccos(m)) - \frac{2\gamma}{3} \tag{4.39}$$

$$\lambda_2 = -2\sqrt{\frac{\gamma^2 - 12\Delta^2}{9}}\cos(\frac{1}{3}\arccos(-m)) - \frac{2\gamma}{3}$$

$$\lambda_3 = -\lambda_1 - \lambda_2 - 2\gamma, \tag{4.40}$$

where

$$m = \frac{\gamma}{(\gamma^2 - 12\Delta^2)^{3/2}}(\gamma^2 + 18\Delta^2(3r_x^2 - 1))). \tag{4.41}$$

These eigenvalues fall into one of two classes, depending on the values of γ, Δ,and $\overrightarrow{\mathbf{r}}$: one real and two imaginary eigenvalues, or three (with possible repetitions) real eigenvalues.

$\mathbf{A}$ is not a symmetric matrix and is therefore not generally diagonalizable. $\mathbf{A}$ could be not diagonalizable if it has repeated eigenvalues (i.e. a degenerate eigenspace). We will see below when we perform a more detailed analysis of the eigenvalues of $\mathbf{A}$ that this only occurs in a vanishingly small parameter range which will not be of interest to us. Therefore we will effectively treat $\mathbf{A}$ as diagonalizable.

Given these properties of the matrix $\mathbf{A}$, let us return to the mixing behavior of a single qubit under the dynamics given by Eq. (4.36). Assuming that the

initial state of the whole register of the system is in the state $|0\rangle^{\otimes n}$ at $t = 0$ the $\langle\sigma_z(t)\rangle$ component of each individual qubit is:

$$\langle\sigma_z(t)\rangle = (0\ 0\ 1)e^{\mathbf{A}t}\langle\overrightarrow{\boldsymbol{\sigma}}(0)\rangle = (0\ 0\ 1)e^{\mathbf{A}t}(0\ 0\ 1)^T = \left(e^{\mathbf{A}t}\right)_{33} \tag{4.42}$$

For diagonalizable $\mathbf{A}$, with eigenvalues λ_j, $e^{\mathbf{A}t}$ can be written as:

$$e^{\mathbf{A}t} = \sum_{j=1}^{3} e^{\lambda_j t} \prod_{k\neq j} \frac{\mathbf{A} - \lambda_k \mathbb{I}}{\lambda_j - \lambda_k}, \tag{4.43}$$

Using this expansion,

$$\langle\sigma_z(t)\rangle = \sum_{\pi(\lambda_1,\lambda_2,\lambda_3)} \frac{e^{\lambda_{\pi_1} t}}{(\lambda_{\pi_1} - \lambda_{\pi_2})(\lambda_{\pi_1} - \lambda_{\pi_3})} \begin{bmatrix} -4\Delta^2 + \gamma^2(1 - r_z^2) \\ +(\lambda_{\pi_2} + \lambda_{\pi_3})\gamma(1 - r_z^2) + \lambda_{\pi_2}\lambda_{\pi_3} \end{bmatrix} \tag{4.44}$$

where $\pi(\lambda_1,\lambda_2,\lambda_3)$ denotes the three cyclic permutations of $(\lambda_1,\lambda_2,\lambda_3)$. From this expression, we can see that $\langle\sigma_z(t)\rangle$ depends on all the *free* parameters in the system: r_x, r_z, γ, Δ (even though the eigenvalues only depend on r_x, γ and Δ).

This expression for $\langle\sigma_z\rangle(t)$ tells us something crucial about the QRW. The exponential envelopes $e^{\lambda_{\pi_1} t}$, and the fact that $\mathbf{A}$ is a strictly stable matrix (has eigenvalues in the left half of the complex plane), imply that $\langle\sigma_z\rangle \xrightarrow{t\to\infty} 0$. Hence, by Eq. (4.33), the limiting distribution for the quantum walk for all $\overrightarrow{\mathbf{r}}$ and $\gamma > 0$ is the uniform distribution. Thus, the decoherence ensures that the random walk mixes to uniform given sufficient time. This is the reason the notion of average mixing time is not interesting or necessary in the presence of decoherence.

Finally, we note that although the above convergence argument was for the particular initial state, $|0\rangle^{\otimes n}$, the strict stability of the dynamical matrix $\mathbf{A}$ (for all parameters except $r_x = 1$) allows us to state more generally that there is an initial state independent steady state $\langle\overrightarrow{\boldsymbol{\sigma}}\rangle \xrightarrow{t\to\infty} \mathbf{0}$, which of course implies $\langle\sigma_z\rangle \xrightarrow{t\to\infty} 0$.

4.5.1 Special Cases: Simple Channels

In this section, to illustrate the utility of our approach we exactly solve the dynamics for several simple single-qubit channels.

For $r_x = 0$, the dynamics is described by the matrix

$$\mathbf{A} = \begin{pmatrix} -\gamma & 0 & 0 \\ 0 & \gamma(r_y^2 - 1) & \gamma r_y r_z - 2\Delta \\ 0 & \gamma r_y r_z + 2\Delta & \gamma(r_z^2 - 1) \end{pmatrix}. \tag{4.45}$$

So $\langle\sigma_x(t)\rangle = e^{-\gamma t}\langle\sigma_x(0)\rangle$, and we have a pair of coupled linear differential equations describing the motion in the y-z plane. The motion in this plane is analogous to a damped simple harmonic oscillator with natural frequency 2Δ and damping rate γ; for $\gamma < 4\Delta$ the system is underdamped, resulting in decaying oscillations around the origin, while in the overdamped regime, $\gamma > 4\Delta$, we see exponential decay. In analyzing the mixing properties of the hypercube quantum walk, we need only consider the behavior of the z-component of the Bloch vector. When the initial state is $\langle\overrightarrow{\boldsymbol{\sigma}}(0)\rangle = (0,\ 0,\ 1)^T$, in the underdamped case, $\gamma < 4\Delta$, we have,

$$\langle\sigma_z(t)\rangle = e^{-\gamma t/2}\left(\cos\omega t + \frac{\gamma(2r_z^2-1)}{2\omega}\sin\omega t\right). \tag{4.46}$$

where $\omega \equiv \sqrt{|\gamma^2 - 16\Delta^2|}/2$. In the overdamped regime, $\gamma > 4\Delta$,

$$\langle\sigma_z(t)\rangle = \frac{1}{\sqrt{\gamma^2-16\Delta^2}}\left((\lambda_+ + \gamma r_z^2)e^{\lambda_+ t} - (\lambda_- + \gamma r_z^2)e^{\lambda_- t}\right), \tag{4.47}$$

where $\lambda_\pm = (-\gamma \pm \sqrt{\gamma^2 - 16\Delta^2})/2$. Finally, in the critically damped case $\gamma = 4\Delta$, $\langle\sigma_z(t)\rangle = (2\Delta(2r_z^2-1)t + 1)\,e^{-2\Delta t}$.

This result applies to two standard, single-qubit decoherence channels [33]; the phase-flip, or dephasing, channel where $\overrightarrow{\mathbf{r}} = (0,0,1)$ and the bit-phase-flip channel $\overrightarrow{\mathbf{r}} = (0,1,0)$.

In the opposite case, where $\overrightarrow{\mathbf{r}} = (1,0,0)$ (the bit-flip channel),

$$\mathbf{A} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\gamma & -2\Delta \\ 0 & 2\Delta & -\gamma \end{pmatrix}, \tag{4.48}$$

and the solution for $z(t)$,with initial condition $\langle\overrightarrow{\boldsymbol{\sigma}}(0)\rangle = (0,\ 0,\ 1)^T$, is simply

$$\langle\sigma_z(t)\rangle = e^{-\gamma t}\cos 2\Delta t. \tag{4.49}$$

Finally we examine the depolarizing channel [33], which although is not an instance of the class of channels described by (4.22), can be obtained by a randomization of such channels (see the Addendum for the detailed derivation of this). The depolarization channel corresponds to a randomization of the decoherence axis, $\overrightarrow{\mathbf{r}}$, and is described by a dynamical matrix:

$$\mathbf{A} = \begin{pmatrix} -\frac{2\gamma}{3} & 0 & 0 \\ 0 & -\frac{2\gamma}{3} & -2\Delta \\ 0 & 2\Delta & -\frac{2\gamma}{3} \end{pmatrix}. \tag{4.50}$$

The solution for the z-component is:

$$\langle\sigma_z(t)\rangle = e^{-\frac{2\gamma}{3}t}\cos 2\Delta t. \tag{4.51}$$

4.5.2 The Complete Classification of Mixing Behavior

For dynamics under a general single qubit channel, the convergence of the $\langle \sigma_z(t) \rangle$ to the limiting value 0 can be essentially of two different types, depending on the eigenvalues of $\mathbf{A}$: exponential decay or dampened oscillator decay. We now characterize the type of decay in terms of the physical parameters of the qubit model, and then use the decay types to make conclusions about the mixing properties of the quantum walk.

As we have already seen the characteristic equation for the matrix $\mathbf{A}$ is:

$$\lambda^3 + 2\gamma\lambda^2 + (\gamma^2 + 4\Delta^2)\lambda + 4\gamma\Delta^2\eta = 0, \tag{4.52}$$

denoting $\eta \equiv 1 - r_x^2$. For a cubic equation with real coefficients the solutions are either: (1) three real, possibly repeated, roots, or (2) one real root and two complex conjugate roots. According to Eq. (4.44) these two classes of roots give rise to two fundamentally different types of convergence of $\langle \sigma_z \rangle(t)$ to zero: exponential decay or dampened oscillator decay. To determine the parameter dependence of these two types of convergence behavior, we can use the discriminant of (4.52):

$$\Lambda(\eta) = 432\gamma^2\Delta^4\eta^2 - (16\gamma^4\Delta^2 + 576\gamma^2\Delta^4)\eta + (4(\gamma^2 + 4\Delta^2)^3 - 4\gamma^2(\gamma^2 + 4\Delta^2)^2). \tag{4.53}$$

There are two fundamentally distinct parameter regions:

(i) (Zeno region) $\Lambda(\eta) \leq 0 \Rightarrow 3$ real roots, that are distinct unless, $\Lambda(\eta) = 0$, in which case either two or all three are repeated,

(ii) (no-Zeno region) $\Lambda(\eta) > 0 \Rightarrow 1$ real root and 2 complex conjugate roots .

The reason for the names for the two regions, *Zeno* and *no-Zeno*, will become clear when we examine the mixing behavior of random walks with dynamics prescribed by a dynamical matrix $\mathbf{A}$ that lies in one of the above regions. First, let us define the border between the Zeno and no-Zeno regions in terms of the parameters r_x, γ and Δ. This border is defined by the values where the discriminant equals zero:

$$\Lambda(\eta) = 0 \Leftrightarrow \eta = \frac{2}{3} + \frac{\gamma^2}{54\Delta^2} \pm \frac{\sqrt{(\gamma^2 - 12\Delta^2)^3}}{54\gamma\Delta^2}; \tag{4.54}$$

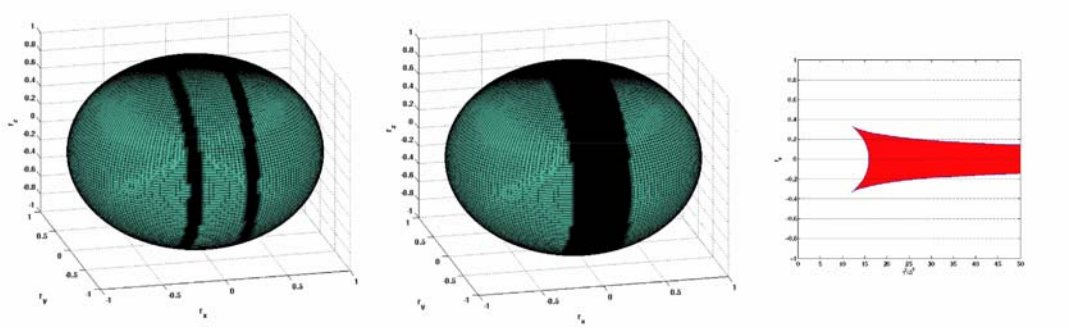


Figure 4.1: Eigenvalue regions of the matrix $\mathbf{A}$ in parameter space. *(Right Fig.)* Phase diagram for eigenvalues of the dynamical matrix $\mathbf{A}$. The red (shaded) region indicates where eigenvalues are purely real, or what is referred to as the Zeno-region in the main text. *(Left and Center Fig.)* $\mathbf{r}$-space diagram of eigenvalue types for matrix $\mathbf{A}$ when $\frac{\gamma}{\Delta} = \sqrt{15}$ and $\frac{\gamma}{\Delta} = \sqrt{20}$ respectively.

When $\gamma < \sqrt{12}\Delta$ the equation $\Lambda(\eta) = 0$ has no real solutions, and therefore we cannot have repeated roots in this parameter range. On the other hand, when $\gamma \geq \sqrt{12}\Delta$ we have two real values for η that define the upper and lower boundary of the no-Zeno parameter region. Since $\eta \equiv 1 - r_x^2$ we get the following expressions:

$$- \text{ for } \sqrt{12}\Delta \leq \gamma \leq 4\Delta, \text{ repeated eigenvalues when} \tag{4.55}$$

$$r_x = \pm\sqrt{\frac{1}{3} - \frac{\gamma^2}{54\Delta^2} \pm \frac{\sqrt{\left(\gamma^2 - 12\Delta^2\right)^3}}{54\gamma\Delta^2}}; \tag{4.56}$$

$$- \text{ for } \gamma > 4\Delta, \text{ repeated eigenvalues when} \tag{4.57}$$

$$r_x = \pm\sqrt{\frac{1}{3} - \frac{\gamma^2}{54\Delta^2} + \frac{\sqrt{\left(\gamma^2 - 12\Delta^2\right)^3}}{54\gamma\Delta^2}}. \tag{4.58}$$

Mathematically, the change in behavior at 4Δ exists because the minus branch of the expression under the square root in equation (4.55) becomes imaginary. The phase diagram in Fig. 4.1 shows the eigenvalue regimes as a function of the parameter ratio γ/Δ. Also, in this figure we explicitly show

the regions in $\overrightarrow{\mathbf{r}}$-space corresponding to the two eigenvalue regimes for two specific values of $\gamma/\Delta : \sqrt{15}$ and $\sqrt{20}$.

From this information about the eigenvalue phase diagram, we can make the following conclusions about the mixing behavior for various parameter regimes:

(i) for $\gamma < \sqrt{12}\Delta$ we have weak decoherence, so that for all decoherence axes (all $\overrightarrow{\mathbf{r}}$) we have a pair of complex conjugate eigenvalues and therefore both instantaneous and classical mixing.

(ii) for $\gamma \geq \sqrt{12}\Delta$ and the decoherence projection direction r_x is such that $|r_x| \leq \sqrt{\frac{1}{3} - \frac{\gamma^2}{54\Delta^2} + \frac{\sqrt{(\gamma^2-12\Delta^2)^3}}{54\gamma\Delta^2}}$ we have two subcases:

(a) if $\sqrt{12}\Delta \leq \gamma \leq 4\Delta$ and decoherence projection direction r_x also such that $|r_x| \geq \sqrt{\frac{1}{3} - \frac{\gamma^2}{54\Delta^2} - \frac{\sqrt{(\gamma^2-12\Delta^2)^3}}{54\gamma\Delta^2}}$ then all eigenvalues are real, possibly repeated. There is no oscillatory behavior in $\langle\sigma_z\rangle(t)$, and therefore no instantaneous mixing time exists. However, the walk of course has a classical mixing time as $\langle\sigma_z\rangle \xrightarrow{t\to\infty} 0$.

(b) if $\gamma > 4\Delta$ all eigenvalues are real, possibly repeated. There is no oscillatory behavior in $\langle\sigma_z\rangle(t)$, and therefore no instantaneous mixing time exists. However, the walk of course has a classical mixing time as $\langle\sigma_z\rangle \xrightarrow{t\to\infty} 0$.

(iii) for $\gamma \geq \sqrt{12}\Delta$ and decoherence projection direction r_x is such that it does not satisfy the conditions from (ii) and (ii-a) then we have a pair of complex conjugate eigenvalues and therefore both instantaneous and classical mixing.

The fact that no finite instantaneous mixing time exists when all eigenvalues are real (for their restricted decoherence model) was interpreted by Alagic and Russell [21] as an analogue of the Zeno effect where the quantum evolution of a system is hindered by its strong interaction with an environment [38]. Following this, we refer to the region where no instantaneous mixing time exists – i.e. regime *(ii)* above where all eigenvalues of $\mathbf{A}$ are real – as the Zeno-region, and the remainder of $(\gamma, \Delta, \overrightarrow{\mathbf{r}})$ parameter space as the no-Zeno region.

4.5.3 Numerical simulations

In this section we numerically evaluate $\langle\sigma_z\rangle(t)$ and calculate the classical mixing time for several parameter values and $\varepsilon = 0.001$. These simulations are summarized in Fig. 4.2 which show mixing time for all values of $\overrightarrow{\mathbf{r}}$ and three values of the physical parameter ratio γ/Δ. We make several observations from these plots:

- The mixing time can vary considerably as $\overrightarrow{\mathbf{r}}$ is varied. Although the change in mixing time is generally smooth with changes in $\overrightarrow{\mathbf{r}}$, there are regions where the change is abrupt (e.g. around $r_x = \pm 1$ when $\gamma/\Delta = 1$). This implies that the mixing time can potentially change drastically with the exact value of $\overrightarrow{\mathbf{r}}$. Hence it is very important to characterize the decoherence process accurately for determining mixing properties. A similar conclusion was arrived at by Strauch in Ref. [3] where he demonstrates that mixing behavior differs greatly depending on the decoherence model chosen.

- The range of mixing times on the $\overrightarrow{\mathbf{r}}$-sphere is smallest when $\gamma/\Delta \approx 1$. The range of mixing time diverges when this parameter ratio is very large or very small. This suggests an optimal γ/Δ parameter ratio where the interplay between Hamiltonian dynamics and decoherence is such that decoherence in any direction yields small mixing times. We will investigate this more thoroughly in the next section.

- In the Hamiltonian dominated regime, where $\gamma/\Delta < 1$, we have fast mixing near the r_x axis but long mixing times in the $r_y - r_z$ plane.

- In the decoherence dominated regime, where $\gamma/\Delta > 1$, we have fairly similar mixing times across nearly all values of $\overrightarrow{\mathbf{r}}$ except for a small region around $\overrightarrow{\mathbf{r}} = (\pm 1, 0, 0)$. The size of this region shrinks as γ/Δ increases, but the value of the mixing time in this region grows with the same parameter. However, note that when the decoherence is *exactly* along the r_x axis we have very short mixing times as evident from the exact solution given by Eq. (refeq::Xbit-dep) for this case.

4.5.4 Optimal decoherence rate

The numerical simulations presented in the last section suggested that the smallest mixing time is achieved for a non-zero value of γ/Δ. The simulations

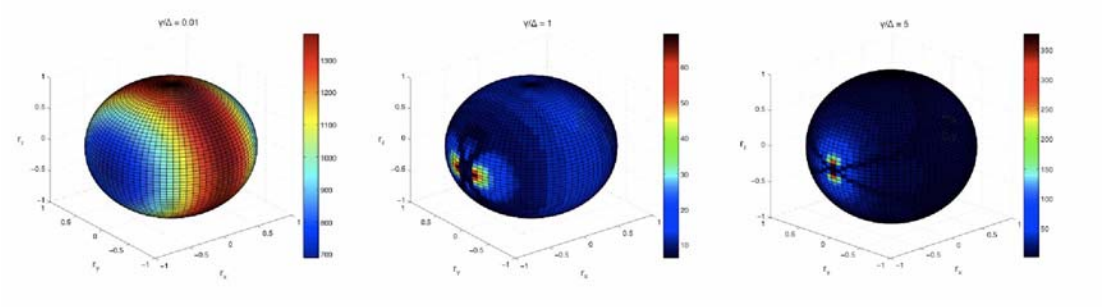


Figure 4.2: Variation of single qubit mixing times with decoherence axis for several values of $\frac{\gamma}{\Delta}$. The color at each point on the Bloch sphere corresponds to the mixing time for single qubit dynamics when the decoherence vector is (r_x, r_y, r_z). The mixing times are in units of Δ and are calculated for $\epsilon = 0.001$. Note that $\langle\sigma_z\rangle(t)$ is invariant under negation of any coordinate of the decoherence axis $(r_x \to -r_x, r_y \to -r_y, r_z \to -r_z)$, and so the portions of the sphere that cannot be seen can be inferred. Figures are for $\frac{\gamma}{\Delta} = \{0.01, 1, 5\}$ respectively from left to right.

also suggested that the optimal γ/Δ is around the critical ratio $\gamma/\Delta = 1$. These observations match with Kendon and Tregenna's conclusions in Ref. [16] where they showed that some amount of decoherence can lead to faster mixing of quantum walks on a line and cycle, and faster hitting times of the quantum walk on a hypercube. Here we fully characterize the scaling of mixing time with decoherence rate (for the hypercube quantum walk) by numerically evaluating the mixing time for several fixed decoherence axes. Figure 4.3 shows how the mixing time varies with γ/Δ for several choices of $\overrightarrow{\mathbf{r}}$. In general, the curves are similar for any latitude in $\overrightarrow{\mathbf{r}}$-space, that is, for any fixed θ. For a given θ the mixing time versus γ/Δ curves (for various ϕ) show maximum variation when $\theta = \pi/2$ (i.e. when $\overrightarrow{\mathbf{r}}$ is in the $x-y$ plane). Therefore we have shown these curves on a separate plot in Fig. 4.3.

The notion of an optimal ratio γ/Δ is accurate for nearly all decoherence axes. And for nearly all decoherence axes, this optimal value is in the range $1 < \gamma/\Delta < 5$. However, when $\overrightarrow{\mathbf{r}}$ is in the $x-y$ plane, there is no finite optimal value for γ/Δ; the mixing time decreases continuously as γ/Δ is increased for $\overrightarrow{\mathbf{r}}$ in the $x-y$ plane. We can gain intuition about this result by viewing the decoherence as a localizing phenomenon – it tends to localize the qubit state along the axis (on its Bloch sphere) defined by $\overrightarrow{\mathbf{r}}$ vector. And the larger γ/Δ is, the faster this localization happens. For a qubit localized in the

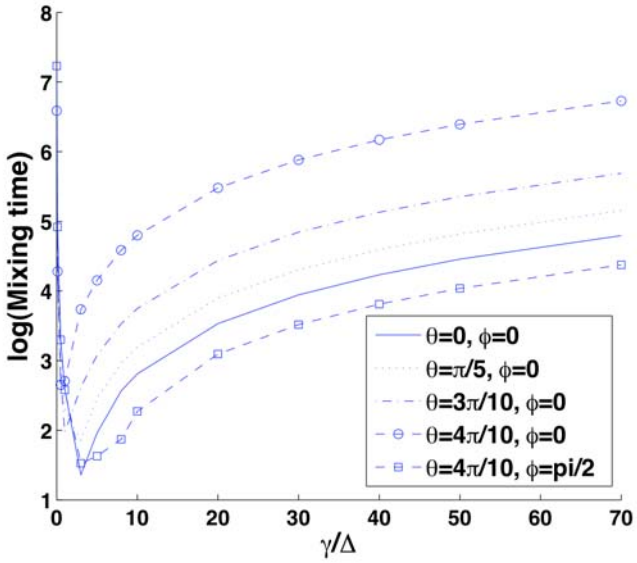


Figure 4.3: Log scaling of single qubit mixing time with the physical parameter ratio $\frac{\gamma}{\Delta}$. The mixing time curves are shown for several values of θ and φ, the two angular parameters of the decoherence axis. This figure shows mixing time curves vs. $\frac{\gamma}{\Delta}$ for $0 \leq \theta < \pi/2$. We primarily only show curves for $\varphi = 0$ in this parameter range for θ, because the behavior of mixing time as $\frac{\gamma}{\Delta} \to 0$ and $\frac{\gamma}{\Delta} \to \infty$. As $\theta \to \pi/2$ these curves show more variation with φ, but they maintain this general shape.

$x - y$ plane, $\langle \sigma_z \rangle = 0$ and hence a fast localization to this plane yields fast mixing. The variation of the mixing time when $\overrightarrow{\mathbf{r}}$ is *within* the $x - y$ plane is an interesting feature of Fig. 4.4. As $\overrightarrow{\mathbf{r}}$ approaches the x-axis ($\varphi = 0$) the mixing time evolution becomes closer and closer to exponential decay (with γ/Δ). However, away from the x-axis, the curves still show a local minimum around $\gamma/\Delta \approx 1$, but the global minima are still for $\gamma/\Delta \to \infty$.

4.6 Conclusion

We complete the picture that exists in the literature for quantum random walks on the hypercube with decoherence under the subspace projection decoherence model. This model of decoherence is the most physically realistic form when the quantum random walk is implemented using a register of qubits.

The following are the important points of our work:

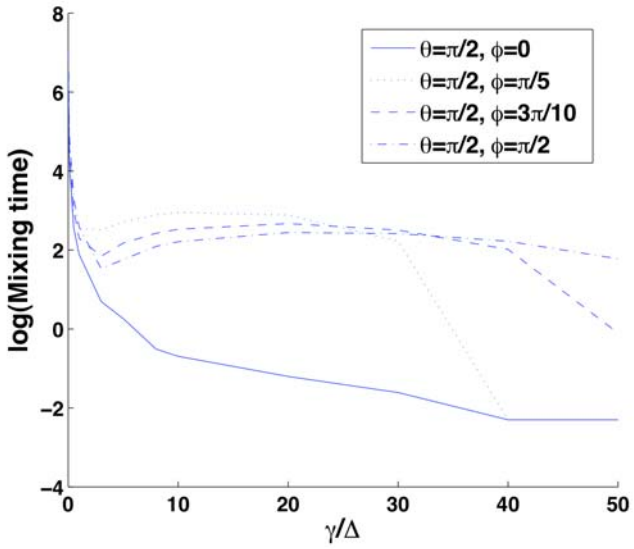


Figure 4.4: Characteristic behavior of mixing time versus $\frac{\gamma}{\Delta}$ for $\theta = \pi/2$.

- The instantaneous mixing time property of the quantum random walk does not necessarily disappear as the decoherence strength is increased. More precisely, Zeno dynamics prevails only in the precisely specified regions of the $\overrightarrow{\mathbf{r}}, \gamma, \Delta$ parameter space defined in 4.5.2. Consequently, the hypercube quantum walk with decoherence does not always limit to its classical version as the decoherence is increased. Depending on the direction of the decoherence vector it is possible to have oscillatory dynamics and instantaneous mixing persist as decoherence is increased.

- We showed numerically that, for almost all decoherence directions, a finite decoherence rate exists for optimally fast mixing. This optimal rate is approximately the same as the rate of Hamiltonian evolution. However, we also showed that for certain decoherence directions ($\overrightarrow{\mathbf{r}}$ in the $x - y$ plane) no finite optimal decoherence rate exists and the mixing rate increases without bound as the decoherence rate is increased. This result is particularly relevant given recent results on quantum walk based modeling of excitation transport in biomolecules [39, 40]. In these works it is argued that decoherence can lead to faster hitting times and walker diffusion, and this is explicitly confirmed for a simplified model of dephasing of the quantum walk. Our results suggest that the exact model of the decoherence matters greatly, and therefore, that an accu-

rate model of the environmental interactions is essential to assess the merits or demerits of decoherence to excitation transport.

- Furthermore, we showed in section 4.5.1 that randomizing the decoherence axis yields the depolarizing channel which exhibits both instantaneous and classical mixing regardless of the relative decoherence rate $\frac{\gamma}{\Delta}$. Yet another example of a quantum walk that does not decohere to a classical walk as the decoherence rate is increased. Therefore, introducing randomized decoherence may be an avenue for controlling the mixing behavior of the hypercube random walk.

4.7 Addendum: Single Qubit Master Equation

The master equation, (4.21), is hard to solve in general but in our case the system Hamiltonian and decoherence operators are a sum of the tensor products that have a special structure. Each summand is a tensor product of elements only one of which is not the identity. We now show formally that this allows one to consider a combination of single qubit evolution equations.

Vectorization is a technique that transforms any $n \times n$ matrix into a n^2 dimensional vector by stacking the transposed rows of the matrix on the top of each other. We will denote a vectorized matrix X as X^v. A useful identity we will utilize involves the vectorization of the matrix product AXB: $(AXB)^v = (B^T \otimes A)X^v$. The action of unitary evolution on a density operator is $U_t \rho U_t^\dagger$, and that is vectorized as $(U_t \rho U_t^\dagger)^v = (U_t^* \otimes U_t)\rho^v = \mathcal{S}_t \rho^v$, where $\mathcal{S}_t$ is the matrix form of the unitary evolution superoperator. Using this formalism it is straight forward to derive the vectorized picture of the master equation. Consider the discretized evolution given by Eq. (4.5) with $\tau \to dt$, and the qubit projection POVMs given by Eq. (4.19):

$$\varrho_{t+dt} = (1 - \gamma dt) U_{dt} \varrho_t U_{dt}^\dagger + \gamma dt \sum_k \sum_\alpha M_\alpha^k [U_{dt} \varrho_t U_{dt}^\dagger] M_\alpha^k \tag{4.59}$$

This can vectorized as:

$$\begin{aligned} \varrho_{t+dt}^v &= (1 - \gamma dt)\{U_{dt} \varrho_t U_{dt}^\dagger\}^v + \gamma dt \mathcal{P}^v (U_{dt} \varrho_t U_{dt}^\dagger)^v && (4.60) \\ &= (1 - \gamma dt)(U_{dt}^* \otimes U_{dt}) \varrho_t^v + \gamma dt \mathcal{P}^v (U_{dt}^* \otimes U_{dt}) \varrho_t^v && (4.61) \\ &= [(1 - \gamma dt) + \gamma dt \mathcal{P}^v][U_{dt}^* \otimes U_{dt}] \varrho_t^v && (4.62) \end{aligned}$$

where $\mathcal{P}^v$ is the operator:

$$\begin{aligned}\mathcal{P}^v &= \sum_{k=1}^{n}\sum_{\alpha=0}^{1}[M_\alpha^{k*}\otimes M_\alpha^k] \\ &= \sum_{k=1}^{n}\mathbb{I}^{\otimes 2(k-1)}\otimes[\mathbb{P}_0^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_0(\overrightarrow{\mathbf{r}})+\mathbb{P}_1^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_1(\overrightarrow{\mathbf{r}})]\otimes\mathbb{I}^{\otimes 2(n-1-k)}\end{aligned}$$

This transition equation for ϱ defines the system dynamics at all times. Now, let $\varrho_t^v=\mathcal{S}_t\varrho_0^v$ where $\mathcal{S}_t$ is the propagator matrix for the dynamics. From Eq. (4.62) we know that $\varrho_{t+dt}^v=[(1-\gamma dt)+\gamma dt\mathcal{P}^v][U_{dt}^*\otimes U_{dt}]\mathcal{S}_t\varrho_0^v$, and since this is true for any initial state ϱ_0^v, we get:

$$\begin{aligned}\mathcal{S}_{t+dt} &= [(1-\gamma dt)+\gamma dt\mathcal{P}^v][U_{dt}^*\otimes U_{dt}]\mathcal{S}_t \\ &= [(1-\gamma dt)\mathbf{1}+\gamma dt\mathcal{P}^v][(\mathbf{1}+iHdt)\otimes(\mathbf{1}-iHdt)]\mathcal{S}_t \qquad (4.63)\\ &= [\mathbf{1}\otimes\mathbf{1}+idt(H\otimes\mathbf{1}-\mathbf{1}\otimes H)-\gamma dt\mathbf{1}\otimes\mathbf{1}+\gamma dt\mathcal{P}^v]\,\mathcal{S}_t, \qquad (4.64)\end{aligned}$$

where we have expanded $U_{dt}=e^{-iHdt}$ to first order, and $\mathbf{1}=\mathbb{I}^{\otimes n}$. Taking into account that $\dot{\mathcal{S}}_t=\frac{d\mathcal{S}_t}{dt}=\lim_{dt\to 0}\frac{S_{t+dt}-S_t}{dt}$ we get the differential form:

$$\dot{\mathcal{S}}_t=[i[H\otimes\mathbf{1}-\mathbf{1}\otimes H]-\gamma\mathbf{1}\otimes\mathbf{1}+\gamma\mathcal{P}^v]\;\mathcal{S}_t \qquad (4.65)$$

Using $H=\Delta\sum_{k=1}^{n}\mathbb{I}^{\otimes(k-1)}\otimes\sigma_x\otimes\mathbb{I}^{\otimes(n-1-k)}$ we expand this as:

$$\begin{aligned}\dot{\mathcal{S}}_t &= \left[\begin{array}{c}\sum_{k=1}^{n}\mathbb{I}^{\otimes 2(k-1)}\\ \otimes\left\{\begin{array}{c} i\Delta[\sigma_x\otimes\mathbb{I}-\mathbb{I}\otimes\sigma_x]-\gamma\mathbb{I}\otimes\mathbb{I}+\\ \gamma[\mathbb{P}_0^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_0(\overrightarrow{\mathbf{r}})+\mathbb{P}_1^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_1(\overrightarrow{\mathbf{r}})]\end{array}\right\}\otimes\mathbb{I}^{\otimes 2(n-1-k)}\end{array}\right] \qquad (4.66)\\ &\equiv \mathcal{A}\mathcal{S}_t. \qquad (4.67)\end{aligned}$$

Since the initial condition is $\mathcal{S}_0=I^{\otimes n}$., the solution to this differential equation is $\mathcal{S}_t=e^{\mathcal{A}t}$:

$$\begin{aligned}\mathcal{S}_t &= \exp\left[\begin{array}{c}\sum_{k=1}^{n}\mathbb{I}^{\otimes 2(k-1)}\\ \otimes t\left\{\begin{array}{c} i\Delta[\sigma_x\otimes\mathbb{I}-\mathbb{I}\otimes\sigma_x]-\gamma\mathbb{I}\otimes\mathbb{I}+\\ \gamma[\mathbb{P}_0^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_0(\overrightarrow{\mathbf{r}})+\mathbb{P}_1^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_1(\overrightarrow{\mathbf{r}})]\end{array}\right\}\otimes\mathbb{I}^{\otimes 2(n-1-k)}\end{array}\right] \qquad (4.68)\\ &= \sum_{k=1}^{n}\mathbb{I}^{\otimes 2(k-1)}\otimes\exp t\left\{\begin{array}{c} i\Delta[\sigma_x\otimes\mathbb{I}-\mathbb{I}\otimes\sigma_x]-\gamma\mathbb{I}\otimes\mathbb{I}+\\ \gamma[\mathbb{P}_0^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_0(\overrightarrow{\mathbf{r}})+\mathbb{P}_1^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_1(\overrightarrow{\mathbf{r}})]\end{array}\right\}\\ &\quad\otimes\mathbb{I}^{\otimes 2(n-1-k)} \qquad (4.69)\\ &= [\exp t\left\{\begin{array}{c} i\Delta[\sigma_x\otimes\mathbb{I}-\mathbb{I}\otimes\sigma_x]-\gamma\mathbb{I}\otimes\mathbb{I}\\ +\gamma[\mathbb{P}_0^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_0(\overrightarrow{\mathbf{r}})+\mathbb{P}_1^*(\overrightarrow{\mathbf{r}})\otimes\mathbb{P}_1(\overrightarrow{\mathbf{r}})]\end{array}\right\}]^{\otimes n}\\ &\equiv [\bar{\mathcal{S}}_t]^{\otimes n}\end{aligned}$$

Therefore dynamics of the system is tensor product of individual qubit dynamics $\bar{\mathcal{S}}_t = e^{\bar{\mathcal{A}}t}$. The single qubit generator can be simplified as follows:

$$\begin{aligned}
\bar{\mathcal{A}} &= i\Delta[\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x] - \gamma \mathbb{I} \otimes \mathbb{I} && (4.70)\\
&\quad +\gamma[\mathbb{P}_0^*(\overrightarrow{\mathbf{r}}) \otimes \mathbb{P}_0(\overrightarrow{\mathbf{r}}) + \mathbb{P}_1^*(\overrightarrow{\mathbf{r}}) \otimes \mathbb{P}_1(\overrightarrow{\mathbf{r}})] && (4.71)\\
&= i\Delta[\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x] - \gamma \mathbb{I} \otimes \mathbb{I} && (4.72)\\
&\quad +\frac{\gamma}{4}[\left(\mathbb{I} + \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}\right)^* \otimes \left(\mathbb{I} + \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}\right) && (4.73)\\
&\quad + \left(\mathbb{I} - \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}\right)^* \otimes \left(\mathbb{I} - \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}\right)] && (4.74)\\
&= i\Delta[\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x] - \frac{\gamma}{2}\mathbb{I} \otimes \mathbb{I} + \frac{\gamma}{2}\overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}^* \otimes \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}. && (4.75)
\end{aligned}$$

It can be easily confirmed that this is the generator for the single qubit dynamics described by Eq. (4.22) once it has been vectorized.

4.7.1 Depolarizing Channel through the Randomized Decoherence Axis

The generator for the single qubit dinamics when the decoherence axis is randomized is:

$$\begin{aligned}
\bar{\mathcal{A}}_d &= i\Delta(\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x) - \gamma \mathbb{I} \otimes \mathbb{I} && (4.76)\\
&\quad +\frac{\gamma}{4\pi} \oint_{S^2} [\mathbb{P}_0^*(\overrightarrow{\mathbf{r}}) \otimes \mathbb{P}_0(\overrightarrow{\mathbf{r}}) + \mathbb{P}_1^*(\overrightarrow{\mathbf{r}}) \otimes \mathbb{P}_1(\overrightarrow{\mathbf{r}})] ds \ . && (4.77)
\end{aligned}$$

We can carry out this integral to get the following:

$$\begin{aligned}
\bar{\mathcal{A}}_d &= \begin{array}{l} i\Delta(\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x) - \gamma \mathbb{I} \otimes \mathbb{I} \\ +\frac{\gamma}{4\pi} \int_{\theta,\varphi} \left[\left(\frac{\mathbb{I} + \overrightarrow{\mathbf{r}} \cdot \boldsymbol{\sigma}}{2} \right)^* \otimes \left(\frac{\mathbb{I} + \overrightarrow{\mathbf{r}} \cdot \boldsymbol{\sigma}}{2} \right) + \left(\frac{\mathbb{I} - \overrightarrow{\mathbf{r}} \cdot \boldsymbol{\sigma}}{2} \right)^* \otimes \left(\frac{\mathbb{I} - \overrightarrow{\mathbf{r}} \cdot \boldsymbol{\sigma}}{2} \right) \right] d\overrightarrow{\mathbf{r}} \end{array} \\
&= \begin{array}{l} i\Delta(\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x) - \gamma \mathbb{I} \otimes \mathbb{I} \\ +\frac{\gamma}{8\pi} \int_{\theta,\varphi} (\mathbb{I} \otimes \mathbb{I} + \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}^* \otimes \overrightarrow{\mathbf{r}} \cdot \overrightarrow{\boldsymbol{\sigma}}) d\overrightarrow{\mathbf{r}} \end{array} \\
&= i\Delta(\sigma_x \otimes \mathbb{I} - \mathbb{I} \otimes \sigma_x) - \frac{\gamma}{2}\mathbb{I} \otimes \mathbb{I} + \frac{\gamma}{6}(\sigma_x \otimes \sigma_x + \sigma_y^* \otimes \sigma_y + \sigma_z \otimes \sigma_z).
\end{aligned}$$

where we have used $r_x = \sin\theta\cos\varphi$, $r_y = \sin\theta\sin\varphi$, $r_z = \cos\theta$. This leads to the solution for the single qubit dynamics:

$$\rho^v(t) = e^{\bar{\mathcal{A}}_d t}\rho^v(0) \ . \qquad (4.78)$$

Changing the basis for $\bar{\mathcal{A}}_d$ to the eigenbasis basis, exponentiating, and returning to the original basis we get:

$$\rho\left(t\right) = \frac{1}{2}\begin{pmatrix} 1 + e^{-\frac{2\gamma t}{3}}\cos(2\Delta t) & -ie^{-\frac{2\gamma t}{3}}\sin(2\Delta t) \\ ie^{-\frac{2\gamma t}{3}}\sin(2\Delta t) & 1 - e^{-\frac{2\gamma t}{3}}\cos(2\Delta t) \end{pmatrix} \quad . \tag{4.79}$$

with $\rho(0) = |0\rangle\langle 0|$. The value of $\rho\left(t\right)_{00}$ and $\rho\left(t\right)_{11}$ determines the probability of measurement in basis $\{|0\rangle, |1\rangle\}$. The eigenvalues of our operator $\sigma(\bar{\mathcal{A}}_d) = \{0, -\frac{2\Delta\gamma t}{3}, -\frac{2\Delta t}{3}(\gamma - 3i), \frac{2\Delta t}{3}(\gamma + 3i)\}$ determine the probability distribution. The expressions for $\rho\left(t\right)_{00}$ and $\rho\left(t\right)_{11}$ show that *regardless* of the rate of decoherence instantaneous mixing exists.

Bibliography

[1] J. Kempe, *Quantum random walks - an introductory overview*, Contemporary Physics, **44** (4), 307 (2003).

[2] W. Strauch, *Connecting the discrete and continuous time quantum walks*, Phys. Rev. A **73** 054302 (2006).

[3] F. Strauch, *Decoherence in quantum walks on the hypercube revisited*, quant-ph/0808.3403v1

[4] E. Farhi and S. Gutmann. Quantum computation and decision trees. *Phys. Rev. A* 58:915-928, 1998.

[5] A. Childs, E. Farhi and S. Gutmann, *An Example of the difference between quantum and classical random walks,* Quantum Information Processing **1** 35 (2002), quant-ph/0103020;

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

[7] N. Shenvi, J. Kempe and K.B. Whaley *, A Quantum Random Walk Search Algorithm,* Phys. Rev. A **67**, 052307 (2003)

[8] A. Ambainis, *Quantum walk algorithm for element distinctness,* SIAM Journal on Computing, 37(1):210-239 (2007); quant-ph/0311001.

[9] M. Szegedy, *Quantum speed-up of Markov Chain based algorithms*, Proceedings of the 45th IEEE Symposium on Foundations of Computer Science, pp. 32-41, 2004; quant-ph/0401053.

[10] D. Aharonov, A. Ambainis, J. Kempe and U. Vazirani *Quantum Walks on Graphs,* Proceedings of ACM Symposium on Theory of Computation (STOC'01), July 2001, p. 50-59; quant-ph/0012090v2.

[11] A. Romanelli, A.C. Sicardi-Schifino, R. Siri, G. Abal, A. Auyuanet and R. Donangelo *Quantum Random Walks on the line as a Markovian Process*, Physica A **338**, 395 (2004).

[12] D. Solenov, L. Fredichkin, *Continuous-time quantum walks on a cycle graph*, Phys. Rev. A. **73**, 012313 (2006).

[13] N.V. Prokof'ev and P.C.E. Stamp, *Decoherence and quantum walks: Anomalous diffusion and ballistic tails*, Phys. Rev. A **74**, 020102(R) (2006).

[14] C.Moore and A. Russell *Quantum Walks on the Hypercube*, Proceedings of the 6th International Workshop on Randomization and Approximation Techniques in Computer Science (2002); quant-ph/0104137.

[15] V. Kendon, *Decoherence in quantum walks - a review*, Math. Struct. in Comp. Sci **17** (6), 1169 (2006); quant-ph/0606016.

[16] V. Kendon and B. Tregenna *Decoherence can be useful in quantum walks*, Phys. Rev. A **67**, 042315 (2003).

[17] T.A. Brun, H.A. Carteret and A. Ambainis. The quantum to classical transition for random walks. *Phys Rev. Lett.* 91:130602, 2003.

[18] T.A. Brun, H.A. Carteret and A. Ambainis. Quantum random walks with decoherent coins. *Phys. Rev. A* 67:032304, 2003.

[19] T.A. Brun, H.A. Carteret and A. Ambainis. Quantum walks driven by many coins. *Phys. Rev. A* 67:052317, 2003.

[20] L. Fedichkin, D. Solenov and C. Tamon. *Mixing and decoherence in continuous time quantum walks on cycles*, Quantum Inf. Comp. **6**(3):263-276 (2006).

[21] G. Alagic and A. Russell. Decoherence in quantum walks on the hypercube. *Phys. Rev. A* 72:062304, 2005.

[22] T.D. Mackay, S.D. Bartlett, L.T. Stephenson and B.C. Sanders. Quantum walks in higher dimensions. *J. Phys. A: Math. Gen.* 35:2745, 2002.

[23] A. Romanelli, R. Siri, G. Abal, A. Auyuanet, and R. Donangelo, *Decoherence in the quantum walk on the line*, Physica A **347**, 137 (2004).

[24] G. Abal, R. Donangelo, F. Severo, and R. Siri, *Decoherent quantum walks driven by a generic coin operation*, Physica A **387**, 335 (2007).

[25] A. Oliveira, R. Portugal, and R. Donangelo, *Decoherence in two-dimensional quantum walk*, Phys. Rev. A **74**, 012312 (2006).

[26] D. Shapira, O. Biham, A. J. Bracken, and M. Hackett, *One-dimensional quantum walk with unitary noise* Phys. Rev. A **68**, 062315 (2003).

[27] R. Stengel, *Optimal Control and Estimation*, Courier Dover Publications, 1994

[28] O. Maloyer and V. Kendon, *Decoherence versus entanglement in coined quantum walks*, New J. Phys. **9**, 87 (2007)

[29] P. Richter. *Almost uniform sampling via quantum walk*, New J. Phys. **9** 72, (2007); *Quantum speedup of classical mixing processes* Phys. Rev. A 76, 042306 (2007).

[30] F.L. Marquezino, R. Portugal, G. Abal and R. Donangelo, *Mixing Times in Quantum Walks on the Hypercube*, Phys. Rev. A **77**, 042312 (2008).

[31] A.P. Hines and P.C.E. Stamp, *Quantum Walks, Quantum Gates, and Quantum Computers*, Phys Rev A **75**, 062321 (2007).

[32] A.P. Hines and P.C.E. Stamp, *Decoherence in quantum walks and quantum computers*, arXiv:quant-ph/0711.155

[33] I. Chuang, M. Nielsen, *Quantum Computation and Quantum Information*, Cambridge, 2000;

[34] R. Alicki and K. Lendi, *Quantum Dynamical Semigroups and Applications*, Physics Lecture Notes, Springer Verlag, 1987

[35] Magniez F. *et. al.*, *Serch via Quantum Walk*, quant-ph/0606016

[36] van Dam W *et. al.*, *How Powerful is Adiabatic Computation*, quant-ph/0206003

[37] L. Lovasz, Random Walks on Graphs: A Survay, Bolyai Soc. Math. Studies, vol. , 353, (1996).

[38] B. Misra, E. C. G. Sudarshan, *The Zeno's paradox in quantum theory*, J. Math. Phys. 18, 756 (1977).

[39] M. Mohseni, P. Rebentrost, S. Lloyd, A. Aspuru-Guzik, *Environment-Assisted quantum walks in energy transfer of photosynthetic complexes*, J. Chem. Phys. 129, 174106 (2008).

[40] M. B. Plenio, S. F. Huelga, *Dephasing assisted transport: quantum networks and biomolecules*, New J. Phys. 10, 113019 (2008).

[41] A. M. Childs, *On the relationship between continuous- and discrete-time quantum walk*, arXiv:0810.0312 [quant-ph] (2008).

[42] A. M. Childs, B. W. Reichardt, R. Spalek, S. Zhang, *Every NAND formula of size N can be evaluated in time* $N^{1/2+o(1)}$ *on a quantum computer*, Proceedings of Foundations of Computer Science 2007, 63 (2007).

Chapter 5

Synopsis and Concluding Discussion

The results that we have presented in this dissertation, in general terms, deal with the mechanisms for the efficient quantum and classical algorithms. Here we summarize the essence of the each result that we have presented.

5.1 On NP vs. BQP

The best known quantum algorithms for finding the unique marked element in the list of N elements run in time $\mathcal{O}\left(\sqrt{N}\right)$, what is in accord with the oracular bounds that we currently know. Continuous time quantum algorithms have proved not to fall within the oracular black box formalism and rely on the following three assumptions.

First, unique unstructured search examples always constructed using the *idealized model*, in which the adversary specifies classical n-bit string that is transliterated into the system Hamiltonian.

Second, all currently known continuous time quantum algorithms rely on the fact that an adversary effectively prepared a problem Hamiltonian whose ground state we would like to find since that would, width the high probability, by measurement reveal the problem solution.

Third, virtually all continuous time quantum algorithms rely on the adiabatic theorem that assures that if the initial state of the quantum evolution is the ground state, provided that we change the system Hamiltonian slowly, system final state will also be the ground state.

Current ongoing efforts in adiabatic algorithm for NP-complete problems have almost exclusively focused on showing that choosing the particular initial Hamiltonian, one avoids the exponentially small gap in the instantaneous Hamiltonian, that occur for the hardest problem instances and prevents the subexponential running time.

In this section we have explored continuous time quantum algorithms in two aspects. *First*, we have introduced in this section a new framework of natural continuous quantum algorithms for optimization problems and *second*, we have explored the freedom of transliteration that arises in the context of continuous time quantum algorithms.

The first aspect. Natural quantum algorithm, unlike the adiabatic quantum algorithms, require neither the ground state initialization nor the adiabatic change of the parameters to ensure the measurement of the ground state with the high probability. For the unique marked element unstructured search natural algorithms that employ either time independent or time dependent Hamiltonian can drive the system outside of ground state but ensure that, at the end of computation, with a high probability measurement will reveal the ground state of the system Hamiltonian part that encodes the problem. Since natural algorithms do not rely on the adiabatic approximation of the ground state our analysis is performed by exact solution of the Schroedinger dynamics.

Overall natural quantum algorithms have the following properties: (a) offer arguably simpler formalism in which the running time can be determined from the gap of the time independent Hamiltonian; (b) running time is determined exactly including the scaling constant what is of potential value for experiments; (c) there is not need to initialize the quantum computer in the ground state of the initial Hamiltonian, i.e. initial state can always be a uniform superposition over the computational basis states; (d) Hamiltonian nature of the quantum system ensures that, at the time of the measurement, with the high probability we will measure the ground state of the part of the system Hamiltonian that encodes the problem that we desire to solve.

The second aspect. The freedom of transliteration that arises in any continuous time quantum algorithm is the following: the unique marked element unstructured search problem is classically completely described with the unknown index Boolean string and the absence of structure. Any attempt to transliterate this problem into the Hamiltonian oracle, that is necessarily used in any continuous time quantum algorithm, must in addition to the unknown index of the marked element and the absence of structure, specify in which

basis the part of the Hamiltonian that encodes the problem has no structure.

We have showed that it is fairly easy to transliterate the unique marked element unstructured search into the basis in which, computation takes linear or constant time and the Hamiltonian is very simply described but yet most likely very hard to implement. Concretely, we consider the Hamiltonian $|0\rangle\langle 0|_z + |\psi\rangle\langle\psi|$. We show that if one can efficiently implement this Hamiltonian on the quantum computer then one can solve unique marked element unstructured search in quantum continuous constant time. Unfortunately, we do not know how to implement this Hamiltonian. Depending of the viewpoint, one can view this as either providing a step towards an unlikely result of inclusion of NP in BQP or as showing that this Hamiltonian is NP-hard to implement.

5.2 Multiplicative Weights Algorithms

We present several results related to matrix and vector version of the multiplicative weights algorithm and the exact qualification of the relation between the two. The results are the following.

We present a bit more general version of the known lower bound for the matrix multiplicative weight algorithm, that is good for larger set of adversary strategies.

Moreover, in the restricted setup of nonnegative zero-sum games, we show that unless objective function is redefined, in some rather non-obvious way, the computationally simpler vector multiplicative weight algorithm has at least as good performance as matrix multiplicative weight algorithm. This implies that, the adversary that is providing the response to our queries basically gains nothing in terms of overall loss from changing the basis in which his events are diagonal matrices.

Furthermore, one might be interested in asking the following question, that is very relevant in any streaming application. Is it possible to redefine the original cumulative matrix multiplicative weight algorithm to the truly iterative multiplicative weight algorithm that achieves the identical performance bound? As we show the answer to this question is: yes, and we construct one such algorithm, again within the framework of nonnegative zero-sum games. Consequently, the updates are much less computationally demanding $\mathcal{O}(n^2)$ rather than $\mathcal{O}(n^3)$.

Finally, we show that our multiplicative update algorithm is converging

in strategy to the Nash equilibrium for a non-zero sum game, namely the augmented Shapley game.

5.3 Circuit Complexity Equivalence

The Distortion Lemma on Finsler manifolds and its corollary provide a general tool for relating distances on the manifold with distances on the tangent space. We have derived a generalized linear bounds for the exact simulation of any path on the manifold, in terms of the minimum circuit size and the simulation parameters.

The equivalence between the path on the manifold and circuit size still persists in the case of approximate simulation, provided that the simulation parameters have the appropriate scaling. However, one can not expect better than n^2 times improvement in the minimum circuit size upper bound over the result for standard circuit simulation derived by Nielsen, Dowling, Gu and Doherty [4].

Moreover, if one defines a metric on the manifold that penalizes the hard-to-simulate directions on the tangent space with high cost, that cost is, in effect, translated to the increased ratio between upper and lower bound in minimum circuit size.

5.4 Complete Characterization of Mixing Time on Hypercube

We complete the picture that exists in the literature for quantum random walks on the hypercube with decoherence under the subspace projection decoherence model. This model of decoherence is the most physically realistic form when the quantum random walk is implemented using a register of qubits.

The following are the important points of our work:

- The instantaneous mixing time property of the quantum random walk does not necessarily disappear as the decoherence strength is increased. More precisely, Zeno dynamics prevails only in the precisely specified regions of the $\overrightarrow{\mathbf{r}}, \gamma, \Delta$ parameter space defined in 4.5.2. Consequently,

the hypercube quantum walk with decoherence does not always limit to its classical version as the decoherence is increased. Depending on the direction of the decoherence vector it is possible to have oscillatory dynamics and instantaneous mixing persist as decoherence is increased.

- We showed numerically that, for almost all decoherence directions, a finite decoherence rate exists for optimally fast mixing. This optimal rate is approximately the same as the rate of Hamiltonian evolution. However, we also showed that for certain decoherence directions ($\overrightarrow{\mathbf{r}}$ in the $x-y$ plane) no finite optimal decoherence rate exists and the mixing rate increases without bound as the decoherence rate is increased. This result is particularly relevant given recent results on quantum walk based modeling of excitation transport in biomolecules [39, 40]. In these works it is argued that decoherence can lead to faster hitting times and walker diffusion, and this is explicitly confirmed for a simplified model of dephasing of the quantum walk. Our results suggest that the exact model of the decoherence matters greatly, and therefore, that an accurate model of the environmental interactions is essential to assess the merits or demerits of decoherence to excitation transport.

- Furthermore, we showed in section 4.5.1 that randomizing the decoherence axis yields the depolarizing channel which exhibits both instantaneous and classical mixing regardless of the relative decoherence rate $\frac{\gamma}{\Delta}$. Yet another example of a quantum walk that does not decohere to a classical walk as the decoherence rate is increased. Therefore, introducing randomized decoherence may be an avenue for controlling the mixing behavior of the hypercube random walk.

Appendix A

Generalized Subspace Overlap Theorem

In this appendix we prove the theorem that relates the lowest eigenvalue of the parametric sum of two matrices to the angle of their lowest subspace. Non parametric version of this theorem appears in [8]

Lemma 34 *Let W and F be a non-negative operators, and $\mathcal{L}_1, \mathcal{L}_2$ their null subspaces, where $\mathcal{L}_1 \cap \mathcal{L}_2 = \{0\}$. Then:*

$$H(s) = W + sF \geq s - \cos\theta \tag{A.1}$$

where:

$$\cos\theta = \max_{|\eta_1\rangle \in \mathcal{L}_1,\, |\eta_2\rangle \in \mathcal{L}_2} |\langle \eta_1 | \eta_2 \rangle| \tag{A.2}$$

for $s \in [0, 1]$

Proof. The system Hamiltonian can be written as:

$$H(s) = W + sF = I + sI - \Pi_1 - \Pi_2 \tag{A.3}$$

We are interested in the bound that relates to the smallest eigenvalue gap for the sum of two matrices. In our case:

$$W = I - |z\rangle\langle z| \equiv I - \Pi_1\,, \text{ where } |z\rangle = \frac{1}{2^n}\sum_{i=1}^{n} |i\rangle \tag{A.4}$$

$$F = I - |m\rangle\langle m| \equiv I - \Pi_2\ , \text{ where } |m\rangle \text{ is the marked element in the list;} \tag{A.5}$$

Let $\lambda \geq 0$ be the eigenvalue of the eigenvector $|\xi\rangle$, then:

$$\lambda = \langle\xi|\Pi_1 + s\Pi_2|\xi\rangle = u_1^2 + su_2^2 \ , \text{ where } \Pi_i |\xi\rangle = u_i |\eta_i\rangle \ , \ i = \{1,2\} \tag{A.6}$$

The equation A.1 is equivalent to:

$$\Pi \equiv \Pi_1 + s\Pi_2 = |z\rangle \langle z| + s |m\rangle \langle m| \leq 1 + \cos\theta \tag{A.7}$$

Now we can see that:

$$\lambda = \langle\xi|\Pi|\xi\rangle = |\langle\xi|z\rangle|^2 + s |\langle\xi|m\rangle|^2 = u_1^2 + su_2^2 \tag{A.8}$$

Moreover, on the other side:

$$\begin{aligned}
\lambda^2 &= \langle\xi|\lambda\cdot\lambda|\xi\rangle = (\langle\eta_1| u_1 + \langle\eta_2| su_2)(u_1 |\eta_1\rangle + su_2 |\eta_2\rangle) && \text{(A.9)}\\
&= u_1^2 + (su_2)^2 + u_1u_2s(\langle\eta_1|\eta_2\rangle + \langle\eta_2|\eta_1\rangle) && \text{(A.10)}\\
&= u_1^2 + (su_2)^2 + 2u_1u_2s\,\mathrm{Re}(\langle\eta_1|\eta_2\rangle) && \text{(A.11)}\\
&\leq u_1^2 + (su_2)^2 + 2u_1u_2s\,|\mathrm{Re}(\langle\eta_1|\eta_2\rangle)| && \text{(A.12)}\\
&\leq u_1^2 + su_2^2 + 2u_1u_2s\,|\mathrm{Re}(\langle\eta_1|\eta_2\rangle)| && \text{(A.13)}\\
&= \lambda + 2u_1u_2s\,|\mathrm{Re}(\langle\eta_1|\eta_2\rangle)| && \text{(A.14)}
\end{aligned}$$

Follows that:

$$\lambda - \lambda^2 \geq -2u_1u_2\,|\mathrm{Re}(\langle\eta_1|\eta_2\rangle)| \equiv -2u_1u_2s\chi \ , \text{ where } \chi \equiv |\mathrm{Re}(\langle\eta_1|\eta_2\rangle)| \ ; \tag{A.15}$$

$$\lambda - \lambda^2 + \lambda\chi \geq -2u_1u_2s\chi + \lambda\chi \tag{A.16}$$

$$\begin{aligned}
(1+\chi)\lambda - \lambda^2 &\geq \left(-2u_1u_2s + u_1^2 + su_2^2\right)\chi && \text{(A.17)}\\
&\geq \left(-2u_1u_2s + u_1^2 + s^2u_2^2\right)\chi && \text{(A.18)}\\
&\geq (u_1 + su_2)^2\chi \geq 0 && \text{(A.19)}
\end{aligned}$$

Therefore, since $\lambda \geq 0$:

$$\lambda^2 \leq (1+\chi)\lambda \tag{A.20}$$

$$\lambda \leq 1 + \chi \leq 1 + \max_{|\eta_1\rangle\in\mathcal{L}_1,\,|\eta_2\rangle\in\mathcal{L}_2} |\langle\eta_1|\eta_2\rangle| \tag{A.21}$$

since $|\mathrm{Re}(\langle\eta_1|\eta_2\rangle)| \leq \max_{|\eta_1\rangle\in\mathcal{L}_1,\,|\eta_2\rangle\in\mathcal{L}_2} |\langle\eta_1|\eta_2\rangle|$.

Substituting back to the original equation we get the desired result:

$$I + sI - (|z\rangle\langle z| + s|m\rangle\langle m|) \geq s - \max_{|\eta_1\rangle\in\mathcal{L}_1,\,|\eta_2\rangle\in\mathcal{L}_2} |\langle\eta_1|\eta_2\rangle| \tag{A.22}$$

■