

Spatial Search by Quantum Walk with a Randomized Local Start State

by

Fen Zhao

Submitted to the Department of Physics
in partial fulfillment of the requirements for the degree of

Bachelor of Science in Physics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2004

© Massachusetts Institute of Technology 2004. All rights reserved.

Author

Department of Physics

May 7, 2004

Certified by

Edward Farhi

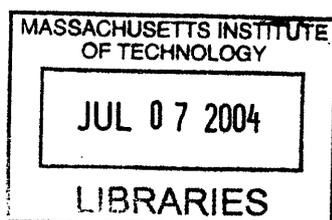
Professor of Physics

Thesis Supervisor

Accepted by

Professor David E. Pritchard

Thesis Coordinator



ARCHIVES

Spatial Search by Quantum Walk with a Randomized Local Start State

by

Fen Zhao

Submitted to the Department of Physics
on May 7, 2004, in partial fulfillment of the
requirements for the degree of
Bachelor of Science in Physics

Abstract

In this thesis, we present a quantum walk algorithm for spatial search of a periodic lattice. Our algorithm is a variation of the Childs and Goldstone algorithm for spatial search, but begins in a randomly selected local initial state rather than a uniformly delocalized one. We analytically calculate the running time of our algorithm on the complete graph and find it to be $O(\sqrt{N})$. We reduce the analysis of our algorithm to that of the Childs and Goldstone algorithm by comparing the eigenvalue conditions of the Hamiltonians used in the two algorithms. We numerically show that the two Hamiltonians have similar eigenvalue conditions when the starting state is a certain extremal vertex of the lattice. We also study the behavior of the algorithm when we move the start state away from this extremal vertex. Finally, we numerically analyze the behavior of our algorithm on 5 and 4 dimensional lattices. In the 5 dimensional case, we appear to be able to achieve a $O(\sqrt{N})$ running time. In the 4 dimensional case, previous analysis indicates there may be additional factors of $\log^c N$ in the running time of our algorithm. Numerically, we are not able to determine whether this logarithmic factor exists. However, the numerical evidence does indicate that the running time of our algorithm is $O(\sqrt{N})$, up to some factor of $\log^c N$.

Thesis Supervisor: Edward Farhi
Title: Professor of Physics

Acknowledgments

My thanks go to Professor Edward Farhi for supervising this thesis project, to Professor Jeffrey Goldstone for his helpful comments, and to Andrew Childs, whose help made this project possible.

Contents

1	Introduction and background	11
1.1	Overview of quantum computing	12
1.2	Grover's algorithm	13
1.3	Spatial search	14
1.3.1	Continuous time quantum walks	15
1.3.2	Spatial search using quantum walks	16
2	The complete graph	17
2.1	Previous analysis	17
2.2	New algorithm	18
2.3	The evolution of the system	20
3	Dependence of the algorithm on the initial state	23
3.1	Evolution of the system starting from the extremal vertex	26
3.2	Variation of the start state away from the extremal case	26
4	Numerical analysis of algorithm for $d = 5$ and $d = 4$	33
4.1	The 5 dimensional lattice	33
4.2	The 4 dimensional lattice	34
5	Conclusions	45

List of Figures

2-1	Plot of the probability of finding $ w\rangle$, $ b\rangle$, and $ s\rangle$ over time for the complete graph.	21
3-1	Dependence of NG_1 on N for $d = 5$. We find that $G_1 = O(N^{-0.6})$. . .	25
3-2	The probability of finding the system in the start state, the target state, or in the uniform linear superposition of states at different points in time for $N = 8^5$	27
3-3	$NG_1(b)$ as a function of distance between the starting vertex and the extremal point	29
3-4	The probability of measuring the target state at time t_e for different starting states differentiated by their NG_1 value	30
3-5	Probability of measuring the target state over time for a starting state far from the extremal vertex.	31
4-1	Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 6^5$ and $N = 7^5$. . .	36
4-2	Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 8^5$ and $N = 9^5$. . .	37
4-3	t_e for different values of \sqrt{N} for $d = 5$; $t_e = O(\sqrt{N})$	38
4-4	Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 6^4$ and $N = 7^4$. . .	39
4-5	Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 8^4$ and $N = 9^4$. . .	40

4-6	Average success probability of the algorithm for different values of N for $d = 4$	41
4-7	t_e for different values of N for $d = 4$	42
4-8	Log plot of t_e for different values of N for $d = 4$; we see that $t_e =$ $O(N^{0.53 \pm 0.01})$	43

Chapter 1

Introduction and background

In this paper, we present a quantum walk algorithm for doing spatial search of a d dimensional periodic lattice. This algorithm is based on the spatial search algorithm presented by Childs and Goldstone in [9]. Their algorithm starts from a uniform superposition over all vertices. This uniform state can be prepared locally from a single vertex starting state, but then the full algorithm does not use a single time-independent Hamiltonian. Our algorithm attempts to make the entire algorithm time independent by beginning at a randomly selected vertex and evolving using a time independent Hamiltonian.

Chapter one begins with an overview of quantum computing and outlines the previous research done on spatial search and quantum walks. We will also introduce the basic structure of our algorithm.

Chapter two examines the high dimensional case of spatial search on a complete graph. We analytically calculate the behavior of our algorithm in this highly symmetric case and use it as motivation for the numerical analysis of more complicated systems.

Chapter three presents analytic and numerical analysis of the dependence of the behavior of our algorithm on the location of the initial state. We find that there is one ideal starting vertex where the behavior of the system becomes simpler to analyze. We then numerically study the behavior of our algorithm for other starting vertices.

Chapter four presents numerical analysis of our algorithm for 5 and 4 dimensional

lattices. In 5 dimensions, the results are consistent with a running time of $O(\sqrt{N})$; this is the same as the running time found in [9]. In 4 dimensions, our results are not conclusive, but our data suggests that the running time has a $O(\sqrt{N})$ polynomial dependence.

1.1 Overview of quantum computing

The physical representation of information controls the rules of how that information can be processed. Modern, or “classical,” computers store information using electronics, and manipulation of that information is governed by the laws of classical electromagnetism. A quantum computer, however, represents information as the states of a quantum system; computations on a quantum computer are governed by the rules of quantum mechanics. Because the laws of quantum mechanics offer ways of manipulating a physical system that are impossible in classical mechanics, one can make quantum computations with running times lower than their classical analogs.

Richard Feynman was the first to suggest the possibility of computation speedup by using a quantum computer. In 1994, Peter Shor created a quantum algorithm for factoring and discrete log which ran exponentially faster than any known classical algorithm [15]. In 1996, Lov Grover developed an algorithm for quantum search that was provably faster than any classical algorithm [13],[4]. The quantum complexity hierarchy has been deeply studied as well (for more information, see [14]), and many quantum complexity classes are believed to be larger than their classical counterparts.

A quantum bit is called a qubit. It is represented as a normalized vector that is the linear combination of two orthogonal states: $|\psi_1\rangle = c_0|0\rangle + c_1|1\rangle$. Operations on this bit can be seen mathematically as 2×2 unitary matrices. Multiple qubit states are described by higher dimensional vectors; in 4 dimensions, the vector is $|\psi_2\rangle = c_{00}|00\rangle + c_{01}|01\rangle + c_{10}|10\rangle + c_{11}|11\rangle$.

Quantum computation has traditionally been done with qubit gates, which are discrete operations that manipulate qubits in similar ways to logic gates for bits. These gates are unitary operations that act on the quantum state. Qubit gates are

created by manipulating a Hamiltonian via an external control mechanism. A system with time-independent Hamiltonian H evolves as $|\Psi(t)\rangle = e^{-iHt/\hbar} |\Psi(0)\rangle$. By applying specific Hamiltonians for some set amount of time, one can simulate a quantum gate. This is a discrete time conception of quantum computing. Quantum computing can also be done using continuous time algorithms. Such algorithms generally evolve a system by some unchanging Hamiltonian [12] or slowly changing Hamiltonian [10]. Our algorithm is a continuous time algorithm, using an time-independent Hamiltonian.

1.2 Grover's algorithm

In [13], L. Grover presented an algorithm to search N items for a marked item. We will assume that only one item is marked in our discussion of Grover's algorithm and all later search algorithms. Classically this problem requires $O(N)$ operations. Grover's algorithm required only $O(\sqrt{N})$ operations. This algorithm can be applied to speed up many algorithms that use search heuristics.

A database of N elements can be represented by a function on $\log(N)$ bits; each element is referenced by its index. To mark an item, have the function return a 1 on the input of the item's index. The function will return 0 on the index of an unmarked item. Say we are given an oracle for a database that acts as f . Classically, we would, in the worst case, have to call the oracle $N - 1$ times to find a marked item.

Grover's algorithm uses a quantum oracle only $O(\sqrt{N})$ times to find a marked item. The initial state is in a uniform superposition of all the indices of the items

$$|s\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$$

The oracle is of the form

$$|x\rangle \rightarrow (-1)^{f(x)} |x\rangle$$

Each round of the Grover algorithm applies the oracle, and then applies $2|s\rangle\langle s| - I$ to the state. The algorithm requires $\pi\sqrt{N}/4$ rounds. This operation can be visualized as

a rotation in the two dimensional subspace spanned by the marked, or “target,” state, $|w\rangle$, and the orthogonal component of the uniform starting state, $|r\rangle = |s\rangle - \langle w|s\rangle |w\rangle$. Each round rotates the state of the system from $|r\rangle$ to $|w\rangle$ by $\text{asin}\frac{2\sqrt{N-1}}{N}$. Therefore, after $\pi\sqrt{N}/4$ rotations, the system will have rotated approximately $\pi/2$ radians and be mostly in the $|w\rangle$ state. The system can then be measured in the item index basis and have a high probability of being found in the target state.

1.3 Spatial search

Spatial search is a variant of the Grover search problem (see [3], [9], [1]). In this case, we imagine that the N items are stored in N different memory locations, and that we can only move between certain memory locations. For example, if the N items are stored in a d dimensional square lattice, we can only make local moves from an item to the $2d$ items adjacent to it. In terms of formulating the problem as a Hamiltonian acting on the item indices, we restrict ourselves to Hamiltonians with non-zero matrix elements only between states representing vertices that are connected to each other.

Currently, there are many different spatial search algorithms. In [1] Ambainis and Aaronson presented a discrete recursive algorithm that searches a d dimensional database in $O(\sqrt{N})$ oracle calls for $d \geq 3$, and in $O(\sqrt{N}\log^2 N)$ oracle calls for $d = 2$. In [9], Childs and Goldstone presented an algorithm for spatial search using continuous time quantum walks. This algorithm performed in $O(\sqrt{N})$ time for $d \geq 5$, and in $O(\sqrt{N}\log N)$ time for $d = 4$. The algorithm failed to give any speedup for lower dimensions. In [2], Ambainis presented an algorithm using discrete time quantum walks with a coin label that performed spatial search in $O(\sqrt{N})$ oracle calls for $d \geq 3$, and in $O(\sqrt{N}\log N)$ oracle calls for $d = 2$. Recently, Childs and Goldstone have developed another continuous time quantum walk algorithm for spatial search using the Dirac Hamiltonian and extra spin degrees of freedom [7]. This algorithm has a running time of $O(\sqrt{N}\log N)$ in $d = 2$.

All of the previously mentioned quantum walk algorithms for spatial search have an initial state that is delocalized over the entire graph. In the spirit of the this

“local” problem, one might require beginning from a local state. However, one can generate this delocalized starting state from a localized state with $O(N^{1/d})$ local operations. One can also run the continuous or discrete time algorithm backward, by setting the initial local state as the target state. Running the algorithm backwards will produce a state that is close to a uniform superposition, which can then be used with the original Hamiltonian. In this paper, we will investigate the possibility of running the algorithm from a local state directly, with just a single unchanging oracle Hamiltonian.

1.3.1 Continuous time quantum walks

We will use the technique of continuous time quantum walks [11] in our algorithm. A quantum walk on a graph is like a classical random walk on a graph. An undirected graph a set of vertices with edges between pairs of vertices. It can be defined by an adjacency matrix

$$A_{jk} = \begin{cases} 1 & \text{edge } (j, k) \text{ in graph} \\ 0 & \text{otherwise} \end{cases}$$

The degree of a vertex i is the number of edges that have i as one of its vertices. Define the Laplacian $L = A - D$ where D is diagonal with $D_{ii} = \text{degree}(i)$. Note that this Laplacian is “local;” a vertex only interacts with the vertices connected to it.

In a continuous classical walk, there is some probability per unit time that one moves from a vertex to the vertices sharing an edge with it. The probability of being at any vertex j , $p_j(t)$ evolves according to

$$\frac{dp_j(t)}{dt} = \gamma \sum_k L_{jk} p_k(t) \quad (1.1)$$

Quantum mechanically, we represent each vertex in a graph with N vertices as an orthogonal state, $|j\rangle$, in a N dimensional Hilbert space. In the state $|\psi\rangle$, the probability of being at vertex j is defined as $|\langle j|\psi\rangle|^2$, the probability of measuring state $|j\rangle$ in the orthogonal basis of vertices. Note that the Laplacian previously

defined is a Hermitian matrix. If we define the Hamiltonian acting on the system as the Laplacian previously described, Schrödinger's equation gives an equation for the evolution of the system

$$i \frac{d \langle j | \psi \rangle}{dt} = \sum_k H_{jk} \langle k | \psi \rangle \quad (1.2)$$

which is analogous to Equation 1.1. The difference lies in the factor of i , which allows for significant interference effects that make quantum walks algorithmically useful in ways that classical walks are not. For example, in [8], an exponential quantum speedup was shown for a graph problem using a quantum walk algorithm.

1.3.2 Spatial search using quantum walks

Childs and Goldstone's [9] first algorithm for spatial search uses the Hamiltonian

$$H = -\gamma L - |w\rangle \langle w|$$

where L is the Laplacian for a d dimensional periodic square lattice of N vertices, and $|w\rangle$ is the marked vertex. The initial state is the uniform superposition of all vertices, $|s\rangle$. For certain values of γ , this Hamiltonian drives oscillations between the $|s\rangle$ and $|w\rangle$ state so that the amplitude of the target state is $O(1)$ in $O(\sqrt{N})$ time for $d > 4$ and $O(1/\sqrt{\log N})$ in $O(\sqrt{N \log N})$ time for $d = 4$.

In our algorithm, we will use the same value of γ , but begin in some randomly selected vertex, $|b\rangle$. We also add a extra term to the Hamiltonian

$$H = -\gamma L - |w\rangle \langle w| - \alpha |b\rangle \langle b|.$$

We will select $\alpha = 1$ because we want a symmetry between the $|w\rangle$ and $|b\rangle$ states; we want this Hamiltonian to drive oscillations between the $|b\rangle$ and the $|w\rangle$ states with $|s\rangle$ as an intermediate state. Also note that this oscillation may only occur for certain values of b . We hope that this Hamiltonian will give an algorithm for spatial search, which will be the case if this oscillatory behavior occurs quickly enough for sufficiently many values of b .

Chapter 2

The complete graph

2.1 Previous analysis

A complete graph on N vertices is the N -vertex graph where every vertex is connected to every other vertex. Analysis of an optimal quantum search algorithm for a complete graph using initial state $|s\rangle$ can be found in [12]. It was found that evolving with the Hamiltonian

$$H = E(|s\rangle\langle s| + |w\rangle\langle w|) \quad (2.1)$$

for the complete graph with N vertices made the amplitude of the marked vertex 1 at

$$t = \frac{\pi N^{1/2}}{2E}$$

Analysis was done by noting that the Hamiltonian only affected the two-dimensional subspace spanned by $|w\rangle$ and the orthogonal vector $|s\rangle - \langle w|s\rangle|w\rangle$.

In [9], Equation 2.1 was interpreted as a quantum walk Hamiltonian of the form previously discussed by noting that we can add a multiple of the identity to the Laplacian of the complete graph to have the Hamiltonian found in [12].

$$L + NI = N|s\rangle\langle s|$$

This makes the Hamiltonian

$$H = -\gamma N |s\rangle \langle s| - |w\rangle \langle w|$$

The critical value of γ was found to be $\gamma = 1/N$, giving the Hamiltonian proposed in [12].

2.2 New algorithm

Now we will analyze the complete graph problem starting from some randomly selected starting state $|b\rangle \neq |w\rangle$, and evolving with the Hamiltonian

$$H = E(|s\rangle \langle s| + |w\rangle \langle w| + |b\rangle \langle b|)$$

In this case, we begin with a state not in the two dimensional subspace previously analyzed. However, $|b\rangle$ is orthogonal to $|w\rangle$ and mostly orthogonal to $|s\rangle$. Therefore we can see that the Hamiltonian acts non-trivially only on a three dimensional subspace. Note that all bs are equivalent.

Note that $|w\rangle$ and $|b\rangle$ are orthogonal to each other since they are different vertices in the graph. Calling the overlap between a vertex and the superposition $x = \langle w|s\rangle = \langle b|s\rangle = 1/\sqrt{N}$, the third basis vector in our three dimensional subspace is

$$|r\rangle = \frac{|s\rangle - x|w\rangle - x|b\rangle}{\sqrt{1 - 2x^2}}$$

In the basis of $|r\rangle$, $|w\rangle$, and $|b\rangle$, our Hamiltonian is

$$E \begin{pmatrix} 1 - 2x^2 & x\sqrt{1 - x^2} & x\sqrt{1 - x^2} \\ x\sqrt{1 - x^2} & x^2 + 1 & x^2 \\ x\sqrt{1 - x^2} & x^2 & x^2 + 1 \end{pmatrix}$$

Table 2.1: The eigenvectors and eigenvalues of H

eigenvectors	eigenvalues
$ f\rangle = \begin{pmatrix} -\frac{2x-\sqrt{2}}{\sqrt{1-2x^2}} \\ 1 \\ 1 \end{pmatrix}$	$E(1+x\sqrt{2})$
$ g\rangle = \begin{pmatrix} -\frac{2x+\sqrt{2}}{\sqrt{1-2x^2}} \\ 1 \\ 1 \end{pmatrix}$	$E(1-x\sqrt{2})$
$ h\rangle = \begin{pmatrix} 0 \\ -1 \\ 1 \end{pmatrix}$	E

The state of the system at time t will be

$$|\psi_b(t)\rangle = e^{-iHt} |b\rangle$$

To find the state of the system, we can decompose $|b\rangle$ into a linear combination of the eigenvectors of H , which can be found in Table 2.1.

$$|b\rangle = p|f\rangle + \left(\frac{1}{2} - p\right)|g\rangle + \frac{1}{2}|h\rangle$$

where $p = \frac{\sqrt{2}(\sqrt{2}+2x)}{8}$. Using

$$e^{-iHt} |\alpha\rangle = e^{-iE_\alpha t} |\alpha\rangle$$

for any eigenvector $|\alpha\rangle$, we can then easily analyze the evolution of the system. With some simple calculation, we can find that the probability of finding the state at time t in $|w\rangle$.

$$|\langle w|\psi_b(t)\rangle|^2 = 2p^2 - p + \frac{1}{2} + 2p\left(\frac{1}{2} - p\right)\cos(2\sqrt{2}xEt) - \frac{1}{2}\cos(\sqrt{2}xEt) \quad (2.2)$$

This is a periodic function that has a maximum value of 1 at

$$t = \frac{\pi}{\sqrt{2Ex}}$$

Therefore, the running time of this algorithm is

$$t_m = \frac{\pi N^{1/2}}{\sqrt{2E}} = O(\sqrt{N})$$

which is optimal by the lower bound found in [4].

2.3 The evolution of the system

In the same manner as previously used to find 2.2, we can find

$$|\langle b|\psi_b(t)\rangle|^2 = 2p^2 - p + \frac{1}{2} + 2p \left(\frac{1}{2} - p \right) \cos(2\sqrt{2x}Et) + \frac{1}{2} \cos(\sqrt{2x}Et) \quad (2.3)$$

$$|\langle s|\psi_b(t)\rangle|^2 = 1 - |\langle b|\psi_b(t)\rangle|^2 - |\langle w|\psi_b(t)\rangle|^2 \quad (2.4)$$

In Fig. 2-1, we can see that the system oscillates from the $|b\rangle$ state to the $|w\rangle$ state with frequency $1/2t_m$, with $|r\rangle$ as an intermediate state. This is the behavior that we will try to emulate in our algorithm for a d dimensional lattice. Instead of a system that oscillates between the superposition and target states, we will try to have the system oscillate between the start state and the target state, using the superposition state as an intermediate.

Childs and Goldstone's algorithm creates a target state amplitude that is $O(1)$ in $O(\sqrt{N})$ time for all $d \geq 5$. We will demonstrate numerically that our algorithm appears to have the same behavior for $d = 5$ for a large enough number of $|b\rangle$ states. Because our algorithm emulates the behavior of the Childs Goldstone algorithm, and because our algorithm creates a target state amplitude that is $O(1)$ in $O(\sqrt{N})$ time in the high dimensional case of the complete graph, we argue that the same should occur for all $d > 5$ as well.

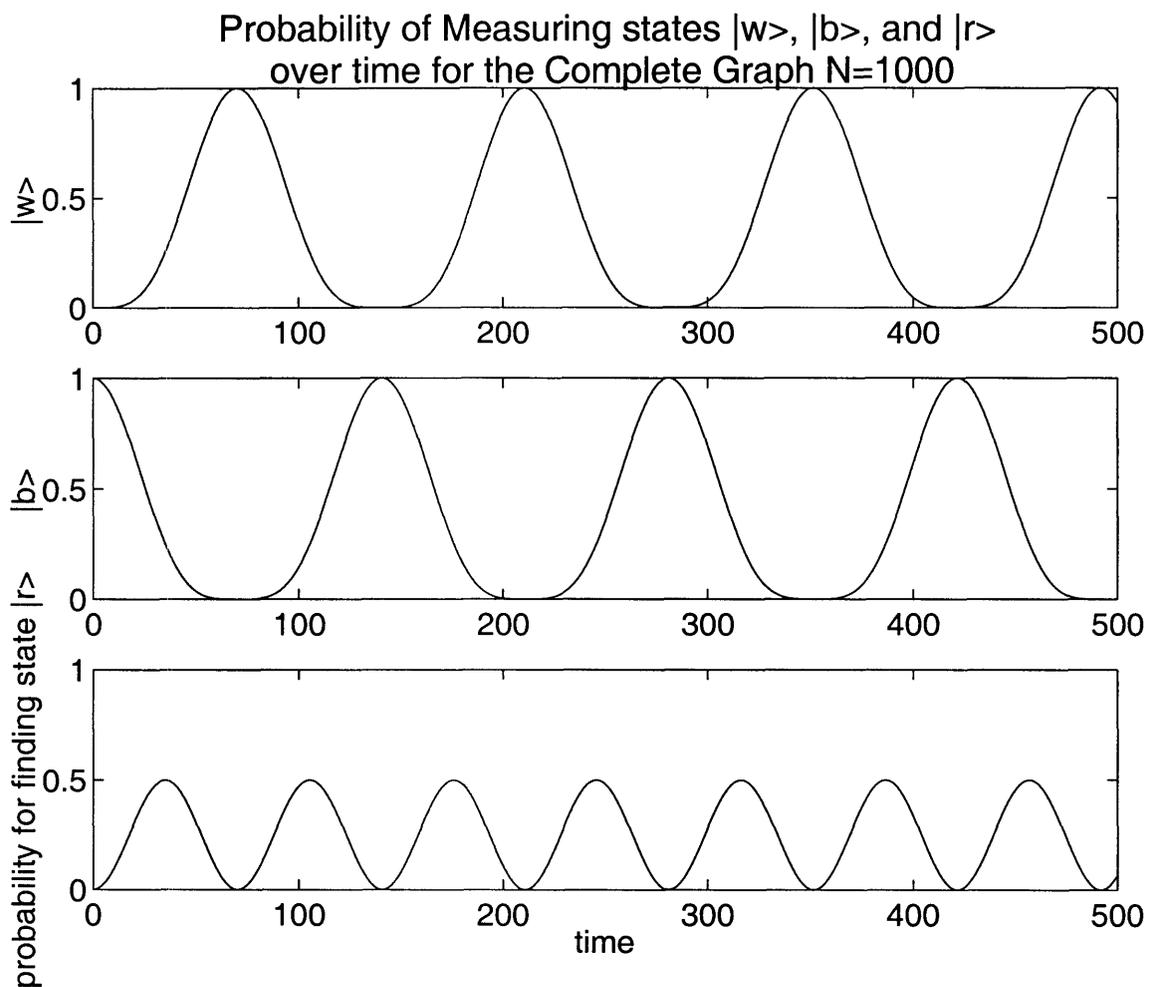


Figure 2-1: Plot of the probability of finding $|w\rangle$, $|b\rangle$, and $|s\rangle$ over time for the complete graph.

Chapter 3

Dependence of the algorithm on the initial state

In this chapter, we will analyze the behavior of our algorithm for different starting states. We will begin by trying to find a starting state where analysis of our algorithm will be similar to the analysis of the Childs and Goldstone algorithm.

The behavior of the quantum system in Childs and Goldstone's algorithm is determined by the spectrum of their Hamiltonian. The eigenvalues, E_{CG} , of their Hamiltonian can be related to the spectrum of the Laplacian by the equation

$$F(E_{CG}) = 1, \quad F(E) = \frac{1}{N} \sum_k \frac{1}{\gamma \mathcal{E}(k) - E}$$

Here, $\mathcal{E}(k)$ are the eigenvalues of the Laplacian.

$$\mathcal{E}(k) = 2 \left(d - \sum_{j=1}^d \cos k_j \right)$$

where

$$k_j = \frac{2\pi m_j}{N^{1/d}}$$

and

$$m_j = \begin{cases} 0, \pm 1, \dots, \pm \frac{1}{2}(N^{1/d} - 1) & \text{if } N^{1/d} \text{ odd} \\ 0, \pm 1, \dots, \pm \frac{1}{2}N^{1/d} & \text{if } N^{1/d} \text{ even.} \end{cases}$$

In [6], Childs calculates that a similar eigenvalue condition is placed on a system driven by the Hamiltonian

$$H = -\gamma L - |w\rangle \langle w| - |b\rangle \langle b|$$

starting from state $|b\rangle$. Childs finds that the constraint for the eigenvalues of our Hamiltonian, E_b , is

$$1 = F(E_b) \pm G(E_b)$$

where $F(E)$ is defined as before, and

$$G(E) = \frac{1}{N} \sum_k \frac{\cos(k \cdot b)}{\gamma \mathcal{E}(k) - E}$$

where b is the d component vector with $b_j \in \{0, 1, \dots, N^{1/d} - 1\}$. Without loss of generality, we set the origin at the marked state: $w_j = 0$ for all j .

Taylor expanding both functions results in

$$\begin{aligned} F(E) &= -\frac{1}{NE} + \frac{1}{N} \sum_{k \neq 0} \frac{1}{\gamma \mathcal{E}(k)} + \frac{E}{N} \sum_{k \neq 0} \frac{1}{\gamma^2 \mathcal{E}(k)^2} \dots \\ G(E) &= -\frac{1}{NE} + \frac{1}{N} \sum_{k \neq 0} \frac{\cos(k \cdot b)}{\gamma \mathcal{E}(k)} + \frac{E}{N} \sum_{k \neq 0} \frac{\cos(k \cdot b)}{\gamma^2 \mathcal{E}(k)^2} \dots \end{aligned}$$

$G(E)$ will typically be difficult to analyze, but in the extremal case where $b_j = \frac{1}{2}N^{1/d}$ for all j , $\cos k \cdot b = \pm 1$, so the second term of $G(E)$ is proportional to

$$G_1 = \frac{1}{N} \sum_{k \neq 0} \frac{\cos k \cdot b}{\mathcal{E}(k)} = \frac{1}{N} \sum_{k \text{ even}} \frac{1}{\mathcal{E}(k)} - \frac{1}{N} \sum_{k \text{ odd}} \frac{1}{\mathcal{E}(k)}.$$

Here, k being odd means $\sum_j m_j$ is odd and vice versa.

We expect the value of G_1 to be very small. In figure 3-1, we plot the values

of NG_1 for $d = 5$ and find that $NG_1 = (0.08 \pm 0.02)N^{0.4 \pm 0.02}$ with an R^2 value of 0.9979¹. This means that $G_1 = o(N^{-1/2})$. Since $E = \Theta(\sqrt{N})$ in this system, $1/NE = \Theta(N^{-1/2})$, and $G(E)$ will be dominated by the $\frac{1}{NE}$ term.

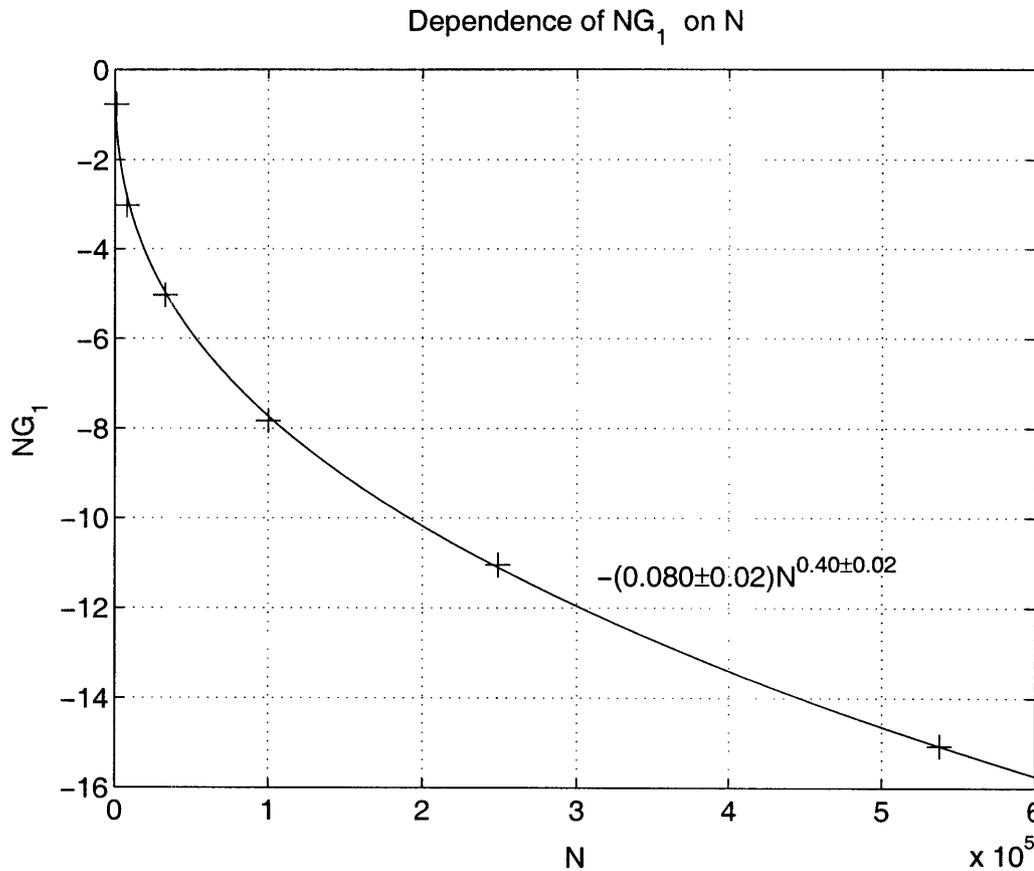


Figure 3-1: Dependence of NG_1 on N for $d = 5$. We find that $G_1 = O(N^{-0.6})$.

Therefore, in the high N limit, the eigenvalue equation becomes

$$1 = F(E) \pm \frac{1}{NE}$$

When $\frac{1}{NE}$ is added, we have the same eigenvalue equation as the Childs and Goldstone algorithm, only with an extra constant of 2 on the $\frac{1}{NE}$ term. We therefore expect our algorithm, starting from this extremal vertex and with the same value of γ as defined

¹ R^2 is the correlation coefficient (It is also known as the product-moment coefficient of correlation or Pearson's correlation). It describes the quality of a least squares fitting of data. $R^2 = \frac{SSR}{ss_{yy}}$, where SSR is the sum of the square residuals, and ss_{yy} is the sum of the residuals of the dependent variable. Complete correlation corresponds to an R^2 value of 1.

in [9], to behave asymptotically like the Childs and Goldstone algorithm; the running time should only be different by a constant multiplicative factor.

3.1 Evolution of the system starting from the extremal vertex

In figure 3-2, we see the evolution of a $d = 5$ and $N = 8^5$ lattice, starting from the extremal vertex. To calculate the evolution of the system, we use a fourth degree Runge Kutta approximation on Equation 1.2.

Comparing this graph to figure 2-1, we see that the system behaves very similarly to the complete graph case. However, the amplitude of the target state never reaches 1. Also note that the sum of the probabilities of being in the three states does not always equal 1; the system is not in a perfect three state system. In general, however, the three dimensional subspace behaves like in the complete graph. In our algorithm, we will measure the system at the time of the first peak of the target state in this extremal case, t_e . In Figure 3-2, $t_e \approx 450$.

3.2 Variation of the start state away from the extremal case

Starting from a different location than the extremal vertex may significantly alter $G(E)$, and therefore change the behavior of our algorithm. The $\frac{1}{NE}$ is independent of starting point, but we only know that G_1 is small for the extremal starting point. Define

$$G_1(b) = \frac{1}{N} \sum_{k \neq 0} \frac{\cos k \cdot b}{\mathcal{E}(k)}$$

for any starting point b . In Figure 3-3, we randomly select some starting points and calculate $NG_1(b)$ as a function of the Euclidean distance² between b and the extremal

²Because the lattice is periodic, there are many different values for the Euclidean distance between two vertices. We pick the smallest one.

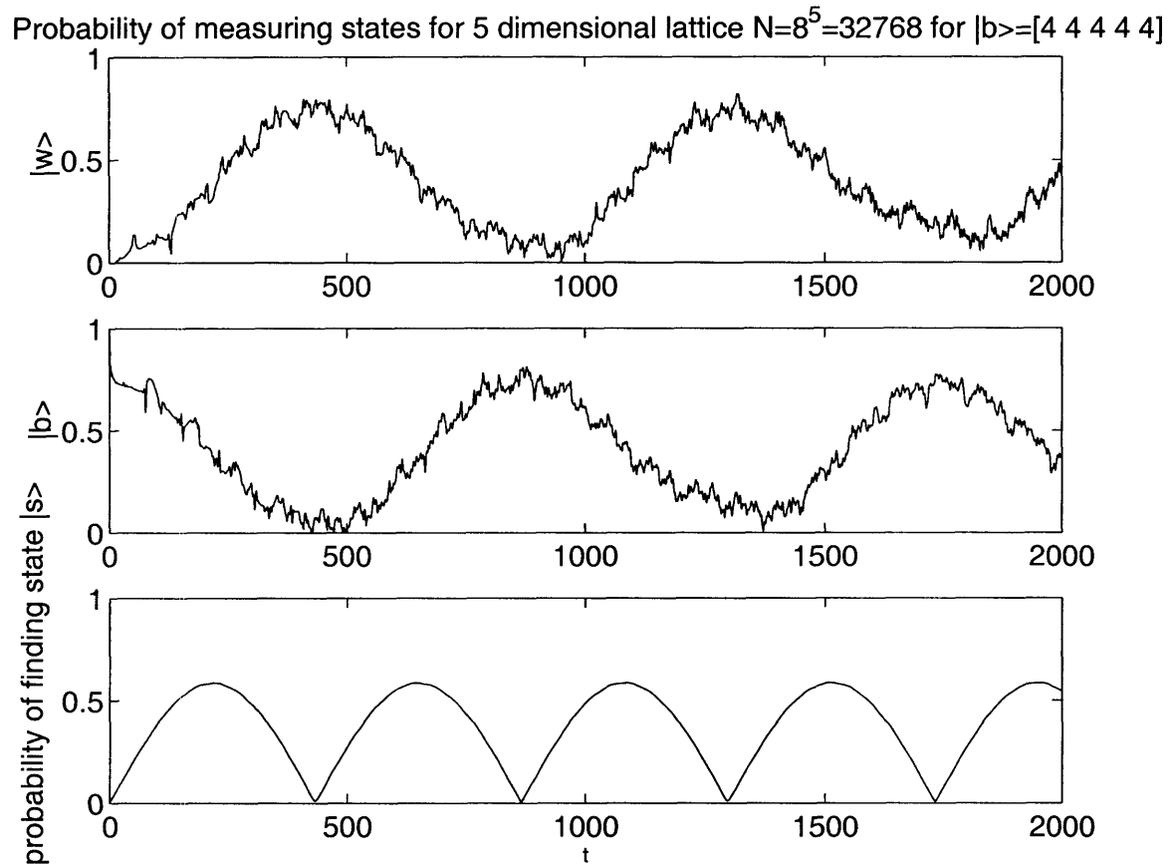


Figure 3-2: The probability of finding the system in the start state, the target state, or in the uniform linear superposition of states at different points in time for $N = 8^5$.

point. In Figure 3-4, we see how the amplitude of the system at time t_e is dependent on the value of $NG_1(b)$.

In Figure 3-4, one can see that large values of G_1 will cause the amplitude of the target state to diminish dramatically. In Figure 3-3, we see that these large values of G_1 occur when the starting point is far away from the extremal vertex. Figure 3-5 shows the behavior of the target state probability over time for a starting state that is far from the extremal vertex for $N = 6^5$. The maximum amplitude is significantly lower than those of points near the extremal vertex. There are also some high frequency oscillations and some asymmetry that we do not see in Figure 3-2.

The question is then, how far can the starting point be from the extremal vertex before the behavior of the algorithm degrades? From Figure 3-3, the value of G_1 for both $N = 6^5$ and $N = 8^5$ does not deviate from G_1 at the extremal vertex until

the distance is approximately $2/3$ the maximum distance possible from the extremal vertex. Note that for this randomly chosen a set of starting points, most points have small G_1 values. In Figure 3-4, note that most points have high amplitudes and small G_1 values, and only a few points have significantly lower amplitudes.

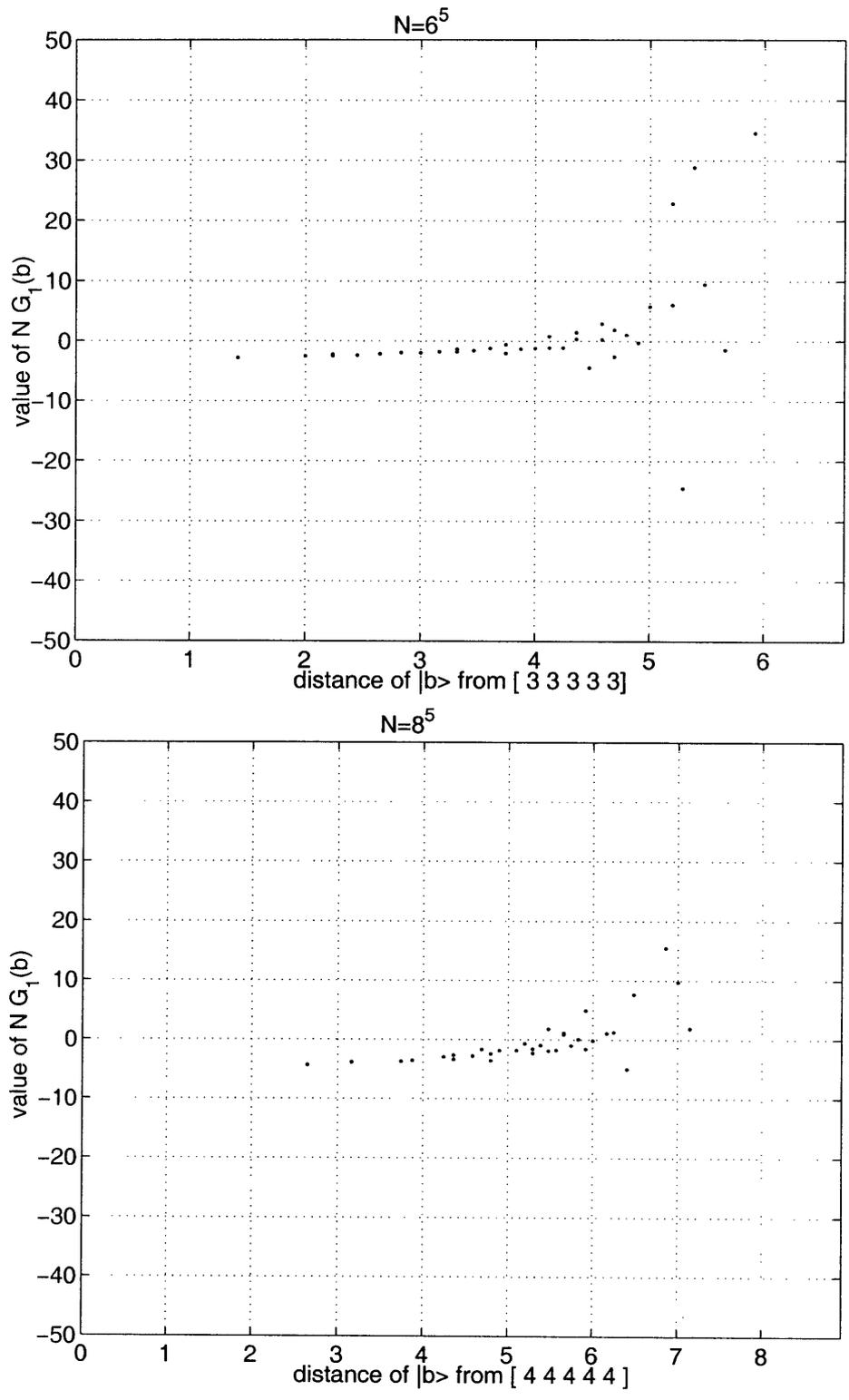


Figure 3-3: $NG_1(b)$ as a function of distance between the starting vertex and the extremal point

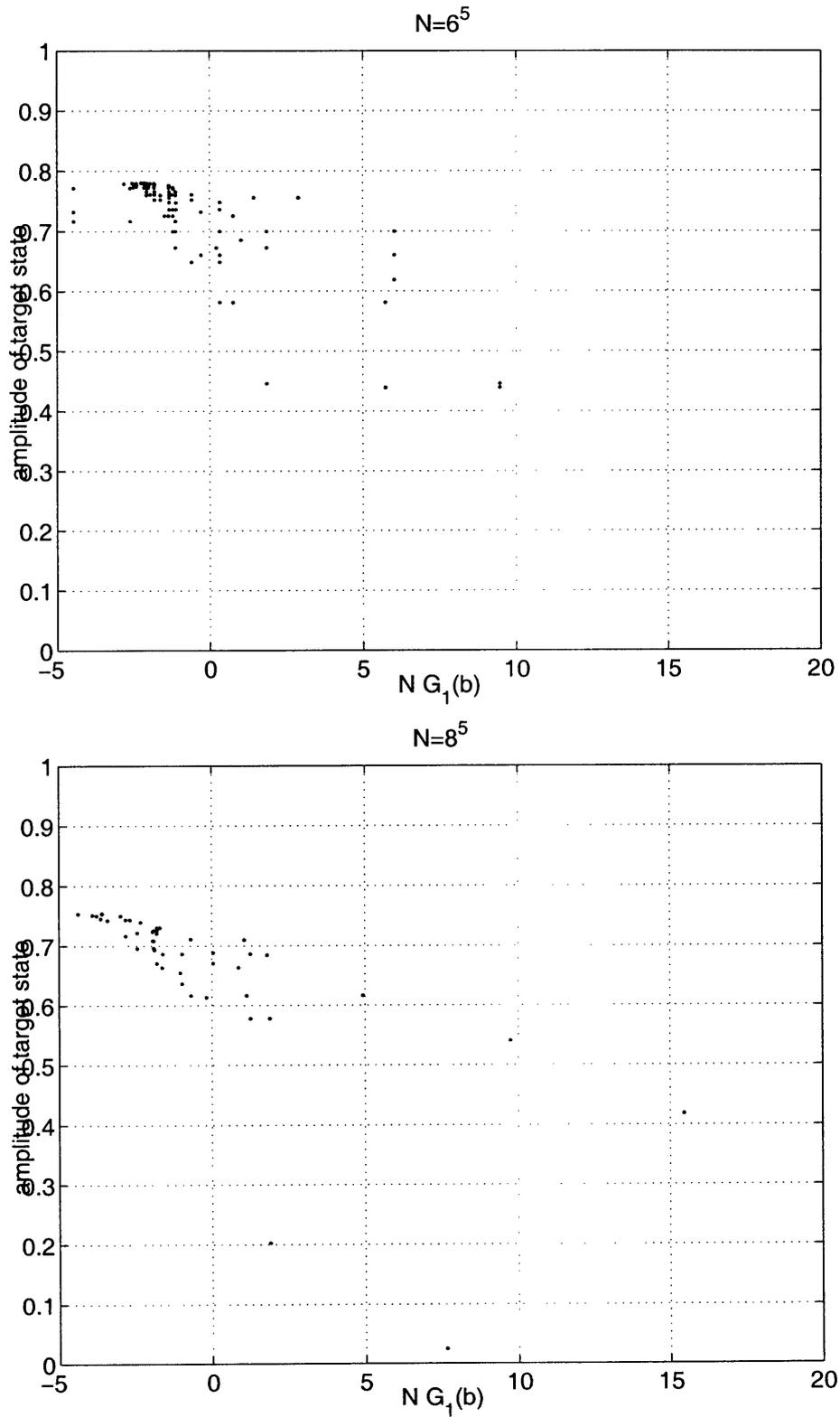


Figure 3-4: The probability of measuring the target state at time t_e for different starting states differentiated by their NG_1 value

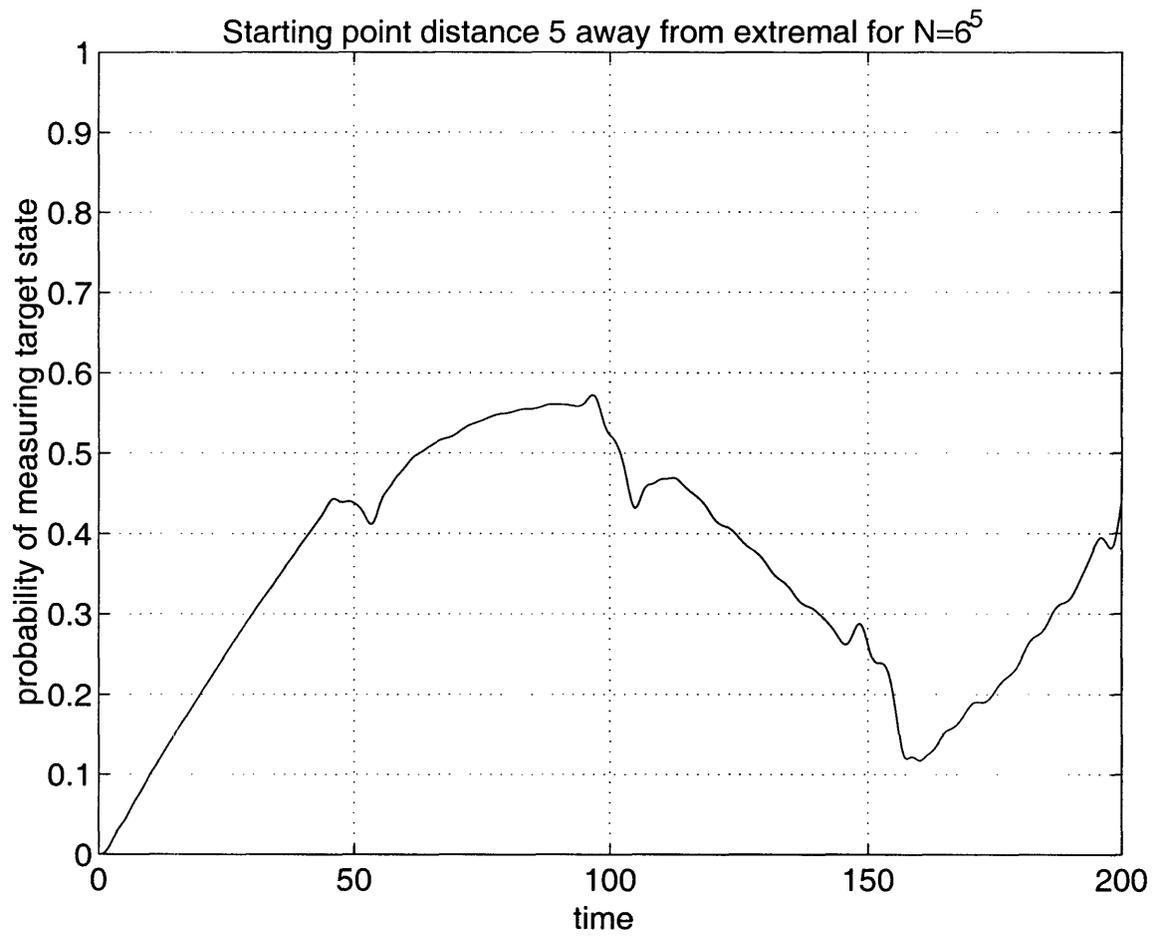


Figure 3-5: Probability of measuring the target state over time for a starting state far from the extremal vertex.

Chapter 4

Numerical analysis of algorithm for $d = 5$ and $d = 4$

4.1 The 5 dimensional lattice

In Figures 4-1 and 4-2, we can see the amplitude decay of the target state probability as a function of the distance of the starting vertex from the extremal vertex. Note that for odd $N^{1/d}$, this is a virtual extremal vertex because the coordinates $x_j = N^{1/d}/2$ do not correspond to an actual lattice site. Looking at the amplitude versus distance plots for different values of N , it does not appear that the general shape of the graphs depend on N . From the graphs, one can see that the amplitude does not seem to decay significantly until the distance is at least half the maximum distance possible from the extremal vertex, $D_e = \sqrt{dN^{1/d}/2}$.

The maximum distance possible from the extremal vertex is $\Theta(N^{1/d})$. There are $\Theta((N^{1/d})^d) = \Theta(N)$ lattice vertices contained in the d dimensional sphere of radius $\Theta(N^{1/d})$. Therefore, assuming similar behavior for large N , there is a constant probability of the algorithm picking a starting point where the target state amplitude will be high. If this is the case, then to get a probability of success arbitrarily close to 1, we only need to repeat the algorithm some constant number of times.

In Figure 4-3, we plot the value of t_e , the time of the first peak in target amplitude, for different values of \sqrt{N} . We determine each value of t_e manually, by visually

approximating the time of the first peak in the amplitude of the target state for each value of N . We found that $t_e = (2.437 \pm 0.005)\sqrt{N} + (0.4774 \pm 0.05)$ with an R^2 value of 0.9975. This suggests that the running time of our algorithm is $O(\sqrt{N})$; if so, we have the same running time in $d = 5$ as the Childs and Goldstone algorithm. Their algorithm has a running time of $1.1694\sqrt{N}$. Our algorithm is a factor of 2.085 slower; we expect this factor to become $\sqrt{2}$ in the high dimensional limit, as indicated in our analysis of the complete graph.

4.2 The 4 dimensional lattice

In Figures 4-4 and 4-5, we again plot the amplitude decay of the target state probability as a function of the distance of the starting vertex from the extremal vertex. Again, the target probability for different lattice starting sites does not seem to decay until the start vertex is some $\Theta(N^{1/d})$ distance from the extremal point. However, note that the points on the graphs of $N = 8^5$ and $N = 9^5$ are much more spread out for $d = 4$ than for $d = 5$. One can see that the points are significantly more dispersed for low distances in $d = 4$ than $d = 5$.

In the Childs and Goldstone algorithm, the success probability for $d = 4$ is $O(1/\log N)$ at time $O(\sqrt{N \log N})$, giving a total running time of $O(\sqrt{N \log^3 N})$ with classical repetition, or $O(\sqrt{N} \log N)$ using amplitude amplification [5]. The variation of amplitudes for at small distances for $d = 4$ may be a result of this $\log N$ factor in the success probability. We plot the average success probability of our algorithm for different values of N in Figure 4-6. We do not have enough data points to be able to conclusively state the N dependence of the average success probability. We also plot t_e as a function of N in Figure 4-7. This data obviously depicts a t_e with a $o(N)$ dependence. In Figure 4-8, we have a logarithmic plot of N and t_e . We find that the data fits $t_e = O(N^{0.53 \pm 0.01})$ with an R^2 value of 0.9986. Unfortunately we cannot determine if there is a log multiplicative factor; the data fits $O(\sqrt{N})$, $O(\sqrt{N \log N})$, and $O(\sqrt{N} \log N)$ equally well, with an R^2 value of 0.9980, 0.9988, and 0.9978, respectively. Therefore, we cannot conclusively state the running time of the algorithm

for $d = 4$, but it appears that t_e has a $O(\sqrt{N})$ polynomial dependence.

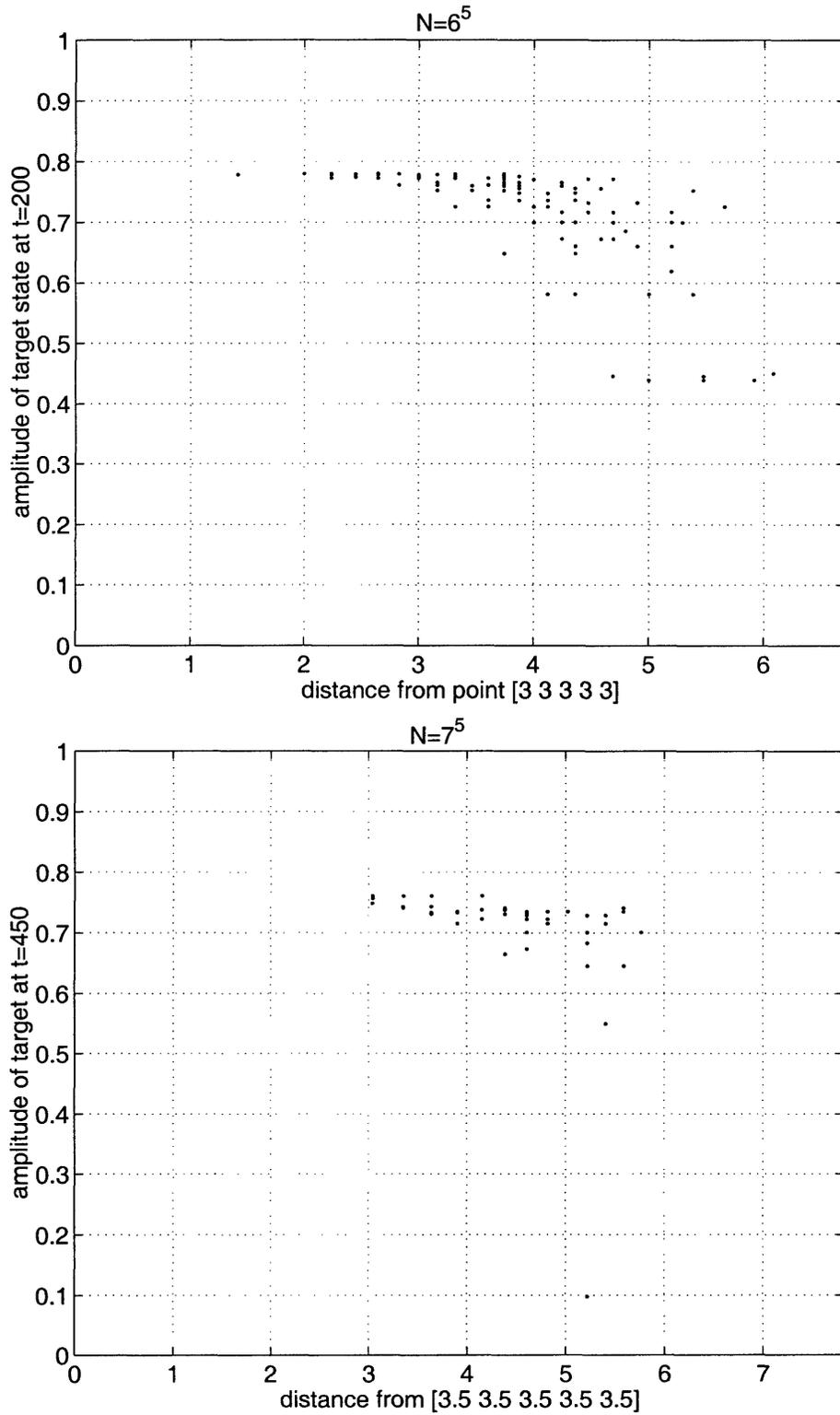


Figure 4-1: Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 6^5$ and $N = 7^5$

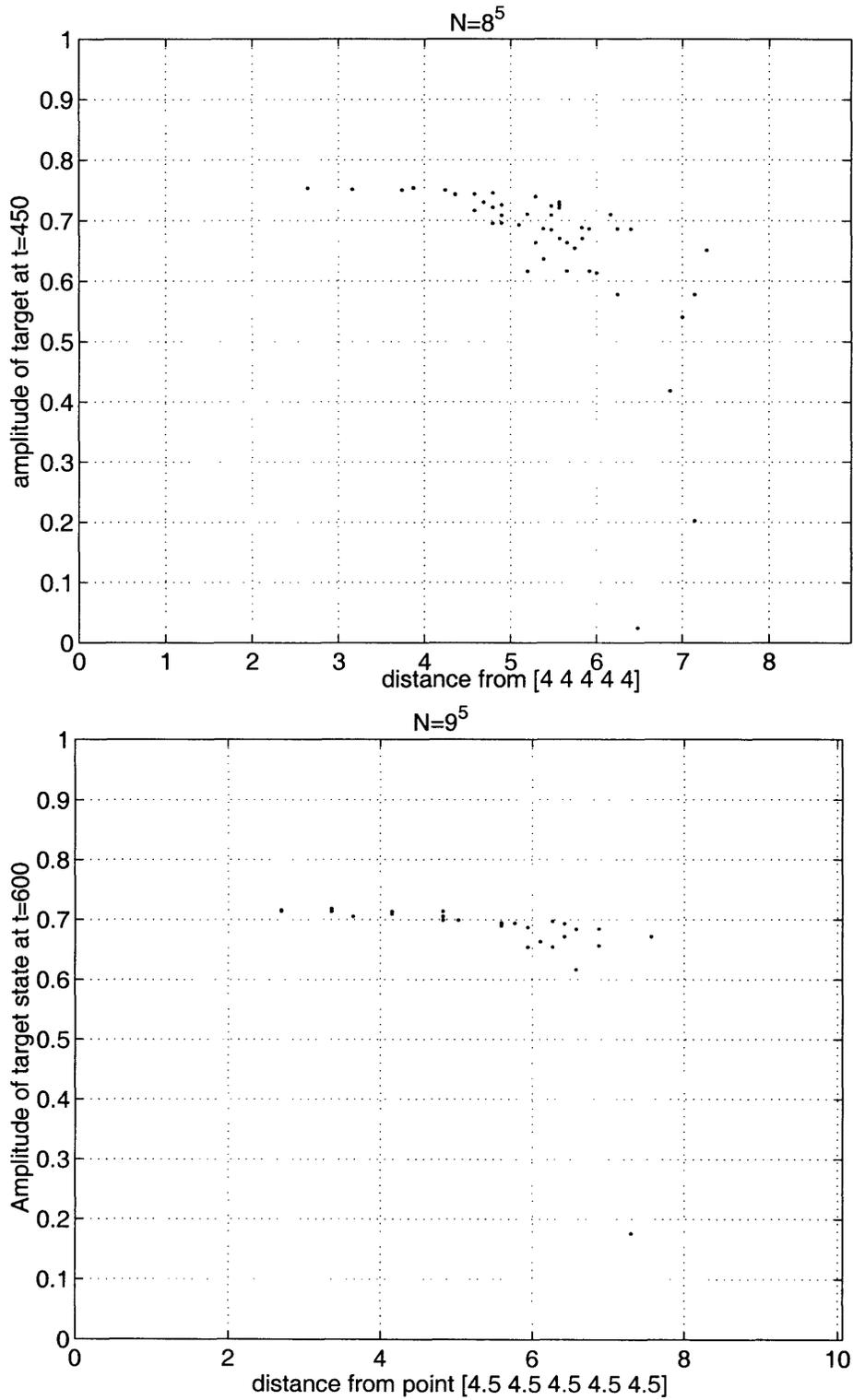


Figure 4-2: Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 8^5$ and $N = 9^5$

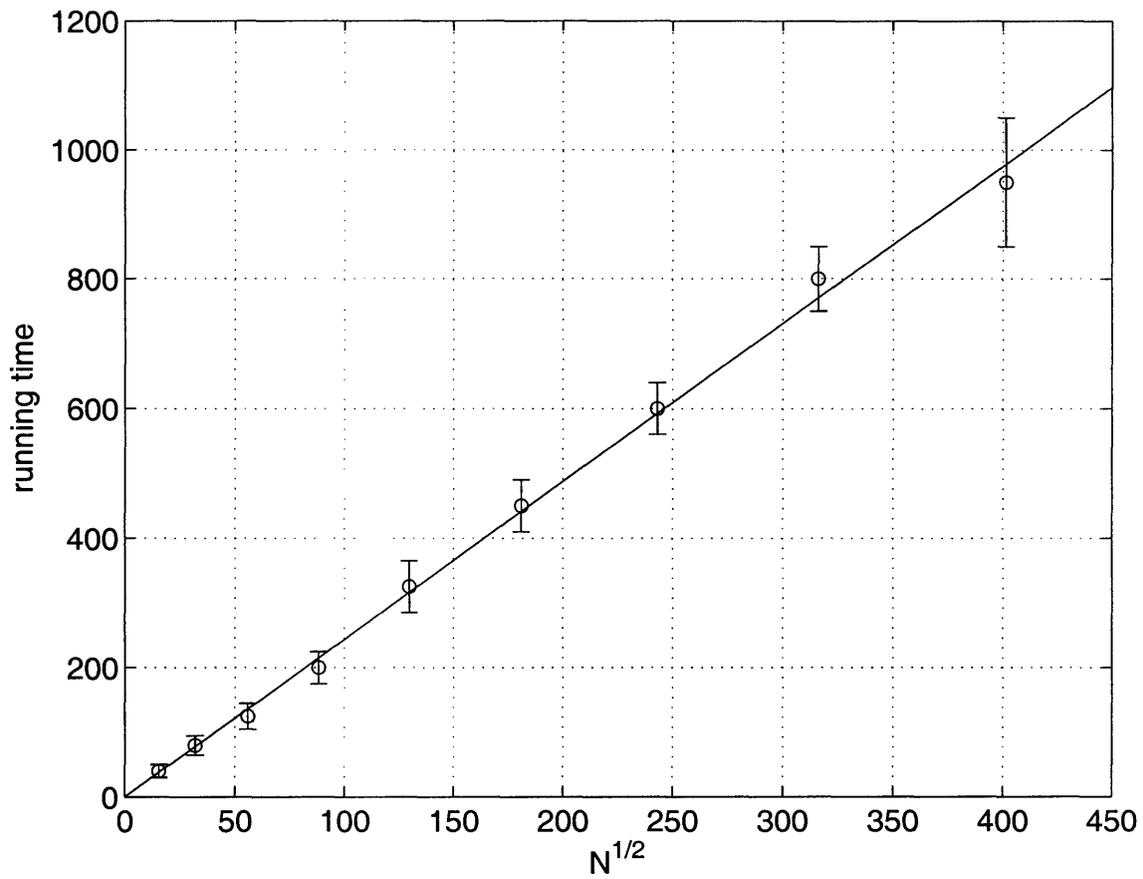


Figure 4-3: t_e for different values of \sqrt{N} for $d = 5$; $t_e = O(\sqrt{N})$

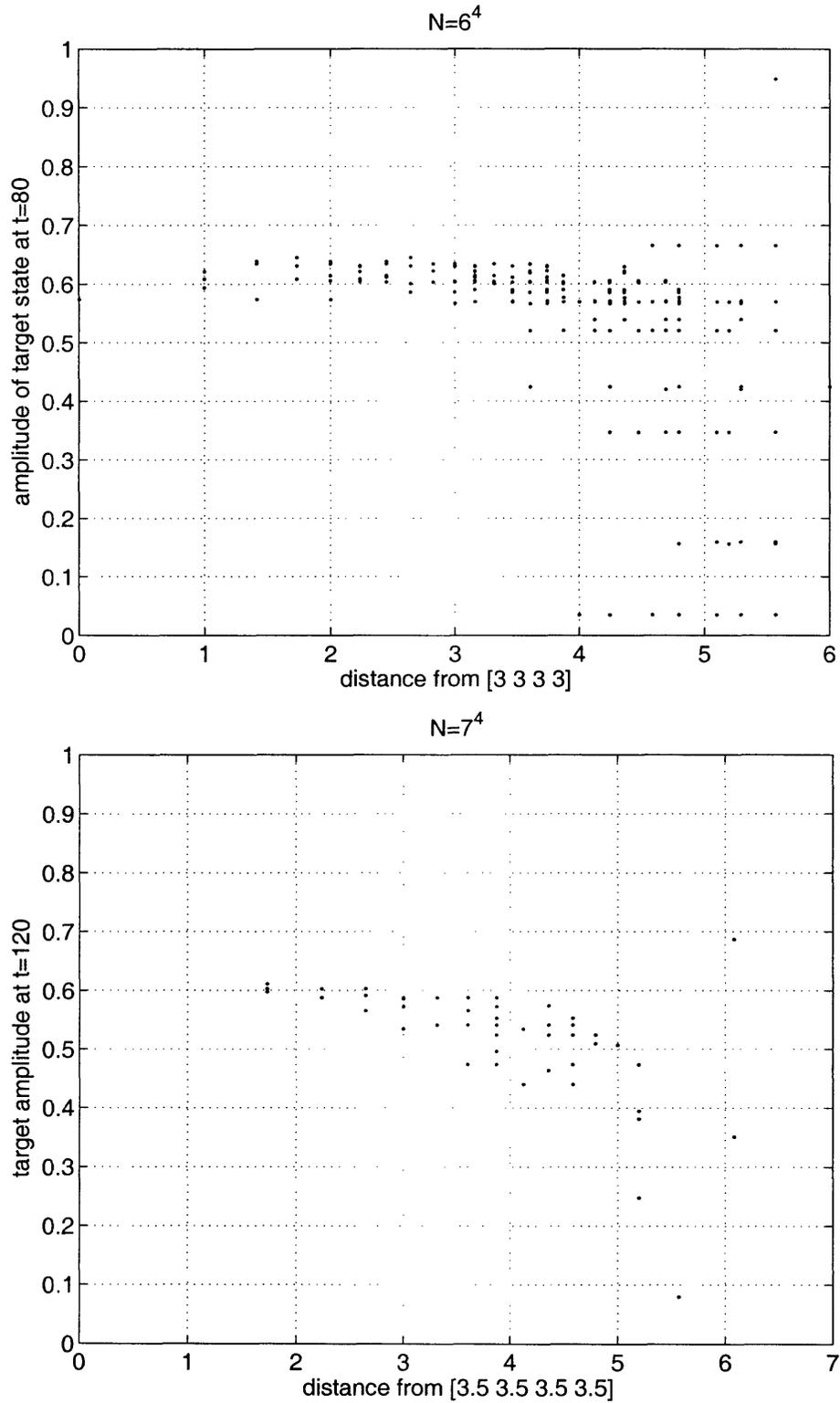


Figure 4-4: Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 6^4$ and $N = 7^4$

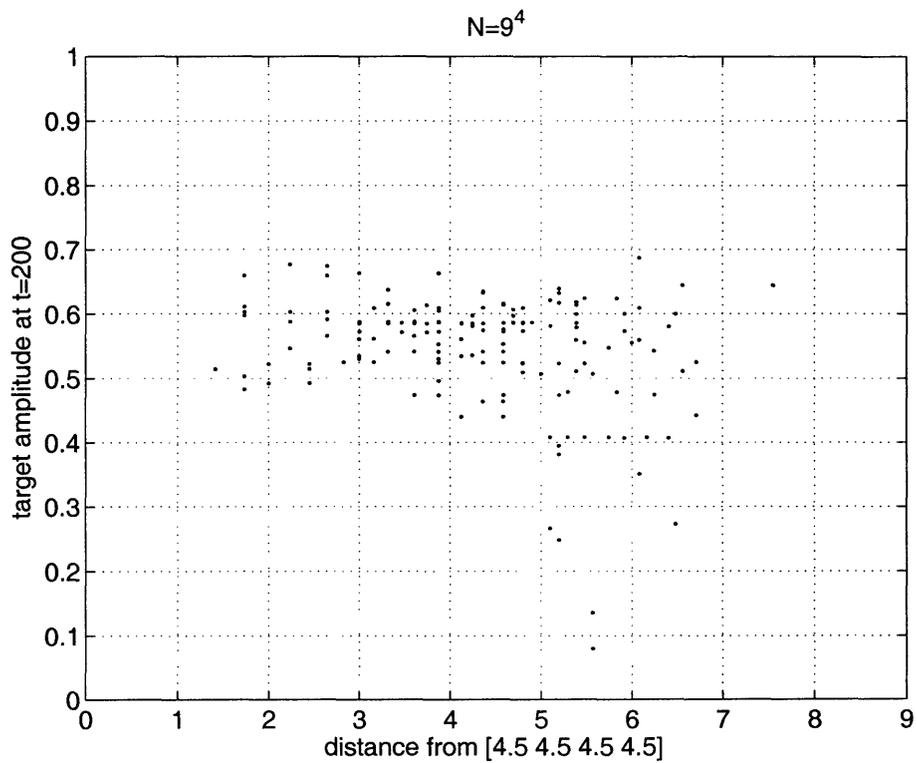
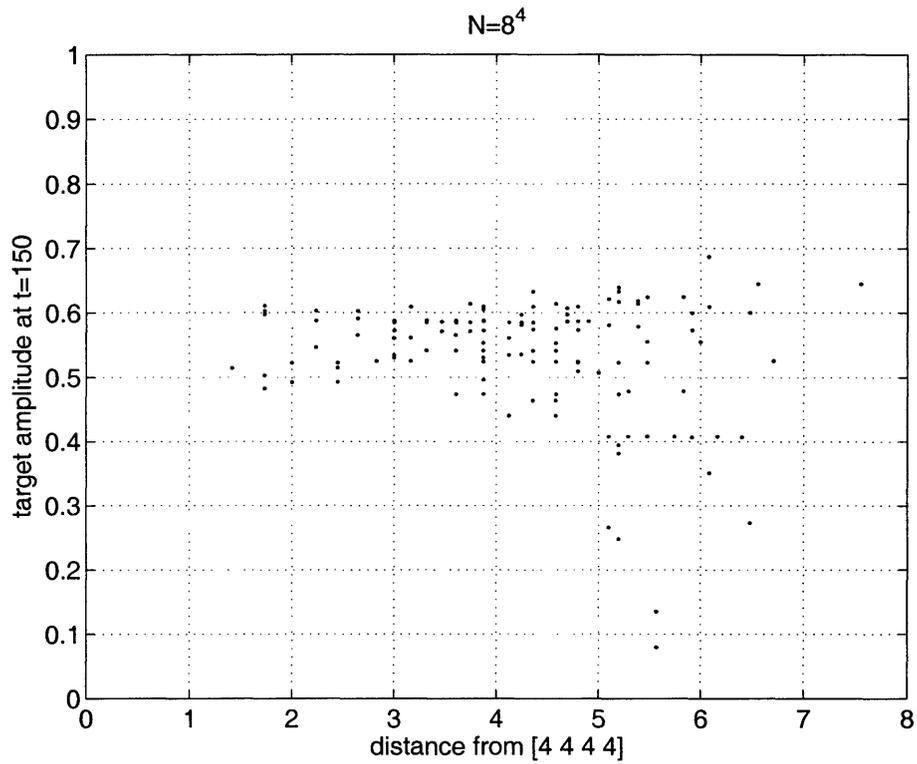


Figure 4-5: Dependence of amplitude of the target state at time t_e on distance of the starting vertex from extremal vertex for $N = 8^4$ and $N = 9^4$

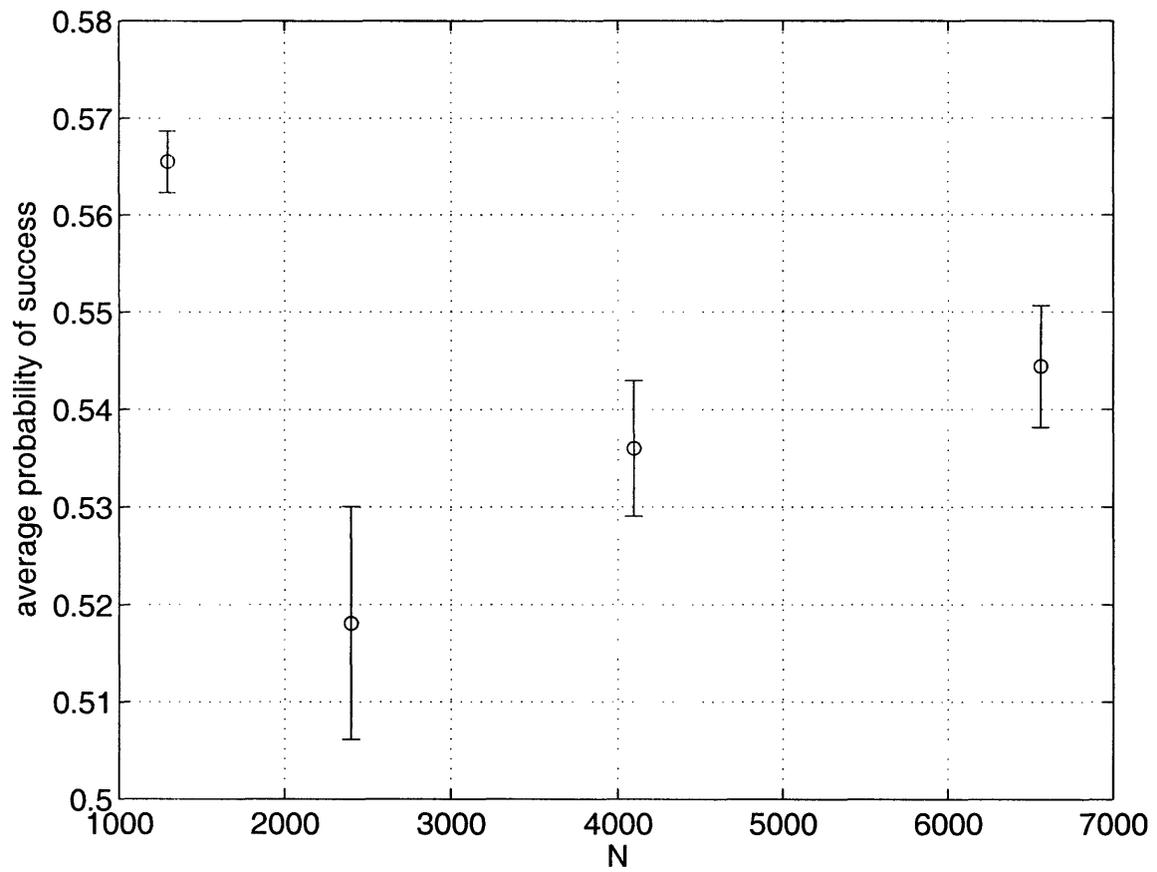


Figure 4-6: Average success probability of the algorithm for different values of N for $d = 4$

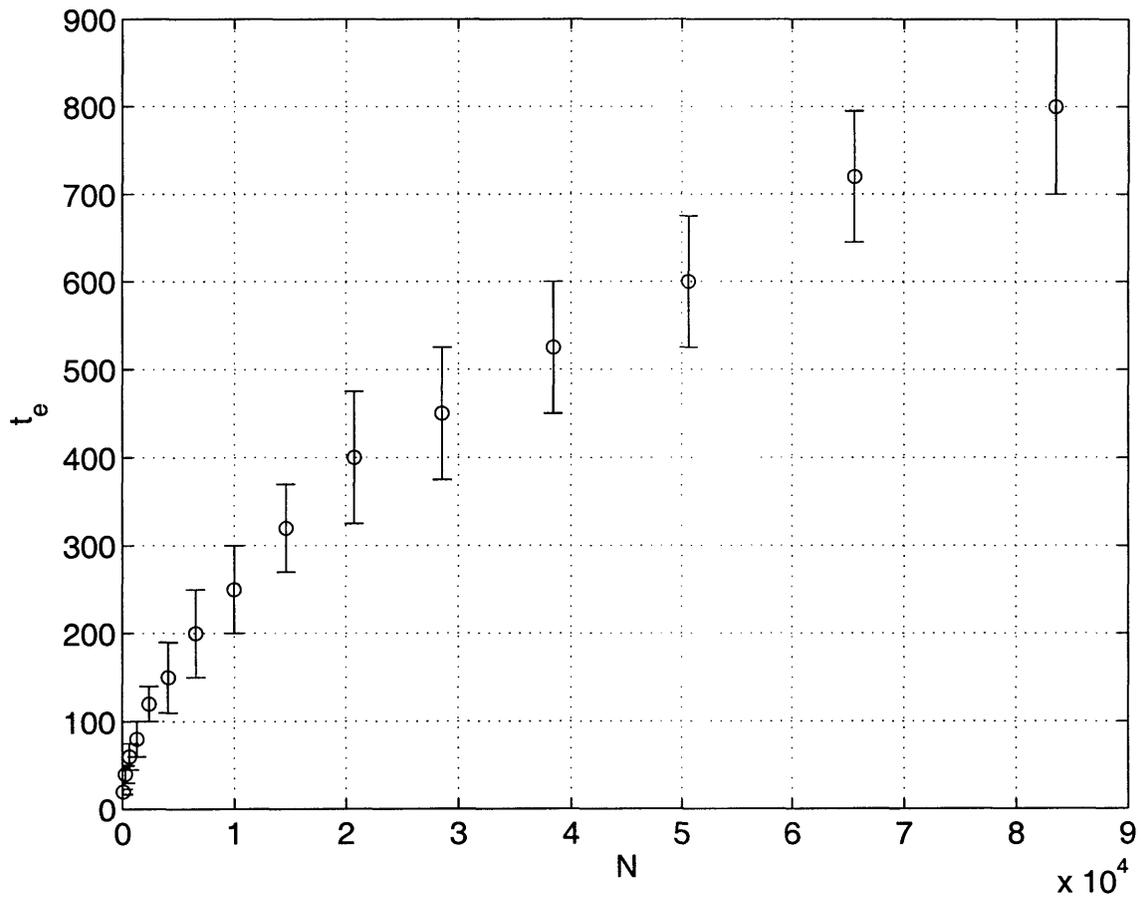


Figure 4-7: t_e for different values of N for $d = 4$

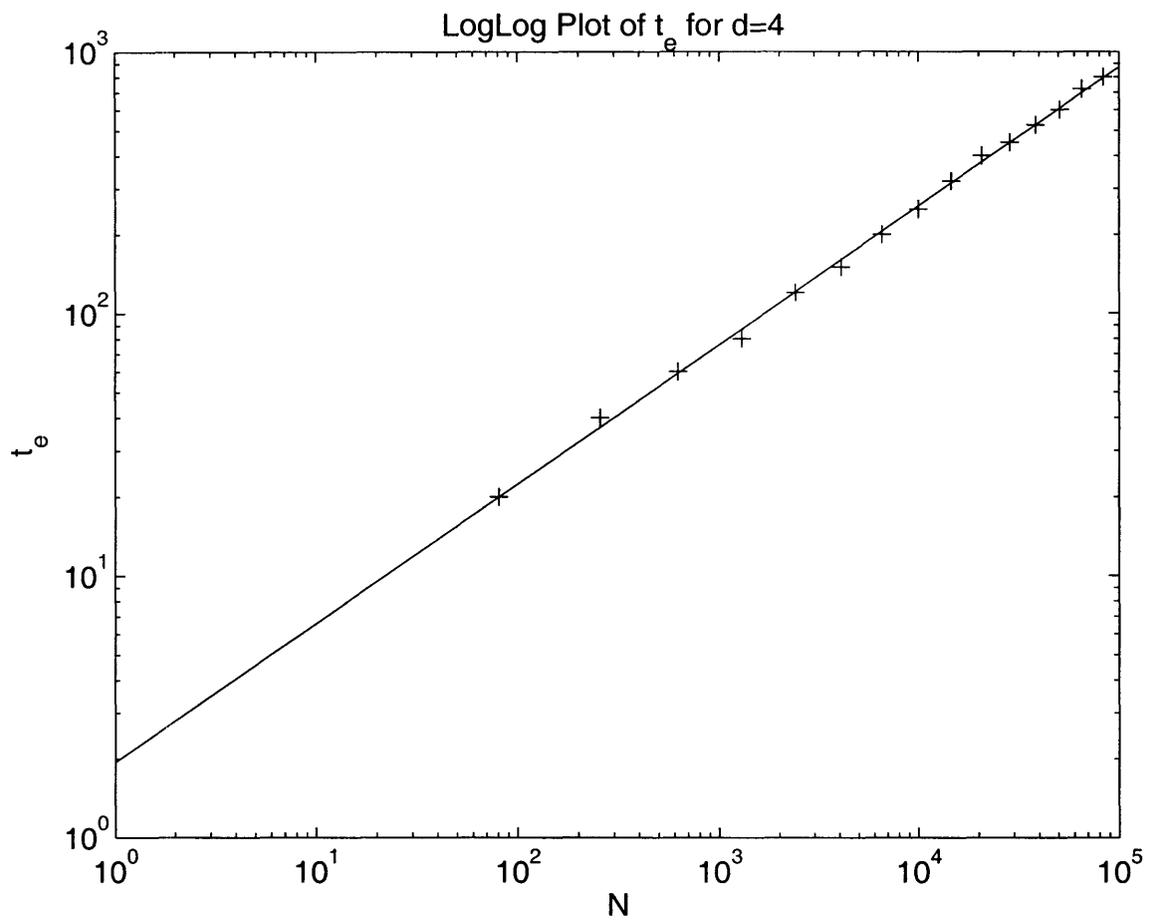


Figure 4-8: Log plot of t_e for different values of N for $d = 4$; we see that $t_e = O(N^{0.53 \pm 0.01})$

Chapter 5

Conclusions

In this thesis, we have presented a quantum walk algorithm for spatial search of a periodic lattice from a randomized local starting state. Our algorithm is a variation of the Childs and Goldstone algorithm for spatial search found in [9], which was a quantum walk algorithm from a uniformly delocalized initial state. We used the Hamiltonian

$$H = -\gamma L - |w\rangle\langle w| - |b\rangle\langle b|.$$

We began by looking at the highly symmetric case of the complete graph. The complete graph is the high d limit of the d dimensional periodic lattice. We reduced the problem to a three state system, and found that the system oscillated between the start state and the target state, using the linear superposition of all vertices as an intermediate state. These oscillations had a period of $O(\sqrt{N})$.

We then tried to reduce the behavior of our algorithm to that of the Childs and Goldstone algorithm by comparing the eigenvalue conditions of the Hamiltonians used in the two algorithms. We showed that for the case of where we began at the extremal vertex of the lattice, the eigenvalue condition for the spectrum of our Hamiltonian is close to their eigenvalue condition for the spectrum of their Hamiltonian. We then give some data that indicates that moving the starting state a distance $O(N^{1/d})$ from this extremal vertex does not dramatically change the eigenvalue condition, and that our algorithm looks to have a constant probability of behaving correctly. Finally, we

numerically analyze the behavior of our algorithm on 5 and 4 dimensional lattices. In the 5 dimensional case, the results are consistent with $O(\sqrt{N})$ running time. In the 4 dimensional case, we are not able to conclude the exact running time of our algorithm, but the data predicts a running time with a $O(\sqrt{N})$ polynomial dependence.

Bibliography

- [1] A. Aaronson and A. Ambainis. Quantum search of spatial regions. In *44th IEEE Symposium on Foundations of Computer Science*, pages 200–209, 2003.
- [2] A. Ambainis, J. Kempe, and A. Rivosh. Coins make quantum walks faster. quant-ph/0402107, 2004.
- [3] P. Benioff. Space searches with a quantum robot. *Quantum Computation and Information*, 305, 2002.
- [4] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(1510), 1997.
- [5] G. Brassard, P. Hoyer, m. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Information*, 2002.
- [6] A. Childs. Unpublished notes. 2003.
- [7] A. M. Childs. *Quantum Information Processing in Continuous Time*. PhD thesis, MIT, 2004.
- [8] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S Gutmann, and D. A. Spielman. Exponential algorithmic speedup by quantum walk. In *Proc 35th ACM Symposium on Theory of Computing*, page 59, 2003.
- [9] A. M. Childs and J. Goldstone. Spatial search by quantum walk. Submitted to Phys. Rev. A; quant-ph/0306054, 2003.

- [10] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser. Quantum computation by adiabatic evolution. 2002. quant-ph/0204013.
- [11] E. Farhi and S. Gutmann. Quantum computation and decision trees. *Phys. Rev. A*, 58(915), 1998.
- [12] E. H. Farhi and S. Gutmann. Analog analogue of a digital quantum computation. *Phys. Rev. A*, 75(2403), 1998.
- [13] L.K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(325), 1997.
- [14] A. Kitaev, A. Shen, and M. Vyalyi. *Classical and Quantum Computation*. American Mathematical Society, Providence, Rhode Island, 2000.
- [15] P. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.