Discrete-Time Quantum Walks with Lazy and Chaotic Coins

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Pranay Naredi

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May 2022

Supervisor: M. S. Santhanam

Certificate

This is to certify that this dissertation entitled Discrete-Time Quantum Walks with Lazy and Chaotic Coins towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Pranay Naredi at Indian Institute of Science Education and Research under the supervision of M. S. Santhanam, Professor, Department of Physics, during the academic year 2021-2022.

M. S. Santhanam

Committee:
M. S. Santhanam
Rejish Nath

To my family and friends...

Declaration

I hereby declare that the matter embodied in the report entitled Discrete-Time Quantum Walks with Lazy and Chaotic Coins are the results of the work carried out by me at the Department of Physics, Indian Institute of Science Education and Research, Pune, under the supervision of M. S. Santhanam and the same has not been submitted elsewhere for any other degree.

Pranay Naredi

Acknowledgments

I would first like to thank my supervisor, Professor M S Santhanam, for being an excellent guide and pushing me to sharpen my thinking and bring my work to a higher level. I am grateful to Bharathi Kanan for helping me with the project and for the productive conversations we had.

I acknowledge the INSPIRE-SHE scholarship that I have received from the Department of Science and Technology, Government of India.

I thank my family and friends for providing the help that I needed. Furthermore, I am very grateful to the IISER community for providing a wonderful research environment that has helped me grow as a researcher.

Abstract

We review and study quantum walks and their properties. We highlight the contrasts between quantum and classical walks by comparing the walker's probability distribution and hitting times on different types of graphs. We focus on discrete-time quantum walks. We see that the quantum walks do not always give an advantage over the classical random walks. As in the case of Lackadaisical quantum walks for a specific range of parameter values, the classical walker has lower hitting times as compared to the quantum walker.

We study the lackadaisical quantum walk search algorithm and its limitations in finding adjacent marked nodes on a graph. We develop a modified algorithm to find any type of marked node at the cost of increased algorithm running time. We carry out the search on a binary tree and a hypercube and build a quantum circuit to search for a marked node on the $3-$dimensional hypercube.

Finally, we study the quantum-chaotic walks by coupling a quantum kicked rotor as a coin to the quantum walker. The kicked rotor affects the walker's probability distribution, and it seems that the quantum walker gets localized with high probability for a short time interval. This behavior occurs for low $\hbar_s$ and kick strengths, $K \lesssim 2$ (for the kicking time period $T = 1$), accompanied by the rapid growth of von Neumann entropy which is used to measure coin-walker entanglement.

Contents

Chapter 1

Introduction

A random walk is a series of independent random processes in a mathematical space [1, 2, 3]. Random walks are used to model phenomena that occur in the real world and are intensively used in mathematics, biology, chemistry, and physics. In computer science, classical random walks are heavily used for developing algorithms, such as Pagerank [4, 5, 6], image segmentation [7], search algorithms, etc. Like classical random walks, quantum walks can be utilized to develop quantum algorithms [8, 9, 10, 11] that can be the basis for modern quantum computers. Quantum walks can be thought of as a quantum generalization of classical random walks [12, 13]. However, the walker for the quantum walk can be in a superposition state, giving rise to interference effects. The quantum walker evolves by exploring multiple paths simultaneously, making the variance grow quadratically faster than a classical random walk. Quantum walks are described by two different models: continuous-time quantum walk (CTQW) and discrete-time quantum walk (DTQW). In [14, 15], quantum walks (continuous or discrete) are shown to be universal for quantum computation. Therefore, studying quantum walks can deepen our understanding of quantum computation.

1.1 Continuous-time quantum walks (CTQW)

The CTQW is the quantization of the continuous-time classical random walk [16, 13, 17]. We require only a position space to define continuous-time classical random walk on a graph $G = (V, E)$. The position space, denoted by $\mathcal{H}_P$, is spanned by the vertex set V. E is the set of edges of the graph. The information about the connections of the vertices by the edges is encoded in the adjacency matrix:

$$A_{i,j} = \begin{cases} 1, & i,j \in E \\ 0, & i,j \notin E \end{cases} \tag{1.1}$$

To describe the random walk on G, we need to define the infinitesimal generator matrix that governs the evolution of the probabilities. The matrix is given by

$$H_{i,j} = \begin{cases} -\gamma, & i \neq j,\ i,j \in E \\ 0, & i \neq j,\ i,j \notin E \\ d_i\gamma, & i = j \end{cases} \tag{1.2}$$

where γ is the probability of transitions between the connected vertices per unit time and is time-independent. d_i is the degree of the vertex i.

Let $p_i(t)$ denote the walker's probability to be on the vertex i at time t, then the differential equation that dictates the transitions on G is

$$\frac{d}{dt}p_i(t) = -\sum_{j\in V} H_{i,j}p_j(t) \tag{1.3}$$

whose solution is of the form

$$\vec{p}(t) = e^{-Ht}\vec{p}(0) \tag{1.4}$$

In the quantum case, the vertices are represented as the orthogonal basis states $\{|j\rangle\}$ of the position Hilbert space $\mathcal{H}_P$. The generator matrix becomes the Hamiltonian, and with some modifications Eq.1.3 become

$$\iota\frac{d}{dt}a_i(t) = \sum_{j\in V} H_{i,j}a_i(t) \tag{1.5}$$

where the probabilities are replaced by quantum amplitudes $a_i(t) = \langle j|\psi(t)\rangle$. Eq. 1.5 is the Schrödinger equation

$$\iota\frac{d}{dt}\,|\psi(t)\rangle = H\,|\psi(t)\rangle \tag{1.6}$$

and its solution is

$$|\psi(t)\rangle = e^{-\iota Ht}\,|\psi(0)\rangle \tag{1.7}$$

where, $U(t) = e^{-\iota Ht}$ is the unitary evolution operator of the walk that transforms the probability amplitude to the connected vertices through the edges.

1.2 Discrete-time quantum walks (DTQW)

A discrete-time classical random walk is defined using a coin toss followed by the walker's movement in the position space, and this movement depends on the coin toss result. In the quantum case [17], the application of a coin operator, defined on the "coin" Hilbert space, replaces the coin toss process. The structure of DTQW includes a "coin" Hilbert space $\mathcal{H}_C$ and the position Hilbert space $\mathcal{H}_P$ such that the state of the total system is in the space $\mathcal{H} = \mathcal{H}_\mathcal{C} \otimes \mathcal{H}_\mathcal{P}$. The walker moves in the position-space according to the instructions given by states from the coin space, i.e., a walk operator consists of a coin operator that performs rotation in the coin space followed by a shift operator that moves the walker.

Generally, $\mathcal{H}_C$ is spanned by the directions a walker can move from a vertex, namely graph's edges. And similar to CTQW, $\mathcal{H}_P$ is spanned by the orthogonal basis states representing the graph's vertices. The unitary evolution operator for a DTQW has the form:

$$U = S(C \otimes I) \tag{1.8}$$

where, C and S are the coin and the shift operators, respectively. Evolution of an initial state $|\Psi(0)\rangle$ of the quantum walk for time t results in a state $|\Psi(t)\rangle = U^t |\Psi(0)\rangle$.

Since quantum walks evolve in superposition, there is nothing random about them. The randomness comes from the measurement of the walker's position, the superposition collapses randomly in one of the measurement basis states. As the quantum walk is a generalization of the classical random walk, it reduces to the classical random walk under certain conditions. These conditions [18] include measurement of the walker or coin at every time step, the use of multiple independent coins, or decoherence in the system.

An example of a DTQW on a linear lattice can be described using the Hilbert spaces $\mathcal{H}_C$ spanned by $\{|\uparrow\rangle, |\downarrow\rangle\}$ and $\mathcal{H}_P$ spanned by $\{|i\rangle : i = -N, ... -1, 0, 1...N\}$. Let the coin operator be a Hadamard coin

$$C_H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{1.9}$$

and the shift operator can be written as

$$S = |\uparrow\rangle \langle\uparrow| \otimes \sum_i |i+1\rangle \langle i| + |\downarrow\rangle \langle\downarrow| \otimes \sum_i |i-1\rangle \langle i| \tag{1.10}$$

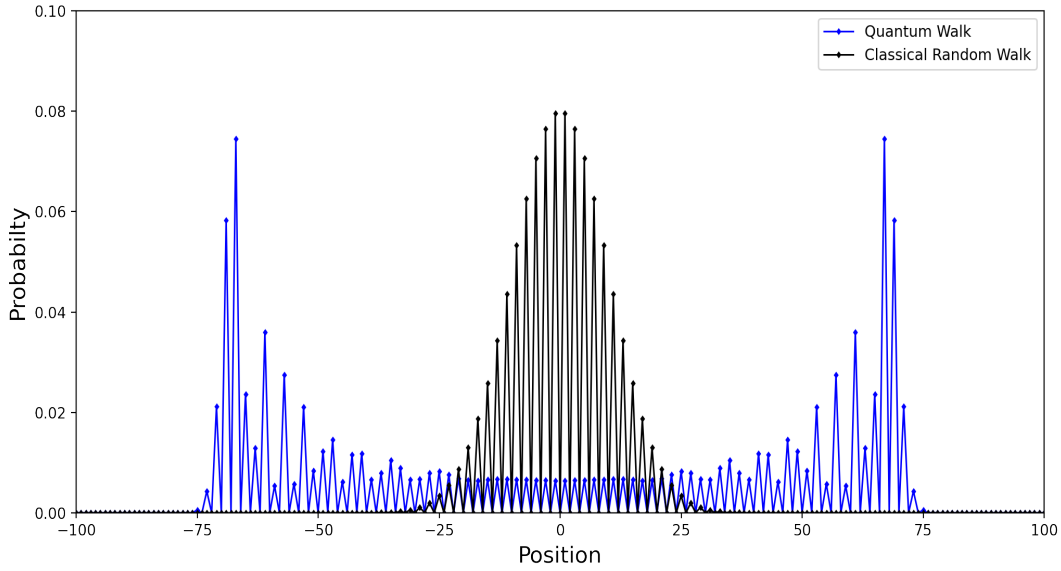


Figure 1.1: Probability distribution of a Hadamard walk (Blue) compared with classical-random walk (black) at time $t = 100$.

This walk is knows as the Hadamard walk. Quantum and classical walker's probability distributions are compared in in Fig.1.1, the initial state for the quantum walk is $\frac{1}{\sqrt{2}}(|\uparrow\rangle + \iota\,|\downarrow\rangle) \otimes |0\rangle$. The Hadamard coin treats $|\uparrow\rangle$ and $|\downarrow\rangle$ differently [17]. Therefore, for the initial state $\frac{1}{\sqrt{2}}(|\uparrow\rangle + |\downarrow\rangle) \otimes |0\rangle$ the final walker probability distribution is asymmetric because of interference effects.

1.2.1 Initial state dependence

The initial state of the coin affects the evolution of the walker's probability distribution. Let us say, in the above example the coin's initial state is $|\uparrow\rangle$ ($|\downarrow\rangle$) then the walker's probability distribution drifts left (right). Few steps of the quantum walk's evolution to demonstrate this drifting for different initial states is carried out. Evolving the initial state $|\psi_1\rangle = |\uparrow\rangle \otimes |0\rangle$:

$$\begin{aligned} U^3 |\psi_1\rangle &= U^2 \frac{1}{\sqrt{2}} [|\uparrow\rangle \otimes |1\rangle + |\downarrow\rangle \otimes |-1\rangle] \\ &= U^1 \frac{1}{2} [|\uparrow\rangle \otimes |2\rangle + (|\uparrow\rangle + |\downarrow\rangle) \otimes |0\rangle - |\downarrow\rangle \otimes |-2\rangle] \\ &= \frac{1}{2\sqrt{2}} [|\uparrow\rangle \otimes |3\rangle + (|\downarrow\rangle + 2\,|\uparrow\rangle) \otimes |1\rangle - |\uparrow\rangle \otimes |-1\rangle + |\downarrow\rangle \otimes |-3\rangle] \end{aligned}$$

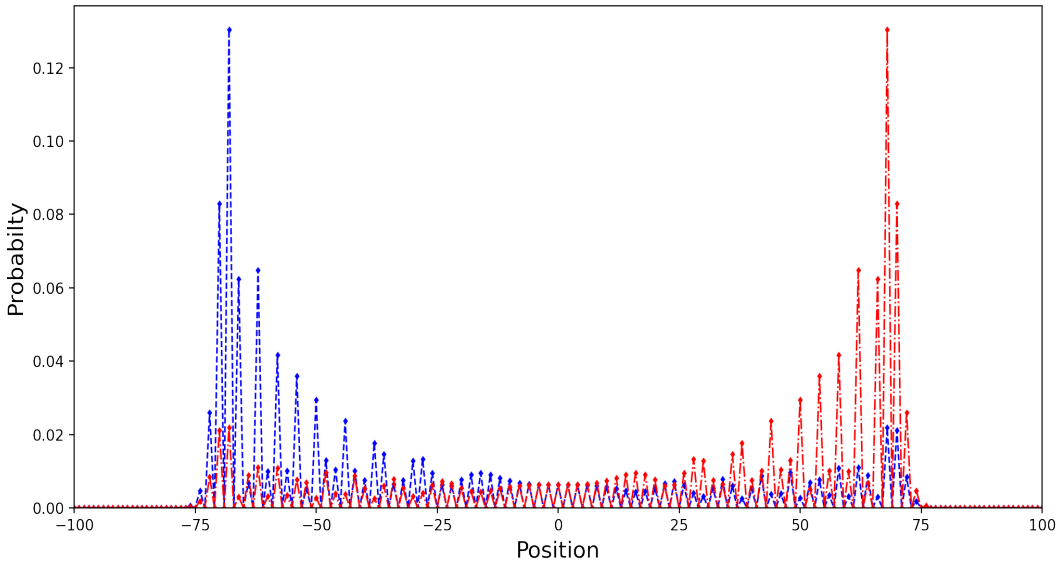


Figure 1.2: Probability distribution of a Hadamard walk at time $t = 100$ for the initial states $|\psi_1\rangle$ (red) and $|\psi_2\rangle$ (blue).

$|\psi_2\rangle = |\downarrow\rangle \otimes |0\rangle$:

$$\begin{aligned} U^3 |\psi_2\rangle &= U^2 \frac{1}{\sqrt{2}} [|\uparrow\rangle \otimes |1\rangle - |\downarrow\rangle \otimes |-1\rangle] \\ &= U^1 \frac{1}{2} [|\uparrow\rangle \otimes |2\rangle + (-|\uparrow\rangle + |\downarrow\rangle) \otimes |0\rangle + |\downarrow\rangle \otimes |-2\rangle] \\ &= \frac{1}{2\sqrt{2}} [|\uparrow\rangle \otimes |3\rangle + |\downarrow\rangle \otimes |1\rangle - (2|\downarrow\rangle - |\uparrow\rangle) \otimes |-1\rangle - |\downarrow\rangle \otimes |-3\rangle] \end{aligned}$$

For $|\psi_1\rangle$ ($|\psi_2\rangle$) the distribution drifts right (left) from the third step of the quantum evolution due to the constructive interference for positive (negative) and destructive for negative (positive) position states as shown in Fig.1.2.

For walker probability distribution to drift depending on the coin's initial is a more general property and is even seen for higher-dimensional graphs. The initial state of the coin determines the direction in which the evolution would have more constructive interference as compared to other directions related to other coin states.

1.2.2 Coins for DTQW

Various coins can be used in discrete-time quantum walks, resulting in different walker distributions. The commonly used coins are the Hadamard, Grover, and DFT coins. An example of a Quantum walk using the Hadamard coin is described above. A more general Hadamard coin for higher dimensional quantum walks is $H_d = C_H \otimes C_H \otimes ... \otimes C_H$.

A Grover coin is the Grover diffusion operator. Grover coin can be used to make the walker move permutationally symmetric along the edges at a vertex of a graph. Grover coin is a very useful coin, and there are quantum walk search algorithms that require this coin to find a marked vertex on a graph. A $d-$dimensional Grover coin is

$$G = \begin{pmatrix} \frac{2}{d} - 1 & \frac{2}{d} & ... & \frac{2}{d} \\ \frac{2}{d} & \frac{2}{d} - 1 & ... & \frac{2}{d} \\ & & ... & \\ \frac{2}{d} & \frac{2}{d} & ... & \frac{2}{d} - 1 \end{pmatrix}$$

Using a DFT (Discrete Fourier Transform) coin, we can have the walker move along the edges at a vertex with equal probabilities. The matrix form of the d-dimensional DFT coin is

$$DFT = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & ... & 1 \\ 1 & \omega & \omega^2 & ... & \omega^d \\ & & & ... & \\ 1 & \omega^{(d-1)} & \omega^{2(d-1)} & ... & \omega^{(d-1)(d-1)} \end{pmatrix}$$

where, $\omega = \exp(2\pi\iota/d)$ is the $d-$th root of unity.

There are many other types of coins that impart specific properties on the quantum walker. In recent studies, chaotic systems like the quantum baker map [19] and Harper map [18] have been used as coins for a quantum walker on a line giving rise to interesting dynamics of the walk. For the walker, these coupled systems can behave as the environment. We construct a walk using the quantum kicked rotor as the coin in chapter 5.

1.2.3 Circuit implementation of DTQW

To implement a quantum walk on a circuit, we require a coin and a walker (position) register. The DTQW is an iterative process and it involves he alternative application of the coin and

the shift operators (Eq.1.8). The circuit implementation of a single step of Hadamard walk (with periodic boundary condition) is shown in Fig.1.3 with an initial state $|0\rangle \otimes |000\rangle$ (or, $|\uparrow\rangle \otimes |000\rangle$). S_+ increments the position state by 1 if the coin state is $|0\rangle$ and S_- decrements the position state for coin state $|1\rangle$. For instance, $S_+ |0\rangle \otimes |000\rangle = |0\rangle \otimes |001\rangle$ and $S_- |1\rangle \otimes |000\rangle = |1\rangle \otimes |111\rangle$.

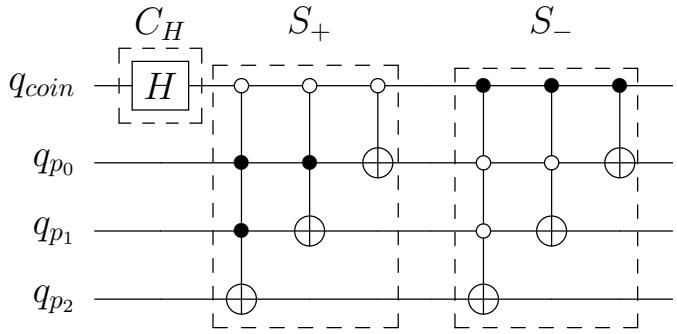


Figure 1.3: One step circuit implementation of a Hadamard walk.

Other walks with different coins and shift operators can be implemented in a similar way. In chapter 4, circuit implementation of a quantum walk on a hypercube is shown.

Chapter 2

Lackadaisical Quantum Walks

Consider a graph with N vertices on which we want to perform a quantum walk, and let the maximum degree (number of edges) for any vertex be d. The walker at a vertex on the graph can move along the edges at that vertex. The Hilbert space $\mathcal{H}_P$ associated with the walker's position would be N-dimensional, and the coin Hilbert space $\mathcal{H}_C$ would be d-dimensional. We represent each vertex on the graph and its edges with a basis states in $\mathcal{H}_P$ and $\mathcal{H}_C$, respectively. For vertices with a degree less than d, we can add self-loops (edge connecting the vertex to itself) at that vertex so that the degree of every vertex becomes d.

In order to describe the Lackadaisical Quantum Walk (LQW) [20, 21], quantum analog of lazy random walk, we increase the dimension of $\mathcal{H}_C$ to add a self-loop with a weight l at each vertex so that $\mathcal{H}_C$ is now a $d+1$-dimensional Hilbert space. The resulting Hilbert space of LQW is $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P = \mathbb{C}^{d+1} \otimes \mathbb{C}^N$.

Given the Hilbert space, we can construct the walk operator $\hat{U}$ by defining a coin flip operator $\hat{C}$ that performs a rotation in the "coin"-space and a shift operator $\hat{S}$ that evolves the walker in the position space. The walk operator has the form $U = \hat{S}.(\hat{C} \otimes I)$, also $\hat{C}$ is given as

$$\hat{C} = 2\left|s_c\right\rangle \left\langle s_c\right| - I_{d+1} \tag{2.1}$$

where, I_{d+1} is $(d+1)-$dimensional identity matrix and

$$\left|s_c\right\rangle = \frac{1}{\sqrt{d+l}} \left(\sum_{i=0}^{d-1} \left|i\right\rangle + \sqrt{l}\left|\circlearrowleft\right\rangle \right) \tag{2.2}$$

the coin subspace is spanned by the basis states $\{|i\rangle : i = 0, 1, ...d-1\}$ and $|\circlearrowleft\rangle$ which represents a self-loop. The initial coin state is $|s_c\rangle$, an eigenstate of the coin operator with eigenvalue 1. The shift operator depends on the graph of the walk.

The walker starts from the vertex denoted by the basis state $|0\rangle$ of the position space, the initial state of the quantum walk is $|\Psi(0)\rangle = |s_c\rangle \otimes |0\rangle$. The state at time t is

$$|\Psi(t)\rangle = \hat{U}^t(|s_c\rangle \otimes |0\rangle) \tag{2.3}$$

Therefore, the probability of finding the walker at a vertex $|n\rangle \in \mathcal{H}_{\mathcal{P}}$ can be calculated using the reduced density matrix of the walker,

$$\rho_W(t) = Tr_C(|\Psi(t)\rangle \langle\Psi(t)|) \tag{2.4}$$

as

$$P_n(t) = \langle n| \rho_W(t) |n\rangle \tag{2.5}$$

which can also be expressed as

$$P_n(t) = \langle n| \left(\sum_{i=0}^{d} \langle i|\Psi(t)\rangle \langle\Psi(t)|i\rangle \right) |n\rangle \tag{2.6}$$

where the sum is over the basis states of the $d+1-$dimensional coin subspace.

Here, we construct quantum walk on different types of graphs— line, hypercube and binary tree.

2.1 Undirected walk on a line

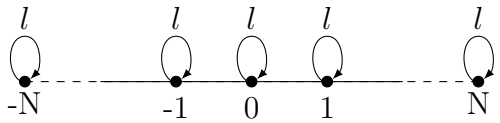


Figure 2.1: Walk on an line with a self-loop on each vertex having weight l

We first consider a quantum walk on a line where the walker starts at the origin and moves in both directions with the nearest-neighbor hopping [Fig.2.1]. The basis states of $N = 2n+1$-dimensional position Hilbert space represent N vertices $\{0, \pm 1, \ldots, \pm n\}$ of the graph. The

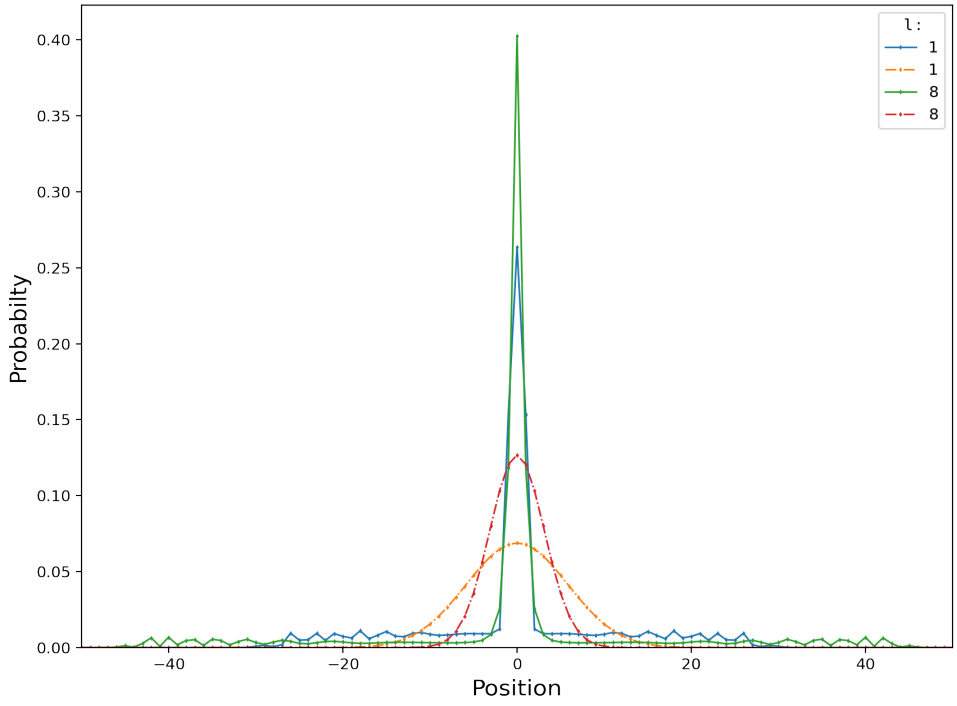


Figure 2.2: Comparison of quantum (solid line) and classical (dashed line) walker's probability distribution for $l = 1, 8$ at $t = 50$.

coin-space is a 3-dimensional Hilbert space spanned by $\{|\leftarrow\rangle, |\rightarrow\rangle, |\circlearrowleft\rangle\}$, the coin has the matrix representation

$$\hat{C} = \frac{2}{2+l}\begin{pmatrix} 1 & 1 & \sqrt{l} \\ 1 & 1 & \sqrt{l} \\ \sqrt{l} & \sqrt{l} & l \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.7}$$

and the shift operator for the walk is

$$\hat{S} = \sum_{x=-n}^{n} |\leftarrow\rangle\langle\leftarrow| \otimes |x-1\rangle\langle x| + |\rightarrow\rangle\langle\rightarrow| \otimes |x+1\rangle\langle x| + |\circlearrowleft\rangle\langle\circlearrowleft| \otimes |x\rangle\langle x| \tag{2.8}$$

The origin is denoted by the state $|0\rangle$, and the initial state of the system is given as $|s_c\rangle \otimes |0\rangle$. For $l = 0$, $\hat{C}$ is the 2-dimensional Pauli-X matrix, so the evolution of the initial state is the

alternate application of +1 and -1 shift resulting in a trivial walk. $l > 0$ leads to a non-trivial coin operator. Fig.2.3 shows the probability distribution for different self-loop weights at $t = 50$ steps. The inset shows that as we increase the self-loop weight the probability distribution spreads.

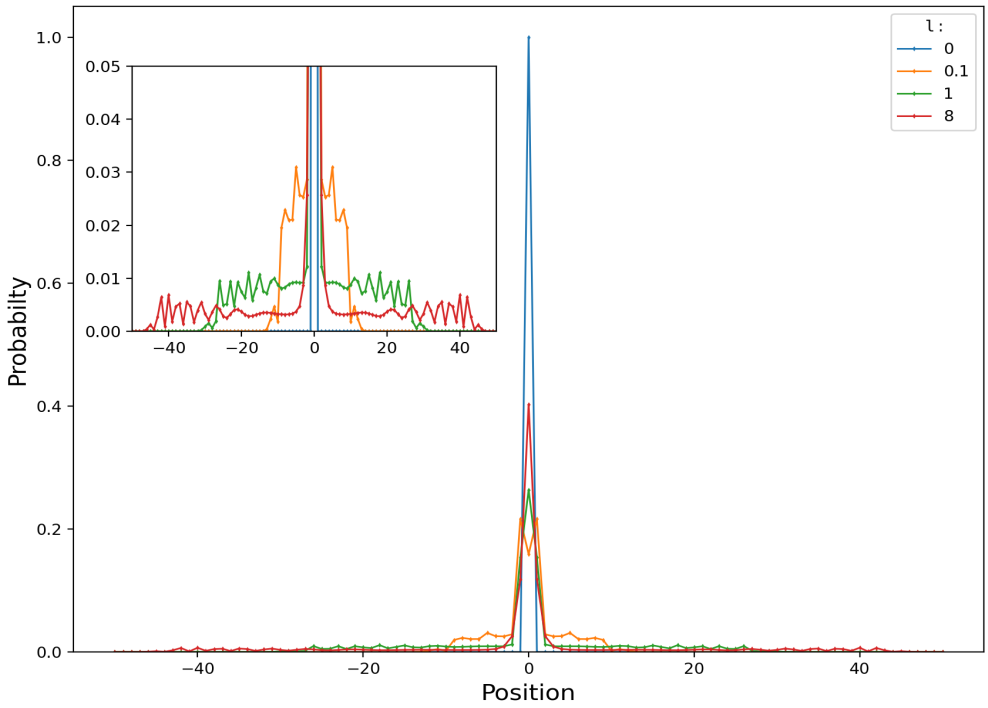


Figure 2.3: Walker distribution of LQW on a line with $n = 50$ at $t = 50$ steps for different self-loop weights.

We also plot the distribution for the corresponding classical random walk. This is done by evolving the initial state of the classical walker at the origin using the transfer matrix for time t. The classical walker's probability distribution is a gaussian, and the width of the distribution decreases as we increase the self-loop weight, unlike the quantum case. Fig.2.2 and 2.4 shows a comparison between the classical and the quantum walker's distribution. The variance in quantum case increases as we increase l up to a certain value of l giving high variances as compared to the classical case, which only decreases as l increases.

Undirected LQW on a line are intensively studied in [12]. In this section, we looked at the undirected walk on a line to give an idea of the behavior of lackadaisical and the regular quantum walks as well as the corresponding classical random walk.

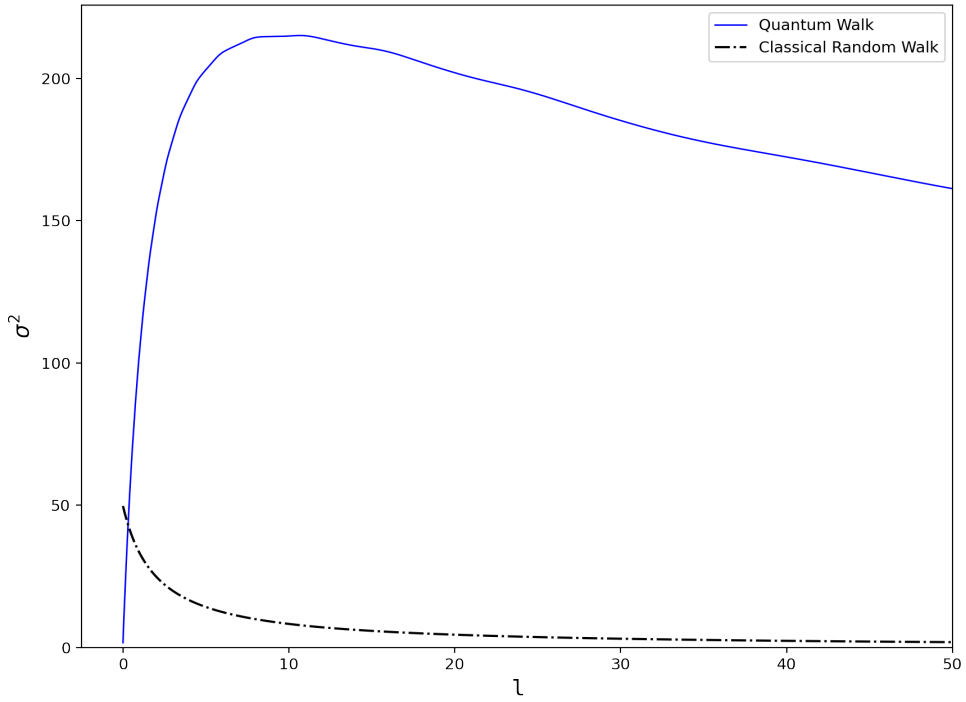


Figure 2.4: Variance of the probability distribution of quantum and classical walks at $t = 50$

2.2 Directed walk on a line

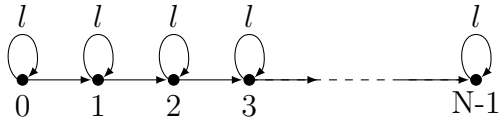


Figure 2.5: Directed walk on a line with a self-loop on each vertex having weight l

Next, consider a quantum walk on a line as in Fig.2.5 with periodic boundary conditions where the walker can only move in a single direction. The line has N vertices $\{0, 1, \ldots, N-1\}$ and we represent them with the basis states of the N-dimensional position Hilbert space. The coin subspace for $l \neq 0$ is a 2-dimensional Hilbert space spanned by $\{|\rightarrow\rangle, |\circlearrowleft\rangle\}$ with the matrix representation

$$\hat{C} = \frac{2}{1+l}\begin{pmatrix} 1 & \sqrt{l} \\ \sqrt{l} & l \end{pmatrix} - \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{2.9}$$

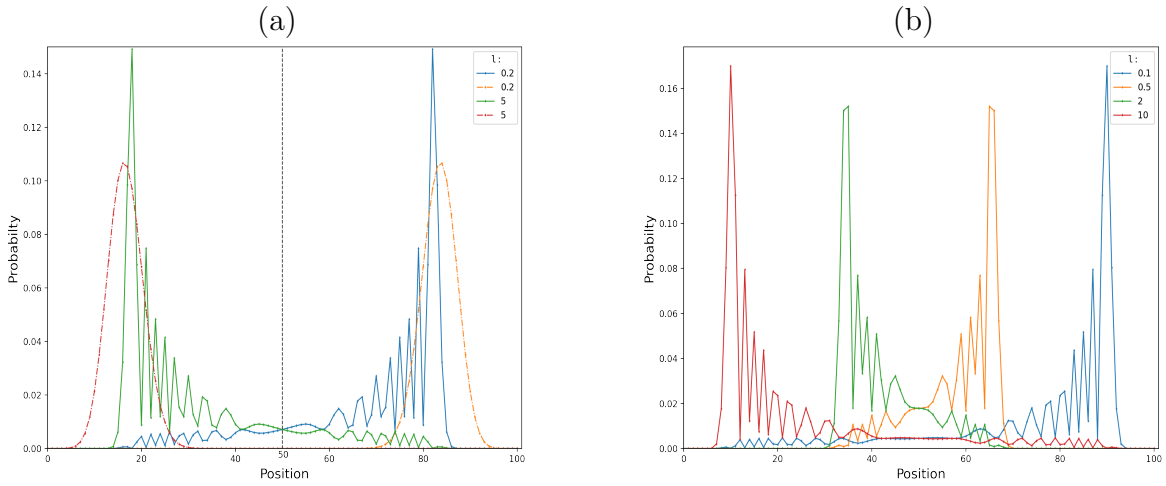


Figure 2.6: LQW on a directed-line. (a) Comparison of quantum (solid line) and classical (dashed line) walker's probability distribution for $l = 0.2, 5$ at $t = 100$. (b) Walker distribution with $N = 101$ at $t = 100$ for different self-loop weights.

also the shift operator is

$$\hat{S} = \sum_{x=0}^{N-1} |\rightarrow\rangle \langle\rightarrow| \otimes |x+1\rangle \langle x| + |\circlearrowleft\rangle \langle\circlearrowleft| \otimes |x\rangle \langle x| \tag{2.10}$$

The walks where self-loop weight is equal to 0 or 1 are trivial because, in the former case, the coin is one-dimensional, and for the latter case, the coin is just the two-dimensional Pauli-X matrix. So, for $l = 1$, the evolution is the alternate application of $+1$ shift and a self-loop.

Walker's distribution for different self-loop weights is shown in Fig.2.6b. Due to the symmetry of the 2-dimensional coin [Eq.(2.9)], the distributions for weights l and $1/l$, after evolution for time t, are mirror images along the $x = t/2$ line. The same is true for the classical random walk, as shown in Fig.2.6a. For $l < 1$, the probability of the walker moving forward is higher than staying at the same vertex. In this case, the classical walker's distribution is ahead of the quantum walker's distribution. For $l > 1$, the walker's probability of remaining at the same vertex is higher, and the quantum distribution is ahead in this case.

Fig.2.7 compares the variance of quantum and classical distributions. The classical variance has a maxima at $l = 1$, which is expected because the probabilities to hop one step forward and be at the same vertex are equal. Whereas, the quantum distribution has a minima

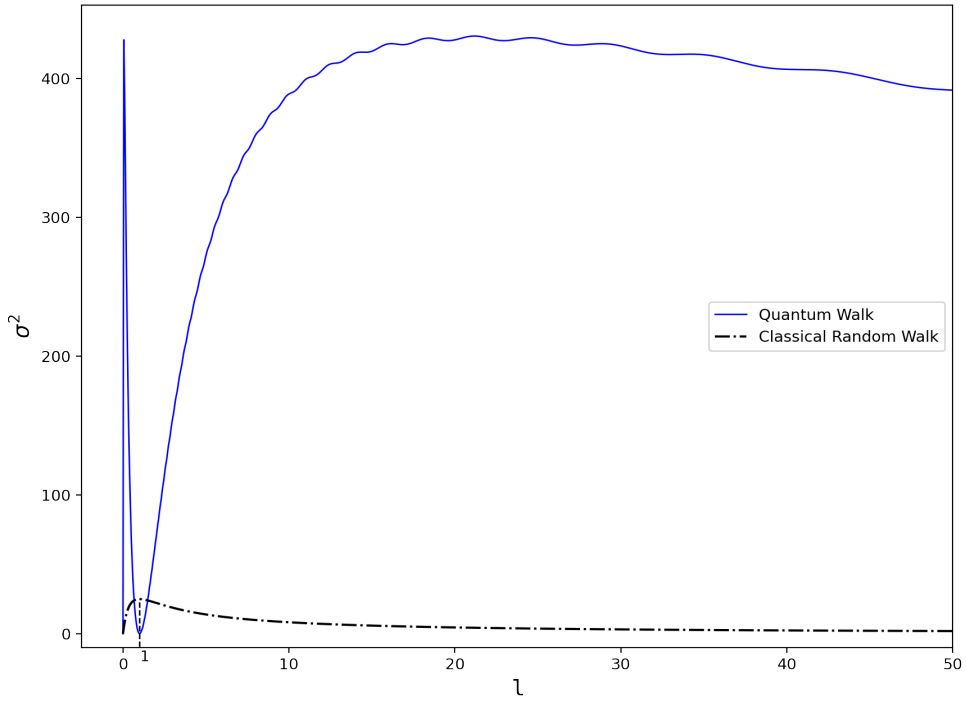


Figure 2.7: Variance of the probability distribution of directed quantum and classical walks on a line at $t = 100$

at $l = 1$ as the coin is a Pauli-X matrix. Similar to the case of the undirected walk for $l > 1$, the spread of the distribution increases as l increases but also the probability at the peak increases giving a maximum in the variance. Since the distribution is symmetric about $x = t/2$ for l and $1/l$, the same argument can be made for $0 < l < 1$.

2.3 Hypercube

An n-dimensional hypercube [11, 22] has $N = 2^n$ vertices with a degree of n at every vertex, so the Hilbert space of the walk is $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P = \mathbb{C}^n \otimes \mathbb{C}^{2^n}$. Every node is an n-bit string starting from (00...0) to (11...1) represented as $|0\rangle$ through $|2^n - 1\rangle$ in the position Hilbert space. An edge connects two vertices $\vec{x}$ and $\vec{y}$ if they differ only in a single bit, i.e., $|\vec{x} - \vec{y}| = 1$, where $|\vec{x}|$ denotes the Hamming weight of $\vec{x}$. The n edges connected to a vertex are n-dimensional vectors $\vec{e_i}$ (labeled $|i\rangle$ in $\mathcal{H}_C$) with the i^{th} bit-string equal to 1

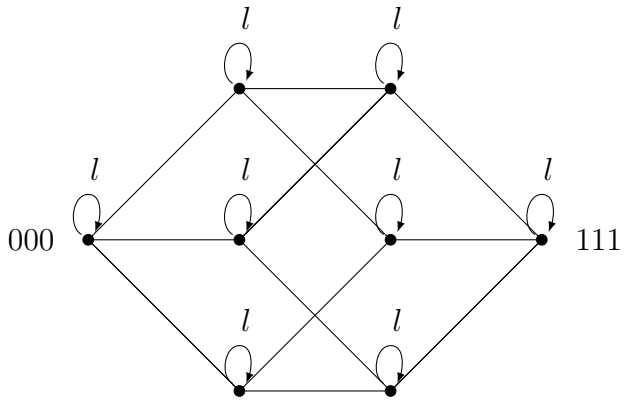


Figure 2.8: Walk on a 3-dimensional hypercube with a self-loop on each vertex with weight l

and rest zero. Therefore, a vertex $\vec{x}$ is connected to n other vertices, which are $\vec{x} \oplus \vec{e_i}$ for $i \in \{0, 1, \ldots n-1\}$, where $\oplus$ is bit-wise addition *modulo* 2.

For the LQW we add a dimension to the coin-space such that $\mathcal{H}_C$ is now spanned by $\{|0\rangle, |1\rangle, ..., |n-1\rangle, |\circlearrowleft\rangle\}$. We can write the matrix representation of the coin operator for a 3-dimensional hypercube in Fig.2.8 from Eq.(2.1) as

$$\hat{C} = \frac{2}{3+l} \begin{pmatrix} 1 & 1 & 1 & \sqrt{l} \\ 1 & 1 & 1 & \sqrt{l} \\ 1 & 1 & 1 & \sqrt{l} \\ \sqrt{l} & \sqrt{l} & \sqrt{l} & l \end{pmatrix} - \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} \tag{2.11}$$

The Shift operator for an n-dimensional hypercube with a self-loop is given by

$$\hat{S} = \sum_{\vec{x}} \left(\sum_{i=0}^{N-1} |i\rangle \langle i| \otimes |\vec{x} \oplus \vec{e_i}\rangle \langle \vec{x}| + |\circlearrowleft\rangle \langle \circlearrowleft| \otimes |\vec{x}\rangle \langle \vec{x}| \right) \tag{2.12}$$

Due to the permutation symmetry of the coin and the initial state being $|s_c\rangle$ (eigenvector with eigenvalue 1), the vertices with equal hamming-weight are equivalent, and their amplitudes are also equal at any time t. This means for $l = 0$, the walk on hypercube can be reduced to walk on a line [11]. Fig.2.9 shows the quantum and classical walker's probabilities as a function of the Hamming weight for different self-loop weights. The occupied states either have odd or even hamming weights without the self-loop weight. This effect disappears as l is increased.

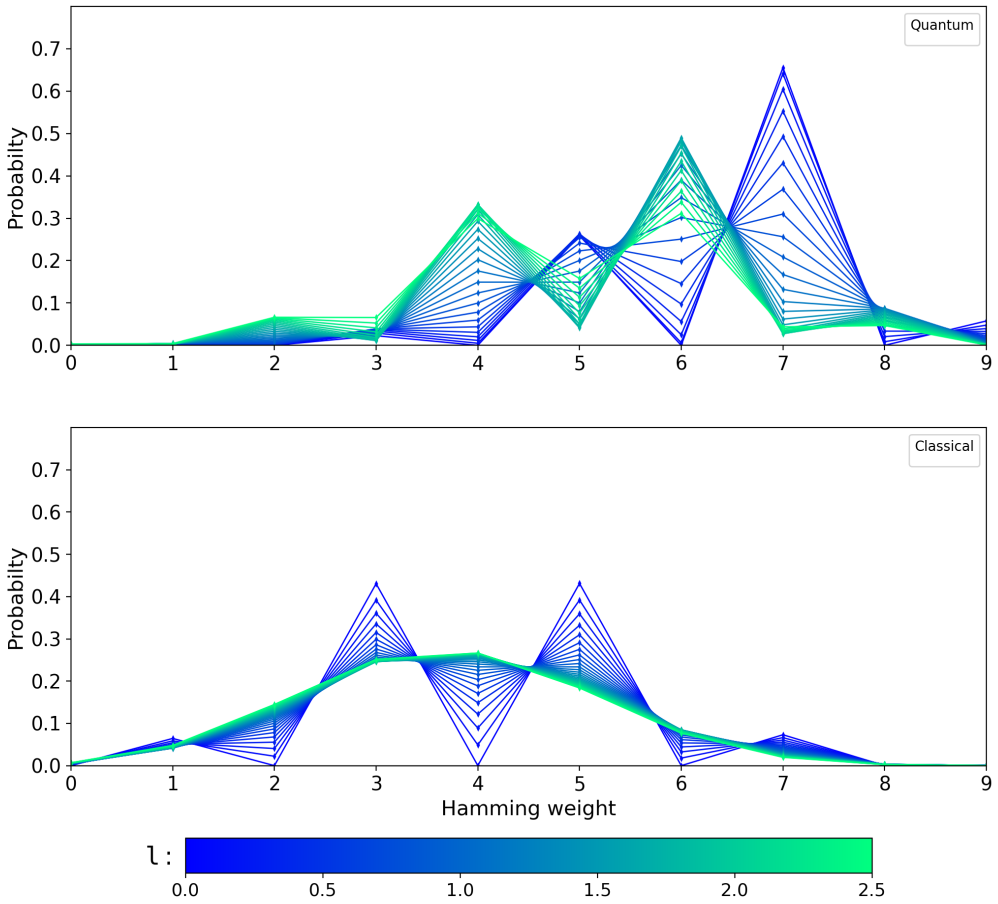
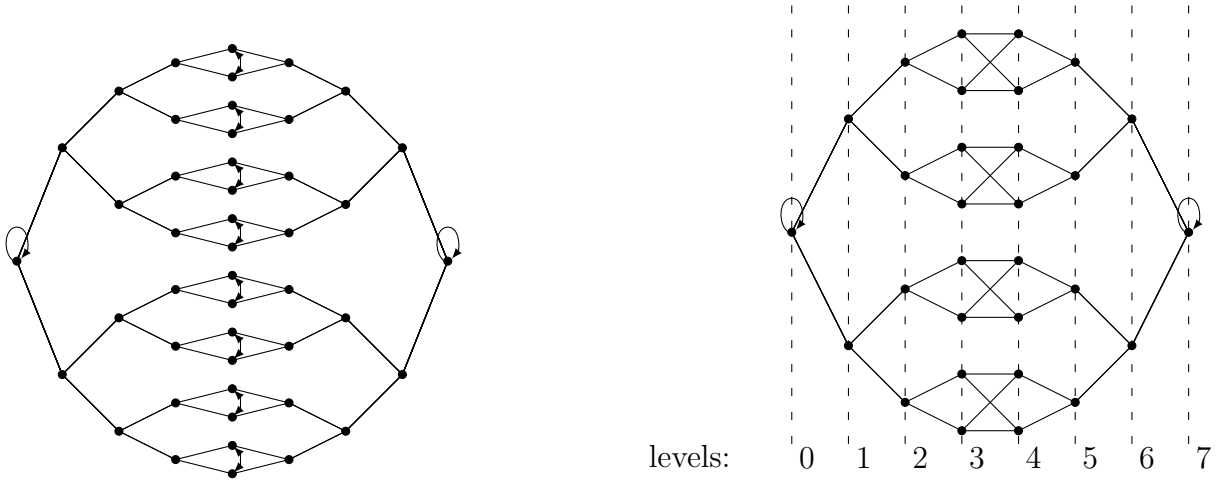


Figure 2.9: Probability as a function of hamming weight on the 9-dimensional hypercube. Comparison of quantum and classical walker's probability distribution for different values of self-loop weight at $t = 9$.

2.4 Binary tree

To perform a coined quantum walk on a glued binary tree, we have to make the degree of all the vertices equal, which can be done either by adding a self-loop at the 2-degree vertices or by connecting two 2-degree vertices by an edge; examples are shown in Fig.(2.10a) and Fig.(2.10b).

Now, every vertex is connected with three edges which makes $\mathcal{H}_c$ a three-dimensional Hilbert space spanned by $\{|1\rangle, |2\rangle, |3\rangle\}$. We can represent the vertices by the states $\{|0\rangle, |1\rangle, ..., |N-1\rangle\}$ in $\mathcal{H}_P$. To construct the shift operator, we label each edge by either 1,2, or 3 such that a



(a) (b)

Figure 2.10: Modified Glued Binary tree. (a)Two binary trees of depth $d = 4$ are glued together at their leafs and extra edges are added at the 2-degree vertices. Total vertices $N = 3.2^d - 2$. (b)Two binary trees of depth $d = 3$ are connected to each other through the leafs and one self-loop is added to each of the root nodes. Total vertices $N = 2^{d+2} - 2$.

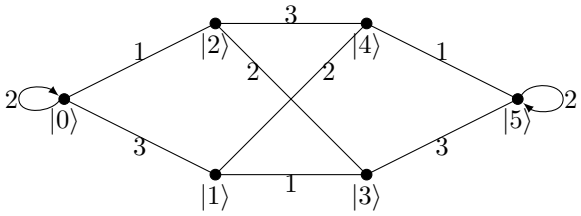


Figure 2.11: Labeling edges of a graph without any vertex having two edges with the same label.

vertex has no two edges with the same label. Then, the shift operator will be

$$\hat{S} = \sum_{i=1}^{3} |i\rangle \langle i| \otimes \hat{S}_i \tag{2.13}$$

where $\hat{S}_i$ performs the shift on the edges labeled i. For instance, let us say the walker is on the vertex $|4\rangle$ of graph in Fig.2.11, then the operator $\hat{S}_1$ shifts the walker to the vertex $|5\rangle$, $\hat{S}_2$ shifts it to $|1\rangle$, and $\hat{S}_3$ to $|2\rangle$. Therefore, $\hat{S}$ would act on a state $|a\rangle = \left(\frac{1}{6}|1\rangle\langle 1| + \frac{1}{3}|2\rangle\langle 2| + \frac{1}{2}|3\rangle\langle 3|\right) \otimes |4\rangle\langle 4|$ as

$$\hat{S}|a\rangle = \frac{1}{6}|1,5\rangle\langle 1,5| + \frac{1}{3}|2,1\rangle\langle 2,1| + \frac{1}{2}|3,2\rangle\langle 3,2| \tag{2.14}$$

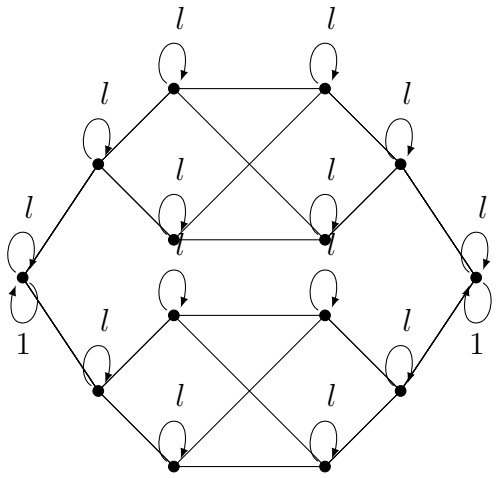


Figure 2.12: LQW on a glued tree. The roots have two self-loops having weight one and l.

where, $|1,5\rangle \equiv |1\rangle \otimes |5\rangle$. Matrix form of S_1 is

$$\hat{S}_1 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.15}$$

and similarly, we can write $\hat{S}_2$ and $\hat{S}_3$.

For LQW on the binary tree, since the degree of each vertex is three plus one self-loop, the coin-space can be spanned by $\{|1\rangle, |2\rangle, |3\rangle, |\circlearrowleft\rangle\}$, and the coin operator is the same as in Eq.(2.11). The shift operator can be written as

$$\hat{S} = \sum_{i=1}^{3} |i\rangle \langle i| \otimes \hat{S}_i + |\circlearrowleft\rangle \langle \circlearrowleft| \otimes I_N \tag{2.16}$$

For the walk on binary trees using this coin, the vertices at the same level are equivalent. Fig.2.13 compares classical and quantum walker's probability amplitudes on the tree levels. The classical walker slows down on increasing self-loop weight, but the probability of finding the quantum walker gets concentrated in the center.

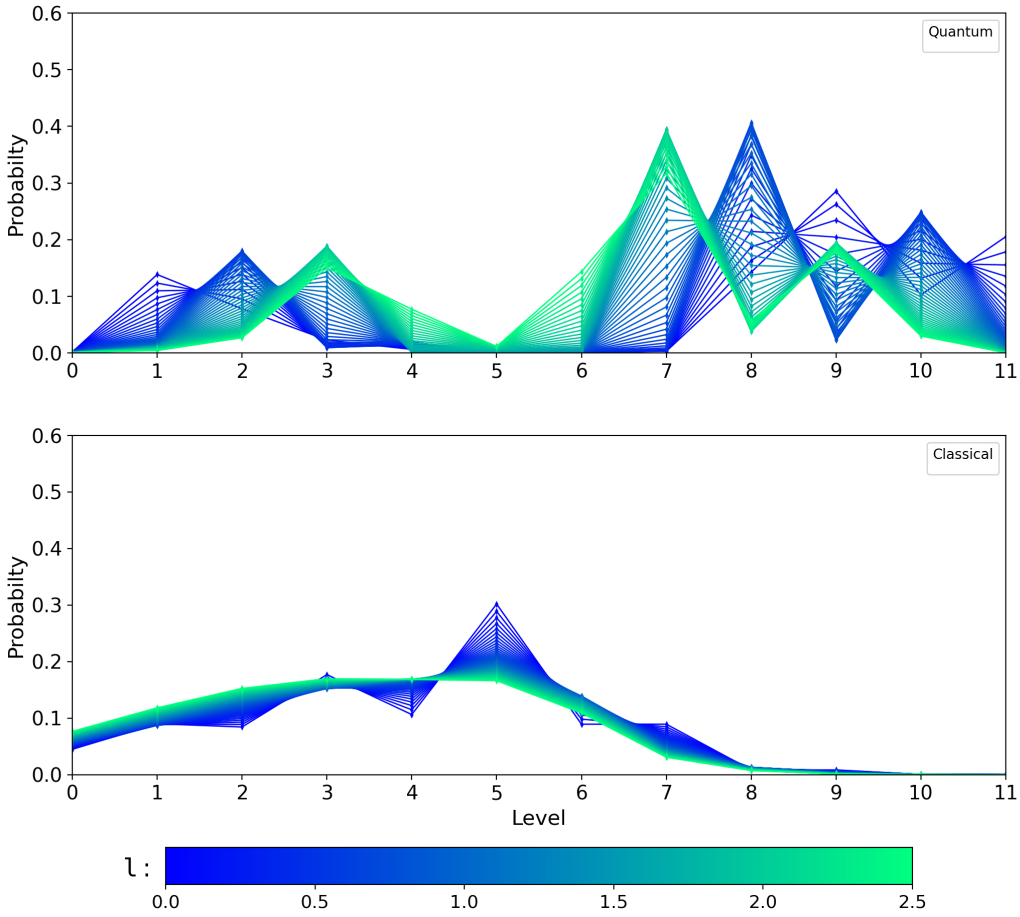


Figure 2.13: Probability as a function of levels on the modified binary tree ($N = 2^{5+2} - 2$). Comparison of quantum and classical walker's probability amplitudes for different values of self-loop weights at $t = 11$.

This chapter compared the probability distribution of lackadaisical quantum walks with its classical counterpart, i.e., the lazy random walks. The behavior of the walks differs significantly. For an undirected walk on a line, the variance of the quantum distribution increases to a maximum on increasing l and reduces on the further increase. However, the classical variance is maximum for $l = 0$ and approaches 0 on increasing l.

For a directed walk on the line, $l < 1$ implies faster classical distribution, whereas $l > 1$ implies faster quantum distribution. However, the classical variance is maximum for $l = 0$ and approaches 0 on increasing l.

For $l \geq 0$ The quantum distribution is faster than the classical distribution for the hypercube and binary tree due to the interference effects in the quantum evolution.

Chapter 3

Hitting time

Classical random walks can be characterized by time scales like hitting time, correlation time mixing time and so on. In [19, 23] different definitions of hitting times that can be used for a quantum walk are provided, here we would use them to compare classical and quantum walks.

We apply the method developed in [22] to calculate the average hitting time of a quantum walk on a graph having N vertices with degree d, where the walker starts on a vertex labeled as x_0 and stops if it reaches some x_f. Using the *measured quantum walk*, an alternating application of the walk operator $\hat{U}$ and a projective measurement (to check whether the walker has reached x_f), we determine the first-crossing probabilities at every time step. The projective measurement has two outcomes P and $Q = I - P$, where $P = I_d \otimes |x_f\rangle \langle x_f|$ is the projector onto the final vertex.

Let us start with an initial density matrix $\rho_0 = |\psi_0\rangle \langle \psi_0|$ in the coin walker space, then, the first-crossing probability at a time step t is,

$$p(t) = Tr(\hat{P}\hat{U}[\hat{Q}\hat{U}]^{t-1}\rho_0[\hat{U}^\dagger\hat{Q}]^{t-1}\hat{U}^\dagger\hat{P}) \tag{3.1}$$

And the average hitting time is defined as,

$$\tau = \sum_{t=1}^{\infty} tp(t) \tag{3.2}$$

We compute the estimate (a lower bound) of the average hitting time by iterating the quantum walk for the shortest time T such that

$$\sum_{t=1}^{T} p(t) \geq 1 - \epsilon \tag{3.3}$$

Therefore, the estimated hitting time is

$$\tau_{est} = \sum_{t=1}^{T} tp(t) \tag{3.4}$$

Taking ϵ closer to 0 gives better estimates of τ. In the process of computing the average hitting time, we have also calculated the concurrent hitting time $\tau_c(1-\epsilon) = \mathrm{T}$, which is defined as the shortest time τ_c for which,

$$\sum_{t=1}^{\tau_c(p)} p(t) \geq p \tag{3.5}$$

holds true.

We use Eq.(3.4) to compute the estimated hitting time of LQW for the graphs in chapter 2. We also compare the quantum hitting time with classical hitting time, where we use the transfer matrix for the classical walker evolution. The classical hitting times are calculated similarly to the quantum hitting times.

3.1 Directed walk on a line

The walker begins from the state $|0\rangle$ and stops when it reaches $|N-1\rangle$. From Eq.(2.9) and (2.10), we get the operator $\hat{U} = \hat{S}.(\hat{C} \otimes I)$ and perform the measured quantum walk on the directed-line using the projection operators,

$$\hat{P} = I_2 \otimes |N-1\rangle \langle N-1|, \ \hat{Q} = I_{2N} - \hat{P}. \tag{3.6}$$

For $l = 0$ the coin operator is $1-$dimensional, i.e., the walker moves to the next site [Fig.2.5] on the line at every step with probability 1. Hitting time can be trivially calculated in this case.

We calculate the hitting times as a function of the self-loop weight value l for a walk with $N = 101$ by summing the series in Eq.(3.4) with $\epsilon = 10^{-6}$. We compare classical and quantum values in Fig.3.1.

The quantum and classical average hitting times are equal for $l = 0, 1$. The average hitting time for both the quantum and the classical case, increases as l is increased. The concurrent hitting time for $l < 1$ decreases for the quantum and increases for the classical case as we increase l from 0 to 1. For $l > 1$, it increases for both the cases. We can see that the average classical hitting time is lower for $l < 1$, implying that the classical walker hits faster as compared to the quantum walker.

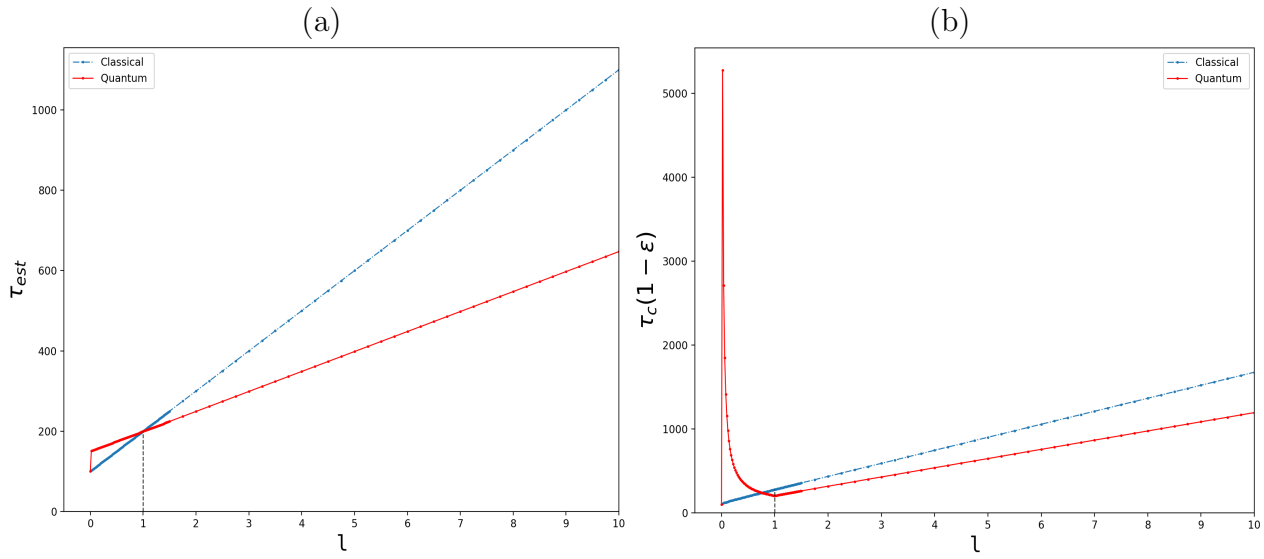


Figure 3.1: Quantum walk on a directed line. Comparison between the quantum (solid red) and classical (dashed blue) case for $N = 101$ and $\epsilon = 10^{-6}$ (a) Average hitting time, (b) Concurrent hitting time.

3.2 Hypercube

We assume that the walk starts at the vertex $|0\rangle$ and stops at $|2^n - 1\rangle$. The projection operators for an n-dimensional hypercube are

$$\hat{P} = I_n \otimes |2^n - 1\rangle \langle 2^n - 1|, \ \hat{Q} = I_{n2^n} - \hat{P} \tag{3.7}$$

It has been shown in [22] that the average hitting time as a function of hypercube dimension, grows polynomially for the quantum walker and exponentially for the classical walker.

Here, we compare the quantum and classical hitting times for a 5−dimensional hypercube

with $\epsilon = 10^{-6}$ as a function of self-loop weight. The average and concurrent hitting times vary linearly with l as shown in Fig.3.2. By making the walker slower, i.e., increasing the self-loop weight, a linear advantage of the quantum over the classical walker can be seen.

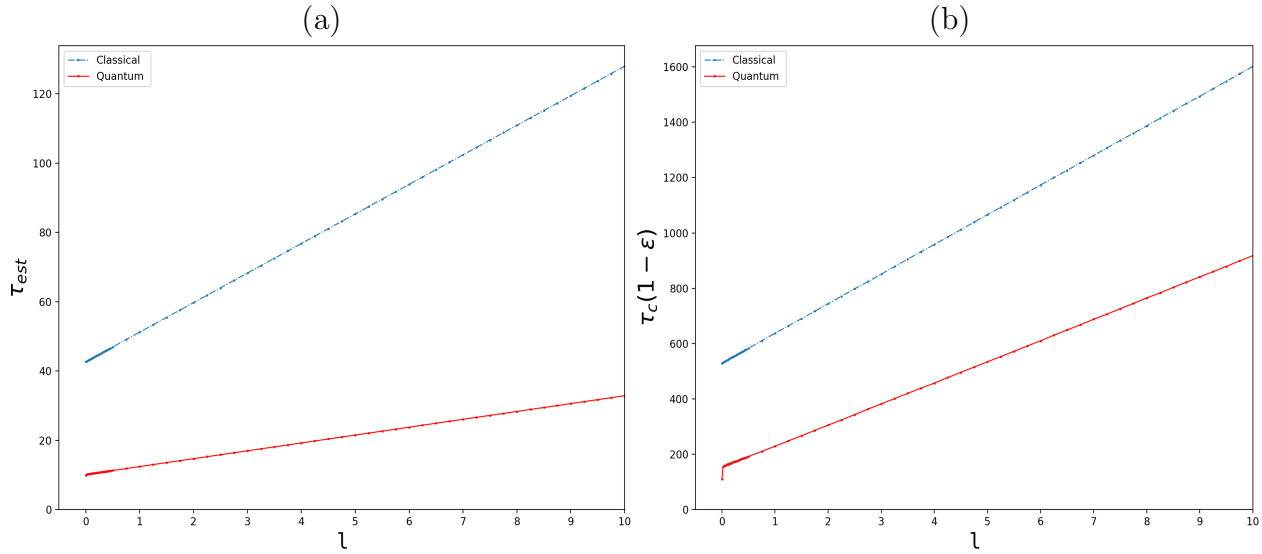


Figure 3.2: Walk on 5-dimensional hypercube. Comparison between the quantum and classical case for $\epsilon = 10^{-6}$ (a) Average hitting time, (b) Concurrent hitting time.

3.3 Binary Tree

To compute the hitting time on the binary tree, we consider the left root node as the starting vertex and the right as the exit vertex. The projection operators can be written as

$$\hat{P} = I_4 \otimes |N-1\rangle \langle N-1|, \ \hat{Q} = I_{4N} - \hat{P} \tag{3.8}$$

Hitting times for both types of graphs (Fig.2.10a and 2.10b) show similar behavior. Fig.3.3 shows hitting time as a function of d (depth) for the modified binary tree with $N = 2^{d+2} - 2$. The classical average hitting shows exponential growth compared to the polynomial growth of the quantum average hitting time as a function of d. As with a hypercube, quantum walk on a binary tree also yields an exponential advantage [Fig.3.3a].

Due to the fact that concurrent hitting time is the running time of the algorithm, it has larger values as compared to the average hitting times.

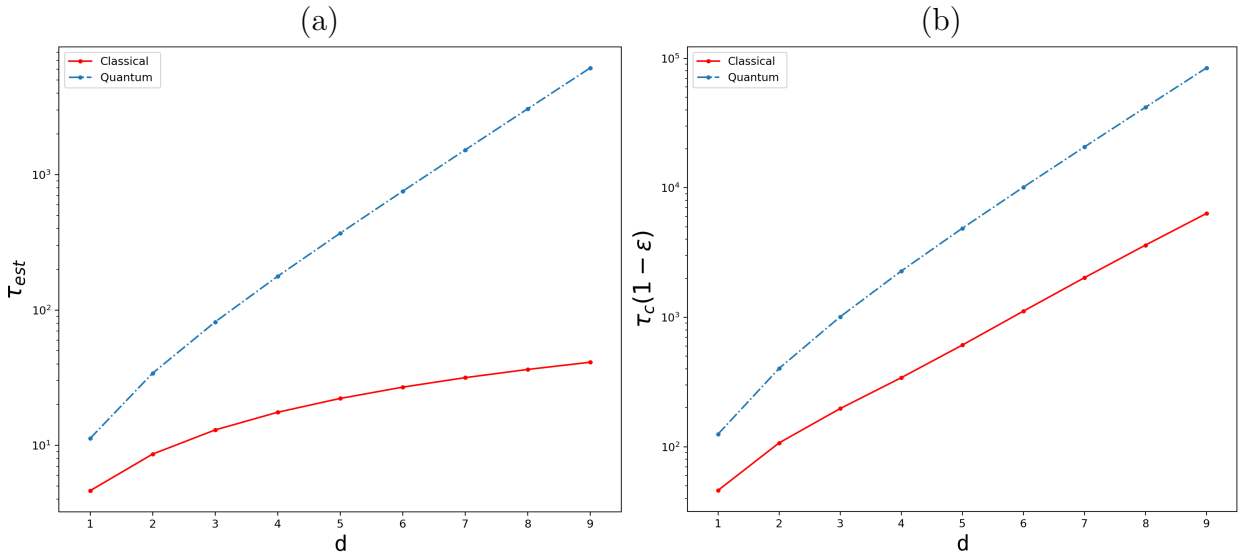


Figure 3.3: Quantum and classical hitting times as a function of d for binary trees with $N = 2^{d+2} - 2$ and $\epsilon = 10^{-6}$. (a) Average hitting time, (b) Concurrent hitting time.

Quantum and classical hitting times for $d = 4$ and $N = 2^{4+2} - 2$ modified binary tree as a function of self-loop weight is shown in Fig.3.4. These hitting times vary linearly with l. Similar to the case of a hypercube, if we make the walker slower by increasing the self-loop weight in a binary tree, a linear advantage of the quantum over the classical walker can be seen. However, there is no advantage in concurrent hitting times on increasing the self-loop weight.

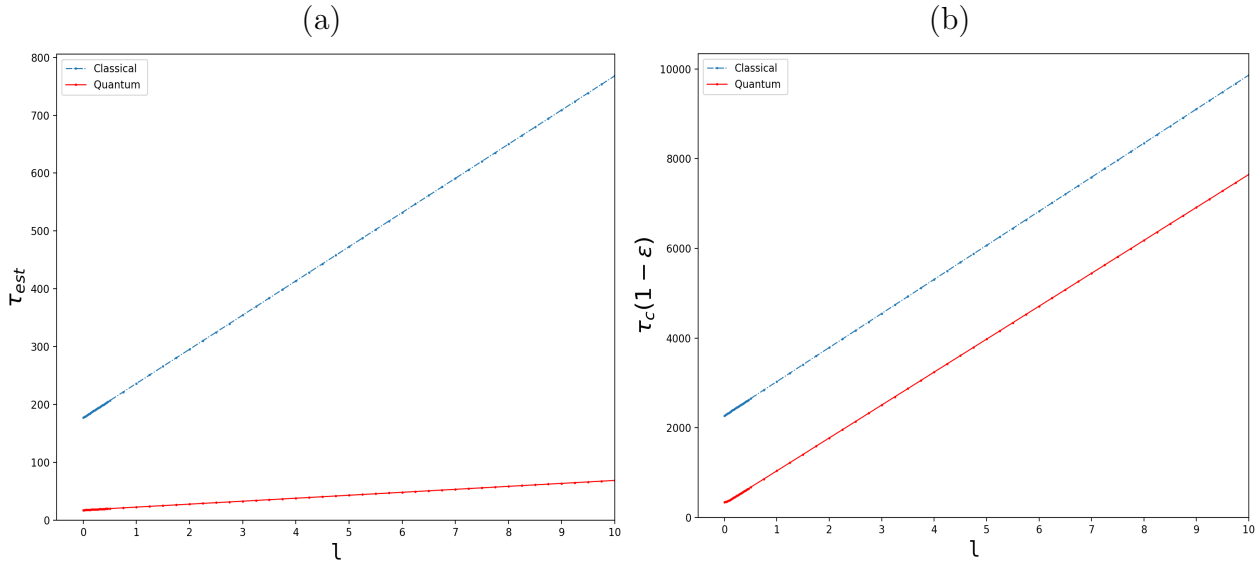


Figure 3.4: Walk on binary tree. Comparison between the quantum and classical case for $N = 2^{4+2} - 2$ and $\epsilon = 10^{-6}$ (a) Average hitting time, (b) Concurrent hitting time.

Chapter 4

Searching a marked node using LQW

This chapter explores the use of Lackadaisical Quantum-Walk for finding the marked node(s) on a graph. It has been used to search on complete graphs [20], arbitrary 2D grids [21], and hypercubes [24], significantly improving the probability of finding the marked node(s) compared to the algorithms that use regular quantum walks. Here, we will search the marked node(s) using the Lackadaisical Quantum-walk.

Grover's algorithm is an iterative process that works by rotating the state of the system closer to the marked states. This rotation depends directly on the number of marked states and inversely to the total number of vertices.

We start with the search algorithm using a regular quantum walk (no self-loops). The Hilbert space for the Quantum walk is $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P = \mathbb{C}^d \otimes \mathbb{C}^N$. The search algorithm is the repeated application of the unitary operator $U_{search} = S(C \otimes Q)$. C is the coin operator (Grover diffusion operator), S is the shift operator, and Q is the query that flips the sign of the marked node's amplitude. For instance, if M is the set of marked node(s), then the query Q is given as

$$Q = I_N - 2 \sum_{m \in M} |m\rangle \langle m| \tag{4.1}$$

and the coin operator is

$$C = 2 |s\rangle \langle s| - I_d \tag{4.2}$$

where we have

$$|s\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d-1} |i\rangle \,, \quad |s_p\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \tag{4.3}$$

The system starts in the state $|\psi(0)\rangle = |s\rangle \otimes |s_p\rangle$, i.e., all nodes are equally guessed and after t time steps the state of the system becomes $|\psi(t)\rangle = U^t |\psi(0)\rangle$.

The use of LQW in the algorithm requires adding a dimension to the coin subspace, modifying the coin's initial state as

$$|s_c\rangle = \frac{1}{\sqrt{d+l}} \left(\sum_{i=0}^{d-1} |i\rangle + \sqrt{l} \, |\circlearrowleft\rangle \right) \tag{4.4}$$

and the coin operator as

$$C = 2 \, |s_c\rangle \, \langle s_c| - I_d \tag{4.5}$$

The unitary operator U of the algorithm is applied iteratively so that the probability accumulates at the marked node. The optimal self-loop weight proposed in [21, 25] for finding k marked nodes with high probability is

$$l = k\frac{d}{N}, \tag{4.6}$$

where d is the degree of each vertex and N is the total number of vertices.

4.1 LQW Search on a Binary Tree

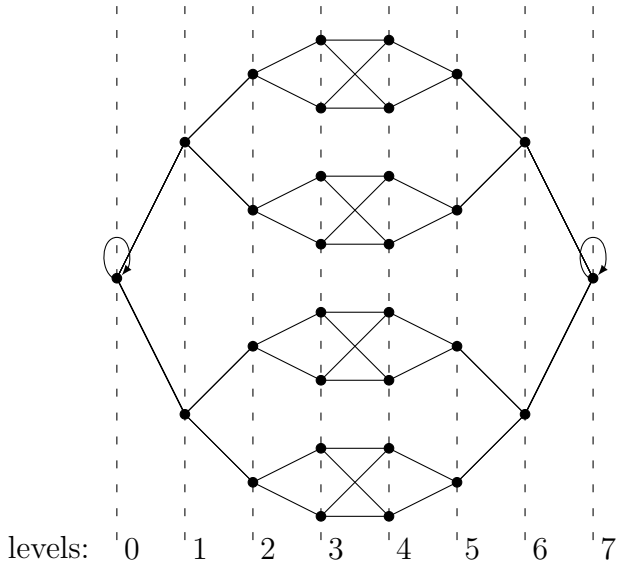


Figure 4.1: Two binary trees of depth $d = 3$ are connected to each other through the leafs and one self-loop is added to each of the root nodes. Total vertices $N = 2^{d+2} - 2$.

Let us perform a search on the binary tree shown in Fig.4.1, for which $d = 3$ and $N = 2^{3+2} - 2$.

First, consider a single marked node other than the roots. The success probability as a function of time is shown in Fig.4.2. We can see that the first peak of success probability is highest for the optimal self-loop weight $l = 0.1$.

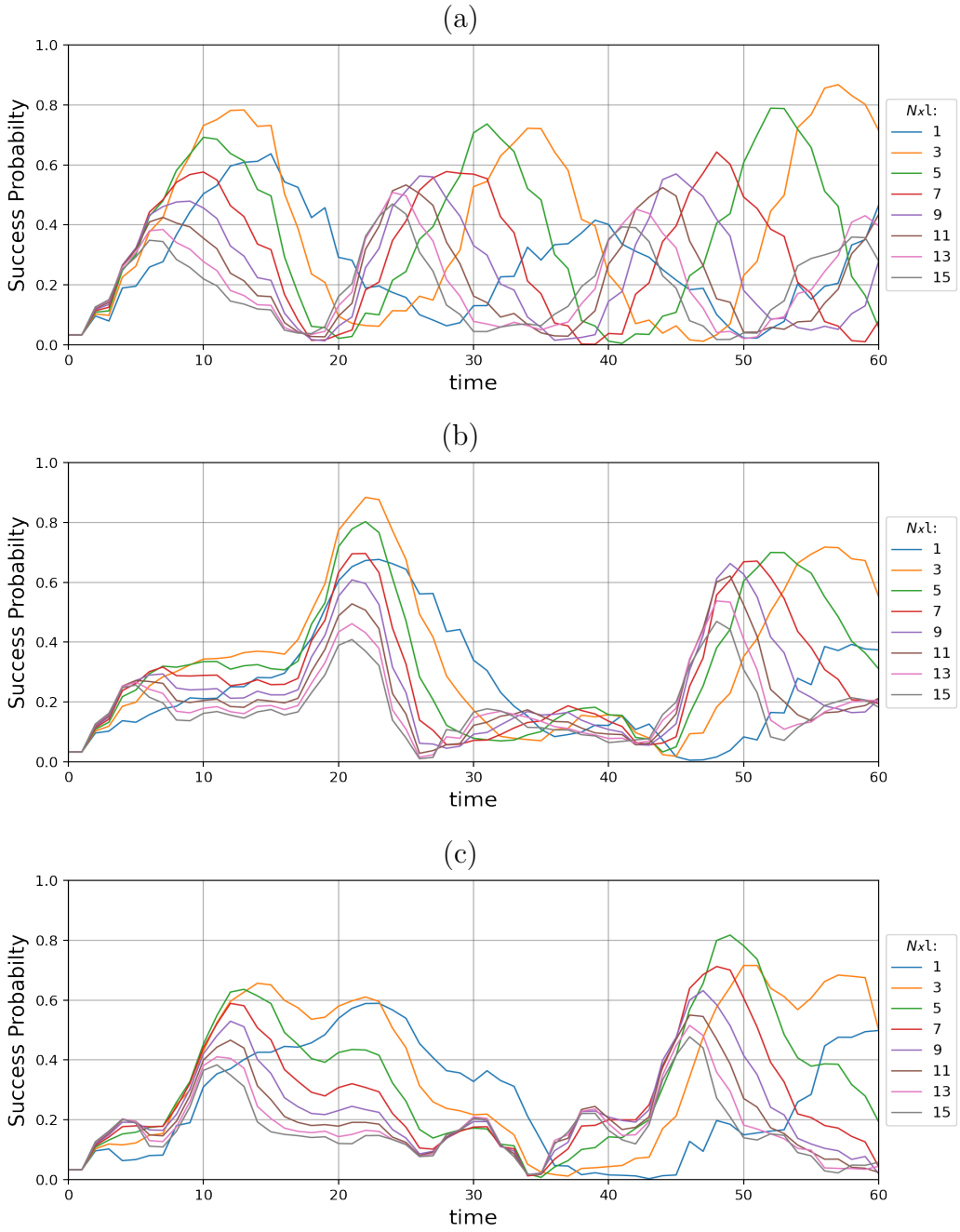


Figure 4.2: Single marked node on binary tree with $N = 30$. Probability of finding the walker at the marked node after 60 time steps for different values of self loop weights. Marked node at level (a) 1 or 6, (b) 2 or 5, and (c) 3 or 4 of Fig.4.1

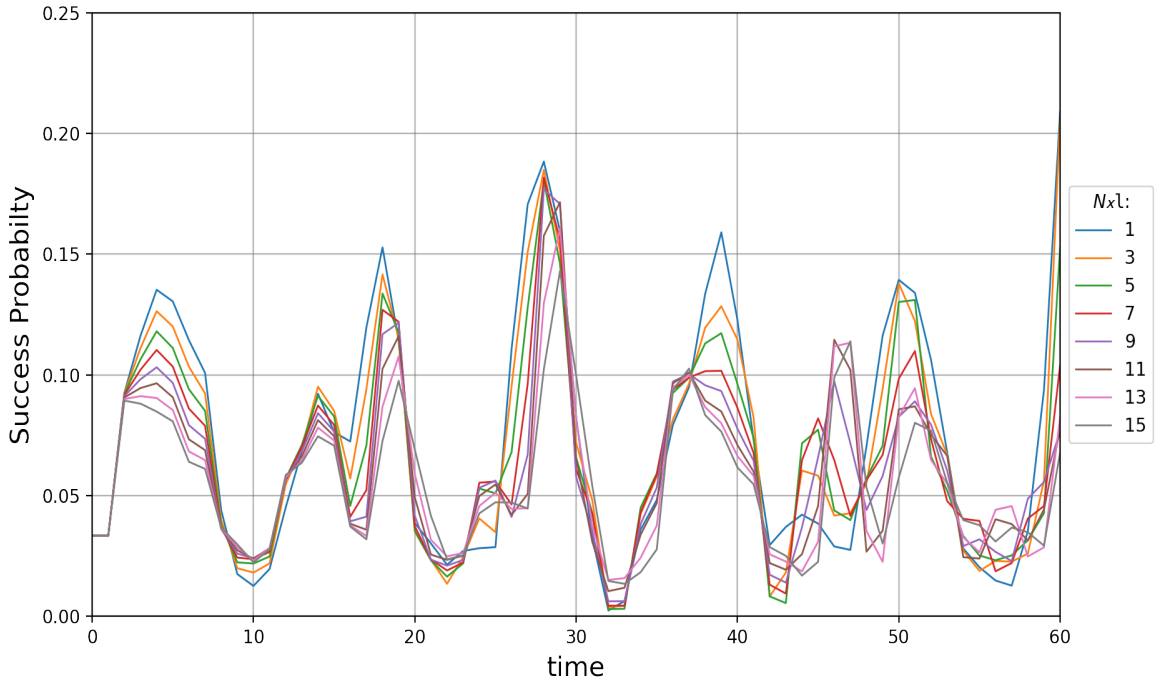


Figure 4.3: Success probability for single marked node at level 0 or 7 of Fig.4.1.

Vertices at a given level are equivalent [Fig.4.1] due to the symmetry of the graph and the quantum walk. Therefore, we can mark any node at a particular level because they would have the same behavior of the success probability as a function of time.

If a root node is marked, the implementation of the algorithm does not return high probabilities of finding that node [Fig.4.3]. This is because of another self-loop at the root node, introduced to make all the vertices to be of the same degree. The resultant self-loop weight at the root nodes is $1 + l$.

To search for the root nodes (or node with extra self-loops), we can modify the query operator such that it only flips the sign of amplitude of the states $|\circlearrowleft\rangle \otimes |m\rangle$ for $m \in M$. We can say that the coin state $|\circlearrowleft\rangle$ controls the application of query operator, or the search is on the entire coin-walker space rather than just the position space. The new query operator can be written as

$$Q' = I_{(d+1)N} - 2 \sum_{m \in M} |\circlearrowleft\rangle \langle\circlearrowleft| \otimes |m\rangle \langle m| \tag{4.7}$$

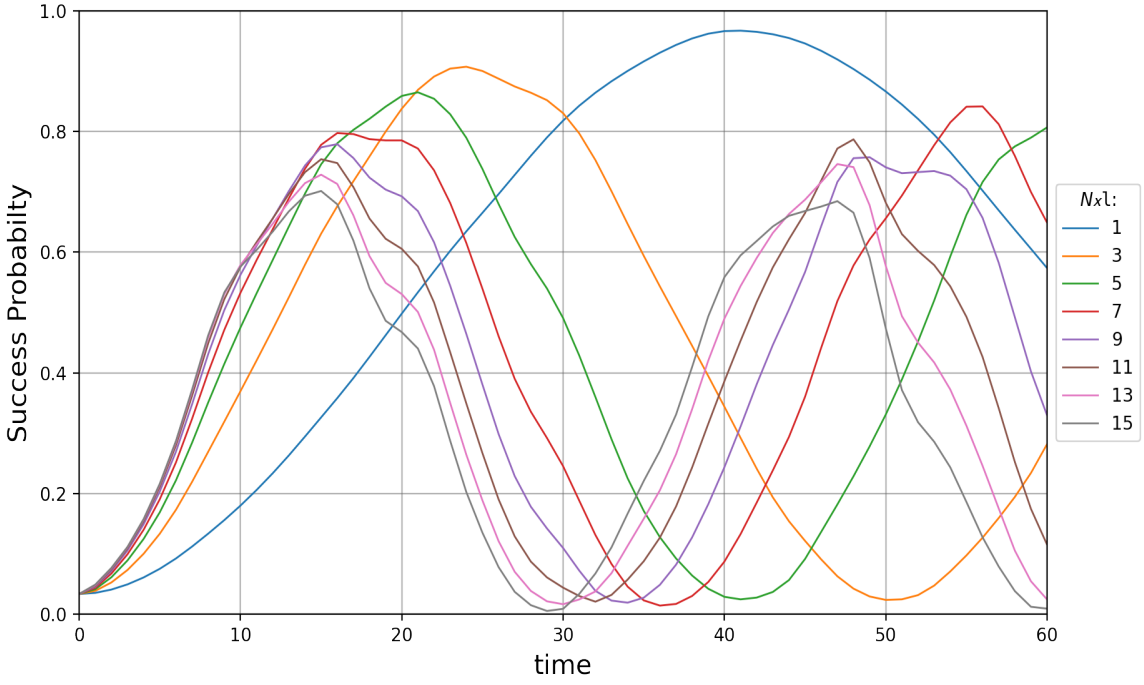


Figure 4.4: Success probability for single marked node at level 0 or 7 of Fig.4.1 when Q' is used.

Note that Q is an operator in position Hilbert space, but the operator Q' is an operator in the entire Hilbert space of the quantum walk. Now, we can implement the algorithm by repeated application of the unitary operator $U' = S(C \otimes I_N)Q'$.

Fig.4.4 shows the probability as a function of time when the marked node is one of the root nodes. With this modification of the algorithm, the root node can be searched. Fig.4.5 shows success probabilities for the marked nodes on the remaining levels of the binary tree. We can see that the first peak probability for the search with $l = 1/N$ is higher than that with optimal self-loop weight, although the time at which the probability reaches its maximum is also higher.

Fig.4.6(a) shows that the first maximum value of success probability increases as we decrease the self-loop weight irrespective of node type (root or non-root nodes).

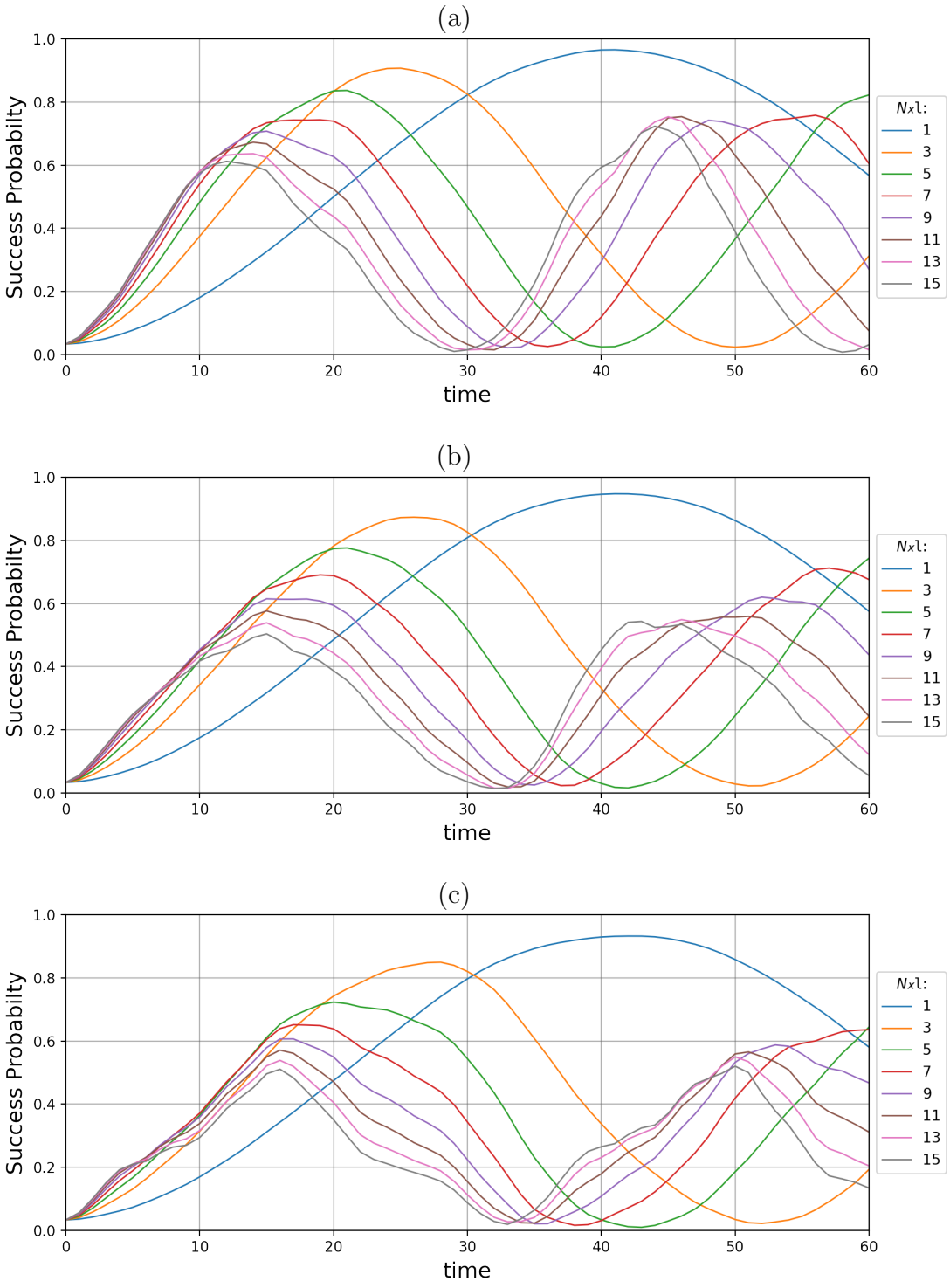


Figure 4.5: Probability of finding the walker at the single marked node on the binary tree for different values of self-loop weights. Marked node at level (a) 1 or 6, (b) 2 or 5, and (c) 3 or 4 of Fig.4.1

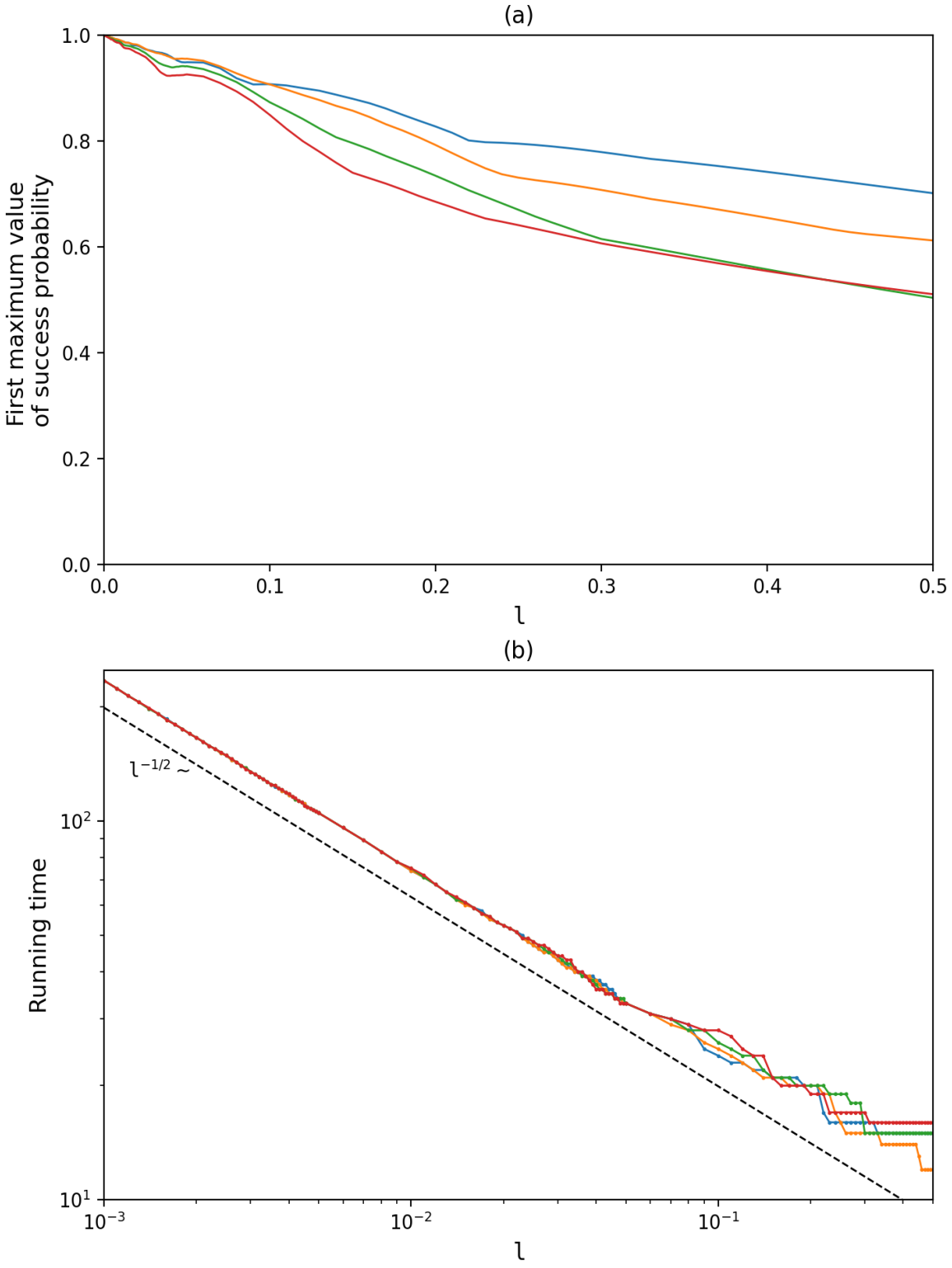


Figure 4.6: (a) First maximum value of success probabilities, (b) time to attain the first maximum (Runtime) as a function of self-loop weight for $l > 0$. A single marked node at level 0 or 7 (blue), 1 or 6 (orange), 2 or 5 (green), 3 or 4 (red).

Next, we consider non-adjacent multiple marked nodes (randomly marked). The cumulative probability of finding the walker on the marked nodes shows similar behavior to the case of the single marked node, i.e., the success probability grows to a maximum value that increases as we decrease the self-loop weight. Decreasing the self-loop weight also increases the time to attain that maximum value. Results are shown in Fig.4.7 and 4.8.

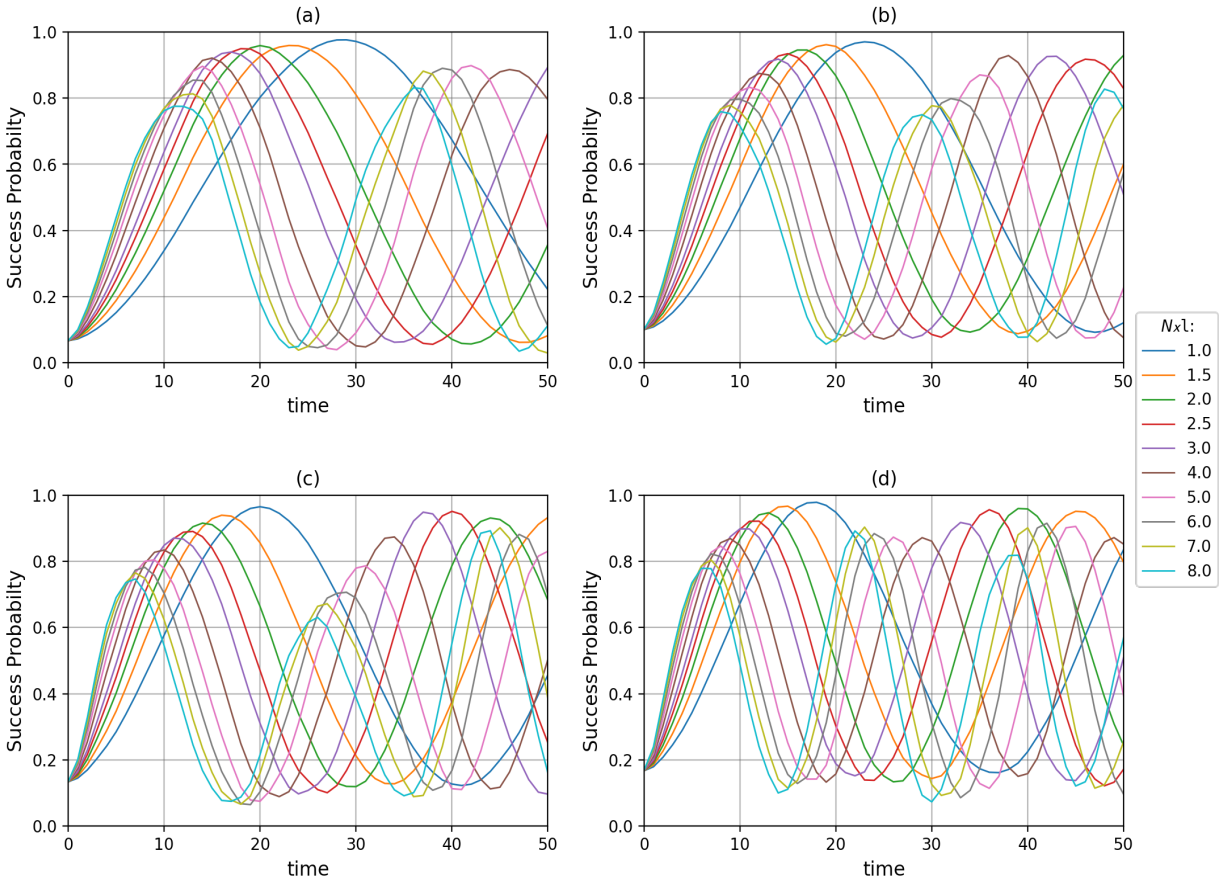


Figure 4.7: Success probability for (a) two, (b) three, (c) four, and (d) five non-adjacent multiple marked nodes after 50 time steps.

We can see that as the state is evolved, the probability of finding the walker oscillates between the marked states and the unmarked states. Our query operator applies a phase of $e^{\iota\pi}$ to the states $\sum_{m\in M} |\circlearrowleft\rangle \otimes |m\rangle$ in the complete coin-walker Hilbert space. So effectively, the marked states are $\sum_{m\in M} |\circlearrowleft\rangle \otimes |m\rangle$. These states have a high contribution in the success probability, although it is the sum of probabilities for the states $\sum_{m\in M} |m\rangle$ irrespective of the coin state.

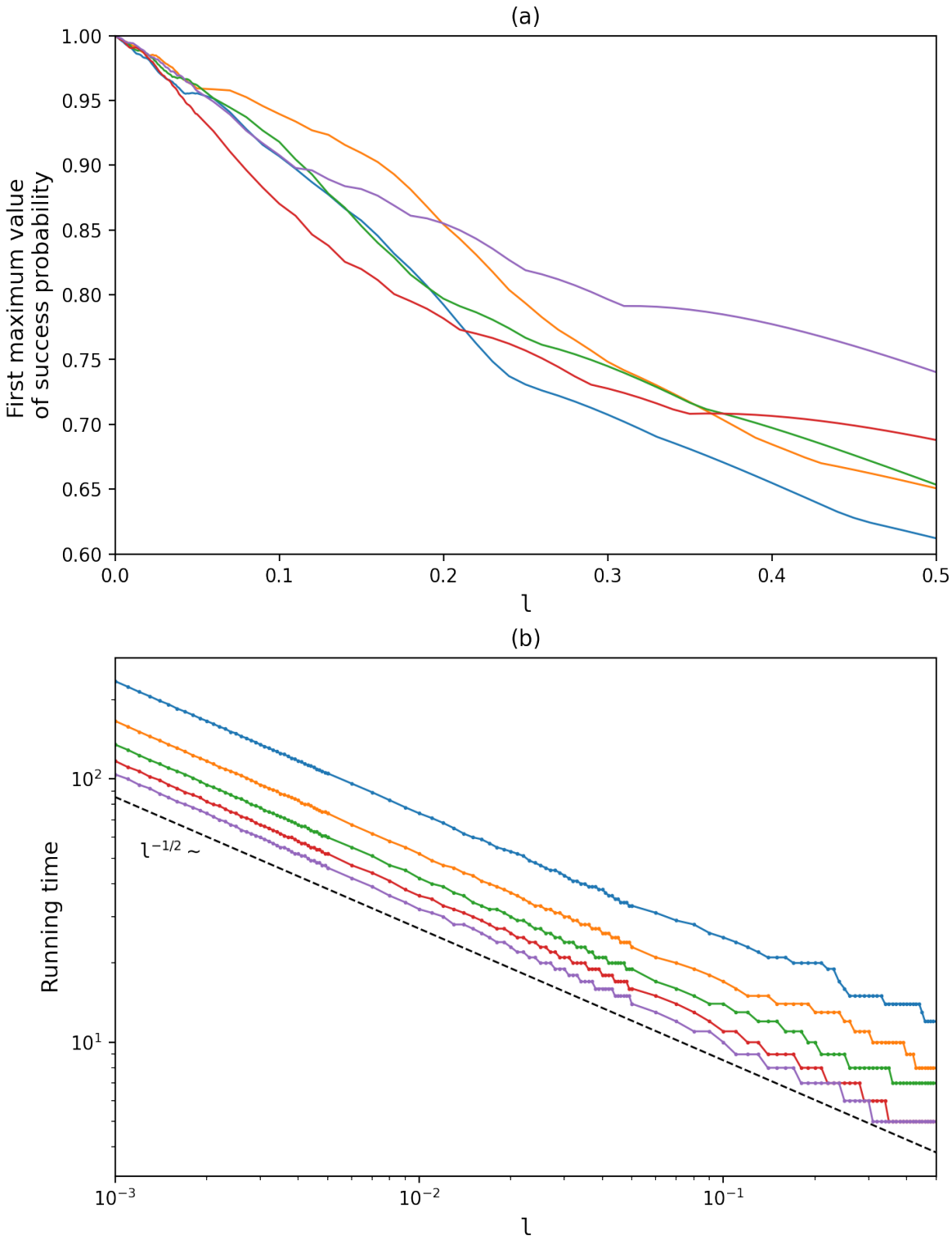


Figure 4.8: (a) First maximum value of success probabilities, (b) time to attain the first maximum (Running time) as a function of self-loop weight for $l > 0$. 1 (blue), 2 (orange), 3 (green), 4 (red) and 5 (purple) non-adjacent marked nodes.

We would now address the problem of adjacent marked nodes. Searching two adjacent marked nodes using the query operator Q is challenging, and the algorithm does not give high success probabilities. Whereas if we use Q', even the adjacent marked nodes can be searched. Fig.4.9 shows the search results for adjacent marked nodes.

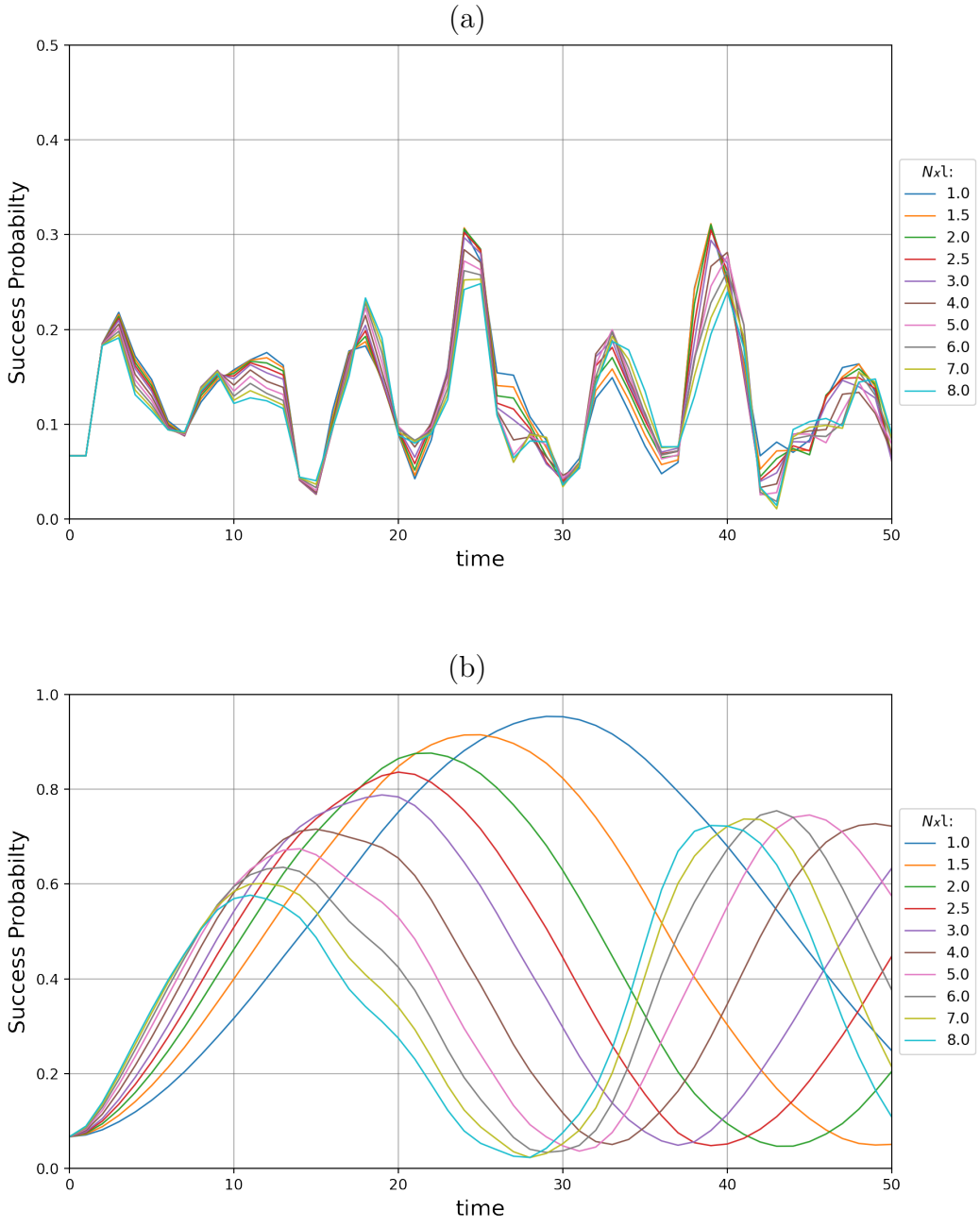


Figure 4.9: Success probabilities if adjacent nodes are marked and (a) Q is used, (b) Q' is used.

4.2 LQW Search on a Hypercube

In [24] LQW search algorithm on a hypercube is studied using the query operator Q. Here, we will use the modified query operator Q' and show that the results obtained for the binary tree also hold for search on a hypercube. As pointed out earlier, searching two adjacent marked nodes using Q is difficult in most graphs, including hypercube. However, the query operator Q' can give high success probabilities even with higher values of self-loop weight.

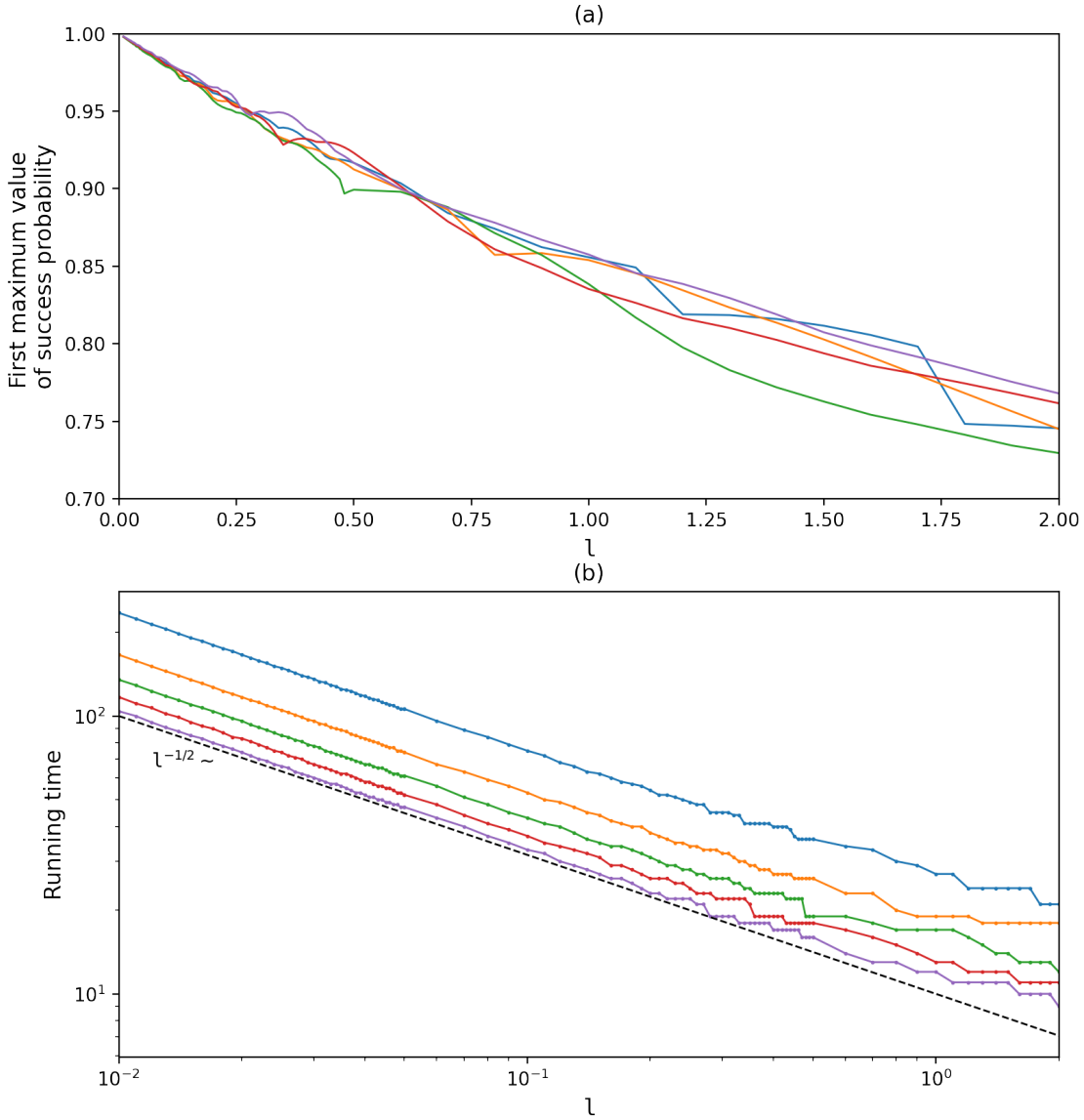


Figure 4.10: LQW search with Q' on a 7 dimensional hypercube. (a) First maximum value of success probabilities, (b) time to attain the first maximum (Running time) as a function of self-loop weight. 1 (blue), 2 (orange), 3 (green), 4 (red), and 5 (purple) non-adjacent marked nodes.

The algorithm with the query operator Q is faster if we have single or non-adjacent multiple marked nodes, i.e., the running time is less than the algorithm which uses Q'. But the limitations of using Q is that we cannot control the success probability, and for a certain type of marked solutions, the algorithm does not work. For instance, a marked node that already has a self-loop, two adjacent marked nodes or marked nodes that form a cycle in the graph. Whereas using Q', the success probability can be controlled, and any type of marked node(s) can be searched.

Let us consider a simple example of a 3−dimensional hypercube in which a node $|m\rangle$ is marked. And for simplicity take $l = 1$. The coin operator with a single self-loop is

$$C = \frac{2}{3+l}\begin{pmatrix} 1 & 1 & 1 & \sqrt{l} \\ 1 & 1 & 1 & \sqrt{l} \\ 1 & 1 & 1 & \sqrt{l} \\ \sqrt{l} & \sqrt{l} & \sqrt{l} & l \end{pmatrix} - \begin{pmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} -1 & 1 & 1 & 1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \\ 1 & 1 & 1 & -1 \end{pmatrix} \tag{4.8}$$

and

$$|s_c\rangle = \frac{1}{2}\sum_{i=0}^{3}|i\rangle \equiv \frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \tag{4.9}$$

is the coin state, which is an eigenstate of the coin operator.

At every application of Q' a phase of $e^{\iota\pi}$ is added to the state $|3\rangle \otimes |m\rangle = |\circlearrowleft\rangle \otimes |m\rangle$. As a result, amplitude of the coin state $|3\rangle$ entangled with $|m\rangle$, instead of getting diffused, accumulates at the marked node.

$$CQ'\frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} \otimes |m\rangle = C\frac{1}{2}\begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \end{pmatrix} \otimes |m\rangle = \frac{1}{2}\begin{pmatrix} -0.5+0.5+0.5-0.5 \\ 0.5-0.5+0.5-0.5 \\ 0.5+0.5-0.5-0.5 \\ 0.5+0.5+0.5+0.5 \end{pmatrix} \otimes |m\rangle \tag{4.10}$$

This gives an intuitive idea of why we can search for any marked solution on a graph using Q' with higher self-loop weights. Note that the state $Q'|s_c\rangle$ is not an eigenstate of C.

Fig.4.10 shows search results for different number of marked nodes on a 7−dimensional hypercube.

4.2.1 Circuit implementation

Since we can search with higher values of self-loop weights using Q', a quantum circuit can be easily constructed for $l = 1$. The coin operator is the Grover diffusion operator for a quantum walk on a d-dimensional hypercube with $l = 1$. In this section we will build a quantum circuit for LQW search on 3−dimensional hypercube. Fig.4.11 and 4.12 shows the quantum circuit implementation of the coin and the shift operators, where q_c and q_p represents the coin and the position quantum registers, respectively.

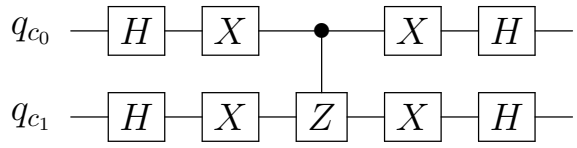


Figure 4.11: Implementation of a 2-qubit Grover diffusion operator as coin for LQW on a 3−dimensional hypercube.

a

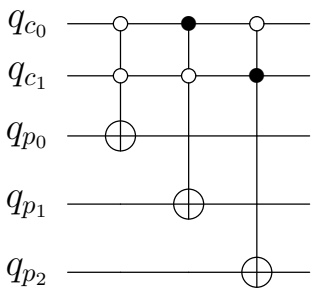


Figure 4.12: Shift operator for LQW on a 3−dimensional hypercube.

In a hypercube, the adjacent nodes have a hamming distance of 1. Therefore, the shift operator performs the bit flip operation on any one of the three qubits of the position register depending on any of the three states (here $|00\rangle$, $|01\rangle$, $|10\rangle$) in the coin subspace. The remaining coin state ($|11\rangle$) does nothing to the walker's position, i.e., it represents a self-loop coin state.

Fig.4.13 shows quantum circuit implementation of a single step of LQW search on a 3−dimensional hypercube. The state is initialized in the uniform superposition of coin-walker states using the Hadamard operators. The query operator Q' marks the node $|11\rangle \otimes |111\rangle$ by adding a phase of $e^{\iota\pi}$. It is followed by the usual coin and shift operators.

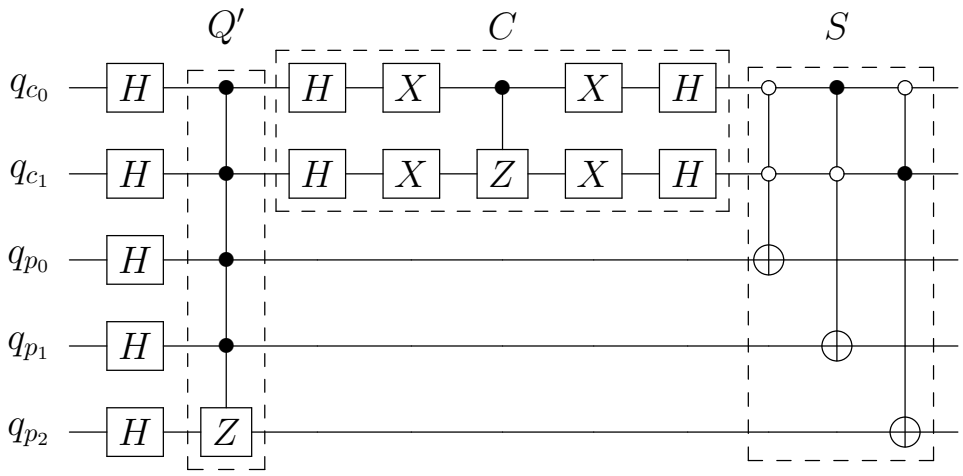


Figure 4.13: Implementation of one step of LQW search on a 3−dimensional hypercube.

We simulate LQW search on a 3−dimensional hypercube using qiskit's qasm simulator. Fig.4.14 shows the probability of the walker states after 5 iterations of $U = S(C \otimes I)Q'$. The walker has high probability ~ 0.8 to be found at the marked node $|111\rangle \in \mathcal{H}_P$. Since this quantum walk itself is an iterative process, implementing the circuit on a real quantum computer does not yield the desired result due to noise.

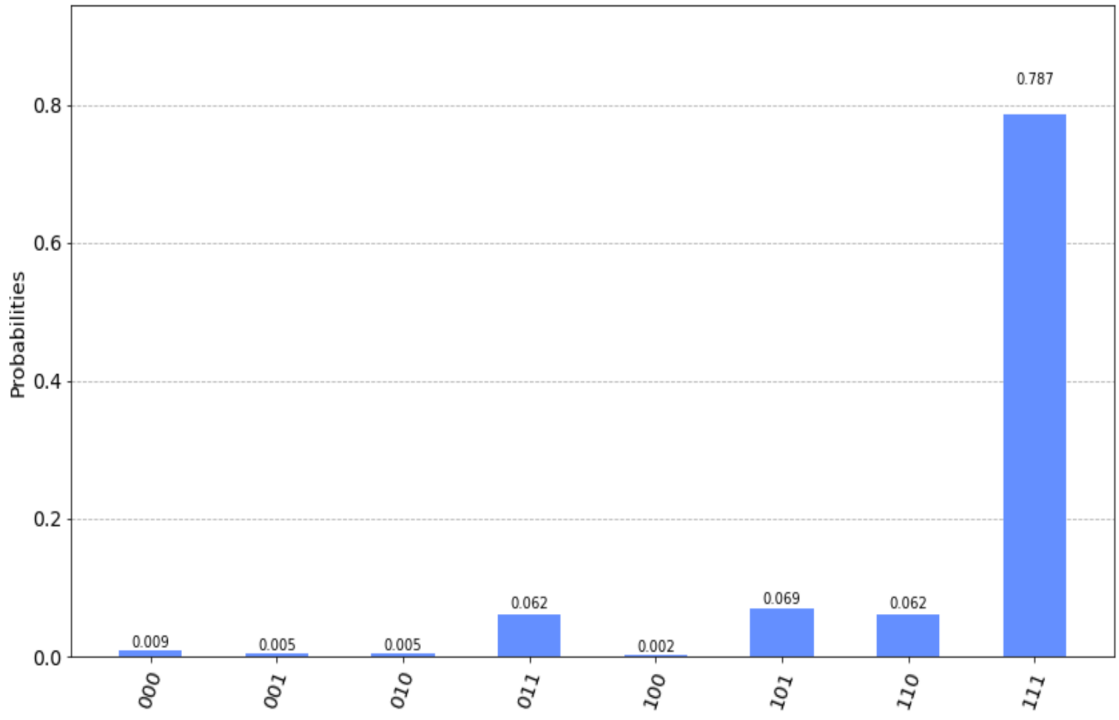


Figure 4.14: Qiskit simulation of LQW search on a 3−dimensional hypercube. Probabilities after 5 iterations of $U = S(C \otimes I)Q'$

In this chapter, we studied the Lackadaisical quantum walk search algorithm first introduced in [20]. The algorithm has some limitations if the query operator $Q = I_N - 2\sum_{m\in M} |m\rangle \langle m|$ is used. Using Q, we cannot: search two adjacent marked nodes, search a node that already has a self-loop, search marked nodes that form a cycle on the graph, control the success probability. We modified Q to $Q' = I_{(d+1)N} - 2\sum_{m\in M} |\circlearrowleft\rangle \langle\circlearrowleft| \otimes |m\rangle \langle m|$ overcoming these limitations. Also, Q' makes it possible to search using higher self-loop values ($l = 1$) for a hypercube, which enabled us to implement the search algorithm on a quantum circuit.

Chapter 5

DTQW with Complex coins

In this chapter, we will use the kicked rotor, a complex system, as a coin for the quantum walk. The study involving dynamical systems like the quantum Baker's map [19] and the Harper map [18] as complex coins showed that in some scenarios, the classically chaotic limit of the coin could lead to thermalization of the quantum walk. Classically, the kicked rotor can be integrable or chaotic depending on the kick strength. Therefore, we will study the effects of the coin's internal dynamics on the quantum walker by observing the evolution of the walker's variance and the coin-walker entanglement. These types of bipartite systems can be useful in studying quantum chaos and the decoherence induced by a chaotic environment in quantum systems.

5.1 Quantum kicked rotor

Kicked rotor [26, 27] Hamiltonian with kicking time period T and kick strength K is

$$H = \frac{\hat{p}^2}{2} + K\cos\hat{x}\sum_{n=0}^{\infty}\delta(t-nT) \tag{5.1}$$

where p is the angular momentum and x is the angle. The p and x of the kicked rotor are taken to be on a torus. We represent angular momentum eigenstates by $\{|m\rangle : -d/2 \leq m \leq d/2-1\}$ and angular eigenstates by $\{|k\rangle : 0 \leq k < d\}$. Therefore,

$$\hat{p}\,|m\rangle = m\hbar_s\,|m\rangle \tag{5.2}$$

and

$$\langle k|m\rangle = \frac{1}{\sqrt{2\pi}} e^{\iota m\theta_k} \ , \quad m = 0, \pm 1, \pm 2... \tag{5.3}$$

where, $\theta_k = \dfrac{2\pi k}{d}$ and $\hbar_s$ is the scaled Planck's constant. We take a large enough basis size for the kicked rotor (i.e., large d) such that the boundary effects do not come into play.

Here, we will assume $T = 1$. Then, the Floquet operator for the kicked rotor is given by

$$U_{kr} = \exp\left[-\iota \frac{\hat{p}^2}{2\hbar_s}\right] \exp\left[-\iota \frac{K}{\hbar_s} \cos\hat{x}\right] \tag{5.4}$$

The matrix elements of the Floquet operator in the angular momentum basis can be written as

$$\begin{aligned}
\langle m| U_{kr} |m'\rangle &= \langle m| \exp\left[-\iota \frac{\hat{p}^2}{2\hbar_s}\right] \exp\left[-\iota \frac{K}{\hbar_s} \cos\hat{x}\right] |m'\rangle \\
&= \sum_{k=0}^{d-1} \langle m| \exp\left[-\iota \frac{\hat{p}^2}{2\hbar_s}\right] |k\rangle \langle k| \exp\left[-\iota \frac{K}{\hbar_s} \cos\hat{x}\right] |m'\rangle \\
&= \exp\left[-\iota \frac{m^2\hbar_s}{2}\right] \sum_{k=0}^{d-1} \langle m|k\rangle \exp\left[-\iota \frac{K}{\hbar_s} \cos\theta_k\right] \langle k|m'\rangle \\
&= \exp\left[-\iota \frac{m^2\hbar_s}{2}\right] \sum_{k=0}^{d-1} \exp\left[-\iota \frac{K}{\hbar_s} \cos\theta_k\right] \exp[\iota(m' - m)\theta_k]
\end{aligned} \tag{5.5}$$

Quantum dynamics of a kicked rotor exhibit dynamical localization in momentum space. Coupling a kicked rotor to a quantum walker may lead to some interesting behavior of the walker.

5.2 Quantum chaotic walk

We consider a quantum walk on a linear lattice having N vertices with periodic boundary conditions [18]. The walker starts at the origin and can hop to the nearest neighbors depending on the coin states, i.e., angular momentum states of the kicked rotor. The Hilbert space for the quantum walk is $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P = \mathbb{C}^d \otimes \mathbb{C}^N$. The unitary operator for the walk is defined as

$$U = (P_- \otimes S_- + P_+ \otimes S_+)(U_{kr} \otimes I_N) \tag{5.6}$$

U_{kr} serves as the coin operator. The shift operator acts as $S_\pm |i\rangle = |i \pm 1\rangle$ on a walker state $|i\rangle$. $P_\pm$ are the projection operators over the angular momentum states of the kicked

rotor and control the application of the shift operator on the walker space. $P_{\pm}$ should be orthogonal and satisfy $P_+ + P_- = I_d$. Here we will use $P_- = \sum_{m=-d/2}^{-1} |m\rangle \langle m| = |0\rangle \langle 0| \otimes I_{d/2}$ and $P_+ = \sum_{m=0}^{d/2-1} |m\rangle \langle m| = |1\rangle \langle 1| \otimes I_{d/2}$.

The system is initialized at the origin $|0\rangle_W$ with zero-momentum of the kicked rotor, i.e., in the state $|\Psi(0)\rangle = |0\rangle_C \otimes |0\rangle_W$. After $t-$time steps, the state of the system is

$$|\Psi(t)\rangle = U^t(|0\rangle_C \otimes |0\rangle_W) \tag{5.7}$$

and the density matrix is $\rho(t) = |\Psi(t)\rangle \langle \Psi(t)|$.

The reduced density matrices of the coin and the walker state can be computed by taking a partial trace.

$$\rho_C(t) = tr_W \rho(t) = Tr_W(|\Psi(t)\rangle \langle \Psi(t)|) \tag{5.8}$$

$$\rho_W(t) = tr_C \rho(t) = Tr_C(|\Psi(t)\rangle \langle \Psi(t)|) \tag{5.9}$$

Therefore, the probability of finding the walker at a vertex $|n\rangle \in \mathcal{H}_{\mathcal{P}}$ can be calculated as,

$$p_n(t) = \langle n| \rho_W(t) |n\rangle \tag{5.10}$$

From the probability distribution of the walker, we can study the evolution of its variance. The variance $\sigma^2(t)$ is

$$\sigma^2(t) = \sum_{n=-d/2}^{d/2-1} n^2 p_n(t) - \left(\sum_{n=-d/2}^{d/2-1} n p_n(t) \right)^2 \tag{5.11}$$

The von Neumann entropy is used to measure the coin-walker entanglement, which is defined for the reduced density matrices as

$$S(t) = -Tr(\rho_W(t) \ln \rho_W(t)) = -Tr(\rho_C(t) \ln \rho_C(t)) \tag{5.12}$$

that can also be written as

$$S(t) = \sum_{\lambda_W \neq 0} \lambda_W \ln \lambda_W \tag{5.13}$$

where the sum is over the non-zero eigenvalues λ_W of $\rho_W(t)$.

Next, we numerically compute the walker's variance and the coin-walker entanglement.

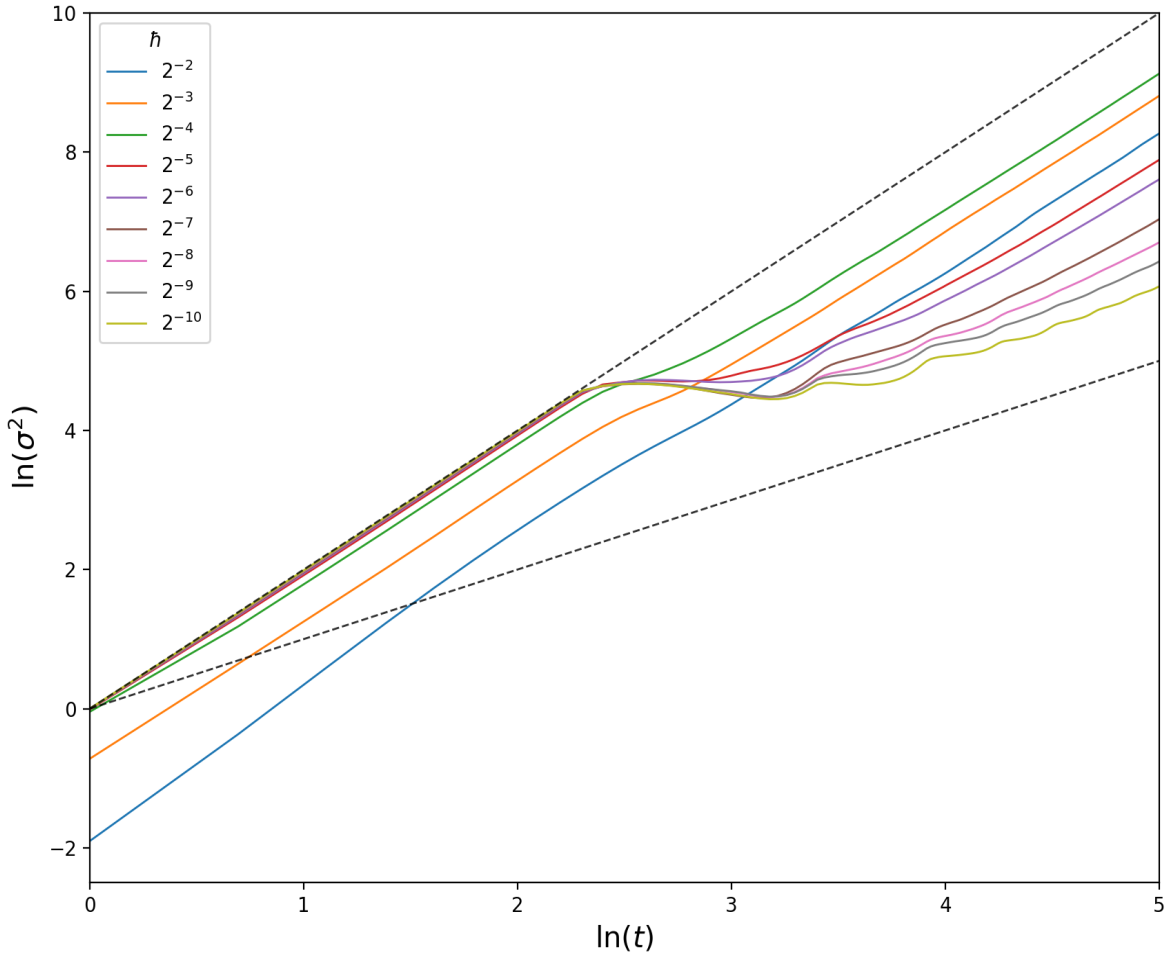


Figure 5.1: Walker's variance as a function of time with $K = 0.1$ for different values of $\hbar_s$. The dashed lines have slopes 1 and 2.

Fig.5.1 shows the walker's variance as a function of time for $K = 0.1$. The quantum walker seems to get localized for a short time during its evolution, and is followed by a diffusive growth for lower values of the scaled Planck's constant $\hbar_s$. This behavior of the quantum walker occurs only at kick strengths $K \lesssim 2$ and lower $\hbar_s$ for $T = 1$. In reality, the walker does not localize completely. There seems to be a combination of ballistic and non-ballistic growths. Furthermore, the coin-walker entanglement entropy grows rapidly during this behavior of the quantum walker. Fig.5.2 shows the behavior of the entropy as a function of time for $K = 0.1$.

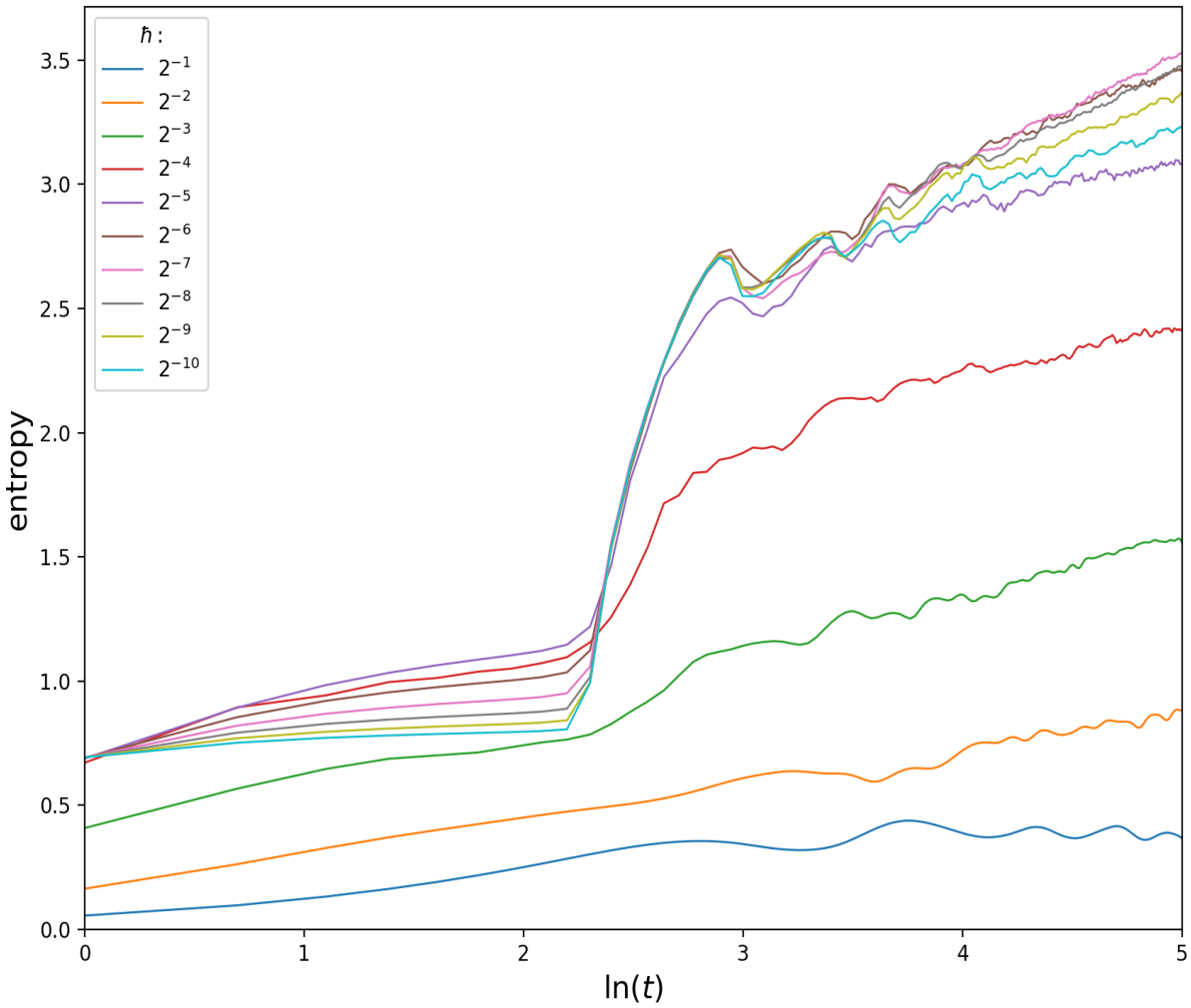


Figure 5.2: Entropy of coin-walker entanglement as a function of time with $K = 0.1$ for different values of $\hbar_s$.

Fig.5.3 and 5.4 shows evolution of the walker's variance and the entanglement entropy for $K = 0.5, 1$. These plots also exhibit similar behavior to the $K = 0.1$ case. Note that the time at which the rapid growth of entropy is observed gets lower for higher kicking strengths.

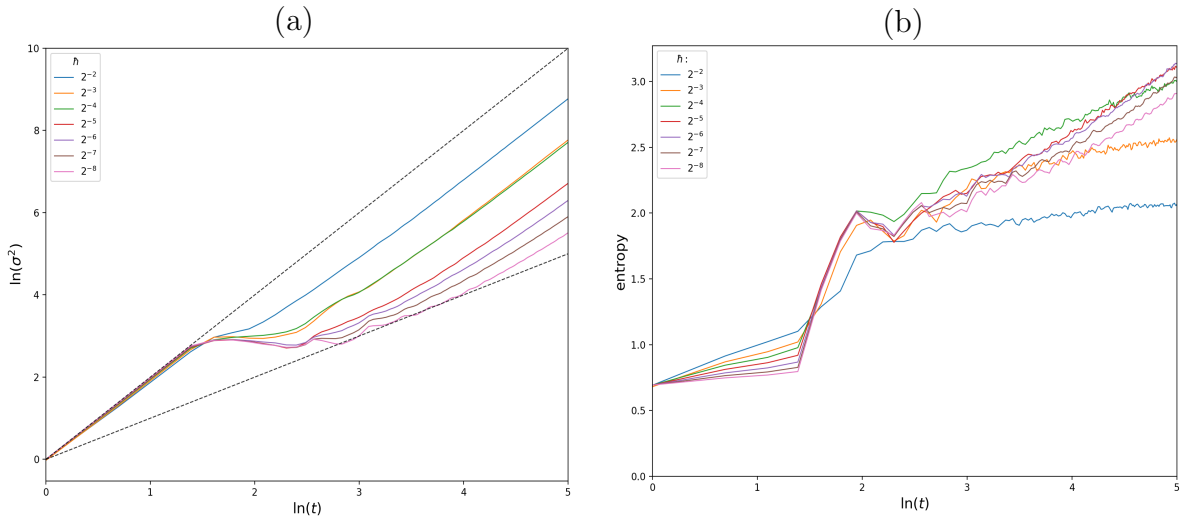


Figure 5.3: Kicked rotor with $K = 0.5$ used as coin. (a) Walker's variance as a function of time. The dashed lines have slopes 1 and 2. (b) Entropy of coin-walker entanglement as a function of time for different values of $\hbar_s$

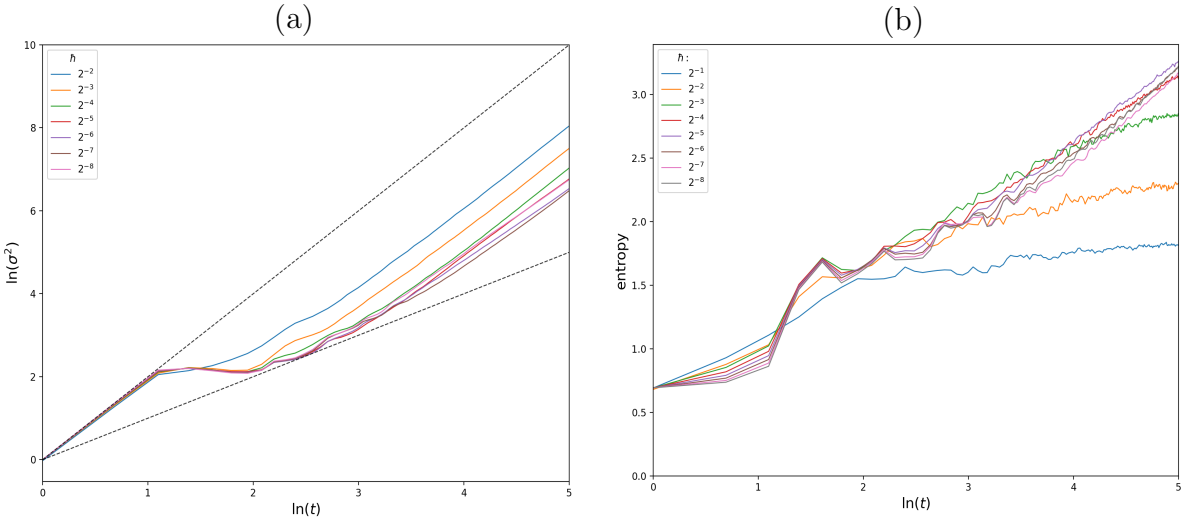


Figure 5.4: Kicked rotor with $K = 1$ used as coin. (a) Walker's variance as a function of time with $K = 1$. The dashed lines have slopes 1 and 2. (b) Entropy of coin-walker entanglement as a function of time for different values of $\hbar_s$

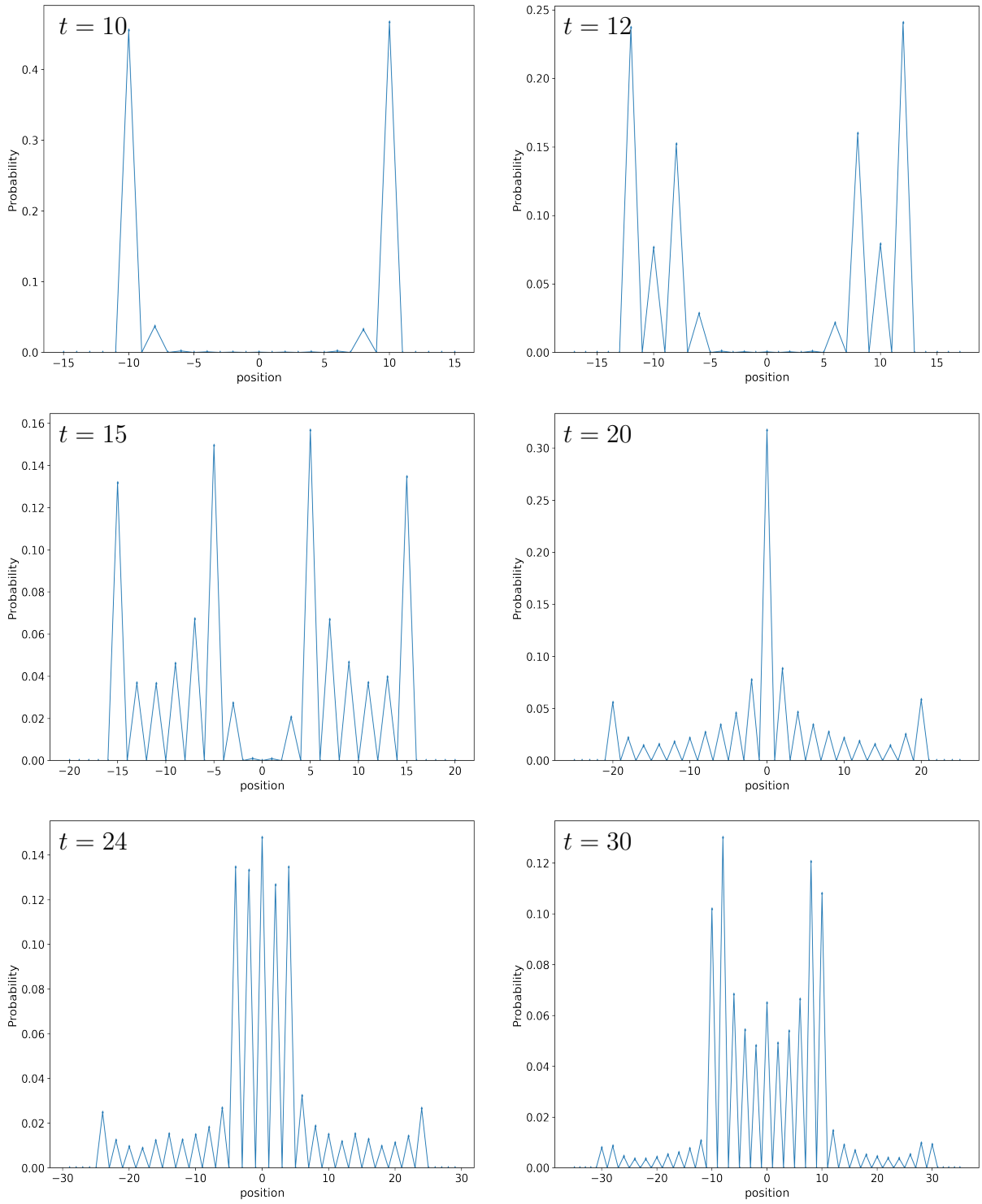


Figure 5.5: Quantum walker's probability distribution as a function of position on the line at various times t. Kicked rotor with $K = 0.1$ and $\hbar_s = 2^{-10}$ used as a coin.

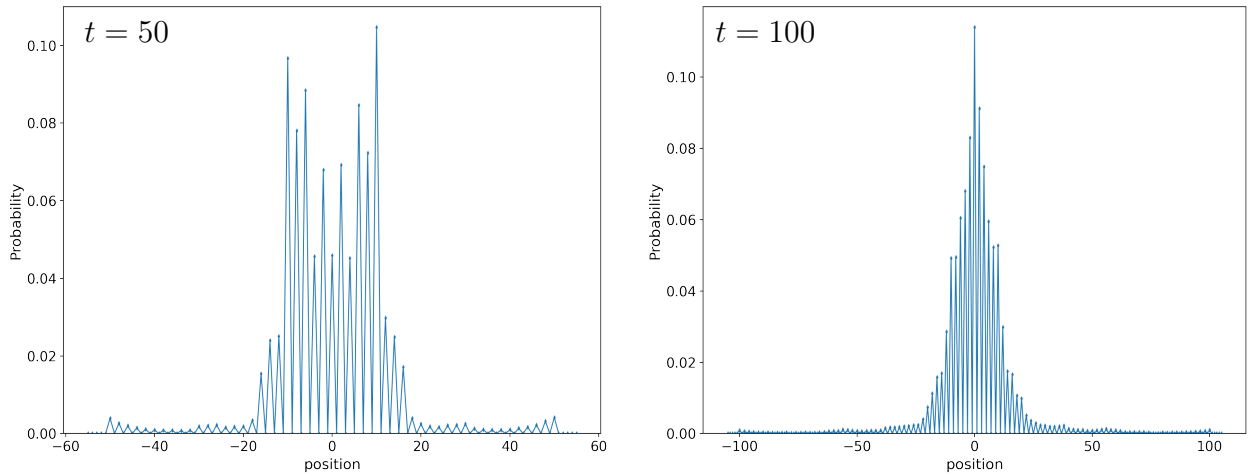


Figure 5.6: Quantum walker's probability distribution as a function of position on the line at longer times $t = 50, 100$. Kicked rotor with $K = 0.1$ and $\hbar_s = 2^{-10}$ used as coin.

Fig.5.5 and 5.6 shows the probability distribution of the quantum walker for $K = 0.1$ with $\hbar_s = 2^{-10}$. Only a small probability amplitude grows in a ballistic manner when the walker seems to be localized. The walker gets localized near the origin with a high probability around $t \sim 20$.

At longer times $t \sim 100$ the quantum walker's distribution start looking like a Gaussian. It may indicate thermalization of the quantum walker.

We can use quantum walker dynamics to study a real physical system with these types of quantum walks. Also, the information shared between the two can be controlled with the help of the projection operators as in Eq.5.6. Here, coarse-grained information is shared, i.e., the angular momentum $\hat{p}$ being greater or lower than 0. Even then, in particular cases (as in the Fig.5.6), it is still possible to see signs of quantum walker's thermalization.

Chapter 6

Summary and Conclusions

In this thesis we studied discrete-time quantum walks with the main focus on lackadaisical quantum walks. We highlighted the contrasts between quantum and classical walks by comparing the walker's probability distribution and hitting times on a linear lattice, hypercube and binary tree. We saw that the quantum walks do not always give an advantage over the classical random walks. Since, for a directed walk on the line, $l < 1$ implies faster classical distribution, whereas $l > 1$ implies faster quantum distribution and is confirmed by the average hitting time. For $l < 1$ classical walker hits faster as compared to the quantum walker.

We modified the lackadaisical quantum walk search algorithm and overcame the limitations due to the query operator of the previously developed algorithm. We proposed a different query operator given as $Q' = I_{(d+1)N} - 2\sum_{m\in M} |\circlearrowleft\rangle \langle\circlearrowleft| \otimes |m\rangle \langle m|$ to be used in the search algorithm. Q' allowed us to search marked nodes on a hypercube with higher self-loop weights ($l \sim 1$), which we then implemented on a quantum circuit and simulated using qiskit.

We studied the quantum-chaotic walks by coupling a quantum kicked rotor as a coin to the quantum walker. The kicked rotor dynamics affect the walker's probability distribution, and the quantum walker seems to get localized with high probability for a short time interval. This behavior occurs for low kick strengths, $K \lesssim 2$ (given $T = 1$), accompanied by the rapid growth of von Neumann entropy which is used to measure coin-walker entanglement. Such a coupling can allow us to study any real physical system by controlling the information shared between the two subsystems.

Bibliography

[1] KARL PEARSON. The problem of the random walk. *Nature*, 72(1865):294–294, Jul 1905.

[2] Frank Spitzer. *Principles of random walk*, volume 34. Springer Science & Business Media, 2001.

[3] Feng Xia, Jiaying Liu, Hansong Nie, Yonghao Fu, Liangtian Wan, and Xiangjie Kong. Random walks: A review of algorithms and applications. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 4(2):95–107, 2020.

[4] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking : Bringing order to the web. In *WWW 1999*, 1999.

[5] Dániel Fogaras and Balázs Rácz. Towards scaling fully personalized pagerank. In Stefano Leonardi, editor, *Algorithms and Models for the Web-Graph*, pages 105–117, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.

[6] T.H. Haveliwala. Topic-sensitive pagerank: a context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

[7] Marina Meilă and Jianbo Shi. A random walks view of spectral segmentation. In Thomas S. Richardson and Tommi S. Jaakkola, editors, *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics*, volume R3 of *Proceedings of Machine Learning Research*, pages 203–208. PMLR, 04–07 Jan 2001. Reissued by PMLR on 31 March 2021.

[8] G. D. Paparo and M. A. Martin-Delgado. Google in a quantum network. *Scientific Reports*, 2(1):444, Jun 2012.

[9] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96*, 1996.

[10] A. Ambainis. Quantum walk algorithm for element distinctness. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31, 2004.

[11] Neil Shenvi, Julia Kempe, and K. Birgitta Whaley. Quantum random-walk search algorithm. *Phys. Rev. A*, 67:052307, May 2003.

[12] Kun Wang, Nan Wu, Ping Xu, and Fangmin Song. One-dimensional lackadaisical quantum walks. *Journal of Physics A: Mathematical and Theoretical*, 50(50):505303, nov 2017.

[13] C. M. Chandrashekar. Discrete-time quantum walk - dynamics and applications. 2010.

[14] Andrew M. Childs. Universal computation by quantum walk. *Phys. Rev. Lett.*, 102:180501, May 2009.

[15] Neil B. Lovett, Sally Cooper, Matthew Everitt, Matthew Trevers, and Viv Kendon. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A*, 81:042330, Apr 2010.

[16] Edward Farhi and Sam Gutmann. Quantum computation and decision trees. *Physical Review A*, 58:915–928, 1998.

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

[18] Sivaprasad Omanakuttan and Arul Lakshminarayan. Quantum walks with quantum chaotic coins: Loschmidt echo, classical limit, and thermalization. *Phys. Rev. E*, 103:012207, Jan 2021.

[19] Leonardo Ermann, Juan Pablo Paz, and Marcos Saraceno. Decoherence induced by a chaotic enviroment: A quantum walker with a complex coin. *Phys. Rev. A*, 73:012302, Jan 2006.

[20] Thomas G Wong. Grover search with lackadaisical quantum walks. *Journal of Physics A: Mathematical and Theoretical*, 48(43):435304, oct 2015.

[21] Nikolajs Nahimovs and Raqueline A M Santos. Lackadaisical quantum walks on 2d grids with multiple marked vertices. *Journal of Physics A: Mathematical and Theoretical*, 54(41):415301, sep 2021.

[22] Hari Krovi and Todd A. Brun. Hitting time for quantum walks on the hypercube. *Phys. Rev. A*, 73:032341, Mar 2006.

[23] Julia Kempe. Quantum random walks hit exponentially faster. *ArXiv*, quant-ph/0205083, 2002.

[24] Luciano S. de Souza, Jonathan H. A. de Carvalho, and Tiago A. E. Ferreira. Lackadaisical quantum walk in the hypercube to search for multiple marked vertices. In André Britto and Karina Valdivia Delgado, editors, *Intelligent Systems*, pages 249–263, Cham, 2021. Springer International Publishing.

[25] Mason L. Rhodes and Thomas G. Wong. Search on vertex-transitive graphs by lack-adaisical quantum walk. *Quantum Information Processing*, 19(9):334, Sep 2020.

[26] Giulio Casati and Joseph Ford. Stochastic behavior in classical and quantum hamiltonian systems : Volta memorial conference, como, 1977. 1979.

[27] M.S. Santhanam, Sanku Paul, and J. Bharathi Kannan. Quantum kicked rotor and its variants: Chaos, localization and beyond. *Physics Reports*, 956:1–87, 2022. Quantum kicked rotor and its variants: Chaos, localization and beyond.