IMPLEMENTATION OF HITTING TIMES OF DISCRETE TIME QUANTUM RANDOM WALKS ON CUBELIKE GRAPHS.

JAIDEEP MULHERKAR, RISHIKANT RAJDEEPAK, AND V SUNITHA

Abstract. We demonstrate an implementation of the hitting time of a discrete time quantum random walk on cubelike graphs using IBM’s Qiskit platform. Our implementation is based on efficient circuits for the Grover and Shift operators. We verify the results about the one-shot hitting time of quantum walks on a hypercube as proved in [17]. We extend the study to another family of cubelike graphs called the augmented cubes [11]. Based on our numerical study, we conjecture that for all families of cubelike graphs there is a linear relationship between the degree of a cubelike graph and its hitting time which holds asymptotically. That is, for any cubelike graph of degree $\Delta$, the probability of finding the quantum random walk at the target node at time $\frac{\pi \Delta}{2}$ approaches 1 as the degree $\Delta$ of the cubelike graph approaches infinity.

Keywords. Quantum Random Walks, Cubelike graphs, Hitting times

1. Introduction

Classical random walks have found applications in many areas such as computer science, physics, finance and engineering. In particular, classical random walks have been used in the development of faster algorithms for estimating the volume of a convex body, finding a satisfying assignment to a Boolean formula (2-SAT problem) and finding if two vertices of an undirected graph are connected [21]. Quantum random walks [16, 23, 25] are quantum mechanical analogues of classical random walks. The first quantum random walk models were proposed in [5, 12]. It has since been observed that there are some startling differences between classical and quantum walks. For instance, it has been shown that the hitting time from one vertex to the antipodal vertex is linear in the dimension of the hypercube, which is exponentially fast in quantum walks compared to classical walks [17]. Moreover, there is also a speed up obtained in the mixing time of quantum versus the classical walks [22]. These differences between classical and quantum walks led to search for quantum walk based algorithms that can outperform classical counterparts. Some of the quantum algorithms that have been developed based on quantum walks are searching a marked element on a grid, the element distinctness problem and the triangle finding problem in a graph [6]. Quantum walks have also been shown to be universal for quantum computation [10, 20].

In quantum random walks there is a quantum system such as a spin system on a graph and the evolution is given by a unitary operator acting on a Hilbert space of the graph. Generally speaking quantum walks are of two types: Discrete and
Continuous. In the discrete time quantum walk the associated Hilbert space is a joint system of a coin space and position space. For a regular graph the coin space has dimension equal to the number of edges per node and the dimension of the position space is equal to the number of vertices in the graph. The discrete time quantum walk starts with an initial state of the coin and position. Each time step in the evolution of the discrete random walk consists of the following: 1. A coin operator acts on the coin space changing the original coin state to a new state 2. A Shift operator shifts the current position of the random walker conditioned on the coin state. This step is then repeated many times thus evolving the random walker over the state space of the graph. After a certain number of time steps a measurement is done of the position space. The continuous random walks are the evolution of a quantum system on a graph according to the unitary \(e^{iAt}\) where \(A\) is the adjacency matrix of the graph. A quantity of interest in quantum walks is the hitting time for a pair of vertices. For a pair of vertices \((u, v)\), if starting from vertex \(u\), after a certain number of time steps \(T\) of the walker is observed to be in vertex \(v\) with probability \(p\) then \((T, p)\) is called the hitting time for vertex \(v\) starting from vertex \(u\). The hitting times are based on the graph topology and faster hitting times could mean speed up algorithms. Quantum walks have been studied on a variety of graphs such as cycles, trees, and cubelike graphs and general Cayley graphs [4, 3]. A related question is of perfect state transfer between two vertices, which has been studied in [14]. Apart from the results in [17] in hypercubes, cubelike graphs have been studied in the context of continuous time random walks. Criterion for the perfect state transfer on cubelike graphs have been given in [4, 9]. In this paper we restrict our study to the hitting times of discrete time quantum walks on cubelike graphs.

Because of their universality and potential use in developing algorithms there is a need for good physical implementations of quantum random walks. Physical implementations based on ion-traps, optical cavities and optical lattices has been suggested in [18, 24, 27]. Circuit implementations of various standard quantum algorithms using IBM’s Qiskit platform was done in [15]. Recently in [2] implementation of staggered quantum walks on cycles, two-dimensional lattices and complete graph was studied on IBM quantum computers. A comparison on two different implementation approaches of quantum walks is given in [13]. An implementation of the discrete time quantum walk on hypercubes, complete graphs, complete bipartite graphs and 2-d lattice is recently given in [20]. Our work is in the spirit of implementation of QRW’s for NISQ (Nosiy Intermediate Scale Quantum computing) era computers. Our main contributions in this paper are 1) An implementation of efficient circuits for the QRW on a cubelike graph based on decompositions of the Grover and Shift operator. 2) An application of these circuits to find the hitting times in cubelike graphs. Here we implement our circuits on the IBM’s Qiskit platform and verify the one-shot hitting times in families of cubelike graphs such as hypercubes and augmented cubes.

This paper is organized as follows: In section 2 we describe the construction of the discrete time quantum random walk based on the Grover coin operator and the Shift operator on a cubelike graph. In section 3 we define the one-shot hitting time of quantum random walks and review the past results. In section 4 we describe our implementation of the efficient circuits for the QRW and finally in section 5
of the system is described by a unitary operator $U$ defined by

$$|\text{state}\rangle = \{C|\text{state}\rangle\}.$$ 

If $H$ space, then the associated quantum system in a Hilbert space. If $H$ space, then the associated quantum system in a Hilbert space. If $H$ space, then the associated quantum system in a Hilbert space.

$2.1$. The quantum walk. A discrete-time quantum walk (DTQW) on a graph $\Gamma$ is described by the evolution of an associated quantum system in a Hilbert space. If $\Gamma$ is $\Delta$-regular with $N$ vertices, then the associated Hilbert space is the joint Hilbert space $\mathcal{H} = \mathcal{H}_C \otimes \mathcal{H}_P$, where $\mathcal{H}_C \cong \mathbb{C}^\Delta$ is known as the coin space, and $\mathcal{H}_P \cong \mathbb{C}^N$ is known as the position space. Suppose vertices and edges in $\Gamma$ are labeled by $\{v_0, v_1, \ldots, v_{N-1}\}$ and $\{\alpha_0, \alpha_1, \ldots, \alpha_{\Delta-1}\}$, respectively, then the computational basis of $\mathcal{H}$ is given by

$$\{|\alpha_k\rangle |v_a\rangle : 0 \leq k \leq \Delta - 1, 0 \leq a \leq N - 1\}.$$

This association between $\Gamma$ and $\mathcal{H}$ is such that vertex $v_a$ represents the position state $|v_a\rangle$ in $\mathcal{H}_P$, and edge $\alpha_k$ represents the coin state $|\alpha_k\rangle$ in $\mathcal{H}_C$. The evolution of the system is described by a unitary operator $U$, called the evolution operator, defined by

$$U = S(C \otimes I),$$

where, $C$ is the coin operator analogous to $\Delta$-dimensional classical coin and $S$ is the shift operator that shifts the quantum state $|\alpha_k\rangle |v_a\rangle$ to the quantum state $|\alpha_k\rangle |v_b\rangle$ such that $\alpha_k$ is the label of the edge $(v_a, v_b)$.

$2.2$. Cubelike graphs. A Cayley graph is a graph defined over a pair $(G, \Omega)$, where $G$ is a finite group and $\Omega$ is a generating set of $G$, with the properties:

1. $\Omega$ does not contain the identity element, and
2. $\Omega$ is closed under the group
inverse, i.e., $x^{-1} \in \Omega$ for all $x \in \Omega$. The Cayley graph, denoted by $\text{Cay}(G, \Omega)$, is a graph whose vertices are identified with the elements of $G$ and the edge set is $\{(x, y) : xy^{-1} \in \Omega\}$. Clearly, $\text{Cay}(G, \Omega)$ is a regular graph on $N = |G|$ vertices with regularity $\Delta = |\Omega|$. The Cayley graph $\text{Cay}(\mathbb{Z}_2^n, \Omega)$ defined over the Boolean group $\mathbb{Z}_2^n$ is called cubelike graph of dimension $n$. For a cubelike graph $\text{Cay}(\mathbb{Z}_2^n, \Omega)$ when the ordering on $\Omega$ is not specified then we assume the lexicographical ordering, i.e., we assume that the cubelike graph is given in its canonical form. A vertex $(x_{n-1}, \ldots, x_1, x_0) \in \mathbb{Z}_2^n$ is denoted by the binary string $x_{n-1} \cdots x_1 x_0$, and is given the label $v_a$, where $a = x_{n-1}2^{n-1} + \cdots + x_1 2^1 + x_0 2^0$. An edge $(v_a, v_b)$ is labeled by $\alpha_{k-1}$, $1 \leq k \leq \Delta$, if the XOR operation of $v_a$ and $v_b$, i.e., $v_a \oplus v_b$, is the $k$-th element $\Omega(k)$ of $\Omega$. Suppose $2^{m-1} \leq \Delta \leq 2^m$, for some integer $m$, then the edge labeled by $\alpha_{k-1}$ can be denoted by the $k$-th binary string $y_{m-1} \cdots y_1 y_0$ in $\mathbb{Z}_2^m$, i.e., $k - 1 = y_{m-1}2^{m-1} + \cdots + y_1 2^1 + y_0 2^0$.

Some special subfamily of cubelike graphs of interest are (a) Hypercubes $Q_n$ with $\Omega = \{0^i 1^j : i + j = n - 1\}$, and (b) Augmented cubes $AQN_n$ with $\Omega = \{0^i 1^j : i + j = n - 1\} \cup \{0^n - 1^i : 1 \leq i \leq n\}$. In Figure 1(a) edge $(v_0, v_4)$ is labeled by $\alpha_2$ because $v_0 \oplus v_4 = 100$ is the third element of its generating set $\{001, 010, 100\}$, while in Figure 1(b) it is labeled by $\alpha_3$ as 101 is the fourth element of its generating set $\{001, 010, 011, 100, 111\}$. The binary representations of $\alpha_0$, $\alpha_1$, $\alpha_2$ in $Q_3$ (Figure 1a) are 00, 01, 10, respectively, while in $AQN_3$ (Figure 1b), the binary representations of $\alpha_0$, $\alpha_1$, $\alpha_2$, $\alpha_3$, $\alpha_4$ are 000, 001, 010, 011, 100, respectively.

Let $U = S(C \otimes I)$ be the evolution operator for DTWQ on $\text{Cay}(\mathbb{Z}_2^n, \Omega)$, with $\Delta = |\Omega|$. Then, the shift operator $S$ is defined by

\begin{equation}
S = \sum_{k=1}^{\Delta} \sum_{a=0}^{2^{n-1}-1} |\alpha_{k-1}, v_a \oplus \Omega(k)\rangle \langle \alpha_{k-1}, v_a|
\end{equation}

Any unitary operator on $\mathbb{C}^\Delta$ can be chosen as the coin operator. A popular coin operator is the Grover coin $C$ which is defined by $2 |D\rangle \langle D| - I$, where,$\begin{align*}
|D\rangle &= \frac{1}{\sqrt{\Delta}} \sum_{k=1}^{\Delta} |\alpha_{k-1}\rangle \implies C &= \begin{bmatrix}
\frac{2}{\Delta} - 1 & \frac{2}{\Delta} & \frac{2}{\Delta} & \cdots & \frac{2}{\Delta} \\
\frac{2}{\Delta} & \frac{2}{\Delta} - 1 & \frac{2}{\Delta} & \cdots & \frac{2}{\Delta} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\frac{2}{\Delta} & \frac{2}{\Delta} & \frac{2}{\Delta} & \cdots & \frac{2}{\Delta} - 1
\end{bmatrix}.
\end{align*}$

In the following example DTQW is illustrated on $C_4$, a cycle on four vertices.

**Example 1** (Discrete-time quantum walk on $\text{Cay}(\mathbb{Z}_2^2, \{01, 10\})$). The Hilbert space associated with the graph is $\mathbb{C}^2 \otimes \mathbb{C}^4$, and the computational basis is $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle \}$. The evolution operator $U$ is $S(C \otimes I)$, where,$\begin{align*}
C &= \begin{bmatrix}
0 & 1 \\
1 & 0
\end{bmatrix}.
\end{align*}$
and,

\[ S = \sum_{k=1}^{2} \sum_{a=0}^{3} |\alpha_{k-1} \rangle \langle v_a \oplus \Omega(k)| (\alpha_{k-1} \rangle \langle v_a|). \]

The evolution, with the initial state \(|D \rangle \langle 00| = \frac{|0\rangle + |1\rangle}{\sqrt{2}} \langle 00|\), occurs as:

1. The first time step;

\[ U |D \rangle \langle 00| = S(C \otimes I) \left( \frac{1}{\sqrt{2}} (|0\rangle \langle 00| + |1\rangle \langle 00|) \right) \]

\[ = \frac{1}{\sqrt{2}} (|1\rangle \langle 10| + |0\rangle \langle 01|). \]

2. The second time step;

\[ U^2 |D \rangle \langle 00| = S \frac{1}{\sqrt{2}} \left[ \left(\frac{2}{\sqrt{2}} \frac{|0\rangle + |1\rangle}{\sqrt{2}} - |1\rangle \right) \langle 10| \right. \]

\[ + \left. \left(\frac{2}{\sqrt{2}} \frac{|0\rangle + |1\rangle}{\sqrt{2}} - |0\rangle \right) \langle 01| \right] \]

\[ = S \frac{1}{\sqrt{2}} (|0\rangle \langle 10| + |1\rangle \langle 01|) = \frac{|0\rangle \langle 11| + |1\rangle \langle 11|}{\sqrt{2}} \]

\[ = |D \rangle \langle 11|. \]

Consequently, we get the probability distribution of the quantum states. After the first step, the quantum system is at state \(|01\rangle\) or \(|10\rangle\) with probability \(\frac{1}{2}\), and after two steps the probability of the quantum system to be at \(|11\rangle\) is 1.

3. Hitting times of quantum walks

The DTQW starts from a particular starting vertex \(v_0\) and evolves under the dynamics of the evolution operator given in equations (3) and (4). Letting the system evolve for a finite number of steps \(T\) is equivalent to applying \(U^T\) to an initial state \(|v_0\rangle\) of the system. A measurement is then performed in the position space basis to see if the walker is found in a specific target vertex. At time \(T\), if we find the walker in the state \(|v\rangle\) corresponding to the target vertex \(v\) with probability at least \(p\), then we say \((T, p)\) is the quantum hitting time for \(v\), starting from vertex \(v_0\). This hitting time was called the one-shot hitting time defined by Kempe in [17]. More formally,

**Definition 1** ([17]). The discrete-time quantum walk is said to admit \((T, p)\) one-shot \((|v_0\rangle, |v\rangle)\) hitting time if \(|\langle v|U^T|v_0\rangle|^2 \geq p\).

It was shown in [17] that the one-shot hitting time of a quantum random walk on a hypercube is exponentially faster than its classical analogue. We state the following result from [17].

**Theorem 1** ([17]). The discrete-time quantum walk with Grover coin on the hypercube of dimension \(n\) has a \((T, p)\) one-shot \((|x\rangle, |\bar{x}\rangle)\) hitting time, i.e. \(\left| \langle \bar{x}|U^T|x\rangle \right|^2 \geq p\), where \(T\) is either \(\left\lfloor \frac{\pi}{2} n \right\rfloor\) or \(\left\lceil \frac{\pi}{2} n \right\rceil\) and \(p = 1 - O\left(\frac{\log^3 n}{n}\right)\).
For example, $Q_2$ has $(2,1)$ one-shot $(|00\rangle, |11\rangle)$ hitting time. The quantum circuit for the DTWQ on $Q_2$ is shown in Figure 2. The first two qubits represent the position state which is initially at zero state $|00\rangle$. The third qubit represents the coin state, and is initialized to the diagonal state by applying the Hadamard gate $H$. The composition of the next three operators, namely, two CNOT-gates and one NOT-gate, forms the evolution operator $U$. The evolution operator $U$ is applied twice, which are separated by a dotted vertical line called the barrier. The following calculations show that the circuit (Figure 2) is indeed a DTQW on $Q_2$.

We use $X(a)$ to denote the X-gate that flips the value of the qubit $a$, and $C_X(a,b)$ to denote the CNOT-gate with control qubit $a$ and target qubit $b$.

1. The system is initialized to $(H \otimes I \otimes I) |00\rangle$, where $I$ is the identity operator. Since,
\[
(H \otimes I \otimes I) |00\rangle = \frac{|000\rangle + |100\rangle}{\sqrt{2}} = |D\rangle |00\rangle,
\]
the system is thus initialized to $|D\rangle |00\rangle$.

2. The application of $U$ to this initial state evolves the system to the quantum state $\frac{|110\rangle + |001\rangle}{\sqrt{2}}$ as shown below.
\[
\frac{|000\rangle + |100\rangle}{\sqrt{2}} \xrightarrow{C_X(2,0)} \frac{|000\rangle + |101\rangle}{\sqrt{2}} \xrightarrow{X(2)} \frac{|100\rangle + |001\rangle}{\sqrt{2}} \xrightarrow{C_X(2,1)} |110\rangle + |001\rangle,
\]

3. By applying $U$ to the current state, we see that the system evolves to the state $|D\rangle |11\rangle$, as shown below.
\[
\frac{|110\rangle + |001\rangle}{\sqrt{2}} \xrightarrow{C_X(2,0)} \frac{|111\rangle + |001\rangle}{\sqrt{2}} \xrightarrow{X(2)} \frac{|011\rangle + |101\rangle}{\sqrt{2}} \xrightarrow{C_X(2,1)} \frac{|011\rangle + |111\rangle}{\sqrt{2}} = |D\rangle |11\rangle.
\]

Now, the measurement operator outputs $|11\rangle$ with probability 1. If the measurement was done after the first application of $U$ and then after the second application, then, we would have obtained either $|00\rangle$ or $|11\rangle$ with probability $\frac{1}{2}$. 
4. The Quantum Circuit for DTQW on Cubelike Graphs

A general quantum circuit representing the DTQW on $Cay(\mathbb{Z}_2^2, \Omega)$, with $\Delta = |\Omega|$, is depicted in Figure 3. A unitary operator that can be appended to a quantum circuit has dimension $2^t$, for some positive integer $t$. Since the coin operator acting on the coin space is of dimension $\Delta$, we study the case where $\Delta = 2^m$, for some positive integer $m$. All the qubits are initialized to state $|0\rangle$. The position state is represented by the first $n$ qubits, which is initially $|v_0\rangle = |0\rangle^\otimes n$. The coin state, represented by the last $m$ qubits, is initialized to diagonal state $|D\rangle = \frac{1}{\sqrt{\Delta}} \sum_{k=1}^{\Delta} |\alpha_k\rangle$ by applying the Hadamard gate $H$ to each coin qubit. After the initialization, the evolution operator $U$, which is the composition of the Grover operator $C$ and the shift operator $S$, is applied to the circuit for a specific number of times before the measurement is taken. It is important to decompose the evolution operator to run the circuit more efficiently.

![Quantum circuit for DTQW on cubelike graph](image)

**Figure 3.** Quantum circuit for DTQW on cubelike graph. $H$ is applied to each coin qubit to initialize the coin state to $|D\rangle$. The evolution operator $U = S(C \otimes I)$ is applied for a finite number of times. Each application of $U$ is separated by a barrier. Finally, measurement operators are applied to the position qubits.

4.1. Decomposition of the Evolution Operator $U$.

4.1.1. Decomposition of the Grover Operator. The decomposition of the Grover operator $C$ has been discussed in [7, 19] (see Figure 4), which is;

$$C = H^\otimes m X^\otimes m H \otimes I^\otimes m-1 C_X^{(m-1)}([0, 1, \ldots, m - 2, m - 1]H \otimes I^\otimes m - 1 X^\otimes m H^\otimes m,$$

where, $C_X^{(m-1)}(c, t)$ is the generalized Toffoli gate with a set of $m - 1$ control qubits, denoted by $c$, and a target qubit $t$, i.e., it flips the value of the target only if each qubit in $c$ is at state $|1\rangle$. 


4.1.2. Decomposition of the Shift operator. The shift operator $S$ can be expressed as

$$S = \sum_{k=1}^{d} S_{k-1}, \text{ where } S_{k-1} = \sum_{i=0}^{2^n-1} |\alpha_{k-1}\rangle \langle v_i| \Omega(k)\langle \alpha_{k-1}| \langle v_i|. $$

The operator $S_{k-1}$ shifts the walker only along the edge $\alpha_{k-1}$. Suppose $\Omega(k)$ has non-zero entries at positions $p_1, p_2, \ldots, p_l$, then we append the extended Toffoli gates $C_X^m(c, p_j, 1), 1 \leq j \leq l$, to the quantum circuit corresponding to $S_{k-1}$, where, $c$ is the control consisting of all coin qubits and $p_j$-th position qubit is the target. These gates do not flip the values of the target qubits unless the coin state $|\alpha_{k-1}\rangle$ is $|1^m\rangle$. Therefore, we need to first apply X gate to each coin qubit with value zero in $|\alpha_{k-1}\rangle$ and then apply the extended Toffoli gates. Thus, the sequence of gates appended to the quantum circuit corresponding to $S_{k-1}$ is given by

$$\tilde{S}_{k-1} = (f(k))(C_X^m(c, p_l, 1) \ldots C_X^m(c, p_2, 1)C_X^m(c, p_1, 1)), $$

where, $f(k)$ is an operator equivalent to appended X gates that transforms $|\alpha_{k-1}\rangle$ to $|1^m\rangle$. Thus, the sequence of operators appended to the quantum circuit corresponding to $S$ is

$$S \equiv \tilde{S}_0\tilde{S}_1 \cdots \tilde{S}_{d-1}. $$

where, $S_{k-1}, 1 \leq k \leq d$, is expressed by equation (6).

**Example 2.** We can understand the decomposition of the shift operator through an example of DTQW on $Cay(\mathbb{Z}_2^3, \Omega)$, with $\Omega = \{0101, 0111, 1001, 1010\}$. Suppose the walker is at vertex $v = 1101$, then the shift operation at position state $|v\rangle = |1101\rangle$ is described as;

$$S |\alpha_0\rangle |1101\rangle = |\alpha_0\rangle |1101 \oplus 0101\rangle = |\alpha_0\rangle |1000\rangle $$
$$S |\alpha_1\rangle |1101\rangle = |\alpha_1\rangle |1101 \oplus 0111\rangle = |\alpha_1\rangle |1010\rangle $$
$$S |\alpha_2\rangle |1101\rangle = |\alpha_2\rangle |1101 \oplus 0101\rangle = |\alpha_2\rangle |0100\rangle $$
$$S |\alpha_3\rangle |1101\rangle = |\alpha_3\rangle |1101 \oplus 1010\rangle = |\alpha_3\rangle |0110\rangle. $$

The shift operator, mentioned in Equation (8), can be decomposed as shown in Figure 4. For each edge $\alpha_{k-1}, 1 \leq k \leq 4$, we transform $|\alpha_{k-1}\rangle$ to $|\alpha_3\rangle \equiv |11\rangle$ by applying NOT gates, and then apply Toffoli gates to flip the value of position qubits corresponding to non-zero entries in $\Omega(k)$. See Table 3 where we have illustrated...
Figure 5. The quantum circuit for the shift operator used in DTQW on $\text{Cay}(\mathbb{Z}_2^4, \{0101, 0111, 1001, 1010\})$.

Table 1. Transforming each coin state $|\alpha_{k-1}\rangle$, $1 \leq k \leq 4$, to $|11\rangle$ by applying NOT gates.

The transformation of coin qubits. The shift along each direction is given by:

$$
S_0 = (X \otimes X)(C_X^{(2)}([4, 5], 0)C_X^{(2)}([4, 5], 2))
$$

$$
S_1 = (I \otimes X)(C_X^{(2)}([4, 5], 0)C_X^{(2)}([4, 5], 1)C_X^{(2)}([4, 5], 2))
$$

$$
S_2 = (X \otimes X)(C_X^{(2)}([4, 5], 0)C_X^{(2)}([4, 5], 3))
$$

$$
S_3 = (I \otimes X)(C_X^{(2)}([4, 5], 1)C_X^{(2)}([4, 5], 3))
$$

We now discuss how we keep track of changes made to each coin state while implementing the shift operation. For $y \in \mathbb{Z}_2^m$, define permutations $P_y$ by

$$
P_y(x) = y \oplus x, \forall x \in \mathbb{Z}_2^m.
$$

We construct a sequence of binary strings $B(\alpha_{k-1}) = 0^{m-r_k}1^{r_k}$, $1 \leq k \leq 2^m$, where $m-r_k$ is the position of the last non-zero bit of the $k$-th binary string $\alpha_{k-1}$ in $\mathbb{Z}_2^m$. Note that $B(0^m) = 1^m$.

Claim:

$$
P_{B(\alpha_{k-1})}(P_{B(\alpha_{k-2})}(\cdots (P_{B(\alpha_0)}(\alpha_{k-1}) \cdots )) = 1^m.
$$

If the claim is true, then $f(k) = I^{m-r_k}X^{r_k}$, $1 \leq k \leq 2^m$, transforms each coin state $|\alpha_{k-1}\rangle$ to $|1^m\rangle$, one at a time and therefore the operator given by Equation 6 is correct.

We now prove the claim by induction on $k$. For $k = 1$, we have $\alpha_0 = 0^m$ and $B(\alpha_0) = 1^m$. Thus,

$$
P_{B(\alpha_0)}(\alpha_0) = 1^m \oplus 0^m = 1^m.
$$
Assume that at $k$-th iteration, $\alpha_{k-1}$ gets transformed to $1^m$. Since $\alpha_{k-1}$ and $\alpha_k$ differ at the last $r$ positions, for some $1 \leq r \leq m$, $\alpha_k$ gets transformed to $1^{m-r}0^r$ at $k$-th iteration. Therefore, $B(\alpha_k) = 0^{m-r}1^r$ and at $(k+1)$-th iteration we get $B(\alpha_k) \oplus \alpha_k = 1^m$. Hence, the claim is True.

Notice that, the transformation of a coin state corresponding to an edge alters other coin states corresponding to other edges. At the end of the shift operation in Example 2 we retrieve the original generic coin state as shown in the last column of Table I. Indeed, after the $2^m$-th iterations, notice that the $i$-bit of $\alpha_{k-1}$, $1 \leq i \leq m$ and $1 \leq k \leq 2^m$, flips only if $B(\alpha_{j-1})$, $1 \leq j \leq 2^m$, is of the form $0^{m-s}1^s$, with $s \geq i$, which corresponds to a binary string $e_{j-1}$ of the form $x_{m-1} \cdots x_0$. Since the number of such binary strings are $2^{m-i}$, the $i$-th bit of $\alpha_{k-1}$ flips even number of times. Therefore, each coin state $|\alpha_{k-1}\rangle$, $1 \leq k \leq 2^m$, remains unchanged after the $2^m$-th iterations. Therefore, the shift operation is performed successfully without altering the coin state.

4.1.3. Analysis of the quantum circuit for the shift operator. The number of $X$ gates $Num(X)$ used in the decomposition of the shift operator is equal to

$$Num(X) = \sum_{x \in \mathbb{Z}_2^m} wt(B(x)) = 2^{m+1} - 2,$$

where $wt(B(x))$ is the Hamming weight of $B(x)$. This can be proved by induction on $m$. For $m = 1$, $Num(X) = 2$ because $B(0) = 1$ and $B(1) = 1$. Assume by way of induction that, for $m = k$, $Num(X) = 2^{k+1} - 1$. Then, for $m = k + 1$, notice that if $x \in \mathbb{Z}_2^m$ then $0x, 1x \in \mathbb{Z}_2^{m+1}$ and,

$$B(bx) = \begin{cases} B(x), & \text{if } x \neq 0^m \\ B(x) + 1, & \text{if } x = 0^m. \end{cases}$$

where $b = 0$ or $1$. Therefore, $Num(X)$ for $m = k + 1$ is equal to

$$2 \times (2^{k+1} - 2) + 2 = 2^{k+2} - 2.$$ 

$Num(X)$ is of order $O(|\Omega|)$. The number of generalized Toffoli gates used is equal to the sum of Hamming weights of all elements of the generating set $\Omega$, which is

$$Num(C_X^{(m)}) = \sum_{x \in \Omega} wt(x), \quad wt(x) = \text{number of 1’s in } x.$$

4.2. Quantum circuit for cubelike graph of arbitrary degree. If the degree of the cubelike graph is not a power of 2 (for example $Q_3$) then we have to make certain modifications to implement our circuit. Consider a cubelike graph $Cay(\mathbb{Z}_2^n, \Omega)$, with $\Delta = |\Omega|$. Let $m$ be a positive integer satisfying $2^{m-1} < \Delta \leq 2^m$. The Grover coin $C = 2 |D \rangle \langle D | - I$ is a $\Delta \times \Delta$ matrix, where $\Delta$ may not be equal to $2^m$. We, therefore, define a new $2^m \times 2^m$ operator $C'$ by

$$C' = \begin{bmatrix} 2 |D \rangle \langle D | & 0 \\ 0 & 0 \end{bmatrix} - I = 2 |D' \rangle \langle D' | - I,$$

where, $|D' \rangle = \frac{1}{\sqrt{\Delta}} \sum_{k=1}^{\Delta} |\alpha_{k-1}\rangle \in \mathbb{C}^{2^m}$ is a projection of the diagonal state $|D \rangle$ in the higher dimensional Hilbert space $H_{C'} = \mathbb{C}^{2^m}$ that contains the coin space $H_C = \mathbb{C}^\Delta$ as a subspace. In other words, $H_{C'}$ is a new coin space with the computational basis $\{|\alpha_{k-1}\rangle : 1 \leq k \leq 2^m\}$, and the quantum walk occurs in $Cay(\mathbb{Z}_2^n, \Omega')$, where
Ω' has $2^m$ elements containing Ω, i.e., $\text{Cay}(\mathbb{Z}_2^n, \Omega)$ is a subgraph of $\text{Cay}(\mathbb{Z}_2^n, \Omega')$. The new coin operator $C'$ changes the coefficients of the initialized vector, which is $|D'_1\rangle$, and therefore, the coefficients of coin states $|\alpha_k-1\rangle$, with $k \geq d+1$, remain 0 throughout the evolution, i.e. the generic coin state $|c\rangle$, in our case, is

$$|c\rangle = \sum_{k=1}^{\Delta} \lambda_k |\alpha_{k-1}\rangle + \sum_{k=\Delta+1}^{2^m} |\alpha_{k-1}\rangle.$$  

We define a new shift operator $S'$ by

$$S' = \sum_{k=1}^{2^n-1} \sum_{a=0}^{2^n-1} |\alpha_{k-1}\rangle \langle v_a \oplus \Omega(k) | \langle \alpha_{k-1}\rangle \langle v_a | + \sum_{k=\Delta+1}^{2^m} \sum_{a=0}^{2^n-1} |\alpha_{k-1}\rangle \langle v_a | \langle \alpha_{k-1}\rangle \langle v_a |.$$  

The new shift operator $S'$ fixes $|\alpha_{k-1}\rangle \langle v_a |$, $0 \leq a \leq 2^n-1$, if $k \geq \Delta + 1$, i.e., $S'|\alpha_{k-1}\rangle \langle v_a | = |\alpha_{k-1}\rangle \langle v_a |$. The Figure 6 represents the quantum circuit for $Q_3$. Notice that the circuit does not shift a position state along the direction $|\alpha_3\rangle \equiv |11\rangle$. Finally, the original coin state is retrieved at the end of the shift operation because each quantum wire contains even number of X gates.

5. Application to hitting times on cubelike graphs

We use Qiskit [1] to implement DTQW on IBM’s Quantum simulators and quantum computers. The Qiskit simulator qasm_simulator runs locally and other IBM’s simulators such as simulator_mps, simulator_extended_stabilizer, etc, are accessed via IBM provider ibm-q.

5.1. DTQW on Quantum computers. The quantum circuit for DTWQ on $Q_2$, see Figure 2, was run on IBM’s quantum computer ibmq_manila v1.0.3, the result of which is compared with the result on qasm_simulator, as shown in Figure 7. The quantum circuits corresponding to higher dimensional cubelike graphs could not be run successfully on real quantum computers due to unavoidable gate errors and noise from the environment around the quantum computer. In case of $Q_3$, the errors and noise were low and therefore could be run successfully.

5.2. Hitting times of Hypercubes and Augmented cubes. Upon implementing quantum circuits for DTQW on cubelike graphs, we have observed that the specific vertex, called the target vertex, to which the walker reach with highest
probability, is unique and equal to the binary XOR operations of elements of the generating set $\Omega$, i.e.,

$$\text{target vertex } = \bigoplus_{x \in \Omega} x.$$  

Figure 7. Probability distribution of DTQW on $Q_2$ after two steps when run on Qiskit simulator (a) *qasm_simulator* and on IBM’s quantum computer (b) *ibmq_manila*.

(a) DTQW on $Q_3$ with steps $T = 3$, target vertex $7 \equiv 111$, and target probability $0.804$.

(b) DTQW on $AQ_3$ with steps $T = 9$, target vertex $3 \equiv 011$, and target probability $0.812$.

Figure 8. Probability distribution of Hypercubes and Augmented cubes of dimension 3. Target vertices correspond to the tallest bar in each bar graph.
(a) DTQW on $Q_4$ with steps $T = 6$, target vertex $15 \equiv 1111$, and target probability 0.562.

(b) DTQW on $AQ_4$ with steps $T = 11$, target vertex $4 \equiv 0100$, and target probability 0.993.

(c) DTQW on $Q_5$ with steps $T = 7$, target vertex $31 \equiv 11111$, and target probability 0.722.

(d) DTQW on $AQ_5$ with steps $T = 13$, target vertex $11 \equiv 01011$, and target probability 0.953.

Figure 9. Probability distribution of Hypercubes and Augmented cubes of dimension 4 and 5.
Table A. $Q_n$. Table B. $AQ_n$.

In Table A and Table B, $n$ denotes dimension of $Q_n$ and $AQ_n$. $T$ is the hitting time with probability $p$, and $\Delta$ is the degree.

In Figure 10 the target probabilities for Augmented cubes are higher than that of Hypercubes of same dimensions. In either case, the target probability gets closer to 1 as the dimension increases. In Table A and B, we have displayed the hitting time $T$ along with target and target probability corresponding to dimensions of $Q_n$ and $AQ_n$. In case of Hypercubes, $T$ is linear with $n$ that tallies with the theoretical result stated by Kempe in [17], while in Augmented cubes it is linear.
with the degree of the graph. See Figure 10b, where we have shown that $T$ is linear with the degree $2n - 1$ of $AQ_n$.

![Plots of steps T required to attain the target probability verses the degree $|\Omega|$ of cubelike graphs of degree $n + k$, where $k$ is fixed and $n$ is the dimension.](image)

**Figure 11.** Plots of steps $T$ required to attain the target probability verses the degree $|\Omega|$ of cubelike graphs of degree $n + k$, where $k$ is fixed and $n$ is the dimension.

5.3. A conjecture on hitting times of general cubelike graphs. The result mentioned by J. Kempe in [17] can be generalized to other cubelike graphs. In Figure 11 and 12 we have shown plots of steps $T$ verses degrees $n + k$ of some cubelike graphs, where $T$ is the number of iterations required to hit the target vertex, and $n$ is the dimension of cubelike graph of degree $n + k$. In the first figure, we fix the value of $k$ and increase $n$, and in the second figure, we fix the dimension $n$ and increase the value of $k$. In either case, we find that $T$ verses $n + k$ plot is
linear and follows the relation:

\[ T \approx (n + k) \times \frac{\pi}{2}. \]

It is to be noted that it is just a coincidence that \( T \) vs dimension plot is same as \( T \) vs degree plot for DTQW on Hypercubes.

\[ v_{\text{targ}} = \bigoplus_{x \in \Omega} x. \]

Then, there exists a hitting time \( T \approx \frac{\Delta}{2} \) such that the target probability \( p_{\text{targ}} \), the probability by which DTWQ reaches \( v_{\text{targ}} \), is asymptotically equal to 1. Moreover, the parity of \( T \) is same as that of \( \Delta \), i.e., \( T \) is even only if \( \Delta \) is even.

6. Conclusion

Quantum random walks and their hitting times are an important area of study in quantum computation due to their applications in development of faster quantum algorithms. In this paper we have implemented efficient quantum circuits for discrete quantum random walks on families of cubelike graphs such as hypercubes and augmented cubes. Our implementations show that the hitting-times of all cubelike
graphs is asymptotically linear in the degree $\Delta$ of the graphs. That is, for the hitting time $\frac{\Delta^2}{2}$, probability that the walker is found at the target vertex approaches 1 as $\Delta$ approaches infinity. Our circuits run on IBM’s quantum computing platform Qiskit, both on real quantum computers as well as simulators.

We note that $T \approx \frac{\Delta^2}{2}$ is not necessarily the minimum hitting time or its multiple for a $\Delta$-regular cubelike graph. For example, in DTQW on $Q_3$, see Figure 13, the walker hits the target vertex $|111\rangle$ after steps $T = 23$ with probability 1. Another example is that of a complete graph. The cubelike graphs $\text{Cay}(\mathbb{Z}_2^n, \Omega)$, with $\Omega = \mathbb{Z}_2^n \setminus \{0^n\}$, are complete graphs. In Figure 14, the hitting time for the complete graphs on $2^n$ vertices has been shown constant with the value $T = 4$, with target probability equal to 1, irrespective of dimensions and degrees of the graphs. We have theoretically verified that $T = 4$ is the minimum hitting time with the target probability equal to 1 for the complete graphs on $2^n$ vertices. It will of interest to find other hitting times for a cubelike graph, specially the minimum one for which the walker hit the target with high probability or with probability equal to 1.

![Figure 13. Probability distribution of $Q_3$ after step $T = 23$.](image)

![Figure 14. Plot for hitting time $T$ versus degree $2^n - 1$ of complete graph on $2^n$ vertices.](image)

Further, it would be interesting to study the same problem on other types of regular graphs particularly cayley graphs that are not cubelike graphs. Another
implementation problem to study along the same lines is of perfect state transfer on regular graphs in the continuous time random walks. It would be interesting to build efficient quantum circuits for the perfect state transfer problem.

References

[1] Abraham, H., and et. al. Qiskit: An open-source framework for quantum computing, 2019.
[2] Acasiete, F., Agostini, F., Khatibi Moqadam, J., and Portugal, R. Implementation of quantum walks on ibm quantum computers. *Quantum Information Processing* 19, 426 (2020).
[3] Acevedo, O. L., and Gobron, T. Quantum walks on cayley graphs. *Journal of Physics A: Mathematical and General* 39, 3 (Dec 2005), 585–599.
[4] Aharonov, D., Ambainis, A., J., K., and Vazirani, U. Quantum walks on graphs. *Proceedings of ACM Symposium on Theory of Computation (STOC’01)* (2001), 50–59.
[5] Aharonov, Y., Davidovich, L., and N., Z. Quantum random walks. *Phys. Rev. A* 48(2), 6 (1993), 1687–1690.
[6] Ambanis, A. Quantum walks and their algorithmic applications. *International Journal of Quantum Information* 1 (2003), 507–518.
[7] Batt, M., Braunstein, S., Duncan, A., and Rees, S. Quantum algorithms in group theory.
[8] Bernasconi, A., Godsil, C., and Severini, P. Quantum networks on cubelike graphs. *Phys. Rev. A* 78, 052320 (2008).
[9] Cheung, W., and Godsil, C. Perfect state transfer in cubelike graphs. *Linear Algebra and its Applications* 435 (2011), 2468–2474.
[10] Childs, A. Universal computation by quantum walk. *Phys. Rev. Lett.* 102:180501 (2009).
[11] Choudum, S., A., and Sunitha, V. Augmented cubes. *Networks* 40(2) (2002), 71–84.
[12] Farhi, E., and Gutmann, S. Quantum computation and decision trees. *Phys. Rev. A* 58, 2 (1998), 915–928.
[13] Georgopoulos, K., Emary, C. , and Zuliani, P. Comparison of quantum-walk implementations on noisy intermediate-scale quantum computers. *Phys. Rev. A* 103, 022408 (2021).
[14] Godsil, C. State transfer on graphs. *Discrete Mathematics* 312(1) (2012), 129–147.
[15] J., A., and et. al. Quantum algorithm implementations for beginners, 2020.
[16] Kempe, J. Quantum random walks: An introductory overview. *Contemporary Physics* 44, 4 (2003), 307–327.
[17] Kempe, J. Quantum random walks hit exponentially faster. *Probability Theory and Related Fields* 133(2) (2005), 215–235.
[18] Knight, P., Roldán, E., and Sipe, J. Optical cavity implementations of the quantum walk. *Optics Communications* 227 (2003), 147–157.
[19] Lavor, C., Manssur, L., and Portugal, R. Grover’s algorithm: Quantum database search. arXiv: Quantum Physics (02 2003).
[20] Lovett, N., Cooper, S., Everitt, M., Trevers, M., and Kendon, V. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A* 81(4):042330 (2010).
[21] Lovász, L. Random walks on graphs: A survey, 1993.
[22] Moore, C., and Russell, A. Quantum walks on the hypercube. In *Randomization and Approximation Techniques in Computer Science* (Berlin, Heidelberg, 2002), J. D. P. Rolim and S. Vadhan, Eds., Springer Berlin Heidelberg, pp. 164–178.
[23] Portugal, R. *Quantum Walks and Search Algorithms*. Springer Publishing Company, Incorporated, 2013.
[24] Roldán, E., and Soriano, J. Optical implementability of the two-dimensional quantum walk. *Journal of Modern Optics* 52 (2005), 2649–2657.
[25] Venegas-Andraca, S. E. Quantum walks: a comprehensive review. *Quantum Information Processing* 11, 5 (2012), 1015–1016.
[26] Wanzambe, E., and Anderson, S. Quantum computing: Implementing hitting time for coined quantum walks on regular graphs. https://arxiv.org/pdf/2108.02723.pdf (2021).
[27] Xue, P., Sanders, C., and Leibfried, D. Quantum walk on a line for a trapped ion. *Phys. Rev. Lett.* 103(18):183602 (2009).