Quantum Random Walks on Cayley Graphs

Marina Knittel[1] and Roozbeh Bassirian[1]

[1]*Department of Computer Science , University of Maryland*

Abstract

We investigate the problem of quantum random walk on Cayley graphs. Quantum random walks are proved to be useful in different aspect of quantum computation. Our motivation is finding the current limitations of quantum random walks, and how these results can be applied to graphs with certain structures. For this purpose, we focus on Cayley graphs, the diagrammatic representation of groups. Aside from containing useful classes of graphs - hypercubes, grids etc. - one can use quantum random walks on Cayley graphs for random element generation and element finding which makes this class interesting for our study.

1 Introduction

This project concerns the use of different coins for quantum discrete random walks on Cayley graphs. We plan to survey the impact of different coin strategies on the distribution of walks and the efficiency of algorithms, and evaluate how this varies across the different Cayley graphs.

A lot of different well-established ideas have already been suggested for using quantum random walks, such as algorithmic speedups [1, 2, 3] and using quantum random walks as a universal computational primitive [4]. However, it is still not clear how their improved hitting time, compared to the classical case, can help us achieve algorithmic speedups. In some cases this speedup is exponential because of the intrinsic properties of the underlying graphs. This includes the hypercubes, a subset of Cayley graphs.

Classical random walks on Cayley graphs are well-studied. A Cayley graph is constructed to represent a group structure based off of how nodes interact with the group generators through the group operator. Therefore, a traversal through a Cayley graph is equivalent to a construction of group elements through applying generators. The authors of [5] describe this process as random element generation. Random group elements are used in a number of group theoretic algorithms, including strong generation, manipulation of certain permutation groups and Sylow subgroups, as well as a number of computational group theory libraries.

We can further investigate walks on Cayley graphs through the use of quantum random walks. In these walks, the random selection of walk direction is modeled by the use of a coin, which is a unitary transformation that is applied to the current state of the particle. We have seen how the choice of the coin can affect the efficiency of different algorithms that are based on quantum random walks. In [6], Ambainis *et al.* show how the choice of coin changes the outcome of a spatial search algorithm.

In the case of random generation, the outcome of random walks should be uniform. This is because algorithms that use randomness for computational improvements generally prefer to select elements with equal probability: or, from a uniform distribution. Coins can be used to control the bias and variance of the quantum walk's distribution [7]. There exist a number of coin strategies utilized by different algorithms based off of the goal of the algorithm and the topology of the graph.

Therefore, this survey strives to develop a deeper understanding of the relationships between coins and quantum random walks on Cayley graphs. Specifically, we should evaluate the impact of different coins on the effectiveness of these algorithms to achieve practical goals.

The structure of this manuscript is as follows. We start with explaining the basic notations and definitions on random walks and Cayley graphs in section 2. In section 3, we summarize multiple results and limitations of quantum random walks on certain well-defined Cayley graphs such as grids, hypercubes and the Cayley graph constructed from Free group. Finally, we talk about possible improvements in section 4.

2 Preliminaries

2.1 Classical Random Walks

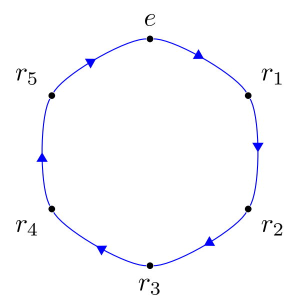
Classical random walks are used for modeling many phenomena across different fields. Shares prices in economics, random movements of molecules in physics, decision making in psychology, and search algorithms in computer science are a number of interesting applications of classical random walks. Informally, a random walk models the random movement of an object in a mathematical space. A number of interesting properties derived from describing the problem using this stochastic model are:

- Stationary distribution: For any Markov chain transition matrix P, a distribution π is stationary if $\pi = P\pi$. This is basically the distribution of possible locations after we reach the equilibrium.
- Hitting time: The average number of steps required for the random walk to reach a certain state.
- Mixing time: The average number of steps required to reach the stationary distribution.
- Cover time: The average number of steps required to visit every node.
- Commute time: The average number of steps it takes to start from a node v, reach node u and get back to v.

A simple example of a classical random walk is a random walk on integer number line, where at each step the location moves to an adjacent number with probability $\frac{1}{2}$.

2.2 Cayley Graphs

For any group G and $S \subseteq G$ s.t. $I \notin S$, the nodes of the Cayley graphs are the elements of G, and the edge (e_1, e_2) exists if and only if $e_1 e_2^{-1} \in S$. Sometimes it is also assumed that the set S has to be a set of generators of G. A simple example of a Cayley graph can be given from the cyclic group, where there exists an element g such that $G = \langle g \rangle$. A simple graph for the case where $g^6 = I$ is as below:



Where the blue edges are application of g.

2.3 Quantum random walks

In this survey, we focus on the discrete model of quantum random walks, where the particles are put on a finite space such as a lattice or a graph. Let us start with defining a simple model of quantum random walk in one dimension. Informally, we would like to have a Hilbert space $\mathcal{H}_P = \{|i\rangle \in \mathbb{Z}\}$, and an operator $\mathcal{E}$, where:

$$|i\rangle \xmapsto{\mathcal{E}} \begin{cases} |i+1\rangle \text{ With probability } \frac{1}{2} \\ |i-1\rangle \text{ With probability } \frac{1}{2} \end{cases}$$

Clearly, this is not a reversible operation. However, it is possible to create the desired unitary operator by purifying the position Hilbert space and adding a "coin Hilbert space" $\mathcal{H}_C = \{|0\rangle, |1\rangle\}$. To replace $\mathcal{E}$, we can define the unitary operator S acting on $\mathcal{H}_P \otimes \mathcal{H}_C$, where:

$$S|i\rangle|0\rangle = |i+1\rangle|0\rangle$$
$$S|i\rangle|1\rangle = |i-1\rangle|1\rangle$$

This is called a shift operator and it acts to move the particle executing the walk. It is possible to simulate a classical random walk by "flipping" the coin using a Hadamard operator and applying S:

$$\frac{1}{\sqrt{2}}S|i\rangle\big(|0\rangle + |1\rangle\big) = \frac{1}{\sqrt{2}}\big(|i+1\rangle|0\rangle + |i-1\rangle|1\rangle\big)$$

Thus the entire transformation at each step is $U = SC$, where S is our shift operator, and H is our coin. After measuring the coin qubit in computational basis we get the desired output. The goal of quantum random walk is to eliminate the measurement step and achieve a speed up by letting different paths of random walk interfere with each other. Starting from $|0\rangle|0\rangle$, it is easy to verify that this coin is not balanced and does not produce symmetric amplitudes. This is just because $H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$. One way to fix this problem is to start from $\frac{1}{\sqrt{2}}|0\rangle(|0\rangle + i|1\rangle)$. This way the real amplitudes and imaginary amplitudes will not interfere with each other and we get a balanced coin.

2.4 Coins on Quantum Random Walks

We have already seen how a simple choice of initial state for the coin state can effect the probability outputs of the quantum random walk. There has been different studies on how it is possible to leverage the coin state and coin operator to increase efficiency, simulate classical walks, etc. We are going to see in the next section how the choice of coin can even affect the running time of algorithms that are based on quantum random walks exponentially [6]. The effect of decoherence on coins is also well studied [8].

It is also worth mentioning that it is in fact possible to do discrete quantum random walk without using entangled coins [9], and only use the superposition of states to implement the desired outcome. However, this perspective is still handy since it can helps us grasp a better understanding of the difference between quantum and classical random walks.

2.5 Hadamard Coin

One of the natural choices of coin is the Hadamard operator, that is also used in Section 2.3. It almost implements the process of coin tossing at each step of the random walk. Generally, a d-dimensional DFT can simulate the coin toss for arbitrary number of choices. For example, for the case of d-dimensional hypercube an appropriate operator would be:

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

Where $\omega = e^{2\pi i/d}$ is the d-th primitive root of unity.

2.5.1 Grover Coin

The name Grover coin comes from the Grover operation that was used in the original Grover algorithm for searching unstructured database [7]. One of the nice characteristics of this coin is that it is permutation symmetric, and it is shown that the only unitary operator to have this property is of form:

$$G_{a,b} = \begin{pmatrix} a & b & b & \dots & b & b \\ b & a & b & \dots & b & b \\ b & b & a & \dots & b & b \\ & & & \dots & & \\ b & b & b & \dots & a & b \\ b & b & b & \dots & b & a \end{pmatrix}$$

Where, $a = \frac{2}{d} - 1$ and $b = \frac{2}{d}$, where d is the dimension of this matrix (possible number of choices).

2.6 The Limiting Distribution

A common problem addressed in classical random walks is mixing time. Mixing time represents how long it takes for a random walk to achieve a within error distance from a stationary distribution. However, because quantum random walks use unitary transformations, the distance between a state and its next state, or $|\alpha\rangle - U|\alpha\rangle$, does not converge. Additionally, the probability distributions, $P_t(v|\alpha_0) = \sum_{a \in A} |\langle a, v|\alpha_t\rangle|^2$, of the nodes in the graph does not converge over t. Although, we find that the average probability distributions as a function of t do converge. So if $|\alpha_0\rangle$ is our system at time 0, and $|\alpha_t\rangle$ is our system at time t, then $\overline{P}_T$ converges.

$$\overline{P}_T(v|\alpha_0) = \frac{1}{T}\sum_{t=0}^{T} P_t(v|\alpha_t).$$

Intuitively, $\overline{P}_T$ is a distribution of nodes such that, if you uniformly select a time t from 0 to T to measure the system, $\overline{P}_T$ represents the probability distribution of measured states. This is our limiting distribution.

Definition 1 *The limiting distribution of a quantum random walk is defined as the limit of the average probability distribution of the system. It is denoted as follows.*

$$\pi(v) = \overline{P}_T(v|\alpha_0).$$

The limiting distribution acts as a proxy for the classical stationary distribution. Despite its dependence on the time of measurement, it can be used in similar ways to the stationary distribution. Therefore, we define the quantum mixing time in terms of the limiting distribution. We use $P(\cdot|a,v)$ to denote parameterization with starting state $|a,v\rangle$.

Definition 2 *The mixing time of a quantum random walk defines the time it takes to for the average probability distribution to approach the limiting distribution of the system. We parameterize it with acceptable margin of error, ϵ.*

$$M_\epsilon = \min\{T \,|\, \forall t \geq T, |a,v\rangle \,:\, ||\pi(\cdot|a,v) - \overline{P}_t(\cdot|a,v)|| \leq \epsilon\}.$$

We use quantum mixing time as an analog to its classical counterpart.

2.7 Quantum Walk for Search

A simple search algorithm can be achieved by what is known as a perturbed random walk. Say the vertices in our graph are either marked or unmarked, and we are searching for a marked vertex. Then, we define a perturbed quantum walk for search as follows.

Definition 3 *A perturbed quantum random walk with marked vertex v and "marking coin" $C_1 = -I$ with regular coin C_0 is defined as $U' = SC'$ where*

$$C' = C_0 \otimes (I - |v\rangle\langle v|) + C_1 \otimes |v\rangle\langle v|\,.$$

Here, C_1 acts as the coin flip for the marked vertex. We utilize an auxiliary qubit which C_1 sets to $|1\rangle$, and C_0 sets to $|0\rangle$. So when we measure the outcome of the random walk, we can determine if we are at a marked or unmarked vertex. Interestingly, Ambainis et al show show that such a perturbed random walk on a complete graph can be viewed as an application of Grover's Algorithm [6].

3 Results

Next, we are going to mention some of the notable results in this area to emphasize the importance of the coin selection, and describe the state-of-the-art for quantum random walks for Cayley graphs. We are interested in exploring results for a number of known and simpler Cayley graphs. Results for hitting and/or mixing algorithms are broken down by graph type. Finally, we address the more general topic of free groups, and explore some theoretical properties of quantum random walks for free groups.

3.1 The Cycle

One of the simplest groups is the finite cyclic group. Generated by a single element, the structure of the cyclic group can be characterized by starting at the generator, applying it repeatedly, and ending up back at the generator. So for every node in the graph, there is only one way in, and one way out, both by applications of the single generator. It can then be represented as a cycle graph.

Properties of the quantum random walk on the cycle graph on the were studied fairly early on, by Aharonov et al [10]. Specifically, they characterized the limiting distribution and found a bound on the mixing time of a quantum random walk on the cycle with a Hadamard coin. Their techniques involved examining the eigenvectors of the unitary transformation U to find the following limited distribution.

Theorem 4 *The limiting distribution π for the coined quantum walk on the n-cycle, with n odd, using the Hadamard transform as the coin, is uniform on the nodes, independent of the initial state $|\alpha_0\rangle$.*

They then place a bound on the mixing time.

Theorem 5 *For the quantum walk on the n-cycle, with n odd, using the Hadamard coin, we have*

$$M_\epsilon \leq O\left(\frac{n \log n}{\epsilon^3}\right).$$

This provides an almost quadratic speed up over classical random walks on the cycle graph. Aharonov et al. also proved this is optimal for the cycle graph. While this is significant results, they also showed that we may not try to look much further.

Corollary 6 *For a general quantum walk on a bounded degree graph, the mixing time is at most quadratically faster than the mixing time of the simple classical random walk on that graph.*

Therefore, quantum random walk algorithms on any graph cannot achieve certain speedup improvements that we may hope to discover. However, a quadratic speedup is a significant improvement, that could have practical value.

It is also possible to compare the hitting time of quantum random walks to its classical counter part and consider how it can be used to speed up search algorithms. Wong et al. in [11], use Szegedy's quantum random walk, which is a quantization of a classical random walk using reflection operators, to gain separations from classical algorithms for hitting time.

In this method, they first consider the bipartite double cover of a cycle, where a 2 partite graph is constructed by cloning vertices, and one original vertex and a clone of another are connected if and only if they are connected in the original graph. The random walk occurs on the edges on this new graph. The edges are spanned by the following orthonormal basis:

$$\{|x, y\rangle : x \in X, y \in Y\}$$

Where X, Y are partitions of the bipartite double cover graph. Szegedy's walk is done by applying $W_P = R_a R_b$, where:

$$R_a = 2 \sum_{x \in X} |\phi_x\rangle\langle\phi_x| - I$$

$$R_b = 2 \sum_{y \in Y} |\psi_y\rangle\langle\psi_y| - I$$

Are the reflection operators, and:

$$|\phi_x\rangle = |x\rangle \otimes \sum_{y \in Y} \sqrt{P_{xy}}\, |y\rangle$$
$$|\psi_y\rangle = \sum_{x \in X} \sqrt{P_{xy}}\, |x\rangle \otimes |y\rangle$$

With $P_{xy} = \frac{1}{deg(x)}$ if and only if x and y are adjacent.

Using this walk, they are able to prove that arbitrary hitting time separation can be achieved depending on the number of marked vertices. For example, if $k = N/\log N$ number of them are marked and contiguous, the hitting time is $N/k = \log N$, compared to the classical case where the hitting time is $O(N^2)$. However, for a search problem, both hitting time and mixing time matters, because we need to run the algorithm a number of times to mix the probabilities, and we have to repeat this process based on hitting time to get a constant probability of success by union bound, which in this case results in $O(N^2 logN/k)$ which is roughly a quadratic speedup. In [12], Wong also shows that it is possible to convert this Szegedy's walk, to an equivalent version using Grover's coin and flip flop operators.

3.2 Grids

Infinite grids on hyper plane R^n can be generated from Z^n with the set of generators S, where:

$$S = \{(\pm 1, 0 \ldots, 0), (0, \pm 1, \ldots, 0), (0, \ldots, 0, \pm 1)\}$$

Ambainis et al. discuss search algorithms on grids using perturbed random walks [6]. Instead of examining coins, they compare the use of different shift operators, which can have similar impacts on the effectiveness and distribution of the resulting algorithm. They formulate their problems with vertices being marked by their x, y coordinates, so in the 2-dimensional case, they look like $|x, y\rangle$. Edges, or transitions, are represented by $\downarrow, \uparrow, \leftarrow, \rightarrow$. Their work utilizes two different shift transformations.

Definition 7 *The "flip-flop" shift continually reverses the direction of traversal.*

$$\begin{aligned} S_{ff} : &|\rightarrow, x, y\rangle \rightarrow |\leftarrow, x+1, y\rangle, \\ &|\leftarrow, x, y\rangle \rightarrow |\rightarrow, x+1, y\rangle, \\ &|\uparrow, x, y\rangle \rightarrow |\downarrow, x+1, y\rangle, \\ &|\downarrow, x, y\rangle \rightarrow |\uparrow, x+1, y\rangle. \end{aligned}$$

Definition 8 *The "moving" shift continues to travel in the same direction.*

$$\begin{aligned} S_m : &|\rightarrow, x, y\rangle \rightarrow |\rightarrow, x+1, y\rangle, \\ &|\leftarrow, x, y\rangle \rightarrow |\leftarrow, x+1, y\rangle, \\ &|\uparrow, x, y\rangle \rightarrow |\uparrow, x+1, y\rangle, \\ &|\downarrow, x, y\rangle \rightarrow |\downarrow, x+1, y\rangle. \end{aligned}$$

For a grid on 2-dimensions, they show that S_{ff} is a "good" coin, and S_m is a "bad" coin. This is counter-intuitive, because S_{ff} seems like a walk algorithm that would be stuck going back and forth between points in the classical case, whereas the S_m moves in a single direction and never backtracks. However, they do a Grover's algorithm-like analysis to show the following.

Theorem 9 *For the quantum walk search algorithm associated to the quantum walk* $U = S_{ff}(C \otimes I)$*, there is a* $T = O(\sqrt{NlogN})$*, such that after* T *steps the probability to determine the marked state is* $O\left(\frac{1}{\log n}\right)$ *.*

And using the worse coin, they found a bad lower bound.

Theorem 10 *The quantum walk search algorithm associated with* S_m *takes at least* $\Omega(N)$ *steps to determine the marked state with constant probability.*

With the results from Theorem 9, they could find a $O(\sqrt{N}\log N)$ algorithm to find a marked vertex with probability $O\left(\frac{1}{\log N}\right)$. Similar results were found for higher dimensional grids.

3.3 Hypercubes

Another interesting Cayley graph is the n-dimensional hypercube. It is, in effect, the extrapolation of a square or a cube into n dimensions. It is based off of the group of n-dimensional vectors on $\mathbb{Z}/2\mathbb{Z}$, commonly represented as n-bit strings.

Krovi and Brun achieved important but limited results on search algorithms on the hypercube [13]. They examined the problem of search on a hypercube, where the initial state and the goal state are on opposite sides of the hypercube.

Theorem 11 *A quantum random walk search algorithm using the Grover coin on the hypercube from one corner to its opposite achieves a polynomial hitting time.*

It turns out this is an exponential speedup over classical results, which is the strongest improvement we've seen yet. However, it has only shown to hold in the case of finding the opposite corner of the hypercube. Krovi and Brun also explored the use of a Discrete Fourier Transform coin, which had an infinite hitting time. This underlines the importance of careful coin selection.

In addition, Marquezino et al explored mixing time for the quantum random walk on the hypercube [14]. Their results are specifically for discrete quantum random walks with a symmetric initial condition. Using the Grover coin then, they were able to achieve the following.

Theorem 12 *The mixing time of a quantum walk algorithm using the Grover coin on a hypercube with a symmetric initial condition is* $O(n/\epsilon)$*.*

It should be noted that this bound was verified experimentally, and not mathematically, though it agrees with related work. These results exhibit a subquadratic speedup over the $O(n \log n)$ classical counterpart.

In the study of mixing time, we are most interested in uniform limiting distributions. While the distributions achieved by Marquezino et al are not uniform across nodes in the graph, they do show a uniformity of a measure known as Hamming distance. Since vertices of hypercubes can represent binary strings, and edges connect strings that differ in only one digit, it is interesting to define the lexicographic distance between vertices.

Definition 13 *The Hamming distance of two binary string is the number of digits in which the two strings differ.*

Thus in a hypercube, edges represent Hamming distances of 1, and the Hamming distance between two vertices is the length of the shortest path between them. Then Marquezino's results are as follows.

Theorem 14 *The limiting distribution of a quantum walk algorithm using the Grover coin on a hypercube with a symmetric initial condition achieves any Hamming distance with equal probability.*

Therefore, while the Grover coined walk on hypercubes does not achieve the desired uniform distribution, it does exhibit a uniformity of travel distance across the hypercube. Again, all results were found empirically.

3.4 Characteristic of Quantum Random Walks on Cayley Graphs

Quantum random walks on Cayley graphs of free groups are well studied in Acevedo et al [15]. Acevedo et al mostly focuses on Cayley graphs of free group in this work, and they also find some results for general abelian groups, which is a more complicated case. They generalize the concept of coins in quantum random walks with "internal states", where instead of tossing a coin at every step, an evolution operator is applied on the internal state of each node. Formally, the evolution operator W, where $|\psi_{t+1}\rangle = W|\psi_t\rangle$ is defined as:

$$W = \sum_{x\in X}\sum_{z\in E_x} M_{x,z}T_{x\to z}$$

Where E_x is the set of neighbors of x, and X is the set of all nodes. $T_{x\to z}$ is the transition operator from state x to z operating on Hilbert space H_G of graph nodes:

$$\langle\phi| T_{x\to z}|\psi\rangle = \langle\phi|z\rangle\langle x|\psi\rangle$$

And $M_{x,z}$ is the internal state operator acting on H_I which is some operation similar to the coin operator in previous models. The overall state is represented in $H_I \otimes H_G$. The unitarity of W is satisfied if and only if:

$$\sum_{z\in E_x\cap E_{x'}} M^\dagger_{x,z}M_{x',z} = \delta_{x,x'}I$$

In the case of Cayley graphs, the evolution operator can be rewritten using the fact that the edges represent the generators. Suppose that Δ is a subset of a Group G, then:

$$W = \sum_{\delta\in\Delta} M_\delta \otimes T_\delta$$

Where:

$$T_\delta = \sum_{x\in X} T_{x\to x\delta}$$

It is also possible to rewrite the unitarity condition in terms of generators:

$$\sum_{\delta_1\delta_2^{-1}} M^\dagger_{\delta_1}M_{\delta_2} = \delta_{\{u=e\}}I$$

These conditions should also be true for the conjugated internal operators since $WW^\dagger = W^\dagger W = I$.

Their main results for free group is the necessary and sufficient conditions for the coin state so that the universal operation of quantum random walk on general graphs is unitary. They also characterize all possible walks on this group. Furthermore, they also suggest some possible solutions for general abelian groups.

3.4.1 Free group

It is possible to simplify the previous condition for if we assume that the Cayley graph is constructed using the free group:

$$M_{\delta_1}^\dagger M_{\delta_2} = M_{\delta_1} M_{\delta_2}^\dagger = 0$$
$$\sum_\delta M_\delta^\dagger M_\delta = M_\delta M_\delta^\dagger = I$$

The following result summarizes the conditions on the evolution operator for free groups.

Theorem 15 *On the Cayley graph of free group, the quantum walk evolution operator is unitary if and only if the internal operators are of the form*

$$M_\delta = UP_\delta$$

Where U is a unitary matrix on H_I, and $\{P_\delta\}_{\delta in \Delta}$ is a complete family of orthogonal operators:

$$\sum_\delta P_\delta = I$$

The internal space is of dimension larger or equal to $|\Delta|$.

This comes from the fact that elements of the free group have no relation with each other, and thus the application of one generator versus another should be orthogonal operations. By writing the unitary condition where $WW^\dagger = I$, one can prove Theorem 15.

4 Conclusion and Outlook

There are lots of possibilities for improvements on known quantum random walk algorithms on Cayley graphs. To our knowledge, little is known about lower bounds of these algorithms, whether its hitting time, mixing time or search algorithms. One of the few lower bounds that works only for spatial search of grids[16], which is also not tight for all cases.

It is interesting that all these results depend on either Grover's coin or Hadamard coin, even when a large class of unitary evolution operators can be constructed without the use of these coins in the case of the Free group. Even if it is intuitive since all these methods rely on amplitude amplification to achieve speedups from their classical counter parts, it might be worth knowing how using other coins might affect running time of these algorithms.

One method that might make implementing these algorithms more plausible is reducing the size of the Hilbert space of the position and coin states. It is also possible to use known methods to eliminate the entanglement of the coin state, and use only the superposition of position states to do quantum walks. It might also be possible to work on experiments to implement known models.

One extension of these works that is still unexplored is the quantum random walk on non-abelian Cayley graphs. To our knowledge there is only one recent work that constructs a model for the dihedral group [17], and it seems like this is the first step to quantum random walk on non-abelian Cayley graphs. However, there are still no simple description of limiting distributions of quantum walks for non-abelian groups [10].

References

[1] N. Shenvi, J. Kempe, and K. B. Whaley, “Quantum random-walk search algorithm,” *Physical Review A*, vol. 67, no. 5, p. 052307, 2003.

[2] A. Ambainis, “Quantum walk algorithm for element distinctness,” *SIAM Journal on Computing*, vol. 37, no. 1, pp. 210–239, 2007.

[3] M. Szegedy, “Quantum speed-up of markov chain based algorithms,” in *Foundations of Computer Science, 2004. Proceedings. 45th Annual IEEE Symposium on.* IEEE, 2004, pp. 32–41.

[4] A. M. Childs, “Universal computation by quantum walk,” *Physical review letters*, vol. 102, no. 18, p. 180501, 2009.

[5] L. Babai, “Local expansion of vertex-transitive graphs and random generation in finite groups,” *ACM Symposium on Theory of Computing*, pp. 164–174, 1991.

[6] A. Ambainis, J. Kempe, and A. Rivosh, “Coins make quantum walks faster,” in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms.* Society for Industrial and Applied Mathematics, 2005, pp. 1099–1108.

[7] J. Kempe, “Quantum random walks: an introductory overview,” *Contemporary Physics*, vol. 44, no. 4, pp. 307–327, 2003.

[8] T. A. Brun, H. A. Carteret, and Ambainis, “Quantum random walks with decoherent coins,” *Physical Review A*, vol. 67, no. 5, p. 032304, 2003.

[9] A. Patel, K. Raghunathan, and P. Rungta, “Quantum random walks do not need a coin toss,” *Physical Review A*, vol. 71, no. 3, p. 032347, 2005.

[10] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, “Quantum walks on graphs,” in *Proceedings of the thirty-third annual ACM symposium on Theory of computing.* ACM, 2001, pp. 50–59.

[11] T. G. Wong and R. A. Santos, “Exceptional quantum walk search on the cycle,” *Quantum Information Processing*, vol. 16, no. 6, p. 154, 2017.

[12] T. G. Wong, “Equivalence of szegedyâĂŹs and coined quantum walks,” *Quantum Information Processing*, vol. 16, no. 9, p. 215, 2017.

[13] H. Krovi and T. A. Brun, “Hitting time for quantum walks on the hypercube,” *Phys. Rev. A*, vol. 73, p. 032341, Mar 2006. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.73.032341

[14] F. L. Marquezino, R. Portugal, G. Abal, and R. Donangelo, “Mixing times in quantum walks on the hypercube,” *Phys. Rev. A*, vol. 77, p. 042312, Apr 2008. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevA.77.042312

[15] O. L. Acevedo and T. Gobron, “Quantum walks on cayley graphs,” *Journal of Physics A: Mathematical and General*, vol. 39, no. 3, p. 585, 2005.

[16] S. Aaronson and A. Ambainis, “Quantum search of spatial regions,” in *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on.* IEEE, 2003, pp. 200–209.

[17] W. Dai, J. Yuan, and D. Li, "Discrete-time quantum walk on the cayley graph of the dihedral group," *arXiv preprint arXiv:1810.00158*, 2018.

[18] A. Ambainis, "Quantum walks and their algorithmic applications," *International Journal of Quantum Information*, vol. 1, no. 04, pp. 507–518, 2003.

[19] F. Magniez, A. Nayak, J. Roland, and M. Santha, "Search via quantum walk," *SIAM Journal on Computing*, vol. 40, no. 1, pp. 142–164, 2011.

[20] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, "Exponential algorithmic speedup by a quantum walk," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing.* ACM, 2003, pp. 59–68.

[21] T. A. Brun, H. A. Carteret, and A. Ambainis, "Quantum walks driven by many coins," *Physical Review A*, vol. 67, no. 5, p. 052317, 2002.