UCLouvain

FACULTE DES SCIENCES

Faculté des sciences

Classical Random Walks vs Quantum Walks

An empirical comparison of the mixing time on finite abelian groups

Auteure : Alyssa Noël

Promoteur : Jean-Charles Delvenne

Lecteurs : Marco Saerens et Pierre-Emmanuel Caprace

Année académique 2021-2022

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
FACULTÉ DES SCIENCES
ÉCOLE DE MATHÉMATIQUES

Classical random walks vs Quantum walks

An empirical comparison of the mixing time on finite abelian groups

---

Alyssa Noël

Advisor: Jean-Charles Delvenne
Readers: Marco Saerens, Pierre-Emmanuel Caprace

Master thesis
August 2022

Acknowledgements

I would like to thank my supervisor, Jean-Charles Delvenne, for suggesting this topic, taking the time to answer my questions and for his comments on the script. It has been a pleasure to delve into different domains of mathematics and applied mathematics and to combine a variety of concepts on this subject.

Special thanks also go to Léonie, Rodrigue and Pia, for reading this document, sharing their comments and ideas for improvement and being my friends for five years.

Finally, I thank my parents and my brother for supporting me and listening to me, even though most of the time, they did not understand what I was talking about.

Contents

Introduction

Random walks on graphs have applications in many domains. They are widely used in randomised algorithms for recommender systems such as PageRank, sampling problems like Gibbs sampling in statistical physics [9] or approximation problems, like approximating the volume of a convex body [18] or the permanent of a non-negative matrix [13], just to give a few examples. Under reasonable assumptions, a random walk converges to a limiting distribution, called the stationary distribution. We are interested in the convergence speed of the random walk in question, i.e. how many steps of the random walk it takes for the distribution to be sufficiently close to the stationary distribution. This convergence speed is measured by the mixing time. Many applications depend on a fast mixing time to the stationary distribution, so there has been a lot of research into reducing the mixing time via several different approaches. In this document, we will consider two approaches : lifted random walks and quantum random walks.

A lifted random walk is a random walk on what is called a lifted graph, which is obtained by taking each vertex of the original graph and splitting it into several vertices containing some information about the direction the walker came from. This way, we construct a graph with an extended state space and add some memory to the random walk to reduce the probability of the walker stepping back and forth between the same two vertices.

A quantum random walk is the quantum mechanical counterpart of a random walk. In contrast to the classical random walk where the evolution is described by a stochastic matrix, quantum random walks evolve trough unitary transformations. This allows for quantum effects such as superposition and interference. These effects can have a positive influence on the mixing time.

In the literature, classical random walks with memory and quantum walks are often treated separately. Our objective is to explore both the classical and the quantum case in parallel, introducing the concepts necessary to the understanding of classical as well as quantum random walks. Some results are stated without proof, as the proofs would be

too extensive to be included in a clear manner. In this case, the exact reference is cited. Throughout this work, we have take care to illustrate the construction of the different random walk types on examples. To compare classical random walks with memory and quantum walks, we will implement a generalisation of the Diaconis lift and two quantum walk models and empirically compare the mixing times for the different types of random walks on Cayley graphs of finite abelian groups.

To do so, in chapter 1 we will start by introducing the framework in which quantum random walks take place, namely the quantum computing model. The quantum computing model is based on quantum bits, or qubits, rather than the classical bits used in everyday classical computation. We start by defining qubits and systems of qubits. Then we describe how the state of a qubit system evolves as well as how to extract classical information from a system of qubits.

Chapter 2 introduces the vocabulary of finite Markov chains to describe classical random walks on graphs. We define the stationary distribution of a random walk and determine under which conditions a random walk converges to its stationary distribution. Next, we define the mixing time of a random walk and give upper and lower bounds for this mixing time. In section 2.2, we then define lifted random walks. A lifted walk can be designed under different sets of constraints given by the underlying non-lifted walk and these sets of constraints can affect the mixing time of the lifted walk, which we will briefly see at the end of the section.

There are many ways of defining a quantum random walk on an undirected graph. A first distinction to make is between discrete-time and continuous-time quantum walks. Throughout this work, we will concentrate on the discrete case. In chapter 3, we describe two quantum walk models, both of which rely on a so-called coin operator. First, the shunt-decomposition model, which is a walk on the vertices of a regular graph, and second the arc-reversal model on the arcs of an arbitrary undirected graph.

Since quantum walks evolve through unitary transformations, they do not converge to a limiting distribution in a straight forward way as the random walks in the classical framework. We will explain what is meant by limiting distribution in the quantum case in chapter 4. The limiting distribution can be expressed explicitly using the eigenvalues and eigenprojections of the unitary transition matrix. These will also be of use to obtain upper bounds on the quantum mixing time.

Our goal is to compare the mixing time of the different random walk models mentioned above on regular graphs. To this end, we have chosen the Cayley graphs of finite

abelian groups with symmetric generating set, as they allow a natural way to construct both lifted walks and quantum walks. In chapter 5, we generalise the Diaconis lift defined on the cycle [8] to Cayley graphs of finite abelian groups, which is the lifted walk used for the implementation. In the last section, we finally state the results of the empiric comparison of the mixing times between the different models.

Chapter 1

Introduction to quantum computing

Shor's quantum factorisation algorithm (1994), which runs exponentially faster than the best known classical algorithm for factorisation, as well as Grover's quantum search algorithm (1996) giving a quadratic speedup for the search of an unstructured database have sparked an interest in designing quantum algorithms, part of which are also quantum random walks. But to be able to understand quantum algorithms and quantum random walks we first need to know how quantum computing works. This chapter as well as some of the concepts recalled in appendix A aim to give an introduction to the quantum computing model. It is inspired by [1] and [19]. The reader is referred to [19] for a more detailed description of the mathematical and physical notions at hand.

1.1 Qubits and systems of qubits

In the classical computing model, the bit, or binary digit (0 or 1), is the basic unit of information. In the quantum computing model, we use quantum bits, or qubits, to carry information. A qubit is the generalisation of a classical bit to quantum mechanics.

Definition 1.1 (Qubit)**.** *A qubit is a quantum mechanical system described by the two-dimensional Hilbert space $\mathcal{H}$ with orthonormal basis $\{|0\rangle, |1\rangle\}$. The information contained in a qubit $|\phi\rangle \in \mathcal{H}$ is encoded in the qubit state*

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle,$$

where $\alpha, \beta \in \mathbb{C}$ *and* $|\alpha|^2 + |\beta|^2 = 1$. *A qubit satisfying this normalisation condition is a valid state.*

We can identify the orthonormal basis $\{|0\rangle, |1\rangle\}$ of $\mathcal{H}$ with the standard basis of $\mathbb{C}^2$,

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \text{ and } |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix},$$

due to the isomorphism $\mathcal{H} \cong \mathbb{C}^2$ (see A.3). Thus, the state of a qubit can also be presented as a two-dimensional complex vector,

$$|\phi\rangle = \alpha \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \beta \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}.$$

It is possible to construct higher-dimensional quantum systems based on subsystems. Let us consider the single qubit system $\mathcal{H}$. The two-qubit system can be described by $\mathcal{H} \otimes \mathcal{H} = \mathcal{H}^{\otimes 2}$, which is a Hilbert space of dimension 4 (see A.5) and proposition A.6) and an orthonormal basis of $\mathcal{H}^{\otimes 2}$ can be constructed as follows:

$$|00\rangle := |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|01\rangle := |0\rangle \otimes |1\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

$$|10\rangle := |1\rangle \otimes |0\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$|11\rangle := |1\rangle \otimes |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}.$$

This is the standard basis of the tensor product $\mathcal{H}^{\otimes 2}$. The combined state in the two-

qubit system of two single qubits

$$|\phi\rangle = \alpha|0\rangle + \beta|1\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \text{ and } |\psi\rangle = \gamma|0\rangle + \delta|1\rangle = \begin{pmatrix}\gamma\\ \delta\end{pmatrix},$$

is obtained by taking the tensor product

$$\begin{aligned}|\phi\rangle \otimes |\psi\rangle &= \alpha\gamma|00\rangle + \alpha\delta|01\rangle + \beta\gamma|10\rangle + \beta\delta|11\rangle \\ &= \begin{pmatrix}\alpha\\ \beta\end{pmatrix} \otimes \begin{pmatrix}\gamma\\ \delta\end{pmatrix} = \begin{pmatrix}\alpha\gamma\\ \alpha\delta\\ \beta\gamma\\ \beta\delta\end{pmatrix}.\end{aligned}$$

This is indeed a valid state in $\mathcal{H}^{\otimes 2}$, since it is still normalised :

$$\begin{aligned}|\alpha\gamma|^2 + |\alpha\delta|^2 + |\beta\gamma|^2 + |\beta\delta|^2 &= |\alpha|^2|\gamma|^2 + |\alpha|^2|\delta|^2 + |\beta|^2|\gamma|^2 + |\beta|^2|\delta|^2 \\ &= \left(|\alpha|^2 + |\beta|^2\right)\left(|\gamma|^2 + |\delta|^2\right) \\ &= 1.\end{aligned}$$

Moreover, we observe that taking the tensor product of the states of the single qubits is equivalent to computing the Kronecker product of the associated complex vectors.

Taking the tensor product $\mathcal{H}^{\otimes 2} \otimes \mathcal{H} = \mathcal{H}^{\otimes 3}$, we can construct the 3-qubit system. Taking the n-fold tensor product of $\mathcal{H}$ then yields an iterative way to construct an n-qubit system.

Definition 1.2 (n-qubit system)**.** *A n-qubit system is a quantum mechanical system described by the 2^n-dimensional Hilbert space $\mathcal{H}^{\otimes n} \cong \mathbb{C}^{2^n}$ with basis $\{|0\rangle, |1\rangle\}^{\otimes n}$.*

The state of an element in the n-qubit system is represented by a normalised complex vector of dimension 2^n.

1.1.1 Inner and outer product

We define the notions of inner and outer product in the single-qubit system. Those definitions can easily be generalised to n-qubit systems.

Notation 1.3. [1] Let $\bar{\alpha}, \bar{\beta}$ be the complex conjugates of $\alpha, \beta \in \mathbb{C}$. Then

$$|\phi\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix} \Leftrightarrow \langle\phi| = \begin{pmatrix} \bar{\alpha} & \bar{\beta} \end{pmatrix}.$$

We call $|\cdot\rangle$ the ket-notation and $\langle\cdot|$ the bra-notation, as introduced by Dirac. So $\langle\phi|$ is a linear form on $\mathcal{H}$.

Multiplying a bra-vector with a ket-vector yields the usual bracket notation for the inner product.

Definition 1.4 (Inner product)**.** *For two single-qubit states* $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ *and* $|\psi\rangle = \gamma|0\rangle + \delta|1\rangle$*, the inner product on* $\mathcal{H}$ *is defined as*

$$\langle\psi|\phi\rangle = \begin{pmatrix} \bar{\gamma} & \bar{\delta} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \bar{\gamma}\alpha + \bar{\delta}\beta \in \mathbb{C}.$$

The inner product of two quantum states yields a scalar quantity measuring the overlap of those states. We have

$$\langle\phi|\phi\rangle = |\alpha|^2 + |\beta|^2 = 1,$$

which represents the fact that $|\phi\rangle$ is maximally aligned with itself. The inner product between a given quantum state $|\phi\rangle$ and a basis state is given by the corresponding coefficient,

$$\begin{aligned} \langle 0|\phi\rangle &= \alpha\langle 0|0\rangle + \beta\langle 0|1\rangle = \alpha \\ \langle 1|\phi\rangle &= \alpha\langle 1|0\rangle + \beta\langle 1|1\rangle = \beta. \end{aligned}$$

The norm of $|\phi\rangle \in \mathcal{H}$ is given by

$$||\phi|| = \sqrt{\langle\phi|\phi\rangle}.$$

We do not use the Dirac notation inside the norm to keep the notations light.

By multiplying a ket-vector with a bra-vector, we obtain the outer product, which generates a matrix.

Definition 1.5 (Outer product)**.** *For two single-qubit states* $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ *and* $|\psi\rangle = \gamma|0\rangle + \delta|1\rangle$*, the outer product is defined as*

$$|\phi\rangle\langle\psi| = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}\begin{pmatrix}\bar{\gamma} & \bar{\delta}\end{pmatrix} = \begin{pmatrix}\alpha\bar{\gamma} & \alpha\bar{\delta}\\ \beta\bar{\gamma} & \beta\bar{\delta}\end{pmatrix}.$$

Outer products provide a way to write matrices in terms of bra-ket notations. Let $A \in \mathbb{C}^{2\times 2}$ be an operator acting on the single-qubit space $\mathcal{H}$, then

$$A = \begin{pmatrix}a_{00} & a_{01}\\ a_{10} & a_{11}\end{pmatrix} = a_{00}|0\rangle\langle 0| + a_{01}|0\rangle\langle 1| + a_{10}|1\rangle\langle 0| + a_{11}|1\rangle\langle 1|.$$

When A is applied to $|\phi\rangle$, this yields

$$\begin{aligned} A|\phi\rangle &= a_{00}|0\rangle\langle 0|\phi\rangle + a_{01}|0\rangle\langle 1|\phi\rangle + a_{10}|1\rangle\langle 0|\phi\rangle + a_{11}|1\rangle\langle 1|\phi\rangle \\ &= (a_{00}\alpha + a_{01}\beta)|0\rangle + (a_{10}\alpha + a_{11}\beta)|1\rangle. \end{aligned}$$

1.1.2 Superposition and entanglement

The **superposition principle** is the physical assumption that the combination of physically valid states is still physically valid. Mathematically, it means that any normalised linear combination of quantum states forms another valid quantum state. This explains why it is so convenient to work with Hilbert spaces. Every quantum state can be seen as an element of the unit ball of a Hilbert space and since this is a vector space, every linear combination is still in the unit ball of the Hilbert space and therefore automatically satisfies the superposition principle. Moreover, superposition implies that any quantum state can be obtained by a normalised linear combination of basis states. A single qubit for example is a superposition of $|0\rangle$ and $|1\rangle$ as can be seen in definition 1.1.

Superposition marks a huge difference between classical and quantum computing. In the classical case, a bit can be either 0 or 1. A qubit can be in the basis state $|0\rangle$ or $|1\rangle$ or any superposition of those states.

Proposition 1.6 (Entanglement)**.** *A state* $|\phi\rangle \in \mathcal{H}^{\otimes n}$ *is entangled if and only if there*

exist no states $|\phi_1\rangle \in \mathcal{H}, \ldots, |\phi_n\rangle \in \mathcal{H}$ *such that*

$$|\phi\rangle = |\phi_1\rangle \otimes \cdots \otimes |\phi_n\rangle.$$

So, an entangled state is a state that can not be written as the tensor product of single qubit states. This becomes more clear when looking at a few examples.

Example 1.7. Consider the state

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle).$$

Suppose there exist $\alpha, \beta, \gamma, \delta \in \mathbb{C}$ such that

$$\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) = (\alpha|0\rangle + \beta|1\rangle) \otimes (\gamma|0\rangle + \delta|1\rangle).$$

Then we would have

$$\begin{cases} \alpha\gamma &= \beta\delta = 0 \\ \alpha\delta &= \beta\gamma = \frac{1}{\sqrt{2}}, \end{cases}$$

which is impossible. Thus, $|\phi\rangle$ is an entangled state.

Example 1.8. Consider the state

$$|\psi\rangle = \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle).$$

We can write

$$\frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle) = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \otimes \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle).$$

So $|\psi\rangle$ is not entangled. Such a state is called **separable**.

1.2 Measurement

When we perform the measurement of a qubit, we transform the quantum information it contains into classical information. Every measurement is a projection and therefore affects the state of the system.

Proposition 1.9 (Projection postulate)**.** [19] *If the state of the system is given by* $|\phi\rangle \in \mathcal{H}$ *and we measure the system at eigenvalue* λ*, then the state of the system after the measurement is given by*

$$\frac{P_\lambda|\phi\rangle}{||P_\lambda|\phi\rangle||}.$$

To understand exactly what this means, we first need to introduce some vocabulary.

An **observable** is a measurable quantity of a quantum system, which is represented by a Hermitian operator A on the Hilbert space $\mathcal{H}$. The possible measurement results of an observable A are given by the eigenvalues $\{\lambda_i\}_{i\in I}$ of A. Suppose we perform a measurement on a state $|\phi\rangle$. Then the **expectation value** of the observable A on the state $|\phi\rangle$ is defined as

$$\langle A\rangle_\phi = \langle\phi|A\phi\rangle.$$

If all eigenvalues $\{\lambda_i\}_{i\in I}$ are distinct and $\{|e_i\rangle\}$ are the associated orthonormal eigenvectors, by spectral decomposition (see A.9), $A = \sum_i \lambda_i|e_i\rangle\langle e_i|$ and

$$\langle A\rangle_\phi = \sum_i \lambda_i|\langle e_i|\phi\rangle|^2.$$

We can interpret the quantity $|\langle e_i|\phi\rangle|^2$ as the probability to measure the value λ_i. The expectation value can then be seen as the average over all the possible outcomes of the measurement weighted by their probability of occurrence. The following definition is true for the general case, where the eigenvalues are not necessarily all distinct.

Definition 1.10 (Measurement probability)**.** *Let* A *be an observable. The probability to measure the eigenvalue* λ_i *for a system in the state* $|\phi\rangle$ *is given by*

$$\Pr_\phi(\lambda_i) = ||P_{\lambda_i}|\phi\rangle||^2,$$

where P_{λ_i} *is the orthogonal projection onto the eigenspace associated to* λ_i*.*

This definition does indeed define a probability distribution on the set of eigenvalues of A, $\{\lambda_i\}_{i\in I}$. Let d_i be the dimension of the eigenspace associated to the eigenvalue λ_i and $\{|e_{i,k}\rangle | i \in I \text{ and } k \in \{1,\dots,d_i\}\}$ an orthonormal basis of eigenvectors. Then

$P_{\lambda_i} = \sum_{k=1}^{j_i} |e_{i,k}\rangle\langle e_{i,k}|$ and

$$\begin{aligned}\Pr_\phi(\lambda_i) &= ||P_{\lambda_i}|\phi\rangle||^2 \\ &= \left|\left| \sum_{k=1}^{d_i} |e_{i,k}\rangle\langle e_{i,k}|\phi\rangle \right|\right|^2 \\ &= \sum_{k=1}^{d_i} \overline{\langle e_{i,k}|\phi\rangle}\langle e_{i,k}|e_{i,k}\rangle\langle e_{i,k}|\phi\rangle \\ &= \sum_{k=1}^{d_i} |\langle e_{i,k}|\phi\rangle|^2 \geq 0.\end{aligned}$$

Moreover,

$$\begin{aligned}\sum_{i\in I} \Pr_\phi(\lambda_i) &= \sum_{i\in I}\sum_{k=1}^{d_i} |\langle e_{i,k}|\phi\rangle|^2 \\ &= ||\phi||^2 = 1.\end{aligned}$$

Proposition 1.11. [19] *The probability of measuring a system in state $|\phi\rangle$ in state $|\psi\rangle$ is given by*

$$\Pr_\phi(|\psi\rangle) = |\langle\psi|\phi\rangle|^2.$$

Proof. Let $|\phi\rangle, |\psi\rangle \in \mathcal{H}$ such that $||\phi|| = 1 = ||\psi||$ and take as observable the orthogonal projection $P_\psi = |\psi\rangle\langle\psi|$ onto the state $|\psi\rangle$.

Since P_ψ is an orthogonal projection, thus idempotent, the eigenvalues satisfy $\lambda^2 = \lambda$ and we deduce that the observable has eigenvalues 0 and 1. Now,

$$P_\psi|\psi\rangle = |\psi\rangle,$$

so $\lambda = 1$ is a non-degenerate eigenvalue with eigenvector $|\psi\rangle$ and the projection onto the eigenspace of $\lambda = 1$ is $P_1 = |\psi\rangle\langle\psi|$. By definition 1.10,

$$\Pr_\phi(1) = ||P_1|\phi\rangle||^2 = |||\psi\rangle\langle\psi|\phi\rangle||^2 = ||\psi||^2|\langle\psi|\phi\rangle|^2 = |\langle\psi|\phi\rangle|^2.$$

□

In the two-dimensional Hilbert space $\mathcal{H}$ with orthonormal basis $\{|0\rangle, |1\rangle\}$ represent-

ing a single qubit system, an interesting observable is

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} = |0\rangle\langle 0| - |1\rangle\langle 1|.$$

Z has eigenvalue 1 with associated eigenvector $|0\rangle$ and eigenvalue -1 with eigenvector $|1\rangle$. So, a measurement of Z projects the qubit onto a state of the orthonormal basis of $\mathcal{H}$. For a qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$, by proposition 1.11,

$$\begin{aligned} \Pr_\phi(|0\rangle) &= \Pr_\phi(\lambda = 1) = |\langle 0|\phi\rangle|^2 = |\alpha|^2 \\ \Pr_\phi(|1\rangle) &= \Pr_\phi(\lambda = -1) = |\langle 1|\phi\rangle|^2 = |\beta|^2. \end{aligned}$$

The observable Z allows to establish a link between classical and quantum bits. If the result of a measurement of Z is 1, then the quantum system is in the state $|0\rangle$, which represents the classical bit 0. On the other hand, if the measurement of Z yields -1, by the projection postulate, the state of the quantum system is $|1\rangle$, representing the classical bit value 1. A general qubit state $|\phi\rangle = \alpha|0\rangle + \beta|1\rangle$ has no corresponding classical bit value.

When performing a measurement in a multiple qubit system, we do not need to measure the whole system, we can choose to do a **partial measurement**. In a two-qubit system with $|\phi\rangle := |\phi_1, \phi_2\rangle \in \mathcal{H}^{\otimes 2}$ for example, we obtain the probability to measure the first qubit in state $|0\rangle$ by

$$\Pr_\phi(\psi_1 = 0) = \textstyle\sum_{\psi_2 \in \{0,1\}} |\langle 0, \psi_2|\phi_1, \phi_2\rangle|^2.$$

1.3 Unitary transformations

When applying a transformation to a qubit, we need to make sure that the resulting qubit state is still a valid state of norm 1. The transformations allowing this are the unitary operators.

Definition 1.12 (Unitary operator)**.** [19] *An operator $U : \mathcal{H} \to \mathcal{H}$ is a unitary if for all $|\psi\rangle, |\phi\rangle \in \mathcal{H}$,*

$$\langle U\psi|U\phi\rangle = \langle\psi|\phi\rangle.$$

Proposition 1.13. [19] *The following are equivalent,*

(i) U *is unitary*

(ii) $U^*U = \mathrm{Id} = UU^*$

(iii) $||U\psi|| = ||\psi||$ *for all* $|\psi\rangle \in \mathcal{H}$.

Proof. First, we suppose that U is unitary and we show that $||U\psi|| = ||\psi||$ for all $|\psi\rangle \in \mathcal{H}$. By definition,

$$||U\psi|| = \sqrt{\langle U\psi|U\psi\rangle} = \sqrt{\langle\psi|\psi\rangle} = ||\psi||.$$

Using $||U\psi|| = ||\psi||$, we can now prove that for any $U^*U = \mathrm{Id}$. Indeed,

$$\begin{aligned} ||U\psi|| = ||\psi|| &\implies \sqrt{\langle U\psi|U\psi\rangle} = \sqrt{\langle\psi|\psi\rangle} \\ &\implies \sqrt{\langle U^*U\psi|\psi\rangle} = \sqrt{\langle\psi|\psi\rangle} \\ &\implies U^*U = \mathrm{Id}\,. \end{aligned}$$

Finally, we demonstrate that $U^*U = \mathrm{Id}$ implies that U is unitary, since for all $|\psi\rangle, |\phi\rangle \in \mathcal{H}$,

$$\langle U\psi|U\phi\rangle = \langle U^*U\psi|\phi\rangle = \langle\psi|\phi\rangle.$$

□

Proposition 1.14. *If* U_1 *and* U_2 *are two unitary matrices acting on the same space, then* U_1U_2 *is a unitary matrix on the combined state space.*

Indeed,

$$(U_1U_2)^*(U_1U_2) = U_2^*U_1^*U_1U_2 = U_2^*U_2 = I.$$

Proposition 1.15. *If* U_1 *and* U_2 *are two unitary matrices, then* $U_1 \otimes U_2$ *is a unitary matrix.*

Indeed,

$$(U_1 \otimes U_2)^*(U_1 \otimes U_2) = (U_1^* \otimes U_2^*)(U_1 \otimes U_2) = (U_1^*U_1) \otimes (U_2^*U_2) = I_1 \otimes I_2.$$

This proposition allows us to construct operators for systems of more than one qubit. If U_1 and U_2 act on separate qubits, then the transformation on the total two-qubit system is given by $U_1 \otimes U_2$, where $\otimes$ is the Kronecker product between the matrices U_1 and U_2.

Quantum transformations satisfy the relation

$$U|\psi\rangle = |\phi\rangle \Leftrightarrow U^*|\phi\rangle = |\psi\rangle,$$

since the inverse of a unitary transformation U is given by its conjugate transpose U^*. This implies that all transformations on qubits (except measurement) are reversible, which is not the case for transformations on classical bits.

1.4 Mixed states and density matrices

So far, we have only encountered pure states, which are a normalised superposition of basis states. A mixed state is a statistical ensemble of pure states and can not be written as a normalised linear combination of pure states.

Consider example 1.7 again, with the state $|\phi\rangle = \frac{1}{\sqrt{2}}(|0_A 1_B\rangle + |1_A 0_B\rangle$ in the combined system $\mathcal{H}^{\otimes 2} = \mathcal{H}_A \otimes \mathcal{H}_B$. Now, we can perform a partial measurement on the first qubit, in $\mathcal{H}_A$. Two cases arise :

- the probability to find the first qubit equal to 0_A is given by

  $$\sum_{\psi_B \in \{0,1\}} |\langle 0\psi_B|\phi\rangle|^2 = \left|\frac{1}{\sqrt{2}}\langle 00|01\rangle + \frac{1}{\sqrt{2}}\langle 00|10\rangle\right|^2 + \left|\frac{1}{\sqrt{2}}\langle 01|01\rangle + \frac{1}{\sqrt{2}}\langle 01|10\rangle\right|^2 = \frac{1}{2}.$$

  But if we measure the first qubit in state $|0_A\rangle$, then the combined state has to be $|0_A 1_B\rangle$.

- the probability to find the first qubit equal to 1_A is given by

  $$\sum_{\psi_B \in \{0,1\}} |\langle 0\psi_B|\phi\rangle|^2 = \left|\frac{1}{\sqrt{2}}\langle 00|01\rangle + \frac{1}{\sqrt{2}}\langle 00|10\rangle\right|^2 + \left|\frac{1}{\sqrt{2}}\langle 01|01\rangle + \frac{1}{\sqrt{2}}\langle 01|10\rangle\right|^2 = \frac{1}{2}.$$

  So if we measure the first qubit in state $|1_A\rangle$, then the combined state has to be $|1_A 0_B\rangle$.

Then, after the measurement on $\mathcal{H}_A$, the qubit ϕ_B in $\mathcal{H}_B$ is in the pure state $|1_B\rangle$ with probability $\frac{1}{2}$ and in pure state $|0_B\rangle$ with probability $\frac{1}{2}$. ϕ_B is a mixed state (which is not written using the Dirac notation).

So, when we have a mixed state, there are different possible outcomes, which are pure states, and each of these outcomes has a certain probability to occur. This has to be distinguished from a system being in a pure state $|\phi\rangle = \sum_i \frac{1}{\sqrt{p_i}|\phi_i\rangle}$, which is a superposition of states. In this case, the system is described by $|\phi\rangle$. Density matrices allow us to represent both pure and mixed states.

Definition 1.16 (Density matrix)**.** *A density matrix is a Hermitian positive semidefinite matrix ρ with* $\mathrm{Tr}(\rho) = 1$.

The definitions of Hermitian matrix, positive semidefinite matrix and trace are recalled in A.8, A.10 and A.11. In the following, we will use the operator norm induced by the Frobenius inner product.

Definition 1.17 (Frobenius norm)**.** *The Frobenius inner product between two operators A and B is defined as*

$$\langle A|B\rangle = \mathrm{Tr}(A^*B),$$

which induces the norm

$$||A|| = \sqrt{\mathrm{Tr}(A^*A)}. \tag{1.1}$$

Consider a mixed state where $|\phi_i\rangle$ are the possible pure states and p_i the probability of each possible outcome $|\phi_i\rangle$. Then the density matrix for this mixed state is given by

$$\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|.$$

The density matrix of the mixed state ϕ_B in the example above is therefore given by

$$\rho_B = \frac{1}{2}|0_B\rangle\langle 0_B| + \frac{1}{2}|0_B\rangle\langle 0_B| = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix}.$$

Every density matrix satisfies the following properties.

Proposition 1.18. [19] *For every density matrix ρ,*

(i) there exist $p_i \in \mathbb{R}$ *such that* $0 \leq p_i$ *and* $\sum_i p_i = 1$ *and there exists an orthonormal basis* $\{|\phi_i\rangle\}$ *such that*

$$\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|,$$

(ii) $0 \leq \rho^2 \leq \rho$,

(iii) $||\rho|| \leq 1$.

Proof. (i) As ρ is Hermitian by definition, all its eigenvalues p_i are real and we can find orthonormal eigenvectors $\{|\phi_i\rangle\}$ such that

$$\rho = \sum_i p_i |\phi_i\rangle\langle\phi_i|.$$

ρ is also positive semidefinite, which implies for all i,

$$0 \leq \langle\phi_i|\rho\phi_i\rangle = \sum_j p_j \langle\phi_i|\phi_j\rangle\langle\phi_j|\phi_i\rangle = p_i.$$

From $\mathrm{Tr}(\rho) = 1$, we deduce

$$1 = \mathrm{Tr}(\rho) = \sum_i \langle\phi_i|\rho\phi_i\rangle = \sum_i \sum_j p_j \langle\phi_i|\phi_j\rangle\langle\phi_j|\phi_i\rangle = \sum_i p_i.$$

(ii) First, we show that $\rho^2 \geq 0$. Indeed, for any $|\phi\rangle \in \mathcal{H}$,

$$\langle\phi|\rho^2\phi\rangle = \langle\rho^*\phi|\rho\phi\rangle = \langle\rho\phi|\rho\phi\rangle = ||\rho\phi||^2 \geq 0.$$

Next, we need to show that $\rho \geq \rho^2$, which equals showing $\rho - \rho^2 \geq 0$. We have that

$$\rho^2 = \left(\sum_i p_i |\phi_i\rangle\langle\phi_i|\right)^2 = \sum_i \sum_j p_i p_j |\phi_i\rangle\langle\phi_i|\phi_j\rangle\langle\phi_j| = \sum_i p_i^2 |\phi_i\rangle\langle\phi_i|.$$

Therefore, for all $|\phi\rangle \in \mathcal{H}$,

$$\langle\phi|(\rho - \rho^2)\phi\rangle = \sum_i p_i^2 \langle\phi|\phi_i\rangle\langle\phi_i|\phi\rangle = \sum_i p_i^2 |\langle\phi|\phi_i\rangle|^2 \geq 0.$$

(iii) Finally, let us show that $||\rho|| \leq 1$. By the definition of the Frobenius norm,

$$||\rho||^2 = \mathrm{Tr}(\rho^*\rho) = \mathrm{Tr}(\rho^2) = \sum_i \langle\phi_i|\rho^2\phi_i\rangle = \sum_i \sum_j p_j^2 \langle\phi_i|\phi_j\rangle\langle\phi_j|\phi_i\rangle = \sum_i p_i^2 \leq 1,$$

because $0 \leq p_i \leq 1$ for all i. □

We can also represent pure states using density matrices. If $|\phi\rangle \in \mathcal{H}$ is a pure state, then the probability of occurrence is $p = 1$ and the associated density matrix is $\rho = |\phi\rangle\langle\phi|$.

In the case of mixed states represented by a density operator, the system changes through $\rho \mapsto U\rho U^*$. We need to make sure that this is a valid mixed state, i.e. $U\rho U^*$ is again a density operator, which is assured by the following property.

Proposition 1.19. [19] *If ρ is a density operator and U a unitary transformation, then $U\rho U^*$ is a density operator.*

Proof. We need to verify that $U\rho U^*$ is Hermitian, positive semidefinite and of trace 1. Since $\rho^* = \rho$,

$$(U\rho U^*)^* = (U^*)^*\rho^* U^* = U\rho U^*.$$

Now, let $|\phi\rangle \in \mathcal{H}$. Then

$$\langle\phi|(U\rho U^*)\phi\rangle = \langle U^*\phi|\rho(U^*\phi)\rangle \geq 0$$

because ρ is positive semidefinite. Last, by the cyclic property of the trace (see A.12) and since U is unitary,

$$\mathrm{Tr}(U\rho U^*) = \mathrm{Tr}(U^*U\rho) = \mathrm{Tr}(\rho) = 1.$$

□

Measurement rules can also be generalised to mixed states and the density matrix representation. Let $\rho = \sum_i p_i|\phi_i\rangle\langle\phi_i|$ and A an observable. Then the expectation value in mixed state ρ is given by

$$\langle A\rangle_\rho = \mathrm{Tr}(\rho A) = \sum_i p_i \langle A\rangle_{\phi_i}.$$

The measurement probability for mixed states is defined as

$$\Pr_\rho(\lambda) = \mathrm{Tr}(\rho P_\lambda) = \langle \rho | P_\lambda \rangle$$

and the projection postulate for mixed states transforms into

$$\rho \xmapsto{measurement} \frac{P_\lambda \rho P_\lambda}{\mathrm{Tr}(\rho P_\lambda)}.$$

Chapter 2

Classical random walks

The many applications of random walks in randomised algorithms and the need for random walks with fast mixing lead us to the first approach attempting to reduce the mixing time : lifted random walks. Before we define lifted random walks in section 2.2, in section 2.1 of this chapter we start by introducing the formalism of finite Markov chains to describe random walks, stationary distributions and mixing times.

2.1 Classical random walks

A classical random walk on a graph can be described as a finite Markov chain. This section follows the introduction to finite Markov chains and mixing times given in [17].

Definition 2.1 (Finite Markov chain)**.** *A finite Markov chain is a sequence of random variables* $(X_0, X_1, X_2, \dots)$ *taking values in a finite state space* $\mathcal{X}$ *and evolving according to a transition matrix* $P = (p_{ij})_{i,j\in\mathcal{X}}$ *such that*

$$\Pr(X_{t+1} = j | X_t = i, X_{t-1} = x_{t-1}, \dots, X_0 = x_0) = \Pr(X_{t+1} = j | X_t = i) =: p_{ij}. \quad (2.1)$$

Property (2.1) implies that the transition from time step t to $t+1$ only depends on X_t and none of the previous steps, the chain is memoryless. As p_{ij} does not depend on t, then we call the Markov chain **homogeneous**. In the following, all the Markov chains we consider will be finite homogeneous Markov chains.

P is a stochastic matrix and $P_{i:}$, the i^{th} row of P, defines a probability distribution on the state space $\mathcal{X}$ given that the current state is i. Thus, for all $i, j \in \mathcal{X}$, $p_{ij} \geq 0$ and

$$\sum_{j \in \mathcal{X}} p_{ij} = 1$$

for any $i \in \mathcal{X}$. We denote μ_t the probability distribution over $\mathcal{X}$ after t steps, i.e. $\mu_t(i) = \Pr(X_t = i)$. Then

$$\mu_{t+1} = \mu_t P = \mu_0 P^t. \tag{2.2}$$

Alternatively, we have

$$\mu_{t+1}(j) = \sum_{i \in \mathcal{X}} \mu_t(i) p_{ij},$$

summing over the probability to be in state i after t steps and to move from state i to j at step $t + 1$.

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. We will only consider graphs with no multiple edges. The vertices u and v are neighbours, we write $u \sim v$, if $\{u, v\} \in E$ and the degree $\deg(u)$ is the number of neighbours of u.

A classical random walk on G can be described as a homogeneous Markov chain with state space the set of vertices V. The edges between vertices represent feasible transitions between states and transitions occur according to some probability distribution. The sequence $(X_0, X_1, X_2, \dots)$ can be seen as the path of a walker on the graph G.

In the **simple random walk** on G, at each vertex u, the walker picks one of the neighbours of u uniformly at random, which yields the transition matrix $P = (p_{uv})_{u,v \in V}$ with

$$p_{uv} = \begin{cases} \frac{1}{\deg(u)}, & \text{if } u \sim v. \\ 0, & \text{otherwise.} \end{cases}$$

If G is a weighted graph, we can take the weight on each edge into account to construct the transition matrix. Let w_{uv} be the weight assigned to the edge $\{u, v\}$, then we define

$$p_{uv} = \begin{cases} \frac{w_{uv}}{\sum_{v \in V} w_{uv}}, & \text{if } u \sim v. \\ 0, & \text{otherwise.} \end{cases}$$

A classical random walk on a graph G is completely determined by its transition matrix P and the initial distribution on the position of the walker μ_0, due to equation (2.2).

2.1.1 Stationary distribution

We are interested in the long term behaviour of the probability distribution μ_t on the state space. What happens when $t \to \infty$? We will show that under reasonable conditions, μ_t converges to a stationary distribution.

Definition 2.2 (Stationary distribution)**.** *A probability distribution π on $\mathcal{X}$ satisfying the property*

$$\pi = \pi P \tag{2.3}$$

is called a stationary distribution.

We can also express the property (2.3) elementwise as

$$\pi_j = \sum_{i \in \mathcal{X}} \pi_i p_{ij} \tag{2.4}$$

for all $j \in \mathcal{X}$.

Example 2.3. Consider the simple random walk on a graph G, where $p_{uv} = \frac{1}{\deg(u)}$ if $u \sim v$. Let $m := \sum_{u \in V} \deg(u)$. Then $\pi_u = \frac{\deg(u)}{m}$ is the stationary distribution for the simple random walk on G. Indeed,

$$\sum_{u \in V} \pi_u p_{uv} = \sum_{u \sim v} \frac{\deg(u)}{m} \frac{1}{\deg(u)} = \frac{\deg(v)}{m} = \pi_v$$

for all $v \in V$, which satisfies equation (2.4).

In particular, if G is a d-regular graph, i.e. $\deg(u) = d$ for all $u \in V$, then the stationary distribution is the uniform distribution over V, since $m = d|V|$. Thus for all $v \in V$,

$$\pi_v = \frac{d}{d|V|} = \frac{1}{|V|}. \tag{2.5}$$

We would like to show that under reasonable conditions, a Markov chain converges to its stationary distribution. To prove this, we need to introduce some definitions.

Definition 2.4 (Irreducible Markov chain)**.** *A Markov chain with transition matrix P is irreducible if for all $i, j \in \mathcal{X}$, there exists an integer t such that $(P^t)_{ij} > 0$.*

Irreducibility of a Markov chain means that any state can be reached by any other state using a sequence of transitions of positive probability. Any chain representing a random walk on an undirected graph is irreducible if the underlying graph is connected.

Existence and uniqueness of a stationary distribution for irreducible Markov chains is admitted. A detailed proof can be found in [17, Sections 1.5.3 and 1.5.4].

Let $\tau(i) = \{t \geq 1 : (P^t)_{ii} > 0\}$. $\tau(i)$ is the set of times where it is possible to return to the starting state i. The period of a state i is the greatest common divisor of the elements $\tau(i)$ and the period of an irreducible Markov chain is the period common to all states.

Definition 2.5 (Aperiodic Markov chain)**.** *A Markov chain is aperiodic if all states have period* 1*.*

To be able to compare probability distributions and talk about convergence, we need to define a distance between probability distributions, which will be the total variation distance.

Definition 2.6 (Total variation distance)**.** *Let μ and ν be two probability distributions. The total variation distance is defined as*

$$||\mu - \nu||_{TV} = \frac{1}{2} \sum_{i \in \mathcal{X}} |\mu_i - \nu_i|.$$

Now, we have everything we need to formulate the convergence theorem.

Theorem 2.7 (Convergence theorem [17])**.** *Let P be the transition matrix of an irreducible, aperiodic Markov chain with stationary distribution π. Then*

$$\lim_{t \to \infty} \max_{i \in \mathcal{X}} ||(P^t)_{i:} - \pi||_{TV} = 0.$$

Because μ_0 is a probability distribution over the states $i \in \mathcal{X}$, we have

$$\max_{i \in \mathcal{X}} ||(P^t)_{i:} - \pi|| = \sup_{\mu_0} ||\mu_0 P^t - \pi||. \tag{2.6}$$

Therefore, theorem 2.7 implies that $\mu_t = \mu_0 P^t$ converges to π for any initial distribution μ_0.

The following lemma will be used in the proof of the convergence theorem.

Proposition 2.8. [17] *Let P be the transition matrix of an irreducible and aperiodic Markov chain. Then there exists an integer $T \geq 0$ such that for all $i, j \in \mathcal{X}$ and $t > T$,*

$$(P^t)_{ij} > 0.$$

Proof. From number theory, we know that any set of integers S such that $s \geq 0$ for all $s \in S$, which is closed under addition and has $\gcd(S) = 1$, contains all but finitely many of the non-negative integers [17, Lemma 1.30].

First, we show that $\tau(i) = \{t \geq 1 : (P^t)_{ii} > 0\}$ is closed under addition and has greatest common divisor 1. By assumption, the chain is aperiodic, hence $\gcd(\tau(i)) = 1$ for all $i \in \mathcal{X}$. Let $s, t \in \tau(i)$, then

$$(P^{s+t})_{ii} \geq (P^s)_{ii}(P^t)_{ii} > 0$$

and $s + t \in \mathcal{X}$. Thus, there exists $t(i)$ such that for all $t > t(i)$, we have $t \in \tau(i)$.

Since the chain is irreducible, for all $i, j \in \mathcal{X}$, there exists r (depending on i and j), such that $(P^r)_{ij} > 0$. For $t \geq t(i) + r$,

$$(P^t)_{ij} \geq (P^{t-r})_{ii}(P^r)_{ij} > 0.$$

Let $T = \max_{i \in \mathcal{X}}\{t(i) + \max_{j \in \mathcal{X}} r\}$. Then for all $i, j \in \mathcal{X}$ and $t \geq T$,

$$(P^t)_{ij} > 0.$$

□

Proof of the Convergence Theorem. By proposition 2.8, there exists T such that $(P^T)_{ij} > 0$ for all $i, j \in \mathcal{X}$. Then there also exists $\delta > 0$ such that

$$(P^T)_{ij} \geq \delta \pi_j.$$

Let Π be a square matrix containing the stationary distribution π in each row. We can write

$$P^T = \delta \Pi + (1 - \delta) Q, \tag{2.7}$$

where Q is a stochastic matrix.

Let $\theta = 1 - \delta$. We show by induction that

$$P^{Tk} = (1 - \theta^k)\Pi + \theta^k Q^k. \tag{2.8}$$

The case $k = 1$ is true by (2.7à. Supposing the equality is true for $k = n$,

$$\begin{aligned}
P^{T(n+1)} &= P^{Tn} P^T \\
&= ((1 - \theta^n)\Pi + \theta^n Q^n) P^T \\
&= (1 - \theta^n)\Pi P^T + \theta^n Q^n((1 - \theta)\Pi + \theta Q) \\
&= (1 - \theta^n)\Pi P^T + \theta^n (1 - \theta) Q^n \Pi + \theta^{n+1} Q^{n+1} \\
&= (1 - \theta^n)\Pi + \theta^n (1 - \theta)\Pi + \theta^{n+1} Q^{n+1} \\
&= (1 - \theta^{n+1})\Pi + \theta^{n+1} Q^{n+1},
\end{aligned}$$

where we have used that π is the stationary distribution for P, thus $\Pi P^T = \Pi$, and since Q is a stochastic matrix, $Q^n \Pi = \Pi$.

By multiplying both sides of (2.8) by P^ℓ and rearranging the terms, we obtain

$$P^{Tk+\ell} - \Pi = \theta^k (Q^k P^\ell - \Pi),$$

which implies for all $i, j \in \mathcal{X}$,

$$|(P^{Tk+\ell})_{ij} - \pi_j| = \theta^k |(Q^k P^\ell)_{ij} - \pi_j|.$$

Summing both sides over j and multiplying them by $\frac{1}{2}$ yields

$$||(P^{Tk+\ell})_{i:} - \pi|| = \theta^k ||(Q^k P^\ell)_{i:} - \pi|| \leq \theta^k$$

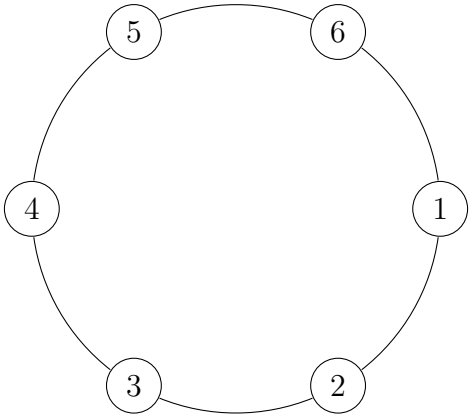


Figure 2.1: 6-cycle

for all $i \in \mathcal{X}$, since $||(Q^k P^\ell)_{i:} - \pi|| \leq 1$. Because $0 < \theta < 1$, θ^k tends to 0 as $k \to \infty$, which concludes the proof. □

Aperiodicity is indeed a necessary condition for this theorem to work.

Example 2.9. Consider the simple random walk on a 6-cycle represented in figure 2.1, which has transition matrix

$$P = \begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 0 \end{pmatrix}.$$

The stationary distribution of this random walk is $\pi = (\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6})$, since the 6-cycle is a regular graph with 6 vertices (see equation 2.5). However, if the walk starts on an even vertex, then for all $i \in \{1, \ldots, 6\}$,

$$(P^{2t})_{ij} = 0 \text{ if } j \text{ is even},$$
$$(P^{2t+1})_{ij} = 0 \text{ if } j \text{ is odd},$$

because this walk is of period 2. So, $(P^t)_{i:}$ can never converge to π.

2.1.2 Mixing time

Now that we have established that irreducible and aperiodic Markov chains converge to a limiting distribution, which coincides with the stationary distribution, we are interested in the convergence speed of a Markov chain to its limiting distribution, which is measured by the mixing time. Let

$$d(t) := \sup_{\mu_0} ||\mu_0 P^t - \pi||.$$

Definition 2.10 (Mixing time)**.** *Given $\varepsilon > 0$, the mixing time is defined as the smallest integer at which μ_t gets ε-close to π for any initial distribution μ_0,*

$$M_\varepsilon = \min\{T | \forall t \geq T : d(t) \leq \varepsilon\}.$$

In the literature, usually $\varepsilon = \frac{1}{4}$. We can relate $M_{\frac{1}{4}}$ to M_ε for any other ε via the following equation,

$$M_\varepsilon \leq \log_2\left(\frac{1}{\varepsilon}\right) M_{\frac{1}{4}}.$$

There are many different ways to bound the mixing time of a Markov chain. The mixing time of a reversible, irreducible and aperiodic Markov chain can be upper and lower bounded using the absolute spectral gap. Let P be the transition matrix of a given Markov chain. Then P is said to be **reversible** if

$$\pi_i p_{ij} = \pi_j p_{ji}$$

for all $i, j \in \mathcal{X}$. The simple random walk and the random walk on a weighted graph are examples of reversible Markov chains. Let $\{\lambda_i\}$ be the set of eigenvalues of P such that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$. Since P is a non-negative matrix describing an irreducible and aperiodic Markov chain, $\lambda_1 = 1$ and -1 is not an eigenvalue of P [17, Lemma 12.1]. Let $\lambda_* = \max\{|\lambda|, \lambda \neq 1\}$. The absolute spectral gap is given by $1 - \lambda_*$.

Theorem 2.11 (Spectral gap bound on the mixing time)**.** [17, Theorem 12.4 and 12.5] *Let $\varepsilon > 0$. For an irreducible, aperiodic and reversible Markov chain,*

$$\frac{\lambda_*}{1-\lambda_*}\frac{1}{\log(2\varepsilon)} \leq M_\varepsilon \leq \frac{1}{1-\lambda_*}\log\left(\frac{1}{\varepsilon \pi_{min}}\right),$$

where $\pi_{min} = \min_{i \in \mathcal{X}} \pi_i$.

We can also relate the mixing time of a random walk to the geometrical structure of the underlying graph G through the conductance. The conductance measures how difficult it is for the walker to leave a subset of vertices.

Definition 2.12 (Conductance)**.** *Let P be the transition matrix of a Markov chain with stationary distribution π. For $S \subset \mathcal{X}$, let $\pi(S) = \sum_{i \in S} \pi_i$. The conductance is defined as*

$$\Phi(P) = \min_{\substack{S \subset \mathcal{X} \\ \pi(S) \leq \frac{1}{2}}} \frac{\sum_{i \in S, j \notin S} \pi_i p_{ij}}{\pi(S)}.$$

Theorem 2.13. [17, Theoremm 7.4] *Let $\varepsilon > 0$ and $\Phi(P)$ be the conductance of the Markov chain P. Then*

$$M_\varepsilon \geq \frac{1}{4\Phi(P)}.$$

However, both the spectral gap and the conductance of a random walk on a graph are difficult to calculate in general.

2.2 Lifted random walks

Let $G = (V, E)$ be an undirected connected graph with n vertices and P the transition matrix of a random walk on G with stationary distribution π. If P for example describes the simple random walk on G, it is possible that the walker moves from a vertex u to v and back to u at the next step, which increases the mixing time. We would like to develop methods to decrease the mixing time of a random walk. One possibility are lifted Markov chains, which provide some memory to the random walk and reduce the probability of the walker going directly back from where it came. A lifted Markov chain is a Markov chain on a lifted graph with an extended state space. Chen, Lovász and Pak established bounds on the mixing time of lifted walks in [7]. Apers, Ticozzi and Sarlette later studied lifted Markov chains and their mixing time in a broader context in [5], studying the constraints on the lifting that affect the mixing time. The bound established in [7] is true for one set of constraints elaborated in [5]. This section is based on both of these papers.

Definition 2.14 (Lifted graph)**.** *Let $G = (V, E)$ be a graph with n vertices. Then $\hat{G} = (\hat{V}, \hat{E})$ with $\hat{n}$ vertices is a lifted graph of G if there exists a surjective map $c : \hat{V} \to V$ such that $(i, j) \in \hat{E}$ only if $(c(i), c(j)) \in E$.*

The map c can be seen as a projection of the graph $\hat{G}$ onto the underlying graph G and the map c^{-1} associates to each vertex u of G the set of vertices in $\hat{G}$ that are projected onto u, i.e. all $i \in \hat{G}$ such that $c(i) = u$. The map c can be represented as a matrix C of dimension $\hat{n} \times n$ with

$$C_{iu} = \begin{cases} 1 & \text{if } c(i) = u \\ 0 & \text{otherwise.} \end{cases}$$

Let ν denote a probability distribution on the set of vertices $\hat{V}$ of the lifted graph. We can obtain the associated marginal distribution μ on the vertices of the underlying graph by

$$\mu = \nu C. \tag{2.9}$$

By definition of C, this equation is equivalent to $\mu(u) = \sum_{i \in c^{-1}(u)} \nu(i)$.

On the lifted graph $\hat{G}$, we perform a random walk evolving through repeated application of a stochastic matrix A,

$$\nu_{t+1} = \nu_t A = \nu_0 A^t,$$

for an initial distribution ν_0. A needs to satisfy the locality constraint of the underlying graph G, which means that a transition from i to j in $\hat{G}$ is only admissible if the edge $(c(i), c(j))$ exists in G. Therefore, if $(c(i), c(j)) \notin E$, we necessarily have $A_{ij} = 0$. The distribution μ_t on the vertices of the underlying graph G after t steps of the random walk on $\hat{G}$ is obtained by

$$\mu_t = \nu_t C.$$

When defining a lifted random walk, the lifted walk can be affected by many different constraints. A first parameter to be specified is the initial distribution ν_0. We can either take any probability distribution on $\hat{V}$ (s) or a probability distribution obtained from the initial distribution μ_0 on V (S). In the latter case, $\mu_0(u)$ is the probability attributed to each vertex $u \in V$ and this probability needs to be lifted onto the set of vertices $c^{-1}(u)$ in a way that for all μ_0, the linear map F that maps μ_0 to ν_0 satisfies

$$\mu_0 F C = \mu_0,$$

which means that $F_{ui} = 0$ if $c(i) \neq u$. So the relevant initial distributions in this case are of the form $\nu_0 = \mu_0 F$.

Furthermore, our goal is still to make μ_t converge as fast as possible to π, which means that for some t, $\mu_t = \nu_t C = \pi$. However, the convergence of the marginal distribution μ_t does not necessarily imply that $\nu_{t+1}C = \pi$. So we can either impose that $\nu_t C = \pi$ for all $t > T$ such that $\nu_T C = \pi$ (i) or accept that for some $t > T$, we have $\nu_t C \neq \pi$ (I).

Another constraint concerns the reducibility. For the random walk on G described by transition matrix P to converge to a unique stationary distribution, it has to be irreducible. The lifted Markov chain described by A can be defined to be either reducible (R) or irreducible (r).

A further constraint on the ergodic flows of the lifted Markov chain is presented in [5]. The ergodic flow of an irreducible aperiodic Markov chain P is defined as $Q^{(P)}_{uv} = p_{uv}\pi_v$. The ergodic flow for the lifted chain A with stationary distribution $\hat{\pi}$ is similarly given by $\hat{Q}^{(A,\hat{\pi})}_{ij} = a_{ij}\hat{\pi}_j$. To compare the flows for A and P, first we define

$$\hat{Q}^{(A,\hat{\pi})}_{c^{-1}(u),c^{-1}(v)} = \sum_{i,j:c(i)=u,c(j)=v} \hat{Q}^{(A,\hat{\pi})}_{ij}.$$

Then we construct the Markov chain $\tilde{P}^{(A,\hat{\pi})}$, which satisfies $\tilde{P}^{(A,\hat{\pi})}_{uv}\pi_v = \hat{Q}^{(A,\hat{\pi})}_{c^{-1}(u),c^{-1}(v)}$. We call $\tilde{P}^{(A,\hat{\pi})}$ the induced Markov chain on G by A and $\hat{\pi}$. Now, we can define a lifted Markov chain without any restriction on the ergodic flows (E) or with ergodic flows matching those of P, i.e. $\tilde{P}^{(A,\hat{\pi})} = P$ (e).

Example 2.15 (Diaconis lift [8] on the 5-cycle)**.** Let G be the 5-cycle with $V = \{1, 2, 3, 4, 5\}$ and consider the simple random walk on G. Then a walker on vertex $v \in V$ moves to one of its neighbours $v + 1 \mod 5$ or $v - 1 \mod 5$ with probability $\frac{1}{2}$, as represented in figure 2.2. The transition matrix for the simple random walk is given by

$$\begin{pmatrix} 0 & \frac{1}{2} & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \end{pmatrix}$$

and the stationary distribution is $\pi = \left(\frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}, \frac{1}{5}\right)$.

First, let us construct the lifted graph $\hat{G}$. For every vertex n in G, we construct two vertices v^+ and v^- in the lifted graph $\hat{G}$, adding a sense of direction to every node.

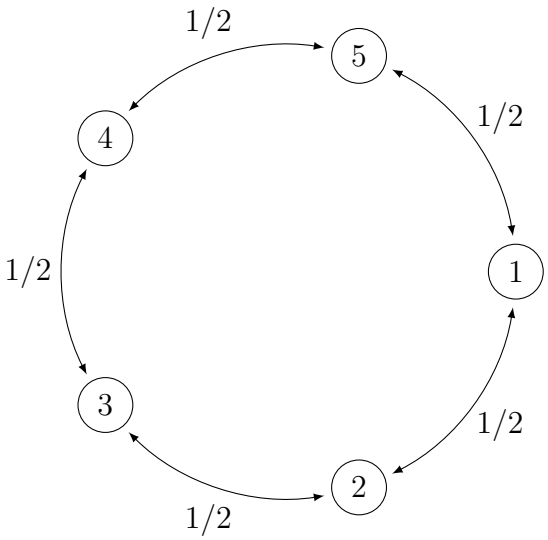


Figure 2.2: Simple random walk on 5-cycle

The projection of $\hat{G}$ onto G is given by $c(v^+) = c(v^-) = v$ for all $v \in V$. The set of allowed edges according to definition 2.14 is given by $\hat{E} = \{(v^s, (v \pm 1 \mod 5)^s) : v \in V, s \in \{+, -\}\}$.

Now, we would like to define a random walk on $\hat{G}$, where the walker's next step depends on the previous one. Intuitively, the walker will continue in the same direction as the previous step with high probability and switch direction with low probability. To put transition probabilities on the edges, we proceed as follows. Let P_+ be the permutation matrix for a clockwise rotation on G and P_- the counter-clockwise rotation on G. If the walker is for example on vertex v^+, the "+" indicates that the previous position was vertex $(v - 1 \mod 5)^+$, i.e. the walker moves in clockwise direction. At the next step, the walker will move to vertex $(v + 1 \mod 5)^+$, continuing in clockwise direction, with probability $1 - \frac{1}{|V|} = \frac{4}{5}$ or to vertex $(v + 1 \mod 5)^-$, switching from the clockwise to the counter-clockwise direction, with probability $\frac{1}{|V|} = \frac{1}{5}$. The adjacency matrix of this walk is given by the block matrix

$$A = \begin{pmatrix} \frac{4}{5}P_+ & \frac{1}{5}P_+ \\ \frac{1}{5}P_- & \frac{4}{5}P_- \end{pmatrix}.$$

This lifted random walk on the 5-cycle is represented in figure 2.3. On the outer circle, the walker only moves clockwise, on the inner circle only counter-clockwise and there is a small probability to switch from one circle to the other. Note that we only use a subset of the allowed edges $\hat{E}$.

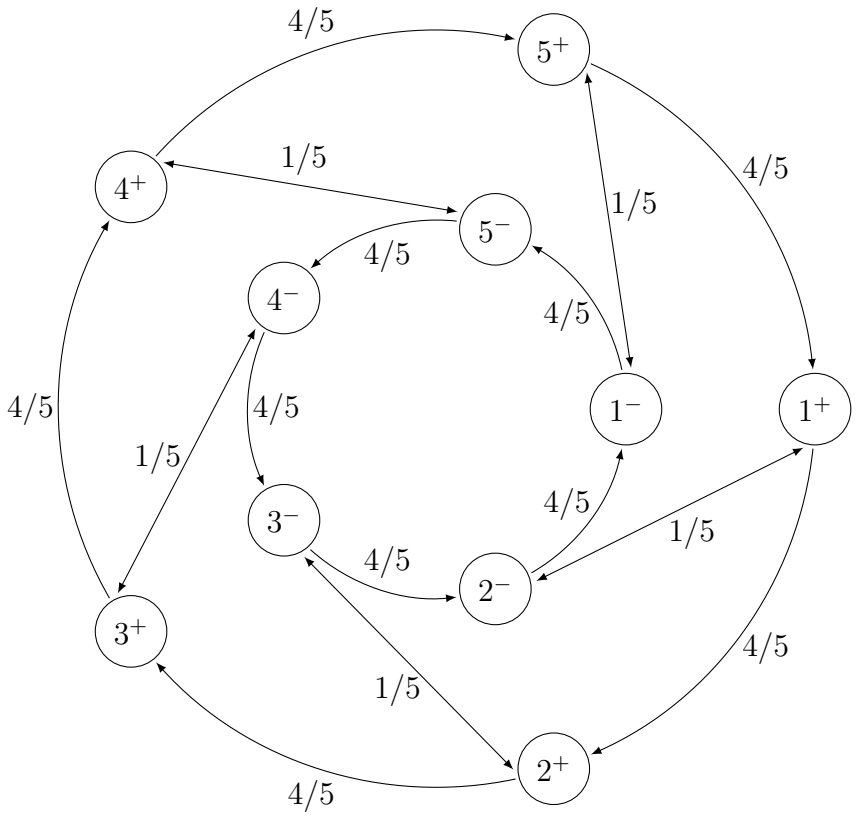


Figure 2.3: Diaconis lift on the 5-cycle

The Diaconis lift on the 5-cycle satisfies the set of of constraints (sIre). Indeed, the lifted graph $\hat{G}$ is connected and the lifted Markov chain is irreducible. The stationary distribution of the lifted walk is the uniform distribution $\hat{\pi}_i = \frac{1}{|\hat{G}|} = \frac{1}{10}$ for all $i \in \hat{G}$. A simple calculation allows to show that the induced chain $\tilde{P}^{(A,\hat{\pi})}$ on G coincides with P, thus the lifted Markov chain has matching ergodic flows. Moreover, the Diaconis lift does not satisfy the invariance of the target distribution constraint. Take for example the following initial distribution on the vertices of $\hat{G}$,

$$\nu_0 = \begin{pmatrix} 1^+ & 2^+ & 3^+ & 4^+ & 5^+ & 1^- & 2^- & 3^- & 4^- & 5^- \\ 0 & \frac{1}{10} & \frac{1}{5} & \frac{1}{10} & \frac{1}{10} & \frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} & \frac{1}{10} \end{pmatrix}.$$

Then the marginal distribution on the vertices of G is given by

$$\nu_0 C = \begin{pmatrix} \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} & \frac{1}{5} \end{pmatrix} = \pi.$$

However, when applying one step of the lifted walk, we obtain

$$\nu_1 = \nu_0 A = \begin{pmatrix} \frac{1}{10} & 0 & \frac{1}{10} & \frac{9}{50} & \frac{3}{25} & \frac{1}{10} & 0 & \frac{1}{10} & \frac{3}{25} & \frac{9}{50} \end{pmatrix},$$

and

$$\nu_1 C = \begin{pmatrix} \frac{1}{5} & 0 & \frac{1}{5} & \frac{3}{10} & \frac{3}{10} \end{pmatrix} \neq \pi.$$

Therefore, the Diaconis lift on the 5-cycle takes place in a (sIre) scenario.

2.2.1 Mixing time

There are to ways to measure the mixing time of a lifted random walk. One way is to calculate the mixing time of the probability distribution ν_t to its stationary distribution $\hat{\pi}$ on the set of lifted vertices $\hat{V}$, i.e.

$$M_\varepsilon = \min\{T | \forall t \geq T : \sup_{\nu_0} ||\nu_0 A^t - \hat{\pi}||_{TV} \leq \varepsilon\}.$$

Another possibility is to calculate the marginal mixing time of $\nu_t C$ to the stationary distribution π on V, since our objective is to accelerate the mixing time of the random walk on the underlying graph G.

Definition 2.16 (Marginal mixing time)**.** *Let π be the stationary distribution of a random walk on G and A the transition matrix of a lifted random walk on $\hat{G}$. Let $\varepsilon > 0$. The marginal mixing time is defined as*

$$M_{marg,\varepsilon} = \min\{T | \forall t \geq T : \max_{\nu_0} ||\nu_0 A^t C - \pi||_{TV} \leq \varepsilon\}.$$

Different combinations of constraints on the lifted Markov chain give rise to different bounds on the mixing time. The study and proof of these bounds is the purpose of the paper [5] by Apers, Ticozzi and Sarlette. The proofs make use of basic building blocks to construct lifted random walks satisfying the different constraints. However, those constructions are more existential than practical from an algorithmic point of view, as they might rely on extensive optimisation processes. We will content ourselves with stating the results, the interested reader will find the details in [5].

For the combination of constraints (SI), it is possible to determine an upper bound on the mixing time that depends on the diameter of the underlying graph G. The **graph diameter** $D(G)$ is the maximum distance between two vertices in G.

Theorem 2.17. [5, Theorem 2] *For the set of constraints (SI), except (SIre), there exists a lifted Markov chain such that*

$$M_{\frac{1}{4}} \leq M_{marg,\frac{1}{4}} \leq D(G) + 1.$$

This is the best possible reduction of mixing time we can achieve for lifted random walks, since no random walk on a graph can mix in a time shorter than the graph diameter. In the opposite case of (si) however, it was shown that this set of constraints is too restrictive and does not allow to reduce the mixing time.

Theorem 2.18. [5, Theorem 1] *For a lifted Markov chain A on $\hat{G}$ with constraints (si), there exists a non-lifted Markov chain P that behaves like A.*

Now, let us consider the cases (Si) and (sI). In both of these cases it is possible to establish a lower bound on the mixing time in terms of conductance.

Theorem 2.19. [5, Theorem 4 and 5] *For scenarios including (Si), we have the lower conductance bound*

$$M_{\frac{1}{4}} \geq M_{marg,\frac{1}{4}} \geq \frac{1}{8\Phi}.$$

Note that we have used the notation Φ instead of $\Phi(P)$. Φ denotes the graph conductance and is obtained by taking the maximum over all admissible transition matrices P with stationary distribution π. Finally,

Theorem 2.20. [5, Theorem 3] *For a lifted Markov chain satisfying the constraint (s), we have*

$$M_{\frac{1}{4}} \geq M_{marg,\frac{1}{4}} \geq \frac{1}{4\Phi},$$

and if we add the constraint on matching ergodic flows, (se),

$$M_{\frac{1}{4}} \geq M_{marg,\frac{1}{4}} \geq \frac{1}{4\Phi(P)}.$$

To illustrate how lifted walks can reduce the mixing time compared to non-lifted walks, let us once again take the example of a cycle.

Example 2.21. In the non-lifted case, the mixing time on the N-cycle is of order N^2, as shown in [10], and the simple random walk is the best possible walk on the cycle. Now, since the Diaconis lift satisfies the constraints (sIre) and the conductance $\Phi(P)$ of the underlying simple random walk is of order $\frac{1}{N}$, by theorem 2.20, the marginal mixing time satisfies a lower bound of order N. (It is possible to build a lift on the cycle satisfying the constraints (Si) and therefore satisfying the upper bound $D(G)+1$. With the diameter of the N-cycle being $\frac{N}{2}$, this shows the mixing time of a lifted walk is of order N [5].)

Chapter 3

Discrete-time quantum walks

Quantum walks are the quantum counterpart of classical random walks. The key idea for quantum walks is to use quantum states instead of classical states. Since quantum states change through unitary transformations, this allows us to use a wider range of transformations, using matrices with negative or complex entries, as opposed to the stochastic matrices used for classical random walks. The hope is to achieve better properties than classical random walks, for example a speedup in mixing times.

3.1 First approach : quantum walk on the line

We will start by taking a look at quantum walks on the integer line, before defining quantum walks on general graphs. This section is based on [3] and [16].

A classical random walk on the line starts in 0 and the walker moves left (-1) with probability $\frac{1}{2}$ or right $(+1)$ with probability $\frac{1}{2}$, such that the set of possible positions is $\mathbb{Z}$. Now, we want to adapt this process to the quantum case.

The first idea is to take the basis states $|n\rangle$ with $n \in \mathbb{Z}$ and apply a unitary transformation

$$|n\rangle \to U|n\rangle = a|n-1\rangle + b|n+1\rangle,$$

where $a, b \in \mathbb{C}$ do not depend on n and a represents the amplitude of moving one step to the left, i.e. the probability of moving left is given by $|a|^2$, and b denotes the amplitude of moving to the right. Unfortunately, U is only unitary for the trivial transformations, where the walker always moves left or always moves right.

We can solve this problem by adding a coin space to the position space of the walker. The coin space is introduced to decide if the walker moves left or right on the line, like defining a classical random walk by tossing a coin and deciding to move left if it show tail and move right if it shows head. The position space $\mathcal{H}_P$ is the Hilbert space spanned by the position's basis states $\{|n\rangle \mid n \in \mathbb{Z}\}$. The coin space $\mathcal{H}_C$ is spanned by the two basis states $\{|\ell\rangle, |r\rangle\}$, since at each position the walker can go left or right. We define the states of the total system in the Hilbert space

$$\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C,$$

with basis states

$$|n, \ell\rangle := |n\rangle \otimes |\ell\rangle \text{ and } |n, r\rangle := |n\rangle \otimes |r\rangle.$$

When applying a step of the quantum walk, we first perform a coin flip transformation C, which acts on the coin space and leaves the position state unchanged :

$$\begin{aligned}(I \otimes C)|n, \ell\rangle &= a|n, \ell\rangle + b|n, r\rangle \\ (I \otimes C)|n, r\rangle &= c|n, \ell\rangle + d|n, r\rangle.\end{aligned}$$

A frequently used coin transformation is the Hadamard coin

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} = \frac{1}{\sqrt{2}}(|\ell\rangle\langle\ell| + |\ell\rangle\langle r| + |r\rangle\langle\ell| - |r\rangle\langle r|).$$

A different coin operator will result in a quantum walk with a different behaviour.

After the coin flip, we execute a shift operator S according to the result of the coin flip. The shift operator can be defined as

$$S = \sum_{n\in\mathbb{Z}} |n-1\rangle\langle n| \otimes |\ell\rangle\langle\ell| + \sum_{n\in\mathbb{Z}} |n+1\rangle\langle n| \otimes |r\rangle\langle r|$$

such that when the coin state is $|\ell\rangle$, the walker moves one step to the left and when the coin state is $|r\rangle$, the walker moves one step to the right,

$$\begin{aligned}S|n, \ell\rangle &= |n-1, \ell\rangle \\ S|n, r\rangle &= |n+1, r\rangle.\end{aligned}$$

A single step of the discrete quantum walk is thus equivalent to applying the unitary

transformation

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

to the initial quantum state. U is indeed unitary by 1.14 and 1.15, because S, C and I are unitary.

Definition 3.1 (Discrete quantum walk on the line)**.** *A discrete quantum walk of t steps on the integer line is defined as the unitary transformation U^t, where U acts on a state in the Hilbert space $\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$ and is given by*

$$U = S \cdot (C \otimes I).$$

The total state of the system after t steps starting in state $|\psi_0\rangle \in \mathcal{H}$ is

$$|\psi_t\rangle = U^t|\psi_0\rangle.$$

Example 3.2. Let $|\psi_0\rangle = |0, \ell\rangle$ be the initial state of the quantum system on the integer line, i.e. we start in position $|0\rangle$ pointing to the left. If we apply one step of the quantum walk with the Hadamard coin H and the shift operator defined as above, we obtain the state

$$\begin{aligned}
|\psi_1\rangle &= U|\psi_0\rangle \\
&= S \cdot (I \otimes H)(|0, \ell\rangle) \\
&= S\left(\frac{1}{\sqrt{2}}|0, \ell\rangle + \frac{1}{\sqrt{2}}|0, r\rangle\right) \\
&= \frac{1}{\sqrt{2}}(S|0, \ell\rangle + S|0, r\rangle) \\
&= \frac{1}{\sqrt{2}}(|-1, \ell\rangle + |1, r\rangle).
\end{aligned}$$

When measuring this state, we observe that, starting from initial position $|0\rangle$, the probability to be at position $|1\rangle$ is $\frac{1}{2}$ and the probability to be at position $|-1\rangle$ is $\frac{1}{2}$. So measuring the quantum walk with the Hadamard coin H after one step is equivalent to one step of the simple classical random walk on the integer line.

We can let the quantum walk evolve for several steps, which yields

$$\begin{aligned}
|\psi_2\rangle &= U|\psi_1\rangle \\
&= \frac{1}{\sqrt{2}} U(|-1,\ell\rangle + |1,r\rangle) \\
&= \frac{1}{\sqrt{2}} \cdot S\left(\frac{1}{\sqrt{2}}|-1,\ell\rangle + \frac{1}{\sqrt{2}}|-1,r\rangle + \frac{1}{\sqrt{2}}|1,\ell\rangle - \frac{1}{\sqrt{2}}|1,r\rangle\right) \\
&= \frac{1}{2}(|-2,\ell\rangle + |0,r\rangle + |0,\ell\rangle - |2,r\rangle).
\end{aligned}$$

A measurement after the second time step returns a probability of $\frac{1}{4}$ to be in position state $|-2\rangle$ or $|2\rangle$ and $\frac{1}{2}$ to be in state $|0\rangle$.

$$\begin{aligned}
|\psi_3\rangle &= U^3|\psi_0\rangle \\
&= \frac{1}{2\sqrt{2}}\left(|-3,\ell\rangle + |-1,r\rangle + |-1,\ell\rangle - |1,r\rangle + |-1,\ell\rangle + |1,r\rangle - |1,\ell\rangle + |3,r\rangle\right) \\
&= \frac{1}{2\sqrt{2}}\left(|-3,\ell\rangle + |-1,r\rangle + 2|-1,\ell\rangle - |1,\ell\rangle + |3,r\rangle\right).
\end{aligned}$$

After the third step of the quantum walk, we start to see some differences between the classical and the quantum case. In contrast to the classical random walk, we observe that due to quantum interference some states cancel out and other states add up. After measurement, we obtain the position states $|-3\rangle$, $|3\rangle$ and $|1\rangle$ with probability $\frac{1}{8}$ and $|-1\rangle$ with probability $\frac{5}{8}$, which is not the same probability distribution as for the simple classical walk on the line after 3 steps. The probability distribution of the simple classical walk on the line after a great number of steps is symmetric, peaked around the origin and dropping of exponentially on both sides. The quantum walk with the Hadamard coin and initial state $|0,\ell\rangle$ yields a distribution that is peaked on the left.

If we start the walk on state $|0,r\rangle$, however, the probability distribution after a large number of steps will be peaked on the right. A symmetric starting state, $\frac{1}{\sqrt{2}}|0,\ell\rangle + \frac{1}{\sqrt{2}}|0,r\rangle$ will lead to a symmetric distribution that is approximately uniform around the origin and peaked on the far left and right. We can also obtain a symmetric distribution on the positions by taking a coin that treats both initial states $|0,\ell\rangle$ and $|0,r\rangle$ the same way, for example

$$C = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & \imath \\ \imath & 1 \end{pmatrix}.$$

So the choice of the initial state and the coin operator have an impact on the probability

distribution on the positions we obtain after t steps of the walk.

Another difference between the classical and the quantum walk appears when considering the expected distance from the origin. In the case of a simple classical walk on the line, after t steps the expected distance of the walker from 0 is of order $\sqrt{t}$. On the other hand, the expected distance from origin of a quantum walk is of order t, which implies a quantum walk on the line propagates quadratically faster than its classical counterpart [4].

3.2 Quantum walks on general graphs

The quantum walk on the line is an interesting first approach and allows us to get a first idea of how a discrete-time quantum walk evolves and differs from a classical random walk. However, applications-wise it would be more interesting to be able to describe quantum walks on more general graphs in higher dimensions.

Discrete-time quantum walks can be described by several different models. In this work, we will only present two of these models. First, the shunt-decomposition model, which is usually applied to regular graphs and also uses a coin space, like the quantum walk on the line we described in the previous section. The second model is the arc-reversal model. This model does not use a separate position and coin space. The description of these two models follows [11] and [12].

3.2.1 Shunt-decomposition model

Let us now consider a d-regular undirected graph $G = (V, E)$ in which each vertex is of degree d. As for the quantum walk on the line, the possible positions of the walker are the positions it can take in the graph, which are the different vertices. Thus the position space $\mathcal{H}_P$ is spanned by $\{|u\rangle \mid u \in V\}$. We would also like to specify a direction at each vertex. But instead of only being able to go "left" or "right" as on the line, at each vertex, there are d possible directions and the basis of the coin space $\mathcal{H}_C$ is given by $\{|1\rangle, \ldots, |d\rangle\}$. For every vertex $u \in V$, we label all the edges adjacent to u with $j \in \{1, \ldots, d\}$. Then the total quantum space for the walker is $\mathcal{H} = \mathcal{H}_P \otimes \mathcal{H}_C$ and the state associated to a position of the walker on vertex u and pointing along the edge labeled j on u's end is given by $|u, j\rangle := |u\rangle \otimes |j\rangle$.

In a similar way to the quantum walk on the line, we want to define a conditional shift operator S according to some labeling of the edges. For every $u \in V$, let

$$f_u : \{1, \dots d\} \to \{v : u \sim v\}$$

be a linear order describing the labels on the edges. If the directed arc (u, v) is labeled j, which means the edge $\{u, v\}$ is labelled j on u's end and $f_u(j) = v$, then the walker moves along the edge labeled j to vertex v,

$$S|u, j\rangle = |f_u(j), j\rangle = |v, j\rangle.$$

By definition, S is a binary matrix and each column contains exactly one 1. Moreover, S has to be unitary, which implies that the set of linear orders $\{f_u : u \in V\}$ can not be arbitrary. In fact, each label j needs to induce a permutation P_j on the set of vertices V. These permutations are called shunts.

Definition 3.3 (Shunt decomposition)**.** *A shunt on an undirected graph G is a permutation of its vertices, that maps each vertex of the graph to one of its neighbours. A shunt decomposition is a collection of shunts partitioning the arcs of G.*

Since G is d-regular, a shunt decomposition consists of d shunts and because a shunt decomposition partitions the arcs in G, the permutation matrices for each label add up to the adjacency matrix $A(G)$ of G,

$$A(G) = P_1 + \cdots + P_d.$$

We then define

$$S = \begin{pmatrix} P_1 & & & \\ & P_2 & & \\ & & \ddots & \\ & & & P_d \end{pmatrix}.$$

Next, we need to choose a unitary coin transformation C. Since the coin space $\mathcal{H}_C$ is d-dimensional, the coin transformation for a d-regular graph has to be a d-dimensional unitary transformation. The choice of coin transformations is very large and depends on the specific properties we wish to achieve with the quantum walk. If we want a balanced coin for example, where every direction occurs with the same probability $\frac{1}{d}$, we can use

the Fourier coin

$$F = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{d-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \dots & \omega^{(d-1)(d-1)} \end{pmatrix}$$

where $\omega = e^{\frac{2i\pi}{d}}$ is the d-th root of unity. The Fourier-coin is a generalisation of the Hadamard coin to higher-dimensional coin spaces. Another frequently used coin transformation is Grover's operator

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

The unitary transition matrix U defining a shunt-decomposition walk on a d-regular graph G is given by

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

where I_n is the $n \times n$ identity matrix.

Example 3.4. Let us perform a shunt-decomposition walk on the undirected graph $G = (V, E)$ represented in figure 3.1.

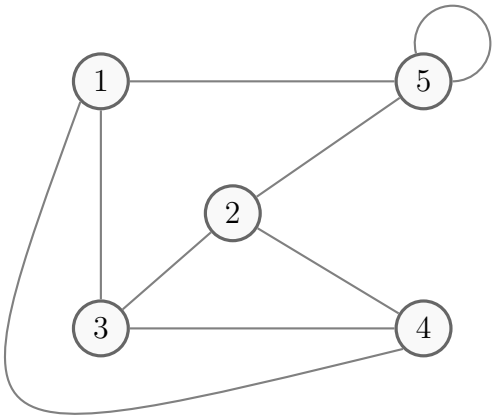


Figure 3.1: 3-regular graph G

We define the following linear order on the neighbours of every node $u \in V$:

$$\begin{aligned}
&f_1(1) = 5, && f_1(2) = 4, && f_1(3) = 3\\
&f_2(1) = 4, && f_2(2) = 3, && f_2(3) = 5\\
&f_3(1) = 2, && f_3(2) = 1, && f_3(3) = 4\\
&f_4(1) = 3, && f_4(2) = 2, && f_4(3) = 1\\
&f_5(1) = 1, && f_5(2) = 5, && f_5(3) = 2.
\end{aligned}$$

This set of linear orders induces the permutations

$$P_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, P_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0 & 0\\ 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}, P_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 \end{pmatrix},$$

and we can verify that $A(G) = P_1 + P_2 + P_3$, so the set of linear orders is valid. Thus, the shift operator for this shunt-decomposition is given by

$$S = \begin{pmatrix} P_1 & 0 & 0\\ 0 & P_2 & 0\\ 0 & 0 & P_3 \end{pmatrix}.$$

Next, we need to choose a coin, for example the Grover coin

$$G = \begin{pmatrix} \frac{-1}{3} & \frac{2}{3} & \frac{2}{3}\\ \frac{2}{3} & \frac{-1}{3} & \frac{2}{3}\\ \frac{2}{3} & \frac{2}{3} & \frac{-1}{3} \end{pmatrix}.$$

The unitary transition matrix for the Grover walk on G with the linear orders specified above is

$$U = S \cdot (I_5 \otimes G).$$

With initial state $|\psi_0\rangle = \frac{1}{\sqrt{3}}(|1,1\rangle + |1,2\rangle + |1,3\rangle)$, the state after 1 and 2 time steps

is given by

$$\begin{aligned}
|\psi_1\rangle &= U|\psi_0\rangle \\
&= \frac{1}{\sqrt{3}} S \left(\frac{-1}{3}|1,1\rangle + \frac{2}{3}|1,2\rangle + \frac{2}{3}|1,3\rangle + \frac{2}{3}|1,1\rangle + \frac{-1}{3}|1,2\rangle + \frac{2}{3}|1,3\rangle \right. \\
&\qquad \left. + \frac{2}{3}|1,1\rangle + \frac{2}{3}|1,2\rangle + \frac{-1}{3}|1,3\rangle \right) \\
&= \frac{1}{\sqrt{3}} S \left(|1,1\rangle + |1,2\rangle + |1,3\rangle \right) \\
&= \frac{1}{\sqrt{3}} S \left(|5,1\rangle + |4,2\rangle + |3,3\rangle \right), \\
|\psi_2\rangle &= \frac{1}{3\sqrt{3}} \left(-|1,1\rangle + 2|1,2\rangle + 2|1,3\rangle + 2|2,1\rangle - |2,2\rangle + 2|2,3\rangle \right. \\
&\qquad \left. + 2|3,1\rangle - |4,3\rangle + 2|5,2\rangle \right).
\end{aligned}$$

So after two steps, we will find the walker on vertices 1 and 2 with probability $\frac{1}{3}$, on vertices 3 and 5 with probability $\frac{4}{27}$ and on vertex 4 with probability $\frac{1}{27}$.

Remark 3.5. The set of d linear orders partitioning the arcs of G can be interpreted as a d-colouring of the edges in the bipartite graph $K_2 \times G$. The graph $K_2 \times G$ contains the edge $\{u, v'\}$ if $\{u, v\}$ is an edge in G. The edges of one colour form a perfect matching of $K_2 \times G$.

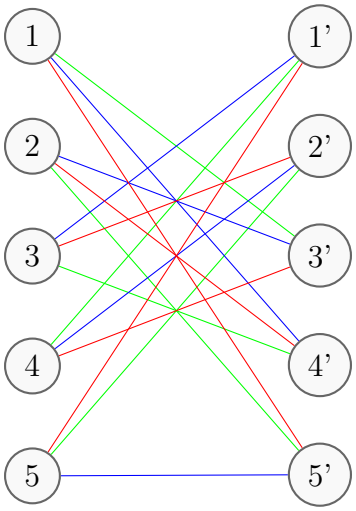


Figure 3.2: edge-colouring of the bipartite graph $K_2 \times G$

On the graph G of example 3.4, we can give each of the three labels a colour and colour the edges in $K_2 \times G$ according to these labels,

- $\{u, v'\}$ is red if $f_u(1) = v$
- $\{u, v'\}$ is blue if $f_u(2) = v$
- $\{u, v'\}$ is green if $f_u(3) = v$,

as it is shown in figure 3.2. So there is a bijection between the decomposition of the adjacency matrix into permutation matrices

$$A(G) = P_1 + P_2 + P_3$$

according to the linear orders and a d-colouring of $K_2 \times G$. This is true for every d-regular graph G.

This remark allows us to prove the existence of a shunt-decomposition for every d-regular graph.

Proposition 3.6. [12] *Every d-regular graph G admits a shunt-decomposition.*

Proof. We can construct the bipartite d-regular graph $K_2 \times G$. The proof follows from the fact that every d-regular bipartite graph admits an edge-colouring with d colours [15]. In remark 3.5, we constructed a bijection between a d-colouring of the edges in $K_2 \times G$ and a shunt-decomposition. □

3.2.2 Arc-reversal model

Let $G = (V, E)$ be an arbitrary undirected graph. We will view G as a directed graph by replacing each edge $\{u, v\}$ by a pair of arcs in opposite directions, (u, v) and (v, u). As the name of this model implies, in the arc-reversal model we perform a walk on the arcs of the graph rather than the vertices. Suppose G has m arcs. Then the space of the system is a m-dimensional Hilbert space with basis states $|(u, v)\rangle$ associated with the arcs (u, v).

Remark 3.7. A state $|u, j\rangle$ where the walker is at vertex u pointing along the edge labelled j is equivalent to $|(u, v)\rangle$, where v is the endpoint of the edge labelled j.

To define an arc-reversal walk on a graph G, we need to define two unitary matrices. First, a permutation matrix

$$R = \sum_{(u,v)\in E} |(v,u)\rangle\langle(u,v)|$$

reversing the arcs such that

$$R|(u,v)\rangle = |(v,u)\rangle.$$

Since R is a binary symmetric matrix, it is clear that R is unitary. The second matrix is the block diagonal coin matrix

$$C = \begin{pmatrix} C_1 & & & \\ & C_2 & & \\ & & \ddots & \\ & & & C_n \end{pmatrix}$$

For every vertex $u \in V$, C_u is a unitary matrix of order $\deg(u)$ that sends the j^{th} arc of u to a superposition of all outgoing arcs of u. To be able to talk about the j^{th} arc of a vertex, as in the shunt-decomposition model, for each vertex u we need to define a linear order on the set of neighbours of u,

$$f_u : \{1, 2, \ldots, \deg(u)\} \to \{v : u \sim v\}.$$

This way, $f_u(j)$ describes the j^{th} neighbour of u and $(u, f_u(j))$ the j^{th} arc. The linear order is used to index the rows and columns of R and C by the arcs in the order

$$(1, f_1(1)), \ldots, (1, f_1(d_1)), (2, f_2(1)), \ldots, (2, f_2(d_2)), \ldots, (n, f_n(1)), \ldots, (n, f_n(d_n)),$$

with $d_u := \deg(u)$.

Note that both R and C are matrices of dimension $m \times m$. Finally, the unitary transition matrix U of the arc-reversal quantum walk is given by

$$U = R \cdot C,$$

so that at every step of the walk, we first perform a coin flip and then an arc reversal.

Example 3.8. Consider the undirected graph $G = (V, E)$ in figure 3.3.

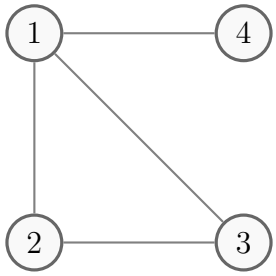


Figure 3.3: undirected graph G

First, we replace each edge by two arcs going in opposite directions. Then we choose an arbitrary linear order on the neighbours of every node $u \in V$,

$$\begin{aligned} &f_1(1) = 2, && f_1(2) = 3, && f_1(3) = 4, \\ &f_2(1) = 1, && f_2(2) = 3, \\ &f_3(1) = 1, && f_3(2) = 2, \\ &f_4(1) = 1. \end{aligned}$$

Indexed according to the linear order, the arc-reversal matrix for G is given by

$$R = \begin{array}{c} \begin{array}{cccccccc} (1,2) & (1,3) & (1,4) & (2,1) & (2,3) & (3,1) & (3,2) & (4,1) \end{array} \\ \left(\begin{array}{cccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{array} \right) \end{array} \begin{array}{c} (1,2) \\ (1,3) \\ (1,4) \\ (2,1) \\ (2,3) \\ (3,1) \\ (3,2) \\ (4,1) \end{array}$$

Now, we provide every vertex $u \in V$ with the balanced Fourier coin. Since $\deg(1) = 3$, C_1 is of dimension 3×3, $\deg(2) = \deg(3) = 2$, so C_2 and C_3 are 2×2 matrices and $\deg(4) = 1$, thus C_4 is a 1×1 matrix. This yields the following block diagonal coin

matrix

$$C = \begin{array}{c} \begin{matrix} (1,2) & (1,3) & (1,4) & (2,1) & (2,3) & (3,1) & (3,2) & (4,1) \end{matrix} \\ \begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{3} & e^{\frac{2\pi i}{3}}/\sqrt{3} & e^{\frac{4\pi i}{3}}/\sqrt{3} & 0 & 0 & 0 & 0 & 0 \\ 1/\sqrt{3} & e^{\frac{4\pi i}{3}}/\sqrt{3} & e^{\frac{8\pi i}{3}}/\sqrt{3} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \begin{matrix} (1,2) \\ (1,3) \\ (1,4) \\ (2,1) \\ (2,3) \\ (3,1) \\ (3,2) \\ (4,1) \end{matrix}$$

Suppose the initial state of the system is $|\psi_0\rangle = |(3,1)\rangle$. Then after one iteration of the arc-reversal walk, the system will be in the state

$$\begin{aligned} |\psi_1\rangle &= RC|(3,1)\rangle \\ &= R\left(\frac{1}{\sqrt{2}}|(3,1)\rangle + \frac{1}{\sqrt{2}}|(3,2)\rangle\right) \\ &= \frac{1}{\sqrt{2}}|(1,3)\rangle + \frac{1}{\sqrt{2}}|(2,3)\rangle. \end{aligned}$$

A second iteration yields the state

$$|\psi_2\rangle = \frac{1}{\sqrt{6}}|(2,1)\rangle + \frac{e^{\frac{2\pi i}{3}}}{\sqrt{6}}|(3,1)\rangle + \frac{e^{\frac{4\pi i}{3}}}{\sqrt{6}}|(4,1)\rangle + \frac{1}{2}|(1,2)\rangle - \frac{1}{2}|(3,2)\rangle.$$

After two steps of this walk, the probability of the walker being at vertex 1 is $\frac{1}{4}$, the probability of being at vertex 2 or 4 is $\frac{1}{6}$ and the probability of being at vertex 3 is $\frac{5}{12}$.

Remark 3.9. For the arc-reversal model with the Fourier coin, the linear orders f_u influence the walk, but all linear orders yield valid states. The walk with the Grover coin is indifferent to the linear order.

Chapter 4

Limiting distribution and mixing time

As for the classical random walks, in the quantum case we are interested in the limiting distribution and mixing time of the quantum walk. Since quantum walks use unitary transformations, the limiting distribution is not as straight-forward to obtain as in the classical case, which is why we introduce average states and probabilities in section 4.2. To obtain a definition of limiting distribution that supports both the arc-reversal model on the arcs of a graph and the shunt-decomposition model describing a quantum walk on the vertices, in this chapter we will work with arc probabilities. They are the more suitable choice, as they can be converted to a probability distribution on the vertices. Aharonov et al. in [2] studied the mixing time on the vertices. Godsil and Zhan extended this work to arc probabilities in [11], [12] and [20]. This chapter is based on the latter references.

4.1 Probabilities in terms of density matrices

Let $G = (V, E)$ be an undirected graph with n vertices and m arcs, where we have replaced every edge $\{u, v\} \in E$ by the set of arcs (u, v) and (v, u). Take a discrete quantum walk on G evolving according to the unitary matrix U. If the quantum walk starts at initial state $|\psi_0\rangle$, then after t steps, the state of the system is given by

$$|\psi_t\rangle = U^t|\psi_0\rangle.$$

After measurement, we obtain that the walker is found on arc (u, v) after t steps when starting in state $|\psi_0\rangle$ with probability

$$\Pr_t((u,v)|\psi_0) = |\langle(u,v)|\psi_t\rangle|^2.$$

Now, the density matrix for the pure basis state $|(u,v)\rangle$ associated with arc (u,v) is

$$\rho_{(u,v)} = |(u,v)\rangle\langle(u,v)|.$$

For initial pure state $\rho_0 = |\psi_0\rangle\langle\psi_0|$, the state of the system after t steps can be described by the density matrix

$$\rho_t = U^t\rho_0(U^t)^* = U^t|\psi_0\rangle\langle\psi_0|(U^t)^* = |\psi_t\rangle\langle\psi_t|.$$

Since $\rho_t^* = \rho_t$ and $\{|(u,v)\rangle\}$ forms an orthonormal basis, by definition of the trace (see A.11),

$$\begin{aligned}
\langle\rho_t|\rho_{(u,v)}\rangle &= \mathrm{Tr}(\rho_t\rho_{(u,v)}) \\
&= \sum_{(u',v')} \langle(u',v')|\rho_t\rho_{(u,v)}|(u',v')\rangle \\
&= \sum_{(u',v')} \langle(u',v')|\psi_t\rangle\langle\psi_t|(u,v)\rangle\langle(u,v)|(u',v')\rangle \\
&= \langle(u,v)|\psi_t\rangle\langle\psi_t|(u,v)\rangle \\
&= |\langle(u,v)|\psi_t\rangle|^2.
\end{aligned}$$

The equation above allows us to express the probability of finding the walker on arc (u, v) starting from state $|\psi_0\rangle$ in terms of density matrices,

$$\Pr_t((u,v)|\psi_0) = \langle\rho_t|\rho_{(u,v)}\rangle.$$

We could also be interested in the probability of finding the walker on a set of arcs. For any subset S of arcs, we define the uniform mixed state density matrix over S as

$$\rho_S = \frac{1}{|S|}\sum_{(u,v)\in S}\rho_{(u,v)},$$

and the probability of the walker being on any of the arcs in the set S is given by

$$\Pr_t(S|\psi_0) = \sum_{(u,v)\in S} \Pr_t((u,v)|\psi_0).$$

This allows us to determine the probability of finding the walker on a specific vertex u instead of an arc by taking the sum of $Pr_t((u,v)|\rho_0)$ over all the outgoing arcs of u,

$$\Pr_t(u|\psi_0) = \textstyle\sum_{v:u\ v} \Pr_t((u,v)|\psi_0).$$

We can rewrite

$$\Pr_t(S|\psi_0) = \sum_{(u,v)\in S} \langle\rho_t|\rho_{(u,v)}\rangle = |S|\left\langle \rho_t \middle| \tfrac{1}{|S|} \sum_{(u,v)\in S} \rho_{(u,v)} \right\rangle = |S|\langle\rho_t|\rho_S\rangle, \tag{4.1}$$

which is a more convenient notation.

4.2 Average limiting distributions

As in the case of classical random walks, we are interested in the limiting behaviour of the walk. However, the quantum walk evolves trough a unitary transformation U, so neither the probabilities on the arcs defined by $\Pr_t(S|\psi_0)$ nor ρ_t do converge, since for all t,

$$||U\psi_0 - \psi_0|| = ||U^{t+1}\psi_0 - U^t\psi_0||.$$

To avoid this problem, we define the average state and the average probability.

Definition 4.1 (Average state after T steps)**.** *The average state of a quantum walk after T steps is defined as*

$$\overline{\rho_T} = \frac{1}{T}\sum_{t=0}^{T-1} \rho_t.$$

Definition 4.2 (Average probability after T steps)**.** *The average probability of a quantum walk after T steps is given by*

$$\overline{\Pr_T}((u,v)|\rho_0) = \frac{1}{T}\sum_{t=0}^{T-1} \Pr_t((u,v)|\rho_0).$$

The average state and average probability converge and we can express their limits using the spectral decomposition of U as detailed in A.9, with $\lambda_j = e^{i\Theta_j}$,

$$U = \sum_j e^{i\Theta_j} P_j, \tag{4.2}$$

and $P_j^* = P_j = P_j^2$.

Theorem 4.3. [12] *Let U be the unitary transition matrix of the discrete quantum walk on G with spectral decomposition (4.2). Then*

$$\lim_{T\to\infty} \overline{\rho_T} = \sum_j P_j \rho_0 P_j.$$

Proof. First, let us show that we can write

$$\overline{\rho_T} = \sum_j P_j \rho_0 P_j + \frac{1}{T} \sum_{j\neq k} \left(\frac{1 - e^{iT(\Theta_j - \Theta_k)}}{1 - e^{i(\Theta_j - \Theta_k)}} \right) P_j \rho_0 P_k. \tag{4.3}$$

The spectral decomposition of U is given by

$$U = \sum_j e^{i\Theta_j} P_j,$$

where P_j is Hermitian, thus $P_j = P_j^*$. Moreover, the complex conjugate of $e^{i\Theta_j}$ is $e^{-i\Theta_j}$ and $U^t = \sum_j e^{it\Theta_j} P_j$. It follows that

$$\begin{aligned}
\rho_t &= (U^t)\rho_0 (U^t)^* \\
&= \left(\sum_j e^{it\Theta_j} P_j \right) \rho_0 \left(\sum_k e^{-it\Theta_k} P_k \right)^* \\
&= \sum_{j,k} e^{it\Theta_j} e^{-it\Theta_k} P_j \rho_0 P_k^* \\
&= \sum_{j,k} e^{it(\Theta_j - \Theta_k)} P_j \rho_0 P_k \\
&= \sum_j P_j \rho_0 P_j + \sum_{j\neq k} e^{it(\Theta_j - \Theta_k)} P_j \rho_0 P_k.
\end{aligned}$$

Therefore,

$$\overline{\rho_T} = \frac{1}{T} \sum_{t=0}^{T-1} \rho_t$$

$$= \sum_j P_j \rho_0 P_j + \frac{1}{T} \sum_{j \neq k} \left(\sum_{t=0}^{T-1} e^{it(\Theta_j - \Theta_k)} \right) P_j \rho_0 P_k$$
$$= \sum_j P_j \rho_0 P_j + \frac{1}{T} \sum_{j \neq k} \frac{1 - e^{iT(\Theta_j - \Theta_k)}}{1 - e^{i(\Theta_j - \Theta_k)}} P_j \rho_0 P_k.$$

To conclude the proof of the theorem,

$$\left\| \overline{\rho_T} - \sum_r P_j \rho_0 P_j \right\| = \left\| \frac{1}{T} \sum_{j \neq k} \frac{1 - e^{iT(\Theta_j - \Theta_k)}}{1 - e^{i(\Theta_j - \Theta_k)}} P_j \rho_0 P_k \right\|$$
$$\leq \frac{1}{T} \sum_{j \neq k} \left| \frac{1 - e^{iT(\Theta_j - \Theta_k)}}{1 - e^{i(\Theta_j - \Theta_k)}} \right| ||P_j \rho_0 P_k||$$
$$\leq \frac{1}{T} \sum_{j \neq k} \frac{2}{\left|1 - e^{i(\Theta_j - \Theta_k)}\right|} ||P_j \rho_0 P_k||$$

Since the sum on the right hand side of the inequality does not depend on T, we deduce that

$$\lim_{T \to \infty} \overline{\rho_T} = \sum_j P_j \rho_0 P_j.$$

□

Theorem 4.4. [12] *Let U be the unitary transition matrix of the discrete quantum walk on G with spectral decomposition (4.2). Then*

$$\lim_{T \to \infty} \overline{\Pr_T}(S|\psi_0) = |S| \sum_j \langle P_j \rho_0 P_j | \rho_S \rangle.$$

Proof. The proof follows from theorem 4.2 and equation (4.1),

$$\lim_{T \to \infty} \overline{\Pr_T}(S|\psi_0) = \lim_{T \to \infty} \frac{1}{T} \sum_{t=0}^{T-1} |S| \langle \rho_t | \rho_S \rangle$$
$$= |S| \left\langle \lim_{T \to \infty} \sum_{t=0}^{T-1} \rho_t \middle| \rho_S \right\rangle$$
$$= |S| \left\langle \sum_j P_j \rho_0 P_j \middle| \rho_S \right\rangle$$
$$= |S| \sum_j \langle P_j \rho_0 P_j | \rho_S \rangle.$$

□

4.3 Mixing time

The mixing time in the quantum case measures how fast the average probabilities converge to their limiting distribution.

Definition 4.5 (Mixing time)**.** *For $\epsilon > 0$, the mixing time $M_\epsilon(\psi_0, S)$ of a quantum walk starting in state $|\psi_0\rangle$ is the smallest integer L such that for all $T > L$,*

$$\left|\overline{\Pr_T}(S|\psi_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| \leq \epsilon.$$

Remark 4.6. In the classical case, it was possible to define the mixing time of a Markov chain independently of the initial distribution since the stationary distribution does not depend on the initial distribution. This is not the case for quantum walks, as the limiting distribution depends on ρ_0, therefore the mixing time has to depend on the initial state of the walk.

Theorem 4.7. [12] *Let U be the unitary transition matrix of the discrete quantum walk on G with spectral decomposition (4.2). Then*

$$M_\varepsilon(\psi_0, S) \leq \frac{2|S|}{\varepsilon}\sum_{j\neq k}\frac{|\langle P_j\rho_0 P_k|\rho_S\rangle|}{|e^{i\Theta_j} - e^{i\Theta_k}|}. \tag{4.4}$$

Proof. Using equations (4.1) and (4.3), we have

$$\begin{aligned}
\overline{\Pr_T}(S|\psi_0) &= \frac{1}{T}\sum_{t=0}^{T-1}|S|\langle\rho_t|\rho_S\rangle \\
&= |S|\langle\overline{\rho_T}|\rho_S\rangle \\
&= |S|\left\langle\sum_j P_j\rho_0 P_j + \frac{1}{T}\sum_{j\neq k}\left(\frac{1-e^{iT(\Theta_j-\Theta_k)}}{1-e^{i(\Theta_j-\Theta_k)}}\right)P_j\rho_0 P_k\middle|\rho_S\right\rangle \\
&= |S|\sum_j\langle P_j\rho_0 P_j|\rho_0\rangle + \frac{|S|}{T}\sum_{j\neq k}\left(\frac{1-e^{iT(\Theta_j-\Theta_k)}}{1-e^{i(\Theta_j-\Theta_k)}}\right)\langle P_j\rho_0 P_k|\rho_S\rangle.
\end{aligned}$$

This implies

$$\left|\overline{\Pr_T}(S|\psi_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| = \frac{|S|}{T}\left|\sum_{j\neq k}\left(\frac{1-e^{iT(\Theta_j-\Theta_k)}}{1-e^{i(\Theta_j-\Theta_k)}}\right)\langle P_j\rho_0 P_k|\rho_S\rangle\right|$$
$$\leq \frac{|S|}{T}\sum_{j\neq k}\left|\frac{1-e^{iT(\Theta_j-\Theta_k)}}{1-e^{i(\Theta_j-\Theta_k)}}\right||\langle P_j\rho_0 P_k|\rho_S\rangle|$$

by the triangle inequality. Since

$$\left|\frac{1-e^{iT(\Theta_j-\Theta_k)}}{1-e^{i(\Theta_j-\Theta_k)}}\right| = \left|\frac{e^{i\Theta_k}(1-e^{iT(\Theta_j-\Theta_k)})}{e^{i\Theta_k}-e^{i\Theta_j}}\right| \leq \frac{|e^{i\Theta_k}||1-e^{iT(\Theta_j-\Theta_k)}|}{|e^{i\Theta_k}-e^{i\Theta_j}|} \leq \frac{2}{|e^{i\Theta_j}-e^{i\Theta_k}|},$$

we obtain

$$\left|\overline{\Pr_T}(S|\psi_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| \leq \frac{2|S|}{T}\sum_{j\neq s}\frac{|\langle P_j\rho_0 P_k|\rho_S\rangle|}{|e^{i\Theta_j}-e^{i\Theta_k}|}. \tag{4.5}$$

Now, set

$$K = \frac{2|S|}{\varepsilon}\sum_{j\neq k}\frac{|\langle P_j\rho_0 P_k|\rho_S\rangle|}{|e^{i\Theta_j}-e^{i\Theta_k}|}.$$

Then for all $T > K$,

$$\left|\overline{\Pr_T}(S|\rho_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| \leq \varepsilon$$

and $M_\varepsilon(\psi_0, S) \leq K$. □

We can obtain a series of weaker bounds, which require less knowledge of the quantum walk transition matrix U.

Corollary 4.8. [12] *Let ℓ be the number of pairs of distinct eigenvalues and $\Delta = \min\{|e^{i\Theta_j} - e^{i\Theta_k}| : j \neq k\}$. Then*

$$M_\varepsilon(\psi_0, S) \leq \frac{2}{\varepsilon}\sum_{j\neq k}\sum_{a\in S}\frac{\sqrt{(P_j)_{aa}(P_k)_{aa}}}{|e^{i\Theta_j}-e^{i\Theta_k}|} \tag{4.6}$$
$$\leq \frac{2|S|}{\varepsilon}\sum_{j\neq k}\frac{1}{|e^{i\Theta_j}-e^{i\Theta_k}|} \tag{4.7}$$
$$\leq \frac{2\ell|S|}{\varepsilon\Delta}. \tag{4.8}$$

Proof. By inequality (4.5),

$$\begin{aligned}\left|\overline{\Pr_T}(S|\psi_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| &\leq \frac{2|S|}{T}\sum_{j\neq k}\frac{1}{|e^{i\Theta_j}-e^{i\Theta_k}|}|\langle P_j\rho_0 P_k|\rho_S\rangle| \\ &= \frac{2|S|}{T}\sum_{j\neq k}\frac{1}{|e^{i\Theta_j}-e^{i\Theta_k}|}|\langle \rho_0|P_j\rho_S P_k\rangle|\end{aligned}$$

since P_j and P_k are Hermitian. Moreover, the Cauchy-Schwarz inequality implies that

$$|\langle \rho_0|P_j\rho_S P_k\rangle| \leq ||\rho_0||||P_j\rho_S P_k|| \leq ||P_j\rho_S P_k||,$$

as $||\rho_0|| \leq 1$ by proposition 1.18(iii). Thus,

$$\left|\overline{\Pr_T}(S|\psi_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| \leq \frac{2|S|}{T}\sum_{j\neq k}\frac{||P_j\rho_S P_k||}{|e^{i\Theta_j}-e^{i\Theta_k}|}. \tag{4.9}$$

Taking $K_1 = \frac{2|S|}{\varepsilon}\sum_{j\neq k}\frac{||P_j\rho_S P_k||}{|e^{i\Theta_j}-e^{i\Theta_k}|}$ yields

$$\left|\overline{\Pr_T}(S|\rho_0) - |S|\sum_j \langle P_j\rho_0 P_j|\rho_S\rangle\right| \leq \varepsilon$$

for all $T > K_1$ and $M_\varepsilon(\rho_0, S) \leq K_1$.

To find the bound (4.7), recall that $\rho_S = \frac{1}{|S|}\sum_{a\in S}|a\rangle\langle a|$ and that P_j is idempotent. Hence,

$$\begin{aligned}||P_j\rho_S P_k|| &\leq \frac{1}{|S|}\sum_{a\in S}||P_j\rho_a P_k|| \\ &= \frac{1}{|S|}\sum_{a\in S}\sqrt{\mathrm{Tr}(P_k^*\rho_a P_j^* P_j\rho_a P_k)} \\ &= \frac{1}{|S|}\sum_{a\in S}\sqrt{\mathrm{Tr}(\langle a|P_j^2|a\rangle\langle a|P_k^2|a\rangle)} \\ &= \frac{1}{|S|}\sum_{a\in S}\sqrt{\mathrm{Tr}(\langle a|P_j|a\rangle\langle a|P_k|a\rangle)} \\ &= \frac{1}{|S|}\sum_{a\in S}\sqrt{(P_j)_{a,a}(P_k)_{a,a}},\end{aligned}$$

where we have used the cyclic property A.12 of the trace. We deduce that

$$\left|\overline{\Pr_T}(S|\psi_0) - |S|\sum_r \langle P_j\rho_0 P_j|\rho_S\rangle\right| \leq \frac{2}{T}\sum_{r\neq s}\sum_{a\in S}\frac{\sqrt{(P_j)_{a,a}(P_k)_{a,a}}}{|e^{i\Theta_r}-e^{i\Theta_s}|} \tag{4.10}$$

and obtain (4.7) by taking $K_2 = \frac{2}{\varepsilon}\sum_{r\neq s}\sum_{a\in S}\frac{\sqrt{(P_j)_{a,a}(P_k)_{a,a}}}{|e^{i\Theta_r}-e^{i\Theta_s}|}$.

For the last bound, (4.8), we have that $\sum_j P_j = \mathrm{Id}$, therefore $0 \leq (P_j)_{a,a} \leq 1$ for all j and

$$\sum_{a\in S}\sqrt{(P_j)_{a,a}(P_k)_{a,a}} \leq |S|.$$

If $\Delta = \min\{|e^{i\Theta_j} - e^{i\Theta_k}| : j \neq k\}$, then

$$\begin{aligned}\left|\overline{\Pr_T}(S|\rho_0) - |S|\sum_r \langle P_j\rho_0 P_j|\rho_S\rangle\right| &\leq \frac{2|S|}{T}\sum_{j\neq k}\frac{1}{\Delta}\\ &\leq \frac{2\ell|S|}{T\Delta},\end{aligned}$$

with ℓ the number of pairs of distinct eigenvalues. Taking $K_3 = \frac{2\ell|S|}{\varepsilon\Delta}$ gives us bound (4.8). □

Example 4.9. Aharanov et al. [2] showed that the limiting distribution of the shunt-decomposition walk with Fourier coin on an N-cycle with odd N is uniform, independently of the initial state $|\psi_0\rangle$. Through bounds similar to those above, they also proved

$$M_\varepsilon \leq \mathcal{O}\left(\frac{L\log N}{\varepsilon^3}\right),$$

compared to the order of N^2 for the classical simple random walk.

Chapter 5

Comparison between classical random walks with memory and quantum random walks on graphs with symmetry

The aim of this chapter is to define a lifted random walk on the Cayley graph of any finite abelian group with symmetric generating set and to compare the mixing time of this lifted walk to the mixing time of both the shunt-decomposition and the arc-reversal model, using the Grover coin or the Fourier coin.

5.1 Cayley graphs

Cayley graphs provide a suitable framework to compare lifted random walks and quantum walks, especially the shunt-decomposition walk, since the generators used to construct the Cayley graph of a given group provide a natural way of defining a lifted random walk, as we will see in section 5.2, as well as the different permutations on the vertices necessary for the shunt-decomposition model.

Definition 5.1 (Cayley graph)**.** [6] *Let G be a finite group and $S \subset G$ a subset of G that does not contain the identity element. The Cayley graph $\Gamma(G, S)$ is the directed graph with vertex set the elements $g \in G$ and edges (g, sg) if $s \in S$.*

For $\Gamma(G, S)$ to be connected, S has to be a generating set of G. Moreover, if the set S is symmetric, i.e. $s \in S$ implies $s^{-1} \in S$, then every edge in Γ is bidirectional.

In this chapter, we will work with abelian groups, such that both (g, sg) and (g, gs) form the same edge, and symmetric generating sets S. The Cayley graphs we build are therefore undirected d-regular graphs of degree $d = |S|$. We will consider the binary operation on G to be additive.

The following structure theorem allows us to write any finite abelian group as Cartesian product of cyclic groups, which will be very useful for the implementation of the classical and quantum random walks.

Theorem 5.2 (Fundamental structure theorem for finite abelian groups)**.** [14, Theorem 13.4] *Let G be a finite abelian group. Then*

$$G \cong \mathbb{Z}_{p_1^{m_1}} \times \mathbb{Z}_{p_2^{m_2}} \times \cdots \times \mathbb{Z}_{p_n^{m_n}},$$

where $p_1, p_2, \ldots, p_n$ (not necessarily distinct) are prime numbers and $m_1, m_2, \ldots, m_n \in \mathbb{Z}$.

Example 5.3 (Abelian group with 15 elements)**.** Due to the fundamental structure theorem for finite abelian groups, the abelian group with 15 elements is isomorphic to $\mathbb{Z}_5 \times \mathbb{Z}_3$. The Cayley graph of $\mathbb{Z}_5 \times \mathbb{Z}_3$ with symmetric generating set $S = \{(1,0), (4,0), (0,1), (0,2)\}$ is represented in figure 5.1. Note that we have taken a generator of $\mathbb{Z}_5$ and its inverse in the first entry and did the same for $\mathbb{Z}_3$ in the second entry, such that $\Gamma(\mathbb{Z}_5 \times \mathbb{Z}_3, S)$ can also be seen as the Cartesian product of the two Cayley graphs $\Gamma(\mathbb{Z}_5, \{1,4\})$ and $\Gamma(\mathbb{Z}_3, \{1,2\})$.

To construct the adjacency matrix of $\Gamma(\mathbb{Z}_5 \times \mathbb{Z}_3, S)$, we can first construct the adjacency matrices of $\Gamma_1(\mathbb{Z}_5, \{1,4\})$ and $\Gamma_2(\mathbb{Z}_3, \{1,2\})$, which are the 5-cycle and the 3-cycle,

$$A(\Gamma_1) = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{pmatrix}, \; A(\Gamma_2) = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}.$$

The adjacency matrix of Γ is then given by

$$A(\Gamma) = A(\Gamma_1) \otimes \mathrm{Id}_3 + \mathrm{Id}_5 \otimes A(\Gamma_2).$$

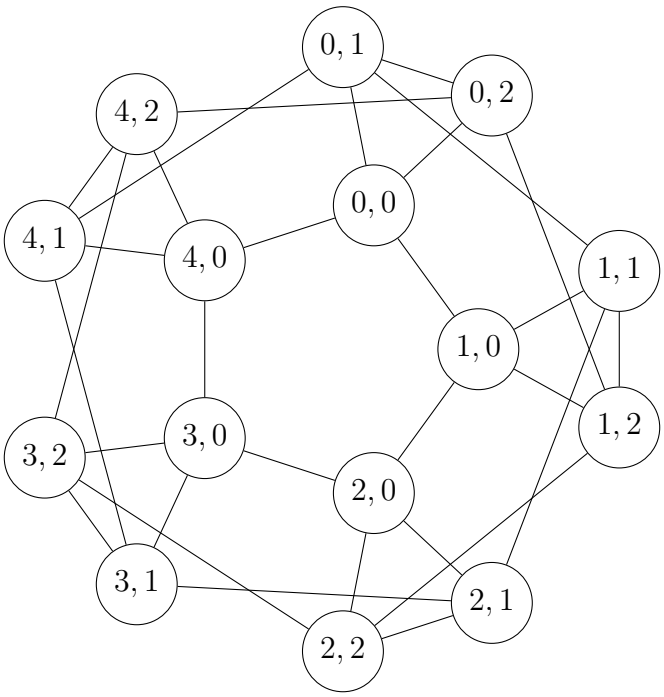


Figure 5.1: Cayley graph $\Gamma(\mathbb{Z}_5 \times \mathbb{Z}_3, S)$

Remark 5.4. In the above example, and in general, we are not restricted to choose the generators in each factor of the Cartesian product separately. The set $S = \{(1,1),(4,2)\}$ with "mixed" generators $(1,1)$ and $(4,2)$ is also a valid symmetric generating set for a Cayley graph of $\mathbb{Z}_5 \times \mathbb{Z}_3$.

5.2 Generalised Diaconis lift on Cayley graphs

In this section, we want do define a lifted random walk on the Cayley graph of a finite abelian group with given generating set S. To do so, let us reconsider the Diaconis lift on the cycle, as defined in example 2.15. As a reminder, to construct the Diaconis lift on the N-cycle, for each vertex v in the original graph, we have created two vertices v^+ and v^- in the lifted graph, to add information as to which vertex the walker came from, since there are only two possible directions on a cycle. We can see this as a graph with two layers which are N-cycles : the first layer where the walker can only move in clockwise direction and the second layer, where the walker only moves counter-clockwise. At each step, there is a probability of $1 - \frac{1}{N}$ that the walker continues to move in the same direction as previously and a small probability of $\frac{1}{N}$ that the walker switches from one layer to the other.

The N-cycle can also be seen as the Cayley graph of the abelian group $\mathbb{Z}_N$ with generating set $S = \{-1, +1\}$. The element $s = +1$ induces a permutation on the vertices which is equivalent to the layer with the clockwise direction and similarly $s = -1$ induces the permutation used in the second layer going counter-clockwise. We want to generalise this procedure to more general Cayley graphs.

Let $\Gamma(G, S)$ be the Cayley graph of a finite abelian group G with symmetric generating set S. The graph Γ has $n = |G|$ vertices and every vertex has degree $d = |S|$. We want to construct the lifted graph $\hat{\Gamma}$. Each $s \in S$ defines a direction for the walker and induces a permutation P_s on the vertices of Γ, which defines one layer of the lifted graph. As in the Diaconis lift, the walker continues along the same direction with probability $1 - \frac{1}{n}$. Now, instead of having two layers in the lifted graph and only other choice when switching to a different layer, as the Diaconis lift, $\hat{\Gamma}$ will have $d = |S|$ different layers and $d - 1$ choices when switching to another layer. The walker chooses one of the other layers uniformly at random, which implies that the walker goes from its current layer to any other layer with probability $\frac{1}{(d-1)n}$.

If $s_1, \ldots, s_d$ are the generators in S, then the transition matrix A for the generalised Diaconis lift on $\Gamma(G, S)$ is given by

$$A = \sum_{i,j=1}^{d} Q_{ij} e_i e_j^T \otimes P_{s_i},$$

where e_i is the d-dimensional unit vector with 1 on its i^{th} entry and

$$Q_{ij} = \begin{cases} 1 - \frac{1}{n} & \text{if } i = j \\ \frac{1}{(d-1)n} & \text{if } i \neq j. \end{cases}$$

To clarify this construction, let us build the transition matrix A for the generalised Diaconis lift on a small example.

Example 5.5. Consider the Cayley graph $\Gamma(\mathbb{Z}_5, S)$, with $S = \{1, 4, 2, 3\}$ as in figure 5.2.

Since $n = 5$ and $d = 4$, the lifted graph $\hat{\Gamma}$ will have 20 vertices. Every vertex v from the original graph is splitted into the four vertices $v^{(1)}, v^{(4)}, v^{(2)}, v^{(3)}$, one for every

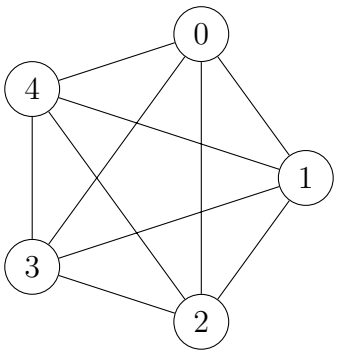


Figure 5.2: Cayley graph $\Gamma(\mathbb{Z}_5, \{1, 4, 2, 3\})$

possible direction. Now, we need to find the permutations P_s for each $s \in S$,

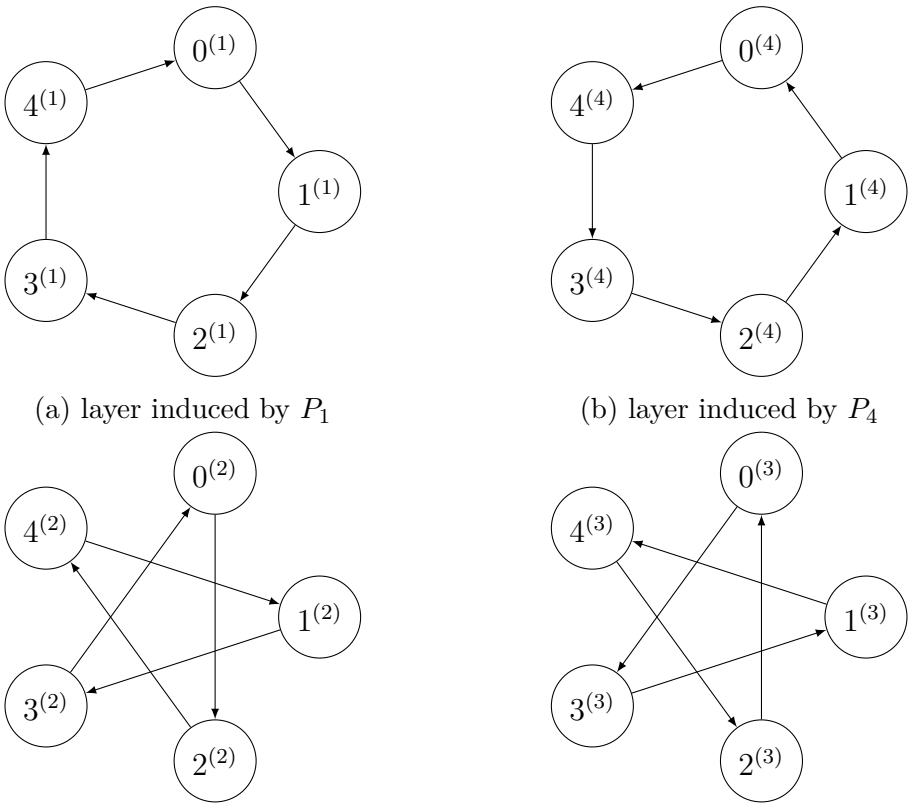
$$P_1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \qquad P_4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix},$$

$$P_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}, \qquad P_3 = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

These four permutation matrices induce the four layers, each with one direction, of the lifted graph represented in the figure 5.3 beneath.

To see how the different layers are connected, suppose the walker is at vertex $3^{(4)}$ in the lifted graph. Then it will continue in the direction given by P_4, moving to $2^{(4)}$, with probability $\frac{4}{5}$ or switch to a different layer, moving to $2^{(1)}, 2^{(2)}$ or $2^{(3)}$, with probability $\frac{1}{15}$. The transition matrix A of the lifted walk on $\hat{\Gamma}$ is the $(nd) \times (nd)$ block matrix

$$A = \begin{pmatrix} \frac{4}{5}P_1 & \frac{1}{15}P_1 & \frac{1}{15}P_1 & \frac{1}{15}P_1 \\ \frac{1}{15}P_4 & \frac{4}{5}P_4 & \frac{1}{15}P_4 & \frac{1}{15}P_4 \\ \frac{1}{15}P_2 & \frac{1}{15}P_2 & \frac{4}{5}P_2 & \frac{1}{15}P_2 \\ \frac{1}{15}P_3 & \frac{1}{15}P_3 & \frac{1}{15}P_3 & \frac{4}{5}P_3 \end{pmatrix}.$$



(a) layer induced by P_1

(b) layer induced by P_4

(c) layer induced by P_2

(d) layer induced by P_3

Figure 5.3

Remark 5.6. Note that the four permutation matrices P_1, P_4, P_2, P_3 add up to the adjacency matrix of Γ. So they also form a valid shunt-decomposition for the shunt-decomposition quantum walk.

5.3 Comparison

This last section is dedicated to an empiric comparison between the simple classical random walk as defined in section 2.1 (SW), the generalised Diaconis lifted random walk (LW), the shunt-decomposition quantum walk with Fourier coin or Grover coin (SD_F and SD_G respectively) and the arc-reversal quantum walk with Fourier or Grover coin (AR_F and AR_G respectively). As a reminder, the mixing time for the simple random walk is given by 2.10, for the generalised Diaconis lift we consider the marginal mixing time 2.16 and the quantum mixing time is defined as 4.5 and our objective is to minimise the mixing time.

n	G	d	SW	LW	SD_F	SD_G	AR_F	AR_G
31	$\mathbb{Z}_{31}$	2	183	43	24	2	6	24
32	-	-	-	-	-	-	-	-
33	$\mathbb{Z}_{11} \times \mathbb{Z}_3$	2+2	12	7	14	10	8	5
34	-	-	-	-	-	-	-	-
35	$\mathbb{Z}_7 \times \mathbb{Z}_5$	2+2	9	10	11	4	4	4
36	$\mathbb{Z}_9 \times \mathbb{Z}_4$	2+2	31	11	13	6	4	4
	$\mathbb{Z}_4 \times \mathbb{Z}_3 \times \mathbb{Z}_3$	2+2+2	4	8	7	4	8	2
37	$\mathbb{Z}_{37}$	2	260	51	29	2	6	28
38	-	-	-	-	-	-	-	-
39	$\mathbb{Z}_{13} \times \mathbb{Z}_3$	2+2	16	8	17	8	10	6
40	$\mathbb{Z}_8 \times \mathbb{Z}_5$	2+2	11	11	13	6	6	5
41	$\mathbb{Z}_{41}$	2	319	57	32	2	6	31
42	-	-	-	-	-	-	-	-
43	$\mathbb{Z}_{43}$	2	351	60	34	2	6	33
44	$\mathbb{Z}_{11} \times \mathbb{Z}_4$	2+2	46	13	16	8	4	6
45	$\mathbb{Z}_9 \times \mathbb{Z}_5$	2+2	11	10	14	7	6	4
	$\mathbb{Z}_5 \times \mathbb{Z}_3 \times \mathbb{Z}_3$	2+2+2	5	9	8	5	9	3
46	-	-	-	-	-	-	-	-
47	$\mathbb{Z}_{47}$	2	419	66	37	2	6	36
48	$\mathbb{Z}_{16} \times \mathbb{Z}_3$	2+2	25	10	22	10	12	8
	$\mathbb{Z}_4 \times \mathbb{Z}_4 \times \mathbb{Z}_3$	2+2+2	6	10	10	4	12	4
49	$\mathbb{Z}_{49}$	2	456	68	39	2	6	37
	$\mathbb{Z}_7 \times \mathbb{Z}_7$	2+2	13	17	17	5	6	5
50	-	-	-	-	-	-	-	-
51	$\mathbb{Z}_{17} \times \mathbb{Z}_3$	2+2	28	11	24	11	14	8

Table 5.1: Mixing times ($\varepsilon = \frac{1}{4}$) on Cayley graphs of finite abelian groups with $n = 31$ to $n = 51$ elements and generating set $\{+1, -1\}$ on each factor.

To get a first idea of the behaviour of the different walks, we can compute the mixing time for graphs of $n = 31$ to $n = 51$ vertices. To do so, we apply the structure theorem to obtain all groups with $n = 31$ to $n = 51$ elements and consider all the factorisations that do not include a factor $\mathbb{Z}_2$, such that in each factor, the two elements $+1$ and -1 are distinct. Then we construct a Cayley graph of each group as Cartesian product of the Cayley graphs of the factors with generating set $S = \{+1, -1\}$, as in example 5.3. We perform each walk taking as initial state the unit vector e_1 with 1 on its first entry of the appropriate dimension and compute the mixing time to its limiting distribution for $\varepsilon = \frac{1}{4}$. The results are represented in table 5.1. The lowest mixing time for each Cayley graph is marked orange.

First of all, we observe that there is not one model that performs better than all the others in all cases. But the way we interpret and analyse this table depends on our final objective. If all we need is a random walk that converges as fast as possible to any distribution, then the quantum walks SD_G, AR_F and AR_G show the lowest mixing times. The shunt-decomposition walk with Grover coin for example has mixing time 2 on all cycles. The relevance of this result however is debatable. Since the degree of every vertex in a cycle is $d = 2$, the Grover coin in this case is

$$G = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

and will not result in a superposition of states. Moreover, as the initial distribution of the walk in our case is concentrated on a single state, the walker only moves between one vertex and one of its neighbours, without exploring the rest of the graph.

From a practical point of view also, the three models SD_G, AR_F and AR_G are more difficult to analyse. Contrary to SW, LW and SD_F, which converge to the uniform distribution independently of the initial state, SD_G, AR_F and AR_G converge to different limiting distributions for every initial state and the limiting distribution needs to be computed using the expression in theorem 4.4 based on the eigenprojections. This means that the computation time increases rapidly for larger graphs.

As mentioned above, both the simple random walk and the generalised Diaconis lift converge to the uniform stationary distribution. Of the quantum walks, only the shunt-decomposition walk with Fourier coin has uniform limiting distribution for every Cayley graph we considered and independently of the initial state. The arc-reversal walk converges to the uniform distribution only for the cycles. So it seems appropriate to make a more detailed comparison between the simple classical walk, the generalised Diaconis lift and the shunt-decomposition quantum walk with Fourier coin. Let us mark the best mixing time to the uniform distribution in blue in table 5.1. A first glance tells us that the simple classical walk is outperformed by the lifted walk in many cases. More precisely, it is outperformed by either the lifted walk or the shunt-decomposition walk on 13 graphs and outperformed by both on 10 graphs. It is still not possible to determine one random walk model that produces better mixing times to the uniform distribution than all others. The shunt-decomposition walk for example seems to do very well on cycles and even more so as the size of the cycle increases. However, on the groups $\mathbb{Z}_7 \times \mathbb{Z}_5$, $\mathbb{Z}_4 \times \mathbb{Z}_3 \times \mathbb{Z}_3$, $\mathbb{Z}_5 \times \mathbb{Z}_3 \times \mathbb{Z}_3$, $\mathbb{Z}_4 \times \mathbb{Z}_4 \times \mathbb{Z}_3$ and $\mathbb{Z}_7 \times \mathbb{Z}_7$, the simple random walk yields the lowest mixing time. This might be because all of these groups are the

d	SW	LW	SD_F	d	SW	LW	SD_F
2+2	122	48	71	4+2	43	56	51
2+4	120	40	59	4+4	12	42	27
2+6	160	35	79	4+6	10	37	22
2+8	200	33	86	4+8	12	34	25
2+10	240	31	106	4+10	17	31	33
2+12	279	30	114	4+12	20	30	40
6+2	55	78	70	8+2	69	100	70
6+4	14	50	25	8+4	17	61	31
6+6	7	12	17	8+6	8	48	16
6+8	6	38	15	8+8	5	42	11
6+10	7	36	16	8+10	5	39	11
6+12	8	33	17	8+12	5	37	11
10+2	83	122	102	12+2	96	144	107
10+4	20	72	34	12+4	23	83	39
10+6	9	56	18	12+6	10	63	20
10+8	5	48	11	12+8	5	53	11
10+10	4	43	10	12+10	4	47	10
10+12	5	40	10	12+12	4	43	9

Table 5.2: Mixing times on the Cayley graphs of group $\mathbb{Z}_{29} \times \mathbb{Z}_{17}$ depending on the number of generators in $\mathbb{Z}_{29}$ and $\mathbb{Z}_{17}$.

Cartesian product of cyclic groups of small order which are very well connected even if the degree is only 4 (resp. 6 if there are 3 factors).

So far, we have only tested the different walks on Cayley graphs with a relatively small number of vertices and the two generators +1 and -1 for each factor given by the fundamental structure theorem for finite abelian groups. During the simulations, it became clear that the actual generators chosen for each factor do not affect the mixing time. This means that the mixing time for SW, LW and SD_F on the Cayley graphs $\Gamma(\mathbb{Z}_{11}, \{1, -1, 2, -2\})$ and $\Gamma(\mathbb{Z}_{11}, \{3, -3, 4, -4\})$ for example is respectively the same. The number of generators chosen for each factor greatly affects the mixing time as well as which model performs best. The tables 5.2 and 5.3 contain the mixing times of those three models according to the number of generators, on graphs with $n = 493$ and $n = 765$ vertices . For $\mathbb{Z}_{17} \times \mathbb{Z}_9 \times \mathbb{Z}_5$ for example, the entry $d = 2 + 6 + 4$ means that we take 2 generators of $\mathbb{Z}_{17}$, 6 generators of $\mathbb{Z}_9$ and 4 generators in $\mathbb{Z}_5$ such that the total degree of the Cayley graph is 12. The lowest mixing time in each case is marked orange.

d	SW	LW	SD_F	d	SW	LW	SD_F	d	SW	LW	SD_F
2+2+2	42	20	39	2+4+2	55	22	47	2+6+2	69	27	51
2+2+4	55	24	54	2+4+4	69	21	52	2+6+4	83	22	60
4+2+2	16	27	38	4+4+2	9	26	21	4+6+2	10	30	27
4+2+4	20	27	42	4+4+4	8	21	20	4+6+4	9	21	18
6+2+2	20	34	36	6+4+2	9	32	27	6+6+2	9	35	25
6+2+4	24	33	55	6+4+4	7	25	18	6+6+4	6	24	14

Table 5.3: Mixing times on the Cayley graphs of the group $\mathbb{Z}_{17} \times \mathbb{Z}_9 \times \mathbb{Z}_5$ depending on the number of generators in $\mathbb{Z}_{17}$, $\mathbb{Z}_9$ and $\mathbb{Z}_5$.

We can observe in both tables that when the first factor only has 2 generators, then both the LW and the SD_F perform better then the SW, with the LW showing the lowest mixing times. As the number of generators in the first factor increases, the mixing time of the simple random walk decreases drastically and the simple random walk outruns the other two models, with the lifted random walk performing even worse than the shunt-decomposition model in most cases.

When considering a first factor of larger order however and a second factor of small order, we obtain the results in table 5.4. We can see that either the generalised Diaconis lift or the shunt-decomposition walk generate lower mixing times as we slightly increase the number of generators in the factor of higher order. But for a larger number of generators, we fall back into the same pattern as previously, with the simple random walk outrunning both other models.

G	d	SW	LW	SD_F	G	d	SW	LW	SD_F
$\mathbb{Z}_{53} \times \mathbb{Z}_3$	2+2	267	79	79	$\mathbb{Z}_{97} \times \mathbb{Z}_3$	2+2	893	258	146
	4+2	80	24	59		4+2	268	71	110
	8+2	23	13	32		8+2	75	23	62
	16+2	7	22	18		16+2	20	22	32
	32+2	11	42	43		32+2	11	42	39

Table 5.4: Mixing times on the Cayley graphs of the groups $\mathbb{Z}_{53} \times \mathbb{Z}_3$ and $\mathbb{Z}_{97} \times \mathbb{Z}_3$ depending on the number of generators.

The only case where the mixing times seem to always follow the same pattern are the Cayley graphs of abelian groups with $n = p^3$ elements, where p is prime, and $|S| = 6$. There are three possible finite abelian groups with p^3 elements, $\mathbb{Z}_{p^3}$, $\mathbb{Z}_{p^2} \times \mathbb{Z}_p$ and $\mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p$. On the Cayley graph of $\mathbb{Z}_{p^3}$, the shunt-decomposition model performs best, for the Cayley graph of $\mathbb{Z}_{p^2} \times \mathbb{Z}_p$, the lowest mixing time is obtained using the

generalised Diaconis lift and finally, for the graph associated to $\mathbb{Z}_p \times \mathbb{Z}_p \times \mathbb{Z}_p$, the simple random walk yields the best result. This can be seen on the two example in table 5.5 and has been tested for other prime numbers as well.

$n = 5^3$	d	SW	LW	SD_F
$\mathbb{Z}_{125}$	6	106	182	56
$\mathbb{Z}_{25} \times \mathbb{Z}_5$	2+4	89	18	46
$\mathbb{Z}_5 \times \mathbb{Z}_5 \times \mathbb{Z}_5$	2+2+2	8	18	16
$n = 7^3$	d	SW	LW	SD_F
$\mathbb{Z}_{343}$	6	1196	475	174
$\mathbb{Z}_{49} \times \mathbb{Z}_7$	2+4	342	39	93
$\mathbb{Z}_7 \times \mathbb{Z}_7 \times \mathbb{Z}_7$	2+2+2	16	28	26

Table 5.5: Mixing times on the Cayley graphs of abelian groups with $n = p^3$ elements and degree 6.

All of these results show that it is not possible to conclude which walk performs best in general. Which random walk model achieves the lowest mixing time to the uniform distribution depends both on the order of each of the factors given by the structure theorem as well as the number of generators chosen in each factor. We can say however, that the simple random walk should not be entirely disregarded. Trying to determine when the simple random walk performs better than the generalised Diaconis lift and the shunt-decomposition model with Fourier coin could be the subject of future work.

Conclusion

Throughout this work, we have explored different types of random walks with the objective to determine which random walk models allow for fast mixing.

In chapter 1, we started by giving the necessary mathematical background on the quantum computing model. This was important to understand the differences between the classical and the quantum random models, as a quantum system can be in a superposition of states and quantum states evolve through unitary transformations.

The classical random walk models have been presented in chapter 2. In section 2.1, we formalised random walks as finite Markov chains and defined the simple random walk. We then defined the stationary distribution, demonstrated the convergence theorem for irreducible, aperiodic Markov chains and finished the section by mathematically defining the mixing and giving some bounds based on the spectral gap and the conductance. Section 2.2 introduced lifted random walks as well as the Diaconis lift on the cycle. These were followed by the definition of marginal mixing time and a discussion of the bounds on the mixing time depending on the constraints imposed on the lifted walk.

Chapter 3 concentrated on discrete-time quantum walk models. We started by adapting the classical simple random walk on the line to the quantum case. Then we generalised the idea of a coined walk to general regular graphs in the shunt-decomposition model. Finally, we defined the arc-reversal quantum walk on general undirected graphs. Both models have been illustrated by examples. In chapter 4, we defined the average state and average probabilities of a quantum walk, which allowed us to obtain a limiting distribution for quantum walks. To end this chapter, we defined the mixing time for quantum walks and demonstrated bounds for this mixing time depending on the eigenvalues of the unitary transition matrix.

In chapter 5, we were finally able to compare the mixing times of the different random walk models with the objective to establish which model produces the lowest mixing time. Our contributions were the generalisation of the of the Diaconis lift to Cayley graphs of finite abelian groups and the analysis of the mixing times given by our simulations of the simple random walk, the generalised Diaconis lift, the shunt-decomposition walk and the arc-reversal model (using the Fourier and the Grover coin) on Cayley graphs of finite abelian groups with symmetric generating set. First, we observed that not all models converge to the uniform distribution. For the three models converging to the uniform distribution (simple random walk, generalised Diaconis lift and shunt-decomposition walk with Fourier coin), we could not conclude that one model always produces the lowest mixing time. Which model performs best seems to depend on the order of every factor given by the fundamental structure theorem of finite abelian groups and the number of generators chosen in each factor.

This raises some interesting questions for future research. It could be useful to investigate in which situations the simple random walk outperforms the generalised Diaconis lift and the shunt-decomposition walk with Fourier coin. In our simulations, we only considered generating sets, where we have taken generators in each cyclic factor seperately, but how do the different random walk models perform on Cayley graph with "mixed" generators as mentioned in remark 5.4? And how does the generalised Diaconis lift behave in comparison to the other models if we change the probability $1 - \frac{1}{n}$ of moving in the same direction and the probability $\frac{1}{(d-1)n}$ of switching to a different direction?

Appendix A

Hilbert spaces and operators

Definition A.1 (Hilbert space)**.** [19] *A Hilbert space $\mathcal{H}$ is a complex vector space with an inner product*

$$\langle \cdot | \cdot \rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{C} : (\phi, \psi) \mapsto \langle \phi | \psi \rangle$$

that induces a norm

$$|| \cdot || : \mathcal{H} \to \mathbb{R} : \phi \mapsto ||\phi|| := \sqrt{\langle \phi | \phi \rangle}$$

such that $\mathcal{H}$ is complete for this norm.

Most Hilbert spaces we consider are of finite dimension.

Notation A.2. [19] We denote $|\phi\rangle \in \mathcal{H}$ a vector of the Hilbert space $\mathcal{H}$ in Dirac's ket-notation. Moreover, the dual space of $\mathcal{H}$ is given by

$$\mathcal{H}^* = \{f : \mathcal{H} \to \mathbb{C} \mid f \text{ linear and continuous}\},$$

which is in bijection with $\mathcal{H}$. Elements in $\mathcal{H}^*$ are denoted $\langle \psi |$, called the bra-notation. To every $|\phi\rangle \in \mathcal{H}$ corresponds a vector $\langle \phi | \in \mathcal{H}^*$. With this notation, a function $\langle \psi |$ in $\mathcal{H}^*$ applied to a vector $|\phi\rangle$ in $\mathcal{H}$ can be written as $\langle \psi | \phi \rangle \in \mathbb{C}$, yielding the bra-ket notation for the inner product.

Given an orthonormal basis $\{|e_i\rangle\}$ of $\mathcal{H}$, every $|\phi\rangle \in \mathcal{H}$ can be written as

$$|\phi\rangle = \sum_i |e_i\rangle\langle e_i|\phi\rangle = \sum_i \phi_i |e_i\rangle.$$

Likewise, $\{\langle e_i|\}$ is the dual orthonormal basis of $\mathcal{H}^*$ and every $\langle\phi| \in \mathcal{H}^*$ can be described as

$$\langle\phi| = \sum_i \langle\phi|e_i\rangle\langle e_i| = \sum_i \overline{\langle e_i|\phi\rangle}\langle e_i| = \sum_i \overline{\phi_i}\langle e_i|.$$

Remark A.3. Let $\mathcal{H}$ be a Hilbert space with $\dim(\mathcal{H}) = n$ and $\{|e_i\rangle\}_{i=1,\dots,n}$ an orthonormal basis of $\mathcal{H}$. We can identify $\mathcal{H}$ with $\mathbb{C}^n$ by defining an isomorphism $\tau : \mathcal{H} \to \mathbb{C}^n$ mapping the orthonormal basis of $\mathcal{H}$ to the standard basis of $\mathbb{C}^n$:

$$\tau(|e_1\rangle) = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \cdots, \tau(|e_n\rangle) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}.$$

The same can be done for the dual basis $\{\langle e_i|\}$, where we define $\iota : \mathcal{H}^* \to (\mathbb{C}^n)^*$ as

$$\iota(\langle e_1|) = \begin{pmatrix} 1 & 0 & \cdots & 0 \end{pmatrix}, \cdots, \iota(\langle e_n|) = \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix}.$$

By convention, we do not explicitly write τ and ι, but simply write an equality between the element of the orthonormal basis of $\mathcal{H}$ (resp. $\mathcal{H}^*$) and the corresponding element in the standard basis of $\mathbb{C}^n$ (resp. $(\mathbb{C}^n)^*$).

Definition A.4 (Tensor product of Hilbert spaces)**.** [19] *Let $|\phi\rangle \in \mathcal{H}^A$ and $|\psi\rangle \in \mathcal{H}^B$ be two vectors in Hilbert spaces. We define the tensor product of two vectors of $\mathcal{H}^A$ and $\mathcal{H}^B$ as the map*

$$|\phi \otimes \psi\rangle := |\phi\rangle \otimes |\psi\rangle : \mathcal{H}^A \times \mathcal{H}^B \to \mathbb{C} : (\mu, \nu) \mapsto \langle\mu|\phi\rangle_{\mathcal{H}^A}\langle\nu|\psi\rangle_{\mathcal{H}^B}$$

which is anti-linear in each entry and continuous. The tensor product of the Hilbert spaces is defined as

$$\mathcal{H}^A \otimes \mathcal{H}^B = \{\Lambda : \mathcal{H}^A \times \mathcal{H}^B \to \mathbb{C} \mid \textit{anti-linear and continuous}\}.$$

For $|\phi_1\rangle, |\phi_2\rangle \in \mathcal{H}^A$ and $|\psi_1\rangle, |\psi_2\rangle \in \mathcal{H}^B$, we define

$$\langle\phi_1 \otimes \psi_1|\phi_2 \otimes \psi_2\rangle = \langle\phi_1|\phi_2\rangle_{\mathcal{H}^A}\langle\psi_1|\psi_2\rangle_{\mathcal{H}^B}.$$

Proposition A.5. [19] *Let* $\{|e_a\rangle\} \subset \mathcal{H}^A$ *and* $\{|f_b\rangle\} \subset \mathcal{H}^B$ *be two orthonormal bases with* $a \in I$ *and* $b \in J$*. The set* $\{|e_a \otimes f_b\rangle\}$ *is an orthonormal basis of* $\mathcal{H}^A \otimes \mathcal{H}^B$ *and*

$$\dim(\mathcal{H}^A \otimes \mathcal{H}^B) = \dim(\mathcal{H}^A)\dim(\mathcal{H}^B).$$

Proof. First, we show that $\{|e_a \otimes f_b\rangle\} \subset \mathcal{H}^A \otimes \mathcal{H}^B$ is orthonormal. For any $a_1, a_2 \in I$ and $b_1, b_2 \in J$,

$$\langle e_{a_1} \otimes f_{b_1} | e_{a_2} \otimes f_{b_2} \rangle = \langle e_{a_1} | e_{a_2} \rangle \langle f_{b_1} | f_{b_2} \rangle = \delta_{a_1 a_2} \delta_{b_1 b_2},$$

where

$$\delta_{ij} = \begin{cases} 1, & \text{if } i = j. \\ 0, & \text{otherwise.} \end{cases}$$

The set $\{|e_a \otimes f_b\rangle\}$ is generating for $\mathcal{H}^A \otimes \mathcal{H}^B$. Let $\Lambda \in \mathcal{H}^A \otimes \mathcal{H}^B$, then using the anti-linearity of Λ,

$$\begin{aligned} \Lambda(\mu, \nu) &= \Lambda\left(\sum_a |e_a\rangle\langle e_a|\mu\rangle, \sum_b |f_b\rangle\langle f_b|\nu\rangle\right) \\ &= \sum_{a,b} \overline{\langle e_a|\mu\rangle}\,\overline{\langle f_b|\nu\rangle}\,\Lambda(|e_a\rangle, |f_b\rangle) \\ &= \sum_{a,b} \Lambda(|e_a\rangle, |f_b\rangle)\langle \mu|e_a\rangle\langle \nu|f_b\rangle. \end{aligned}$$

By definition A.4, $\langle \mu|e_A\rangle\langle \nu|f_b\rangle = |e_a \otimes f_b\rangle(\mu, \nu)$. Furthermore, if we define $\Lambda_{ab} := \Lambda(|e_a\rangle, |f_b\rangle)$, we obtain

$$\Lambda(\mu, \nu) = \sum_{a,b} \Lambda_{ab} |e_a \otimes f_b\rangle(\mu, \nu).$$

Thus, every $\Lambda \in \mathcal{H}^A \otimes \mathcal{H}^B$ can be expressed as a linear combination of $\{|e_a \otimes f_b\rangle\}$,

$$|\Lambda\rangle = \sum_{a,b} \Lambda_{ab} |e_a \otimes f_b\rangle.$$

Now, suppose

$$0 = \sum_{a,b} \Lambda_{ab} |e_a \otimes f_b\rangle.$$

In particular, for any $(e_{a'}, f_{b'}) \in \mathcal{H}^A \times \mathcal{H}^B$ with $a' \in I$ and $b' \in J$,

$$\begin{aligned} 0 &= \sum_{a,b} \Lambda_{ab} |e_a \otimes f_b\rangle (e_{a'}, f_{b'}) \\ &= \sum_{a,b} \Lambda_{ab} \langle e_{a'}|e_a\rangle \langle f_{b'}|f_b\rangle \\ &= \Lambda_{a'b'}. \end{aligned}$$

We deduce that $\{|e_a \otimes f_b\rangle\}$ is also linearly independent. □

Proposition A.6. [19] *Let $\mathcal{H}^A$ and $\mathcal{H}^B$ be two Hilbert spaces. Then the tensor product $\mathcal{H}^A \otimes \mathcal{H}^B$ is again a Hilbert space.*

Proof. $\mathcal{H}^A \otimes \mathcal{H}^B$ is a complex vector space :

- linearity : For all $\Lambda, \Phi \in \mathcal{H}^A \otimes \mathcal{H}^B$ and $a, b \in \mathbb{C}$, the map

  $$a\Lambda + b\Phi : \mathcal{H}^A \times \mathcal{H}^B \to \mathbb{C} : (\mu, \nu) \mapsto a\Lambda(\mu, \nu) + b\Phi(\mu, \nu)$$

  is continuous since Λ and Φ are continuous. It is also anti-linear in both entries. So $a\Lambda + b\Phi \in \mathcal{H}^A \otimes \mathcal{H}^B$.

- additive neutral element :

  $$0 : \mathcal{H}^A \times \mathcal{H}^B \to \mathbb{C} : (\mu, \nu) \mapsto 0$$

- additive inverse :

  $$-\Lambda : \mathcal{H}^A \times \mathcal{H}^B \to \mathbb{C} : (\mu, \nu) \mapsto -\Lambda(\mu, \nu) := \sum_{a,b} -\Lambda_{ab} |e_a \otimes f_b\rangle (\mu, \nu).$$

  Indeed,

  $$\Lambda - \Lambda = \sum_{a,b} (\Lambda_{ab} - \Lambda_{ab}) |e_a \otimes f_b\rangle = 0 = -\Lambda + \Lambda.$$

Furthermore, we need to define an inner product on $\mathcal{H}^A \otimes \mathcal{H}^B$. The corresponding "bra"-notation of $|\Lambda\rangle \in \mathcal{H}^A \otimes \mathcal{H}^B$ is given by

$$\langle \Lambda| = \sum_{a,b} \overline{\Lambda_{ab}} \langle e_a \otimes f_b|.$$

We can define the inner product

$$\begin{aligned}\langle \Lambda | \Phi \rangle &= \sum_{a_1,b_1} \sum_{a_2,b_2} \overline{\Lambda_{a_1 b_1}} \Phi_{a_2 b_2} \langle e_{a_1} \otimes f_{b_1} | e_{a_2} \otimes f_{b_2} \rangle \\ &= \sum_{a,b} \overline{\Lambda_{ab}} \Phi_{ab}.\end{aligned}$$

This inner product induces the norm

$$||\Lambda||^2 = \langle \Lambda | \Lambda \rangle = \sum_{a,b} \Lambda_{ab}^2.$$

□

Proposition A.5 and proposition A.6 can be generalised to the tensor product between a finite number of Hilbert spaces.

Notation A.7. We write $\mathcal{H}^{\otimes n} = \underbrace{\mathcal{H} \otimes \cdots \otimes \mathcal{H}}_{n}$

Definition A.8 (Adjoint operator)**.** [19] *Let $A : \mathcal{H} \to \mathcal{H}$ be an operator on the Hilbert space Hil. Then $A^* : \mathcal{H} \to \mathcal{H}$ such that*

$$\langle A^* \psi | \phi \rangle = \langle \psi | A \phi \rangle$$

is the adjoint operator. If $A = A^*$, we call A self-adjoint

In the case of finite-dimensional Hilbert spaces, A^* is the conjugate transpose matrix of A and if $A = A^*$, then A is called a Hermitian matrix.

Proposition A.9 (Spectral decomposition)**.** [19] *For any normal operator ($A^*A = AA^*$), there exists an orthonormal basis of eigenvectors $\{|e_{i,k}\rangle\}$ such that*

$$A = \sum_{i \in I} \lambda_i P_i,$$

where for all $i \in I$, $P_i = \sum_{k=1}^{d_i} |e_i, k\rangle\langle e_{i,k}|$ with d_i the multiplicity of eigenvalue λ_i is the projection onto the eigenspace of A associated with λ_i.

If A is Hermitian, all its eigenvalues are real. Indeed, if λ is an eigenvalue with associated eigenvector $|\phi\rangle$, then $A|\phi\rangle = \lambda|\phi\rangle$ and

$$\lambda\langle\phi|\phi\rangle = \langle\phi|\lambda\phi\rangle = \langle\phi|A\phi\rangle = \langle A^*\phi|\phi\rangle = \langle A\phi|\phi\rangle = \langle\lambda\phi|\phi\rangle = \bar{\lambda}\langle\phi|\phi\rangle.$$

So $\lambda = \bar{\lambda}$ and λ has to be real.

If U is unitary, then

$$||\phi|| = ||U\phi|| = ||\lambda\phi|| = |\lambda|||\phi||,$$

from which we deduce that $|\lambda| = 1$. So the eigenvalues of a unitary matrix all have modulus 1.

Definition A.10 (Positive semidefinite operator)**.** [19] *A self-adjoint operator $A : \mathcal{H} \to \mathcal{H}$ is positive semidefinite if for any $|\phi\rangle \in \mathcal{H}$,*

$$\langle\phi|A\phi\rangle \geq 0.$$

Definition A.11 (Trace)**.** [19] *Let $A : \mathcal{H} \to \mathcal{H}$ be an operator and $\{|e_i\rangle\}$ any orthonormal basis of $\mathcal{H}$. The trace of an operator A is defined as*

$$\mathrm{Tr}(A) = \sum_i \langle e_i|Ae_i\rangle = \sum_i A_{ii} \in \mathbb{C}.$$

Proposition A.12 (Cyclic property)**.** *For operators A, B, C,*

$$\mathrm{Tr}(ABC) = \mathrm{Tr}(CAB) = \mathrm{Tr}(BCA).$$

More generally, the trace is invariant under cyclic permutations of the matrix product.

The cyclic property follows from the following equation and associativity of the matrix product,

$$\mathrm{Tr}(AB) = \sum_i (AB)_{ii} = \sum_i \sum_j A_{ij}B_{ji} = \sum_j \sum_i B_{ji}A_{ij} = \sum_j (BA)_{jj} = \mathrm{Tr}(BA).$$

Appendix B

Python code for simulations

This appendix contains the Python code used for the simulations of the simple random walk, the generalised Diaconis walk, the shunt-decomposition walk and the arc-reversal walk. There are probably many ways to ameliorate these implementations. We imported the following packages.

```
import numpy as np
import math
import cmath
from scipy.linalg import schur,eigvals
```

The following function has been used in the implementation of the four random walk models. It returns the list of permutation matrices given by the generators. In the simple random walk and the arc-reversal walk implementation, the sum of these permutation matrices allows us to obtain the adjacency matrix of the graph. This function only works for groups with three cyclic factors (and where we choose generators in each factor seperately), as more were not really necessary for our purposes. It can be extended to more than three factors, but in this case a recursive algorithm would probably be recommended.

```
# returns a list of permutation matrices, given a list M containing the orders in
# decreasing order of the factors and the list S_M containing a list of generators
# for each factor
def Cayley_permutations(M,S_M):

    P = []

    for i in range(len(M)):
        m = M[i]
        S = S_M[i]

        for s in S:
```

```
            p = np.zeros((m,m))
            for v in range(0,m):
                w = (v+s) % m
                p[v,w]=1

            if len(M) == 1 :
                P.append(p)

            if len(M) == 2 :
                if i == 0 :
                    P.append(np.kron(p,np.identity(M[1])))
                else :
                    P.append(np.kron(np.identity(M[0]),p))

            if len(M) == 3 :
                if i == 0 :
                    P.append(np.kron(np.kron(p,np.identity(M[1])),np.identity(M[2])))
                if i == 1 :
                    P.append(np.kron(np.kron(np.identity(M[0]),p),np.identity(M[2])))
                if i == 2 :
                    P.append(np.kron(np.kron(np.identity(M[0]),np.identity(M[1])),p))

    return P


# returns the total variation distance between two probability distributions
def total_variation_distance(q,p):

    tv = 0
    for i in range(len(q)):
        tv=tv+abs(q[i]-p[i])
    return 1/2 * tv
```

B.1 Simple random walk

```
# returns the distribution on the vertices of the graph after t steps of the
# simple random walk, given transition matrix A, initial distribution q_0 and
# number of steps t
def simple_walk(A,q_0,t):

    q_t = np.matmul(q_0,np.linalg.matrix_power(A,t))
    return q_t


# returns the mixing time of the simple random walk with transition matrix A
# and initial distribution q_0 to the stationary distribution limite, epsilon>0
def mixing_time_sw(A,q_0,limite,epsilon):

    Q = np.zeros((10000,),dtype=np.ndarray)
    Q[0] = q_0
    for t in range(1,10000):
        Q[t] = np.matmul(Q[t-1],A)
```

```
        if total_variation_distance(Q[t],limite)<=epsilon:
            return t
    print("t>10000")


# example

M = [5,3]
S_M = [[1,4],[1,2]]
d = 4

P = Cayley_permutations(M,S_M)
A = 1/d * sum(P)

q_0 = np.zeros((np.prod(M),))
q_0[0] = 1

epsilon = 1/4
uniform = 1/np.prod(M)*np.ones((np.prod(M),))

mix = mixing_time_sw(A,q_0,uniform,epsilon)
print(mix)
print(simple_walk(A,q_0,mix))
```

B.2 Generalised Diaconis lift

```
# returns the transition matrix for the generalised Diaconis lift
def lifted_transition_matrix(M,S_M):

    # permutations
    P = Cayley_permutations(M,S_M)

    for i in range(len(M)) :
        m = M[i]
        S = S_M[i]

        # transition matrix for lifted walk
        for j in range(len(P)):
            for k in range(len(P)):
                if j==k:
                    if k==0:
                        A_row = (1-1/m)*P[j]
                    else:
                        A_row = np.concatenate((A_row,(1-1/m)*P[j]),axis=1)
                else:
                    if k==0:
                        A_row = (1/((len(P)-1)*m))*P[j]
                    else:
                        A_row = np.concatenate((A_row,(1/((len(P)-1)*m))*P[j]),axis=1)

            if j==0:
                A = A_row
```

```
            else:
                A = np.concatenate((A,A_row),axis=0)

    return(A)


# returns the marginal distribution on the vertices of the underlying graph,
# given a probability distribution x on the vertices of the lifted graph
def collapse(M,d,x):

    C=np.identity(np.prod(M))
    for i in range(d-1):
        C=np.concatenate((C,np.identity(np.prod(M))),axis=0)

    return np.matmul(x,C)


# returns the marginal probability distribution after t steps of the lifted walk,
# given the transition matrix A and the initial distribution x_0 on the lifted graph
def lifted_walk(A,x_0,t):

    # probabilities on the vertices of the lifted graph
    x_t = np.matmul(x_0,np.linalg.matrix_power(A,t))

    # probabilities on the vertices of the original graph
    q_t = collapse(M,d,x_t)

    return q_t


# returns the marginal mixing time to the stationary distribution limite on the
# underlying graph, given initial distribution x_0, epsilon>0
def marginal_mixing_time(M,S_M,x_0,limite,epsilon):

    A=lifted_transition_matrix(M, S_M)
    X = np.zeros((10000,),dtype=np.ndarray)
    X[0] = x_0
    for t in range(1,10000):
        X[t] = np.matmul(X[t-1],A)
        if total_variation_distance(collapse(M,d,X[t]),limite)<=epsilon:
            return t
        if t%100==0 : print(t)
    print("t>10000")


# example

M = [343]
S_M = [[1,342,2,341,3,340]]
d = 6

#initial distribution on the vertices of the lifted graph
x_0 = np.zeros((np.prod(M)*d,))
x_0[0] = 1
```

```
#transition matrix of lifted walk
A = lifted_transition_matrix(M,S_M)

epsilon = 1/4
uniform = 1/np.prod(M)*np.ones((np.prod(M),))

marginal_mix = marginal_mixing_time(M,S_M,x_0,uniform,epsilon)
print(marginal_mix)
print(lifted_walk(A,x_0,marginal_mix))
```

B.3 Shunt-decomposition quantum walk

The following functions were used for both the shunt-decomposition walk and the arc-reversal walk.

```
# returns the probabilities on the vertices given the amplitudes on the states
def proba(q,n,d):

    q_a = np.abs(q)
    q_s = np.square(q_a)
    P = np.zeros((n,))
    for i in range(0,n):
        for j in range(0,d):
            P[i] = P[i]+q_s[d*i+j]

    return P


# returns the average probabilities on the vertices after T steps
def average_probability(U,q_0,T):

    Q = np.zeros((10000,),dtype=np.ndarray)
    Q[0] = q_0
    average_P = 0
    for t in range(1,T):
        Q[t] = np.matmul(U,Q[t-1])
        average_P = average_P + proba(Q[t],n,d)
    average_P = 1/T*average_P
    return average_P


# returns the eigenprojection of a vector v
def eigenprojection(v):

    v_ct = np.transpose(v.conjugate())
    F_v = np.outer(v_ct,v)
    return np.transpose(F_v)


# returns the spectral decomposition of a transition matrix U
```

```
def spectral_decomposition(U):

    D,Q = schur(U,output='complex')
    Q_T = np.transpose(Q)

    eigenvalues = np.zeros((len(D),)).astype(complex)
    F = []

    for i in range(len(D)):
        eigenvalues[i] = D[i,i]
        F.append(eigenprojection(Q_T[i]))

    return eigenvalues, F


# returns the matrix inner product between two matrices A and B
def matrix_inner_product(A,B):

    p = np.matmul(np.transpose(A.conjugate()),B)
    innerp = np.trace(p)
    return innerp


# returns the limiting distribution of quantum walk with transition matrix U
# starting from initial state q_0
def limiting_distribution(U,q_0):

    n = len(q_0)
    rho_0 = np.outer(np.conjugate(q_0),q_0)
    eigenvalues, F = spectral_decomposition(U)

    limiting_proba_arcs = np.zeros((n,)).astype(complex)
    for i in range(n):
        rho_S = np.zeros((n,n))
        rho_S[i,i] = 1
        proba_arc = 0
        for r in range(n):
            proba_arc = proba_arc + matrix_inner_product(np.matmul(F[r],
                                                    np.matmul(rho_0,F[r])),rho_S)
        limiting_proba_arcs[i] = proba_arc

    limiting_proba_vertices = np.zeros((np.prod(M),)).astype(complex)
    for i in range(0,np.prod(M)):
        for j in range(0,d):
            limiting_proba_vertices[i] = limiting_proba_vertices[i]+
                                                    limiting_proba_arcs[d*i+j]

    return limiting_proba_vertices


def round_complex(x, PlacesReal, PlacesImag):
    return complex(round(x.real,PlacesReal),round(x.imag,PlacesImag))
```

These are the functions implemented in addition to the ones above for the shunt-

decomposition walk.

```
# returns the unitary transition matrix U for the shunt-decomposition walk,
# given a Coin and a list of permutations on the vertices P
def shunt_decomposition_matrix(P,Coin):

    n = len(P[0][0])      # number of vertices
    d = len(P)            # degree of each vertex

    # shift operator
    S = 0
    for i in range(0,len(P)):
        E = np.zeros((len(P),len(P)))
        E[i,i] = 1
        S = S + np.kron(np.transpose(P[i]),E)

    # coin operator
    C = np.kron(np.eye(n),Coin)

    # total unitary operator
    U = np.matmul(S,C)

    return U


# returns the probability distribution on the vertices after t steps of the
# shunt-decomposition walk with transition matrix U, initial state q_0
def shunt_decomposition_walk(U,q_0,t):

    # state and probability on the vertices after t steps
    q_t = np.matmul(np.linalg.matrix_power(U,t),q_0)
    P_t = proba(q_t,n,d)

    return P_t


# returns the mixing time of the shunt-decomposition walk with Coin and shunts P
# starting in state q_0 and with limiting distribution limite, epsilon>0
def mixing_time_sd(P,Coin,q_0,limite,epsilon):

    U = shunt_decomposition_matrix(P,Coin)
    Q = np.zeros((10000,),dtype=np.ndarray)
    Q[0] = q_0
    average_P = 0
    for t in range(1,10000):
        Q[t] = np.matmul(U,Q[t-1])
        average_P = average_P + proba(Q[t],n,d)
        if total_variation_distance(1/t*average_P,limite)<=epsilon:
            return t
    print("t>10000")


# example

M = [7,5]
```

```
S_M = [[1,6],[1,4]]
d = 4
n = np.prod(M)

# Grover coin
G = np.multiply(2/d,np.ones((d,d))) - np.identity(d)

# DFT coin
F = np.zeros((d,d)).astype(complex)
for i in range(d):
    for k in range(d):
        F[i,k] = 1/math.sqrt(d)*round_complex(cmath.exp(complex(0,(2*i*k*math.pi)/d)),15,15)

P = Cayley_permutations(M,S_M)
Coin = G

# initial state
q_0 = np.zeros((n*d,))
q_0[0] = 1

epsilon = 1/4
limite = limiting_distribution(U,q_0)

mix = mixing_time_sd(P,Coin,q_0,limite,epsilon)
print(mix)
print(average_probability(U,q_0,mix))
```

B.4 Arc-reversal quantum walk

```
# returns the unitary transition matrix U for the arc-reversal walk,
# given a Coin and adjacency matrix A
def arc_reversal_matrix(A,Coin):

    n = len(A)
    d = len(Coin)

    # reversal operator
    R = np.zeros((n*d,n*d))
    edge_dict = {}
    k = 0
    for i in range(0,n):
        for j in range(0,n):
            if A[i,j] != 0:
                edge_dict[k] = [i,j]
                k = k+1
    for k in range(0,n*d):
        i = edge_dict[k][0]
        j = edge_dict[k][1]
        l = list(edge_dict.keys())[list(edge_dict.values()).index([j,i])]
        R[k,l] = 1

    # coin operator
```

```
    I = np.eye(n)
    C = np.kron(I,Coin)

    U = np.matmul(R,C)

    return U


# returns the probability distribution on the vertices after t steps of the
# arc-reversal walk with transition matrix U and initial state q_0
def arc_reversal_walk(U,q_0,t):

    # state and probability on the vertices after t steps
    q_t = np.matmul(np.linalg.matrix_power(U,t),q_0)
    P_t = proba(q_t,n,d)

    return P_t


# returns the mixing time of the arc-reversal walk with Coin and
# adjacency matrix A, initial state q_0 and with limiting distribution limite,
# epsilon>0
def mixing_time_ar(A,Coin,q_0,limite,epsilon):

    U = arc_reversal_matrix(A,Coin)
    Q = np.zeros((10000,),dtype=np.ndarray)
    Q[0] = q_0
    average_P = 0
    for t in range(1,10000):
        Q[t] = np.matmul(U,Q[t-1])
        average_P = average_P + proba(Q[t],n,d)
        if total_variation_distance(1/t*average_P,limite)<=epsilon:
            return t
    print("t>10000")


# example

M = [3,3,3]
S_M = [[1,2],[1,2],[1,2]]
d = 6
n = np.prod(M)

# Grover coin
G = np.multiply(2/d,np.ones((d,d))) - np.identity(d)

# DFT coin
F = np.zeros((d,d)).astype(complex)
for i in range(d):
    for k in range(d):
        F[i,k] = 1/math.sqrt(d)*round_complex(cmath.exp(complex(0,(2*i*k*math.pi)/d)),15,15)

A = sum(Cayley_permutations(M,S_M))
Coin = G
```

```
#initial state
q_0 = np.zeros((n*d,))
q_0[0] = 1

epsilon = 1/4
limite = limiting_distribution(U,q_0)

mix = mixing_time_ar(A,Coin,q_0,limite,epsilon)
print(mix)
print(average_probability(U,q_0,mix))
```

Bibliography

[1] Abhijith, J., Adedoyin, A., Ambrosiano, J., Anisimov, P., Bärtschi, A., Casper, W., Chennupati, G., Coffrin, C., Djidjev, H., Gunter, D., et al. Quantum algorithm implementations for beginners. *arXiv e-prints* (2018), arXiv–1804.

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

[3] Ambainis, A. Quantum walks and their algorithmic applications. *International Journal of Quantum Information 1*, 04 (2003), 507–518.

[4] Ambainis, A., Bach, E., Nayak, A., Vishwanath, A., and Watrous, J. One-dimensional quantum walks. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing* (2001), pp. 37–49.

[5] Apers, S., Ticozzi, F., and Sarlette, A. Lifting markov chains to mix faster: limits and opportunities. *arXiv preprint arXiv:1705.08253* (2017).

[6] Babai, L. Automorphism groups, isomorphism, reconstruction (chapter 27 of the handbook of combinatorics). *North-Holland–Elsevier* (1995), 1447–1540.

[7] Chen, F., Lovász, L., and Pak, I. Lifting markov chains to speed up mixing. In *Proceedings of the thirty-first annual ACM symposium on Theory of computing* (1999), pp. 275–281.

[8] Diaconis, P., Holmes, S., and Neal, R. M. Analysis of a nonreversible markov chain sampler. *Annals of Applied Probability* (2000), 726–752.

[9] Gelfand, A. E. Gibbs sampling. *Journal of the American statistical Association 95*, 452 (2000), 1300–1304.

[10] Gerencsér, B. Markov chain mixing time on cycles. *Stochastic processes and their applications 121*, 11 (2011), 2553–2570.

[11] Godsil, C., and Zhan, H. Discrete-time quantum walks and graph structures. *Journal of Combinatorial Theory, Series A 167* (2019), 181–212.

[12] Godsil, C., and Zhan, H. Discrete quantum walks, 2020.

[13] Jerrum, M., Sinclair, A., and Vigoda, E. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM) 51*, 4 (2004), 671–697.

[14] Judson, T. W. *Abstract Algebra: Theory and Applications*. Orthogonal Publishing, 2013.

[15] Jungers, R. *Lecture notes in Graph Theory*. Université Catholique de Louvain, 2018.

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

[17] Levin, D. A., and Peres, Y. *Markov chains and mixing times*, vol. 107. American Mathematical Soc., 2017.

[18] Lovász, L., and Simonovits, M. Random walks in a convex body and an improved volume algorithm. *Random structures & algorithms 4*, 4 (1993), 359–412.

[19] Scherer, W. *Mathematics of Quantum Computing*. Springer, 2019.

[20] Zhan, H. *Discrete Quantum Walks on Graphs and Digraphs*. PhD thesis, University of Waterloo, 2018.

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences
Place des sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | **www.uclouvain.be/sc**