Quantum Walk Frameworks and Quantum Speedup of Markov Mixing

Jeremy Bernstein
Trinity College, Cambridge, UK
jb866@cam.ac.uk

Mentors: Thomas Vidick and Stacey Jeffery

September 25, 2015

Abstract

We give a novel quantum algorithm for Markov mixing, using ideas from two frameworks for quantising algorithms on reversible, discrete time Markov chains. The *MNRS quantum walk framework* (MNRS) is based on the spectral properties of Markov chains, and the *Belovs electric network framework* (Belovs) is based on the resistor network interpretation.

Given a Markov chain with spectral gap δ and minimal stationary probability π_*, classical mixing takes $\mathrm{O}(\frac{1}{\delta}\log\frac{1}{\pi_*})$ applications of the transition matrix. In line with quadratic speedups via quantum walks in other areas, Richter conjectured a quantum mixing time of $\mathrm{O}(\frac{1}{\sqrt{\delta}}\log\frac{1}{\pi_*})$ queries to the quantum walk operator. This problem is still open. We give a novel algorithm inspired by Belovs, which mixes in $\mathrm{O}(\frac{1}{\sqrt{\delta'}}+\frac{1}{\sqrt{\delta}})$ queries, where δ' is the spectral gap of the originial Markov chain modified by the addition of self loops.

Our algorithm is as follows: we first mix to the stationary distribution of the modified Markov chain (self loops added). We then mix to the stationary distribution of the original chain. This process guarantees O(1) overlap between initial, intermediate and final state. We show that in certain cases this algorithm beats Richter's conjecture, and we provide a means of quantifying when our algorithm might beat both existing algorithms and Richter's conjecture in terms of Cheeger's inequality, providing a link between quantum mixing times and Markovian bottlenecks.

For cases where adding self loops does not asymptotically affect the spectral gap, our algorithm can be prepended to the MNRS quantum walk framework yielding a new spectral quantum walk framework that can start from an arbitrary initial state at no extra cost over MNRS. This is a feature of the Belovs' framework absent from MNRS. That our algorithm is based on Belovs' walk operator is suggestive of power in Belovs' framework absent from MNRS.

1 Introduction

Algorithms based on random walks have a wide applicability: from problems in theoretical Computer Science to Google's PageRank algorithm. It is natural to ask whether the power of quantum computing might yield faster such algorithms. The answer is yes, as exemplified by element distinctness: the problem of determining whether a list of N items contains any repeated entries. Classically we require at least N queries to the list, as we must examine the entire list. But Ambainis gives a quantum walk algorithm requiring only $\mathrm{O}(N^{\frac{2}{3}})$ quantum queries [1].

A quantum walk algorithm makes use of a unitary walk operator W analogous to a classical Markov transition matrix P. The algorithm *walks* towards a solution, by which we mean we update our initial state by repeated application of W. At intermediate steps along the walk we perform a *checking* operation, which is analogous to classically checking whether we've reached a valid solution. Efficient

algorithms in this paradigm incrementally update a data structure at every step of the walk to ensure a cheap checking operation.

We can abstract away the problem of specifying quantum walk algorithms by using tools called quantum walk frameworks. These frameworks tell us the cost of implementing general quantum walk algorithms in terms of

1. setup cost – of setting up the initial state,
2. update cost – of implementing the quantum walk operator,
3. checking cost – of checking the validity of a solution,
4. properties of underlying Markov chain – either spectral or electric.

One spectral framework, due to Magniez, Nayak, Roland and Santha (MNRS [2]), takes the quantum encoding of the Markov chain's stationary distribution and sends it to some desired state. Symbolically we map $|\pi\rangle \to |\mu\rangle$. The reverse of this map can be thought of as a quantum analogue of a process known as *Markov mixing*. Markov mixing is the problem of generating the stationary distribution of a Markov chain. This is at the heart of Google's PageRank algorithm for ranking websites [3], and it is also important in Physics in the context of Markov Chain Monte Carlo [4].

Although simply reversing the MNRS framework yields a quantum speedup of Markov mixing in some cases, the speedup is not general. Other more advanced algorithms have been developed that offer quantum speedup in other special cases. But still a conjecture due to Richter of a general quadratic quantum speedup of Markov mixing goes unproven [5].

In this paper we offer a novel quantum mixing algorithm based on ideas from quantum walk frameworks – the MNRS spectral framework and Belovs' electric network framework in particular. Our algorithm uses the spectral properties of a walk operator inspired by that of Belovs' framework [6]. We explain our algorithm intuitively with the notion of adding self loops to the original Markov chain.

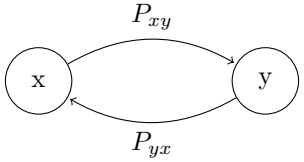
Self loops have traditionally been added to Markov chains to make them *lazy* – i.e. a walker stays put in a given state with probability $\frac{1}{2}$. More recently the idea of adding self loops of variable strength yet in a uniform manner has been considered in the context of quantum algorithms, with Wong coining the term *lackadaiscal quantum walks* [7]. But we go further by considering adding self loops in proportion to some distribution.

The structure of the paper is as follows: in Section 2 we give the theoretical background of classical and quantum walks, as well as some necessary quantum algorithm primitives. In Section 3 we present our algorithm. In Sections 4 and 5 we discuss our results and sketch ideas for further work. Throughout the paper we place the emphasis on building intuition, and as often as possible defer rigorous proof to the references.

2 Theoretical background

2.1 Classical random walks

A classical random walk, or Markov chain, involves moving around a set of states $x \in X$ with transition probabilities $\mathbb{P}(x \to y) = P_{xy}$.



If there are n states, then the initial probability distribution over states can be given by an n dimensional row vector $\vec{\sigma}_0$. It follows that after a single time step the new distribution $\vec{\sigma}_1 = \vec{\sigma}_0 P$ and after t time steps $\vec{\sigma}_t = \vec{\sigma}_0 P^t$.

In broad terms, a Markov chain is *ergodic* if after starting in any state and taking some large number of steps there is a non-zero probability of being in any state. Ergodic Markov chains always relax to a stationary distribution, i.e. $\vec{\sigma}_\infty$ is unique and independent of $\vec{\sigma}_0$. Proving this relies on the fact that $\vec{\sigma}_\infty$ is the unique 1-eigenvector of the chain, and that there is a spectral gap $\delta = 1 - |\lambda_2|$ between 1 and the second largest magnitude eigenvalue. The intuition is that the existence of the spectral gap ensures other eigenvectors will be damped away as they have an eigenvalue of magnitude less than one. Having reached the stationary distribution, we refer to the Markov chain as *at equilibrium*. See for example [8] for more detail.

2.1.1 Mixing to the stationary distribution

An important time scale associated with a classical Markov chain is the mixing time τ, which gives a measure of how many steps of a classical walk are needed to relax to the stationary distribution. We can bound the classical mixing time tightly by the spectral gap [8]. The intuition for this is that a larger spectral gap corresponds to non-stationary eigenvectors with smaller magnitude eigenvalues, hence these eigenvectors will be damped away faster.

We define $\tau(\epsilon)$ as the number of steps needed to be ϵ close to the stationary distribution, maximised over starting states. The notion of closeness can be taken to be the total variation distance. For small δ and $\pi_* = \min_{j \in X} \pi_j$, then

$$\tau(\epsilon) \leq \frac{1}{\delta} \log\left(\frac{1}{\epsilon \pi_*}\right) = \tilde{\mathrm{O}}\left(\frac{1}{\delta}\right) \tag{1}$$

Mixing is generally slowed down if bottlenecks are present in the chain, i.e. the state space contains some *narrow* regions – see Figure 1 for an example. In Section 2.1.3 we will introduce Cheeger's inequality, which bounds the spectral gap and hence the mixing time in terms of the narrowest bottleneck of a graph.

2.1.2 Reversible Markov chains

We can visualise the evolution of a Markov chain as many random walkers moving around the nodes of a graph, and think of a current of walkers moving between the nodes. At equilibrium the fraction of walkers moving from node x to node y, w_{xy}, is $\pi_x P_{xy}$. The Markov chain is reversible if and only if $w_{xy} = w_{yx}$, i.e. equal numbers of walkers moving in each direction, symmetric under time reversal. A reversible Markov process at equilibrium is therefore fully specified by the symmetric weight matrix w_{xy}.

Defining the total weight $W = \sum_{xy} w_{xy}$, it is quick to check the following relations:

$$P_{xy} = \frac{w_{xy}}{\sum_v w_{xv}} \qquad \pi_x = \frac{\sum_v w_{xv}}{W} \qquad \pi_x P_{xy} = \frac{w_{xy}}{W} \tag{2}$$

2.1.3 Cheeger's inequality

Cheeger's inequality provides a means of bounding the spectral gap of a Markov chain. Here we explain the intuition, and then state the result.

Consider a Markov chain after it has relaxed to the stationary distribution, and partition its state space in two. Call the subset with smaller total stationary probability the *less likely* subset. We're less likely to observe a random walker at equilibrium in this subset. If we can show that the probability a random walker jumps from the less likely subset to the more likely subset in a single step is very small (we'll call this probability ϕ), then this indicates there is a region of the state space that takes a

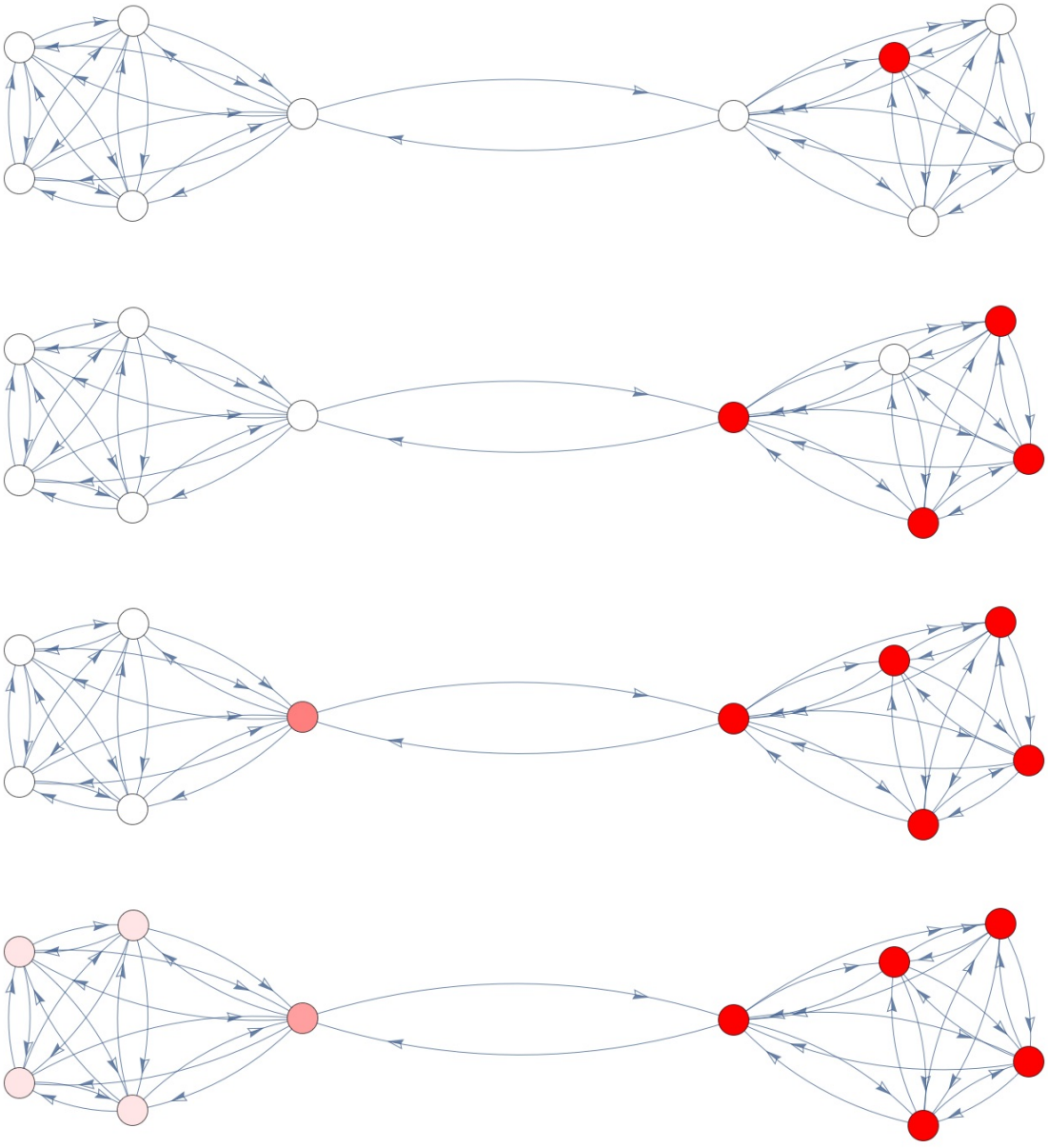


Figure 1: From top to bottom: the right hand subgraph mixes rapidly, but it takes many more steps to cross the bottleneck.

long time to escape. As our initial state can lie within the less likely subset, we expect a long mixing time, and hence a small spectral gap.

Formally, we partition the state space $X = S \cup \bar{S}$ and define $\pi(S) = \sum_{x \in S} \pi_x$. Then ϕ as a function of the partitioning satisfies $\pi(S)\phi(S) = \sum_{x \in S, y \in \bar{S}} w_{xy}$, where w_{xy} is the usual Markov weight matrix. We are interested in the tightest bound, so we want to minimise ϕ with respect to S. We define the Cheeger constant

$$\phi_* = \min_{S:\pi(S) \leq \frac{1}{2}} \phi(S) \tag{3}$$

and then the bound on the spectral gap [8] (without proof) is

$$\frac{\phi_*^2}{2} \leq \delta \leq 2\phi_* \tag{4}$$

2.2 Quantum random walks

A single step in a classical Markov chain is implemented via the map

$$\sigma_x \to \sum_y P_{xy} \sigma_y \tag{5}$$

Therefore a natural first attempt to define a quantum walk might involve operations looking like

$$|x\rangle \to \sum_y \sqrt{P_{xy}} \, |y\rangle \tag{6}$$

Where a classical random walk probabilistically travels from one definite state to another definite state, a quantum walk would diffuse probability over a state's neighbours, only collapsing to a definite state upon measurement. But unfortunately the operation in Equation 6 is not reversible.

In quantum algorithms, the conventional way to make a non-reversible operation reversible is to keep track of the input. So instead we define our quantum update operation as

$$|x\rangle \, |0\rangle \to \sum_y \sqrt{P_{xy}} \, |x\rangle \, |y\rangle \tag{7}$$

and we also require the symmetric operation

$$|0\rangle \, |y\rangle \to \sum_x \sqrt{P_{yx}} \, |x\rangle \, |y\rangle \tag{8}$$

2.2.1 The quantum walk operator

We define the Szegedy walk operator [9] as the composition of reflections $W = \mathrm{Ref}(\mathcal{B}) \circ \mathrm{Ref}(\mathcal{A})$ where subspaces $\mathcal{A}$ and $\mathcal{B}$ are given by

$$\mathcal{A} = \mathrm{Span} \left\{ \sum_y \sqrt{P_{xy}} \, |x\rangle \, |y\rangle : x \in X \right\} \tag{9}$$

$$\mathcal{B} = \mathrm{Span} \left\{ \sum_x \sqrt{P_{yx}} \, |x\rangle \, |y\rangle : y \in X \right\} \tag{10}$$

It turns out (as explained later in Section 2.2.5) that these reflections are easy to implement provided access to the maps given in Equations 7 and 8. The procedure is essentially to reverse the map, reflect in the computational basis, then reimplement the map.

One thing we like about the classical transition matrix P is that it picks out the stationary vector $\vec{\pi}$ as special. $\vec{\pi}$ has eigenvalue 1 and spectral gap δ distinguishing it from all other eigenvalues. Remarkably the quantum walk operator W has similar properties.

We can encode the stationary distribution into a quantum state naturally as $\sum_x \sqrt{\pi_x}\,|x\rangle$. Consider applying the quantum update operation of Equation 7 to this state. We get

$$|\pi\rangle = \sum_{xy} \sqrt{\pi_x}\sqrt{P_{xy}}\,|x\rangle\,|y\rangle \tag{11}$$

Restricting down to the class of reversible Markov chains, we have $\pi_x P_{xy} = \pi_y P_{yx}$. It's easy to see then that $|\pi\rangle$ lies in the space $\mathcal{A}\cap\mathcal{B}$ and so is invariant under reflections in those spaces. Hence $|\pi\rangle$ is invariant under action of the walk operator. Where $\vec{\pi}$ is a one-eigenvector of the classical transition matrix P, $|\pi\rangle$ is a one-eigenstate of the quantum walk operator W.

But the analogy goes deeper according to Szegedy's theorem. Szegedy's theorem tells us that if we restrict our attention to the space $\mathcal{A}\cup\mathcal{B}$ then $|\pi\rangle$ is the unique one-eigenstate, and it is separated by a phase gap of $\sqrt{\delta}$ from all other eigenvalues. This square root increase in the gap over the classical transition matrix is the source of quantum walk speedup. Our statement of the theorem below is based on that in [2]. It's important to note that $D(P)$ is similar to P so has the same spectrum.

Theorem 1 (Szegedy's theorem)**.** *Let P be an irreducible Markov chain, and let $\cos\theta_1$, ..., $\cos\theta_l$ be an enumeration of those singular values (possibly repeated) of $D_{xy} = \sqrt{\frac{\pi_x}{\pi_y}}P_{xy}$ that lie in the open interval (0, 1). Then:*

1. *On $\mathcal{A}\cup\mathcal{B}$ those eigenvalues of $W(P)$ that have non-zero imaginary part are exactly $\exp(\pm 2i\theta_1)$, ..., $\exp(\pm 2i\theta_l)$, with the same multiplicity.*
2. *On $\mathcal{A}\cap\mathcal{B}$ the operator $W(P)$ acts as the identity. The linear subspace $\mathcal{A}\cap\mathcal{B}$ is spanned by the left (and right) singular vectors of $D(P)$ with singular value 1.*
3. *On $\mathcal{A}\cap\mathcal{B}^\perp$ and $\mathcal{A}^\perp\cap\mathcal{B}$ the operator $W(P)$ acts as -Id. The linear subspace $\mathcal{A}\cap\mathcal{B}^\perp$ (respectively, $\mathcal{A}^\perp\cap\mathcal{B}$) is spanned by the set of left (respectively, right) singular vectors of $D(P)$ with singular value 0.*
4. *$W(P)$ has no other eigenvalues on $\mathcal{A}\cup\mathcal{B}$; on $(\mathcal{A}\cup\mathcal{B})^\perp$ the operator $W(P)$ acts as Id.*

2.2.2 Quantum mixing

Just as we have a classical notion of mixing from some initial distribution $\vec{\sigma}$ to the stationary distribution $\vec{\pi}$, we can define quantum mixing as the process of mapping from some arbitrary initial state $|\sigma\rangle$ to the stationary state $|\pi\rangle$.

We defined the classical mixing time as the number of applications of the classical transition matrix P needed to reach some measure of closeness to the stationary distribution $\vec{\pi}$. We make an analogous definition of quantum mixing time as the number of applications of the Szegedy walk operator W needed to reach some measure of closeness to the stationary state $|\pi\rangle$.

This definition of quantum mixing time is motivated by the close resemblance of the quantum update rule $|x\rangle\,|0\rangle \to \sum_y \sqrt{P_{xy}}\,|x\rangle\,|y\rangle$ to the classical update rule $\sigma_x \to \sum_y P_{xy}\sigma_y$.

Classically we have a mixing time $\tau_{cl} = \tilde{\mathrm{O}}\left(\frac{1}{\delta}\right)$ [8] which led Richter to conjecture a quadratic quantum speedup of $\tau_q = \tilde{\mathrm{O}}\left(\frac{1}{\sqrt{\delta}}\right)$ [5].

2.2.3 State generation via reflection

Since allowed quantum operations are unitary, meaning unit vectors get sent to unit vectors, we can view the transformation of one quantum state into another as a rotation. Grover search and

amplitude amplification provide a means of implementing such a rotation by composing a sequence of sub-rotations. Each sub-rotation keeps the state of the system lying within the plane defined by the initial and final state. See Figure 2 for the general geometric picture.

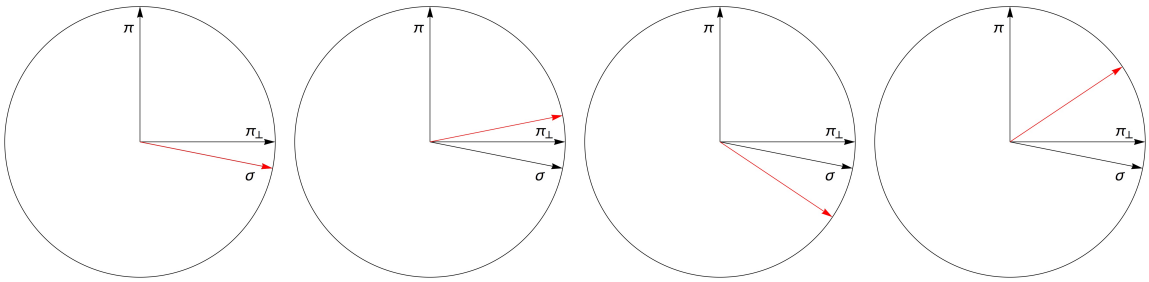


Figure 2: From left to right: $|\sigma\rangle$ is rotated towards $|\pi\rangle$ by a series of reflections.

Let $|\sigma\rangle$ be our initial state, $|\pi\rangle$ our desired state, and $|\pi^\perp\rangle$ the orthogonal to $|\pi\rangle$ in the plane spanned by $|\pi\rangle$ and $|\sigma\rangle$. The required sub-rotation can be implemented by first reflecting the system in $|\sigma\rangle$ and then in $|\pi^\perp\rangle$. This can be appreciated by inspecting the geometry of Figure 2.

The trick to this method of state generation is knowing how many sub-rotations to implement before we start to overshoot. From the geometry we can tell that the angle 2θ of each sub-rotation satisfies $\langle\sigma|\pi\rangle = \sin\theta \approx \theta$. Since the total angle is $\mathrm{O}(1)$ this means we need $\mathrm{O}(\frac{1}{\langle\sigma|\pi\rangle})$ subrotations. Here we have assumed θ is small, which is justified because otherwise $|\sigma\rangle$ and $|\pi\rangle$ would have large overlap and we could easily prepare $|\pi\rangle$ with high probability by one round of phase estimation on $|\sigma\rangle$ to precision $\frac{1}{\sqrt{\delta}}$ followed by a measurement of the phase bit (see Sections 2.2.4 and 2.2.5).

So we have reduced the problem of state generation to the problem of being able to reflect in our initial and final state [2]. Given access to $\mathrm{Ref}(\sigma)$ at cost C_σ and $\mathrm{Ref}(\pi)$ at cost C_π then we can rotate $|\sigma\rangle$ into $|\pi\rangle$ at total cost $\frac{C_\sigma + C_\pi}{\langle\sigma|\pi\rangle}$.

2.2.4 Phase estimation

Phase estimation is a quantum algorithm which, given a unitary operator U, maps eigenvectors of U to corresponding eigenvalues. The algorithm takes an eigenstate of U denoted $|\phi_k\rangle$ with unknown eigenvalue $\exp(i\theta_k)$, and carries out the map

$$|\phi_k\rangle\,|0\rangle \to |\phi_k\rangle\,|\tilde{\theta}_k\rangle \tag{12}$$

where $\tilde{\theta}_k$ is an approximation to θ_k. The precision of this approximation is improved by allowing the algorithm to make more queries to the unitary U. Specifically, to estimate θ_k to precision δ requires $\mathrm{O}(\frac{1}{\delta})$ applications of U.

Strictly speaking, the algorithm only outputs a state with high overlap with $|\tilde{\theta}_k\rangle$, although the overlap can be made arbitrarily large at additional logarithmic cost, so we do not worry. We base our statement of phase estimation on the one in [10].

Theorem 2 (Phase estimation)**.** *Let U be some $2m$-dimensional unitary. Then for any $s \geq 1$, there exists a quantum algorithm Phase(U, s) such that*

1. *Phase(U, s) acts on $m + s$ qubits;*
2. *Phase(U, s) consists of $O(s^2)$ elementary gates, and $2s + 1$ controlled calls to U;*
3. *if $U\,|\psi\rangle = |\psi\rangle$ then Phase(U, s) sends $|\psi\rangle\,|0^s\rangle$ to itself;*

4. *and if* $U|\psi\rangle = \exp i\theta |\psi\rangle$*, for* $\theta \in (-\pi, 0) \cup (0, \pi]$*, then Phase*$(U, s)$ *sends* $|\psi\rangle|0^s\rangle \to |\psi\rangle|\omega\rangle$ *for some* $|\omega\rangle$ *such that* $|\langle 0^s|\omega\rangle| \leq \frac{\pi}{|\theta|2^s}$.

We say that Phase(U, s) *estimates the phase of* U *with precision* $\frac{1}{2^s}$.

We are interested in phase estimation as it allows us to implement a reflection in gapped eigenvectors of unitary operators, as described in Section 2.2.5.

2.2.5 Means of implementing reflections

We have reduced the problem of implementing the map $|\sigma\rangle \to |\pi\rangle$ to the problem of implementing the reflections $\mathrm{Ref}(\sigma)$ and $\mathrm{Ref}(\pi)$. Three typical means of implementing reflections in quantum algorithms are:

1. access to oracle,
2. ability to create a state,
3. MNRS reflection via phase estimation.

The first method is the simplest – we assume we have access to a reflection oracle as in Grover search, which acts as the identity on unmarked states and flips the sign of marked states.

The second method assumes we have access to a creation map $|0\rangle \to |\sigma\rangle$ and its reverse. To reflect our state $|\psi\rangle$ around $|\sigma\rangle$, we run the map in reverse, reflect around $|0\rangle$ and then run the map forwards. This process is a change of basis to one where the reflection is easy, a reflection, and then a change back.

The third method is the most interesting, and applies when we wish to reflect around an eigenvector of a unitary with a gapped eigenvalue [2]. We describe the method for the case of reflecting around the 1-eigenvector, $|\pi\rangle$, of the Szegedy walk operator, W, which is surrounded by a phase gap of $\sqrt{\delta}$. Formally, we can use phase estimation to implement the map

$$|\pi\rangle + \sum_i |\pi_i^\perp\rangle \to |\pi\rangle - \sum_i |\pi_i^\perp\rangle \tag{13}$$

where the $|\pi_i^\perp\rangle$ are arbitrary states orthogonal to $|\pi\rangle$. The reflection procedure is:

$$\begin{aligned}
&|\pi\rangle|0\rangle + \sum_i |\pi_i^\perp\rangle|0\rangle \\
\to &|\pi\rangle|0\rangle + \sum_i |\pi_i^\perp\rangle|\tilde{\theta}_i\rangle \quad \text{(phase estimate to precision } \sqrt{\delta} \text{ in register 2)} \\
\to &|\pi\rangle|0\rangle - \sum_i |\pi_i^\perp\rangle|\tilde{\theta}_i\rangle \quad \text{(apply negative sign to any non-zero register 2)} \\
\to &|\pi\rangle|0\rangle - \sum_i |\pi_i^\perp\rangle|0\rangle \quad \text{(uncompute register 2)}
\end{aligned} \tag{14}$$

Phase estimation to precision $\sqrt{\delta}$ requires $\mathrm{O}(\frac{1}{\sqrt{\delta}})$ applications of W, each of cost U. Therefore this procedure enables us to reflect around $|\pi\rangle$ at cost $\frac{U}{\sqrt{\delta}}$.

2.2.6 Naïve quantum mixing algorithm – reverse-MNRS

Combining the results of the previous sections gives us a simple mixing algorithm for mapping $|\sigma\rangle \to |\pi\rangle$. We showed that implementing this mapping as a rotation had cost $\frac{C_\sigma + C_\pi}{\langle\sigma|\pi\rangle}$, where C_σ and C_π were the costs of the respective reflections. Reflection around $|\sigma\rangle$ can be carried out at cost S, the cost of

creating $|\sigma\rangle$. Reflection around $|\pi\rangle$ can be carried out at cost $\frac{U}{\sqrt{\delta}}$ via MNRS reflection. Assuming U = S = 1, we have a total quantum mixing time of

$$\tau_q \sim \frac{1}{\langle\sigma|\pi\rangle\sqrt{\delta}} \tag{15}$$

This expression improves the classical result with respect to the spectral gap by a square root factor, which is highly desirable. But we have an unpleasant dependence on the initial overlap – if $\langle\sigma|\pi\rangle$ is exponentially small, then this naïve algorithm will take exponentially long to mix.

We can say more if we are able to cheaply prepare the uniform mixture over states, $|\sigma\rangle = \frac{1}{\sqrt{N}}\sum_x |x\rangle$. Then we are guaranteed $\langle\sigma|\pi\rangle > \frac{1}{\sqrt{N}}$, and $\tau_q < \sqrt{\frac{N}{\delta}}$. For Markov chains where the spectral gap scales with system size, this quantum algorithm will present an advantage over the classical case for $\delta = \mathrm{O}\left(\frac{1}{N}\right)$.

This quantum mixing algorithm is essentially the MNRS quantum walk framework run in reverse. Much of the further work on quantum mixing algorithms consists of trying to improve upon the dependence on initial overlap of reverse-MNRS.

2.2.7 More advanced quantum mixing algorithms

Wocjan and Abeyesinghe suggest one improvement to the reverse-MNRS mixing algorithm [11]. Instead of rotating $|\sigma\rangle$ directly into $|\pi\rangle$ when the overlap might be poor, they suggest rotating through a series of intermediate states $|\pi_i\rangle$ which are stationary with respect to intermediate Markov chains and walk operators W_i. Hence we send $|\sigma\rangle \to |\pi_0\rangle \to |\pi_1\rangle \to ... \to |\pi_n\rangle \to |\pi\rangle$. Provided neither the gaps of the intermediate W_i nor the intermediate overlaps $\langle\pi_i|\pi_{i+1}\rangle$ are too small, this algorithm will present an advantage over plain reverse-MNRS.

This method can be seen as an analogue of quantum mixing via adiabatic state generation [12]. In adiabatic state generation, we take Hamiltonians $\mathcal{H}_\sigma$ and $\mathcal{H}_\pi$ that have ground states $|\sigma\rangle$ and $|\pi\rangle$ respectively. We start the system in $|\sigma\rangle$ under $\mathcal{H}_\sigma$, and by shifting the Hamiltonian slowly enough towards $\mathcal{H}_\pi$ the system will remain in the ground state and end up in $|\pi\rangle$. This relies on the intermediate Hamiltonians being gapped, just as Wocjan and Abeyesinghe's algorithm relies on the intermediate Markov chains being gapped. For a reversible Markov transition matrix P_{xy} with stationary distribution π_x, the appropriate Hamiltonian is $\mathcal{H}_{xy} = \delta_{xy} - \sqrt{\frac{\pi_x}{\pi_y}}P_{xy}$. This is appropriate because $\mathcal{H}_{xy}$ defined in this way has ground state $|\pi\rangle$ and spectral gap δ[1]. Wocjan and Abeyesinghe's algorithm presents a speedup[2] over adiabatic state generation precisely because the Szegedy walk operator W has a larger gap than $\mathcal{H}_{xy}$. Still it does not reach Richter's conjectured bound on mixing time.

Another quantum mixing algorithm of interest is due to Richter himself and is based on dechohering quantum walks [5]. Richter shows that this algorithm achieves his conjectured bound in certain cases, although not in general.

3 Contribution

3.1 General idea

The reverse-MNRS mixing algorithm suffers when the initial state has poor overlap with the stationary distribution. We look for ways to modify the Markov chain to increase the overlap. We can then first mix to the stationary distribution of the modified chain, and then mix to the stationary distribution of the original chain.

[1] Not to be confused with the Kronecker delta in the definition of $\mathcal{H}_{xy}$.

[2] The magnitude of the speedup is slightly vague as different authors use different adiabatic theorems, so the speed of the adiabatic algorithm isn't entirely clear.

In Figure 3, the starred node in the left hand graph has exponentially small stationary weight if we assume a is exponentially small. A random walk on this graph will spend the overwhelming majority of time in the outer four nodes, and the reverse-MNRS algorithm would perform poorly at mixing from the starred node. Adding a self loop can overcome this problem by causing the random walk to spend more time in the starred node at equilibrium, i.e. by boosting the stationary weight of this node.

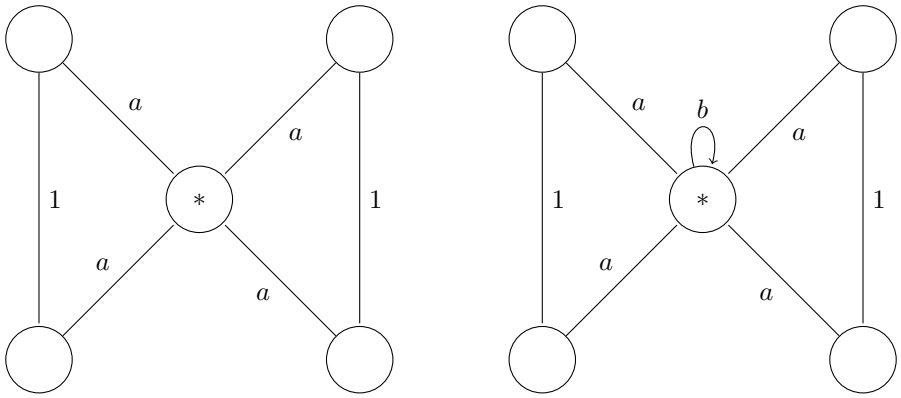


Figure 3: The edge labels correspond to the weights, w_{xy}. Assume a is exponentially small. In the right hand graph, a self loop has been added to the starred node. Tuning b can dramatically increase the overlap of a state initialised to this node with the stationary distribution.

3.2 The Belovs walk operator

3.2.1 Definition

Take the original reversible Markov chain, which is specified by the weight matrix w_{xy}. We'll assume that the original Markov chain has no self loops, corresponding to zeroes on the leading diagonal of w_{xy}. Then the Szegedy walk operator for this Markov chain involves reflections in states of the form:

$$|\psi_x\rangle = \sqrt{\pi_x}\,|x\rangle \left(\sum_y \sqrt{P_{xy}}\,|y\rangle\right) = \sum_y \sqrt{\frac{w_{xy}}{W}}\,|x\rangle\,|y\rangle \tag{16}$$

$$|\phi_y\rangle = \left(\sum_x \sqrt{P_{yx}}\,|x\rangle\right) \sqrt{\pi_y}\,|y\rangle = \sum_x \sqrt{\frac{w_{xy}}{W}}\,|x\rangle\,|y\rangle \tag{17}$$

We note that the state

$$|\pi\rangle = \sum_x |\psi_x\rangle = \sum_y |\phi_y\rangle = \sum_{xy} \sqrt{\frac{w_{xy}}{W}}\,|x\rangle\,|y\rangle \tag{18}$$

is stationary with respect to the Szegedy walk operator, since it is invariant under the reflections.

Now consider a new weight matrix formed by adding self loops to the old one. Let $w'_{xy} = w_{xy} + A\sigma_x$. Here A is some constant that can be tuned to vary the total strength of the self loops, and σ_x is a probability distribution corresponding to our initial state $|\sigma\rangle$. The new total weight is $W' = W + A$. We define the Belovs walk operator to be the Szegedy walk operator for this modified chain. The addition of self loops of finite strength does not break properties such as connectedness or ergodicity,

therefore the stationary distribution is still uniquely defined. We note that the state

$$|\rho\rangle = \sum_x \sqrt{\frac{\sigma_x A}{W+A}} |x\rangle |x\rangle + \sum_{xy} \sqrt{\frac{w_{xy}}{W+A}} |x\rangle |y\rangle \tag{19}$$

$$= \sqrt{\frac{A}{W+A}} |\sigma\rangle + \sqrt{\frac{W}{W+A}} |\pi\rangle \tag{20}$$

is stationary with respect to the Belovs walk operator.

This walk operator is inspired by the reflections made in Belovs' electric network quantum walk framework, which couples the walk operator to the initial state. Our conceptual shift is to put this coupling on the same footing as the other terms in the walk operator by introducing the notion of self loops.

3.2.2 Implementation

If the Szegedy walk operator can be implemented at cost U, then so can the Belovs walk operator. To show this we assume access to two update operations: the Szegedy update from the original chain and another simple update:

$$|x\rangle |0\rangle \to \sum_y \sqrt{\frac{w_{xy}}{W}} |x\rangle |y\rangle \tag{21}$$

$$|x\rangle |1\rangle \to \sqrt{\sigma_x} |x\rangle |x\rangle \tag{22}$$

Provided the original chain has no self loops, both these maps are clearly reversible. The first costs U and the second should be less expensive.

The Belovs update can then be implemented by the two step map

$$|x\rangle |0\rangle \to \sqrt{\frac{A}{W+A}} |x\rangle |1\rangle + \sqrt{\frac{W}{W+A}} |x\rangle |0\rangle \tag{23}$$

$$\to \sqrt{\frac{A\sigma_x}{W+A}} |x\rangle |x\rangle + \sum_y \sqrt{\frac{w_{xy}}{W+A}} |x\rangle |y\rangle \tag{24}$$

In the next section we will let $A = W$ for simplicity. Now that we have set up the idea of the Belovs walk operator, our mixing algorithm is simple.

3.3 Novel mixing algorithm

We wish to map $|\sigma\rangle \to |\pi\rangle$ and we have access to the Belovs walk operator B and the Szegedy walk operator W.

W has stationary state $|\pi\rangle$ and phase gap $\sqrt{\delta}$. B has stationary state $|\rho\rangle = \frac{1}{\sqrt{2}} |\sigma\rangle + \frac{1}{\sqrt{2}} |\pi\rangle$ and phase gap $\sqrt{\delta'}$, the phase gap of the Markov chain with self loops added. We opt to implement the map $|\sigma\rangle \to |\rho\rangle \to |\pi\rangle$.

3.3.1 Statement of algorithm

Our algorithm is as follows:

1. Phase estimate on $|\sigma\rangle$ with respect to B to precision δ'
2. Measure the phase bit
3. Phase estimate with respect to W to precision δ
4. Measure the phase bit

3.3.2 Analysis

$|\sigma\rangle = \sum_x \sigma_x |x\rangle |x\rangle$ and $|\pi\rangle = \sum_{xy} \sqrt{\frac{w_{xy}}{W}} |x\rangle |y\rangle$ are orthogonal since the weight matrix w_{xy} contains no on diagonal elements. Therefore defining $|\rho^\perp\rangle = \frac{1}{\sqrt{2}} |\sigma\rangle - \frac{1}{\sqrt{2}} |\pi\rangle$ we can write $|\sigma\rangle = \frac{1}{\sqrt{2}} |\rho\rangle + \frac{1}{\sqrt{2}} |\rho^\perp\rangle$. From this expression we can see that the outcome of the first measurement will be $|\rho\rangle$ with probability $\frac{1}{2}$. The outcome of the second measurement will therefore be $|\pi\rangle$ with probability $\frac{1}{4}$.

The algorithm involves two phase estimations in series. If we assume the walk operator can be implemented at unit cost, the algorithm has total cost

$$\tau \sim \frac{1}{\sqrt{\delta'}} + \frac{1}{\sqrt{\delta}} \tag{25}$$

3.3.3 Applying the algorithm

Our mixing algorithm has a time complexity $\mathrm{O}\left(\frac{1}{\sqrt{\delta'}} + \frac{1}{\sqrt{\delta}}\right)$ and therefore achieves Richter's conjectured bound in any case when adding self loops to the Markov chain does not asymptotically decrease its spectral gap.

For an example of such a Markov chain, we return to our earlier example drawn in Figure 3. The corresponding weight and transition matrices are

$$w = \begin{pmatrix} 0 & 1 & a & 0 & 0 \\ 1 & 0 & a & 0 & 0 \\ a & a & b & a & a \\ 0 & 0 & a & 0 & 1 \\ 0 & 0 & a & 1 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 0 & \frac{1}{1+a} & \frac{a}{1+a} & 0 & 0 \\ \frac{1}{1+a} & 0 & \frac{a}{1+a} & 0 & 0 \\ \frac{a}{4a+b} & \frac{a}{4a+b} & \frac{b}{4a+b} & \frac{a}{4a+b} & \frac{a}{4a+b} \\ 0 & 0 & \frac{a}{1+a} & 0 & \frac{1}{1+a} \\ 0 & 0 & \frac{a}{1+a} & \frac{1}{1+a} & 0 \end{pmatrix} \tag{26}$$

Assume a is small, and we'll first take the case $b = 0$. In this case the spectral gap $\delta = \frac{a}{1+a}$, and the unnormalised stationary distribution is given by $\vec{\pi} = (1, 1, \frac{4a}{1+a}, 1, 1)$. We observe that the stationary distribution has small $\mathrm{O}(a)$ weight on the central node. We note that the classical bound for mixing time is $\tilde{\mathrm{O}}\left(\frac{1}{a}\right)$. Reverse-MNRS achieves $\mathrm{O}\left(\frac{1}{\sqrt{a}}\right)$ for starting states in any of the outer nodes, since they have $\mathrm{O}(1)$ overlap with the stationary distribution. But reverse-MNRS is no better than classical for starting in the central node, with $\mathrm{O}\left(\frac{1}{a}\right)$ complexity. In the case that we can only prepare an initial state in the central node, we'll need a better algorithm.

Consider adding a self loop to the central node by setting $b = 4 + 8a$. We again get spectral gap $\delta' = \mathrm{O}(a)$. But this time we get unnormalised stationary distribution $\vec{\rho} = (1, 1, \frac{4+12a}{1+a}, 1, 1)$. This time we have $\mathrm{O}(1)$ stationary weight on all the vertices, hence we can mix in time $\mathrm{O}\left(\frac{1}{\sqrt{a}}\right)$ for all initial states, and we have achieved Richter's conjectured bound.

But what about cases when δ' is far worse than δ? Then our algorithm will not present an advantage. Consider the $n \times n$ weight matrix $w_{xy} = 1 - \delta_{xy}$, and its counterpart modified with a single self loop. We give the example $n = 5$:

$$w = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \qquad w' = \begin{pmatrix} 20 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{27}$$

Then as $n \to \infty$, $\delta \to 1$ but $\delta' \to 0$. This case constitutes in some sense a blind application of our algorithm: we took a graph with no bottlenecks that mixes fine, and ruined its spectral gap by adding a self loop. A general means of understanding when adding self loops is expected to asymptotically change the spectral gap is given in the next section.

3.3.4 Self loops and Cheeger's inequality

The effectiveness of adding self loops can be understood in terms of Cheeger's inequality from Section 2.1.3. To recap, Cheeger's inequality bounds the spectral gap by minimising over 2-partitions a function that measures how hard it is to escape one of the subsets. The bound is $\delta = \mathrm{O}(\phi_*)$, where ϕ_* is the Cheeger constant.

The observation is that adding a self loop can only make it harder to escape a subset, so adding self loops should in general maintain or decrease the upper bound on the spectral gap. Consider the particular partition of the state space $X = S_* \cup \bar{S}_*$ such that $\phi(S_*) = \phi_*$, i.e. this is the partition that fixes the Cheeger constant. We might picture this partition graphically as two sets of nodes separated by a bottleneck, as in Figure 4.

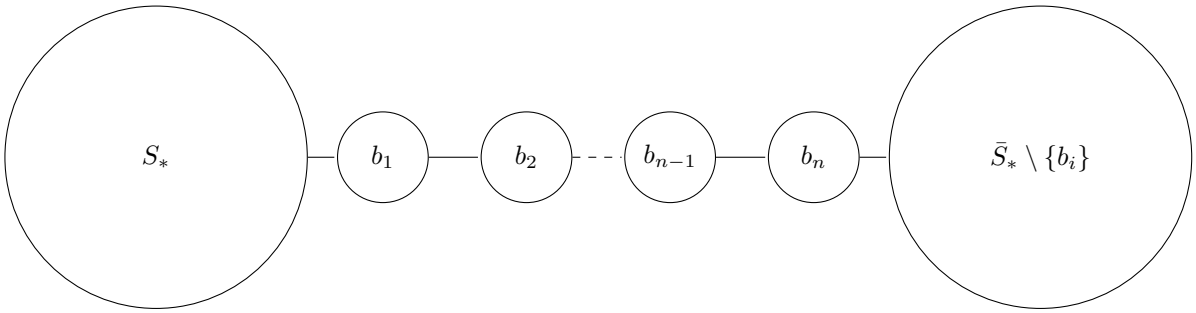


Figure 4: Graphical depiction of a bottleneck.

Adding self loops to states in S_* only makes this region harder to escape, so we'd expect this to reduce the spectral gap. Adding self loops to the bottleneck states $\{b_i\}$ or the remaining states in $\bar{S}_*$ does not have this effect, and we might expect them not to reduce the gap. This intuition can be easily applied to explain the successful case of the algorithm in Section 3.3.3. The unsuccessful case can also be explained, since by the high degree of symmetry in the problem any self loop we add will always be in S_*.

4 Discussion

Quantum mixing algorithms are used for preparing the state $|\pi\rangle = \sum_x \sqrt{\pi_x}\,|x\rangle$, where the π_x vector is the stationary distribution of some Markov chain. This is known as *q-sampling* the distribution. Whilst the q-sample might be useful in its own right, measuring the state corresponds to sampling from the classical probability distribution, so this is useful in applications such as Markov Chain Monte Carlo.

Our novel mixing algorithm mixes in time $\tau \sim \frac{1}{\sqrt{\delta'}} + \frac{1}{\sqrt{\delta}}$ (where the notion of time corresponds to number of applications of the walk operator). Here δ' is the spectral gap of the original Markov chain modified by the addition of self loops. Our algorithm is useful for any Markov chain in which $\delta' \sim \delta$, since we then achieve Richter's conjectured bound on quantum mixing times.

Our algorithm is also interesting in the context of quantum walk frameworks. Our algorithm takes our initial state $|\sigma\rangle$ and maps $|\sigma\rangle \to |\pi\rangle$. The MNRS quantum walk framework is used to implement the map $|\pi\rangle \to |\mu\rangle$ where $|\mu\rangle$ is some desired state. We can compose these two algorithms to map $|\sigma\rangle \to |\mu\rangle$, a process which clearly resembles Belovs' electric network framework. The cost of this composition goes like $\frac{1}{\sqrt{\delta'}} + \frac{1}{\sqrt{\delta}} + \frac{1}{\sqrt{\delta\langle\pi|\mu\rangle}}$. Clearly if $\delta' \sim \delta$ then this algorithm has no extra asymptotic cost over running the MNRS algorithm from $|\pi\rangle$. It is interesting that this power comes from ideas in Belovs' framework, and might indicate power present in Belovs' framework but absent from MNRS.

So our algorithm is most useful in the cases when we expect $\delta' \sim \delta$. Drawing from our discussion of Cheeger's inequality in Section 3.3.4, we suggest that this is the class of Markov chains where classical mixing is limited by some region of the state space that is hard to escape. If we can prepare an initial state outside this region, then a self loop on that initial state should not severely affect the gap, and we would expect our algorithm to mix in time $\mathrm{O}\left(\frac{1}{\sqrt{\delta}}\right)$.

It is interesting that the speed of our algorithm is affected by a non-trivial property of the chain – specifically the presence of bottlenecks. Bottlenecks are very important in the context of classical mixing, so it is surprising that as far as the author is aware they have been absent from the discussion of quantum mixing times until now.

5 Further work

Here we pose some questions for further work.

Using Cheeger's inequality and the language of bottlenecks, we provide intuition for when we might expect the spectral gap not to asymptotically change under the addition of self loops to the Markov chain. Can this be written in more formal terms?

Our algorithm is based on the walk operator of Belovs' electric network, and composing it with MNRS we are able to give a spectral quantum walk framework that starts from an arbitrary initial state. In cases when the spectral gap of the modified chain is not asymptotically worse than the original chain, this more powerful quantum walk framework has cost no worse than MNRS alone. This could support the claim that Belovs' framework is more powerful than MNRS – to what extent is this true?

6 Acknowledgements

I'm very grateful to Stacey Jeffery for her time and patience mentoring me for the 10 weeks over which this work was carried out. It was really kind of Thomas Vidick to set up this project with Dr Jeffery, not to mention the endless supply of chocolate he provided for the summer students. Thanks also to Nick, Patrick, Tim and Shannon for interesting conversations as we worked, the whole Preskill group for being so welcoming. And thanks to SFP and Caltech for organising an inspiring summer program.

References

[1] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. Comput.*, 37(1):210–239, April 2007.

[2] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM J. Comput.*, 40(1):142–164, February 2011.

[3] Kurt Bryan and Tanya Leise. The $25,000,000,000 eigenvector: The linear algebra behind google. *SIAM Rev.*, 48(3):569–581, March 2006.

[4] Gareth O. Roberts and Jeffrey S. Rosenthal. General state space markov chains and mcmc algorithms. *Probab. Surveys*, 1:20–71, 2004.

[5] Peter C. Richter. Quantum speedup of classical mixing processes. *Phys. Rev. A 76*, 2007.

[6] Aleksandrs, Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *Proceedings of the 40th International Conference on Automata, Languages, and Programming - Volume Part I*, ICALP'13, pages 105–122, Berlin, Heidelberg, 2013. Springer-Verlag.

[7] Thomas G. Wong. Grover search with lackadaisical quantum walks. *http://arxiv.org/abs/1502.04567*, 2015.

[8] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[9] Mario Szegedy. Quantum speed-up of markov chain based algorithms. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '04, pages 32–41, Washington, DC, USA, 2004. IEEE Computer Society.

[10] S. Jeffery. Frameworks for quantum algorithms. *Doctoral thesis*, 2014.

[11] Pawel Wocjan and Anura Abeyesinghe. Speed-up via quantum sampling. *Physical Review A*, 2008.

[12] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation. *SIAM J. Comput.*, 37(1):47–82, April 2007.

[13] Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik*, 1998.

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2011.