Quantum Information and Computation, Vol. 19, No. 3&4 (2019) 0181–0213

QUANTUM FAST-FORWARDING: MARKOV CHAINS AND GRAPH PROPERTY TESTING

SIMON APERS
Team SECRET, INRIA Paris, France
CWI, the Netherlands

ALAIN SARLETTE
QUANTIC lab, INRIA Paris, France
Department of Electronics and Information Systems, Ghent University, Belgium

{*simon.apers@inria.fr,alain.sarlette@inria.fr*}

Received December 16, 2018
Revised February 18, 2019

We introduce a new tool for quantum algorithms called quantum fast-forwarding (QFF). The tool uses quantum walks as a means to quadratically fast-forward a reversible Markov chain. More specifically, with P the Markov chain transition matrix and $D = \sqrt{P \circ P^T}$ its discriminant matrix ($D = P$ if P is symmetric), we construct a quantum walk algorithm that for any quantum state $|v\rangle$ and integer t returns a quantum state ϵ-close to the state $D^t |v\rangle / \| D^t |v\rangle \|$. The algorithm uses $O\left(\| D^t |v\rangle \|^{-1} \sqrt{t \log(\epsilon \| D^t |v\rangle \|)^{-1}} \right)$ expected quantum walk steps and $O(\| D^t |v\rangle \|^{-1})$ expected reflections around $|v\rangle$. This shows that quantum walks can accelerate the transient dynamics of Markov chains, complementing the line of results that proves the acceleration of their limit behavior.

We show that this tool leads to speedups on random walk algorithms in a very natural way. Specifically we consider random walk algorithms for testing the graph expansion and clusterability, and show that we can quadratically improve the dependency of the classical property testers on the random walk runtime. Moreover, our quantum algorithm exponentially improves the space complexity of the classical tester to logarithmic. As a subroutine of independent interest, we use QFF for determining whether a given pair of nodes lies in the same cluster or in separate clusters. This solves a robust version of s-t connectivity, relevant in a learning context for classifying objects among a set of examples. The different algorithms crucially rely on the quantum speedup of the transient behavior of random walks.

Keywords: quantum algorithms, quantum walks, property testing
Communicated by: R Cleve & A Harrow

1 Introduction and Summary

Quantum walks (QWs) have been shown to provide a speedup over classical Markov chains in a variety of settings. In the class of search problems, there exist quantum walk algorithms that accelerate tasks such as detecting element distinctness [1], finding triangles [2], and hitting marked elements [3, 4, 5]. In the class of sampling problems, there exist quantum

walk algorithms that speed up mixing on graphs [6, 7, 8] and simulated annealing [9, 10], and allow for quantum state generation [11]. A broader overview is given in the surveys by Ambainis [12] and Santha [13].

In this work we further develop this list by showing that quantum walks can be used in a very natural way to speed up random walk algorithms for graph property testing. Central to this result is a new tool which we call quantum walk fast-forwarding, allowing to quadratically fast-forward the full dynamics of a reversible Markov chain. Whereas most existing quantum walk algorithms build on a quadratic speedup towards the Markov chain limit behavior, quantum fast-forwarding allows to accelerate the transient dynamics as well. This feature is crucial towards speeding up the classical algorithms for property testing.

Quantum Walk Fast-Forwarding

Many of the above mentioned algorithms are to some extent preceded and inspired by the work of Watrous [14]. In this work, he introduced quantum walks as a means to *quantum simulate* random walks as a superposition on a quantum computer, without resorting to intermediate measurements. With P the transition matrix of a random walk on a regular graph, and $|v\rangle$ some arbitrary initial quantum state, he shows that it is possible to create the quantum state

$$|P^t v\rangle = P^t |v\rangle / \| P^t |v\rangle \|$$

using $O(\| P^t |v\rangle \|^{-2} t)$ expected QW steps, and $O(\| P^t |v\rangle \|^{-2})$ expected copies of $|v\rangle$. This allowed him to quantum simulate the famous random walk algorithm by Aleliunas et al [15] for undirected graph connectivity, thereby proving that the complexity class symmetric logspace is contained in a quantum analogue of randomized logspace.

In this work we show that quantum walks can create the state $|P^t v\rangle$ quadratically faster. Indeed, we show that *quantum walks can quadratically fast-forward a general reversible Markov chain.* More specifically, let P be the transition matrix of a reversible Markov chain on a finite state space with discriminant matrix $D = \sqrt{P \circ P^T}$, where the square root and "$\circ$"-product are elementwise. Note that if P is symmetric, as in the work of Watrous, then $D = P$. Following the work of Szegedy [4], generalizing the approach in [14], we can associate a quantum walk to P whose spectral properties are closely tied to those of P. These results provide the ground for most existing quantum walk algorithms, building on a quadratic speedup of the Markov chain limit behavior. For intermediate times however the behavior of these quantum walks will in general be unrelated to the Markov chain behavior. We prove that applying a technique called *linear combination of unitaries* [16, 17, 18] on the QW operator allows to mediate this shortcoming. Indeed, combining this technique with a truncated Chebyshev expansion of the Markov chain eigenvalue function allows to simulate and accelerate the (spectral) dynamics of the Markov chain. We name this scheme *quantum walk fast-forwarding* (QFF), and it condenses into the following theorem:

Theorem 1 (Quantum walk fast-forwarding with reflection) *Given any quantum state $|v\rangle$, $t \geq 0$ and $\epsilon > 0$, QFFg (Algorithm 2) outputs a quantum state ϵ-close to $|D^t v\rangle$ using*

$$O\Big(\| D^t |v\rangle \|^{-1} \sqrt{t \log(\epsilon \| D^t |v\rangle \|)^{-1}} \Big)$$

expected QW steps and $O(\| D^t |v\rangle \|^{-1})$ expected reflections around $|v\rangle$.

Much of the previous work that builds on Szegedy's quantum walk, such as [4, 9], relies on the quadratic improvement of the spectral gap when compared to the original Markov chain. This suffices when one is interested in the limit behavior of the dynamics. Our result, however, captures the transient dynamics which are governed by the complete spectrum of eigenvalues and corresponding eigenvectors. Similarly to both the preceding work and the existing classical algorithms, our algorithm makes use of only local information on the graph and Markov chain. Indeed we show that our algorithm allows quantum walks to simulate the dynamics of this entire classical spectrum, all the while retaining a quadratic acceleration.[a]

Upon completion of this work, we became aware of the recent work on quantum singular value transformation by Gilyén, Su, Low and Wiebe [20]. This work generalizes a wide range of advances in quantum algorithms for Hamiltonian simulation, Gibbs sampling and others. In Section 2.3 we discuss how our algorithm and its properties can alternatively be proved using this framework.

Quantum Graph Property Testing

We will show that QFF allows to very naturally speed up random walk algorithms for graph property testing. Given query access to a graph, property testing aims to determine whether it has a certain property, or whether it is far from having this property. Among the graphs with degree bound d, as we will be focusing on, two N-node graphs are said to be ϵ-far from each other if at least ϵdN edges have to be removed or added to turn one graph into the other. As an example, one can ask whether a given graph is bipartite, or whether it is at least ϵ-far from any bipartite graph. Testing bipartiteness is a relaxation as compared to effectively deciding whether the graph is bipartite or not (but possibly very close to bipartite), allowing for algorithms to work in sublinear time, i.e., scale as $o(N)$ with N the number of nodes in the graph. This is in contrast to the complexity of deciding properties exactly, which typically requires a number of queries at least linear in the graph size. In many realistic settings, see for instance the discussion of *massive graphs* in [21], linear in the graph size is no longer computationally feasible, hence sparking the interest in sublinear time algorithms.

We will consider property testers for the expansion and the clusterability of graphs. We start by discussing the expansion tester of Goldreich and Ron (GR) [22], and we prove how QFF allows to accelerate this tester. Specifically the problem is to determine whether the given graph has vertex expansion $\geq \Upsilon$, or whether it is ϵ-far from any graph having expansion $\geq c\Upsilon^2$ for some constant $c > 0$. The expansion of a graph forms a measure for the random walk mixing time over the graph. The idea behind the GR tester is therefore to run a number of random walks and count the number of pairwise collisions between the end points. If a random walk is congested in some low expansion set, then this number will be greater than when the random walk mixes efficiently. It thus forms a measure for the mixing behavior and expansion of the random walk. The runtime of their algorithm is

$$O(N^{1/2+\mu}\Upsilon^{-2}d^2\epsilon^{-1}\log N),$$

for any fixed $\mu > 0$, and with the $d^2\Upsilon^{-2}$-factor determined by the random walk runtime.

[a] This is reminiscent of the work by Miclo and Diaconis [19] on second order Markov chains, where they show that decreasing the probability that a Markov chain backtracks improves not only the spectral gap, but the entire spectrum. In contrast to quantum walks, however, this improvement will generally only be a constant factor, rather than quadratic.

We show that QFF very naturally allows to speed up this algorithm by fast-forwarding the random walk, and then using quantum amplitude estimation to estimate the 2-norm of the random walk probability distribution. This 2-norm will similarly be large if the random walk congests and small otherwise, thus allowing to detect whether the random walk is able to efficiently spread out or not. The runtime of our quantum algorithm is

$$O(N^{1/2+\mu}\Upsilon^{-1}d^{3/2}\epsilon^{-1}\log N),$$

which basically follows from quadratically improving the random walk runtime. In addition, our algorithm only requires polylog(N) space, as compared to the poly(N) space requirements of the GR tester. We note that in preceding work Ambainis, Childs and Liu [23] have also used quantum walks to speed up the GR tester, be it in an indirect way. Roughly they apply Ambainis' element distinctness algorithm [1] to speed up the search of collisions between random walk end points from $N^{1/2}$ to $N^{1/3}$. Compared to our result, they find a complimentary speedup to $O(N^{1/3+\mu}\Upsilon^{-2}d^2\epsilon^{-1}\log N)$. Due to the use of the element distinctness algorithm, their algorithm does require poly(N) space.

We continue by discussing the more recent line of algorithms for testing graph clusterability [24, 25], forming a natural generalization of the work of Goldreich and Ron. We discuss how these techniques make use of algorithms for classifying nodes in clusters, and show how QFF allows to accelerate these algorithms. Such node classification is of relevance beyond the setting of property testing, allowing for instance nearest-neighbor classification of nodes in a learning problem.

We remark that work by Valiant and Valiant [26] shows that estimating the distance in 2-norm between given probability distributions is much easier and more stable than estimating the distance in 1-norm, which would otherwise be the natural choice. This underlies the fact that many graph property testing algorithms estimate the 2-distance between random walk distributions. QFF allows to cast a probability distribution p as a quantum state $|p\rangle = p/\|p\|$, which is naturally associated to the 2-norm. As a consequence, QFF very naturally leads to quantum algorithms for estimating the 2-norm distance between random walk distributions, directly leading to the quantization and speedup of the above graph property testers.

Subsequent Work

In follow-up work, Ambainis, Gilyén, Jeffery and Kokainis [27] have used QFF to resolve the open problem of quantum search of multiple marked elements. Their algorithm crucially builds on the fast-forwarding of transient dynamics as is allowed by the QFF algorithm.

2 Quantum Walk Fast-Forwarding

In this section we elaborate the details of the quantum walk fast-forwarding scheme. First, we formally introduce the concept and characteristics of a quantum walk associated to a reversible Markov chain. These results provide the ground for most existing quantum walk algorithms, building on a quadratic speedup of the Markov chain limit behavior. We discuss how these results fall short for speeding up any transient behavior of the Markov chain. Second, we prove how a technique called *linear combinations of unitaries* can be used to mediate this shortcoming. By combining this technique with a truncated Chebyshev expansion of the general Markov chain eigenvalue function, we arrive at our quantum algorithm for quantum walk fast-forwarding.

2.1 Preliminaries: Quantum Walk Schemes

In this section we review the aforementioned quantum walk scheme by Watrous [14], and show how it gives rise to the subsequent work on quantum walk speedups by Ambainis [1], Szegedy [4], Magniez et al [28], and many others. Apart from a new proof of Proposition 1, the results in this section are known, and if necessary a reader could skip the section. For the rest of this paper we will only consider simple graphs $G = (\mathcal{V}, \mathcal{E})$ with node set $\mathcal{V}$ and edge set $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We will also refer to a Markov chain by its stochastic transition matrix P.

2.1.1 Watrous Scheme

Consider a Markov chain P on a graph $G = (\mathcal{V}, \mathcal{E})$, and an initial probability distribution v over $\mathcal{V}$. In early work, Watrous [14] proposed a quantum walk scheme for creating the quantum state $|P^t v\rangle$ associated to the classical distribution $P^t v$, defined by

$$\left|P^t v\right\rangle = \frac{1}{\|P^t v\|} \sum_j (P^t v)(j)\, |j\rangle\,, \tag{1}$$

where $(P^t v)(j)$ denotes the j-th component of the probability vector $P^t v$. For a general nonzero vector w, we will use the notation $|w\rangle = \frac{1}{\|w\|} \sum w(j)\, |j\rangle$ which associates a quantum state $|w\rangle$ to w. The quantum walk associated to P takes places on the extended or "coined" node space $\hat{\mathcal{V}} = \mathcal{V} \times \{\bar{0}, \mathcal{V}\} = \{(i,j) \mid i \in \mathcal{V}, j \in \{\bar{0}, \mathcal{V}\}\}$, where "$\bar{0}$" denotes some canonical initialization state. The associated Hilbert space is $\mathcal{H} = \text{span}\{|i,j\rangle \mid (i,j) \in \hat{\mathcal{V}}\}$. We will often be interested in the subspace $\mathcal{H}_{\bar{0}} = \text{span}\{|i,\bar{0}\rangle \mid i \in \mathcal{V}\}$, associated to the projector $\Pi_{\bar{0}} = I \otimes |\bar{0}\rangle\langle\bar{0}|$, with I the identity operator on the first register. The discrete-time quantum walk is described by a unitary operator U_P on $\mathcal{H}$, defined by a *shift operator* S and a *coin toss operator* V,

$$U_P = V^\dagger S V. \tag{2}$$

We will often write U instead of U_P when the context allows it. The coin toss operator is defined as $V = \sum_i |i\rangle\langle i| \otimes V_i$, where V_i is such that

$$V\,|i,\bar{0}\rangle = |i\rangle \otimes V_i\,|\bar{0}\rangle = |i\rangle \otimes |\psi_i\rangle = |i\rangle \otimes \sum_j \sqrt{P(j,i)}\,|j\rangle\,. \tag{3}$$

By the design of the QW scheme, as we will see later, it suffices to characterize the action of V_i on the state $|\bar{0}\rangle$. The operators V_i can then be arbitrarily completed into unitary matrices. The shift operator is defined by the permutation

$$|i,j\rangle \mapsto S\,|i,j\rangle = \begin{cases} |j,i\rangle\,, & (i,j) \in \mathcal{E}, \\ |i,j\rangle\,, & \text{otherwise}, \end{cases}$$

and $S\,|i,\bar{0}\rangle = S\,|i,\bar{0}\rangle$. It is now easy to prove the below lemma, stating that the restriction of U to the subspace $\mathcal{H}_{\bar{0}}$ implements the *discriminant matrix*

$$D = \sqrt{P \circ P^T},$$

with the square root and "$\circ$"-product elementwise. The discriminant matrix is closely related to the original Markov chain P, and if P is reversible then they share the same eigenvalues. We will often write $D\,|v,\bar{0}\rangle$ as shorthand for $(D \otimes I)\,|v,\bar{0}\rangle$.

Lemma 1 ([14]) *For any quantum state $|v,\bar{0}\rangle$, it holds that*

$$\Pi_{\bar{0}} U |v,\bar{0}\rangle = D |v,\bar{0}\rangle .$$

Proof. This directly follows from the fact that for any node i it holds that $U|i,\bar{0}\rangle = \sum_j \sqrt{P(i,j)P(j,i)}\,|j,\bar{0}\rangle + |\psi^{\perp}\rangle$, where $|\psi^{\perp}\rangle$ is some state perpendicular to the subspace $\mathcal{H}_{\bar{0}}$. By linearity, the proposition follows for general $|v,\bar{0}\rangle$. □

Following the terminology of more recent literature [29, 30, 20], this lemma shows that U forms a "block encoding" of the discriminant matrix D. This encoding gives rise to an easy QW algorithm for creating $|D^t v\rangle$. Namely, do t times: (i) apply a single step of the QW U, (ii) perform the measurement corresponding to the measurement operators $\Pi_{\bar{0}}$ and $I - \Pi_{\bar{0}}$. If each of the measurements returns "$\bar{0}$", which happens with a probability

$$\frac{\|D^t |v\rangle\|^2}{\|D^{t-1}|v\rangle\|^2}\frac{\|D^{t-1}|v\rangle\|^2}{\|D^{t-2}|v\rangle\|^2}\cdots\frac{\|D|v\rangle\|^2}{\||v\rangle\|^2} = \|D^t|v\rangle\|^2,$$

then the output state is $|D^t v,\bar{0}\rangle$. For symmetric P, as in Watrous' original paper, it holds that $D = P$, and so this approach effectively returns the quantum state $|P^t v\rangle$ that we were looking for. It requires t QW steps and succeeds with a probability $\|D^t|v\rangle\|^2$.

2.1.2 Quadratically Improved Spectrum

The main idea of our new QFF tool is that we can quadratically accelerate the number of QW steps required: we can create the state $|D^t v\rangle$ using $O(\sqrt{t})$ QW steps, succeeding with the same probability $\|D^t|v\rangle\|^2$. To do so we make use of work that followed up on the QW approach by Watrous, mainly initiated by Ambainis [1] and Szegedy [4] with the aim of accelerating classical search problems. We will discuss how in a certain sense this operator quadratically improves the Markov chain spectrum, yet falls short of speeding up its full dynamics. In the next section we then present our more fine-grained scheme that resolves this issue.

They proposed an alternative QW, essentially adding a reflection around the subspace $\mathcal{H}_{\bar{0}}$ to the QW operator U by Watrous:

$$W = R_{\bar{0}} U = (2\Pi_{\bar{0}} - I)U. \tag{4}$$

Their key insight is captured in the following proposition, for which we provide a new proof which explicitly builds on the insight from Watrous' work. We will denote by T_t the t-th Chebyshev polynomial of the first kind.

Proposition 1 ([4]) *For any $|v,\bar{0}\rangle$, it holds that*

$$\Pi_{\bar{0}} W^t |v,\bar{0}\rangle = T_t(D)\,|v,\bar{0}\rangle .$$

As a consequence, if $(\cos\theta, |v\rangle)$ is an eigenpair of D, then

$$\Pi_{\bar{0}} W^t |v,\bar{0}\rangle = T_t(\cos\theta)\,|v,\bar{0}\rangle = \cos(t\theta)\,|v,\bar{0}\rangle .$$

Proof. We easily find a recursion formula for $\Pi_{\bar{0}} W^t$:

$$\Pi_{\bar{0}} W^t = \Pi_{\bar{0}} R_{\bar{0}} U (2\Pi_{\bar{0}} - I) U W^{t-2} = 2\Pi_{\bar{0}} U (\Pi_{\bar{0}} W^{t-1}) - \Pi_{\bar{0}} W^{t-2}$$

using the fact that $\Pi_{\bar{0}} R_{\bar{0}} = \Pi_{\bar{0}}$, and $U^\dagger = U$ so that $U^2 = UU^\dagger = I$. Since $\Pi_{\bar{0}} W^0 = \Pi_{\bar{0}}$ and $\Pi_{\bar{0}} W = \Pi_{\bar{0}} U$, this shows that we can express $\Pi_{\bar{0}} W^t$ as a polynomial in $\Pi_{\bar{0}} U$. The Chebyshev polynomials of the first kind T_t are defined by

$$T_0(x) = 1, \quad T_1(x) = x, \quad T_t(x) = 2xT_{t-1}(x) - T_{t-2}(x).$$

Setting $x = \Pi_{\bar{0}} U$ and $T_0(\Pi_{\bar{0}} U) = \Pi_{\bar{0}}$, this shows that we can express $\Pi_{\bar{0}} W^t$ as $\Pi_{\bar{0}} W^t = T_t(\Pi_{\bar{0}} U)$. From Lemma 1 we know that $(\Pi_{\bar{0}} U)^t |v, \bar{0}\rangle = D^t |v, \bar{0}\rangle$, and therefore

$$\Pi_{\bar{0}} W^t |v, \bar{0}\rangle = T_t(D) |v, \bar{0}\rangle .$$

Using the geometric definition of T_t, $T_t(\cos\theta) = \cos(t\theta)$, we see that if $D |v, \bar{0}\rangle = \cos\theta |v, \bar{0}\rangle$ then

$$\Pi_{\bar{0}} W^t |v, \bar{0}\rangle = T_t(\cos\theta) |v, \bar{0}\rangle = \cos(t\theta) |v, \bar{0}\rangle .$$

□

This proposition constitutes the basis from which most of the aforementioned quantum algorithms start, and it will be the basis from which this work starts. The gist of the speedup comes from comparing the original action of D^t on an eigenpair $(\cos\theta, |v\rangle)$, $D^t |v, \bar{0}\rangle = \cos^t(\theta) |v, \bar{0}\rangle$, with the action of $\Pi_{\bar{0}} W^t$, $\Pi_{\bar{0}} W^t |v, \bar{0}\rangle = \cos(t\theta) |v, \bar{0}\rangle$. Taylor expanding the respective *eigenvalue functions* $g^t(\theta) = \cos^t(\theta)$ and $f_{t'}(\theta) = \cos(t'\theta)$ yields

$$g^t(\theta) = 1 - \frac{t\theta^2}{2} + O(t^2\theta^4), \quad \text{whereas} \quad f_{t'}(\theta) = 1 - \frac{t'^2\theta^2}{2} + O(t'^4\theta^4).$$

Setting $t' = \sqrt{t}$, we see that both expressions are equal up to second order in t. This suggests that the quantum walk *quadratically fast-forwards* the Markov chain, and so $\Pi_{\bar{0}} W^{\sqrt{t}} \approx \Pi_{\bar{0}} D^t$.

This observation underlies a range of quantum walk speedup results which are mainly concerned with accelerating the Markov chain asymptotics, where one is interested in the limit regime $\lim_{t\to\infty} P^t v = \pi$ and one wishes to approximate the quantum state $|\pi\rangle$. In these cases, the timescale for the classical Markov chain is for instance set by the inverse of the spectral gap $\delta^{-1} = (1 - \lambda_2)^{-1}$ (for mixing tasks and Gibbs sampling, see [9, 31]), or by the sum of the inverses $\sum(1 - \lambda_k)^{-1}$ (for hitting tasks, see [4]), where $\{\lambda_k = \cos\theta_k\}$ denotes the set of eigenvalues of P. For these purposes, the low order conclusions from the above expansion generally suffice to achieve a quantum walk speedup in generating $|\pi\rangle$.

The main issue with the above analysis is that it breaks down for t and eigenvalues θ such that $t\theta \approx 1$: $g^t(\theta)$ and $f_t(\theta)$ start to diverge from each other, thus preventing the quantum walk from simulating the full dynamics of the Markov chain. As the main contribution in the following section we will construct a more involved and fine-grained quantum walk scheme whose eigenvalue function closely approximates the Markov chain eigenvalue function $g^t(\theta)$ for all values of t and θ, without losing the quadratic fast-forward.

2.2 Quantum Fast-Forwarding Algorithm

In this section we develop our main tool: a quantum walk algorithm for quantum simulating Markov chains quadratically faster than the original dynamics. Thereto we will make use of the concept of *linear combinations of unitaries*. We will use this technique to manipulate the eigenvalues of the quantum walk such that they better approximate the Markov chain eigenvalues.

2.2.1 LCU and Chebyshev Expansion

We can create some wiggle room on the implementation of the quantum walk $\Pi_{\bar{0}}W^t$, and therefore on its eigenvalue function, by implementing linear combinations of $\Pi_{\bar{0}}W^t$ for different t. A similar approach has been used in for instance [16, 17] for Hamiltonian simulation and in [18] for optimizing quantum SDP solvers, where they call this technique *linear combination of unitaries* (LCU). We extract the below Lemma 2 from this existing work, and elaborate its details for completeness. Below the lemma we discuss how it can be used for our purpose.

The lemma shows how to implement a linear combination

$$\sum_{l=0}^{\tau} q_l \Pi_{\bar{0}} W^l,$$

where we assume that $q_l \geq 0$ and $\sum q_l = 1$. To do so, we will again enlarge the state space from $\mathcal{H}_{\hat{\mathcal{V}}}$ to $\mathcal{H}_{\hat{\mathcal{V}}\times[\tau]}$, with $[\tau] = \{0, 1, 2, \ldots, \tau\}$. We will identify $|0\rangle = |\bar{0}\rangle$ and be interested in the span of states $|j, \bar{0}, \bar{0}\rangle \equiv |j, \bar{0}\bar{0}\rangle$, $j \in \mathcal{V}$, which we denote as $\mathcal{H}_{\hat{\mathcal{V}}\times[\tau]}$. The projector $\Pi_{\bar{0}}$ will either denote the projector on the subspace $\mathcal{H}_{\hat{\mathcal{V}}}$ or $\mathcal{H}_{\hat{\mathcal{V}}\times[\tau]}$, whichever it is will be clear from the context. The construction is very similar to the Watrous quantum walk scheme. It builds on a coin toss V_q on $\mathcal{H}_{\hat{\mathcal{V}}\times[\tau]}$, defined by the coefficients q_l as

$$V_q |\psi, \bar{0}\rangle = \sum_{l=0}^{\tau} \sqrt{q_l} |\psi, l\rangle .$$

Then the *controlled W-operator* $W_{\mathrm{ctrl}} = \sum_{l=0}^{\tau} W^l \otimes |l\rangle\langle l|$ is applied which, conditioned on the integer l in the last register, applies the operator W^l:

$$W_{\mathrm{ctrl}} V_q |\psi, \bar{0}\rangle = W_{\mathrm{ctrl}} \sum_{l=0}^{\tau} \sqrt{q_l} |\psi, l\rangle = \sum_{l=0}^{\tau} \sqrt{q_l} W^l |\psi, l\rangle .$$

Finally, as in the Watrous QW, the operator $V_q^\dagger$ is applied, returning a state

$$V_q^\dagger W_{\mathrm{ctrl}} V_q |\psi, \bar{0}\rangle = \sum_{l=0}^{\tau} q_l W^l |\psi, \bar{0}\rangle + |\psi^\perp\rangle , \tag{5}$$

where $|\psi^\perp\rangle$ is some quantum state perpendicular to $\mathcal{H}_{\hat{\mathcal{V}}\times[\tau]}$. This leads to the following lemma, where we set $W_\tau = V_q^\dagger W_{\mathrm{ctrl}} V_q$.

Lemma 2 (LCU) *For any $|v, \bar{0}\bar{0}\rangle$, it holds that*

$$\Pi_{\bar{0}} W_\tau |v, \bar{0}\bar{0}\rangle = \Big(\sum_{l=0}^{\tau} q_l \Pi_{\bar{0}} W^l |v, \bar{0}\rangle \Big) \otimes |\bar{0}\rangle = \Big(\sum_{l=0}^{\tau} q_l T_l(D) |v\rangle \Big) \otimes |\bar{0}\bar{0}\rangle .$$

Implementing the operator W_τ requires $O(\tau)$ QW steps.

Proof. From (5) we see that

$$\Pi_{\bar{0}} V_q^\dagger W_{\mathrm{ctrl}} V_q |v, \bar{0}\bar{0}\rangle = \Big(\sum_{l=0}^{\tau} q_l \Pi_{\bar{0}} W^l |v, \bar{0}\rangle \Big) \otimes |\bar{0}\rangle .$$

Combined with Proposition 1, and by linearity, this proves the equality. In for instance [32, 33] it is discussed that the operator W_{ctrl} can be implemented in $O(\tau)$ QW steps, and the local coin tosses V_q and $V_q^\dagger$ require no QW steps. □

This lemma shows that if we apply the operator W_τ on $|v,\bar{0}\bar{0}\rangle$, and we perform a measurement $\{\Pi_{\bar{0}}, I-\Pi_{\bar{0}}\}$, then we retrieve the state $\left(\sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle\right)\otimes|\bar{0}\bar{0}\rangle$ (up to normalization) with a probability $\|\sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle\|^2$. The corresponding eigenvalue function is then

$$\tilde{f}_t(\theta) = \sum_{l=0}^{\tau} q_l \cos(l\theta).$$

In the following we will choose the coefficients q_l so that $\tilde{f}_{t'}(\theta)$ approximates the original eigenvalue function $g^t(\theta)=\cos^t(\theta)$ for $t'\in O(\sqrt{t})$. For this purpose we can use the Chebyshev expansion of g^t. Indeed, from for instance [34] we know that

$$x^t = \sum_{l=0}^{t} p_l T_l(x),$$

where p_l represents the probability that $|X_t|=l$ for X_t a t step random walk starting in the origin:

$$p_l = \mathbb{P}(|X_t|=l) = \begin{cases} \frac{1}{2^{t-1}}\binom{t}{\frac{t-l}{2}} & l>0,\ t=l \bmod 2,\\ \frac{1}{2^t}\binom{t}{\frac{t}{2}} & l=0,\ t=0 \bmod 2,\\ 0 & \text{elsewhere.}\end{cases} \tag{6}$$

Again using the geometric definition of the Chebyshev polynomials, $T_t(\cos(\theta))=\cos(t\theta)$, and setting $x=\cos(\theta)$, this implies that g^t can be exactly expanded into the eigenfunctions f_t:

$$g^t(\theta)=\cos^t(\theta)=\sum_{l=0}^{t} p_l\cos(l\theta)=\sum_{l=0}^{t} p_l f_t(l\theta). \tag{7}$$

Using the above lemma we can now choose $q_l=p_l$ to exactly simulate the original dynamics. The problem is that in this case $\tau=t$, and implementing W_τ therefore requires $O(t)$ QW steps, giving no speedup with respect to the simple quantum simulation scheme. We can resolve this by noting that p_l approaches a normal distribution with standard deviation $\Theta(\sqrt{t})$, so that we can approximate it exponentially well by its support on a $O(\sqrt{t})$ interval, as we elaborate in the below lemma.

Lemma 3 *Let $\epsilon>0$. If $C\geq 2\ln\frac{2}{\epsilon}$ then for all θ*

$$\left|\cos^t(\theta)-\sum_{l=0}^{\lceil\sqrt{Ct}\rceil} p_l\cos(l\theta)\right|\leq\epsilon.$$

Proof. Let $t'=\lceil\sqrt{Ct}\rceil$. The proof comes down to bounding the quantity $p_{>t'}=\sum_{l=t'+1}^{t} p_l$. Indeed, by (7) we can easily calculate that

$$\left|\cos^t(\theta)-\sum_{l=0}^{t'} p_l\cos(l\theta)\right|\leq p_{>t'},$$

so that it suffices to prove that $p_{>t'} \leq \epsilon$. We can bound $p_{>t'}$ since it represents the probability that $|X_t| > t'$ where X_t is a t step random walk X_t. By Hoeffding's inequality we know that $p_{>t'} \leq 2\exp\big(-t'^2/(2t)\big)$. For $t' = \lceil\sqrt{Ct}\rceil$ and $C \geq 2\ln\frac{2}{\epsilon}$ this shows that $p_{>\lceil\sqrt{Ct}\rceil} \leq \epsilon$, which proves the lemma. □

This lemma shows that it is possible to *pointwise* approximate the original eigenvalue function $\cos^t(\theta)$, up to an error ϵ, using the truncated Chebyshev expansion

$$g_\tau^t(\theta) = \sum_{l=0}^{\tau} p_l \cos(l\theta)$$

for $\tau \in O(\sqrt{t}\log\frac{1}{\epsilon})$. We note that a similar approximation in combination with LCU was used for a different purpose in [35]. In the next section we combine this approximation lemma with the LCU lemma, leading to our quantum fast-forwarding scheme.

2.2.2 Quantum Fast-Forwarding Algorithm

Combining the above lemmas, we can propose our QFF algorithm. It builds on the operator W_τ from Lemma 2, with coefficients q_l derived from Lemma 3, so that

$$\begin{aligned} \Pi_{\bar{0}} W_\tau \left|v,\bar{0}\bar{0}\right\rangle &= \frac{1}{1-p_{>\tau}} \sum_{l=0}^{\tau} p_l \Pi_{\bar{0}} W^l \left|v,\bar{0}\bar{0}\right\rangle \\ &= \left(\frac{1}{1-p_{>\tau}} \sum_{l=0}^{\tau} p_l T_l(D) \left|v\right\rangle\right) \otimes \left|\bar{0}\bar{0}\right\rangle, \end{aligned} \tag{8}$$

where the p_l are defined in (6).

Algorithm 1 Quantum Fast-Forwarding $\mathbf{QFF}(\left|v\right\rangle, P, t, \epsilon)$

Input: quantum state $\left|v\right\rangle \in \mathcal{H}_\mathcal{V}$, reversible Markov chain P, $t \in \mathbb{N}$, $\epsilon > 0$
Do:

1: set $\epsilon' = \|D^t \left|v\right\rangle\|\epsilon/2$ and $\tau = \left\lceil\sqrt{2t\ln(2/\epsilon')}\right\rceil$
2: initialize the registers $\mathbf{R_1R_2R_3}$ with the state $\left|v,\bar{0}\bar{0}\right\rangle$
3: apply the LCU operator W_τ as in (8) on $\mathbf{R_1R_2R_3}$
4: perform the measurement $\{\Pi_{\bar{0}\bar{0}}, I - \Pi_{\bar{0}\bar{0}}\}$
5: **if** outcome $\neq$ "$\bar{0}\bar{0}$" **then** output "`Fail`" and stop

Output: registers $\mathbf{R_1R_2R_3}$

Theorem 2 (Quantum Fast-Forwarding) *The QFF algorithm* $\mathbf{QFF}(\left|v\right\rangle, P, t, \epsilon)$ *outputs a state* ϵ*-close to* $\left|D^t v,\bar{0}\bar{0}\right\rangle$ *with success probability at least* $(1-\epsilon)\|D^t \left|v\right\rangle\|^2$*. Otherwise it outputs "`Fail`". The algorithm uses a number of QW steps*

$$\tau \in O\left(\sqrt{t}\,\log^{1/2}\frac{1}{\epsilon\|D^t \left|v\right\rangle\|}\right).$$

Proof. Let $\{(\cos\theta_k, \left|v_k\right\rangle), 1 \leq k \leq |\mathcal{V}|\}$ be a complete orthonormal set of eigenpairs of D. Then we can write $\left|v\right\rangle = \sum_k \alpha_k \left|v_k\right\rangle$ and the goal state $\left|D^t v\right\rangle = \sum_k \alpha_k \cos(\theta_k)^t \left|v_k\right\rangle / \|D^t \left|v\right\rangle\|$.

From Lemma 2 we know that if we apply the operator W_τ on $|v,\bar{0}\bar{0}\rangle$, and we perform a measurement $\{\Pi_{\bar{0}}, I-\Pi_{\bar{0}}\}$, then we retrieve the state

$$\frac{1}{\left\|\frac{1}{1-p_{>\tau}}\sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle\,\right\|}\left(\frac{1}{1-p_{>\tau}}\sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle\right)\otimes|\bar{0}\bar{0}\rangle$$
$$=\frac{1}{\left\|\sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle\,\right\|}\left(\sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle\right)\otimes|\bar{0}\bar{0}\rangle$$

with a success probability $\|\frac{1}{1-p_{>\tau}}\sum_{l=0}^{\tau} p_l T_l(D)\,|v\rangle\,\|^2$. We will denote the state $|\psi_\tau\rangle = \sum_{l=0}^{\tau} q_l T_l(D)\,|v\rangle$. By the approximation from Lemma 3 we know that if $\tau = \left\lceil\sqrt{2t\ln\frac{2}{\epsilon'}}\right\rceil$ then $|\psi_\tau\rangle$ will be ϵ'-close to $D^t\,|v\rangle$:

$$\left\||\psi_\tau\rangle - D^t\,|v\rangle\right\| = \sqrt{\sum_k\left|\sum_{l=0}^{\tau} q_l\cos(l\theta)-\cos^t(\theta_k)\right|^2\cdot|\alpha_k|^2}$$
$$\leq\sqrt{\epsilon'^2\sum_k|\alpha_k|^2}=\epsilon'.$$

Using standard manipulations we know that for any two nonzero vectors it holds that $\|v/\|v\| - w/\|w\|\| \leq 2\|v-w\|/\|w\|$. As a consequence we can bound

$$\left\|\frac{|\psi\rangle}{\|\,|\psi\rangle\,\|}-\frac{D^t\,|v\rangle}{\|D^t\,|v\rangle\,\|}\right\|\leq\frac{2\epsilon'}{\|D^t\,|v\rangle\,\|}=\epsilon,$$

using the fact that we chose $\epsilon' = \|D^t\,|v\rangle\,\|\epsilon/2$. We can now also bound the success probability using the reverse triangle inequality:

$$\left\|\frac{1}{1-p_{>\tau}}\sum_{l=0}^{\tau} p_l T_l(D)\,|v\rangle\right\|^2\geq\left\|\sum_{l=0}^{\tau} p_l T_l(D)\,|v\rangle\right\|^2$$
$$\geq(\|D^t\,|v\rangle\,\|-\epsilon')^2\geq(1-\epsilon)\|D^t\,|v\rangle\,\|^2.$$

By Lemma 2 we know that implementing the operator W_τ requires a number of QW steps

$$\tau=\left\lceil\sqrt{2t}\ln^{1/2}\frac{4}{\epsilon\|D^t\,|v\rangle\,\|}\right\rceil\in O\left(\sqrt{t}\,\log^{1/2}\frac{1}{\epsilon\|D^t\,|v\rangle\,\|}\right).$$

This finalizes the proof. □

This theorem establishes our algorithm for quantum fast-forwarding Markov chains. It winds back the quantum walk speedup of the Ambainis-Szegedy scheme on the Markov chain asymptotic behavior to the original problem of quantum simulating Markov chains for any fixed timestep, showing that we can achieve the same quadratic acceleration that is characteristic for this scheme. The success probability is proportional to $\|D^t\,|v\rangle\,\|^2$, so that $\|D^t\,|v\rangle\,\|^{-2}$ expected iterations are necessary for the scheme to succeed. In the next section we show how to quadratically improve this. We mention that the norm $\|D^t\,|v\rangle\,\|$ will be small when the

Markov chain spreads out from a small set to a large set, e.g., going from a single node to the uniform distribution yields $\|D^t |v\rangle\| = N^{-1/2}$. This reflects the fact that it is costly for the quantum algorithm to create a superposition over a large number of queried elements (see [36] for a discussion and a more rigorous analysis of this phenomenon).

2.2.3 Amplitude Amplification

We can improve the success probability to a constant by replacing the final measurement in the algorithm with amplitude amplification, provided that we can reflect around the initial state $|v,\bar{0}\bar{0}\rangle$ ($\equiv |v,\bar{0}\rangle$, we will use the shorthand $\bar{0}$ to denote $\bar{0}\bar{0}$), i.e., implement the reflection operator

$$R_v = 2|v,\bar{0}\rangle\langle v,\bar{0}| - I.$$

Thereto we will use the following proposition by Brassard et al [37], demonstrating how we can retrieve a component $\Pi_{\bar{0}}|\psi\rangle$ of some quantum state $|\psi\rangle$ by performing reflections around $|\psi\rangle$ and around the image of $\Pi_{\bar{0}}$.

Proposition 2 (Amplitude amplification [37]) *Given an initial state $|\psi\rangle$ and a projection operator $\Pi_{\bar{0}}$, with $\Pi_{\bar{0}}|\psi\rangle \neq 0$. Define the reflections $R_\psi = 2|\psi\rangle\langle\psi| - I$ and $R_{\bar{0}} = 2\Pi_{\bar{0}} - I$, and set $m = \lfloor \pi/(4\theta) \rfloor$ with $\theta \in (0, \pi/2]$ such that $\sin\theta = \|\Pi_{\bar{0}}|\psi\rangle\|$. If we apply the operator $(-R_\psi R_{\bar{0}})$ m times on the state $|\psi\rangle$, and we perform a measurement $\{\Pi_{\bar{0}}, I - \Pi_{\bar{0}}\}$, then we find back the state $\Pi_{\bar{0}}|\psi\rangle / \|\Pi_{\bar{0}}|\psi\rangle\|$ with probability at least $\max\{1 - \sin^2\theta, \sin^2\theta\} \geq 1/2$.*

This implementation of amplitude amplification requires a very good estimate of the initial success probability $\|\Pi_{\bar{0}}|\psi\rangle\|^2$ to determine m. If m is chosen either too small or too large, then the guarantees on the success probability are lost, a problem often referred to as the "soufflé problem". A remedy is however proposed in [37], in which iteratively different guesses for m are used. They show that this approach also yields a success probability $\geq 1/2$, while still applying the operator $(-R_\psi R_{\bar{0}})$ only $O(1/\|\Pi_{\bar{0}}|\psi\rangle\|)$ times. For clarity we will implicitly rely on this fact, and throughout assume that we can appropriately determine the parameter m. As a sidenote, we mention that the "fixed point amplitude amplification" algorithm from [38] (and used in Appendix A) allows to perform amplitude amplification when only a lower bound on $\alpha \leq \|\Pi_{\bar{0}}|\psi\rangle\|$ is known. It allows to retrieve the goal state using $O(1/\alpha)$ implementations of $R_{\bar{0}}$ and R_ψ. For our purpose it will always hold that $\|\Pi_{\bar{0}}|\psi\rangle\| \geq 1/\sqrt{N}$, so it would be possible to use this lower bound. In the below Theorem 3 however this would yield a number of QW steps in $\tilde{O}(\sqrt{tN})$, which could be much worse than $\tilde{O}(\sqrt{t}/\|D^t|v\rangle\|)$.

In our QFF algorithm we have $|\psi\rangle = W_\tau|v,\bar{0}\rangle$, and we wish to retrieve the component $\Pi_{\bar{0}}|\psi\rangle = \Pi_{\bar{0}}W_\tau|v,\bar{0}\rangle$. Amplitude amplification shows that we can do so with constant success probability by implementing the operator $(-R_\psi R_{\bar{0}})$ for a number of times in $\Theta(1/\|\Pi_{\bar{0}}|\psi\rangle\|)$. The reflection $R_{\bar{0}} = 2\Pi_{\bar{0}} - I$ is considered an elementary operation on the basis states, which we can implement with a negligible cost. The following lemma shows that we can implement the reflection $R_\psi = 2|\psi\rangle\langle\psi| - I$ using $O(\tau)$ QW steps.

Lemma 4 *The operator R_ψ can be implemented using $O(\tau)$ QW steps and a reflection R_v around the initial state $|v,\bar{0}\rangle$.*

Proof. We can rewrite the reflection $R_\psi = 2|\psi\rangle\langle\psi| - I = W_\tau R_v W_\tau^\dagger$, so that we can implement the reflection by implementing the operators W_τ, $W_\tau^\dagger$, and R_v. To implement the operator $W_\tau^\dagger$, we recall that $W_\tau = V_q^\dagger\Big[\sum_{l=0}^{\tau}|l\rangle\langle l| \otimes (R_{\bar{0}}U)^l\Big]V_q$ and so $W_\tau^\dagger = V_q^\dagger\Big[\sum_{l=0}^{\tau}|l\rangle\langle l| \otimes (UR_{\bar{0}})^l\Big]V_q$.

Here we also used the fact that $U = V^\dagger S V$ with $S = S^\dagger$, as in (2), so that $U^\dagger = U$. We already discussed in Lemma 2 that a controlled operator $\sum_{l=0}^{\tau} |l\rangle\langle l| \otimes U R_{\bar{0}}$ can be implemented in $O(\tau)$ QW steps, so that both W_τ and $W_\tau^\dagger$ can be implemented in $O(\tau)$ QW steps. □

It follows that the total operator $(-R_\psi R_{\bar{0}})$ can be implemented using $O(\tau)$ QW steps, a reflection around the initial state $|v,\bar{0}\rangle$, and a number of elementary operations. In many cases the initial state will be an elementary basis state, so that the reflection R_v will also be elementary, and the main cost boils down to $O(\tau)$ QW steps. We can now propose the improved QFF algorithm, **QFFg**, in Algorithm 2. Theorem 3 proves its correctness and complexity. We note that this describes the Monte Carlo version of QFF. We can easily retrieve the Las Vegas version, as stated in Theorem 1 in the introduction, by repeating the below algorithm until it is successful. As mentioned at the end in previous section, we improve the expected number of QW steps for the QFF algorithm to succeed from $\widetilde{\Theta}(\|D^t|v\rangle\|^{-2}\sqrt{t})$ to $\widetilde{\Theta}(\|D^t|v\rangle\|^{-1}\sqrt{t})$.

Algorithm 2 Quantum Fast-Forward with Reflections $\mathbf{QFFg}(|v\rangle, P, t, \epsilon)$

Input: quantum state $|v\rangle \in \mathcal{H}_\mathcal{V}$, Markov chain P, $t \in \mathbb{N}$, $\epsilon > 0$
Do:
1: set $\epsilon' = \|D^t|v\rangle\|\epsilon/2$ and $\tau = \left\lceil \sqrt{2t\ln(2/\epsilon')} \right\rceil$
2: set $m = \lfloor \pi/(4\theta) \rfloor$, where $0 < \theta \leq \pi/2$ s.t. $\sin\theta = \|\Pi_{\bar{0}} W_\tau |v\rangle\|$
3: initialize registers $\mathbf{R_1 R_2 R_3}$ with the state $|v,\bar{0}\bar{0}\rangle$
4: apply the LCU operator W_τ on $\mathbf{R_1 R_2 R_3}$
5: apply the operator $(-R_\psi R_{\bar{0}})^m = (-W_\tau R_v W_\tau^\dagger R_{\bar{0}})^m$ ▷ *Amplitude Amplification*
6: perform the measurement $\{\Pi_{\bar{0}\bar{0}}, I - \Pi_{\bar{0}\bar{0}}\}$
7: **if** outcome $\neq$ "$\bar{0}\bar{0}$" **then** output "`Fail`" and stop
Output: registers $\mathbf{R_1 R_2 R_3}$

Theorem 3 *The QFFg algorithm* $\mathbf{QFFg}(|v\rangle, P, t, \epsilon)$ *outputs a state* ϵ*-close to* $|D^t v\rangle$ *with success probability at least* $1/2$*. Otherwise, it outputs "*`Fail`*". The algorithm uses* $\Theta(1/\|D^t|v\rangle\|)$ *reflections around the initial state, and a number of QW steps*

$$O(m\tau) \in O\left(\frac{\sqrt{t}}{\|D^t|v\rangle\|} \log^{1/2} \frac{1}{\epsilon \|D^t|v\rangle\|} \right).$$

Proof. The algorithm straightforwardly applies the amplitude amplification scheme on the state $W_\tau|v,\bar{0}\rangle$. From Proposition 2 we know that the scheme has a success probability $\geq \max\{1-\sin^2\theta, \sin^2\theta\} \geq 1/2$. The number of QW steps for implementing W_τ and $(-R_\psi R_{\bar{0}})^m$ is $O(\tau)$ respectively $O(m\tau)$. We know that $m \in O(1/\|\Pi_{\bar{0}} W_\tau |v,\bar{0}\rangle\|)$, and from the proof of Theorem 2 we can bound $\|\Pi_{\bar{0}} W_\tau |v,\bar{0}\rangle\| \geq \|D^t|v\rangle\| - \epsilon' = (1-\epsilon/2)\|D^t|v\rangle\| \in \Theta(\|D^t|v\rangle\|)$ for all $\epsilon < 1/2$. □

2.3 Quantum Singular Value Transformation

After completion of this work, we were pointed to the recent work of Chakraborty et al [30] on block-encoding, and Gilyén et al [20] on quantum singular value transformation. These develop a framework that generalizes and unifies the principles underlying a large number

of quantum algorithms for problems such as Hamiltonian simulation, Gibbs sampling, and many more. In the following we note that an alternative derivation of our QFF algorithm and its properties can be constructed from this framework. Specifically, [20] consider a general projected unitary encoding of an operator $A = \Pi U \Pi'$, where Π, Π' are projectors and U is a unitary operator. We can see the quantum walk encoding $D = \Pi_{\bar{0}} U \Pi_{\bar{0}}$ of the discriminant matrix of a Markov chain, as in Lemma 1, as a special case of such encoding. If A has a singular value decomposition $A = W \Sigma V^\dagger$, then they show that it is possible to transform the singular values of A. More precisely, given some degree-d polynomial p, they construct a quantum circuit that implements the operator $Wp(\Sigma)V^\dagger$ using the operators U and $U^\dagger$ at most d times. They then cite a result from Sachdeva and Vishnoi [39] showing that if $p(\Sigma) = \Sigma^d$, then this polynomial can be efficiently approximated using a polynomial with degree $O(\sqrt{d})$ (this result also follows from our Chebyshev truncation in Lemma 3). By applying their quantum singular value transformation on this approximating polynomial, we can retrieve our QFF algorithm.

3 Quantum Property Testing

In this section we show how QFF allows to speed up random walk algorithms for property testing on graphs. Specifically, we will consider property testers for the expansion and the clusterability of graphs. We leave it as an open question whether similar speedups can be found for other graph property testers, an interesting example of which is the recent random walk algorithm by [40] for testing the occurrence of forbidden minors. In the first section we will discuss the expansion tester of Goldreich and Ron (GR), which they presented in later work [22], and we prove how QFF allows to accelerate this tester very naturally. We compare this speedup to the preceding work by Ambainis, Childs and Liu [23], and discuss how they achieve a complementary speedup to ours. Then we discuss the more recent line of algorithms for testing graph clusterability, aimed at probing for instance the community structure of a given graph. We discuss how these testers make use of algorithms for classifying nodes in clusters, and show how QFF allows to accelerate these algorithms. Such node classification is of relevance beyond the setting of property testing, allowing for instance nearest-neighbor classification of nodes in a learning problem.

3.1 Classical Expansion Tester

To formalize the concept of a graph property tester, we must introduce the notion of *oracle or query access* to a graph as used throughout the literature on property testing. Query access to an N-node undirected graph with degree bound d means that we can query the graph with a string $(v, i) \in [N] \times [d]$, upon which we receive either the i-th neighbor $w \in [N]$ of v, or a special symbol in case v has less than i neighbors. Clearly this model allows to perform a random walk over the graph. In addition it allows to generate a uniformly random node by simply generating a random number in $[N]$. This differs from the more classical Markov chain setting where possibly we are only given a single node, and we must explore the graph in a completely local manner.

Given such query access to a graph, the task of a property tester is to determine whether the graph has a certain property or is far from any graph having that property. To formalize this, a distance measure between two N-node graphs G and G' is defined, equaling the number

of edges that have to be added or removed from G to transform it into G'. With $\mathcal{E}$ and $\mathcal{E}'$ the edge sets of G resp. G', this equals the size of the symmetric difference $|\mathcal{E} \triangle \mathcal{E}'|$. We say that two N-node graphs G and G' with degree bound d are ϵ-far from each other if $|\mathcal{E} \triangle \mathcal{E}'| \geq \epsilon N d$. Given a graph and a parameter ϵ, a property tester should correctly distinguish between the graph having a certain property, and the graph being at least ϵ-far from any graph with that property (i.e., the distance between the graph and any graph with that property is $\geq \epsilon N d$). When given a graph that is neither, the algorithm can do whatever.

Goldreich and Ron [22] studied a property tester for the expansion of a graph.[b] The expansion or vertex expansion of a graph $G = (\mathcal{V}, \mathcal{E})$ is defined by

$$\min_{|\mathcal{S}| \leq |\mathcal{V}|/2} \frac{|\partial \mathcal{S}^c|}{|\mathcal{S}|},$$

where $\partial \mathcal{S}^c$ denotes the outer boundary of $\mathcal{S}$, i.e., the set of nodes in $\mathcal{S}^c = \mathcal{V} \backslash \mathcal{S}$ that have an edge going to $\mathcal{S}$. For some given parameter Υ, an expansion tester should determine whether a given graph has expansion $\geq \Upsilon$, or whether it is at least ϵ-far from any such graph. GR, and the subsequent literature [41, 42, 43, 23], have relaxed this setting somewhat. They propose the following definition:

Definition 1 *An algorithm is an $(\Upsilon, \epsilon, \mu)$-expansion tester if there exists a constant $c > 0$, possibly dependent on d, such that given parameters N, d, and query access to an N-node graph with degree bound d it holds that*

- *if the graph has expansion $\geq \Upsilon$, then the algorithm outputs "`accept`" with probability at least $2/3$,*
- *if the graph is ϵ-far from any graph having expansion $\geq c\mu\Upsilon^2$, then the algorithm outputs "`reject`" with probability at least $2/3$.*

In the strict setting of property testing, the expression "$\geq c\mu\Upsilon^2$" in the second bullet should be replaced by "$\geq \Upsilon$". Although unproven, the relaxation in this definition seems necessary to allow for efficient (sublinear) testing. GR [44] conjectured that the below Algorithm 3 is a $(\Upsilon, \epsilon, \mu)$ expansion tester. They also proved that any classical expansion tester must make at least $\Omega(\sqrt{N})$ queries to the graph.

The intuition behind the algorithm is very clear. It builds on the use of a random walk P on the given graph, which starting from a node v jumps to any of its d_v neighbors with a probability $1/(2d)$, and stays put otherwise:

$$P(u,v) = \begin{cases} 1/(2d) & (v,u) \in E \\ 1 - d_v/(2d) & u = v \\ 0 & \text{elsewhere.} \end{cases} \tag{9}$$

This walk is lazy and symmetric, and hence converges to the uniform distribution. If the graph has vertex expansion Υ, then one can prove that the mixing time of this random walk is $O(d^2 \Upsilon^{-2} \log N)$. As a consequence, the probability distribution of the random walks in

[b] They actually studied a property tester for the spectral gap of a graph, for which currently there is no known sublinear algorithm. All follow-up work however, as well as our work, considers the closely-related graph expansion.

Algorithm 3 Graph Expansion Tester

Input: parameters N and d; query access to N-node graph G with degree bound d; expansion parameter Υ; accuracy parameter ϵ; running time parameter $\mu < 1/4$

Do:

1: **for** $T \in \Theta(\epsilon^{-1})$ times **do**
2: select a uniformly random starting vertex s
3: perform $m \in \Theta(N^{1/2+\mu})$ independent random walks of length $t \in \Theta(d^2\Upsilon^{-2}\log N)$, starting in s
4: count number of pairwise collisions between the endpoints of the m random walks
5: if the count is greater than $M \in \Theta(N^{2\mu})$, abort and output "`reject`"

Output: if no "`reject`", output "`accept`"

step 2 of the algorithm must be close to uniform. If p describes the probability distribution of the endpoint of a random walk starting from some fixed node, then the probability of a pairwise collision between two independent endpoints, i.e., the probability that the random walks end in the same node, is given by

$$\sum p(j)^2 = \|p\|^2.$$

This will be close to $1/N$ if p is close to uniform. If on the other hand the expansion of the graph is $\ll \Upsilon$, then random walks can get stuck in a small region, leading to an increase in the 2-norm or collision probability. It follows that the collision probability of a random walk forms a measure for the expansion of the graph. The key insight is then that, by a refinement of the birthday paradox, $\Theta(N^{1/2+\mu})$ independent samples of the same random walk suffice to estimate the collision probability to within a multiplicative factor $1 + O(N^{-2\mu})$. As a consequence, it is possible to estimate the 2-norm of a t-step random walk on an N-node graph using $\Theta(N^{1/2+\mu}t\epsilon^{-1})$ random walk steps.

The conjecture that Algorithm 3 is an expansion tester was later resolved as the conclusion of a series of papers by Czumaj and Sohler [41], Kale and Seshadhri [42] and Nachmias and Shapira [43], leading to the following theorem.

Theorem 4 ([43]) *If $d \geq 3$, then Algorithm 3 is a $(\Upsilon, \epsilon, \mu)$ expansion tester with runtime*

$$O(N^{1/2+\mu}\Upsilon^{-2}d^2\epsilon^{-1}\log N).$$

In the following section we show that we can use QFF to accelerate this tester very naturally by quantum simulating the random walks, and using quantum techniques to estimate the 2-norm.

3.2 Quantum Expansion Tester

It is possible to extend the classical query model to the quantum setting, a proper definition of which can be found in [45, 23]. For this work it suffices to know that (i) we can generate a uniformly random node as in the classical case, and (ii) we can implement a single step of the quantum walk operator as defined in (4) using $O(\sqrt{d})$ queries to the graph, where d is the maximum degree.

To accelerate the classical tester we will quantum simulate the random walks, and then perform quantum amplitude estimation to estimate the 2-norm of the simulated random walks. Together with the aforementioned amplitude amplification scheme, the amplitude estimation scheme is described in the work by Brassard et al [37]. It is captured by the following lemma. We note that in the original statement in [37] the number of reflections scales as $1/\delta$ for a success probability $1-\delta$. We use standard tricks to improve this to $\log(1/\delta)$.

Lemma 5 (Quantum Amplitude Estimation) *Consider a quantum state $|\psi\rangle$ and a general projector $\Pi_{\bar{0}}$. Give some $\delta > 0$, there exists a quantum algorithm that outputs an estimate a such that $\big|\|\Pi_{\bar{0}}|\psi\rangle\| - a\big| \leq \epsilon$ with probability at least $1-\delta$, using $O(\log(1/\delta)\epsilon^{-1})$ reflections around $|\psi\rangle$ and around the image of $\Pi_{\bar{0}}$.*

Proof. We can use the quantum amplitude estimation algorithm from [37, Theorem 12] to output an $\epsilon' = \epsilon/3$-close estimate with success probability at least $5/6$. This algorithm requires $O(1/\epsilon)$ reflections around $|\psi\rangle$ and around the image of $\Pi_{\bar{0}}$. We can boost the success probability to $1-\delta$ by running their algorithm $T = \lceil 18 \ln \delta^{-1} \rceil$ times, which by Hoeffding's inequality implies that, with a probability at least $1-\delta$, at least $2T/3$ iterations have been successful. Therefore, with probability $1-\delta$, it holds that (i) at least $2T/3$ estimates lie in the interval $[\|\Pi_{\bar{0}}|\psi\rangle\| - \epsilon', \|\Pi_{\bar{0}}|\psi\rangle\| + \epsilon']$, and therefore (ii) we can find a subset $\mathcal{S}$ of estimates, $|\mathcal{S}| \geq 2T/3$, all of which lie in a $2\epsilon'$-interval. This subset must overlap with the interval $[\|\Pi_{\bar{0}}|\psi\rangle\| - \epsilon', \|\Pi_{\bar{0}}|\psi\rangle\| + \epsilon']$.

If now we output any element of this subset $\mathcal{S}$, we know that it lies in the interval $[\|\Pi_{\bar{0}}|\psi\rangle\| - 3\epsilon', \|\Pi_{\bar{0}}|\psi\rangle\| + 3\epsilon']$. This proves that with probability $1-\delta$ we can output an estimate of $\|\Pi_{\bar{0}}|\psi\rangle\|$ with precision $3\epsilon' = \epsilon$, using T runs of the quantum amplitude estimation algorithm in [37], each of which requires $O(1/\epsilon)$ reflections around $|\psi\rangle$ and around the image of $\Pi_{\bar{0}}$. This proves the claimed statement. □

We will use this amplitude estimation algorithm to estimate the 2-norm of a random walk. Thereto we recall the QFF scheme as discussed in Section 2.2.2. Note that the random walk (9) proposed in the GR tester is symmetric, so that we can simply replace the discriminant matrix D in the QFF algorithm by the random walk matrix P. Given a quantum state $|s,\bar{0}\bar{0}\rangle$, QFF applies an operator W_τ so that

$$\Pi_{\bar{0}} W_\tau |s,\bar{0}\bar{0}\rangle = \left(\frac{1}{1-p_{>\tau}} \sum_{l=0}^{\tau} p_l T_l(P)|s\rangle\right) \otimes |\bar{0}\bar{0}\rangle \approx (P^t|s\rangle) \otimes |\bar{0}\bar{0}\rangle\,,$$

as in (8), with the summation corresponding to the truncated Chebyshev expansion of P^t. Implementing this operator requires $O(\tau)$ QW steps and $O(\tau\sqrt{d})$ queries to the graph. If we set $\tau \in \Theta\big(\sqrt{t\log(N\epsilon^{-1})}\big)$ (replacing $\|P^t|s\rangle\|$ by its lower bound $N^{-1/2}$ in Algorithm 2) then the 2-norm of $\frac{1}{1-p_{>\tau}}\sum_{l=0}^{\tau} p_l T_l(P)|v\rangle$ approximates the 2-norm of $P^t|v\rangle$ to precision $O(\epsilon)$. Applying quantum amplitude estimation on the state $W_\tau|v,\bar{0}\bar{0}\rangle$ and projector $\Pi_{\bar{0}}$ will therefore allow to estimate the 2-norm of $P^t|v\rangle$, as was our initial goal. This scheme is easily formalized in the below algorithm and theorem.

Algorithm 4 Quantum 2-norm Estimator

Input: parameters N and d; query access to N-node graph G with degree bound d; starting vertex s; running time t; accuracy parameter ϵ; confidence parameter δ

Do:

1: set $\tau \in O(\sqrt{t \log(N/\epsilon)})$
2: apply the QFF operator W_τ on the quantum state $|s, \bar{0}\bar{0}\rangle$
3: use quantum amplitude estimation to create estimate a of $\|\Pi_{\bar{0}} W_\tau |s, \bar{0}\bar{0}\rangle\|$ to error $\epsilon/2$ with probability $1-\delta$

Output: estimate a

Theorem 5 (Quantum 2-norm Estimator) *With probability at least $1-\delta$, Algorithm 4 outputs an estimate a such that $\big|\|P^t |s\rangle\| - a\big| \leq \epsilon$. The algorithm requires a number of QW steps bounded by $O\big(\frac{\sqrt{t}}{\epsilon} \log \frac{1}{\delta} \log^{1/2} \frac{N}{\epsilon}\big)$.*

Proof. We will prove the theorem for the algorithm parameter $\tau = \left\lceil \sqrt{2t \ln\left(8\sqrt{N}/\epsilon\right)} \right\rceil$. By this choice, and the fact that $\|P^t |s\rangle\| \geq N^{-1/2}$ on any N-node graph, we can deduce from the proof of Theorem 2 that

$$\big|\|\Pi_{\bar{0}} W_\tau |s\rangle\| - \|P^t |s\rangle\|\big| \leq \epsilon/2.$$

Applying quantum amplitude estimation on $\Pi_{\bar{0}} W_\tau |s\rangle$ with a precision $\epsilon/2$ therefore leads to an estimate of $\|P^t |s\rangle\|$ up to error ϵ. By Lemma 5 we can do so with a probability $1-\delta$ using $O(\epsilon^{-1} \log \delta^{-1})$ reflections around $W_\tau |s\rangle$. We can implement a single such reflection using 2τ QW steps, and a reflection around the initial state (which is a basis state and can hence be neglected). □

We can compare this with the classical 2-norm tester proposed by Czumaj, Peng and Sohler [24, Lemma 3.2], building on the GR tester. For $\delta = 1/3$ their tester requires $O(t/\epsilon)$ queries to the graph, whereas our algorithm only requires $\widetilde{O}(\sqrt{t}/\epsilon)$ queries. We can now use our quantum 2-norm tester to create a quantum tester for the graph expansion. The proof makes use of some details from the classical proof of Nachmias and Shapira [43].

Algorithm 5 Quantum Graph Expansion Tester

Input: parameters N and d; query access to N-node graph G with degree bound d; expansion parameter Υ; accuracy parameter ϵ; running time parameter $\mu < 1/4$

Do:

1: set $t \in O(d^2 \Upsilon^{-2} \log N)$, $M \in O(N^{-1/2})$, $\epsilon' \in O(N^{-1/2+\mu})$, $\delta \in O(\epsilon)$
2: **for** $T \in O(\epsilon^{-1})$ times **do**
3: select a uniformly random starting node s
4: use Algorithm 4 to create estimate a of $\|P^t |s\rangle\|$ to precision ϵ', with probability $1-\delta$
5: if $a > M + \epsilon'$, abort and output "`reject`"

Output: if no "`reject`", output "`accept`"

Theorem 6 (Quantum Graph Expansion Tester) *If $d \geq 3$ then Algorithm 5 is a $(\Upsilon, \epsilon, \mu)$ expansion tester. The runtime and number of queries of the algorithm are bounded by*

$$O(N^{1/2+\mu} d^{3/2} \Upsilon^{-1} \epsilon^{-1} \log(\epsilon^{-1}) \log N).$$

Proof. We will prove the theorem for the algorithm parameters $t = 16 d^2 \Upsilon^{-2} \log N$, $M = \sqrt{N^{-1}(1+N^{-1})}$, $\epsilon' = N^{-1/2+\mu}/(16\sqrt{2})$, $\delta = \epsilon/300$ and $T = 90/\epsilon$.

In each iteration the estimate a will be such that $\big|a - \|P^t |s\rangle\|\big| \leq \epsilon'$ with probability $1-\delta$, and hence

$$\big|a^2 - \|P^t |s\rangle\|^2\big| = \big|(a - \|P^t |s\rangle\|)(a + \|P^t |s\rangle\|)\big| \leq 2\epsilon'.$$

Nachmias and Shapira [43] showed that if G has a conductance $\geq \Upsilon$, then for all nodes s it holds that

$$\|P^t |s\rangle\| \leq M = \sqrt{N^{-1}(1+N^{-1})}.$$

Given such a graph, in each iteration the estimate $a \leq M + \epsilon'$ with probability $1-\delta$, so that the probability of a faulty rejection is at most δ per iteration. The total probability of a faulty rejection can then be bounded by $T\delta < 1/3$.

In the negative case, [43] showed that there exists a constant $c > 0$ such that if G is ϵ-far from any graph with max degree d and conductance $\geq c\mu\Upsilon^2$, then there exist at least $\epsilon N/128$ vertices s for which it holds that

$$\|P^t |s\rangle\| \geq \sqrt{N^{-1}(1+32^{-1}N^{-2\mu})}.$$

Given such a graph, in each iteration the estimate a will now be such that $a \geq \|P^t |s\rangle\| - \epsilon' \geq \sqrt{N^{-1}(1+32^{-1}N^{-2\mu})} - \epsilon'$ with probability $1-\delta$. To show that this quantity $> M + \epsilon'$, we bound $M = \sqrt{N^{-1}(1+N^{-1})} \leq N^{-1/2}(1+N^{-1/2})$ and $\sqrt{N^{-1}(1+32^{-1}N^{-2\mu})} \geq N^{-1/2}(1 - N^{-\mu}/(4\sqrt{2}))$, which shows that

$$\sqrt{N^{-1}(1+32^{-1}N^{-2\mu})} - M \geq N^{-1/2-\mu}/(4\sqrt{2}).$$

This proves that indeed

$$\sqrt{N^{-1}(1+32^{-1}N^{-2\mu})} - \epsilon' > M + \epsilon'$$

for $\epsilon' = N^{-1/2-\mu}/(16\sqrt{2})$. As a consequence, a single iteration will correctly output a rejection with probability $(1-\delta)\epsilon N/128$. The total probability of correctly rejecting at least once is therefore lower bounded by $T(1-\delta)\epsilon/128 \geq 2/3$. This concludes the proof that Algorithm 5 is a $(\Upsilon, \epsilon, \mu)$ graph expansion tester. The required number of QW steps is given by T times the number required by the 2-norm tester, which by Theorem 5 is

$$O\bigg(\frac{\sqrt{t}}{\epsilon'} \log\frac{1}{\delta} \log^{1/2}\frac{N}{\epsilon'}\bigg) \in O\bigg(\big(d\Upsilon^{-1}\log^{1/2} N\big) N^{1/2+\mu} \log\frac{1}{\epsilon} \log^{1/2} N^{1+\mu}\bigg).$$

We can implement a single QW step using $O(\sqrt{d})$ graph queries, so that we find the claimed bound. □

We recall that the classical GR tester has a runtime

$$O(N^{1/2+\mu} d^2 \Upsilon^{-2} \epsilon^{-1} \log N).$$

Up to our $\log(\epsilon^{-1})$-factor we improve this runtime with a factor Υ^{-1}, which basically follows from the fact that we quadratically accelerate the random walk runtime to $t \in \widetilde{\Theta}(\Upsilon^{-1})$. There exist bounded-degree graphs for which $\Upsilon \in \Omega(1/N)$, so that in some cases we improve the runtime by a factor $\Upsilon^{-1} \in \Theta(N)$. Concerning the space complexity, we note that the classical GR tester must store and compare the endpoints of $\Omega(N^{1/2})$ independent random walks. By direct inspection we see that our algorithm only requires $O(\log(Nt \log \epsilon^{-1})) \in \mathrm{polylog}(N)$ qubits to implement, exponentially improving the space complexity. This is due to the fact that our algorithm compares superpositions that encode the endpoint distribution of the random walks, rather than an explicit list of samples.

We can now compare this result to the preceding work by Ambainis, Childs and Liu [23]. They used a very different approach to speed up the GR expansion tester, using quantum walks only indirectly, which results in a runtime improvement of a different nature. In rough strokes they speed up the classical 2-norm tester by making use of Ambainis' quantum walk algorithm for element distinctness [1] to count collisions between pairs of classical random walks more efficiently. This allows them to improve the runtime of the 2-norm tester to $\widetilde{O}(N^{1/3+\mu}t)$, which provides a speedup complementary to the speedup of our 2-norm tester which in this context has a runtime $\widetilde{O}(N^{1/2+\mu}\sqrt{t})$. Using this 2-norm tester in the above Algorithm 5 leads to a runtime

$$\widetilde{O}(N^{1/3+\mu}d^2\Upsilon^{-2}\epsilon^{-1}).$$

The space complexity of this approach is comparable to the GR tester: the algorithm for element distinctness over the $\sqrt{N}$ random walk endpoints requires to store $N^{1/3}$ elements. We leave it as future work to combine the $\tilde{\Theta}(N^{1/6})$ gain in collision counting of [23] with our $\tilde{\Theta}(d\Upsilon^{-1})$ gain in random walk runtime and our logarithmic space complexity.

We note that a property tester in the same spirit as the GR expansion tester was proposed by Batu et al [46] for testing the mixing time of general Markov chains on a graph. For the special case of symmetric Markov chains it seems feasible that we can speed up their algorithm using the same ideas, yielding a similar speedup on the random walk runtime.

3.3 Clusterability and Robust s-t Connectivity

We can similarly use QFF to speed up a more recent line of algorithms on testing the clusterability of a graph [24, 25]. In clusterability testing the goal is to test whether a graph can be appropriately clustered into k parts for some given k. The proposed algorithms build on a subroutine of independent interest, which allows to determine whether a pair of nodes lie in the same cluster or not. This leads to a robust notion of s-t connectivity, useful e.g. for classifying objects among a set of examples and relevant also outside of the setting of property testing. We show that QFF allows to speed up this subroutine, leading to a speedup on the clusterability testers that use this subroutine.

The observation underlying the clusterability testers in [24, 25] is that the GR technique of counting collision can also be used to estimate the inner product of any two given distributions p and q, defined by

$$\langle p, q\rangle = \sum_j p(j)q(j).$$

Indeed, this quantity is equal to the collision probability between the two distributions. The estimate of the inner product is then used to estimate the 2-distance $\|p - q\|$ between a pair

of random walks, which will be small if both random walks started in the same cluster, and large otherwise. This approach of estimating the distance between distributions was further developed in the work by Batu et al [46], Valiant [26] and Chan et al [47]. We will focus our efforts on showing how QFF allows to speed up this routine of independent interest, following up with an informal discussion of how this leads to a speedup on the clusterability tester of Czumaj et al [24].

2-distance Estimator

To estimate the 2-distance of a pair of random walks, we will combine QFF with the SWAP test: given two quantum states $|\psi\rangle$ and $|\phi\rangle$ and an ancillary qubit in the state $|0\rangle$, yielding the state $|0\rangle|\psi\rangle|\phi\rangle$, we apply the following operations:

$$\begin{aligned} |0\rangle|\psi\rangle|\phi\rangle &\overset{H\otimes I\otimes I}{\to} \tfrac{|0\rangle+|1\rangle}{\sqrt{2}}|\psi\rangle|\phi\rangle \\ &\overset{cS}{\to} \tfrac{1}{\sqrt{2}}|0\rangle|\psi\rangle|\phi\rangle + \tfrac{1}{\sqrt{2}}|1\rangle|\phi\rangle|\psi\rangle \\ &\overset{H\otimes I\otimes I}{\to} \tfrac{1}{2}|0\rangle(|\psi\rangle|\phi\rangle + |\phi\rangle|\psi\rangle) + \tfrac{1}{2}|1\rangle(|\psi\rangle|\phi\rangle - |\phi\rangle|\psi\rangle), \end{aligned}$$

where we used the Hadamard gate $H = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$, and the *conditional swap operation* cS swapping the second and third registers conditional on the first register being in the state $|1\rangle$. We will call the combined unitary operation $U_{\text{SWAP}} = (H\otimes I\otimes I)cS(H\otimes I\otimes I)$. We can now either measure the first register, or apply quantum amplitude estimation to the projector $\Pi_1 = |1\rangle\langle 1|\otimes I\otimes I$, to estimate the quantity

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

This quantity will be small if $|\psi\rangle$ and $|\phi\rangle$ are close and large otherwise, allowing to estimate the distance between the input states $|\psi\rangle$ and $|\phi'\rangle$. We can combine the SWAP test with our QFF algorithm, and the 2-norm estimator in previous section, to obtain a tester for the 2-distance. Due to the straightforward yet technical nature of the details of the tester, we defer its description to Appendix A.

Theorem 7 (Quantum 2-distance Estimator) *With probability at least $1-\delta$, Algorithm 6 (Appendix A) outputs an estimate a such that*

$$\left| \|P^t|u\rangle - P^t|v\rangle\|^2 - a \right| \leq \epsilon.$$

For $\overline{a} = \max\{\|P^t|u\rangle\|, \|P^t|v\rangle\|\}$ and $\underline{a} = \min\{\|P^t|u\rangle\|, \|P^t|v\rangle\|\}$, the algorithm requires an expected number of QW steps bounded by

$$O\left(\sqrt{t}\left(\frac{\overline{a}}{\epsilon} + \frac{\overline{a}^4}{\underline{a}\epsilon^2}\right)\log\frac{\log N}{\delta}\log^{3/2}\frac{N}{\epsilon}\right).$$

For comparison, the classical estimator presented in Czumaj et al [24, Theorem 3.1] requires a number of graph queries or random walk steps $O\big(t\frac{\overline{a}}{\epsilon}\log\frac{1}{\delta}\big)$. Chan et al [47] give an information theoretical proof that classically $\Omega(\overline{a}/\epsilon)$ samples are needed to estimate the 2-distance between a pair of distributions.

Classifying Nodes

Czumaj et al [24] use their classical 2-distance estimator to propose a property tester for the clusterability of a graph. Following for instance Oveis Gharan and Trevisan [48], they say that a graph G is $(k, \Phi_{\text{in}}, \Phi_{\text{out}})$-*clusterable* if and only if there exists a partition $\mathcal{V} = \mathcal{S}_1 \cup \cdots \cup \mathcal{S}_h$, $h \leq k$, such that the clusters are well-connected internally, $\Phi(G[\mathcal{S}_j]) \geq \Phi_{\text{in}}$, and poorly-connected externally, $\Phi(\mathcal{S}_j) \leq \Phi_{\text{out}}$. Here $G[\mathcal{S}_j]$ denotes the graph consisting of the nodes in $\mathcal{S}_j$ and the edges between these nodes, the conductance $\Phi(\mathcal{S}_j)$ is defined as

$$\Phi(\mathcal{S}_j) = \frac{|E(\mathcal{S}_j, \mathcal{S}_j^c)|}{d|\mathcal{S}_j|},$$

and the conductance $\Phi(G')$ of a graph $G' = (\mathcal{V}', \mathcal{E}')$ is

$$\Phi(G') = \min_{\mathcal{T} \subset \mathcal{V}', |\mathcal{T}| \leq |\mathcal{V}'|/2} \frac{|E(\mathcal{T}, \mathcal{V}' \backslash \mathcal{T})|}{d|\mathcal{T}|}.$$

It turns out that graph clusterability can be efficiently tested when the gap between Φ_{in} and Φ_{out} is sufficiently large - typically quadratic, $\Phi_{\text{out}} \in \widetilde{O}(\Phi_{\text{in}}^2)$.

Czumaj et al [24] construct such a clusterability tester using a subroutine for *classifying nodes*, i.e., determining whether two nodes lie in the same cluster or not. As mentioned before, it is possible to classify nodes by comparing random walks starting from the nodes: the 2-distance between random walks starting from nodes of the same cluster will typically be smaller than the 2-distance between nodes from different clusters. This is formalized below in Lemma 6, which we extract from Czumaj et al [24, Lemma 4.1 and 4.3]. Given an appriopriately clusterable graph, having a gap $\Phi_{\text{out}} \in O(\Phi_{\text{in}}^2 / \log N)$, it gives bounds on the 2-distance between pairs of nodes coming from the same or different clusters. The lemma is confined to the *internal nodes* $\tilde{\mathcal{S}} \subseteq \mathcal{S}$ of a cluster $\mathcal{S}$, similar to most work on locally exploring graph clusters, see for instance the work of Spielman and Teng [49].

Lemma 6 ([24]) *Consider a $(k, \Phi_{\text{in}}, \Phi_{\text{out}})$-clusterable graph with degree bound d, and let $\mathcal{S}$ and $\mathcal{S}'$ be clusters of such a partition. Assume that*

$$\Phi_{\text{out}} \leq c\, \Phi_{\text{in}}^2 / \log N,$$

with c some constant dependent on d, k, $|\mathcal{S}|/N$ and $|\mathcal{S}'|/N$. Then there exist subsets $\tilde{\mathcal{S}} \subseteq \mathcal{S}$, $|\tilde{\mathcal{S}}| \geq |\mathcal{S}|/2$, and $\tilde{\mathcal{S}}' \subseteq \mathcal{S}'$, $|\tilde{\mathcal{S}}'| \geq |\mathcal{S}'|/2$, and a universal constant c', such that for $t = \lceil c' k^4 \Phi_{\text{in}}^{-2} \log N \rceil$ it holds that

- *if two nodes $u, v \in \tilde{\mathcal{S}}$ or $u, v \in \tilde{\mathcal{S}}'$, then $\| P^t |u\rangle - P^t |v\rangle \|^2 \leq 1/(4N)$.*
- *if two nodes $u \in \tilde{\mathcal{S}}$ and $v \in \tilde{\mathcal{S}}'$, then $\| P^t |u\rangle - P^t |v\rangle \|^2 \geq 1/N$.*

We can combine this lemma with our quantum 2-distance estimator to prove the below proposition. It speeds up the routine which lies at the basis of the property tester in [24], which essentially solves a robust version of s-t connectivity. Arguably the latter is more relevant to e.g. social networks, where mere connectivity between two nodes is no longer deemed an interesting quantity, yet the community or cluster structure does hold important information.

Proposition 3 (Classifying Nodes)
- *Under the clusterability conditions of Lemma 6, we can use the quantum 2-distance estimator to determine with probability at least* $2/3$ *whether two internal nodes lie in the same cluster or not.*
- *There exists a subset* $\tilde{\mathcal{V}} \subseteq \mathcal{V}$, $|\tilde{\mathcal{V}}| \geq 9|\mathcal{V}|/10$, *such that if in addition both nodes lie in* $\tilde{\mathcal{V}}$, *then the algorithm requires* $O(N^{1/2}k^4\Phi_{\text{in}}^{-1}\log^{3/2} N)$ *expected QW steps.*

Proof. To prove the first bullet, it suffices to use Lemma 6 which states that if both are internal nodes of the same cluster, then $\|P^t|u\rangle - P^t|v\rangle\|^2 \leq 1/(4N)$, whereas if both are internal nodes of different clusters, then $\|P^t|u\rangle - P^t|v\rangle\|^2 \geq 1/N$. By Theorem 7 we can estimate $\|P^t|u\rangle - P^t|v\rangle\|^2$ to error $\epsilon = 1/N$, which allows to distinguish both cases.

To prove the second bullet, let $\tilde{\mathcal{V}}$ denote a set of nodes u for which $\|P^t|u\rangle\| \in O(k/N)$, which by [24, Lemma 4.2] we know we can choose of size at least $9|\mathcal{V}|/10$. If both nodes lie in $\tilde{\mathcal{V}}$, then in Theorem 7 we can set $\overline{a} \in O(k/N)$, and $\underline{a} \in O(1/N)$ since necessarily $\|P^t|u\rangle\| \geq 1/N$ for any node u. In this case, the expected number of QW steps becomes $O(\sqrt{tN}\log^{3/2} N)$. For t as in Lemma 6, this proves the second bullet. □

We can compare the runtime in the second bullet, $O(N^{1/2}k^4\Phi_{\text{in}}^{-1}\log^{3/2} N)$, to the runtime when using classical collision counting, which requires a number of RW steps $\widetilde{O}(N^{1/2}k^4\Phi_{\text{in}}^{-2})$. Applying the element distinctness technique by Ambainis et al [23] requires a number of QW steps $\widetilde{O}(N^{1/3}k^4\Phi_{\text{in}}^{-2})$. Again we also find an improvement in space complexity with respect to these alternative approaches: our algorithm only requires polylog(N) qubits, whereas the other approaches require poly(N) classical or quantum bits.

Lemma 6, combined with a classifier as in Proposition 3, forms the basis of the graph clusterability tester proposed by [24]. Since the tester is in the same vein as the GR expansion tester, we will not state it explicitly but merely summarize the idea. The algorithm selects a uniformly random set of $\Theta(k\log k)$ nodes over which it constructs a *similarity graph* by adding an edge between any pair of nodes if their random walk probabilities are closer than some threshold. This similarity graph serves as a *graph sketch*, reminiscent of the recent surge of results on graph sketching and sparsification [50]. They then prove that if the graph is appropriately clusterable in at most k components, then with high probability this small similarity graph will have at most k connected components, which they then check by brute force. Using the classical 2-distance estimator to estimate the distance between random walk distributions, this leads to a clusterability tester requiring $\widetilde{O}\big(N^{1/2}k^7\Phi_{\text{in}}^{-2}\epsilon^{-5}\big)$ RW steps. We can improve this to $\widetilde{O}\big(N^{1/2}k^7\Phi_{\text{in}}^{-1}\epsilon^{-4}\big)$ QW steps using Proposition 3. It seems feasible that using the element distinctness technique in [36] an alternative speedup to $\widetilde{O}\big(N^{1/3}k^7\Phi_{\text{in}}^{-2}\epsilon^{-5}\big)$ RW steps can be achieved.

4 Discussion and Open Questions

We introduced a new quantum walk tool called quantum fast-forwarding (QFF), allowing to quantum simulate classical reversible Markov chains with a quadratically improved time dependency. The main benefit of this tool is that it allows to effectively simulate the transient dynamics of the Markov chains. We can contrast this to many existing quantum walk algorithms which rely on a speedup of the Markov chain limit behavior. This new feature is crucial for the applications in graph property testing and node classification that we discuss.

Indeed we show that QFF allows to speed up in a very natural way random walk algorithms for testing graph properties such as expansion and clusterability, both of which decisively depend on the transient dynamics of a random walk.

To finalize we mention some avenues for future work:

- *Improving the QFF scheme: parameter dependence and irreversible Markov chains.* QFF allows to create an ϵ-approximation of the state $|D^t v\rangle$ with constant success probability using a number of QW steps

$$O\left(\frac{\sqrt{t}}{\|D^t |v\rangle\|} \log^{1/2} \frac{1}{\epsilon \|D^t |v\rangle\|}\right)$$

  and $O(\|D^t |v\rangle\|^{-1})$ reflections around the initial state $|v\rangle$. It is easy to see that the individual t and ϵ dependency are optimal by looking at the random walk on $\mathbb{Z}$. If we tolerate an ϵ error, then we can confine the probability distribution of a t-step random walk to the $\Theta\big(t^{1/2} \log^{1/2} \epsilon^{-1}\big)$ neighborhood of the initial state. Since the QW has the same locality constraints as the RW, it needs $\Omega\big(t^{1/2} \log^{1/2} \epsilon^{-1}\big)$ QW steps to spread out over this interval. A very similar argument also shows why in general QFF cannot create the state $|P^t v\rangle$ (rather than $|D^t v\rangle$) when P is irreversible. Indeed, consider the Markov chain on $\mathbb{Z}$ which simply moves to the right every step, $P(i+1, i) = 1$ and $P(i-1, i) = 0$. This walk is clearly not reversible, as the direction of its motion reverses when running the time forward or backward. When starting in the origin, the walk will be on node t after t steps. A local QW requires $\Omega(t)$ steps to reach this point, so that no fast-forwarding is possible.

  We leave improvements of the dependency on $\|D^t |v\rangle\|$ as an open question.

- *Local Graph Clustering and Sparsification.* Local graph clustering algorithms, as in [49, 51], aim to explicitly construct a local cluster, rather than merely test whether appropriate clusters exist. They have a similar flavor to the graph expansion tester that we discussed, making use of random walks and other diffusive dynamics as a way of locally exploring a graph. It might be possible to use QFF or similar ideas as a way of speeding up these algorithms. Since these algorithms formed the root of a number of approaches towards graph sparsification and solving symmetric diagonally-dominant linear systems, this might lead to speedups on these highly relevant problems as well.

- *QFF of Hamiltonians or block-encoded matrices.* We restricted our QFF result to fast-forwarding Markov chains. Following the recent work on block-encoded matrices [29, 30, 20], we can straightforwardly extend it to accelerate the implementation of more general matrices, corresponding to the relevant matrices in linear system problems or the Hamiltonian of a quantum system. We leave it as an open question whether for instance the fast-forwarding of Hamiltonians can lead to interesting applications (e.g. in imaginary time evolution [52] or an improved implementation of functions of a Hamiltonian [53]).

References

[1] Ambainis, A.: Quantum walk algorithm for element distinctness. SIAM Journal on Computing 37(1), 210–239 (2007)

[2] Magniez, F., Santha, M., Szegedy, M.: Quantum algorithms for the triangle problem. SIAM Journal on Computing 37(2), 413–424 (2007)

[3] Childs, A.M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., Spielman, D.A.: Exponential algorithmic speedup by a quantum walk. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing. pp. 59–68. ACM (2003)

[4] Szegedy, M.: Quantum speed-up of Markov chain based algorithms. In: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science. pp. 32–41. IEEE (2004)

[5] Krovi, H., Magniez, F., Ozols, M., Roland, J.: Quantum walks can find a marked element on any graph. Algorithmica 74(2), 851–907 (2016)

[6] Ambainis, A., Bach, E., Nayak, A., Vishwanath, A., Watrous, J.: One-dimensional quantum walks. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing. pp. 37–49. ACM (2001)

[7] Aharonov, D., Ambainis, A., Kempe, J., Vazirani, U.: Quantum walks on graphs. In: Proceedings of the 33rd Annual ACM Symposium on Theory of Computing. pp. 50–59. ACM (2001)

[8] Richter, P.C.: Quantum speedup of classical mixing processes. Physical Review A 76(4), 042306 (2007)

[9] Somma, R.D., Boixo, S., Barnum, H., Knill, E.: Quantum simulations of classical annealing processes. Physical Review Letters 101(13), 130504 (2008)

[10] Wocjan, P., Abeyesinghe, A.: Speedup via quantum sampling. Physical Review A 78(4), 042336 (2008)

[11] Aharonov, D., Ta-Shma, A.: Adiabatic quantum state generation and statistical zero knowledge. In: Proceedings of the 35th Annual ACM Symposium on Theory of Computing. pp. 20–29. ACM (2003)

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

[13] Santha, M.: Quantum walk based search algorithms. In: International Conference on Theory and Applications of Models of Computation. pp. 31–46. Springer (2008)

[14] Watrous, J.: Quantum simulations of classical random walks and undirected graph connectivity. Journal of Computer and System Sciences 62(2), 376–391 (2001)

[15] Aleliunas, R., Karp, R.M., Lipton, R.J., Lovasz, L., Rackoff, C.: Random walks, universal traversal sequences, and the complexity of maze problems. In: Proceedings of the 20th Annual IEEE Symposium on Foundations of Computer Science. pp. 218–223. IEEE (1979)

[16] Childs, A.M., Wiebe, N.: Hamiltonian simulation using linear combinations of unitary operations. Quantum Information and Computation 12(11& 12), 901–924 (2012)

[17] Berry, D.W., Childs, A.M., Cleve, R., Kothari, R., Somma, R.D.: Simulating hamiltonian dynamics with a truncated taylor series. Physical Review Letters 114(9), 090502 (2015)

[18] Van Apeldoorn, J., Gilyén, A., Gribling, S., de Wolf, R.: Quantum SDP-solvers: better upper and lower bounds. In: Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science. pp. 403–414. IEEE (2017)

[19] Diaconis, P., Miclo, L.: On the spectral analysis of second-order Markov chains. Annales de la Faculté des Sciences de Toulouse. Mathématiques 22(3), 573–621 (2013)

[20] Gilyén, A., Su, Y., Low, G.H., Wiebe, N.: Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. arXiv:1806.01838 (2018)

[21] Spielman, D.A., Teng, S.H.: A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. SIAM Journal on Computing 42(1), 1–26 (2013)

[22] Goldreich, O., Ron, D.: On testing expansion in bounded-degree graphs. In: Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation, pp. 68–75. Springer (2011)

[23] Ambainis, A., Childs, A.M., Liu, Y.K.: Quantum property testing for bounded-degree graphs. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, pp. 365–376. Springer (2011)

[24] Czumaj, A., Peng, P., Sohler, C.: Testing cluster structure of graphs. In: Proceedings of the 47th Annual ACM Symposium on Theory of Computing. pp. 723–732. ACM (2015)

[25] Chiplunkar, A., Kapralov, M., Khanna, S., Mousavifar, A., Peres, Y.: Testing graph clusterability: Algorithms and lower bounds. In: Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science. IEEE (2018)

[26] Valiant, P.: Testing symmetric properties of distributions. SIAM Journal on Computing 40(6), 1927–1968 (2011)

[27] Ambainis, A., Gilyén, A., Jeffery, S., Kokainis, M.: Quadratic speedup for finding marked vertices by quantum walks. personal communication (2019)

[28] Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. SIAM Journal on Computing 40(1), 142–164 (2011)

[29] Low, G.H., Chuang, I.L.: Optimal hamiltonian simulation by quantum signal processing. Physical review letters 118(1), 010501 (2017)

[30] Chakraborty, S., Gilyén, A., Jeffery, S.: The power of block-encoded matrix powers: improved regression techniques via faster hamiltonian simulation. arXiv preprint arXiv:1804.01973 (2018)

[31] Poulin, D., Wocjan, P.: Sampling from the thermal quantum gibbs state and evaluating partition functions with a quantum computer. Physical Review Letters 103(22), 220502 (2009)

[32] Nielsen, M.A., Chuang, I.: Quantum computation and quantum information. Cambridge University Press (2002)

[33] Berry, D.W., Childs, A.M.: Black-box hamiltonian simulation and unitary implementation. Quantum Information and Computation 12(1& 2), 29–62 (2012)

[34] Gil, A., Segura, J., Temme, N.M.: Numerical methods for special functions, vol. 99. SIAM (2007)

[35] Berry, D.W., Childs, A.M., Cleve, R., Kothari, R., Somma, R.D.: Exponential improvement in precision for simulating sparse hamiltonians. In: Forum of Mathematics, Sigma. vol. 5. Cambridge University Press (2017)

[36] Ambainis, A., Magnin, L., Roetteler, M., Roland, J.: Symmetry-assisted adversaries for quantum state generation. In: Proceedings of the 26th Annual Conference on Computational Complexity. pp. 167–177. IEEE (2011)

[37] Brassard, G., Hoyer, P., Mosca, M., Tapp, A.: Quantum amplitude amplification and estimation. Contemporary Mathematics 305, 53–74 (2002)

[38] Yoder, T.J., Low, G.H., Chuang, I.L.: Fixed-point quantum search with an optimal number of queries. Physical Review Letters 113(21), 210501 (2014)

[39] Sachdeva, S., Vishnoi, N.K., et al.: Faster algorithms via approximation theory. Foundations and Trends® in Theoretical Computer Science 9(2), 125–210 (2014)

[40] Kumar, A., Seshadhri, C., Stolman, A.: Fiding forbidden minors in sublinear time: a $o(n^{1/2+o(1)})$-query one-sided tester for minor closed properties on bounded degree graphs. arXiv:1805.08187 (2018)

[41] Czumaj, A., Sohler, C.: Testing expansion in bounded-degree graphs. Combinatorics, Probability and Computing 19(5-6), 693–709 (2010)

[42] Kale, S., Seshadhri, C.: An expansion tester for bounded degree graphs. SIAM Journal on Computing 40(3), 709–720 (2011)

[43] Nachmias, A., Shapira, A.: Testing the expansion of a graph. Information and Computation 208(4), 309 (2010)

[44] Goldreich, O., Ron, D.: Property testing in bounded degree graphs. In: Proceedings of the 29th Annual ACM Symposium on Theory of Computing. pp. 406–415. ACM (1997)

[45] Montanaro, A., de Wolf, R.: A survey of quantum property testing. Theory of Computing Library Graduate Surveys (7), 1–81 (2016)

[46] Batu, T., Fortnow, L., Rubinfeld, R., Smith, W.D., White, P.: Testing closeness of discrete distributions. Journal of the ACM (JACM) 60(1), 4 (2013)

[47] Chan, S.O., Diakonikolas, I., Valiant, P., Valiant, G.: Optimal algorithms for testing closeness of discrete distributions. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1193–1203. SIAM (2014)

[48] Gharan, S.O., Trevisan, L.: Partitioning into expanders. In: Proceedings of the 25th Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1256–1266. SIAM (2014)

[49] Spielman, D.A., Teng, S.H.: Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In: Proceedings of the 36th Annual ACM symposium on Theory of Computing. pp. 81–90. ACM (2004)

[50] Batson, J., Spielman, D.A., Srivastava, N., Teng, S.H.: Spectral sparsification of graphs: theory and algorithms. Communications of the ACM 56(8), 87–94 (2013)

[51] Andersen, R., Peres, Y.: Finding sparse cuts locally using evolving sets. In: Proceedings of the 41st Annual ACM Symposium on Theory of Computing. pp. 235–244. ACM (2009)

[52] Verstraete, F., Murg, V., Cirac, I.J.: Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. Advances in Physics 57(2), 143–224 (2008)

[53] Childs, A.M., Kothari, R., Somma, R.D.: Quantum algorithm for systems of linear equations with exponentially improved dependence on precision. SIAM Journal on Computing 46(6), 1920–1950 (2017)

A Quantum 2-distance Estimator: Algorithm and Proof

In this appendix we present the algorithm and proof underlying Theorem 7, which concerns the estimation of the distance between two random walk distributions $p = P^t |u\rangle$ and $q = P^t |v\rangle$. To construct our algorithm, we rewrite

$$\|p - q\|^2 = \|p\|^2 + \|q\|^2 - 2\|p\|\|q\| \langle p|q\rangle ,$$

using the notation $\langle p|q\rangle = \langle p, q\rangle / (\|p\|\|q\|)$. As a consequence, we can retrieve an estimate by separately estimating $\|p\|$, $\|q\|$ and $\langle p|q\rangle$. Towards estimating $\|p\|$ and $\|q\|$, we present at the end of this appendix a simple extension of the quantum 2-norm tester presented earlier in Section 3.2 that allows to estimate the 2-norm up to multiplicative error, instead of additive error. Towards estimating $\langle p|q\rangle$, we first create approximations of $|p\rangle = p/\|p\|$ and $|q\rangle = q/\|q\|$, on which we subsequently apply the SWAP test and amplitude estimation. A subtlety is that we cannot simply use our QFF algorithm to create $|p\rangle$ and $|q\rangle$ with high probability. Indeed, in order to apply amplitude estimation for the SWAP test we must reflect around these states, and it is not clear that we can reflect around the output of the QFF algorithm. Instead, we will apply the unitary amplitude amplification operator to the states $W_\tau |u, \bar{0}\bar{0}\rangle$ and $W_\tau |v, \bar{0}\bar{0}\rangle$ to unitarily rotate these states close to $|p\rangle$ and $|q\rangle$, omitting the final measurement in Algorithm 2. This invertible operation will allow to reflect around the output states. Furthermore, instead of the amplitude amplification operator used in Section 2.2.2, we will make use of an enhanced operator by Yoder and Low [38]. This operator, as described in the below lemma, is better suited for the case where we only have a lower bound on the success probability.

Lemma 7 (Fixed Point Amplitude Amplification [38]) *Consider a state $|\psi\rangle$ and a projector $\Pi_{\bar{0}}$ such that $\|\Pi_{\bar{0}}|\psi\rangle\| = \lambda > 0$. For any constant $\delta > 0$, there exists a family of unitary transformations U_L such that if $L \geq \lambda^{-1}\log(2/\delta)$ then*

$$|\langle\psi_{\bar{0}}|U_L|\psi\rangle|^2 \geq 1-\delta^2,$$

where $\psi_{\bar{0}} = \Pi_{\bar{0}}|\psi\rangle / \|\Pi_{\bar{0}}|\psi\rangle\|$. We can implement U_L using $O(L)$ reflections around $|\psi\rangle$ and around the image of $\Pi_{\bar{0}}$.

Using the appropriate operator U_L, we can therefore retrieve approximations $|\psi_u\rangle = U_L W_\tau |u,\bar{0}\bar{0}\rangle \approx |p\rangle$ and $|\psi_v\rangle = U_L W_\tau |v,\bar{0}\bar{0}\rangle \approx |q\rangle$. We can now apply the SWAP test to these states, combined with amplitude amplification, to retrieve an estimation of $\langle p|q\rangle$. To see this, note that

$$\Pi_1(U_{\text{SWAP}}|0\rangle|\psi_u\rangle|\psi_v\rangle) = \frac{1}{2}|1\rangle(|\psi_u\rangle|\psi_v\rangle - |\psi_v\rangle|\psi_u\rangle).$$

As a consequence we can apply quantum amplitude estimation on the state $U_{\text{SWAP}}|0\rangle|\psi_u\rangle|\psi_v\rangle$ with respect to the projector Π_1 to estimate the quantity

$$\frac{1}{2}\big\||\psi_u\rangle|\psi_v\rangle - |\psi_v\rangle|\psi_u\rangle\big\|^2 = 1 - |\langle\psi_u|\psi_v\rangle|^2 \approx 1 - |\langle p|q\rangle|^2.$$

Combined with the former estimates of $\|p\|$ and $\|q\|$ this leads to an estimate of the 2-distance we were looking for. We formalize this in the following algorithm and theorem.

Algorithm 6 Quantum 2-distance Estimator

Input: parameters N and d; query access to N-node graph G with degree bound d; starting vertices u and v; running time t; accuracy parameter ϵ; confidence parameter δ

Do:

1: use Algorithm 7 to create estimates α and β of $\|P^t|u\rangle\|$ resp. $\|P^t|v\rangle\|$ to multiplicative error $1/4$, with probability $1-\delta/4$

2: set $\mu \in O(\epsilon \max(\alpha,\beta)^{-2})$

3: use Algorithm 7 to create new estimates α and β of $\|P^t|u\rangle\|$ resp. $\|P^t|v\rangle\|$ to multiplicative error μ, with probability $1-\delta/4$

4: set $L \in \Omega(\min(\alpha,\beta)^{-1}\log\min(\alpha,\beta)^{-1})$ and $\tau \in \Omega(\sqrt{t\ln(N/\mu)})$

5: apply W_τ, U_L and U_{SWAP} to create the state

$$|\psi\rangle = U_{\text{SWAP}}|0\rangle\big(U_L W_\tau|u,\bar{0}\bar{0}\rangle\big)\big(U_L W_\tau|v,\bar{0}\bar{0}\rangle\big)$$

6: use amplitude estimation to create an estimate γ of $\|\Pi_1|\psi\rangle\|$ to error μ, with probability $1-\delta/2$

Output: estimate $a = \alpha^2+\beta^2-2\alpha\beta\sqrt{1-\gamma^2/2}$

Theorem 8 (Quantum 2-distance Estimator) *With probability at least $1-\delta$, Algorithm 6 outputs an estimate a such that*

$$\big|\|P^t|u\rangle - P^t|v\rangle\|^2 - a\big| \leq \epsilon.$$

With $\overline{a} = \max\{\|P^t |u\rangle\|, \|P^t |v\rangle\|\}$ *and* $\underline{a} = \min\{\|P^t |u\rangle\|, \|P^t |v\rangle\|\}$*, the algorithm requires an expected number of QW steps bounded by*

$$O\left(\sqrt{t}\left(\frac{\overline{a}}{\epsilon} + \frac{\overline{a}^4}{\underline{a}\epsilon^2}\right)\log\frac{\log N}{\delta}\log^{3/2}\frac{N}{\epsilon}\right).$$

Proof. We prove the theorem for

$$\mu = \frac{1}{26}\min\left(1, \frac{9\epsilon}{16\max(\alpha,\beta)^2}\right), \quad L = \left\lceil\frac{1}{\lambda}\log\frac{2}{\nu}\right\rceil, \quad \tau = \left\lceil\sqrt{2t}\ln^{1/2}\frac{4}{\lambda\nu}\right\rceil,$$

with $\lambda = \min(\alpha,\beta)/(1+\nu)$ and $\nu = \mu^2/11$. We will denote $p = P^t |u\rangle$, $q = P^t |v\rangle$, $|p\rangle = p/\|p\|$, $|q\rangle = q/\|q\|$, $\overline{a}^2 = \max(\|p\|, \|q\|)$ and $\underline{a} = \min(\|p\|, \|q\|)$. The algorithm estimates the quantity $\|p-q\|^2 = \|p\|^2 + \|q\|^2 - 2\|p\|\|q\|\langle p|q\rangle$ by separately estimating $\|p\|$, $\|q\|$ and $\langle p|q\rangle$ to error $O(\epsilon/\overline{a}^2)$.

After the first step, we retrieve with probability at least $1-\delta/4$ estimates α and β such that

$$\frac{3}{4}\|p\| \leq \alpha \leq \frac{5}{4}\|p\|, \qquad \frac{3}{4}\|q\| \leq \beta \leq \frac{5}{4}\|q\|.$$

This proves that the parameter

$$\mu = \frac{1}{26}\min\left(1, \frac{\epsilon}{(4\max(\alpha,\beta)/3)^2}\right) \leq \frac{1}{26}\min\left(1, \frac{\epsilon}{\overline{a}^2}\right), \tag{10}$$

and $\mu \in \Theta(\min(1, \epsilon/\overline{a}^2))$. In step 3 we then create new estimates of $\|p\|$ and $\|q\|$ to multiplicative error μ. The combined success probability of both steps is $(1-\delta/4)^2 \geq 1-\delta/2$. Following Theorem 9 these steps require an expected number of QW steps in

$$O\left(\frac{\sqrt{t}\overline{a}}{\epsilon}\log\frac{\log N}{\delta}\log^{1/2}\frac{N}{\epsilon}\right).$$

In the following steps of the algorithm we estimate $\langle p|q\rangle = \frac{\langle p,q\rangle}{\|p\|\|q\|}$ to additive error μ by combining QFF, amplitude amplification, the SWAP test and amplitude estimation. Thereto we first rewrite

$$\langle p|q\rangle = \sqrt{1 - \frac{\| |p\rangle|q\rangle - |q\rangle|p\rangle\|^2}{2}},$$

showing that we can use an estimate on $\| |p\rangle|q\rangle - |q\rangle|p\rangle\|$ to estimate $\langle p|q\rangle$. Indeed, it is easily seen from a function plot that if we create an estimate $\kappa \in [0, \sqrt{2}]$ such that $\left|\| |p\rangle|q\rangle - |q\rangle|p\rangle\| - \kappa\right| \leq \mu^2$, then the estimate $\sqrt{1-\kappa^2/2}$ will be μ-close:

$$\left|\sqrt{1-\kappa^2/2} - \langle p|q\rangle\right| \leq \mu. \tag{11}$$

We now create an estimate of $\| |p\rangle|q\rangle - |q\rangle|p\rangle\|$. By Lemma 7 and Theorem 3, and our choice of L and τ, it holds that

$$\begin{aligned}\|U_L W_\tau |u,\overline{0}\overline{0}\rangle - |p,\overline{0}\overline{0}\rangle\| &\leq (1-\nu^2)\|W_\tau |u,\overline{0}\overline{0}\rangle / \|W_\tau |u,\overline{0}\overline{0}\rangle\| - |p,\overline{0}\overline{0}\rangle\| + \nu \\ &\leq (1-\nu^2)\nu + \nu \leq 2\nu,\end{aligned}$$

with $\nu = \mu^2/11$, and similarly for $U_L W_\tau |v,\bar{0}\bar{0}\rangle$. If we set $|\psi_u\rangle = U_L W_\tau |u,\bar{0}\bar{0}\rangle$ and $|\psi_v\rangle = U_L W_\tau |v,\bar{0}\bar{0}\rangle$, then this implies that

$$\big|\| |\psi_u\rangle |\psi_v\rangle - |\psi_v\rangle |\psi_u\rangle \| - \| |p\rangle |q\rangle - |q\rangle |p\rangle \|\big| \leq 8\nu(1+2\nu).$$

Now we can apply amplitude estimation, as in Lemma 5, to the state $U_{\text{SWAP}} |0\rangle |\psi_u\rangle |\psi_v\rangle$ and projector Π_1 with success probability $1-\delta/2$ and error ν. If successful this returns an estimate γ of $\| |\psi_u\rangle |\psi_v\rangle - |\psi_v\rangle |\psi_u\rangle \|$ to error ν. Combined with the above inequality this shows that

$$\big|\| |p\rangle |q\rangle - |q\rangle |p\rangle \| - \gamma\big| \leq \nu + 8\nu(1+2\nu) \leq \mu^2.$$

By (11) this leads to the promised bound $\big|\sqrt{1-\gamma^2/2} - \langle p|q\rangle\,\big| \leq \mu$.

Implementing W_τ, U_L and U_{SWAP} requires a number of QW steps $O(\tau)+O(L)$, bounded by

$$O\left(\frac{\sqrt{t}}{\underline{a}} \log\frac{\overline{a}}{\epsilon\underline{a}} \log^{1/2}\frac{N\overline{a}}{\epsilon}\right).$$

Applying amplitude estimation with success probability $1-\delta/2$ and error $\nu \in \Theta(\epsilon^2/\overline{a}^4)$ requires $O\big(\frac{\overline{a}^4}{\epsilon^2}\log\frac{1}{\delta}\big)$ reflections around the state $U_{\text{SWAP}} |0\rangle |\psi_u\rangle |\psi_v\rangle$. We can implement each such reflection using the same number of QW steps required to implement the operators W_τ, U_L and U_{SWAP}. This leads to a total number of QW steps bounded by

$$O\left(\frac{\sqrt{t}\overline{a}^4}{\underline{a}\epsilon^2} \log\frac{1}{\delta} \log\frac{\overline{a}}{\epsilon\underline{a}} \log^{1/2}\frac{N\overline{a}}{\epsilon}\right).$$

Combined with the first approximation part, we find estimates α, β and γ such that $|\alpha - \||p\rangle\|| \leq \mu\||p\rangle\|$, $|\beta - \||q\rangle\|| \leq \mu\||q\rangle\|$ and $|\gamma - \langle p|q\rangle\,| \leq \mu$. This allows to prove the claimed error of the estimate

$$\begin{aligned}\big|\alpha^2+\beta^2-2\alpha\beta\gamma - \||p\rangle - |q\rangle\|^2\big| &\leq \mu(2+\mu)(\||p\rangle\|^2+\||q\rangle\|^2)\\ &\quad + 2\||p\rangle\|\||q\rangle\|\big[\mu(2+\mu)(\langle p|q\rangle+\mu)+(1+\mu)^2\mu\big]\\ &\leq 3\mu(\||p\rangle\|^2+\||q\rangle\|^2) + 20\mu\||p\rangle\|\||q\rangle\|\\ &\leq 26\mu\max(\||p\rangle\|,\||q\rangle\|)^2 \leq \epsilon,\end{aligned}$$

using the bound (10). The total success probability can be bounded by $(1-\delta/2)^2 \geq 1-\delta$, and the expected number of QW steps by

$$O\left(\sqrt{t}\left(\frac{\overline{a}}{\epsilon}+\frac{\overline{a}^4}{\underline{a}\epsilon^2}\right)\log\frac{\log N}{\delta}\log^{3/2}\frac{N}{\epsilon}\right).$$

□

2-norm Estimator to Multiplicative Error

In the above estimator for the 2-distance we wish to estimate $\|P^t |u\rangle\|$ to some multiplicative error ϵ, without having a bound on $\|P^t |u\rangle\|$. We present such an estimator in the below algorithm and theorem.

Algorithm 7 Quantum Multiplicative 2-norm Estimator

Input: parameters N and d; query access to N-node graph G with degree bound d; starting vertex u; running time t; accuracy parameter ϵ; confidence parameter δ

Do:

1: **for** $k = 1 \dots T \in O(\log N)$ **do**

2: use Algorithm 4 to create estimate α of $\|P^t |u\rangle\|$ to error $\epsilon_k = \epsilon 2^{-k-2}$, with probability $1 - \delta'$ for $\delta' \in O(\delta \log^{-1} N)$

3: if $\alpha \geq (1+\epsilon)2^{-k}$, abort **for**-loop

Output: α

Theorem 9 (Quantum Multiplicative 2-norm Estimator) *With probability at least* $1-\delta$*, Algorithm 6 outputs an estimate* α *such that*

$$\left| \|P^t |u\rangle\| - \alpha \right| \leq \epsilon \|P^t |u\rangle\|.$$

The algorithm requires an expected number of QW steps bounded by

$$O\left(\frac{\sqrt{t}}{\epsilon \|p\|} \log \frac{\log N}{\delta} \log^{1/2} \frac{N}{\epsilon} \right).$$

Proof. We will prove the theorem for $T = \lceil \frac{1}{2} \log N \rceil$ and $\delta' = \delta/T$. We do so by showing that with probability at least $1 - \delta$ the loop aborts such that the value of α forms an estimate of $\|p\|$ to multiplicative error ϵ, where we denote $p = P^t |u\rangle$. We first assume that every call to Algorithm 4 is successful, the probability of which we will bound afterwards. Let a_k be the value of α in the k-th iteration, so that $|\|p\| - a_k| \leq \epsilon_k$. If the loop is stopped at the k-th iteration then $a_k \geq (1+\epsilon)2^{-k}$ or equivalently $\epsilon_k \leq \frac{\epsilon}{1+\epsilon} a_k$. Combined with the fact that $a_k \leq \|p\| + \epsilon_k$ this shows that $\epsilon_k \leq \frac{\epsilon}{1+\epsilon}(\|p\| + \epsilon_k)$ or equivalently $\epsilon_k \leq \epsilon \|p\|$, so that we find an estimate with multiplicative error ϵ.

If the first $\lceil \log \|p\|^{-1} \rceil$ calls to the 2-norm estimator are successful, then the algorithm stops and outputs a correct estimate. We can bound this number of calls by $T = \lceil \frac{1}{2} \log N \rceil$ using the fact that $\|p\| \geq N^{-1/2}$. The probability that this happens, i.e., that none of the first $\lceil \log \|p\|^{-1} \rceil$ implementations of the 2-norm tester fails, is at least $1 - \lceil \log \|p\|^{-1} \rceil \delta' \geq 1 - \delta$ if we set $\delta' = \delta/T$. This proves the success probability of the algorithm.

To bound the runtime, we first note that the k-th iteration runs the 2-norm tester with error $\epsilon_k = \epsilon 2^{-k}$ and success probability $1 - \delta'$, which by Theorem 5 requires a number of QW steps

$$O\left(\frac{2^k \sqrt{t}}{\epsilon} \log \frac{\log N}{\delta} \log^{1/2} \frac{2^k N}{\epsilon} \right).$$

Now we bound the expected number of iterations. If the algorithm succeeds, then it aborts after $\lceil \log \|p\|^{-1} \rceil$ iterations. If this does not happen, then either it aborts earlier, resulting in a number of iterations smaller than $\lceil \log \|p\|^{-1} \rceil$, or it aborts later. However, after $\lceil \log \|p\|^{-1} \rceil$ iterations, any successful call to the 2-norm tester will abort the algorithm, which happens per iteration with probability at least $1 - \delta$. In such case the expected number of iterations can be bounded by $(1-\delta)^{-1} \leq 2$ under the assumption that $\delta \leq 1/2$. In any case we see that the expected number of iterations is $O(\log \|p\|^{-1})$. Now we can use the fact that

$\sum_{k=0}^{b} 2^k \log^{1/2} 2^k \in O(2^b\sqrt{b}) \in O(\|p\|^{-1} \log^{1/2} \|p\|)$ for $b \in O(\log \|p\|^{-1})$ to bound the total expected number of QW steps by

$$O\left(\frac{\sqrt{t}}{\epsilon\|p\|} \log \frac{\log N}{\delta} \log^{1/2} \frac{N}{\epsilon}\right).$$

This finalizes the proof. □