Quantum Walk Sampling by Growing Seed Sets

Simon Apers
Inria, Paris, France
CWI, Amsterdam, The Netherlands
simon.apers@inria.fr

Abstract

This work describes a new algorithm for creating a superposition over the edge set of a graph, encoding a quantum sample of the random walk stationary distribution. The algorithm requires a number of quantum walk steps scaling as $\widetilde{O}(m^{1/3}\delta^{-1/3})$, with m the number of edges and δ the random walk spectral gap. This improves on existing strategies by initially growing a classical seed set in the graph, from which a quantum walk is then run.

The algorithm leads to a number of improvements: (i) it provides a new bound on the setup cost of quantum walk search algorithms, (ii) it yields a new algorithm for *st*-connectivity, and (iii) it allows to create a superposition over the isomorphisms of an n-node graph in time $\widetilde{O}(2^{n/3})$, surpassing the $\Omega(2^{n/2})$ barrier set by index erasure.

2012 ACM Subject Classification Theory of computation → Quantum computation theory; Theory of computation → Graph algorithms analysis

Keywords and phrases Quantum algorithms, Quantum walks, Connectivity, Graph theory

Digital Object Identifier 10.4230/LIPIcs.ESA.2019.9

Acknowledgements This work benefited from discussions with Alain Sarlette, Stacey Jeffery, Anthony Leverrier, Ronald de Wolf, André Chailloux and Frédéric Magniez. Part of this work was supported by the CWI-Inria International Lab.

1 Introduction and Summary

Sampling from the stationary distribution of a random walk is a common and valuable tool in the design of algorithms [32]. It underlies the Markov chain Monte Carlo paradigm, and plays a central role in a wide range of approximation algorithms for graph problems. In this work we investigate the quantum counterpart of this task - generating quantum samples from the random walk stationary distribution. Given query access to some graph $G = (\mathcal{V}, \mathcal{E})$ with m edges, we wish to create the quantum state

$$|\pi\rangle = \frac{1}{\sqrt{m}} \sum_{(i,j)\in\mathcal{E}} |i,j\rangle\,, \tag{1}$$

which is a superposition over the edges of the graph. Measuring the first register of this state, and discarding the second register, indeed returns the random walk stationary distribution. Creating such a quantum sample of a classical stationary distribution forms a crucial primitive for a range of algorithms: the so-called "setup cost" in quantum walk search algorithms [28, 25] refers to the cost of generating a state such as $|\pi\rangle$, quantum algorithms for speeding up MCMC [2, 33, 39, 30] build on the possibility of efficiently creating quantum samples, and a number of quantum algorithms for solving graph problems [38, 23] require the generation of a superposition over the edges of a graph.

We develop a new quantum algorithm for creating the quantum sample (1), given only local query access to the graph. Our algorithm improves the query and time complexity of the folklore approach to quantum sampling from $\widetilde{O}(m^{1/2}\delta^{-1/2})$ to $\widetilde{O}(m^{1/3}\delta^{-1/3})$. We do so by growing a classical seed set from the initial node. This incurs a payoff in the space complexity, increasing it from $\widetilde{O}(1)$ to $\widetilde{O}(m^{1/3}\delta^{-1/3})$. As a demonstration of our algorithm,

we discuss a new approach to solving *st*-connectivity: generate a superposition over the connected components of s and t, and compare these states. This approach generalizes the notorious quantum state generation strategy for solving graph isomorphism. Concerning the latter, we show that our algorithm allows to create a superposition over the isomorphisms of a given n-vertex input graph in $\widetilde{O}(2^{n/3})$ steps. This surpasses the $\Omega(2^{n/2})$ index erasure barrier by Ambainis et al [6]. In a similar way we can create a superposition over the elements of a black box group in $\widetilde{O}(2^{n/3})$ steps, where 2^n is the number of group elements.

1.1 Query Model

We assume throughout this work that we only have "local" query access to the graph $G = (\mathcal{V}, \mathcal{E})$: we are give an initial node $j \in \mathcal{V}$, and we can query for its degree and neighbors. Such queries fall under the so-called *adjacency array model* [20] or *bounded degree model* [21] (although we do not assume the degree is bounded), which is very natural when studying random walk algorithms. However, departing from these models, and justifying the term "local", we will not assume direct access to or prior knowledge about $\mathcal{V}$, apart from the initial node. For comparison, in [20] the node set $\mathcal{V}$ is given as a list, and in [21] access to uniformly random nodes is assumed. In this sense our work is more in line with graph exploration algorithms as considered e.g. in [34], or more recently in [17].

Since our algorithm strongly builds on the use of quantum walks, we will alternatively express the complexity of our results as a function of the number of quantum walk steps. Also in such case the denominator "local" query access is justified, since a single quantum walk step from a certain node only accesses the neighbors of that node.

1.2 Quantum Walk Sampling Algorithm

Our algorithm builds on the folklore approach to creating $|\pi\rangle$, discussed in e.g. [31, 39, 30, 29]. Starting from some initial state $|j\rangle$ localized on a node $j \in \mathcal{V}$, this approach combines quantum phase estimation and amplitude amplification on the quantum walk operator associated to the graph. We detail this scheme in Section 2.2. The scheme requires $\widetilde{O}(m^{1/2}\delta^{-1/2})$ QW steps on the graph, where δ is the random walk spectral gap, and the factor $m^{1/2}$ stems from the small projection of the initial state onto $|\pi\rangle$.

In the present work we improve on this scheme by initially doing some "classical work": we first use classical means to grow a seed set around the initial vertex. Briefly ignoring the δ-dependency, we grow the set to have size $\Theta(m^{1/3})$. We can then use a special data structure to generate and reflect around a quantum superposition over this set, which now has a $\Omega(m^{-1/3})$ overlap with the target state. Reinvoking the folklore scheme from this state then allows to retrieve $|\pi\rangle$, now only requiring $\widetilde{O}(m^{1/3})$ queries. This approach leads to the following result.

▶ **Theorem 1.** *Given a lower bound $\gamma \leq \delta$ on the spectral gap, it is possible to create the quantum state $|\pi\rangle$ using $\widetilde{O}(m^{1/3}\gamma^{-1/3})$ time, space and QW steps.*

Apart from the log-factors, the combined dependency on m and δ is optimal. Indeed it is tight on e.g. the cycle graph, which has $m = n$ and $\delta = n^{-2}$, giving an $\widetilde{O}(n)$ steps algorithm. Since the diameter of the cycle is $\Omega(n)$, this is optimal when assuming local query access. We also note that, if in addition we are given a bound $D \geq d_M$ on the maximum degree (in e.g. the array model this is always given), then we can implement our algorithm using $\widetilde{O}(m^{1/3}\gamma^{-1/3}D^{1/3})$ degree and neighbor queries.

The algorithm gives a direct bound on the so-called setup cost of quantum walk search algorithms in the MNRS framework [28] as a function of the update cost (i.e., the cost of implementing a quantum walk step). The increased space complexity of our algorithm, $\widetilde{O}(m^{1/3}\delta^{-1/3})$ as compared to $\widetilde{O}(1)$ for the folklore approach, is very similar to the payoff in space versus time or query complexity in the collision finding algorithm of Brassard et al [15] and the element distinctness algorithm of Ambainis [4].

1.3 Application to *st*-connectivity

Our QW sampling algorithm yields a new approach for solving *st*-connectivity, somewhat similar to the approach taken by Watrous in [38]: generate a superposition over the edges in the connected components of s resp. t, and compare the resulting states. As we prove in Proposition 9, this requires $\widetilde{O}(m^{1/3}\gamma^{-1/3})$ QW steps, where γ is a lower bound on the spectral gaps of the connected components of s and t. Our algorithm outperforms the existing quantum algorithms for *st*-connectivity [20, 13, 12, 23] on for instance sparse graphs with a good spectral gap.

The approach generalizes a well-known strategy to solving graph isomorphism on a quantum computer [2] (called "component mixing" in [27]): generate superpositions over the isomorphisms of each graph, and compare the resulting states. In [6], Ambainis et al aimed to prove a lower bound on this approach by abstracting it to the so-called index erasure problem. For this generalized problem, they prove a lower bound of $\Omega(2^{n/2})$. They argue that the same bound holds for creating a superposition over graph isomorphisms, be it under the condition that the algorithm makes no use of the structure of the problem. We show that, by exploiting the structure of the problem, we can indeed use our quantum walk sampling algorithm to surpass this bound. Thereto we consider the graph whose node set consists of isomorphisms of the input graph, and whose edge set arises from performing pairwise transpositions on the nodes (i.e., on the adjacency matrices of the isomorphisms). Using our quantum walk sampling algorithm on this graph yields the following corollary.

▶ **Corollary 2.** *Given an n-node input graph g, it is possible to create a superposition over the isomorphisms of g in $\widetilde{O}(2^{n/3})$ steps.*

Completing the associated *st*-connectivity algorithm, we find an $\widetilde{O}(2^{n/3})$ quantum algorithm for graph isomorphism. Using the existing quantum algorithms for *st*-connectivity, this approach would require $\Omega(2^{n/2})$ steps. Clearly the improved performance still falls terribly short of current (classical) algorithms for graph isomorphism, most notably the quasi-polynomial algorithm by Babai [10], yet it provides a clear demonstration of how the readily accessible structure of the problem allows to surpass the index erasure bound.

A similar strategy exists for solving the group non-membership problem on a quantum computer, as proposed by Watrous [37], requiring to generate a superposition over the elements of a finite black box group. Using the random walk algorithm by Babai [9] for generating uniformly random group elements, we can similarly generate this superposition in $\widetilde{O}(2^{n/3})$ steps, when 2^n is the number of group elements.

1.4 Open Questions

This work leaves open a number of questions and possible applications, some of which we summarize below:

- *Quantum sampling for general Markov chains or stoquastic Hamiltonians.* In this work we only consider the quantum sampling problem for random walks. Generalizing our approach to more general Markov chains could lead to improvements on quantum MCMC

algorithms [2, 33], or the preparation of many body ground states [30] and Gibbs states [36]. The main bottleneck to such generalization seems to be the classical construction of seed sets which have an appropriate overlap with the goal quantum state. Even more generally, one could consider the preparation of ground states of Hamiltonians. For e.g. the special case of stoquastic Hamiltonians, which are known to have a nonnegative ground state, it should be possible to construct a seed set with improved overlap with the ground state.

- *Faster quantum fast-forwarding.* In former work by the author [8], a quantum algorithm was proposed for quantum sampling a t-step Markov chain. If the Markov chain has transition matrix P, and is started from a node i, the algorithm has complexity $\widetilde{O}(\|P^t\,|i\rangle\,\|^{-1}\,t^{1/2}) \in \widetilde{O}(m^{1/2}\,t^{1/2})$. Using ideas from the present work, it seems very feasible that we can improve this complexity to $\widetilde{O}(\|P^t\,|i\rangle\,\|^{-2/3}\,t^{1/2}) \in \widetilde{O}(m^{1/3}\,t^{1/2})$. Rather than using a breadth-first search to grow the seed set, as in the present work, it seems more suitable to use random walk techniques as in [34, 7]. As a byproduct, this would yield an improved quantum expansion tester, combining the speedups of [5] and [8].
- *Quantum search in $\sqrt{HT}$.* Our algorithm does not suffer from the so-called "symmetry barrier" in quantum algorithms: we can go from $|j\rangle$ to $|\pi\rangle$ more easily then from $|\pi\rangle$ to $|j\rangle$. Indeed, if for instance the underlying graph is an expander, then the former takes $O(n^{1/3})$ queries, whereas the latter takes $\Omega(n^{1/2})$ queries by the search lower bound.
  An open problem related to this is the following: given an initial node s in a graph, can we find a node t in $O(HT_{s,t}^{1/2})$ QW steps, with $HT_{s,t}$ the hitting time from s to t? Currently the best algorithm for this problem is by Belovs [12], which solves it in $O(CT_{s,t}^{1/2})$, with $CT_{s,t} = H_{s,t} + H_{t,s}$ the commute time between s and t. Since the commute time is symmetric between s and t, this obeys the aforementioned symmetry barrier. However, the commute time can be much larger than the hitting time from s to t, hence the open question of whether we can improve this performance to $O(HT_{s,t}^{1/2})$, thereby necessarily breaking this symmetry e.g. by using our techniques.

1.5 Outline

In Section 2 we discuss the graph and query model (Section 2.1), and provide the necessarily preliminaries on random walks and quantum walks (Section 2.2). In Section 3 we propose an algorithm for growing a classical seed set (Section 3.1), we discuss the data structure (Section 3.2), and we propose our QW sampling algorithm (Section 3.3). Finally in Section 4 we discuss the application of our QW sampling algorithm for solving st-connectivity (Section 4.1), and we demonstrate it for the special case of graph isomorphism testing (Section 4.2).

2 Preliminaries: Queries and Walks

2.1 Graph and Query Model

Throughout the paper we assume local query access to an undirected graph $G = (\mathcal{V}, \mathcal{E})$, with $\mathcal{E}$ a subset of the ordered pairs $\mathcal{V} \times \mathcal{V}$, such that $(i,j) \in \mathcal{E} \Leftrightarrow (j,i) \in \mathcal{E}$. We denote $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. For any $\mathcal{S} \subseteq \mathcal{V}$, we let $\mathcal{E}(\mathcal{S})$ denote the set of edges starting in $\mathcal{S}$, i.e.,

$$\mathcal{E}(\mathcal{S}) = \{(i,j) \in \mathcal{E} \mid i \in \mathcal{S}\}.$$

For any $i \in \mathcal{V}$, we let $d(i) = |\mathcal{E}(\{i\})|$ denote the degree of i, the maximum degree $d_M = \max_{i\in\mathcal{V}} d(i)$, and $d(\mathcal{S}) = |\mathcal{E}(\mathcal{S})| = \sum_{i\in\mathcal{S}} d(i)$ denotes the total degree of a set $\mathcal{S} \subseteq \mathcal{V}$. A single query consists of either of the following:

- *degree query*: given $i \in \mathcal{V}$, return degree $d(i)$
- *neighbor query:* given $i \in \mathcal{V}$, $k \in [d(i)]$, return k-th neighbor of i

As an alternative query model we will also consider the quantum walk model, or so-called MNRS framework, as proposed in [28] in the context of quantum walk search. The model associates abstract costs to different operations[1]:

- *setup cost:* the cost of preparing the quantum sample $|\pi\rangle = m^{-1/2} \sum_{(i,j)\in\mathcal{E}} |i,j\rangle$
- *update cost:* the cost of implementing a quantum walk step. See Section 2.2 for details.

For search problems an additional *checking cost* is considered, yet this will not be relevant here. In [16] it is proven that the update cost or quantum walk step for a node i can be simulated using $O(d(i)^{1/2})$ degree and neighbor queries. From our work it follows that the setup cost can be simulated using $\widetilde{O}(m^{1/3}\delta^{-1/3})$ QW steps, or $\widetilde{O}(m^{1/3}d_M^{1/3}\delta^{-1/3})$ degree and neighbor queries.

2.2 Random Walks and Quantum Walks

From some initial seed vertex $j \in \mathcal{V}$, we can use degree and neighbor queries to implement a random walk over $\mathcal{V}$. The transition matrix P describing such a walk is defined by $P(i,j) = 1/d(i)$ if $(i,j) \in \mathcal{E}$, and $P(i,j) = 0$ elsewhere. If the graph is connected and nonbipartite, then the random walk converges to its stationary distribution π, defined by $\pi(i) = d(i)/m$ for any $i \in \mathcal{V}$. If we order the eigenvalues of P (with multiplicities) as $1 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n \geq -1$, then the rate at which the walk converges to π is bounded by the spectral gap $\delta = 1 - \max\{|\lambda_2|, |\lambda_n|\}$ [26].

Quantum walks (QWs) form an elegant quantum counterpart to random walks on graphs. Following the exposition in [28], they are naturally defined over a vector space associated to the edge set

$$\mathcal{H}_{\mathcal{E}} = \mathrm{span}_{\mathbb{C}}\{\, |i,j\rangle \mid (i,j) \in \mathcal{E}\}.$$

A quantum walk over $\mathcal{H}_{\mathcal{E}}$ is now defined as the unitary operator $W = SR_{\mathcal{E}}$, where $R_{\mathcal{E}}$ is a reflection around the subspace $\mathrm{span}_{\mathbb{C}}\{|\psi_i\rangle \mid i \in \mathcal{V}\}$, with

$$|\psi_i\rangle = \frac{1}{\sqrt{d(i)}} \sum_{(i,j)\in\mathcal{E}} |i,j\rangle\,,$$

and S represents the swap operator defined by $S\,|i,j\rangle = |j,i\rangle$. The cost of implementing the QW operator W is called the update cost, but can alternatively be implemented using $O(d_M^{1/2})$ degree and neighbor queries, and $\widetilde{O}(1)$ elementary operations.

The spectrum of W is carefully tied to the spectrum of the original random walk matrix P, as was seminally proven by Szegedy in [35] and Magniez et al in [28]. For the purpose of this work, we abstract the following lemma. We say that W has a phase gap Δ if for every eigenvalue $e^{i\theta} \neq 1$ of W it holds that $|\theta| \geq \Delta$. We also recall the state $|\pi\rangle = m^{-1/2} \sum_{(i,j)\in\mathcal{E}} |i,j\rangle$.

▶ **Lemma 3** ([35, 28]). *Let P be the random walk transition matrix having spectral gap δ. Then the quantum walk operator W has a phase gap $\Delta \in \Theta(\sqrt{\delta})$, and $|\pi\rangle$ is the unique eigenvalue-1 eigenvector in the subspace* $\mathrm{span}_{\mathbb{C}}\{|\psi_i\rangle \mid i \in \mathcal{V}\}$.

From this lemma, combined with the quantum algorithms for phase estimation and amplitude amplification, we can derive the folklore approach to quantum walk sampling, discussed in for instance [31, 39, 30, 29]. Since we will use it as a subroutine, we summarize it below. For a general subset $\mathcal{S} \subseteq \mathcal{V}$, we denote the state $|\mathcal{S}\rangle = d(\mathcal{S})^{-1/2} \sum_{(i,j)\in\mathcal{E}(\mathcal{S})} |i,j\rangle$.

[1] They actually consider a more general model, associated to a reversible Markov chain over G. We consider the special case where the Markov chain is a random walk.

▶ **Proposition 4.** *Given an initial set $\mathcal{S} \subseteq \mathcal{V}$ and a lower bound $\gamma \leq \delta$, there exists a quantum routine that generates a state ϵ-close to $|\pi\rangle$. The routine finishes and outputs a success flag after an expected number of $O(d(\mathcal{S})^{-1/2} m^{1/2} \gamma^{-1/2} \log \epsilon^{-1})$ calls to W, $O(d(\mathcal{S})^{-1/2} m^{1/2})$ reflections around $|\mathcal{S}\rangle$, and requires an additional $O(\log \epsilon^{-1} \log^2 \gamma^{-1})$ time and space complexity.*

Proof. Let the operator U be defined by the amplified quantum phase estimation algorithm, as used in [28, Theorem 6]. For some integer k, this operator maps an initial state $|\mathcal{S}\rangle$ to the state

$$U |\mathcal{S}\rangle |0\rangle = \langle \pi | \mathcal{S} \rangle |\pi\rangle |0\rangle + |\Gamma\rangle ,$$

where $|\Gamma\rangle$ is such that $\| (\mathbb{I} \otimes |0\rangle \langle 0|) |\Gamma\rangle \| \leq 2^{-k}$. The operator U can be implemented using $O(k\Delta^{-1}) \in O(k\gamma^{-1/2})$ calls to W and $W^\dagger$, and $O(k \log^2 \gamma^{-1})$ additional space and elementary gates.

On this state we can invoke the amplitude amplification scheme from [14, Theorem 3] to retrieve the projection of $U |\mathcal{S}\rangle |0\rangle$ on the image of $\mathbb{I} \otimes |0\rangle \langle 0|$, which is 2^{-k}-close to $|\pi\rangle$. This requires an expected number of $\Theta(|\langle \mathcal{S} | \pi \rangle|^{-1})$ calls to U, $U^\dagger$ and the reflection operator $\mathbb{I} \otimes (2 |0\rangle \langle 0| - \mathbb{I})$. We prove the proposition by choosing $k \in \Theta(\log \epsilon^{-1})$ and noting that $|\langle \mathcal{S} | \pi \rangle| = d(\mathcal{S})^{1/2} m^{-1/2}$. ◀

On a general graph, and starting from some initial node $\mathcal{S} = \{i\}$, this scheme requires $\widetilde{O}(d(i)^{-1/2} m^{1/2} \gamma^{-1/2}) \in \widetilde{O}(m^{1/2} \gamma^{-1/2})$ QW steps, or $\widetilde{O}(m^{1/2} d_M^{1/2} \gamma^{-1/2})$ degree and neighbor queries.

3 Quantum Walk Sampling

In this section we elaborate our scheme for quantum walk sampling. We separately address the process for growing a seed set, the data structure that we require, and their combination with the folklore QW sampling routine.

3.1 Growing a Seed Set

We propose the below Algorithm 1 to grow a seed set in the graph. It is a variation on the breadth-first search algorithm, returning an edge set of given size.

▶ **Lemma 5.** *If $M \leq m$, then Algorithm 1 outputs a subset $E \subseteq \mathcal{E}$ with $|E| \geq M$. Its time and space complexity, and degree and neighbor query complexity, are $\widetilde{O}(M)$.*

Proof. Assuming the lists are ordered, any of the list and queue operations (enqueueing, dequeueing, adding an element, outputting the size of a list, searching an element in a list) takes polylogarithmic time. As a consequence, the time complexity will be determined up to log-factors by the number of for-loops before the algorithm terminates.

In every for-loop an edge is considered. Since the edges are directed, every edge is encountered at most once, at which point it is added to E. Since the algorithm terminates when $|E| = M$, this implies that the algorithm terminates after less than M for-loops. ◀

Alternatively we can output the node set $\mathcal{S} \subseteq \mathcal{V}$. Since $E \subseteq \mathcal{E}(\mathcal{S})$, we have that $d(\mathcal{S}) \geq M$.

Algorithm 1 Breadth-First Edge Search.

Input: initial node i and query access to a connected graph G, integer M
Do:

create lists $S = \emptyset$ and $E = \emptyset$, and queue $B = (i)$
while $B \neq \emptyset$ **do**
  $i \leftarrow \texttt{dequeue}(B)$, $\texttt{add}(S \leftarrow i)$
  for all j s.t. $(i, j) \in \mathcal{E}$ **do**
    if $j \notin S$ **then**
      $\texttt{add}(E \leftarrow \{(i, j), (j, i)\})$
      if $|E| \geq M$ **then** terminate and output E
    if $j \notin B$ **then** $\texttt{enqueue}(B \leftarrow j)$

3.2 Kerenidis-Prakash Data Structure

After growing the seed set $\mathcal{S} \subseteq \mathcal{V}$, we wish to use it as a resource for our QW sampling algorithm. Specifically we will require the generation of and reflection around the superposition $|\mathcal{S}\rangle$ over edges starting in $\mathcal{S}$. By naive query access to the database containing $\mathcal{S}$, this requires a time complexity $\Omega(d(\mathcal{S})^{1/2})$ per generation or reflection, which follows from the bound on index erasure [6]. Since our QW sampling algorithm will require $\Omega(m^{1/3})$ such operations, the total time complexity for $d(\mathcal{S}) \in \Theta(m^{1/3})$ would become $\Omega(m^{1/2})$, thus providing no speedup on the time complexity as compared to the folklore approach. To remedy this, we use a more efficient data structure proposed by Kerenidis and Prakash [24] in their quantum recommendation algorithm. We extract the following result, abstracted from their Theorem 15 (by setting $\mathtt{m} = 1$, $\mathtt{n} = n^2$ and inputting entries $(1, (i, j), 1)$ for all $(i, j) \in \mathcal{S}$).

▶ **Theorem 6** (Kerenidis-Prakash [24]). *Assume we have query access to a set $\mathcal{S} \subseteq \mathcal{V}$. There exists a classical data structure to store the set $\mathcal{S}$ with the following properties:*

- *the size of the structure is $O(|\mathcal{S}| \log^2(m))$,*
- *the time and query complexity to fill the structure is $O(|\mathcal{S}| \log^2(m))$,*
- *having quantum access to the data structure we can perform the mapping $U : |0\rangle \to |\mathcal{S}\rangle$ and its inverse $U^\dagger$ in time* $\text{polylog}(m)$.

This easily implies the ability to reflect around $|\mathcal{S}\rangle$ in time $\text{polylog}(m)$: we can rewrite the reflection $2 |\mathcal{S}\rangle \langle\mathcal{S}| - \mathbb{I} = U(2 |0\rangle \langle 0| - \mathbb{I})U^\dagger$, so that it comes down to implementing U, $U^\dagger$ and a reflection around the basis state $|0\rangle$.

3.3 QW Sampling Algorithm

Building on the seed set and data structure, we can now propose our quantum sampling algorithm for creating the state $|\pi\rangle$ in $\widetilde{O}(m^{1/3}\delta^{-1/3})$ time, space and quantum walk steps.

▶ **Theorem 7** (Quantum Walk Sampling). *If we choose $\gamma \leq \delta$ then Algorithm 2 returns a state ϵ-close to $|\pi\rangle$. The algorithm requires expected space, time and quantum walk steps in*

$$\widetilde{O}(m^{1/3}\gamma^{-1/3} \log \epsilon^{-1}).$$

Proof. The correctness of the algorithm follows from Proposition 4. By this proposition we know that if $\gamma \leq \delta$ and the algorithm terminates, and hence the routine from Proposition 4 finished, then it effectively outputs a state ϵ-close to $|\pi\rangle$. The complexity of the algorithm for a fixed M is also easily bounded: the complexity of steps 2 and 3 is both $\widetilde{O}(M^{1/3}\gamma^{-1/3})$,

Algorithm 2 Quantum Walk Sampling.

Input: parameters γ and ϵ; initial node i and query access to a graph G
Do:

1: **for** $M = 1, 2, 4, \ldots, 2^k, \ldots$ **do**
2: use BFS to grow a seed set $\mathcal{S}$ with $d(\mathcal{S}) \in \Omega(M^{1/3}\gamma^{-1/3})$
3: load $\mathcal{S}$ in data structure
4: apply the routine from Proposition 4 on $|\mathcal{S}\rangle$
if the routine finishes after $\widetilde{O}(M^{1/3}\gamma^{-1/3}\log\epsilon^{-1})$ steps
then terminate algorithm and return its output
else abort the routine and continue for-loop

which follows from Lemma 5 resp. Theorem 6. Step 4 is automatically terminated after $\widetilde{O}(M^{1/3}\gamma^{-1/3}\log\epsilon^{-1})$ steps, which by Proposition 4 directly bounds the number of calls to W and reflections around $|\mathcal{S}\rangle$. By Theorem 6 the complexity of implementing a single reflection around $|\mathcal{S}\rangle$ is $\widetilde{O}(1)$. The total complexity for a fixed M is therefore $\widetilde{O}(M^{1/3}\gamma^{-1/3}\log\epsilon^{-1})$.

What remains to bound is the M-value at which the algorithm terminates. From Proposition 4 we know that if the number of steps $M^{1/3}\gamma^{-1/3}\log\epsilon^{-1}$ is sufficiently large, i.e.,

$$M^{1/3}\gamma^{-1/3}\log\epsilon^{-1} \in \Omega(|\langle\pi|\mathcal{S}\rangle|^{-1}\gamma^{-1/2}\log\epsilon^{-1}), \tag{2}$$

then the routine finishes with probability $\Omega(1)$. From the fact that $|\pi\rangle = m^{-1/2}\sum_{(i,j)\in\mathcal{E}}|i,j\rangle$ and $d(\mathcal{S}) \in \Omega(M^{1/3}\gamma^{-1/3})$ it holds that $|\langle\pi|\mathcal{S}\rangle| \in \Omega(M^{1/6}\gamma^{-1/6}m^{-1/2})$. As a consequence, if $M \geq m$ then $|\langle\pi|\mathcal{S}\rangle| \in \Omega(m^{-1/3}\gamma^{-1/6})$ and hence (2) will hold, such that the routine will finish with probability $\Omega(1)$. The expected number of for-loops is therefore $\log m + O(1)$, with the total complexity scaling as

$$\widetilde{O}\left(\gamma^{-1/3}\log\epsilon^{-1}\sum_{k=0}^{\log m+O(1)}2^{k/3}\right) \in \widetilde{O}(m^{1/3}\gamma^{-1/3}\log\epsilon^{-1}). \qquad \blacktriangleleft$$

Alternatively we are interested in bounding the algorithm in terms of classical queries. We can naively substitute every quantum walk step for $\widetilde{O}(\sqrt{d_M})$ degree and neighbor queries, yielding a complexity $\widetilde{O}(m^{1/3}d_M^{1/2}\gamma^{-1/3})$. However, if we are given an upper bound $D \geq d_M$, we can improve this complexity by slightly increasing the size of the seed set. We note that in the array model [20] the degrees are assumed to be known beforehand, so we exactly know d_M.

▶ **Corollary 8.** *Given an initial node i, a lower bound $\gamma \leq \delta$ and an upper bound $D \geq d_M$, we can generate a state ϵ-close to $|\pi\rangle$ in expected space, time, and degree and neighbor queries in*

$$\widetilde{O}(m^{1/3}D^{1/3}\gamma^{-1/3}\log\epsilon^{-1}).$$

Proof. We adapt Algorithm 2 as follows: we slightly increase the size of the seed set in step 2 to $\Omega(M^{1/3}D^{1/3}\gamma^{-1/2})$, and limit the number of steps in step 4 to $\widetilde{O}(M^{1/3}D^{-1/6}\gamma^{-1/3}\log\epsilon^{-1})$. Following the proof of Theorem 7, the algorithm then terminates after

$$\widetilde{O}(m^{1/3}D^{1/3}\gamma^{-1/3}\log\epsilon^{-1})$$

classical steps and queries, and $\widetilde{O}(m^{1/3}D^{-1/6}\gamma^{-1/3})$ QW steps. Now we can substitute each QW step with $\widetilde{O}(\sqrt{d_M})$ degree and neighbor queries, yielding the claimed complexity. $\blacktriangleleft$

4 Application: st-Connectivity

4.1 General Algorithm

Let $\delta^{(s)}$ and $\delta^{(t)}$ denote the spectral gaps of the connected components of s resp. t.

▶ **Proposition 9.** *Given $s,t \in \mathcal{V}$ and a lower bound $\gamma \leq \delta^{(s)}, \delta^{(t)}$, we can decide st-connectivity with probability $1-\epsilon$ in $\widetilde{O}(m^{1/3}\gamma^{-1/3}\log\epsilon^{-1})$ QW steps. If we are also given an upper bound $D \geq d_M$, then we can do so in $\widetilde{O}(m^{1/3}D^{1/3}\gamma^{-1/3}\log\epsilon^{-1})$ degree and neighbor queries.*

Proof. Given γ we can create an $(\epsilon' = 1/4)$-approximation $|\psi_s\rangle$ (resp. $|\psi_t\rangle$) of the superposition $|\pi^{(s)}\rangle$ (resp. $|\pi^{(t)}\rangle$) over the edges of the connected component of s (resp. t) in $\widetilde{O}(m^{1/3}\gamma^{-1/3})$ QW steps. If we also have D, then we can do so in $\widetilde{O}(m^{1/3}d_M^{1/3}\gamma^{-1/3})$ degree and neighbor queries.

If s and t are connected, then $|\langle\psi_s|\psi_t\rangle| \geq 1-\epsilon'^2$, whereas if they are not, then $|\langle\psi_s|\psi_t\rangle| \leq 2\epsilon'$. We can distinguish these cases by performing the SWAP-test [2] between these states, using a single copy of both states, and $O(1)$ additional gates. If s and t are connected, then the test returns 1 with probability $(1-|\langle\psi_s|\psi_t\rangle|)/2 \leq \epsilon'^2/2 = 1/32$, if s and t are not connected, the test returns 1 with probability $(1-|\langle\psi_s|\psi_t\rangle|)/2 \geq 1/2-\epsilon' = 1/4$. Repeating this scheme $O(\log\epsilon^{-1})$ times then allows to decide st-connectivity with probability $1-\epsilon$. ◀

This approach best compares to the following classical scheme: use $\widetilde{\Theta}(n^{1/2})$ independent random walks of length $\Theta(\gamma^{-1})$ from s and t to gather samples from the stationary distributions on the connected components of s resp. t. If s and t are connected then with constant probability the sample sets will overlap, which follows from the birthday paradox. This scheme requires $\widetilde{O}(n^{1/2}\gamma^{-1})$ random walk steps, or equivalently, neighbor queries. It lies at the basis of the graph expansion tester by Goldreich and Ron [22], and the subsequent work on testing closeness of distributions [11] and clusterability of graphs [18].

In Table 1 we compare the query complexity of our approach to the existing quantum algorithms for st-connectivity. If no promise is given on negative instances (such as in [23] in the form of a capacitance $C_{s,t}$), then all former algorithms require $\Omega(n^{1/2})$ queries when maximized over all (s,t)-pairs of the graph. As a consequence, for the graph isomorphism problem treated in the next section, they all have a $\Omega(2^{n/2})$ complexity. Our approach however has a $\widetilde{O}(2^{n/3})$ complexity.

4.2 Graph Isomorphism

We consider some given n-node graph g, described by its adjacency matrix. To this graph we can associate a new regular graph $G^{(g)} = (\mathcal{V},\mathcal{E})$ with nodes $\mathcal{V} = \{\sigma(g) \mid \sigma \in S_n\}$, consisting of permutations of the original graph nodes, and edges $\mathcal{E} = \{(h,\sigma_{i,j}(h)) \mid h \in \mathcal{V}, i,j \in [n]\}$, corresponding to all possible transpositions of two elements. We can easily prove the following.

▶ **Lemma 10.** *The random walk on $G^{(g)}$ has a spectral gap $\delta \in \Omega(n^{-1}\log^{-1}n)$.*

Proof. If $|\mathcal{V}| = n!$ (i.e., $g \neq \sigma(g)$ if $\sigma \neq 1$), this graph is isomorphic to the Cayley graph derived from the symmetric group with generators given by transpositions. The mixing time of a random walk on this graph is $O(n\log n)$ by a result of Diaconis and Shashahani [19], implying a lower bound on its spectral gap $\delta \in \Omega(n^{-1}\log^{-1}n)$.

If $|\mathcal{V}| < n!$, the graph is effectively an edge contraction of the random transposition graph. Following Aldous and Fill [3, Proposition 4.44], a random walk on this graph is an *induced chain* of the random walk on the symmetric group, in particular having a spectral gap lower bounded by the spectral gap of the original walk. ◀

Table 1 Query complexity of *st*-connectivity using different quantum algorithms in different models. The array model measures the number of degree and neighbor queries; the adjacency model measures the number of pair queries (e.g., "are i and j neighbors?"); the QW model measures the number of QW steps. The quantities $d_{s,t}$ and $R_{s,t}$ denote the length of the shortest path and the effective resistance, respectively, between s and t. The quantity $C_{s,t}$ denotes the capacitance between s and t in negative instances, i.e., if s and t are disconnected then $C_{s,t}$ quantifies "how" disconnected they are.

	query complexity	model
Dürr et al [20]	$\Theta(n)$	array
Dürr et al [20]	$\Theta(n^{3/2})$	adjacency
Belovs-Reichardt [13]	$O(m^{1/2}\, d_{s,t}^{1/2})$	adjacency
Belovs [12]	$O(m^{1/2} R_{s,t}^{1/2}) \in O(m^{1/2}\, \delta^{-1/2})$	QW
Jarret et al [23]	$O(R_{s,t}^{1/2}\, C_{s,t}^{1/2})$	adjacency
folklore QW sampling	$O(m^{1/2}\, \delta^{-1/2})$	QW
this work	$\widetilde{O}(m^{1/3}\, \delta^{-1/3})$	QW
this work	$\widetilde{O}(m^{1/3}\, \delta^{-1/3}\, d_M^{1/3})$	array

Next we show how to implement a QW step on $G^{(g)}$ in $\widetilde{O}(1)$ steps. By Theorem 7 we can then create a superposition over the edges of $G^{(g)}$ (or, equivalently, its nodes) in time $\widetilde{O}(m^{1/3}) = \widetilde{O}(2^{n/3})$, and by Proposition 9 we can solve *st*-connectivity (i.e. graph isomorphism) in the same time.

▶ **Lemma 11.** *Implementing a quantum walk on $G^{(g)}$ takes time $\widetilde{O}(1)$.*

Proof. Since we may have multi-edges, corresponding to permutations that leave the input graph invariant, we will slightly alter the QW to take place on a node+coin space (as in e.g. [1, 4]) rather than on the edge space. The relevant spectral properties from Lemma 3 however remain unchanged, as is easily seen by following for instance the proof of [25]. We define the QW node+coin space, associated to the input graph g, as $\mathrm{span}_{\mathbb{C}}\{|\sigma(g), i, j\rangle \mid \sigma \in S_n, i, j \in [n]\}$, with S_n the symmetric group of permutations. Similarly to Section 2.2, the QW operator $W = SR_{\mathcal{E}}$ consists of a reflection $R_{\mathcal{E}}$ around a subspace $\mathrm{span}_{\mathbb{C}}\{|\psi_{\sigma(g)}\rangle \mid \sigma \in S_n\}$, now defined as

$$|\psi_{\sigma(g)}\rangle = \frac{1}{n} \sum_{i,j \in [n]} |g, i, j\rangle\,,$$

and the shift operator S defined by $S\,|g', i, j\rangle = |\sigma_{i,j}(g'), i, j\rangle$. Each of these operators can be implemented in $\widetilde{O}(1)$ steps. ◀

References

1 Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC)*, pages 50–59. ACM, 2001. arXiv:quant-ph/0012090

2 Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the 35th ACM Symposium on Theory of Computing (STOC)*, pages 20–29. ACM, 2003. arXiv:quant-ph/0301023

3 David Aldous and Jim Fill. Reversible Markov chains and random walks on graphs. Unfinished monograph, 2002. URL: https://www.stat.berkeley.edu/~aldous/RWG/book.pdf.

4 Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. arXiv:quant-ph/0311001

5 Andris Ambainis, Andrew M Childs, and Yi-Kai Liu. Quantum property testing for bounded-degree graphs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 365–376. Springer, 2011. arXiv:1012.3174

6 Andris Ambainis, Loïck Magnin, Martin Roetteler, and Jérémie Roland. Symmetry-assisted adversaries for quantum state generation. In *Proceedings of the 26th IEEE Conference on Computational Complexity (CCC)*, pages 167–177. IEEE, 2011. arXiv:1012.2112

7 Reid Andersen and Yuval Peres. Finding sparse cuts locally using evolving sets. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, pages 235–244. ACM, 2009. arXiv:0811.3779

8 Simon Apers and Alain Sarlette. Quantum Fast-Forwarding Markov Chains and Property Testing. *Quantum Information and Computation*, 19(3&4):181–213, 2019. arXiv:1804.02321

9 László Babai. Local expansion of vertex-transitive graphs and random generation in finite groups. In *Proceedings of the 23rd ACM Symposium on Theory of Computing (STOC)*, volume 91, pages 164–174. ACM, 1991. doi:10.1145/103418.103440.

10 László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the 48th ACM Symposium on Theory of Computing (STOC)*, pages 684–697. ACM, 2016. arXiv:1512.03547

11 Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D Smith, and Patrick White. Testing closeness of discrete distributions. *Journal of the ACM*, 60(1):4, 2013. arXiv:1009.5397

12 Aleksandrs Belovs. Quantum walks and electric networks. arXiv:1302.3143, 2013.

13 Aleksandrs Belovs and Ben W Reichardt. Span programs and quantum algorithms for st-connectivity and claw detection. In *Proceedings of the 20th European Symposium on Algorithms (ESA)*, pages 193–204. Springer, 2012. arXiv:1203.2603

14 Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002. arXiv:quant-ph/0005055

15 Gilles Brassard, Peter Høyer, and Alain Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News (Cryptology Column)*, 28:14–19, 1997. arXiv:quant-ph/9705002

16 Chris Cade, Ashley Montanaro, and Aleksandrs Belovs. Time and space efficient quantum algorithms for detecting cycles and testing bipartiteness. arXiv:1610.00581, 2016.

17 Flavio Chiericetti, Anirban Dasgupta, Ravi Kumar, Silvio Lattanzi, and Tamás Sarlós. On sampling nodes in a network. In *Proceedings of the 25th International Conference on World Wide Web (WWW)*, pages 471–481. International WWW Conferences, 2016. doi:10.1145/2872427.2883045.

18 Artur Czumaj, Pan Peng, and Christian Sohler. Testing cluster structure of graphs. In *Proceedings of the 47th ACM Symposium on Theory of Computing (STOC)*, pages 723–732. ACM, 2015. arXiv:1504.03294

19 Persi Diaconis and Mehrdad Shahshahani. Generating a random permutation with random transpositions. *Probability Theory and Related Fields*, 57(2):159–179, 1981. doi:10.1007/BF00535487.

20 Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006. arXiv:quant-ph/0401091

21 Oded Goldreich and Dana Ron. Property testing in bounded degree graphs. *Algorithmica*, 32(2):302–343, 2002. doi:10.1007/s00453-001-0078-7.

22 Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 68–75. Springer, 2011. doi:10.1007/978-3-642-22670-0_9.

23 Michael Jarret, Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithms for connectivity and related problems. In *Proceedings of the 26th European Symposium on Algorithms (ESA)*, pages 49:1–49:13. Springer, 2018. arXiv:1804.10591

24 Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 49:1–49:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. arXiv:1603.08675

25 Hari Krovi, Frédéric Magniez, Maris Ozols, and Jérémie Roland. Quantum walks can find a marked element on any graph. *Algorithmica*, 74(2):851–907, 2016. arXiv:1002.2419

26 David A Levin, Yuval Peres, and Elizabeth L Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2017. doi:10.1090/mbk/058.

27 Andrew Lutomirski. Component mixers and a hardness result for counterfeiting quantum money. arXiv:1107.0321, 2011.

28 Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. arXiv:quant-ph/0608026 doi:10.1137/090745854.

29 Davide Orsucci, Hans J. Briegel, and Vedran Dunjko. Faster quantum mixing for slowly evolving sequences of Markov chains. *Quantum*, 2:105, 2018. arXiv:1503.01334

30 David Poulin and Pawel Wocjan. Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. *Physical Review Letters*, 103(22):220502, 2009. arXiv:0905.2199

31 Peter C Richter. Quantum speedup of classical mixing processes. *Physical Review A*, 76(4):042306, 2007. arXiv:quant-ph/0609204

32 Alistair Sinclair. *Algorithms for random generation and counting: a Markov chain approach*. Springer Science & Business Media, 2012. doi:10.1007/978-1-4612-0323-0.

33 Rolando D Somma, Sergio Boixo, Howard Barnum, and Emanuel Knill. Quantum simulations of classical annealing processes. *Physical Review Letters*, 101(13):130504, 2008. arXiv:0804.1571

34 Daniel A Spielman and Shang-Hua Teng. 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. arXiv:0809.3232

35 Mario Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41. IEEE, 2004. arXiv:quant-ph/0401053

36 Joran Van Apeldoorn, András Gilyén, Sander Gribling, and Ronald de Wolf. Quantum SDP-Solvers: better upper and lower bounds. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 403–414. IEEE, 2017. arXiv:1705.01843

37 John Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings of the 41st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 537–546. IEEE, 2000. arXiv:cs/0009002

38 John Watrous. Quantum simulations of classical random walks and undirected graph connectivity. *Journal of Computer and System Sciences*, 62(2):376–391, 2001. arXiv:cs/9812012

39 Pawel Wocjan and Anura Abeyesinghe. Speedup via quantum sampling. *Physical Review A*, 78(4):042336, 2008. arXiv:0804.4259