A Unified Framework of Quantum Walk Search

Simon Apers* András Gilyén†,‡ Stacey Jeffery§

December 10, 2019

Abstract

The main results on quantum walk search are scattered over different, incomparable frameworks, most notably the *hitting time framework*, originally by Szegedy, the *electric network framework* by Belovs, and the *MNRS framework* by Magniez, Nayak, Roland and Santha. As a result, a number of pieces are currently missing. For instance, the electric network framework allows quantum walks to start from an arbitrary initial state, but it only detects marked elements. In recent work by Ambainis et al., this problem was resolved for the more restricted hitting time framework, in which quantum walks must start from the stationary distribution.

We present a new quantum walk search framework that unifies and strengthens these frameworks. This leads to a number of new results. For instance, the new framework not only detects, but finds marked elements in the electric network setting. The new framework also allows one to interpolate between the hitting time framework, which minimizes the number of walk steps, and the MNRS framework, which minimizes the number of times elements are checked for being marked. This allows for a more natural tradeoff between resources. Whereas the original frameworks only rely on quantum walks and phase estimation, our new algorithm makes use of a technique called *quantum fast-forwarding*, similar to the recent results by Ambainis et al. As a final result we show how in certain cases we can simplify this more involved algorithm to merely applying the quantum walk operator some number of times. This answers an open question of Ambainis et al.

1 Introduction

Quantum walk search refers to the use of quantum walks to solve a search problem on a graph. In the last two decades, this topic has received a great deal of attention, with a rich literature attesting to the progress on understanding quantum walk algorithmic techniques [AKR05, Sze04, MNRS11, KMOR16, Bel13, AGJK19, DH17] and developing applications [BŠ06, MSS07, JKM12, BCJ+13, BJLM13, Mon18, KT17, HM18, Kir18]. Despite this long line of progress, the main results on quantum walk search lie somewhat scattered in different frameworks, and a number of pieces are currently missing.

The quantum walk search frameworks that we consider are the *hitting time framework* originally due to Szegedy [Sze04], the *MNRS framework* due to Magniez, Nayak, Roland and Santha [MNRS11], the *electric network framework* due to Belovs [Bel13], and the *controlled quantum amplification framework* by Dohotaru and Høyer [DH17]. We summarize these frameworks, as well as the corresponding complexities, in Table 1. In this work we unify these different

*Inria, France and CWI, the Netherlands. Supported by the CWI-Inria International Lab. simon.apers@inria.fr

†QuSoft, CWI and University of Amsterdam, the Netherlands. Supported by ERC Consolidator Grant QPROGRESS and partially supported by QuantERA project QuantAlgo 680-91-034.

‡Caltech, USA. Supported by the Institute for Quantum Information and Matter, an NSF Physics Frontiers Center (NSF Grant PHY-1733907). agilyen@caltech.edu

§QuSoft and CWI, the Netherlands. Supported by an NWO Veni Innovational Research Grant under project number 639.021.752, an NWO WISE Grant, and QuantERA project QuantAlgo 680-91-03. SJ is a CIFAR Fellow in the Quantum Information Science Program. jeffery@cwi.nl

frameworks, leading to a number of new results and missing pieces. For example, algorithms developed using the electric network framework could only *detect* marked elements. Our unified approach can be used to develop algorithms that *find* marked elements, while incurring at most a logarithmic overhead.

We also give a conceptual bridge between the recent result of Ref. [AGJK19] and the original approaches by Szegedy [Sze04] and Krovi, Magniez, Ozols and Roland [KMOR16]. The latter showed that combining quantum walks with phase estimation or time averaging allows one to quadratically improve the hitting time of a single marked element, when starting from the stationary distribution. Ambainis et al. [AGJK19] used a more involved technique called *quantum fast-forwarding* [AS19] to improve these results to yield quadratic speedups on the hitting time of arbitrary sets. In this work we reprove the same result using only simple quantum walks, thereby proving a conjecture from [AGJK19].

1.1 Different Frameworks

While the frameworks we consider are similar, each has advantages and disadvantages. The earliest *hitting time framework* was due to Szegedy [Sze04], inspired by an algorithm of Ambainis for element distinctness [Amb07]. To illustrate this framework, imagine a classical algorithm that begins by sampling a state from the stationary distribution π of some random walk, described by a transition matrix P. The algorithm starts from a vertex distributed according to π, and simulates the random walk. After every step of the walk it checks whether the current vertex is "marked". The algorithm terminates after $\mathcal{O}(\mathrm{HT}(P, M))$ steps, with $\mathrm{HT}(P, M)$ the *hitting time*, or the expected number of steps from π before a marked vertex in M, the marked set, is reached. As such, the algorithm has a constant probability of having found a marked vertex. To bound the complexity of this algorithm, let the *setup cost* $\mathcal{S}$ denote the complexity of sampling from π, the *update cost* $\mathcal{U}$ denote the complexity of simulating a step of the walk, and the *checking cost* $\mathcal{C}$ denote the complexity of checking whether a vertex is marked. The complexity of the resulting algorithm is then of order $\mathcal{S} + \mathrm{HT}(P, M)(\mathcal{U} + \mathcal{C})$. The hitting time framework essentially shows how to construct a quantum algorithm with complexity

$$\mathsf{S} + \sqrt{\mathrm{HT}(P, M)}(\mathsf{U} + \mathsf{C}),$$

where S, U and C are quantum analogues of $\mathcal{S}$, $\mathcal{U}$ and $\mathcal{C}$, respectively, denoting the costs in terms of *coherent* quantum samples (see Section 2.4 for details). One of the major drawbacks of the original framework was that the resulting quantum algorithm typically *detected* the presence of a marked vertex, without actually *finding* one. In the special case where there is only a single marked element, Krovi at al. [KMOR16] showed how to also find the marked element in the same complexity. To this end they introduced the concept of *interpolated walks*. Combining interpolated walks with another technique called *quantum fast-forwarding*, introduced in [AS19], Ref. [AGJK19] more recently showed how to also find a marked element in the general case. We will refer to this final result as the hitting time framework.

The second framework that we consider is the *MNRS framework* introduced by Magniez, Nayak, Roland and Santha [MNRS11]. This framework also *finds* a marked vertex, but it can be understood as the quantum analogue of a slightly different random walk algorithm. Consider a random walk that begins in the stationary distribution. Rather than checking if the current vertex is marked after every step, the walk takes $1/\delta$ steps between checks, where δ is the *spectral gap* of P. Since $1/\delta$ is approximately the *mixing time* of the random walk, this process effectively samples from the stationary distribution, for each sample checking whether it is marked, and otherwise generating a new sample. If ε is the probability that a vertex sampled from the stationary distribution is marked, then a marked element is found with constant probability after $\mathcal{O}(1/\varepsilon)$ samples. As such, the complexity of this classical algorithm is $\mathcal{S} + \frac{1}{\varepsilon}(\frac{1}{\delta}\mathcal{U} + \mathcal{C})$. The MNRS framework shows how to get a quantum algorithm for finding a marked vertex with

Framework	Complexity
Hitting time framework [Sze04, KMOR16, AGJK19]	$\mathsf{S} + \sqrt{\mathrm{HT}(P,M)}(\mathsf{U}+\mathsf{C})$
MNRS framework [MNRS11]	$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathsf{U}+\mathsf{C})$
Electric network framework [BCJ+13, Bel13]	$\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}}(\mathsf{U}(\sigma)+\mathsf{C})$
Controlled quantum amplification [DH17]	$\mathsf{S} + \sqrt{\mathrm{HT}(P,\{m\})}\mathsf{U} + \frac{1}{\sqrt{\varepsilon}}\mathsf{C}$

Table 1: Comparison of different quantum walk frameworks.

complexity

$$\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\Big(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\Big).$$

Since $\mathrm{HT}(P,M) \leq \frac{1}{\varepsilon\delta}$, this requires at least as many steps of the walk as the hitting time framework. On the other hand, $\mathrm{HT}(P,M) \geq \frac{1}{\varepsilon}$, and so the number of checks can be significantly smaller than in the hitting time framework. In fact, this amount of checks performed in the MNRS framework is easily seen to be optimal by a lower bound on black-box search.[1]

The third framework that we consider is the *electric network framework* by Belovs [Bel13] (published in [BCJ+13]). This is a generalization of the hitting time framework, allowing for the walker to start from an arbitrary initial distribution σ (such as a single vertex), rather than necessarily the stationary distribution. If $\mathsf{S}(\sigma)$ is the complexity of sampling (coherently) from σ, then the resulting quantum algorithm has complexity

$$\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}}(\mathsf{U}(\sigma)+\mathsf{C}),$$

where $\mathsf{U}(\sigma)$ is the complexity of implementing a step of a slightly modified random walk. The quantity $C_{\sigma,M}$ (defined in Section 2.3) is a generalization of the *commute time*. When both σ and M correspond to single vertices u and m, then $C_{\sigma,M}$ equals the commute time from u to m, which is the expected number of steps starting from u to reach m and then return to u. When σ equals the stationary distribution then $C_{\sigma,M} = HT(P,M)$, thus retrieving the hitting time framework. The obvious advantage of the electric network framework is that it does not necessarily require quantum samples from the stationary distribution of P, which might be very costly, and can instead begin in a much easier to produce state. A major disadvantage of this framework, however, is that the quantum algorithm only *detects* the presence of marked vertices, as in the original hitting time framework, rather than actually finding marked vertices.

Finally we also consider the *controlled quantum amplification framework* by Dohotaru and Høyer [DH17]. They use an extra qubit to control the quantum walk operator[2], leading to an additional degree of freedom. For the case of a *unique* marked element $M = \{m\}$, and starting from a quantum sample of the stationary distribution, they achieve a complexity

$$\mathsf{S} + \sqrt{\mathrm{HT}(P,\{m\})}\mathsf{U} + \frac{1}{\sqrt{\varepsilon}}\mathsf{C},$$

which has both an optimal number of walk steps (as the hitting time framework) and an optimal number of checks (as the MNRS framework). The clear downside of this approach is that it is restricted to cases where there is a single marked element, and we start from the stationary distribution.

[1] Consider for instance a quantum walk search algorithm on the complete graph on N vertices. Finding a single marked element then requires $\Omega(\sqrt{N})$ checks by the optimality of Grover's search algorithm.

[2] In fact they consider more general operators, but we will focus on their result for quantum walk operators.

1.2 Contributions

Finding in the electric network framework The electric network framework [Bel13] generalizes the hitting time framework [Sze04] by allowing for arbitrary initial distributions. The downside is that algorithms in this framework only detect rather than actually find marked vertices. On the other hand, the improved hitting time framework of [AGJK19] shows how to actually find marked vertices in the hitting time framework, provided that the walk starts from a quantum sample of the stationary distribution. Both works hence provide complementary but incompatible improvements over the initial hitting time framework.

In Section 4, we fill this gap by generalizing the results of [AGJK19] to the electric network setting, designing a quantum algorithm that not only detects but also *finds* marked elements for any starting distribution σ. This improved version strictly generalizes the results of [AGJK19], and it loses at most a log factor with respect to the original electric network framework [Bel13]. In particular, we show (see Theorem 13):

Theorem 1 (Informal)**.** For any distribution σ, there is a quantum walk search algorithm that finds a marked element from M with constant probability in complexity (up to log factors)

$$\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}}(\mathsf{U}(\sigma) + \mathsf{C}).$$

To analyze our new algorithm, we use techniques similar to those employed in [AGJK19] for finding in the hitting time framework. However, there is an additional difficulty we must overcome. The hitting time, $\mathrm{HT}(P, M)$, has a useful interpretation in terms of the classical random walk – that is, with high probability, a marked vertex is encountered within the first $\mathcal{O}(\mathrm{HT}(P, M))$ steps – and this fact is crucial in the analysis of the quantum algorithm in [AGJK19]. In contrast, to the best of our knowledge, the generalized quantity $C_{\sigma,M}$ (defined in Section 2.3) is not well understood. If σ is supported on a single vertex, u, and M contains a single vertex, m, then $C_{\sigma,M}$ is exactly the *commute time* between u and m. This means that within the first $\mathcal{O}(C_{\sigma,M})$ steps, with high probability, a walker starting from u has visited m, and then returned to u. For general σ and M, no such interpretation was known. We prove that, under certain conditions, a similar interpretation holds: with high probability, a walker starting from σ will hit M, and then return to the support of σ, within the first $\mathcal{O}(C_{\sigma,M})$ steps. We can ensure that these conditions hold by using the same graph and walk modification as used in [Bel13], adding a weighted edge to each vertex in $\mathrm{supp}(\sigma)$. The resulting understanding of $C_{\sigma,M}$ is enough to employ a similar analysis to that of [AGJK19].

A Unified Framework While the electric network framework is a generalization of the hitting time framework, the MNRS framework is incomparable. Since $\mathrm{HT}(P, M) \leq \frac{1}{\varepsilon\delta}$, the hitting time framework always finds a marked vertex using a number of quantum walk steps (updates) less than or equal to that used by the MNRS framework. On the other hand, $\mathrm{HT}(P, M) \geq \frac{1}{\varepsilon}$, and hence the MNRS framework may make fewer calls to the check operation. When the complexity of implementing the checking operation is much larger than that of the update operation, the MNRS framework may hence be preferable to both the hitting time framework and the electric framework. The controlled quantum amplification framework achieves the best of both worlds, but only for a unique marked element.

In Section 5, we present a new framework that unifies all these individual approaches. For the sake of intuition, we first describe this framework when the initial state π is used, which can be seen as a unification between the hitting time framework, the MNRS framework and the controlled quantum amplification framework. To this end, recall that the hitting time framework is the quantum analogue of a random walk algorithm that takes $\mathrm{HT}(P, M)$ steps of the random walk described by P, checking at each step if the current vertex is marked. In contrast, the MNRS framework is the quantum analogue of a random walk algorithm that takes $\frac{1}{\delta}$ steps of P, thus approximately sampling from the stationary distribution π, and then checks if the sampled

vertex is marked. Since ε is the probability that a sampled vertex is marked, this process is repeated $\frac{1}{\varepsilon}$ times.

We can define a natural interpolation between both classical algorithm. To this end, take any t, and consider a classical random walk that repeatedly takes t steps, and then checks whether the current vertex is marked. The expected number of iterations is then $\mathrm{HT}(P^t, M)$, the hitting time of the t-step random walk, described by transition matrix P^t. This classical algorithm finds a marked vertex in complexity $\mathcal{S} + \mathrm{HT}(P^t, M)(t\mathcal{U} + \mathcal{C})$. We give a quantum analogue of this algorithm, generalized to arbitrary initial distributions (see Theorem 27).

Theorem 2 (Informal)**.** For any $t \in \mathbb{N}$ and any distribution σ, there is a quantum walk search algorithm that finds a marked element from M with constant probability in complexity (up to log factors)

$$\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}(P^t)}(\sqrt{t}\mathsf{U}(\sigma) + \mathsf{C}).$$

Setting $t = 1$ we recover our previous theorem, Theorem 1. When $\sigma = \pi$, then $C_{\sigma,M}(P^t) = \mathrm{HT}(P^t, M)$, and hence we find the quantum analogue of the aforementioned random walk algorithm. As such, when $\sigma = \pi$ and $t = 1$, we recover the hitting time framework. When $\sigma = \pi$ and $t = \frac{1}{\delta}$, we recover the MNRS framework, since a $1/\delta$-step random walk essentially samples from π at every step, and so $\mathrm{HT}(P^{1/\delta}, M) \in \mathcal{O}(\frac{1}{\varepsilon})$. When there is a unique marked element $\{m\}$, and $t = \varepsilon\mathrm{HT}(P, \{m\})$, we recover the controlled quantum amplification framework. To see this, we use a result from [DH17, Section 6] which proves that $\mathrm{HT}(P^t, \{m\}) = 1/\varepsilon$ if $t \in \Omega(\varepsilon\mathrm{HT}(P, \{m\}))$. For multiple marked elements, and other intermediate values of t, we obtain new types of algorithms. We summarize these special cases in the table below.

New quantum walk search framework:	$\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}(P^t)}(\sqrt{t}\mathsf{U}(\sigma) + \mathsf{C})$
Hitting time framework	$\sigma = \pi$, $t = 1$
MNRS framework	$\sigma = \pi$, $t = \frac{1}{\delta}$
Electric network framework	any σ, $t = 1$
Controlled quantum amplification	$\sigma = \pi$, $M = \{m\}$, $t = \varepsilon\mathrm{HT}(P, M)$

Table 2: The new quantum walk search framework.

A simpler algorithm for the hitting time and electric network framework Similar to the recent work by Ambainis et al. [AGJK19], our new quantum algorithm makes use of a somewhat involved technique called quantum fast-forwarding. For the case $t = 1$ (recovering the hitting time and electric network framework), we show that a much simpler algorithm works with essentially the same complexity. This algorithm works by (classically) choosing random interpolation parameters, and applies the interpolated quantum walk operator an appropriately chosen number of steps, starting from $|\sqrt{\sigma}\rangle$. It was already conjectured in [AGJK19] that this simpler approach would work (but only for the hitting time framework). In Section 6, we prove that this simple algorithm indeed finds a marked vertex, with at most a logarithmic overhead over the complexity of the more involved fast-forwarding algorithm. Interestingly, our proof relies on the proof of correctness of the fast-forwarding algorithm.

Related independent work While finalizing this manuscript, the authors became aware of the concurrent and independent work of Stephen Piddock, who developed an alternative refinement of Belovs' results for finding marked elements in the electric network framework [Pid19].

2 Preliminaries

2.1 Random walks

Let Y be a random variable over a finite state space X, with $|X| = n$. For $y \in X$, we let $\Pr(Y = y)$ denote the probability that $Y = y$. We can describe the corresponding probability distribution by a vector $\sigma \in \mathbb{R}^n$, where $\sigma_y = \Pr(Y = y)$. For $S \subseteq X$ and a probability distribution σ, we let $\sigma(S) = \sum_{u \in S} \sigma_u$ denote the probability that $Y \in S$, and we let $\sigma|_S$ be the normalized restriction of σ to S, defined as $(\sigma|_S)_u = \sigma_u/\sigma(S)$ for all $u \in S$. A sequence of random variables $Y = (Y_t)_{t=0}^{\infty}$ over X is a *Markov chain* if for all $t \geq 1$, Y_t is independent of $Y_0, \ldots, Y_{t-2}$ given Y_{t-1}. For any distribution σ over X and random variable Z that is a function of Y, we let $\Pr_\sigma(Z = z)$ denote the probability that Z takes value z when Y_0 is distributed as σ. Any Markov chain is described by a stochastic transition matrix P, with $P_{y,y'} = \Pr(Y_t = y' \mid Y_{t-1} = y)$. If $\sigma^{(t)}$ describes the probability distribution of Y_t, then this implies that $\sigma^{(t)} = \sigma^{(t-1)}P$.

We consider weighted, undirected graphs $G = (X, E, w)$ on vertex set X, with $|X| = n$; edge set $E \subseteq X \times X$ with $(u, v) \in E$ if and only if $(v, u) \in E$, and $|E| = 2m$; and edge weights $w : E \to \mathbb{R}_{\geq 0}$. If an edge is not present, we will usually think of it as having edge weight zero. The total weight is then $W = \sum_{u,v \in X} w_{u,v}$, and the total weight of edges leaving a node u is $w_u = \sum_{v \in X} w_{u,v}$. A random walk on G is described by a Markov chain over X with transition matrix P, defined by

$$P_{u,v} = \frac{w_{u,v}}{w_u}.$$

In words, a random walk from a vertex u picks a random neighbor v with probability proportional to the edge weight $w_{u,v}$. This random walk describes a special kind of Markov chain, a so-called *reversible* Markov chain [LPW17]. In fact, it can be shown that any reversible Markov chain can also be described as a random walk on a weighted graph, simply by choosing $w_{u,v} = w_{v,u} = 2\pi_u P_{u,v}$. As such we will interchangeably use the terms "random walk" and "reversible Markov chain". If the graph is connected and non-bipartite, the random walk is called *ergodic* and its probability distribution converges to a unique limiting distribution called the *stationary distribution* $\pi \in \mathbb{R}^n$, defined as

$$\pi_u = \frac{w_u}{W}.$$

This is the unique left eigenvector of P with eigenvalue 1. While the transition matrix P of a reversible Markov chain is not necessarily symmetric, a closely related matrix called the *discriminant matrix* $D(P)$ is symmetric. For ergodic and reversible Markov chains it can be defined as

$$D(P) := \sqrt{P \circ P^T} = \operatorname{diag}\left(\sqrt{\pi}\right) P \operatorname{diag}\left(\sqrt{\pi}\right)^{-1}, \tag{1}$$

with the $\circ$-product and square root acting elementwise and $\operatorname{diag}(\sqrt{\pi})$ being the diagonal matrix with entries $(\operatorname{diag}(\sqrt{\pi}))_{u,u} = \sqrt{\pi_u}$. The second equality implies that P and $D(P)$ share the same eigenvalues, which we denote by $1 = \lambda_0 > \lambda_1 \geq \cdots \geq \lambda_{n-1} > -1$. The convergence time or *mixing time* of P to the stationary distribution is characterized, up to log factors, by the inverse of the *spectral gap* δ of the transition matrix, which is defined as $\delta = \min\{1 - |\lambda_1|, 1 - |\lambda_{n-1}|\}$.

For a subset $M \subseteq X$, the *hitting time* τ_M is the random variable representing the minimum t such that $Y_t \in M$, i.e., the first time at which the random walk hits M. With slight abuse of notation, we will also call $\mathrm{HT}(P, M) = \mathbb{E}_\pi(\tau_M)$ the hitting time of set M, corresponding to the expected hitting time when starting from the stationary distribution π. We similarly define τ_S^M to be the first time at which the random walk has hit M, and then S. As such, when $S = \{s\}$ is a singleton, the quantity $\mathbb{E}_s(\tau_s^M)$ denotes the expected *commute time* from s to M.

2.2 Interpolated walks

Interpolated walks form an important tool in quantum walk search algorithms [KMOR16]. Quite literally, such walks are an interpolation between the original random walk P, and the *absorbing*

random walk P_M in some set M, which corresponds to the walk that halts when it hits M (equivalently, the walk obtained after adding self-loops of infinite weight to the elements in M). For an interpolation parameter $s \in [0,1]$, the interpolated walk $P(s)$ is then defined as

$$P(s) = (1-s)P + sP_M.$$

If P is an ergodic reversible Markov chain, then so is $P(s)$ for every $s \in [0,1)$ [KMOR16, Proposition 12].

2.3 Electric networks

An electric network is described by a weighted, undirected graph $G = (X, E, w)$ (for an introduction to the connection between random walks on graphs and electric networks, see [DS84]). The edge weights in G are interpreted as *conductances* associated to the edges. An edge that is not present has weight zero, and hence also zero conductance. A central notion is that of a *flow*.

Definition 3. Let $M \subset X$ be a set of marked vertices, and let σ be a distribution supported on unmarked vertices. A *unit flow* from σ to M is a function $p : X \times X \to \mathbb{R}$ such that:

- $p_{u,v} = 0$ if $w_{u,v} = 0$;
- $p_{u,v} = -p_{v,u}$ for all u, v;
- for all $u \notin M$, $\sum_v p_{u,v} = \sigma_u$; and
- $\sum_{u \in M} \sum_{v \notin M} p_{u,v} = -1$.

The *effective resistance* $R_{\sigma,M}$ from σ to M is

$$R_{\sigma,M} = \min_p \sum_{u,v \in X : u<v} \frac{p_{u,v}^2}{w_{u,v}} \tag{2}$$

where the minimum runs over all unit flows from σ to M. For an S disjoint from M we define

$$R_{M,S} = R_{S,M} := \min_{\sigma \colon \operatorname{supp}(\sigma) \subseteq S} R_{\sigma,M}$$

In the special case when $S = \{s\}$ and $M = \{t\}$ are singletons, we simply write $R_{s,t} := R_{\{s\},\{t\}}$.

Note that there is a unique σ-M flow that minimizes the expression in (2). To see this, note that if two distinct flows p and p' achieve the same minimum, then their average is again a σ-M flow, but with an even smaller value, which leads to a contradiction. In the special cases where $\sigma = \pi$ or $\operatorname{supp}(\sigma) = \{s\}$, the effective resistance $R_{\sigma,M}$ has a well-known combinatorial interpretation.

Theorem 4 ([CRR+96, Bel13])**.** In a weighted graph of total weight W, for any vertex s and subset $M \subseteq X$, we have $WR_{s,M} = C_{s,M}$, where $C_{s,M}$ is the *commute time* from s to M. Furthermore, we have $WR_{\pi,M} = \mathrm{HT}(P, M)$, with $\mathrm{HT}(P, M)$ the hitting time from π to M.

Since we could only find in the literature a proof of the first statement for the case where M is a singleton, we extend the existing proofs to the more general case in Appendix B.

In the same vein we define the quantity $C_{\sigma,M} = WR_{\sigma,M}$ for any distribution σ and subset M, and define $C_{S,M}$ analogously. Similar to the above theorem, we will later prove a connection of this more general quantity to the behavior of a random walk.

2.4 Quantum walks and quantum walk search algorithms

Let $D(P)$ denote the discriminant matrix of a random walk on a weighted graph, as defined in (1). We can associate a quantum walk operator to this random walk, which is a unitary operator for which the following holds.

Definition 5 (Quantum walk operator)**.** For any reversible Markov chain P on a finite state space X, a *quantum walk operator* $W(P)$ is a unitary on $\mathcal{H}_A \otimes \mathcal{H}_X$, such that $|\bar{0}\rangle \in \mathcal{H}_A$, $\mathrm{span}\{|u\rangle : u \in X\} \subseteq \mathcal{H}_X$, and for all $u \in X$ it holds that

$$(\langle\bar{0}| \otimes \langle u|)W(P)(|\bar{0}\rangle \otimes I_X) = \langle u|D(P) \quad \text{and} \quad (\langle\bar{0}| \otimes I_X)W(P)(|\bar{0}\rangle \otimes |u\rangle) = D(P)|u\rangle.$$

We note that this definition is more general than the usual notion of Szegedy's quantum walk operator [Sze04]. Nevertheless, this definition perfectly fits Szegedy's framework and its later extensions, and enables obtaining all major results (but with increased generality and clarity).

In case $\mathcal{H}_X = \mathrm{span}\{|u\rangle : u \in X\}$ we can simply write $(\langle\bar{0}| \otimes I)W(P)(|\bar{0}\rangle \otimes I) = D(P)$ and such a unitary is called a *block-encoding* of $D(P)$. But as indicated by our definition, all of our results also apply if $W(P)$ is a block-encoding of a matrix M that can be block-diagonalised with $D(P)$ being one of its diagonal blocks – this generalization is relevant for example in applications where a data-structure is attached to each vertex (for more details see [GSLW19]). Thus, for simplicity in the presentation we will use a slight abuse of notation and simply write $(\langle\bar{0}| \otimes I)W(P)(|\bar{0}\rangle \otimes I) = D(P)$, when $(\langle\bar{0}| \otimes I)W(P)(|\bar{0}\rangle \otimes I) = M$ and M is block-diagonal, and the block corresponding to the subspace $\mathrm{span}\{|u\rangle : u \in X\}$ is $D(P)$.

To gain some intuition, we recall the usual construction for Szegedy's quantum walk operator [Sze04]. It starts from the classical walk perspective: a step of a classical random walk from vertex u consists of sampling a new vertex v according to the distribution $P_{u,\cdot}$ given by the u-th row of P. An analogous quantum operation is a unitary $V(P)$ on $\mathrm{span}\{|u, v\rangle : u, v \in X \cup \{\bar{0}\}\}$ defined by

$$|\bar{0}\rangle|u\rangle \mapsto V(P)|\bar{0}\rangle|u\rangle = \left(\sum_{v \in X} \sqrt{P_{u,v}}|v\rangle\right)|u\rangle, \tag{3}$$

and acting arbitrarily (but unitarily, and controlled on the second register) on the rest of the state space. Using such a unitary, one can simulate the classical random walk by measuring the state and re-initializing the first register to $\bar{0}$ in every step. Using the unitary swap operator $\textsc{Shift}\colon |u, v\rangle \mapsto |v, u\rangle$ (for $u, v \in X$), we can now define the operator

$$V(P)^\dagger \,\textsc{Shift}\, V(P), \tag{4}$$

where the dagger $\dagger$ denotes the Hermitian conjugate. One can now verify that this operator is indeed a quantum walk operator, as defined in Definition 5.

In order to understand the relationship between our perspective and most previous works on Szegedy-type quantum walks, we note that a Szegedy-type quantum walk is usually implemented as the following sequence of gates:

$$\ldots \overbrace{V(P)^\dagger \,\textsc{Shift}\, V(P)}^{W(P)}([2|\bar{0}\rangle\langle\bar{0}| - I] \otimes I)V(P)^\dagger \,\textsc{Shift}\, \underbrace{V(P)([2|\bar{0}\rangle\langle\bar{0}| - I] \otimes I)V(P)^\dagger}_{\textsc{Ref}} \ldots$$

This can be viewed [Sze04, MNRS11] as a sequence of unitaries $\ldots \textsc{Shift}\,\textsc{Ref}\,\textsc{Shift}\,\textsc{Ref}\ldots$, where $\textsc{Ref}$ is a reflection around the span of the states in the right-hand side of (3). We instead look at it as the sequence $\ldots W(P)\left([2|\bar{0}\rangle\langle\bar{0}| - I] \otimes I\right) W(P)\left([2|\bar{0}\rangle\langle\bar{0}| - I] \otimes I\right)\ldots$. There are various advantages of our treatment:

- It directly reveals the discriminant matrix, which is at the core of the analysis

- It enables the use of new techniques such as fast-forwarding or block-encoding
- There is no need to work with pairs of vertices $|u\rangle|v\rangle$
- The similarities and differences compared to the classical walk are more apparent

Analogous to the case of classical random walk algorithms, quantum walk search algorithms are assumed to have access to the following (possibly controlled) black-box operations (and their inverses):

- $\texttt{Check}(M)$: checks whether a vertex u is marked. Complexity $\mathsf{C}(M)$ or C. Described by the mapping
$$\forall u \in X, b \in \{0,1\}: \quad |u\rangle|b\rangle \mapsto \begin{cases} |u\rangle|b\rangle & \text{if } u \notin M \\ |u\rangle|b \oplus 1\rangle & \text{if } u \in M. \end{cases}$$
- $\texttt{Setup}(\pi)$: generates the superposition $|\sqrt{\pi}\rangle = \sum_{u \in X} \sqrt{\pi_u}|u\rangle$. Complexity $\mathsf{S}(\pi)$ or S.
- $\texttt{Update}(P)$: implements a (controlled) walk operator $W(P)$, as in Definition 5. Complexity $\mathsf{U}(P)$ or U.

Remark 6. In the literature the update cost is often defined as the cost implementing the unitary $V(P)$ described in (3), which is compatible with our cost notion due to (4). However, in some papers the update cost is defined as the cost of implementing Ref, which is harder to compare directly. Still it seems unlikely that Ref can be implemented much more efficiently than $W(P)$, so we do not devote much attention to this minor conflict in the definitions. When we cite such a paper we re-express their bounds in terms of our cost functions, c.f., Belovs' original paper [Bel13] on electrical network based quantum walks.

Implementing interpolated quantum walks. We can derive other operations by combining the above black-box operations. Say that we wish to implement a quantum walk corresponding to the interpolated walk $P(s)$. Let $\theta = \arccos(\sqrt{s})/2$, and
$$V = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}.$$
Then
$$-YVXV|0\rangle = \sqrt{s}|0\rangle + \sqrt{1-s}|1\rangle, \quad -YVV|0\rangle = |1\rangle.$$
Let $cW(P)$ be a controlled version of the "update unitary" $W(P)$ controlled by the first qubit, and C be the "check unitary" flipping the first qubit for marked vertices. Then the operator
$$[I_A \otimes (V \otimes I_X)\, C\, (VY \otimes I_X)]\, cW(P)\, [I_A \otimes (YV \otimes I_X)\, C\, (V \otimes I_X)]$$
is a quantum walk operator[3] corresponding to $D(P(s))$. This shows that we can implement $\texttt{Update}(P(s))$ using one call to $\texttt{Update}(P)$, two calls to $\texttt{Check}(M)$ and 4 elementary gates. We can hence bound the complexity
$$\mathsf{U}(P(s)) \in \mathcal{O}(\mathsf{U}(P) + \mathsf{C}(M)).$$

[3] In order to show that it satisfies the requirements of Definition 5, note that the subspace $\text{span}(|0\rangle, |1\rangle) \otimes \text{span}\{|u\rangle : u \in X\}$ is invariant under the action of C.

2.5 Quantum fast-forwarding

Similar to interpolated walks, *quantum fast-forwarding* [AS19] has proven to be a useful tool for quantum walk search algorithms. For example, the most recent developments [AGJK19] in the hitting time framework (Theorem 8) are based on this technique.

Theorem 7 (Quantum fast-forwarding [AS19])**.** Let $\varepsilon \in (0,1)$, $t \in \mathbb{N}$ and let P be any reversible Markov chain on state space X. There is a quantum algorithm with complexity $\mathcal{O}\left(\sqrt{t \log \frac{1}{\varepsilon}}\mathsf{U}\right)$, where U is the cost[4] of implementing a quantum walk operator $W(P)$ (and its inverse), that implements a unitary U on $\operatorname{span}\{|a,x\rangle : a \in A, x \in X\}$ for some finite set $A \ni \{\tilde{0}\}$, such that for any $|\psi\rangle \in \operatorname{span}\{|x\rangle : x \in X\}$,

$$\left\|(\langle\tilde{0}| \otimes I)U|\bar{0}\rangle|\psi\rangle - |\tilde{0}\rangle D(P)^t|\psi\rangle\right\|^2 \leq \varepsilon.$$

The resulting unitary U can be equivalently described as a $(1, \log|A|, \varepsilon)$-block-encoding of $D(P)^t$ [CGJ19, GSLW19]. For completeness we will later reprove this theorem, and give an explicit quantum circuit solving this problem, that will play a crucial role in Section 6.

3 Quantum walk frameworks

In this section we survey the different quantum walk search frameworks.

3.1 Hitting time framework

The hitting time framework is the quantum analogue of arguably the simplest random walk search algorithm:[5]

1. Use `Setup`(π) to sample a vertex u according to π.
2. Repeat HT times:
   (a) Check if the current vertex is marked using `Check`(M).
   (b) Sample a neighbour of the current vertex using `Update`(P), and make that the current vertex.

If $\mathrm{HT} \in \Omega(\mathrm{HT}(P,M))$, then this algorithm finds a marked vertex with constant probability. Its complexity is $\mathcal{S} + \mathrm{HT}(\mathcal{U} + \mathcal{C})$.

The quantum analogue of this framework was first introduced by Szegedy [Sze04], giving a quadratic speedup over the update and checking costs of the classical algorithm. His algorithm, however, only *detected* the presence of a marked vertex, rather than actually finding one. Later work by Ambainis et al. [AGJK19] resolved this issue, describing a quantum walk algorithm that effectively *finds* a marked vertex with constant probability. We summarize their result in the theorem below.

Theorem 8 (Hitting time framework)**.** Let P be any reversible Markov chain on a finite state space X, $M \subset X$ a marked set, and HT a known upper bound on $\mathrm{HT}(P,M)$. Then there is a quantum algorithm that outputs a vertex x from M with constant probability in complexity

$$\mathcal{O}\left(\mathsf{S}\sqrt{\log(\mathrm{HT})} + \sqrt{\mathrm{HT}}(\mathsf{U}+\mathsf{C})\sqrt{\log(\mathrm{HT})\log\log(\mathrm{HT})}\right).$$

[4] For simplicity we assume that U is at least the number of qubits on which $W(P)$ act. This is a fair assumption because if $W(P)$ uses fewer gates than qubits, then there must be some unused qubits.

[5] In the case of classical algorithms, the subroutines `Setup`, `Update`, and `Check` are assumed to: sample a vertex according to π; sample a neighbour of the current vertex u; and check if the current vertex is marked. Note that the costs $\mathcal{S}$ and $\mathcal{U}$ of the classical operations might be significantly cheaper than their quantum counterparts S and U (since the quantum checking operation is just the reversible version of the classical checking operation, it will never be significantly harder).

3.2 MNRS framework

While the hitting time framework is optimal in terms of the number of quantum walk steps, it may be suboptimal in the number of calls to $\texttt{Check}(M)$. In contrast, the MNRS framework uses an optimal number of calls to $\texttt{Check}(M)$, at the expense of possibly more calls to $\texttt{Update}(P)$. This framework is again best understood using a random walk perspective. Specifically, consider a classical algorithm that, rather than checking after every step, only checks after every $\frac{1}{\delta}$ steps:

1. Use $\texttt{Setup}(\pi)$ to sample a vertex u according to π.
2. Repeat $\frac{1}{\varepsilon}$ times:
   (a) Check if the current vertex is marked using $\texttt{Check}(M)$.
   (b) Repeat $\frac{1}{\delta}$ times:
   Sample a neighbour of the current vertex using $\texttt{Update}(P)$, and make that the current vertex.

If δ is a lower bound on the spectral gap of P, then by taking $\frac{1}{\delta}$ steps between checks, each random walk vertex that is checked is approximately distributed as an independent sample from π. Hence, if $\varepsilon \leq \pi(M)$, then after $\frac{1}{\varepsilon}$ samples from π a marked vertex will be sampled with constant probability. The total complexity of this classical algorithm is $\mathcal{S} + \frac{1}{\varepsilon}(\frac{1}{\delta}\mathcal{U} + \mathcal{C})$. If $\varepsilon = \pi(M)$ and δ equals the spectral gap of P, then it holds that

$$\frac{1}{\varepsilon} \leq \mathrm{HT}(P, M) \leq \frac{1}{\varepsilon\delta}. \tag{5}$$

Hence this approach is often suboptimal in the number of steps with respect to the algorithm in the previous section[6], but it may be better in the number of checks. This algorithm will hence be preferable in cases where the checking cost $\mathcal{C}$ is much larger than the update cost $\mathcal{U}$.

The quantum analogue of this classical algorithm was introduced by Magniez, Nayak, Roland and Santha [MNRS11], who showed the following:

Theorem 9 (MNRS framework)**.** Let P be any reversible Markov chain on a finite state space X, $M \subset X$ a marked set, ε a known lower bound on $\pi(M)$ and δ a known lower bound on the spectral gap of P. Then there is a quantum algorithm that outputs a vertex x from M with constant probability in complexity

$$\mathcal{O}\left(\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}\left(\frac{1}{\sqrt{\delta}}\mathsf{U} + \mathsf{C}\right)\right).$$

3.3 Controlled quantum amplification framework

Dohotaru and Høyer [DH17] showed that an *optimal* payoff between update cost and checking cost can be obtained for the special case where there is a single marked element $M = \{m\}$. Specifically, consider the following classical algorithm:

1. Use $\texttt{Setup}(\pi)$ to sample a vertex u according to π.
2. Repeat $1/\varepsilon$ times:
   (a) Check if the current vertex is marked using $\texttt{Check}(M)$.
   (b) Repeat $\varepsilon\mathrm{HT}$ times:
   Sample a neighbour of the current vertex using $\texttt{Update}(P)$, and make that the current vertex.

[6] E.g., consider the graph consisting of two cliques on $n/2$ nodes, and a single edge between them. For a single marked element we get that $\pi(m) \in \Theta(1/n)$, $\delta(P) \in \Theta(1/n^2)$ and $\mathrm{HT} \in \Theta(1/n^2)$, so that $1/(\epsilon\delta) \gg \mathrm{HT}$.

If $\varepsilon \in \Theta(\pi(m))$, $\mathrm{HT} \in \Omega(\mathrm{HT}(P, \{m\}))$, and there is a unique marked element $M = \{m\}$, then the above algorithm returns m with constant probability. This can be proven using the following lemma.

Lemma 10 ([DH17, Section 6])**.** Consider a reversible Markov chain P with stationary distribution π, and a single marked element m. If $\tau \in \Omega(\pi(m)\mathrm{HT}(P, m))$, then

$$\mathrm{HT}(P^\tau, m) \in \mathcal{O}\left(\frac{1}{\pi(m)}\right).$$

If we set $\tau = \varepsilon\mathrm{HT}$ then $\mathrm{HT}(P^\tau, m)$ exactly describes the expected number of repetitions of step 2. in the above algorithm that are required to find a marked element. The algorithm has complexity of the order $\mathcal{S} + \mathrm{HT}\mathcal{U} + \frac{1}{\varepsilon}\mathcal{C}$. This corresponds to the update complexity of the classical algorithm in the hitting time framework, and the checking complexity of the classical algorithm in the MNRS framework.

In their *controlled quantum amplification framework*, Dohotaru and Høyer [DH17] construct a quantum analogue of the above result.

Theorem 11 (Controlled quantum amplification framework)**.** Let P be any reversible Markov chain on a finite state space X, $m \in X$ a unique marked element, HT a known upper bound on $\mathrm{HT}(P, \{m\})$, and ε a known lower bound on $\pi(m)$. Then there is a quantum algorithm that outputs m with constant probability in complexity

$$\widetilde{\mathcal{O}}\left(\mathsf{S} + \sqrt{\mathrm{HT}}\,\mathsf{U} + \frac{1}{\sqrt{\varepsilon}}\,\mathsf{C}\right).$$

3.4 Electric network framework

A main drawback of all the aforementioned frameworks is that they require the quantum walk to start from a quantum sample of the stationary distribution, which may in general be much more difficult to construct than, say, a distribution that is supported on a single vertex. Belovs [Bel13] showed that one can combine quantum walks with tools from electric network theory to get rid of this restriction, proving the theorem below. For an arbitrary distribution σ, we let $\mathsf{S}(\sigma)$ denote the complexity of generating the initial state $|\sqrt{\sigma}\rangle = \sum_u \sqrt{\sigma_u}|u\rangle$, which can be much smaller than the setup cost $\mathsf{S} = \mathsf{S}(\pi)$ for generating the quantum sample $|\sqrt{\pi}\rangle$ of the stationary distribution.

We also define $\Lambda(\sigma, C)$, for some choice of C, as a unitary that acts as:

$$|0\rangle|u\rangle \mapsto \frac{\sqrt{\pi_u}|0\rangle + \sqrt{\sigma_u/C}|1\rangle}{\sqrt{\pi_u + \sigma_u/C}}|u\rangle, \tag{6}$$

and let $\mathsf{R}(\sigma)$ be its implementation cost (again including controlled/inverse versions). We define $\mathsf{U}(\sigma) = \mathsf{U} + \mathsf{R}(\sigma)$. With these costs, we can describe Belovs' framework as follows.

Theorem 12 (Electric network framework [Bel13])**.** Let P be any reversible Markov chain on a finite state space X, $M \subset X$ a marked set, σ a distribution on X, and C a known upper bound on $C_{\sigma,M}$. Then there is a quantum algorithm that decides if $M \neq \emptyset$ with bounded error in complexity

$$\mathcal{O}\Big(\mathsf{S}(\sigma) + \sqrt{C}(\mathsf{U}(\sigma) + \mathsf{C})\Big).$$

Recall that if $\sigma = \pi$ then $C = HT(P, M)$. In addition, $\mathsf{R}(\sigma)$ becomes trivial in this case, so that this recovers the hitting time framework (except that it only *detects* marked elements).

4 Finding in the electric network framework

The major drawback of the electric network framework is that it only allows for detecting marked vertices, rather than actually finding one – in fact, if we only want to detect a marked vertex, the electric network framework is a strict generalization of the hitting time framework. In this section, we describe a quantum algorithm that reproduces the electric network framework, but in addition actually *finds* marked elements, making it a strict generalization of the hitting time framework.

Recall that we are given a weighted graph G over X with total edge weight W, a distribution σ over X, and a subset $M \subset X$ of marked elements. The quantity $R_{\sigma,M}$ denotes the effective resistance from σ to M, and we define $C_{\sigma,M} = WR_{\sigma,M}$. We will prove the following.

Theorem 13 (Electric network framework)**.** Let P be any reversible Markov chain on a finite state space X, $M \subset X$ a marked set, σ a distribution on X, and C a known upper bound on $C_{\sigma,M}$. There is a quantum algorithm that finds a marked element from M, or decides that it is empty, with bounded error in complexity

$$\mathcal{O}\Big(\sqrt{\log C}\,\mathsf{S}(\sigma) + \sqrt{C \log C \log\log C}(\mathsf{U}(\sigma) + \mathsf{C})\Big).$$

In earlier work, Belovs [Bel13] showed that it is possible to detect the presence of marked vertices in $\mathcal{O}\left(\sqrt{C}\right)$ quantum walk steps (Theorem 12). Our work strengthens this result by also finding a marked element, at the cost of an additional log factor. Our algorithm and analysis runs along the same lines as [AGJK19]. As described in Section 4.2, the algorithm combines quantum fast-forwarding with a quantum walk derived from an interpolated Markov chain. The analysis, described in Section 4.2.3 reduces the success probability of our quantum algorithm to the probability that a classical random walk starting from σ hits M, and then returns to the support of σ, within $\mathcal{O}(C_{\sigma,M})$ steps of the walk, similar to the analysis in [AGJK19]. In order to lower bound this quantity, we extend the known combinatorial interpretation of the quantity $C_{\sigma,M}$ from the case where σ is a singleton (Theorem 4) to a more general setting. This is described in Section 4.1.

Remark 14. A similar result, but restricted to the special case where the graph is a tree, can be found in the quantum algorithm for backtracking by Montanaro [Mon18]. Starting from the root of a binary tree, this algorithm incurs an additional log factor for actually finding a solution, rather than simply detecting one. The extension however crucially relies on the tree structure of the graph, essentially performing a binary search, and hence seems restricted to this special class of graphs.

4.1 Combinatorial interpretation of $C_{S,M}$ and $C_{\sigma,M}$

In Theorem 4 we mentioned the classic result that for any vertex s and subset M, the electric quantity $C_{s,M} = WR_{s,M}$ equals the commute time $\mathbb{E}_s(\tau_s^M)$ from s to M. A similar interpretation however seems to be lacking for the more general quantity $C_{\sigma,M} = WR_{\sigma,M}$. While one could expect that a similar relation should hold, at least for the special case where $\sigma = \pi|_S$ equals the stationary distribution on some subset S, we provide a counterexample in Appendix A. There we show that in certain cases $C_{\pi|_S,M} = WR_{\pi|_S,M}$ is not equal to the commute time $\mathbb{E}_{\pi|_S}(\tau_S^M)$ from $\pi|_S$ to M and back to S.

Nevertheless, we do succeed in proving a one-way bound, showing that a variant of the commute time can indeed be bounded by the electric quantity $C_{\pi|_S,M}$, which will prove sufficient for our purpose.

Claim 15. Let $S \subseteq X \setminus M$, and $p \in \mathbb{R}, T \in \mathbb{N}$ such that $\frac{2}{T} \leq \pi(S)p \leq 1/C_{S,M}$. Then with probability at least $p/2$ the random walk started from $\sigma = \pi|_S$ first hits M, and then returns to S, in the first T steps.

The conditions on σ might seem a bit strong, we can however adapt any graph, similar to what is implicitly done in Belovs' algorithm [Bel13], to ensure that they do hold for $p = \frac{1}{2}, T = 4C_{\sigma,M}$.

The main technical contribution in the proof of Claim 15 is the following lemma, which we prove in Appendix B. We let τ_M denote the hitting time of M, which is the random variable representing the number of steps to reach M (i.e., the minimum i such that $Y_i \in M$), and τ_S^+ the *first return time* to S (i.e., the minimum $i > 0$ such that $Y_i \in S$).

Lemma 16. Let $S, M \subseteq X$ be disjoint sets, then

$$\Pr_{\pi|_S}(\tau_M < \tau_S^+) = \frac{1}{C_{S,M}\pi(S)} \left(\geq \frac{1}{C_{\pi|_S,M}\pi(S)} \right).$$

This generalizes the classic fact that the probability that a reversible Markov chain starting at s visits t before returning to s is $1/(C_{s,t}\pi_s)$. We can then combine this with the following classic lemma, a proof of which can be found in [LPW17, Lemma 21.13].

Lemma 17 (Kac's Lemma)**.** For any irreducible and reversible Markov chain it holds that

$$\mathbb{E}_{\pi|_S}(\tau_S^+) = \frac{1}{\pi(S)}.$$

The proof of the main claim then easily follows.

Proof of Claim 15. We use a union bound on the events that $\tau_M < \tau_S^+$ and $\tau_S^+ < T$, the union of which implies the claimed statement. From Lemma 16 we know that $\Pr_\sigma(\tau_M < \tau_S^+) = 1/(C_{S,M}\pi(S)) \geq p$. A bound on $\Pr_\sigma(\tau_S^+ < T)$ easily follows from Kac's lemma (Lemma 17). Combined with Markov's inequality this lemma implies that

$$\Pr_\sigma(\tau_S^+ < T) \geq \Pr(\tau_S^+ < 2/(\pi(S)p)) > 1 - p/2.$$

The claim then follows by a union bound:

$$\Pr_\sigma((\tau_M < \tau_S^+) \wedge (\tau_S^+ < T)) \geq \Pr_\sigma(\tau_M < \tau_S^+) + \Pr_\sigma(\tau_S^+ < T) - 1 > p/2. \qquad \square$$

4.2 Quantum walk algorithm

Our algorithm combines quantum fast-forwarding with interpolated quantum walks. Similar to [AGJK19], the analysis of the algorithm then follows from a "box-stretching" argument, which builds on our combinatorial Claim 15. To ensure the conditions of this claim, we will consider an interpolated walk on a slightly modified graph, implicitly used in [Bel13].

4.2.1 Modified graph and Belovs' quantum walk

The input to the search problem is a weighted graph $G = (X, E, w)$, a subset of marked elements $M \subset X$ and an initial distribution σ over X. We assume access to these through black-box operations $\texttt{Check}(M)$, $\texttt{Setup}(\sigma)$, and the operator $\Lambda(\sigma, C)$, as defined in Section 3.4, with C an upper bound on $C_{\sigma,M}$.

We will consider a slightly modified graph $G' = (X', E', w')$, with total weight W', initial distribution σ' and marked elements M'. In Lemma 19 we will prove that we can implement a quantum walk on G' using only the original black-box operations mentioned earlier. The graph G' essentially consists of two copies of the original graph, and is defined by

$$X' = \{0,1\} \times X,$$
$$E' = \{((0,u),(0,v)),\ (u,v) \in E\} \ \cup\ \{((0,u),(1,u)),\ u \in \mathrm{supp}(\sigma)\},$$
$$w'_{(0,u),(0,v)} = w_{u,v},\ \forall (u,v) \in E, \qquad w'_{(0,u),(1,u)} = \sigma_u W/C,\ \forall u \in X.$$

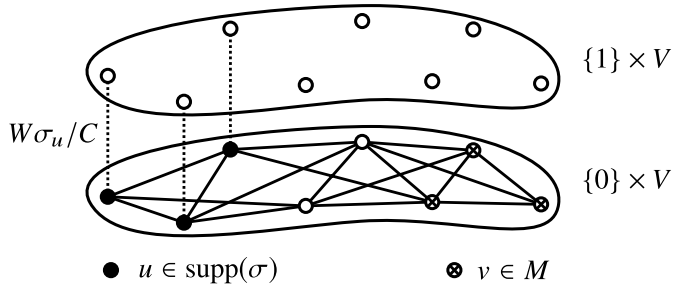

Figure 1: Modified graph.

We illustrate this construction in Figure 1 below. The modified initial state σ' is defined as $\sigma'_{(1,u)} = \sigma_u$, and zero elsewhere, and the marked elements $M' \subseteq V'$ correspond to the set $\{0\} \times M$.

These particular choices ensure that (i) $S' := \mathrm{supp}(\sigma')$ is disjoint from M', (ii) σ' is proportional to π' on its support, with π' the stationary distribution on G', and (iii) the commute time between σ' and M' in G' does not increase too much. We prove the second and third points in the following lemma, letting $R'_{\sigma',M'}$ and $C'_{\sigma',M'} = R'_{\sigma',M'}W'$ denote the effective resistance and commute time, respectively, between σ' and M' on G'. We also use the fact that $W' = W + 2W\sum_u \sigma_u/C = W + 2W/C$.

Lemma 18. We have $\pi'(S') = 1/(C+2)$, moreover if ρ is a probability distribution on $S = \mathrm{supp}(\sigma)$ such that $\sum_{u\in S} \frac{\rho_u^2}{\sigma_u} = 1/p$, then $C'_{\rho',M'} = (C_{\rho,M}/C + 1/p)(C+2)$. Therefore,

$$\frac{1}{C'_{S',M'}\pi'(S')} \geq \frac{1}{2}\min\left\{p, \frac{C}{C_{\rho,M}}\right\}.$$

Proof. Since $W' = W + 2W/C$, and $w'_{(1,u)} = W\sigma_u/C$, we get that

$$\pi'_{(1,u)} = \frac{w'_{(1,u)}}{W'} = \frac{W\sigma_u}{W'C} = \frac{\sigma_u}{(1+2/C)C} = \frac{\sigma_u}{(C+2)} \quad\Longrightarrow\quad \pi'(S') = \frac{1}{(C+2)}.$$

Now observe that

$$R'_{\rho',M'} = R_{\rho,M} + \sum_u \frac{\rho_u^2}{W\sigma_u/C} = R_{\rho,M} + \frac{C}{Wp},$$

yielding

$$C'_{\rho',M'} = W'R'_{\rho',M'} = (W + 2W/C)(R_{\rho,M} + C/(Wp)) = (C+2)(C_{\rho,M}/C + 1/p).$$

Finally,

$$\frac{1}{C'_{S',M'}\pi'(S')} \geq \frac{1}{C'_{\rho',M'}\pi'(S')} = \frac{1}{C_{\rho,M}/C + 1/p} \geq \frac{1}{2\max\{C_{\rho,M}/C, 1/p\}} = \frac{1}{2}\min\{p, C/C_{\rho,M}\}.$$

□

Now we apply the above lemma with $\rho = \sigma$, which gives $p = 1$. If $C \in \Theta(C_{\sigma,M})$ and $C_{\sigma,M} = \Omega(1)$, then we see that $C'_{\sigma',M'} \in \Theta(C_{\sigma,M})$, so the commute time does not increase significantly. Moreover, if $C \geq C_{\sigma,M}$, then for $T := \lceil 4C'_{\sigma',M'} \rceil \leq 8(C+2)+1$ we have

$$\frac{2}{T} \leq \pi'(S')/2 \leq \frac{1}{C'_{\sigma',M'}}.$$

In this case, all conditions of Claim 15 are satisfied with respect to $p' = \frac{1}{2}$.

Finally, we show that we can efficiently implement a quantum walk corresponding to the random walk P' on G':

Lemma 19. The cost of $\texttt{Update}(P')$ is $\mathsf{U}(P') \in \mathcal{O}(\mathsf{U}(P) + \mathsf{R}(\sigma)) = \mathcal{O}(\mathsf{U}(\sigma))$, where $\mathsf{U}(P)$ is the cost of implementing the original quantum walk operator $W(P)$, and $\mathsf{R}(\sigma)$ is the cost of $\Lambda(\sigma, C)$, defined in Section 2.3.

Proof. In the following we will identify P with $|0\rangle\langle 0| \otimes P$ (and similarly for $D(P)$). Observe that

$$P' = \operatorname{diag}\left(\frac{\pi_u}{\pi_u + \sigma_u/C}\right) P + \sum_{u \in \operatorname{supp}(\sigma)} \frac{\sigma_u/C}{\pi_u + \sigma_u/C} (|0,u\rangle\langle 1,u| + |1,u\rangle\langle 0,u|),$$

and therefore $D(P') = \sqrt{P' \circ P'^T}$ expressed in terms of $D(P) = \sqrt{P \circ P^T}$ is

$$\begin{aligned} D(P') = &\operatorname{diag}\left(\sqrt{\frac{\pi_u}{\pi_u + \sigma_u/C}}\right) D(P) \operatorname{diag}\left(\sqrt{\frac{\pi_u}{\pi_u + \sigma_u/C}}\right) \\ &+ \sum_u \sqrt{\frac{\sigma_u/C}{\pi_u + \sigma_u/C}} (|0,u\rangle\langle 1,u| + |1,u\rangle\langle 0,u|). \end{aligned}$$

As in (6) let $\Lambda(\sigma, C)$ be a unitary that uses a single-qubit "flag" register a and acts as

$$|\bar{0}\rangle_{A'} |u\rangle_X \mapsto \frac{\sqrt{\pi_u}|\bar{0}\rangle_a + \sqrt{\sigma_u/C}|1\rangle_a}{\sqrt{\pi_u + \sigma_u/C}} |u\rangle_X.$$

We will use a new qubit register b, and represent the vertices as $(0,u) \mapsto |0\rangle_b |u\rangle$, $(1,u) \mapsto |1\rangle_b |u\rangle$. We will denote by $\bar{c}_{ab} W(P)$ the operator $W(P)$ conditioned on the qubit state $|0\rangle_a |0\rangle_b$, and by $\bar{c}_b \Lambda$ the operator $\Lambda(\sigma, C)$ conditioned on the qubit state $|0\rangle_b$, and using a as the output qubit. Let

$$W(P') := (I_A \otimes (\bar{c}_b \Lambda^\dagger)) \bar{c}_{ab} W(P) (I_A \otimes (\mathrm{SWAP}_{ab} \otimes I_X) \bar{c}_b \Lambda),$$

then

$$\begin{aligned} (\langle 0|_a \otimes I) W(P') (|0\rangle_a \otimes I) = |0\rangle\langle 0|_b \otimes &\left[\operatorname{diag}\left(\sqrt{\frac{\pi_u}{\pi_u + \sigma_u/C}}\right) W(P) \operatorname{diag}\left(\sqrt{\frac{\pi_u}{\pi_u + \sigma_u/C}}\right)\right] \\ &+ (|0\rangle\langle 1|_b + |1\rangle\langle 0|_b) \otimes I_A \otimes \left[\sum_u \sqrt{\frac{\sigma_u/C}{\pi_u + \sigma_u/C}} |u\rangle\langle u|\right], \end{aligned}$$

and therefore $W(P')$ is a walk operator of $D(P')$ whenever $W(P)$ is a walk operator of $D(P)$. □

4.2.2 Interpolated walk and algorithm

The transformation described in Section 4.2.1 can be applied to any input graph G and any distribution σ, with negligible impact on $C_{\sigma,M}$, or the update cost U. This justifies focusing our analysis on the case where $\sigma = \pi|_S$ and $1/C_{\sigma,M} \le \pi(S) \le 2/C_{\sigma,M}$, ensuring that the conditions of Claim 15 are satisfied for $p = 1/2$ and $T \ge 4C_{\sigma,M}$. Moreover, this ensures that M and $S = \operatorname{supp}(\sigma)$ are disjoint, and furthermore, that we can easily reflect around $\operatorname{supp}(\sigma)$.[7] We assume these conditions for the remainder of this section.

For a Markov chain P, and parameter $q = (q_S, q_M) \in [0,1)^2$, we consider the interpolated Markov chain $P(q)$, defined by (here δ_{uv} is the Kronecker delta)

$$P(q)_{u,v} = \begin{cases} (1 - q_S) P_{u,v} + q_S \delta_{uv} & \text{if } u \in \operatorname{supp}(\sigma) \\ (1 - q_M) P_{u,v} + q_M \delta_{uv} & \text{if } u \in M \\ P_{u,v} & \text{else.} \end{cases}$$

[7] To see the second point, note that in the modified graph of Section 4.2.1 this amounts to reflecting around the states whose first register is 1.

Equivalently,

$$P(q) = (1 - q_S - q_M)P + q_S P_{\mathrm{supp}(\sigma)} + q_M P_M, \tag{7}$$

with $P_{\mathrm{supp}(\sigma)}$ and P_M absorbing walks, as defined in Section 2.2. We denote by $D(q)$ the discriminant matrix of $P(q)$. Starting from the state $|\sqrt{\sigma}\rangle$, our algorithm will apply T steps of quantum fast-forwarding of $P(q)$ for appropriately chosen q, and increasing T.

Algorithm 1 Fast-forwarding-based search algorithm

Search(P, σ, M, T).

1. Let $Q = \left\{1, 2^{-1}, 2^{-2}, \ldots, 2^{-\lceil \log(14T) \rceil}\right\}$, and prepare the state

$$|\psi\rangle = \sum_{t \in [T]} \sum_{q_M \in Q} \frac{1}{\sqrt{T|Q|}} |t\rangle |q = (1 - T/2, 1 - q_M)\rangle |\sqrt{\sigma}\rangle.$$

2. Let U be the operator that applies quantum fast-forwarding, controlled on the first two registers, mapping $|t\rangle|q\rangle|\sqrt{\sigma}\rangle$ to $|1\rangle|t\rangle|q\rangle D^t(q)|\sqrt{\sigma}\rangle + |0\rangle|\Gamma\rangle$ for some arbitrary $|\Gamma\rangle$, with precision $\mathcal{O}\left(\frac{1}{\log T}\right)$.

3. Apply $\mathcal{O}\left(\sqrt{\log T}\right)$ rounds of amplitude amplification on U and $|\psi\rangle$, conditioned on the first register. Finally, measure the last register.

In Lemma 25 of Section 4.2.3 we will prove that there exists some $T' \in \mathcal{O}(C_{\sigma,M})$, such that for all $T'' \geq T'$ the algorithm returns a marked vertex with constant probability. Combined with the following lemma, which bounds the complexity of the algorithm, this proves Theorem 13.

Lemma 20. The complexity of Algorithm 1 is

$$\mathcal{O}\left(\sqrt{\log T}\,\mathsf{S}(\sigma) + \sqrt{T \log T \log\log T}\,(\mathsf{U}(\sigma) + \mathsf{C})\right).$$

Proof. For the complexity of step 1., note that creating $|\psi\rangle$ only requires $\mathcal{O}(\log T)$ elementary gates, and a call to $\texttt{Setup}(\sigma)$ costing $\mathsf{S}(\sigma)$. For the complexity of step 2., note that by Theorem 7, the operators U and $U^\dagger$ require $\mathcal{O}\left(\sqrt{T \log\log T}\right)$ calls to $\texttt{Update}(P(q))$, which implements a block-encoding of $W(q) = W(P(q))$. We can implement such a block-encoding, using a block-encoding of P, $W(P)$, which, by assumption (see also Lemma 19), can be implemented in $\mathcal{O}(\mathsf{U}(\sigma))$ complexity; the operation $\texttt{Check}(M)$, costing C; and an analogous operation that checks if a vertex is in $S = \mathrm{supp}(\sigma)$, which can be done in $\mathcal{O}(1)$ cost, by our assumptions on the structure of G (such an implementation of $W(q)$ is straightforward, as we discuss in Section 2.4). Thus, the total cost of step 2. is $\mathcal{O}\left(\sqrt{T \log\log T}(\mathsf{U}(\sigma) + \mathsf{C})\right)$.

By [BHMT02] we can implement step 3. using $\mathcal{O}\left(\sqrt{\log T}\right)$ reflections around $U|\psi\rangle$. A single such reflection can be implemented by $\mathcal{O}(1)$ calls to U and $U^\dagger$, and the preparation circuit of $|\psi\rangle$ and its inverse – yielding a total complexity (neglecting constants):

$$\sqrt{\log T}\left(\mathsf{S}(\sigma) + \sqrt{T \log\log T}(\mathsf{U}(\sigma) + \mathsf{C})\right) = \sqrt{\log T}\mathsf{S}(\sigma) + \sqrt{T \log T \log\log T}(\mathsf{U}(\sigma) + \mathsf{C}). \quad \square$$

4.2.3 Correctness of Algorithm 1

Similar to the argument in [AGJK19], we use a careful choice of the parameters of the interpolated walk to ensure a constant success probability of Algorithm 1. As discussed in the previous section, we can assume without loss of generality that $\sigma = \pi|_S$ for some $S \subseteq X$, and that $1/C_{\sigma,M} \leq \pi(S) \leq 2/C_{\sigma,M}$.

The key quantity is $\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2$, where Π_M is the orthogonal projector onto marked vertices – this is the probability of finding a marked vertex in step 2. of the algorithm (that is, before amplitude amplification). We can bound this quantity in terms of the classical Markov chain $P(q)$ as follows, generalizing [AGJK19, Lemma 8]:

Lemma 21. Let $(Y_i(q))_{i=0}^{\infty}$ be a Markov chain evolving according to $P(q)$ with initial state $Y_0(q)$ distributed according to $\sigma = \pi|_S$, and let $D(q)$ be the associated discriminant matrix. Then for any $t, t' \in \mathbb{N}$ such that $t' > t$ we have that

$$\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\| \geq \Pr_{Y_0(q)\sim\sigma}\big(Y_t(q) \in M, Y_{t'}(q) \in S\big).$$

We will not use this lemma directly, but we state it here for the sake of intuition. Its proof closely follows that of [AGJK19, Lemma 8] (see also the proof of Corollary 24).

By this lemma it suffices to show that with certain probability, for appropriate choice of t and t', the t-th vertex is marked and the t'-th vertex is again in the initial support S. We will be able to ensure these conditions by appropriately tweaking the parameters $q = (q_S, q_M)$.

The sequence of random variables $(Y_i)_{i=0}^{\infty}$ is supported on infinite sequences of vertices from X, which represent *paths* of the random walk. However, in light of Lemma 21, given such a sequence, we will only care about which states in the sequence are in S, and which are in M. Following a similar abstraction in [AGJK19], we model a path of the random walk as a sequence of boxes, with gray boxes representing vertices in S, black boxes representing vertices in M, and white boxes representing vertices in neither S nor M. We depict such a sequence of boxes in Figure 2. In a slight abuse of notation, we will refer to $y = (y_0, y_1, \dots)$ drawn from $(Y_i)_{i=0}^{\infty}$ as a sequence of boxes. A gray or black box at position i then denotes the event that $Y_i \in S$ or $Y_i \in M$, respectively. The indicated times ht and ct in Figure 2 denote the random variables corresponding to the hitting time (first time to reach M) and commute time (first time to return to S after reaching M), respectively.

Figure 2: Sequence of boxes $y = (y_0, y_1, \dots)$ drawn from the random variable $(Y_i)_{i=0}^{\infty}$. Gray boxes, called S-boxes, correspond to $y_k \in S$; black boxes, called M-boxes, correspond to $y_k \in M$. The hitting time from S to M is denoted by ht, the commute time by ct.

In the following we define $r_S = 1/(1 - q_S)$ and $r_M = 1/(1 - q_M)$, representing the expected number of steps that the interpolated walk remains at a vertex in S or M respectively. Given parameters $r = (r_S, r_M)$ and a sequence of boxes $y = (y_0, y_1, \dots)$, we denote by $\gamma^{(r)}$ the sequence derived from y by replacing each M-box with r_M M-boxes, and each S-box with r_S S-boxes. Since r_M is the expected number of steps a walker would stay at a marked vertex in an absorbing walk with parameter q_M (and similarly for r_S), $\gamma^{(r)}$ loosely models a path of the interpolated random walk $P(q)$. We denote by $\mathrm{ht}^{(r_S,r_M)}$ and $\mathrm{ct}^{(r_S,r_M)}$ the hitting time and commute time of the sequence $\gamma^{(r)}$.

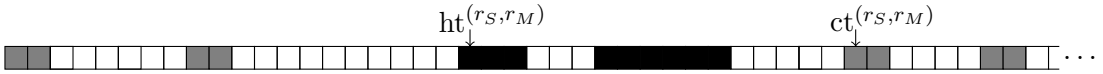

Figure 3: If y is the sequence of boxes shown in Figure 2, then the above sequence represents $\gamma^{(r_S,r_M)}$ for $r_S = 2$ and $r_M = 3$.

For integers $a < b$, we will be interested in bounding the quantities

$$M_y^{(r)}[a,b] = |\{t \in [a,b] : \gamma_t^{(r)} \in M\}| \quad \text{and} \quad S_y^{(r)}[a,b] = |\{t \in [a,b] : \gamma_t^{(r)} \in S\}|.$$

When $r = (1,1)$ we will omit the (r) superscript.

Lemma 22. Let $y = (y_0, y_1, \dots, y_T, \dots)$ denote a sequence of boxes, for some $T \in \mathbb{N}$, such that

- $y_0 \in S$,
- $\mathrm{ct} \le T$,
- $S_y[0, \mathrm{ht}] = 1$.

If we set $r_S = T/2$, then there exists $r_M \in R = \{1, 2, 4, \dots, 2^{\lceil \log(14T) \rceil}\}$ such that

$$M_y^{(r)}[0, 2T] \ge T/2 \quad \text{and} \quad S_y^{(r)}[7T, 15T] \ge T/4.$$

We note that $R = \{1/q_M : q_M \in Q\}$ for Q the set defined in Algorithm 1.

Proof. We first choose $r_M \in \{1, 2, 4, \dots, 2^{\lceil \log(14T) \rceil}\}$ such that $7T \le \mathrm{ct}^{(1,r_M)} \le 14T$. To see that this is possible, note that $\mathrm{ct}^{(1,1)} = \mathrm{ct} \le 14T$ by assumption, and increasing r_M strictly increases $\mathrm{ct}^{(1,r_M)}$, so there is some largest possible r_M such that $\mathrm{ct}^{(1,r_M)} \le 14T$. Since doubling r_M can at most double $\mathrm{ct}^{(1,r_M)}$, we must also have $7T \le \mathrm{ct}^{(1,r_M)}$.

The increase in the commute time when transforming y to $\gamma^{(1,r_M)}$, given by $\mathrm{ct}^{(1,r_M)} - \mathrm{ct}$, comes from adding $\mathrm{ct}^{(1,r_M)} - \mathrm{ct}$ new M-boxes to $\gamma^{(1,r_M)}$ before $\mathrm{ct}^{(1,r_M)}$, meaning that

$$M_y^{(1,r_M)}[0, \mathrm{ct}^{(1,r_M)}] \ge \mathrm{ct}^{(1,r_M)} - \mathrm{ct} \ge \mathrm{ct}^{(1,r_M)} - T.$$

Note that for any positive integer k, we have $M_y^{(1,r_M)}[0, \mathrm{ct}^{(1,r_M)} - k] \ge M_y^{(1,r_M)}[0, \mathrm{ct}^{(1,r_M)}] - k$. Setting $k = \mathrm{ct}^{(1,r_M)} - 3T/2$ thus implies that

$$M_y^{(1,r_M)}[0, 3T/2] \ge T/2.$$

Next we choose $r_S = T/2$. By the conditions $S_y[0, \mathrm{ht}] = 1$ and $y_0 \in S$, the unique S-box in y before ht is y_0, so the second S-box in y is at ct, and similarly, the second S-box in $\gamma^{(1,r_M)}$ is at $\mathrm{ct}^{(1,r_M)}$. Thus, there are $r_S - 1 = T/2 - 1$ new boxes added to $\gamma^{(1,r_M)}$ before $\mathrm{ct}^{(1,r_M)}$ to get $\gamma^{(r_S,r_M)}$, meaning that the $\ge T/2$ M-boxes in the first $3T/2$ boxes of $\gamma^{(1,r_M)}$ must all occur within the first $2T$ boxes of $\gamma^{(r_S,r_M)}$, so

$$M_y^{(r_S,r_M)}[0, 2T] \ge T/2.$$

Similarly, since the first S-box in $\gamma^{(1,r_M)}$ is extended to $T/2$ S-boxes to get to $\gamma^{(r_S,r_M)}$, the second S-box is displaced by $T/2 - 1$, meaning $\mathrm{ct}^{(r_S,r_M)} = \mathrm{ct}^{(1,r_M)} + T/2 - 1$, so

$$15T/2 - 1 \le \mathrm{ct}^{(r_S,r_M)} \le 29T/2 - 1.$$

In $\gamma^{(r_S,r_M)}$, there is a sequence of $r_S = T/2$ S-boxes beginning at $\mathrm{ct}^{(r_S,r_M)}$, so we have:

$$S_y^{(r_S,r_M)}[7T, 15T] \ge T/2. \qquad \square$$

For a fixed Markov process P, and parameters $r = (r_S, r_M)$, let $(Y_i^{(r)})_{i=0}^{\infty}$ be the Markov chain evolving according to $P(q)$, the absorbing chain with $q_S = 1 - \frac{1}{r_S}$ and $q_M = 1 - \frac{1}{r_M}$ (i.e., $Y^{(r)} = Y(q)$).

For a sequence y, for any choice of $r = (r_S, r_M)$, we associate two slightly different sequences of boxes to y:

- As previously defined, $\gamma^{(r)}$ is derived from y by replacing each M- resp. S-box with a fixed number of copies r_M resp. r_S.
- $y^{(r)}$ is derived from y by replacing each M- resp. S-box with an independently random number of copies that are geometrically distributed with mean r_M resp. r_S. Note that if y is distributed according to the random variable $(Y_i)_{i=0}^{\infty}$, then $y^{(r)}$ is distributed according to $(Y_i^{(r)})_{i=0}^{\infty}$.

Lemma 22 allows us to say something about the probability that $\gamma_t^{(r)} \in M$ and $\gamma_{t'}^{(r)} \in S$ under uniform random $r \in R$, $t \in [0, 2T]$, and $t' \in [7T, 15T]$, assuming certain conditions on y are satisfied (we will shortly argue that these conditions are satisfied with high probability when starting from the distribution $\sigma = \pi|_S$). Since we are actually concerned with proving statements about the absorbing walk, we would like to say something similar for $y^{(r)}$ in place of $\gamma^{(r)}$. Fortunately, these two sequences are very similar, leading to the following corollary of Lemma 22.

Corollary 23. Set $r_S = T/2$, and let $r_M \in R$, $t \in \{1, \dots, 30T\}$ and $t' \in \{1, \dots, 30T\}$ be chosen uniformly at random. Let $Y = (Y_i)_{i=0}^{\infty}$ be the Markov chain evolving according to Markov process P (not necessarily reversible) starting in any distribution κ supported on S, with $Y^{(r)}$ the absorbing Markov chain, coupled to Y, as defined above. Let E be the event that: $\mathrm{ct} \leq T$ and $S_Y[0, \mathrm{ht}] = 1$, where ht and ct are the hitting time and commute time in Y. Then

$$\mathbb{E}_{t,t',r_M}\left(\Pr_{Y_0^{(r)} \sim \kappa}(Y_t^{(r)} \in M, Y_{t+t'}^{(r)} \in S \mid E)\right) = \Omega(\log^{-1} T).$$

Proof. By Lemma 22, we know that if E holds then there exists some $r_M \in R$, chosen with probability $1/|R| = \Omega(\log^{-1} T)$, such that with probability 1 we have

$$|\{i \in [0, 2T] : \gamma_i^{(r)} \in M\}| = M_Y^{(r)}[0, 2T] \geq T/2$$

and

$$|\{i \in [7T, 15T] : \gamma_i^{(r)} \in S\}| = S_Y^{(r)}[7T, 15T] \geq T/4.$$

We will show that for *any* y in the support of Y such that E holds, given this choice of r, $\Pr_{t,t'}(y_t^{(r)} \in M, y_{t+t'}^{(r)} \in S)$ is constant, completing the proof. Using the similarities between $y^{(r)}$ and $\gamma^{(r)}$, we now wish to lower bound $|\{i \in [0, 7T/2] : y_i^{(r)} \in M\}|$ and $|\{i \in [7T/2 + 1, 30T] : y_i^{(r)} \in S\}|$.

In going from y to $\gamma^{(r)}$, we replace each S-box with a block of r_S S-boxes, whereas when going from y to $y^{(r)}$, we replace each S-box with a block of S-boxes whose length is geometrically distributed with mean r_S (and similarly for M-boxes). We can equivalently derive $y^{(r)}$ from $\gamma^{(r)}$ by replacing each block of r_S S-boxes with a block of S-boxes whose length is given by a geometric sample. Define $\overline{r}_S[a, b]$ to be the mean over all such geometric samples replacing an S-block of $\gamma^{(r)}$ that has non-trivial overlap with the interval $[a, b]$. Define $\overline{r}_M[a, b]$ similarly.

We will assume the following 6 conditions:

$$\begin{gathered}\overline{r}_M[1, 2T],\ \overline{r}_M[1, 7T],\ \overline{r}_M[7T, 15T] \in [r_M/2, 2r_M], \\ \overline{r}_S[1, 2T],\ \overline{r}_S[1, 7T],\ \overline{r}_S[7T, 15T] \in [r_S/2, 2r_S].\end{gathered} \tag{8}$$

We use [AGJK19, Lemma 10], which states that any sum of i.i.d. geometric random variables will be within a factor 2 of its expected value with probability at least 7/16. While for instance the sums $\overline{r}_M[1, 2T]$ and $\overline{r}_M[1, 7T]$ are not strictly independent, clearly the event that $\overline{r}_M[1, 2T] \in [r_M/2, 2r_M]$ cannot decrease the probability that $\overline{r}_M[1, 7T] \in [r_M/2, 2r_M]$. Hence the probability that all 6 conditions hold is at least $(7/16)^6$, and we will assume that this is the case for the rest of the proof.

Since $r_S/2 \leq \overline{r}_S[0, 2T] \leq 2r_S$ and $r_M/2 \leq \overline{r}_M[0, 2T] \leq 2r_M$, it holds that

$$|\{i \in [0, 7T/2] : y_i^{(r)} \in M\}| \geq M_y^{(r)}[0, 2T]/2 \geq T/4. \tag{9}$$

To see this, notice that in the interval $[0, 2T]$ of $\gamma^{(r)}$, we remove at most half of the $\geq T/2$ M-boxes to get $y^{(r)}$, so there are at least $T/4$ remaining. We at most double the $\leq 3T/2$ S-boxes, so the remaining $\geq T/4$ M-boxes from $\gamma^{(r)}$'s interval $[0, 2T]$ all appear within the first $2T + 3T/2 = 7T/2$ of $y^{(r)}$. From (9), we immediately have

$$|\{i \in [1, 30T] : y_i^{(r)} \in M\}| \geq T/4,$$

so the probability that a uniform random $t \in \{1, \dots, 30T\}$ satisfies $y_t^{(r)} \in M$ is at least $1/120$.

Next, since $\overline{r}_M[0, 7T], \overline{r}_M[7T, 15T] \in [r_M/2, 2r_M]$ and $\overline{r}_S[0, 7T], \overline{r}_S[7T, 15T] \in [r_S/2, 2r_S]$,

$$|\{i \in [7T/2, 30T] : y_i^{(r)} \in S\}| \geq |\{i \in [7T, 15T] : \gamma_i^{(r)} \in S\}|/2 = S_y^{(r)}[7T, 15T]/2 \geq T/8. \quad (10)$$

To see this, note that in the interval $[7T, 15T]$ of $\gamma^{(r)}$, we remove at most half of the $\geq T/4$ S-boxes to get $y^{(r)}$, so there are at least $T/8$ remaining, although they might no longer be contained within the interval $[7T, 15T]$. However, since the position of any element in $y^{(r)}$ is at least half and at most double its position in $\gamma^{(r)}$, these $T/8$ S-boxes will be within the interval $[7T/2, 30T]$ of $y^{(r)}$.

From (10), we want to conclude that $\Pr_{t,t'}(y_{t+t'}^{(r)} \in S | y_t^{(r)} \in M)$ is constant. In fact, by (9), we have that with constant probability $t \leq 7T/2$ *and* $y_t^{(r)} \in M$. Hence it is sufficient to lower bound $\Pr_{t,t'}(y_{t+t'}^{(r)} \in S | t \leq 7T/2, y_t^{(r)} \in M)$ by a constant. To do so, we note that for *any* $t \in [1, \dots, 7T/2]$ the range of possible values of $t + t'$ contains $[7T/2, 30T]$. Hence, by (10),

$$\Pr_{t,t'}(y_{t+t'}^{(r)} \in S | t \leq 7T/2, Y_t^{(r)} \in M) \geq (T/8)/(30T) = 1/240.$$

This bound only holds conditioned on the events in (8), but we already argued that these also hold with constant probability. □

This corollary allows us to prove the following statement.

Corollary 24. Set $r_S = T/2$, and let $r_M \in R$ and $t \in \{1, \dots, 30T\}$ be chosen uniformly at random. Let $q_S = 1 - \frac{1}{r_S}$ and $q_M = 1 - \frac{1}{r_M}$, and let $D(q)$ be the discriminant of $P(q)$ for a reversible ergodic Markov process P on X with stationary distribution π, and $\sigma = \pi|_S$ for some $S \subset X$ with $1/C_{\sigma,M} \leq \pi(S) \leq 2/C_{\sigma,M}$. If $T \geq 4C_{\sigma,M}$ then

$$\mathbb{E}_{t,r_M}\left[\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2\right] \in \Omega(\log^{-1} T).$$

Proof. Let $Y^{(r)} = Y(q)$ be the Markov chain of the absorbing walk $P(q)$ starting from the distribution σ. Then we can define Y as the Markov chain evolving according to P, coupled to $Y^{(r)}$ as above. That is, Y follows the same sequence as $Y^{(r)}$, except that it omits repeated elements that result from using the absorbing self-edges.

Let E denote the event that $\mathrm{ct} \leq T$, where ct is the commute time of Y, and $S_Y[0, \mathrm{ht}] = 1$. Then by Claim 15 we know that E holds with probability at least $1/4$. Combined with Corollary 23, taking t' uniformly at random from $\{1, \dots, 30T\}$, this allows us to conclude

$$\mathbb{E}_{t,t',r}\left(\Pr_{Y_0^{(r)} \sim \sigma}(Y_t^{(r)} \in M, Y_{t'}^{(r)} \in S)\right) \geq \frac{1}{4}\Omega(\log^{-1} T). \quad (11)$$

Next, we compute:

$$\begin{aligned}
\mathbb{E}_{t,r_M}\left[\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2\right] &= \frac{1}{|R|}\sum_{r_M \in R}\frac{1}{30T}\sum_{t=1}^{30T}\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2 \\
&\geq \frac{1}{|R|}\sum_{r_M \in R}\left(\frac{1}{30T}\sum_{t=1}^{30T}\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|\right)^2 \\
&= \frac{1}{|R|}\sum_{r_M \in R}\frac{1}{30T}\sum_{t=1}^{30T}\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|\frac{1}{30T}\sum_{t'=1}^{30T}\left\|\Pi_M D^{t'}(q)|\sqrt{\sigma}\rangle\right\| \\
&\geq \frac{1}{|R|}\sum_{r_M \in R}\frac{1}{30T}\sum_{t=1}^{30T}\frac{1}{30T}\sum_{t'=1}^{30T}\left|\langle\sqrt{\sigma}|D^t(q)\Pi_M D^{t'}(q)|\sqrt{\sigma}\rangle\right|, \quad (12)
\end{aligned}$$

by the Cauchy-Schwarz inequality. Let π' denote the stationary distribution of $P(q)$. Since $P(q)$ is just a twice interpolated walk, its stationary distribution has the form

$$\pi' = a\pi|_S + b\pi|_M + c\pi|_{X\setminus(M\cup S)} = a\sigma + b\pi|_M + c\pi|_{X\setminus(M\cup S)},$$

for some positive constants a, b and c, whose precise values are not important for our purposes (see [KMOR16] for an analysis of the stationary distribution of interpolated walks). Furthermore, we observe that $D(q)^t = \mathrm{diag}(\pi')^{1/2}P(q)^t\mathrm{diag}(\pi')^{-1/2}$, so, since $|\sqrt{\sigma}\rangle$ is supported on S:

$$\begin{aligned}\langle\sqrt{\sigma}|D(q)^t\Pi_M &= \langle\sqrt{\sigma}|\mathrm{diag}(a\sigma)^{1/2}P(q)^t\mathrm{diag}(b\pi|_M)^{-1/2}\\ &= \sqrt{a/b}\sum_{u\in S}\sigma_u\langle u|P(q)^t\mathrm{diag}(\pi|_M)^{-1/2}.\end{aligned}$$

Similarly,

$$\Pi_M D(q)^{t'}|\sqrt{\sigma}\rangle = \sqrt{b/a}\mathrm{diag}(\pi|_M)^{1/2}P(q)^{t'}\sum_{u\in S}|u\rangle.$$

Thus,

$$\begin{aligned}\left|\langle\sqrt{\sigma}|D^t(q)\Pi_M D^{t'}(q)|\sqrt{\sigma}\rangle\right| &= \sum_{u\in S}\sigma_u\langle u|P(q)^t\mathrm{diag}(\pi|_M)^{-1/2}\mathrm{diag}(\pi|_M)^{1/2}P(q)^{t'}\sum_{u\in S}|u\rangle\\ &= \sum_{u\in S}\sigma_u\langle u|P(q)^t\Pi_M P(q)^{t'}\sum_{u\in S}|u\rangle\\ &= \mathrm{Pr}_{Y_0(q)\sim\sigma}(Y_t(q)\in M, Y_{t'+t}(q)\in S).\end{aligned}$$

Recall that $Y^{(r)} = Y(q)$. Thus continuing, from (12), and using (11), we have:

$$\begin{aligned}\mathbb{E}_{t,r_M}\left[\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2\right] &= \frac{1}{|R|}\sum_{r_M\in R}\frac{1}{30T}\sum_{t=1}^{30T}\frac{1}{30T}\sum_{t'=1}^{30T}\mathrm{Pr}_{Y_0(q)\sim\sigma}\big(Y_t(q)\in M, Y_{t'+t}(q)\in S\big)\\ &= \mathbb{E}_{t,t',r_M}\big(\mathrm{Pr}_{Y_0(q)\sim\pi_S}\big(Y_t(q)\in M, Y_{t+t'}(q)\in S\big)\big)\in\Omega(\log^{-1}T).\end{aligned}$$

□

From this corollary, we can straightforwardly prove our final lemma. Combined with Lemma 20 this proves Theorem 13.

Lemma 25. There exists $T'\in\mathcal{O}(C_{\sigma,M})$ such that, for all $T\geq T'$, Algorithm 1 returns a marked element with constant probability.

Proof. By the above Corollary 24 we know that for all $T\geq T'$, for some $T'\in\mathcal{O}(C_{\sigma,M})$, it holds that

$$\frac{1}{T}\sum_{t\in[T]}\frac{1}{|Q|}\sum_{q_M\in Q}\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2 = \frac{1}{T}\sum_{t\in[T]}\frac{1}{|R|}\sum_{r_M\in R}\left\|\Pi_M D^t(q)|\sqrt{\sigma}\rangle\right\|^2\in\Omega(\log^{-1}T).$$

As a consequence, measuring the state

$$|1\rangle\Big(\sum_{t\in[T]}\sum_{q_M\in Q}\frac{1}{\sqrt{T|Q|}}|t\rangle|q\rangle D^t(q)|\pi_S\rangle\Big) + |0\rangle|\Gamma\rangle$$

returns a marked element with probability $\Omega(\log^{-1}T)$. In step 2. of the algorithm we approximate this state up to sufficient precision $\mathcal{O}\big(\log^{-1}T\big)$. Applying $\mathcal{O}\big(\sqrt{\log T}\big)$ rounds of amplitude amplification then indeed suffices to retrieve a marked element with constant probability. □

4.3 A generalization of Belovs' algorithm

Now we sketch a generalization of Algorithm 1.

Corollary 26. Let σ, ρ be probability distributions on $S = \mathrm{supp}(\sigma) \subseteq X$, and $p := 1/\sum_{u \in S} \frac{\rho_u^2}{\sigma_u}$, then there is a quantum algorithm that finds a marked element from M in expected complexity

$$\mathcal{O}\Big(\sqrt{1/p}\,\Big[\log(C_{\rho,M})\mathsf{S}(\sigma) + \sqrt{C_{\rho,M}}(\mathsf{U}(\sigma) + \mathsf{C})\Big]\mathrm{polylog}(C_{\rho,M}/p)\Big).$$

Proof. Suppose that $pC_{\rho,M} \leq C \in \mathcal{O}(pC_{\rho,M})$, then by Lemma 18 and Claim 15 we have that the walk started from σ' on the modified graph first hits M' and then returns to S' with probability at least $p/4$ within the first $T = \lceil \frac{4}{p}(C+2) \rceil = \mathcal{O}(C_{\rho,M})$ steps. The analysis in Section 4.2.3 shows that Algorithm 1 finds a marked element with probability $\Omega(p)$ if we enhance the precision by a factor of $\sim p$ in step 2. After applying $\sqrt{1/p}$ additional rounds of amplitude amplification we find a marked element with probability $\Omega(1)$.

Finally note that we do not need to a priori know ρ or the values of p and $C_{\rho,M}$. We can do binary search to find multiplicative constant approximations of p and $C_{\rho,M}$, only incurring a logarithmic overhead and providing an expected runtime as claimed. □

Intuitively this improvement is somewhat analogous to the HT^+ to HT improvement in the complexity of finding marked elements using quantum walks [AGJK19]. There the HT^+ complexity corresponds to "fair" sampling of a marked vertex from $\pi|_M$, whereas here the runtime $\sqrt{C_{\sigma,M}}$ corresponds to the "democratic" requirement that M should be reachable from the entire σ – but one does not actually need to hit the marked set from everywhere! It is enough if we hit it with high probability from a large fraction of the initial states, cf. Lemma 16. Indeed, if $C = C_{\sigma|_Q,M}$ then for $\rho := \sigma|_Q$ we get $p = \sigma(Q)$, and so we get an efficient algorithm with runtime $\sim \sqrt{C}$ as long as p is not too small, while the quantity $\sqrt{C_{\sigma,M}}$ is less relevant.

To illustrate that this result can be helpful in some cases, we consider the following example. Suppose that G is a regular graph, with marked set M and hitting time HT. Suppose that we can remove an edge of G without affecting the hitting time much. Take, say 3 copies of G, and cyclically connect to each other the vertices adjacent to the removed edges, so that the graphs form a triangle, with a single edge between each pair of the copies of G. Suppose that we unmark the marked vertices of one copy, and set the weight of the three new edges very small, so that the hitting time in the new graph can be arbitrary large. If the vertices are also permuted there is no apparent structure left, and previous quantum walk algorithms seem to fail in finding or even detecting marked vertices faster than $\widetilde{\mathcal{O}}(\sqrt{\mathrm{HT}'})$, where HT' is the hitting time of the new graph. However, our algorithm can find a marked vertex in time $\widetilde{\mathcal{O}}(\sqrt{\mathrm{HT}})$. Thus our walk actually finds a marked vertex much faster than the hitting time $\widetilde{\mathcal{O}}(\sqrt{\mathrm{HT}'})$ of the new graph.

5 The MNRS framework and the electric network framework

In this section we describe our second main result, which is a quantum walk search algorithm that generalizes the MNRS framework, the hitting time framework, the controlled quantum amplification framework, and (our extension of) the electric network framework. It is summarized in the following theorem.

Theorem 27. For any reversible Markov chain P on state space X, any marked set $M \subset X$, any $t \in \mathbb{N}$, and any distribution σ on X, there is a quantum algorithm that finds a marked element with bounded error in complexity

$$\begin{aligned}&\mathcal{O}\Big(\sqrt{\log(C(t))}\mathsf{S}(\sigma) + \sqrt{C(t)\log(C(t))\log\log(C(t))}(\sqrt{t}\mathsf{U}\sqrt{\log(C(t))} + \mathsf{R}(\sigma) + \mathsf{C})\Big)\\ &= \widetilde{\mathcal{O}}\Big(\mathsf{S}(\sigma) + \sqrt{C(t)}(\sqrt{t}\mathsf{U}(\sigma) + \mathsf{C})\Big),\end{aligned}$$

where $C(t)$ is a known upper bound on $C_{\sigma,M}(P^t)$, $\mathsf{S}(\sigma)$ is the cost of $\texttt{Setup}(\sigma)$, U is the cost of the walk operator $W(P)$, $\mathsf{R}(\sigma)$ is as in Theorem 12, $\mathsf{U}(\sigma) = \mathsf{U} + \mathsf{R}(\sigma)$, and C is the cost of the $\texttt{Check}(M)$ operation.

Using standard techniques, we can also handle the case where $C_{\sigma,M}(P^t)$ is unknown, at the cost of an additional $\log(C_{\sigma,M}(P^t))$ factor on the first term, giving, for any t, an algorithm with complexity (neglecting log factors):

$$\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}(P^t)}(\sqrt{t}\mathsf{U}(\sigma) + \mathsf{C}).$$

Setting $t = 1$, we recover the electric network framework. In the special case where σ equals the stationary distribution π of P, and thus also of P^t, we have $C_{\pi,M}(P^t) = \mathrm{HT}(P^t, M)$ and $\mathsf{R}(\sigma) \in \mathcal{O}(1)$, and so the complexity of the algorithm is (neglecting log factors):

$$\mathsf{S} + \sqrt{\mathrm{HT}(P^t, M)}(\sqrt{t}\mathsf{U} + \mathsf{C}).$$

Setting $t = 1$ recovers the hitting time framework (Theorem 8), and setting $t = 1/\delta$ recovers the MNRS framework. To see this, note that since $\frac{1}{\delta}$ is at least the mixing time of P, a single step of $P^{1/\delta}$ approximately samples from π, which finds a marked vertex with probability ε, so $\mathrm{HT}(P^{1/\delta}, M) = \mathcal{O}(1/\varepsilon)$. If in addition there is a unique marked element $M = \{m\}$, we can choose $t \in \Omega(\varepsilon \mathrm{HT}(P, \{m\}))$ to retrieve the controlled quantum amplification framework. This immediately follows from Lemma 10 which shows that $\mathrm{HT}(P^t, \{m\}) \in \mathcal{O}(1/\varepsilon)$ if $t \in \Omega(\varepsilon \mathrm{HT}(P, \{m\}))$. If we could extend this bound to larger sets, then we find an immediate and strict extension of their framework.

Proof of Theorem 27. We will apply Theorem 13 to the reversible Markov chain P^t. This gives an algorithm for finding an element $x \in M$ with complexity:

$$\mathsf{S}(\sigma)\sqrt{\log(C(t))} + \sqrt{C(t)\log(C(t))\log\log(C(t))}(\mathsf{U}_t + \mathsf{R}_\sigma + \mathsf{C}), \tag{13}$$

where U_t is the complexity of implementing the walk operator $W(P^t)$, and R_σ is as described above Theorem 12. We need only describe how to implement a walk operator $W(P^t)$, and upper bound its complexity U_t.

By Theorem 7, since $D(P^t) = D(P)^t$, there is an ε-approximate walk operator $W(P^t)$ for P^t with complexity $\mathcal{O}\Big(\sqrt{t\log(1/\varepsilon)}\mathsf{U}\Big)$. We will call this operator a number of times

$$\tau = \sqrt{C(t)\log(C(t))\log\log(C(t))}.$$

Hence if we set $\varepsilon = \Theta(\frac{1}{\tau})$ then this ensures that the algorithm is correct with bounded error. This gives

$$\mathsf{U}_t = \mathcal{O}\Big(\sqrt{t}\mathsf{U}\sqrt{\log\tau}\Big) = \mathcal{O}\Big(\sqrt{t}\mathsf{U}\sqrt{\log(C(t))}\Big).$$

Plugging this into (13) completes the proof. □

6 Alternative algorithm for finding in the hitting time framework

Our quantum walk algorithm relies on the use of quantum fast-forwarding. This makes it more complicated than the original quantum walk algorithms in e.g. [Sze04, MNRS11, Bel13]. In this section we show that the correctness of our algorithm implies the correctness of a much simpler algorithm, at least in the regimes corresponding to the hitting time framework and the electric network framework. Namely, if we simply pick a random interpolation parameter, run the corresponding quantum walk for about $\sqrt{C}$ steps, and finally measure the walk register, then

we find a marked element with constant probability. This algorithm was proposed in [AGJK19, Section 4] for the hitting time framework, but was only conjectured to be correct.

To derive this result, we literally "dissect" the more complicated fast-forwarding algorithm (Algorithm 1) by considering an explicit construction of the quantum fast-forwarding routine. The structural properties of this quantum circuit then imply that the simpler routine should also succeed. To illustrate this, we give a proof of the correctness of the fast-forwarding technique, Theorem 7, as this is the main tool used in Algorithm 1.

Proof of fast-forwarding scheme, Theorem 7. In order to describe our construction we recall some well-known properties of quantum walks. One of the important basic observations is that for any unitary W for which $D := (\langle\bar{0}|\otimes I)W(|\bar{0}\rangle\otimes I)$ is a Hermitian matrix it holds [Chi10, GSLW19, AGJK19] that

$$(\langle\bar{0}|\otimes I)\Big(([I-2|\bar{0}\rangle\langle\bar{0}|]\otimes I)W^\dagger([I-2|\bar{0}\rangle\langle\bar{0}|]\otimes I)W\Big)^n(|\bar{0}\rangle\otimes I) = T_{2n}(D), \tag{14}$$

where $T_{2n}(x)$ is the $2n$-th Chebyshev polynomial of the first kind.

An intriguing property of Chebyshev polynomials is that [SV14]

$$x^t = \sum_{i=0}^{t} 2^{-t}\binom{t}{i}T_{2i-t}(x). \tag{15}$$

For $t, d \in \mathbb{N}$ even numbers, now define the polynomial

$$p_{t,d}(x) = \sum_{n=-\frac{d}{2}}^{\frac{d}{2}} 2^{-t}\binom{t}{\frac{t}{2}+n}T_{2n}(x),$$

which is simply the sum in (15) truncated. By Chernoff's bound and (15) it follows that for all $\varepsilon > 0$, $d \geq \left\lceil\sqrt{2t\ln(2/\varepsilon)}\right\rceil$, and $x \in [-1,1]$:

$$|x^t - p_{t,d}(x)| \leq \varepsilon.$$

Since $T_n(x) = T_{-n}(x)$, by (14), for all even t we get that

$$p_{t,d}(D) = \sum_{n=-\frac{d}{2}}^{\frac{d}{2}} 2^{-t}\binom{t}{\frac{t}{2}+n}(\langle\bar{0}|\otimes I)\Big(([I-2|\bar{0}\rangle\langle\bar{0}|]\otimes I)W^\dagger([I-2|\bar{0}\rangle\langle\bar{0}|]\otimes I)W\Big)^{|n|}(|\bar{0}\rangle\otimes I). \tag{16}$$

Let $\ell \in \mathbb{N}$, we define $C_k := ([I-2(I_{k-1}\otimes|1\rangle\langle 1|\otimes I_{\ell-k})\otimes|\bar{0}\rangle\langle\bar{0}|]\otimes I)$ as the controlled reflection operator controlled by the kth qubit, where I_m denotes the identity operator on m qubits. Let

$$\mathbb{U}^{(\ell)} := \prod_{k=0}^{\ell-1}\Big(C_kW^\dagger C_kW\Big)^{2^k} = \sum_{n=0}^{2^\ell-1}|n\rangle\langle n|\otimes\Big(([I-2|\bar{0}\rangle\langle\bar{0}|]\otimes I)W^\dagger([I-2|\bar{0}\rangle\langle\bar{0}|]\otimes I)W\Big)^n. \tag{17}$$

Now we use the linear combination of unitaries (LCU) [CW12, BCC+14] technique. Suppose that $d < 2^{\ell+1}$, and R is a unitary such that $R\colon \sqrt{\alpha}|0\rangle \mapsto \sqrt{2^{-t}\binom{t}{t/2}}|0\rangle + \sum_{n=1}^{\frac{d}{2}}\sqrt{2^{1-t}\binom{t}{\frac{t}{2}+n}}|n\rangle$, where $\alpha \in [1-\varepsilon, 1]$ is a normalizing factor. A simple LCU calculation shows, that we have

$$p_{t,d}(D) = \alpha(\langle 0|R^\dagger\otimes\langle\bar{0}|\otimes I)\mathbb{U}_\ell(R|0\rangle\otimes|\bar{0}\rangle\otimes I),$$

and therefore setting $|\tilde{0}\rangle := |0\rangle\otimes|\bar{0}\rangle$ and

$$U := (R^\dagger\otimes I)\mathbb{U}_\ell(R\otimes I) \tag{18}$$

we get

$$\left\|D^t - \alpha(\langle\tilde{0}| \otimes I)U(|\tilde{0}\rangle \otimes I)\right\| \leq \varepsilon.$$

We note that the case of odd t can be handled completely analogously using odd counterparts of (14)-(17). The α factor is also trivial to remove using simple techniques [GSLW19]; alternatively one can apply the triangle inequality and use the slightly weaker error-bound $\left\|D^t - (\langle\tilde{0}| \otimes I)U(|\tilde{0}\rangle \otimes I)\right\| \leq 2\varepsilon$. □

Now we are ready to prove the main statement of this section. We first recall the main technical corollary (Corollary 24) underlying Theorem 13: let $T \geq cC_{\sigma,M}$ for a sufficiently large constant c, set $r_S = (T/30)/2 = T/60$ and let the other interpolation parameter $r_M \in R = \{1, 2, 4, \ldots, 2^{\lceil \log(14T) \rceil}\}$ and time parameter $t \in [T]$ be chosen uniformly at random. Let U be a block-encoding of $D(q)^t = (\langle\bar{0}| \otimes I)U(|\bar{0}\rangle \otimes I)$, with $D(q)$ the discriminant matrix of $P(q)$ defined in (7). Then measuring the state $U(|\bar{0}\rangle \otimes |\sqrt{\sigma}\rangle)$ returns a marked element with probability at least $\Omega\Big(\frac{1}{\log(T)}\Big)$. This is precisely why Algorithm 1 is correct.

In particular we can use the unitary in (18) when $\varepsilon = \Theta\Big(\frac{1}{\log(T)}\Big)$ is small enough. Note that since we are only interested in the measurement statistics of the second register we can also use $\mathbb{U}_\ell(R \otimes I)$ instead of U. Then measuring the first part of the first register commutes with $\mathbb{U}_\ell$, so we can measure this register already before applying $\mathbb{U}_\ell$, without modifying the measurement statistics. Now we have the following algorithm: Apply R on the first half of the first register, then measure it. Finally apply $\mathbb{U}_\ell$ and measure the second register. But this is again equivalent to first (classically) sampling $n \in [-\frac{d}{2}, \frac{d}{2}]$ distributed $\propto 2^{-t}\binom{t}{\frac{t}{2}+n}$, and then applying $2n$ quantum walk steps to the initial state and measuring the second register. This works in the case when t is even; one can also handle odd t analogously by slightly tweaking the circuit U. In fact one can show that sampling an even $t \in [T]$ uniformly at random also works in the algorithm of [AGJK19], which is an alternative solution.

We summarize the resulting algorithm:

Algorithm 2 Simple quantum walk algorithm

1. pick $r_M \in R$ and $t \in [T]$ uniformly at random
2. sample n according to $2^{-t}\binom{t}{t/2+n}$, conditioned on $|n| \in \mathcal{O}\Big(\sqrt{T\log(T)}\Big)$ and having the same parity as t
3. apply $|n|$ steps of the interpolated quantum walk $W(P(q))$ with $q_M = 1 - \frac{1}{r_M}$ and $q_S = 1 - \frac{60}{T}$ to the state $|\sqrt{\sigma}\rangle$
4. measure the second register

Theorem 28. There exists a constant c such that if $T \geq cC_{\sigma,M}$ then Algorithm 2 returns a marked vertex with probability $\Omega\Big(\frac{1}{\log T}\Big)$.

Repeating this procedure $\Omega(\log T)$ times returns a marked vertex with constant probability. This yields an algorithm that only uses ordinary (interpolated) quantum walks and finds a marked element with constant probability. If $T \in \Theta(C_{\sigma,M})$, the algorithm has complexity $\mathcal{O}\big((\mathsf{S}(\sigma) + \sqrt{C_{\sigma,M}\log C_{\sigma,M}}(\mathsf{U}(\sigma) + \mathsf{C}))\log(C_{\sigma,M})\big)$. In the case of $\sigma = \pi$, we are in the hitting time framework, and this complexity becomes $\mathcal{O}\big((\mathsf{S} + \sqrt{\mathrm{HT}\log \mathrm{HT}}(\mathsf{U} + \mathsf{C}))\log(\mathrm{HT})\big)$.

Acknowledgments

We thank Frédéric Magniez, Stephen Piddock and Jérémie Roland for fruitful discussions and useful pointers.

References

[AF02] David Aldous and Jim Fill. Reversible Markov chains and random walks on graphs. Unfinished monograph, 2002. link.

[AGJK19] Andris Ambainis, András Gilyén, Stacey Jeffery, and Martins Kokainis. Quadratic speedup for finding marked vertices by quantum walks. arXiv: 1903.07493, 2019.

[AKR05] Andris Ambainis, Julia Kempe, and Alexander Rivosh. Coins make quantum walks faster. In *Proceedings of the 16th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1099–1108, 2005. arXiv: quant-ph/0402107

[Amb07] Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37:210–239, 2007.

[AS19] Simon Apers and Alain Sarlette. Quantum fast-forwarding: Markov chains and graph property testing. *Quantum Information and Computation*, 19(3–4):181–213, 2019. arXiv: 1804.02321

[BCC+14] Dominic W. Berry, Andrew M. Childs, Richard Cleve, Robin Kothari, and Rolando D. Somma. Exponential improvement in precision for simulating sparse Hamiltonians. In *Proceedings of the 46th ACM Symposium on Theory of Computing (STOC)*, pages 283–292, 2014. arXiv: 1312.1414

[BCJ+13] Aleksandrs Belovs, Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 105–122, 2013.

[Bel13] Aleksandrs Belovs. Quantum walks and electric networks. arXiv: 1302.3143, 2013.

[BHMT02] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *Contemporary Mathematics Series*, pages 53–74. AMS, 2002. arXiv: quant-ph/0005055

[BJLM13] Daniel J. Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer. Quantum algorithms for the subset-sum problem. In *Proceedings of the 5th International Conference on Post-Quantum Cryptography (PQCrypto)*, pages 16–33, 2013.

[Bol13] Béla Bollobás. *Modern graph theory*, volume 184. Springer Science & Business Media, 2013.

[BŠ06] Harry Buhrman and Robert Špalek. Quantum verification of matrix products. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 880–889, 2006.

[CGJ19] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. In *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 33:1–33:14, 2019. arXiv: 1804.01973

[Chi10] Andrew M. Childs. On the relationship between continuous- and discrete-time quantum walk. *Communications in Mathematical Physics*, 294(2):581–603, 2010. arXiv: 0810.0312

[CRR+96] Ashok K. Chandra, Prabhakar Raghavan, Walter L. Ruzzo, Roman Smolensky, and Prasoon Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996.

[CW12] Andrew M. Childs and Nathan Wiebe. Hamiltonian simulation using linear combinations of unitary operations. *Quantum Information and Computation*, 12(11&12):901–924, 2012. arXiv: 1202.5822

[DH17] Cătălin Dohotaru and Peter Høyer. Controlled quantum amplification. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 80, pages 18:1–18:13, 2017.

[DS84] Peter G. Doyle and J. Laurie Snell. *Random walks and electric networks*. Mathematical Association of America, 1984. arXiv: math/0001057

[GSLW19] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st ACM Symposium on Theory of Computing (STOC)*, pages 193–204, 2019. arXiv: 1806.01838

[HM18] Alexander Helm and Alexander May. Subset Sum Quantumly in 1.17^n. In *Proceedings of the 13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC)*, pages 5:1–5:15, 2018.

[JKM12] Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Nested quantum walks with quantum data structures. In *Proceedings of the 24th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1474–1485, 2012.

[Kir18] Elena Kirshanova. Improved quantum information set decoding. In *Proceedings of the 9th International Conference on Post-Quantum Cryptography (PQCrypto)*, pages 507–527, 2018.

[KMOR16] 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

[KT17] Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. In *Proceedings of the 8th International Conference on Post-Quantum Cryptography (PQCrypto)*, pages 69–89, 2017.

[Lov96] László Lovász. Random walks on graphs: A survey. In *Combinatorics, Paul Erdős is eighty*, volume 2, pages 353–398. János Bolyai Mathematical Society, 1996.

[LPW17] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. AMS, Providence, RI, USA, 2nd edition, 2017.

[MNRS11] 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

[Mon18] Ashley Montanaro. Quantum-walk speedup of backtracking algorithms. *Theory of Computing*, 14(15):1–24, 2018. arXiv: 1509.02374

[MSS07] Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–427, 2007.

[Pid19] Stephen Piddock. Quantum walk search algorithms and effective resistance, 2019. Personal Communication.

[SV14] Sushant Sachdeva and Nisheeth K. Vishnoi. Faster algorithms via approximation theory. *Foundations and Trends in Theoretical Computer Science*, 9(2):125–210, 2014.

[Sze04] Mário 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, 2004. arXiv: quant-ph/0401053

A Counterexample

It is a classic result that the combinatorial commute time $\mathbb{E}_s(\tau_s^t)$ equals the electric quantity $C_{s,t} = WR_{s,t}$. We show that this result can be extended to the case where t is a set M, rather than a singleton (see Appendix B). Similarly one could expect that something of the form $\mathbb{E}_{\pi|_S}(\tau_S^M) = WR_{\pi|_S,M}$ should hold. We show here that in fact this does not hold in general. Similarly we show that $\Pr_{\pi|_S}(\tau_M < \tau_S^+) \neq \frac{1}{C_{\pi|_S,M}\pi(S)}$, whereas this does hold when S is a singleton.

Let G be a path on three nodes $u - v - w$ with unit weights. Let $S = \{u, v\}$ and $M = \{w\}$, so that $\pi|_S = \frac{1}{3}e_u + \frac{2}{3}e_v$ and $\pi(S) = 3/4$. The optimal (and only) $\pi|_S$-M flow pushes value $1/3$ along the edge (u, v), and value 1 along the edge (v, w). The effective resistance thus equals $R_{\pi|_S,M} = \frac{1}{3^2} + 1 = \frac{10}{9}$. Since $W = 4$, this shows that $WR_{\pi|_S,M} = \frac{40}{9}$ and $\frac{1}{C_{\pi|_S,M}\pi(S)} = 3/10$.

On the other hand, we can easily calculate that

$$\Pr_{\pi|_S}(\tau_M < \tau_S^+) = \frac{1}{3} > \frac{1}{C_{\pi|_S,M}\pi(S)},$$

since the only possibility is to start from v and take the edge (v, w), which happens with probability $\frac{2}{3}\frac{1}{2} = \frac{1}{3}$. Similarly, we can show that the combinatorial commute time

$$\mathbb{E}_{\pi|_S}(\tau_S^M) = \frac{39}{9} < WR_{\pi|_S,M}.$$

To see this, note that $\mathbb{E}_{\pi|_S}(\tau_S^M) = \mathbb{E}_{\pi|_S}(\tau_M) + 1$, with $\mathbb{E}_{\pi|_S}(\tau_M)$ the expected hitting time of M (after the walk hits M, it necessarily jumps back to S). On its turn, $\mathbb{E}_{\pi|_S}(\tau_M) = \frac{1}{3}\mathbb{E}_u(\tau_M) + \frac{2}{3}\mathbb{E}_v(\tau_M)$ and $\mathbb{E}_u(\tau_M) = 1 + \mathbb{E}_v(\tau_M)$ (a walk from u necessarily jumps to v after one step). To calculate $\mathbb{E}_v(\tau_M)$, note that $\mathbb{E}_v(\tau_M) = \frac{1}{2} + \frac{1}{2}(\mathbb{E}_v(\tau_M) + 2)$, since with probability $1/2$ we jump to M in 1 step, and otherwise we go to u and then back to v, taking 2 steps. This implies that $\mathbb{E}_v(\tau_M) = 3$ and hence $\mathbb{E}_{\pi|_S}(\tau_S^M)$.

B Proof of s-M and S-M commute times

In this appendix we prove Claim 15. It follows by generalizing [LPW17, Proposition 9.5], where the theorem is proven for the special case of S and M being singletons. It builds on *voltages*, which are dual to electric flows. Any voltage is described by a function $h : X \to \mathbb{R}$ that is *harmonic* on all nodes that are not sources or sinks, i.e.,

$$h(u) = \sum_{v \in X} P_{u,v} h(v)$$

for every u which is neither a source (that is, $u \neq s$) nor a sink (that is $u \notin M$). The quantities $C_{\pi|_S,M}$ and $C_{S,M}$ are described in Definition 3.

Lemma 16. Let $S, M \subseteq X$ be disjoint sets, then

$$\Pr_{\pi|_S}(\tau_M < \tau_S^+) = \frac{1}{C_{S,M}\pi(S)} \left(\geq \frac{1}{C_{\pi|_S,M}\pi(S)} \right).$$

Proof. First we prove the claim for a singleton $S = \{s\}$, in which case the claim becomes

$$\Pr_s(\tau_M < \tau_s^+) = \frac{1}{C_{s,M}\pi_s}.$$

Define the *boundary voltages* $h_B(s) = 0$ and $h_B(u \in M) = 1$. By standard results [Bol13], this implies that a total current of magnitude $i = 1/R_{s,M}$ will flow from s to M, and the resulting voltage can be uniquely described as the *escape probability*

$$h(u) = \Pr_u(\tau_M < \tau_s),$$

as shown in [LPW17, Proposition 9.1] (see also [DS84]). Using that $h(s) = 0$ and $i_{u,v} = (h(v) - h(u))/w_{u,v}$ by Ohm's law, we can now rewrite

$$\begin{aligned}\Pr_s(\tau_M < \tau_s^+) &= \sum_{v \in X \setminus \{s\}} P(s,v) \Pr_v(\tau_M < \tau_s) \\ &= \sum_{v \in X \setminus \{s\}} \frac{w_{s,v}}{w_s}(h(v) - h(s)) = \sum_{v \in X \setminus \{s\}} \frac{i_{s,v}}{w_s} = \frac{i}{w_s},\end{aligned}$$

with i the total current. Since $i = 1/R_{s,M} = W/C_{s,M}$ and $\pi_s = w_s/W$, this implies that $\Pr_s(\tau_M < \tau_s^+) = 1/(C_{s,M}\pi_s)$.

Now we reduce the general case to the singleton case. For this we consider the graph G' where we replace S by a single vertex s', so that for $u, v \notin S$ we set $w'_{uv} := w_{uv}$, $w'_{s'v} := \sum_{s \in S} w_{sv}$, and $w'_{s's'} := \sum_{s,r \in S} w_{sr}$. Clearly then $W' = W$, $\pi'(s') = \pi(S)$ and $R'_{s',M} = R_{S,M}$. The latter deserves a little explanation. One can see that in the optimal $S \to M$ flow for any two vertices $s_1, s_2 \in S$ and $v \notin S$ we have $i_{s_1,v}/w_{s_1,v} = i_{s_2,v}/w_{s_2,v}$. Therefore, after merging the flows (currents) on the merged edges (s_1, v), (s_2, v) the dissipated power

$$\frac{(i_{s_1,v} + i_{s_2,v})^2}{w_{s_1,v} + w_{s_2,v}} = (i_{s_1,v} + i_{s_2,v}) \frac{i_{s_1,v} + i_{s_2,v}}{w_{s_1,v} + w_{s_2,v}} = (i_{s_1,v} + i_{s_2,v}) \left(\frac{i_{s_1,v}}{w_{s_1,v}} = \frac{i_{s_2,v}}{w_{s_2,v}} \right) = \frac{i_{s_1,v}^2}{w_{s_1,v}} + \frac{i_{s_2,v}^2}{w_{s_2,v}}$$

remains unchanged. So merging the flows / distributing flows proportionally to the edge weights gives a mapping between the optimal flows ($S \to M$ and $s' \to M$) without changing the objective.

Finally, observe that

$$\begin{aligned}\Pr_{\pi|_S}(\tau_M < \tau_S^+) &= \sum_{s \in S} \frac{\pi(s)}{\pi(S)} \sum_{v \in X \setminus S} P(s,v) \Pr_v(\tau_M < \tau_S) \\ &= \sum_{s \in S} \frac{w_s}{w(S)} \sum_{v \in X \setminus S} \frac{w_{s,v}}{w_s} \Pr_v(\tau_M < \tau_S) \\ &= \sum_{v \in X \setminus S} \sum_{s \in S} \frac{w_{s,v}}{w(S)} \Pr_v(\tau_M < \tau_S) \\ &= \sum_{v \in X \setminus S} P'(s',v) \Pr'_v(\tau_M < \tau_{s'}) \\ &= \Pr_{s'}(\tau_M < \tau_{s'}^+),\end{aligned}$$

and so

$$\Pr_{\pi|_S}(\tau_M < \tau_S^+) = \Pr_{s'}(\tau_M < \tau_{s'}^+) = \frac{1}{C'_{s',M}\pi(s')} = \frac{1}{C_{S,M}\pi(S)}.$$

□

B.1 Special case where $S = \{s\}$

For the special case where S is a singleton, this gives a tight characterization of the commute time. This easily follows from combining Lemma 16 with the expression below. This expression is proven in [Lov96, Proposition 2.3] or [AF02, Corollary 2.8] for the case where M is a singleton, but it is easily extended to the more general case.

Lemma 29. Let s be disjoint from M. Then

$$\Pr_s(\tau_M < \tau_s^+) = \frac{1}{\mathbb{E}_s(\tau_M^s)\pi_s}.$$

Proof. Let $q = \Pr_s(\tau_M < \tau_s^+)$. Then by Kac's Lemma (Lemma 17) we know that $\mathbb{E}_s(\tau_s^+) = 1/\pi_s$. Necessarily, when starting from s, $\tau_s^+ \leq \tau_M^s$, and furthermore $\Pr_s(\tau_s^+ = \tau_M^s) = q$. Now if $\tau_s^+ < \tau_M^s$, we know that the Markov chain is "restarted" at timestep τ_s^+ (that is, it is distributed the same as when it started, namely, it is at s), and hence

$$\begin{aligned}\mathbb{E}_s(\tau_M^s - \tau_s^+) &= q\mathbb{E}_s(\tau_M^s - \tau_s^+ | \tau_M^s = \tau_s^+) + (1-q)\mathbb{E}_s(\tau_M^s - \tau_s^+ | \tau_M^s > \tau_s^+) \\ \mathbb{E}_s(\tau_M^s) - \mathbb{E}_s(\tau_s^+) &= (1-q)\mathbb{E}_s(\tau_M^s).\end{aligned}$$

We can therefore rewrite $q = \mathbb{E}_s(\tau_s^+)/\mathbb{E}_s(\tau_M^s) = 1/(\mathbb{E}(\tau_M^s)\pi_s)$, proving the claim. □

Combining this lemma with our Lemma 16 shows that

$$\Pr_s(\tau_M < \tau_s^+) = \frac{1}{C_{s,M}\pi_s} = \frac{1}{\mathbb{E}_s(\tau_M^s)\pi_s},$$

and therefore $\mathbb{E}_s(\tau_M^s) = C_{s,M}$. This generalizes the classic fact that $C_{s,t} = \mathbb{E}_s(\tau_t^s)$, as we mentioned in Theorem 4.