Finding Is as Easy as Detecting for Quantum Walks

Hari Krovi[1], Frédéric Magniez[2], Maris Ozols[3,4], and Jérémie Roland[3]

[1] University of Connecticut
[2] LRI, Univ Paris-Sud, CNRS
[3] NEC Laboratories America, Inc.
[4] University of Waterloo and Institute for Quantum Computing

Abstract. We solve an open problem by constructing quantum walks that not only detect but also find marked vertices in a graph. The number of steps of the quantum walk is quadratically smaller than the classical hitting time of any reversible random walk P on the graph.

Our approach is new, simpler and more general than previous ones. We introduce a notion of interpolation between the walk P and the absorbing walk P', whose marked states are absorbing. Then our quantum walk is simply the quantum analogue of the interpolation. Contrary to previous approaches, our results remain valid when the random walk P is not state-transitive, and in the presence of multiple marked vertices.

As a consequence we make a progress on an open problem related to the spatial search on the 2D-grid.

1 Introduction

Many classical randomized algorithms rely heavily on random walks or Markov chains. The notion of hitting time is intimately related to the problem of spatial search, where displacement constraints are modeled by an undirected graph G. The set of desired vertices, or marked vertices, is denoted by M. Classically, a simple algorithm to find a marked vertex is to repeatedly apply some random walk P on G until one of the marked vertices is reached. The hitting time of P, $\mathrm{HT}(P, M)$, is precisely the expected number of repetitions, or steps, necessary to reach a marked vertex, starting from the stationary distribution of P.

Quantum walks are natural generalizations of classical random walks. Ambainis [1] was the first to solve a natural problem – the "element distinctness problem" – using a quantum walk. Following this, many quantum walk algorithms were discovered [2,3,4]. Quantum walk algorithms for the spatial search problem [5] were studied for the hypercube [6] and the grid [7,8].

The notion of hitting time has been carried over to the quantum case in [8,9,10,11,12,13]. Usually, the quantum hitting time has a quadratic improvement over the classical one. However, until the present paper, several serious restrictions were imposed: a quantum algorithm could only solve the detection problem of deciding whether there are marked vertices or not [10], but for being able to find them the Markov chain had to be reversible, state-transitive, and with

S. Abramsky et al. (Eds.): ICALP 2010, Part I, LNCS 6198, pp. 540–551, 2010.

a unique marked vertex [14,13]. The detection algorithm is quite intuitive and well understood, whereas the finding algorithm requires an elaborate proof whose intuition is not clear. This is due in part to a modification of the quantum walk, so that the resulting walk is not a quantum analogue of a Markov chain anymore.

Whether this quadratic speed-up for finding a marked vertex also holds for any reversible Markov chain and for multiple marked vertices was an open question. In this paper, we answer this question in the positive. Here we choose another approach by modifying directly P, and by considering the quantum analogue of the modified random walk. Doing that we keep some intuition and get simpler proofs while obtaining more general results. The new walk is simply an interpolation between the walk P and the absorbing walk P', where all outgoing transitions from marked vertices are replaced by self-loops. The interpolation coefficient can be used to tune the overlap between the stationary superposition of the quantum walk and its projection onto marked vertices. For a suitable value, depending on the relative weight of marked vertices in the stationary distribution of P, the overlap with marked and unmarked vertices is balanced, leading to a quantum walk algorithm that finds a marked vertex within $\sqrt{\mathrm{HT}(P, M)}$ steps (**Theorem 5**). The balancing can also be achieved when limited or even no information is available on the weight of marked vertices (**Theorems 6, 7 and 8**). As a consequence, we make a progress on an open problem from [5,14] related to the spatial search on the 2D-grid (**Corollary 2**).

2 Preliminaries

2.1 Spatial Search on Graphs

We fix throughout the paper an undirected graph $G = (X, E)$ with $|X| = n$. Let $M \subseteq X$ be a set of marked vertices and let $m = |M|$. Vertices are encoded in a distinguished *vertex register*. Our goal is to find any of the marked vertices in M using only evolutions that preserve the locality of G on the vertex register, i.e., to perform a *spatial search* on G [5]. Here we define an even more restricted notion of locality than the ones in [5], but it is more intuitive and sufficiently powerful for our purpose. We allow two types of operations on the vertex register: *static* transformations (that can be conditioned on the state of the vertex register but do not modify it) and shift (that exchanges the value of the vertex register and another register). Nonetheless, we want to restrict the executions of shift to $(x, y) \in E$.

Definition 1. *Let*

$$\textsc{Shift}\ (x, y) = \begin{cases} (y, x), & \text{if } (x, y) \in E, \\ (x, y), & \text{otherwise.} \end{cases}$$

In the first case we say that Shift succeeds*, but in the second case it* fails*.*

Definition 2 (Problems). *Under the restrictions that only static transformations and* Shift *are allowed, consider the following problems:*

- Detect(G)*: Detect if there is a marked vertex in* G*;*
- Find(G)*: Find any marked vertex in* G*, with the promise that* $M \neq \emptyset$*.*

$\textsc{Detect}^{(k)}(G)$ (resp. $\textsc{Detect}^{(\geq k)}(G)$) will denote the problem $\textsc{Detect}(G)$ with the promise that $m = 0$ or $m = k$ (resp. $m \geq k$). Similarly, define $\textsc{Find}^{(k)}(G)$ (resp. $\textsc{Find}^{(\geq k)}(G)$) as $\textsc{Find}(G)$ with the promise that $m = k$ (resp. $m \geq k$).

A natural approach to searching on a graph consists in using a random walk. Intuitively, a random walk is an alternation of coin flips and shifts. Namely, a coin is flipped according to the current state x of the vertex register, its value describes a target vertex y, and $\textsc{Shift}$ is a move from x to y. Let p_{xy} be the probability that x is shifted to y. Then $\textsc{Shift}$ always succeeds if $p_{xy} = 0$ whenever $(x, y) \notin E$. In that case, we say that $P = (p_{xy})_{x,y \in X}$ is a Markov chain on G.

We assume from now on that P is an ergodic Markov chain. Therefore P has a unique stationary distribution π. We also assume that P is reversible: $\pi_x p_{xy} = \pi_y p_{yx}$, for all $x, y \in X$. To measure the complexity of implementing a random walk corresponding to P, we introduce the following black-box operations:

- $\texttt{Check}(M)$: Check if a given vertex is marked;
- $\texttt{Setup}(P)$: Draw a sample from the stationary distribution π of P;
- $\texttt{Update}(P)$: Perform one step of P.

Each of these black-box operations have the corresponding associated implementation cost. We denote by C, S and U the respective complexities of the transformations $\texttt{Check}(M)$, $\texttt{Setup}(P)$ and $\texttt{Update}(P)$.

2.2 Quantum Version

In the quantum case, the problem extends as follows. Let $\mathcal{H} = \mathbb{C}^X$ be a fixed Hilbert space with basis $(|x\rangle)_{x \in X}$. Again, a transformation is *static* if it is controlled by the vertex register, that is of type $\sum_{x \in X} |x\rangle\langle x| \otimes V_x$, and Definition 1 of $\textsc{Shift}$ is simply extended by linearity. Then the generalization of random walks to quantum walks is as follows.

Definition 3. *A* quantum walk W *on G is a composition of static unitary transformations and* $\textsc{Shift}$ *on an invariant subspace of $\mathcal{H} \otimes \mathcal{H}$, the* walk space*, such that* $\textsc{Shift}$ *always succeeds when W is restricted to its walk space.*

Implicitly we always restrict a quantum walk to its walk space.

We will only consider quantum walks built from quantum analogues of reversible Markov chains. Thus we extend the operations `Check`, `Setup` and `Update` to the quantum setting as follows. Let $|\overline{0}\rangle \in \mathcal{H}$ be a fixed reference state. In the following, the first register is the vertex register.

- $\texttt{Check}(M)$: Map $|x\rangle|b\rangle$ to $|x\rangle|b\rangle$ if $x \notin M$ and $|x\rangle|b \oplus 1\rangle$ if $x \in M$, for $b = 0, 1$.
- $\texttt{Setup}(P)$: Construct the superposition: $|\pi\rangle = \sum_{x \in X} \sqrt{\pi_x}|x\rangle$;
- $\texttt{Update}(P)$: Apply any of $V(P)$, $V(P)^\dagger$ or $\textsc{Shift}$, where $V(P)$ satisfies $V(P)|x\rangle|\overline{0}\rangle = |x\rangle|p_x\rangle = |x\rangle \sum_{y \in X} \sqrt{p_{xy}}|y\rangle$, for all $x \in X$.

Implicitly, we also allow any controlled version of $\texttt{Check}(M)$, $\texttt{Setup}(P)$ and $\texttt{Update}(P)$, on which we access via oracle.

In terms of applications of $\textsc{Shift}$, `Update` has complexity 1, and, `Setup` has complexity $\mathrm{O}(\delta_G)$ (the diameter of G). Nonetheless, in many algorithmic applications, the situation is more complex and the number of applications of $\textsc{Shift}$ is not the only relevant cost, see for instance [1,2].

2.3 Classical Hitting Time

From now on we will assume that all the eigenvalues of P are within $[0, 1]$. This is without loss of generality by replacing P by $(\mathrm{Id} + P)/2$ if necessary. From the ergodicity of P, the eigenvalue 1 has multiplicity 1. The classical hitting time, $\mathrm{HT}(P, M)$, is defined as the expected number of applications of the Markov chain P required to hit a marked vertex when starting from π. This can be used to design a randomized algorithm for DETECT and FIND based on the corresponding random walk.

Proposition 1. *Let $k \geq 1$. $\textsc{Detect}^{(\geq k)}(G)$ can be solved with high probability and randomized complexity of order*

$$\mathsf{S} + T \times (\mathsf{U} + \mathsf{C}), \quad \textit{where} \quad T = \max_{|M'|=k} \mathrm{HT}(P, M').$$

$\textsc{Find}(G)$ can be solved with high probability and expected randomized complexity of order

$$\mathsf{S} + T \times (\mathsf{U} + \mathsf{C}), \quad \textit{where} \quad T = \mathrm{HT}(P, M).$$

Let P' be the Markov chain obtained from P by turning all outgoing transitions from marked vertices into self-loops. We call P' the *absorbing* version of P. If we arrange the elements of X so that the marked vertices are the last ones, matrices P and P' have the following block structure:

$$P := \begin{pmatrix} P_{UU} & P_{UM} \\ P_{MU} & P_{MM} \end{pmatrix}, \qquad P' := \begin{pmatrix} P_{UU} & P_{UM} \\ 0 & I \end{pmatrix}.$$

where P_{UU} and P_{MM} are matrices of size $(n-m) \times (n-m)$ and $m \times m$, while P_{UM} and P_{MU} are matrices of size $(n-m) \times m$ and $m \times (n-m)$.

We first present the matrix characterization of $\mathrm{HT}(P, M)$. From the reversibility of P, we get that $D(P) = \mathrm{diag}(\sqrt{\pi}) P \,\mathrm{diag}(\sqrt{\pi})^{-1}$ is a symmetric matrix, which is also known as the discriminant of P. The latter was introduced in [10] for symmetric P, and generalized to reversible P in [12]. Set $|\pi\rangle = \sum_{x \in X} \sqrt{\pi_x} |x\rangle$, $p_M = \sum_{x \in M} \pi_x$ and $p_U = \sum_{x \in X \setminus M} \pi_x$. The respective normalized projections of $|\pi\rangle$ on marked and unmarked vertices are $|M\rangle = \sum_{x \in M} \sqrt{\pi_x / p_M} |x\rangle$ and $|U\rangle = \sum_{x \in X \setminus M} \sqrt{\pi_x / p_U} |x\rangle$. Then

$$\mathrm{HT}(P, M) = p_U \times \langle U | (\mathrm{Id}_U - D(P)_{UU})^{-1} | U \rangle.$$

For simplicity, we will from now on omit the quantity p_U in the above definition. Indeed, we assume that $p_M \leq 1/2$, so that the the difference between the two expressions is at most a factor of 2. Note that if $p_M > 1/2$, then there is no need for any (classical or quantum) walk to find a marked vertex.

We now introduce the spectral characterization of $\mathrm{HT}(P, M)$. Note that the reversibility of P also implies the alternative definition for the discriminant

$$(D(P))_{xy} = \sqrt{p_{xy} p_{yx}}.$$

Extending the latter definition of the discriminant to P', we get that $D(P') = D(P)_{UU} \oplus \mathrm{Id}_M$. Let $|v_1\rangle, |v_2\rangle, \ldots, |v_n\rangle$ be a system of orthonormal eigenvectors

of $D(P')$ with respective eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{n-m} < \lambda_{n-m+1} = \ldots = \lambda_n = 1$, so that $D(P') = \sum_k \lambda_k |v_k\rangle\langle v_k|$. Then one can rewrite $\mathrm{HT}(P, M)$ as:

$$\mathrm{HT}(P, M) = \sum_{k \leq n-m} \frac{|\langle v_k|U\rangle|^2}{1 - \lambda_k}.$$

2.4 Quantum Hitting Time

Quantum walks were successfully used for detecting the presence of marked vertices quadratically faster than P. Nonetheless, only little is known on the problem of finding a marked vertex. Below we illustrate the state of the art.

Theorem 1 ([10]). *Let $k \geq 1$. $\textsc{Detect}^{(\geq k)}(G)$ can be solved with high probability and quantum complexity of order*

$$\mathsf{S} + T \times (\mathsf{U} + \mathsf{C}), \quad \textit{where} \quad T = \max_{|M'|=k} \sqrt{\mathrm{HT}(P, M')}.$$

When P is state-transitive and there is a unique marked vertex z (i.e., $m = 1$), $\mathrm{HT}(P, \{z\})$ is independent of z and one can also find z:

Theorem 2 ([14,13]). *Assume that P is state-transitive. $\textsc{Find}^{(1)}(G)$ can be solved with high probability and quantum complexity of order*

$$\mathsf{S} + T \times (\mathsf{U} + \mathsf{C}), \quad \textit{where} \quad T = \sqrt{\mathrm{HT}(P, \{z\})}.$$

Using standard techniques, such as in [5], Theorem 2 can be generalized to any number of marked vertices, with an extra logarithmic multiplication factor. Nonetheless, the complexity of the corresponding algorithms does not decrease when the size of M increases, contrary to the random walk search algorithm (Proposition 1) and the quantum walk detecting algorithm (Theorem 1).

Corollary 1. *Assume that P is state-transitive. $\textsc{Find}(G)$ can be solved with high probability and quantum complexity of order*

$$\log(n) \times \Big(\mathsf{S} + T \times (\mathsf{U} + \mathsf{C})\Big), \quad \textit{where} \quad T = \sqrt{\mathrm{HT}(P, \{z\})}, \textit{ for any } z.$$

2.5 Szegedy's Quantum Analogue

Following the main lines of Szegedy [10], we define a quantum analogue of a reversible Markov chain P. Recall that $|\bar{0}\rangle$ is an arbitrary reference state in $\mathcal{H}$, and $V(P)|x\rangle|\bar{0}\rangle = |x\rangle|p_x\rangle$. Set $\mathcal{X} = \mathcal{H} \otimes \mathrm{span}\,\{|\bar{0}\rangle\} = \mathrm{span}\,\{|x\rangle|\bar{0}\rangle : x \in X\}$, and $\mathrm{ref}_{\mathcal{X}} = 2\sum_{x \in X} |x\rangle\langle x| \otimes |\bar{0}\rangle\langle\bar{0}| - \mathrm{Id}$, the reflection with respect to $\mathcal{X}$.

Definition 4. *The* quantum analogue *of P is*

$$W(P) = V(P)^\dagger \cdot \textsc{Shift} \cdot V(P) \cdot \mathrm{ref}_{\mathcal{X}},$$

and its walk space is the subspace spanned by $\mathcal{X}$ and $W(P)\mathcal{X}$.

Observe that in [10], the quantum walk is actually defined as $(V(P) \cdot W(P) \cdot V(P)^\dagger)^2$. Moreover, $W(P)$ requires 3 calls to $\texttt{Update}(P)$, and $\textsc{Shift}$ always succeeds when $W(P)$ is restricted to its walk space.

Szegedy proved the following useful lemma which relates the spectral decomposition of $W(P)$ in its walk space to the one of P. Let $|v_1\rangle, |v_2\rangle, \ldots, |v_n\rangle$ be the normalized eigenvectors of $D(P)$ with respective eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_{n-1} < \lambda_n = 1$. From the definition of $D(P)$, observe that $|\pi\rangle$ is a 1-eigenvector of $D(P)$, and therefore we set $|v_n\rangle = |\pi\rangle$.

Lemma 1 ([10]). *Define* $\mathcal{B}_k = \text{span}\{|v_k\rangle|\bar{0}\rangle, W(P)|v_k\rangle|\bar{0}\rangle\}$, *for* $k \neq n$, *and* $\mathcal{B}_n = \text{span}\{|v_n\rangle|\bar{0}\rangle\}$. *Then the walk space of* $W(P)$ *is* $\bigoplus_k \mathcal{B}_k$, *where* $W(P)$ *admits the following spectral decomposition:*

- *On* $\mathcal{B}_k$, $k \neq n$: $\mu_k^{\pm} = e^{\pm i\varphi_k}, |\Psi_k^{\pm}\rangle$, *where* $\cos\varphi_k = \lambda_k$ *and* $\frac{|\Psi_k^+\rangle + |\Psi_k^-\rangle}{\sqrt{2}} = |v_k\rangle|\bar{0}\rangle$
- *On* $\mathcal{B}_n$: $\mu_n = 1, |\Psi_n\rangle = |v_n\rangle|\bar{0}\rangle$. $\mathcal{B}^{\perp}$).

Therefore, we will also call $|\Psi_n\rangle = |v_n\rangle|\bar{0}\rangle$ the *stationary distribution* of $W(P)$.

3 Finding via Quantum Walk

3.1 Classical Interpolation

Our starting state is the stationary superposition $|v_n\rangle|\bar{0}\rangle = |\pi\rangle|\bar{0}\rangle$ of $W(P)$. We would like to end in its projection onto marked vertices, namely $|M\rangle|\bar{0}\rangle$, which is also the stationary superposition of $W(P')$. However, in many cases, including the 2D-grid, every iteration of $W(P')$ on $|\pi\rangle|\bar{0}\rangle$ may remain far from $|M\rangle|\bar{0}\rangle$.

Our approach consists in taking a new random walk, namely an interpolation between P and P', This technique is drastically different from the approach of [14,13], and up to our knowledge new.

Definition 5. *The* classical interpolation *of* P *and* P' *is*

$$P(s) = (1-s)P + sP', \quad 0 \leq s \leq 1.$$

This interpolation has some similarities with adiabatic evolutions, where similar interpolations are usually defined for Hamiltonians. Here the interpolation is of different nature after we take the quantum analogue of $P(s)$. Nonetheless, this interpolation still makes sense for adiabatic evolutions, leading to an interesting connection between the hitting time and the adiabatic condition [15].

Note that $P(0) = P$, $P(1) = P'$, and $P(s)$ has block structure

$$P(s) = \begin{pmatrix} P_{UU} & P_{UM} \\ (1-s)P_{MU} & (1-s)P_{MM} + sI \end{pmatrix}.$$

Decompose the stationary distribution of P as $\pi = \begin{pmatrix} \pi_U & \pi_M \end{pmatrix}$. Remember that $p_M = \sum_{x \in M} \pi_x$ is the probability to pick a marked vertex from the stationary distribution. Then

$$\pi(s) = \frac{1}{1-(1-s)p_M} \begin{pmatrix} (1-s)\pi_U & \pi_M \end{pmatrix}$$

is a stationary distribution of $P(s)$, for $s \in [0,1]$. Moreover one can show that:

Fact 1. *For* $s \in [0,1)$, *the Markov chain* $P(s)$ *is ergodic and reversible.*

For $s \in [0,1)$, let $D(s)$ be the discriminant of $P(s)$. Then $D(s)$ and $P(s)$ are similar and therefore have the same eigenvalues. Let $|v_1(s)\rangle, |v_2(s)\rangle, \ldots, |v_n(s)\rangle$

be a system of orthonormal eigenvectors of $D(s)$ with respective eigenvalues $0 \leq \lambda_1(s) \leq \lambda_2(s) \leq \ldots \leq \lambda_{n-1}(s) < \lambda_n(s) = 1$: $D(s) = \sum_k \lambda_k(s)|v_k(s)\rangle\langle v_k(s)|$. We also define $W(s) = W(P(s))$.

For $s = 1$, we extend the above for $P(1) = P'$. Recall that $|v_n(s)\rangle = |\pi(s)\rangle$ is an eigenvector of $D(s)$ with eigenvalue $\lambda_n(s) = 1$. Observe that $|v_n(s)\rangle$ is in the two-dimensional subspace spanned by $|M\rangle$ and $|U\rangle$.

Fact 2. $|v_n(s)\rangle = \cos\theta(s)|U\rangle + \sin\theta(s)|M\rangle$, *where* $\theta(s) = \arcsin\sqrt{\frac{p_M}{1-s(1-p_M)}}$.

Intuitively we want $|v_n(s)\rangle$ to have a large overlap on both $|U\rangle$ and $|M\rangle$, so that the algorithm will proceed in two steps: first, map $|U\rangle$ to $|v_n(s)\rangle$, using the quantum walk (using `Update`); second, map $|v_n(s)\rangle$ to $|M\rangle$ by projecting onto marked vertices (using `Check`). Therefore, ideally we want s to satisfy $\sin\theta(s) = \cos\theta(s) = 1/\sqrt{2}$, namely $s = s(p_M)$, where $s(p_M) = 1 - \frac{p_M}{1-p_M}$.

3.2 Quantum Circuit for $W(s)$

In the following lemma, we assume to know p_{xx} for every x. This is reasonable since in practice the probability of self-loops is known. In many cases, it is even independent of x. For the rest of the paper, we assume that this is not an obstacle (we can assume that one call to $\texttt{Update}(P)$ allows to learn p_{xx} for any x).

Lemma 2. *Assuming that p_{xx} is known for every x,* $\texttt{Update}(P(s))$ *can be implemented with complexity* $\mathsf{C} + \mathsf{U}$.

Proof. From Definition 4, the quantum analogue of $P(s)$ is $W(s) = V(P(s))^\dagger \cdot \textsc{Shift}\cdot V(P(s))\cdot \mathrm{ref}_{\mathcal{X}}$, where $V(P(s))$ is a unitary that maps $|x\rangle|\bar{0}\rangle$ to $|x\rangle|p_x(s)\rangle = |x\rangle\sum_{y\in X}\sqrt{p_{xy}(s)}|y\rangle$. Since Shift only depends on G, we just need to explain how to implement $V(P(s))$ and its inverse. We now explain how to implement $V(P(s))$ using one call to $V(P)$ and 2 calls to $\texttt{Check}(M)$. The algorithm for its inverse is obtained from the reverse algorithm.

Simulation(P, M, s)

1. Let $|x\rangle|\bar{0}\rangle$ be the current state
2. Use a fresh qubit (*marking* register) in state $|0\rangle$ and call $\texttt{Check}(M)$.
3. If the marking register is in state $|0\rangle$, call $V(P)$: $|x\rangle|p_{xy}\rangle|0\rangle$
4. Otherwise
   (a) Use another fresh qubit in state $|0\rangle$: $|x\rangle|\bar{0}\rangle|1\rangle|0\rangle$
   (b) Apply a rotation of angle $\arcsin\sqrt{s}$ on the fourth register: $|x\rangle|\bar{0}\rangle|1\rangle(\sqrt{1-s}|0\rangle + \sqrt{s}|1\rangle)$
   (c) If the fourth register is $|0\rangle$, apply $V(P)$ on the first two registers,
   Otherwise XOR the first two registers:
   $|x\rangle(\sqrt{1-s}|p_x\rangle|1\rangle|0\rangle + \sqrt{s}|x\rangle|1\rangle|1\rangle)$
   (d) If the second register is $|x\rangle$, apply a rotation of angle $-\arcsin\sqrt{s/((1-s)p_{xx}+s)}$ on the fourth register,
   Otherwise do nothing: $|x\rangle|p_x(s)\rangle|1\rangle|0\rangle$
5. Call $\texttt{Check}(M)$ to uncompute the marking register.

□

3.3 Hitting Time in s

Following [15], we define the hitting time in s, which intuitively corresponds to the expected time for $P(s)$ to converge to $\pi(s)$ from $\pi(0)$.

Definition 6. $\mathrm{HT}(s) = \sum_{k \neq n} \frac{|\langle v_k(s)|U\rangle|^2}{1-\lambda_k(s)}$.

In particular, note that $\mathrm{HT}(1) = \mathrm{HT}(P, M)$ is the usual hitting time of P with respect to the set of marked vertices M.

The running time of our quantum search algorithms will depend on $\mathrm{HT}(s)$ for some particular value $s \in [0, 1]$. This can be related to the usual hitting time $\mathrm{HT}(P, M)$ thanks to the following explicit expression for $\mathrm{HT}(s)$ (see [15]).

Theorem 3. $\mathrm{HT}(s) = \sin^4 \theta(s) \cdot \mathrm{HT}(P, M)$.

4 Quantum Search

4.1 Algorithm with Known Parameters

Theorem 4 (Phase estimation [16,17]). *Let W be a unitary operator on $\mathcal{H}$ and $t \in \mathbb{N}$. There exists a quantum circuit* **PhaseEstimation**(W, t) *using 2^t calls to the controlled operator $\mathrm{c}{-}W$ and $\mathrm{O}(t^2)$ additional gates, and acting on eigenstates $|\Psi_k\rangle$ of W as*

$$|\Psi_k\rangle \to |\Psi_k\rangle \frac{1}{2^t} \sum_{l,m=0}^{2^t-1} e^{-\frac{2\pi i l m}{2^t}} e^{i\varphi_k l} |m\rangle,$$

where $e^{i\varphi_k}$ is the eigenvalue of W corresponding to $|\Psi_k\rangle$.

By linearity, Theorem 4 implies that **PhaseEstimation**(W, t) resolves any state along the eigenstates of W, labelling those states with a second register whose measurement yields an approximation of the first t-bit of the binary decomposition of $\varphi_k/(2\pi)$. Here, we will be mostly interested in the $|\Psi_n\rangle$-component, with corresponding phase $\varphi_n = 0$. In that case, the second register is in the state $|0^t\rangle$ and the estimation is exact.

We define our main search algorithm with parameters $0 \leq p^* \leq 1$ (an approximation of p_M) and $t \in \mathbb{N}$. Recall that $s(p^*) = 1 - \frac{p^*}{1-p^*}$. For suitable parameters, the algorithm outputs a marked vertex with high probability, if there is any.

QuantumWalkSearch(P, M, p^*, t)
1. Prepare the state $|\pi\rangle|\bar{0}\rangle$
2. Use a fresh qubit in state $|0\rangle$, apply `Check`(M) and measure the qubit.
3. If the outcome is $|1\rangle$, measure the first register (in the vertex basis) and output the outcome.
4. Otherwise, apply **PhaseEstimation**$(W(s(p^*)), t)$ on the registers.
5. Use a fresh qubit in state $|0\rangle$, apply `Check`(M) and measure the qubit.
6. If the outcome is $|1\rangle$, measure the first register (in the vertex basis) and output the outcome.
7. Otherwise, output "`No marked vertex`"

Theorem 5. *Let* $p^*, \epsilon_1 \in [0,1]$ *be such that* $\cos^2\theta(s)\sin^2\theta(s) \geq \epsilon_1$, *where* $s = s(p^*)$. *Let* $T \geq 1$ *and* $\epsilon_2 \in [0,1]$ *be such that* $T \geq \frac{\pi}{\sqrt{2\epsilon_2}} \times \sqrt{\mathrm{HT}(s)}$. *Then,* $\mathbf{QuantumWalkSearch}(P, M, p^*, \lceil \log T \rceil)$ *outputs a marked vertex with probability at least* $\epsilon_1 - \epsilon_2$ *and complexity of order* $\mathsf{S} + T \times (\mathsf{U} + \mathsf{C})$.
In particular, if $|p^* - p_M| \leq p_M/3$ *and* $T \geq 10\sqrt{\mathrm{HT}(P,M)}$, *then the success probability is at least* $1/20$.

Proof. Let $t = \lceil \log T \rceil$. First observe that the complexity analysis is direct since $\mathbf{QuantumWalkSearch}(P, M, p^*, t)$ has complexity of order $\mathsf{S} + 2^t \times (\mathsf{U} + \mathsf{C})$. We now assume that we reach Step 4. Otherwise a marked vertex is already found. Then the current state before Step 4 is $|U\rangle|\bar{0}\rangle$. Let $\alpha_k(s) = \langle U|v_k(s)\rangle$. From now on, we omit to write the dependence on s explicitly, when there is no ambiguity.

In Step 4, $\mathbf{PhaseEstimation}(W(s), t)$ is applied on the state

$$|U\rangle|\bar{0}\rangle = \alpha_n|v_n\rangle|\bar{0}\rangle + \sum_{k\neq n}\alpha_k|v_k\rangle|\bar{0}\rangle = \alpha_n|\Psi_n\rangle + \frac{1}{\sqrt{2}}\sum_{k\neq n}\alpha_k\left(|\Psi_k^+\rangle + |\Psi_k^-\rangle\right).$$

Theorem 4 shows that $\mathbf{PhaseEstimation}(W(s), t)$ maps $|\Psi_n\rangle$ to $|\Psi_n\rangle|0^t\rangle$ and maps $|\Psi_k^\pm\rangle$ to $|\Psi_k^\pm\rangle\left(\delta_k^\pm|0^t\rangle + |\eta_k^\pm\rangle\right)$, where

$$\delta_k^\pm = \frac{1}{2^t}\sum_{l=0}^{2^t-1} e^{\pm i\varphi_k l} \quad \text{and} \quad |\eta_k^\pm\rangle = \frac{1}{2^t}\sum_{m=1}^{2^t-1}\sum_{l=0}^{2^t-1} e^{-\frac{2\pi i l m}{2^t}} e^{\pm i\varphi_k l}|m\rangle.$$

By definition, $\langle 0^t|\eta_k^\pm\rangle = 0$. Then, the probability p to obtain a marked vertex by measuring the first register is at least the probability to obtain both a marked vertex in the first register and the state $|0^t\rangle$ in the last register (i.e., the phase is estimated to be 0). Since $|\Psi_n\rangle = |v_n\rangle|\bar{0}\rangle$ and using Fact 2, we see that the probability p is lower bounded as

$$p \geq |\alpha_n|^2\|\Pi_M|v_n\rangle\|^2 - \frac{1}{2}\sum_{k\neq n}|\alpha_k|^2\left(|\delta_k^+|^2 + |\delta_k^-|^2\right) = \cos^2\theta\sin^2\theta - \sum_{k\neq n}|\alpha_k|^2\delta_k^2,$$

where $\Pi_M = \sum_{x\in M}|x\rangle\langle x|$ is the projection onto marked vertices and $\delta_k = |\delta_k^+| = |\delta_k^-|$. By hypothesis, we already have $\cos^2\theta\sin^2\theta \geq \epsilon_1$. Therefore, it remains to prove that the second term in the RHS is at least $-\epsilon_2$.

First, using the definition of δ_k, we get: $\delta_k^2 = \frac{\sin^2(2^{t-1}\varphi_k)}{2^{2t}\sin^2(\varphi_k/2)} \leq \frac{\pi^2}{2^{2t}\varphi_k^2}$. We also have by definition of $\mathrm{HT}(s)$:

$$\mathrm{HT}(s) = \sum_{k\neq n}\frac{|\alpha_k|^2}{1-\cos\varphi_k} = \sum_{k\neq n}\frac{|\alpha_k|^2}{2\sin^2(\varphi_k/2)} \geq 2\sum_{k\neq n}\frac{|\alpha_k|^2}{\varphi_k^2},$$

which together with the above implies that $\sum_{k\neq n}|\alpha_k|^2\delta_k^2 \leq \frac{\pi^2}{2}\frac{\mathrm{HT}(s)}{2^{2t}} \leq \epsilon_2$.

We now prove the last part of the theorem. The following fact is easy to prove:

Fact 3. *Let* $\epsilon_1 \leq 1/4$ *be such that* $2\sqrt{\epsilon_1}p_M \leq p^* \leq 2(1-\sqrt{\epsilon_1})p_M$. *Then,* $\cos^2\theta(s)\sin^2\theta(s) \geq \epsilon_1$.

The conditions of Fact 3 are satisfied with $\epsilon_1 = 1/10$. Set $\epsilon_2 = 1/20$. Using $\mathrm{HT}(s) \leq \mathrm{HT}(P,M)$, one can check that the conditions of the theorem are satisfied, and therefore the success probability is at least $\epsilon_1 - \epsilon_2 = 1/20$. □

4.2 General Case

At first, assume we have a correct approximation p^* of p_M. In that case, even if we do not know $\mathrm{HT}(P, M)$, we can use the following algorithm, and still find a marked vertex with an expected cost $\mathrm{O}(\sqrt{\mathrm{HT}(P, M)})$.

QuantumWalkSearch$'(P, M, p^*, k)$
1. Let $t = 1$.
2. Call k times **QuantumWalkSearch**(P, M, p^*, t).
3. If no marked vertex is found, set $t \leftarrow t + 1$ and go back to step 2.

Theorem 6. *Given* p^* *such that* $|p^* - p_M| \leq p_M/3$, **QuantumWalkSearch**$'(P, M, p^*, 28)$ *solves* $\textsc{Find}(G)$ *with expected quantum complexity of order*

$$\log(T) \times \mathsf{S} + T \times (\mathsf{U} + \mathsf{C}), \quad \textit{where} \quad T = \sqrt{\mathrm{HT}(P, M)}.$$

Proof. The general idea is to use **QuantumWalkSearch**(P, M, p^*, t) with increasing accuracy of the phase estimation (parameter t), until it is high enough so that the algorithm outputs a marked element with high probability.

Set $s = s(p^*)$. Let t_0 to be the integer such that $\pi\sqrt{\frac{\mathrm{HT}(s)}{2\epsilon_2}} \leq 2^{t_0} \leq \pi\sqrt{\frac{2\,\mathrm{HT}(s)}{\epsilon_2}}$, and let $T = 2^{t_0} = \mathrm{O}(\sqrt{\mathrm{HT}(s)})$. By Theorem 5, for any $t \geq t_0$, **QuantumWalkSearch**(P, M, p^*, t) outputs a marked vertex with probability at least $1/20$. Then, step 3 is reached without finding any marked vertex with probability at most $p \leq (1 - 1/20)^{28} \leq 1/4$. Moreover, **QuantumWalkSearch**(P, M, p^*, t) has complexity of order $\mathsf{S} + 2^t \times (\mathsf{U} + \mathsf{C})$.

Let t_f be the value of t when **QuantumWalkSearch**$'(P, M, p^*, 28)$ stops, that is, the number of iterations of step 2. Then, the expected complexity of **QuantumWalkSearch**$'(P, M, p^*, 28)$ is of order $N_1 \times \mathsf{S} + N_2 \times (\mathsf{U} + \mathsf{C})$, where N_1 is the expectation of t_f, and N_2 is the expectation of $2 + 4 + \ldots + 2^{t_f}$.

First observe that $N_1 \leq t_0 + \sum_{t=t_0+1}^{\infty} p^{t-t_0} = \mathrm{O}(t_0)$. For N_2 we get

$$N_2 \leq \sum_{t=1}^{t_0} 2^t + \sum_{t=t_0+1}^{\infty} p^{t-t_0} \cdot 2^t = (2 \cdot 2^{t_0} - 2) + 2^{t_0} \sum_{t=1}^{\infty} p^t \cdot 2^t.$$

Then using the fact that $p \leq 1/4$ we finally obtain

$$N_2 \leq 2 \cdot 2^{t_0} + 2^{t_0} \sum_{t=1}^{\infty} 2^{-t} \leq 3 \cdot 2^{t_0}.$$

This concludes the proof since $2^{t_0} = \mathrm{O}(\sqrt{\mathrm{HT}(s)})$ and $\mathrm{HT}(s) \leq \mathrm{HT}(P, M)$. □

For the general case we get two possible situations, depending on whether a lower bound $p_{\min}$ on p_M and/or an upper bound $\mathrm{HT}_{\max}$ on $\mathrm{HT}(P, M)$ is given. In particular, for $\textsc{Find}(G)^{(\geq k)}$, we can set $p_{\min} = \min_{M':|M'|=k} p_{M'}$ and $\mathrm{HT}_{\max} = \max_{M':|M'|=k} \mathrm{HT}(P, M')$.

Theorem 7. *Given* $p_{\min} \leq p_M$, $\textsc{Find}(G)$ *can be solved with expected quantum complexity of order*

$$\sqrt{\log(1/p_{\min})} \times \big[\log(T) \times \mathsf{S} + T \times (\mathsf{U} + \mathsf{C})\big], \quad \textit{where} \quad T = \sqrt{\mathrm{HT}(P, M)}.$$

Moreover, if $\mathrm{HT}_{\max} \geq \mathrm{HT}(P, M)$ *is also given, then* $\textsc{Find}(G)$ *can be solved with expected quantum complexity of order*

$$\sqrt{\log(1/p_{\min})} \times \big[\mathsf{S} + T \times (\mathsf{U} + \mathsf{C})\big], \quad \textit{where} \quad T = \sqrt{\mathrm{HT}_{\max}}.$$

Proof. We simply prove the first part of the theorem. The second one is similar using $\mathbf{QuantumWalkSearch}(P, M, p^*, T)$ instead of $\mathbf{QuantumWalkSearch}'(P, M, p^*, 28)$.

From Theorem 6, it is enough to have a good approximation p^* of p_M, such that $3p^*/4 \leq p_M \leq 3p^*/2$. Moreover, since $p_{\min} \leq p_M \leq 1/2$, this condition will be satisfied for some $p^* \in \{(2/3) \times 2^{-l} : l = 1, \ldots, \lfloor \log(1/p_{\min}) \rfloor\}$.

Let us incorporate step 2 of $\mathbf{QuantumWalkSearch}'(P, M, p^*, 28)$ into a loop on the $\lfloor \log(1/p_{\min}) \rfloor$ possible values of p^*. Then the analysis is basically the same, except that now the complexity of step 2 is multiplied by a factor of order $\log(1/p_{\min})$. Instead of looping on all possible values of p^*, we can search for the right value using Grover's algorithm, following the approach of [18], therefore reducing the multiplication factor to $\sqrt{\log(1/p_{\min})}$. □

Theorem 8. *Given* $\mathrm{HT}_{\max} \geq \mathrm{HT}(P, M)$, $\textsc{Find}(G)$ *can be solved with expected quantum complexity of order*

$$\log(1/p_M) \times \big[\mathsf{S} + T \times (\mathsf{U} + \mathsf{C})\big], \quad \textit{where} \quad T = \sqrt{\mathrm{HT}_{\max}}.$$

Proof. We now use $\mathbf{QuantumWalkSearch}(P, M, p^*, t)$ with $t = \lceil \log \sqrt{\mathrm{HT}_{\max}} \rceil$, and perform a dichotomic search for an appropriate value of p^*. This dichotomic search uses backtracking since the branching in the dichotomy is with bounded error, similarly to the situation in [19].

Initially we set $a = 0$ and $b = 1$. Then for testing the current value of $p^* = (a + b)/2$, we run a constant number of times $\mathbf{QuantumWalkSearch}(P, M, p^*, t)$. If a marked vertex is found we stop. Otherwise, if $\mathbf{PhaseEstimation}(W(s(p^*)), t)$ outputs a minority of 0s, we set $a = p^*$, otherwise we set $b = p^*$. The details of the analysis are given in [19]. □

4.3 Application to the 2D-Grid

Consider a random walk on the 2D-grid of size $\sqrt{n} \times \sqrt{n}$, with self-loops. In this section we consider only the complexity in terms of the number of uses of `Check` and $\textsc{Shift}$. The previous best known quantum complexity of $\textsc{Find}(G)^{(k)}$ and $\textsc{Find}(G)^{(\geq k)}$ was $\mathrm{O}(\sqrt{n}(\log n)^{3/2})$, from Corollary 1. Since the grid is a 5-regular graphs (4 directions and 1 self-loop), P is symmetric, and therefore the stationary distribution of P is uniform, and we simply have $p_M = m/n$. Then `Setup` is realized with $\sqrt{n}$ uses of $\textsc{Shift}$, and $\mathrm{HT}(P, \{z\}) = \Theta(n \log n)$, for any z. Therefore we get the following corollary of Theorem 5 and Theorem 7, by upper bounding $\mathrm{HT}(P, M) = \mathrm{O}(n \log n)$.

Corollary 2. *Let* G *be the* 2D*-grid of size* $\sqrt{n} \times \sqrt{n}$*, and let* $k \geq 1$*. Then* $\textsc{Find}(G)^{(k)}$ *can be solved with expected quantum complexity* $\mathrm{O}(\sqrt{n \log n})$*, and* $\textsc{Find}(G)^{(\geq k)}$ *with expected quantum complexity* $\mathrm{O}(\sqrt{n \times \log n \times \log(n/k)})$.

Acknowledgments

This research has been supported in part by ARO/NSA under grant W911NF-09-1-0569. M. Ozols acknowledges support from QuantumWorks. F. Magniez acknowledges support from French ANR grants CRYQ (ANR-09-JCJC-0067) and QRAC (ANR-08-EMER-012), as well as the European Commission IST Integrated Project Qubit Applications (QAP) 015848.

References

1. Ambainis, A.: Quantum walk algorithm for Element Distinctness. In: Proc. 45th FOCS, pp. 22–31. IEEE Computer Society Press, New York (2004)
2. Magniez, F., Santha, M., Szegedy, M.: Quantum Algorithms for the Triangle Problem. In: Proc. 16th SODA (2005)
3. Buhrman, H., Špalek, R.: Quantum verification of matrix products. In: Proc. 17th ACM-SIAM Symposium on Discrete Algorithms, pp. 880–889 (2006)
4. Magniez, F., Nayak, A.: Quantum complexity of testing group commutativity. In: Caires, L., Italiano, G.F., Monteiro, L., Palamidessi, C., Yung, M. (eds.) ICALP 2005. LNCS, vol. 3580, pp. 1312–1324. Springer, Heidelberg (2005)
5. Aaronson, S., Ambainis, A.: Quantum search of spatial regions. Theory of Computing 1, 47–79 (2005)
6. Shenvi, N., Kempe, J., Whaley, K.: Quantum random-walk search algorithm. Phys. Rev. A 67, Article no. 052307 (2003)
7. Childs, A.M., Goldstone, J.: Spatial search and the Dirac equation. Phys. Rev. A 70(4), 042312 (2004)
8. Ambainis, A., Kempe, J., Rivosh, A.: Coins make quantum walks faster. In: Proc. 16th SODA, pp. 1099–1108 (2005)
9. Kempe, J.: Discrete quantum walks hit exponentially faster. Prob. Th. Rel. Fields 133(2), 215–235 (2005)
10. Szegedy, M.: Quantum speed-up of Markov chain based algorithms. In: Proc. 45th FOCS, pp. 32–41 (2004)
11. Krovi, H., Brun, T.A.: Hitting time for quantum walks on the hypercube. Phys. Rev. A 73(3), 032341 (2006)
12. Magniez, F., Nayak, A., Roland, J., Santha, M.: Search via quantum walk. In: Proc. 39th STOC, pp. 575–584. ACM Press, New York (2007)
13. Magniez, F., Nayak, A., Richter, P.C., Santha, M.: On the hitting times of quantum versus random walks. In: Proc. 19th SODA, SIAM, pp. 86–95. SIAM, Philadelphia (2009)
14. Tulsi, A.: Faster quantum-walk algorithm for the two-dimensional spatial search. Phys. Rev. A 78(1), 012310 (2008)
15. Krovi, H., Ozols, M., Roland, J.: On the adiabatic condition and the quantum hitting time of Markov chains. Technical Report arXiv:1004.2721, arXiv.org (2010)
16. Kitaev, A.: Quantum measurements and the Abelian stabilizer problem. Technical Report quant-ph/9511026, arXiv.org (1995)
17. Cleve, R., Ekert, A., Macchiavello, C., Mosca, M.: Quantum algorithms revisited. Proc. Royal Society A 454(1969), 339–354 (1998)
18. Høyer, P., Mosca, M., de Wolf, R.: Quantum search on bounded-error inputs. In: Baeten, J.C.M., Lenstra, J.K., Parrow, J., Woeginger, G.J. (eds.) ICALP 2003. LNCS, vol. 2719, pp. 291–299. Springer, Heidelberg (2003)
19. Feige, U., Raghavan, P., Feleg, D., Upfal, E.: Computing with noisy information. SIAM Journal on Computing 23(5), 1001–1018 (1994)