Quantum walk based search algorithms *

Miklos Santha[1,2]

[1] CNRS–LRI, Université Paris–Sud, 91405 Orsay, France
[2] Centre for Quantum Technologies, Nat. Univ. of Singapore, Singapore 117543

Abstract. In this survey paper we give an intuitive treatment of the discrete time quantization of classical Markov chains. Grover search and the quantum walk based search algorithms of Ambainis, Szegedy and Magniez et al. will be stated as quantum analogues of classical search procedures. We present a rather detailed description of a somewhat simplified version of the MNRS algorithm. Finally, in the query complexity model, we show how quantum walks can be applied to the following search problems: Element Distinctness, Matrix Product Verification, Restricted Range Associativity, Triangle, and Group Commutativity.

1 Introduction

Searching is without any doubt one of the major problems in computer science. The corresponding literature is tremendous, most manuals on algorithms include several chapters that deal with searching procedures [22, 14]. The relevance of finite Markov chains (random walks in graphs) to searching was recognized from early on, and it is still a flourishing field. The algorithm of Aleliunas et al. [4] that solves s–t connectivity in undirected graphs in time $O(n^3)$ and in space $O(\log n)$, and Schöning's algorithm [33] that provides the basis of the currently fastest solutions for 3-SAT are among the most prominent examples for that.

Searching is also a central piece in the emerging field of quantum algorithms. Grover search [16], and in general amplitude amplification [11] are well known quantum procedures which are provably faster than their classical counterpart. Grover's algorithm was used recursively by Aaronson and Ambainis [2] for searching in grids.

Discrete time quantum walks were introduced gradually by Meyer[27, 28] in connection with cellular automata, and by Watrous in his works related to space bounded computations [37]. Different parameters related to quantum walks and possible speedups of classical algorithms were investigated by several researchers [30, 8, 3, 29, 19, 32].

The potential of discrete time quantum walks with respect to searching problems was first pointed out by Shenvi, Kempe, and Whaley [34] who designed a quantum walk based simulation of Grover search. Ambainis, in his seminal

* Research supported by the European Commission IST Integrated Project Qubit Applications (QAP) 015848, and by the ANR Blanc AlgoQP grant of the French Research Ministry.

paper [7], used quantum walks on the Johnson graphs to settle the query complexity of the Element Distinctness problems. Inspired by the work of Ambainis, Szegedy [35] designed a general method to quantize classical Markov chains, and developed a theory of quantum walk based search algorithms. A similar approach for the specific case of searching in grids was taken by Ambainis, Kempe and Rivosh [9]. The frameworks of Ambainis and Szegedy were used in various contexts to find algorithms with substantial complexity gains over simple Grover search [26, 12, 23, 15]. In a recent work, Magniez, Nayak, Roland and Santha [25] proposed a new quantum walk based search method that expanded the scope of the previous approaches. The MNRS search algorithm is also conceptually simple, and improves various aspects of many walk based algorithms.

In this survey paper we give an intuitive (though formal) treatment of the quantization of classical Markov chains. We will be concerned with discrete time quantum walks, the continuous case will not be covered here. Grover search and the quantum walk based search algorithms of Ambainis, Szegedy and Magniez et al. will be stated as quantum analogues of classical search procedures. We present a rather detailed description of a somewhat simplified version of the MNRS algorithm. Finally, in the query complexity model, we show how quantum walks can be applied to the following search problems: Element Distinctness, Matrix Product Verification, Restricted Range Associativity, Triangle, and Group Commutativity. For a detailed introduction to quantum walks the reader is referred to the excellent surveys of Kempe [20] and Ambainis [5]. Another survey on quantum search algorithms is also due to Ambainis [6].

2 Classical search algorithms

At an abstract level, any search problem may be cast as the problem of finding a marked element from a set X. Let $M \subseteq X$ be the set of marked elements, and let ε be a known lower bound on $|M|/|X|$, the fraction of marked elements, whenever M is non-empty. If no further information is available on M, we can choose ε as $1/|X|$. The simplest approach, stated in **Search Algorithm 1**, to solve this problem is to repeatedly sample from X uniformly until a marked element is picked, if there is any.

Search Algorithm 1
Repeat for $t = O(1/\varepsilon)$ steps
1. Sample $x \in X$ according to the uniform distribution.
2. If x is in M then output it and stop.

More sophisticated approaches might use a Markov chain on the state space X for generating the samples. In that case, to generate the next sample, the resources expended for previous generations are often reused.

Markov chains can be viewed as random walks on directed graphs with weighted edges. We will identify a Markov chain with its transition matrix

$P = (p_{xy})$. A chain is *irreducible* if every state is accessible from every other state. An irreducible chain is *ergodic* if it is also aperiodic. The eigenvalues of a Markov chain are at most 1 in magnitude. By the Perron-Frobenius theorem, an irreducible chain has a unique stationary distribution $\pi = (\pi_x)$, that is a unique left eigenvector π with eigenvalue 1 and positive coordinates summing up to 1. If the chain is ergodic, the eigenvalue 1 is the only eigenvalue of P with magnitude 1. We will denote by $\delta = \delta(P)$ the *eigenvalue gap* of P, that is $1 - |\lambda|$, where λ is an eigenvalue with the second largest magnitude. It follows that when P is ergodic then $\delta > 0$.

The *time-reversed Markov chain* P^* of P is defined by the equations $\pi_x p_{xy} = \pi_y p^*_{yx}$. The Markov chain P is said to be *reversible* if $P^* = P$. Reversible chains can be viewed as random walks on undirected graphs with weighted edges, and in these chains the probability of a transition from a state x to another state y in the stationary distribution is the same as the probability of the transition in the reverse direction. The Markov chain P is *symmetric* if $P = P^t$ where P^t denotes the transposed matrix of P. The stationary distribution of symmetric chains is the uniform distribution. They can be viewed as random walks on regular graphs, and they are time-reversible.

We consider two search algorithms based on some ergodic and symmetric chain P. **Search Algorithm 2** repeatedly samples from approximately stationary distributions, and checks if the element is marked. To get a sample the Markov chain is simulated long enough to mix well. **Search Algorithm 3** is a greedy variant: a check is performed after every step of the chain.

Search Algorithm 2
1. Initialize x to a state sampled from the uniform distribution over X.
2. Repeat for $t_2 = O(1/\varepsilon)$ steps
   (a) If the element reached in the previous step is marked then output it and stop.
   (b) Simulate $t_1 = O(1/\delta)$ steps of P starting with x.

Search Algorithm 3
1. Initialize x to a state sampled from the uniform distribution over X.
2. Repeat for $t = O(1/\varepsilon\delta)$ steps
   (a) If the element reached in the previous step is marked then output it and stop.
   (b) Simulate one step of P starting with x.

We state formally the complexity of the three algorithms to clarify their differences. They will maintain a data structure d that associates some data $d(x)$ with every state $x \in X$. Creating and maintaining the data structure incurs a certain cost, but the data $d(x)$ can be helpful to determine if $x \in M$. We distinguish three types of cost.

Setup cost S**:** The cost to sample $x \in X$ according to the uniform distribution, and to construct $d(x)$.
Update cost U**:** The cost to simulate a transition from x to y according to P, and to update $d(x)$ to $d(y)$.
Checking cost C**:** The cost of checking if $x \in M$ using $d(x)$.

The cost may be thought of as a vector listing all the measures of complexity of interest, such as query and time complexity. The generic bounds on the efficiency of the three search algorithms can be stated in terms of the cost parameters.

Proposition 1. *Let P be an ergodic and symmetric Markov chain on X. Then all three algorithms find a marked element with high probability if there is any. The respective costs incurred by the algorithms are of the following order:*

1. **Search Algorithm 1***:* $(\mathsf{S} + \mathsf{C})/\varepsilon$,
2. **Search Algorithm 2***:* $\mathsf{S} + (\mathsf{U}/\delta + \mathsf{C})/\varepsilon$,
3. **Search Algorithm 3***:* $\mathsf{S} + (\mathsf{U} + \mathsf{C})/\delta\varepsilon$.

The generic bound of $O(1/\delta\varepsilon)$ in **Search Algorithm 3** on the hitting time is not always optimal, which in some cases, for example in the 2-dimensional grid, can be significantly smaller.

3 Quantum analogue of a classical Markov chain

We define a quantum analogue of an arbitrary irreducible Markov chain P as it is given by Magniez et al. [25]. This definition is based on and slightly extends the concept of quantum Markov chain due to Szegedy [35]. The latter was inspired by an earlier notion of quantum walk due to Ambainis [7]. We also point out that a similar process on regular graphs was studied by Watrous [37].

The quantum walk may be thought of as a walk on the *edges* of the original Markov chain, rather than on its vertices. Thus, its state space is a vector subspace of $\mathcal{H} = \mathbb{C}^{X \times X} \cong \mathbb{C}^X \otimes \mathbb{C}^X$. For a state $|\psi\rangle \in \mathcal{H}$, let $\Pi_\psi = |\psi\rangle\langle\psi|$ denote the orthogonal projector onto $\mathsf{Span}(|\psi\rangle)$, and let $\mathrm{ref}(\psi) = 2\Pi_\psi - \mathrm{Id}$ denote the reflection through the line generated by $|\psi\rangle$, where Id is the identity operator on $\mathcal{H}$. If $\mathcal{K}$ is a subspace of $\mathcal{H}$ spanned by a set of mutually orthogonal states $\{|\psi_i\rangle : i \in I\}$, then let $\Pi_\mathcal{K} = \sum_{i \in I} \Pi_{\psi_i}$ be the orthogonal projector onto $\mathcal{K}$, and let $\mathrm{ref}(\mathcal{K}) = 2\Pi_\mathcal{K} - \mathrm{Id}$ be the reflection through $\mathcal{K}$. Let $\mathcal{A} = \mathsf{Span}(|x\rangle|p_x\rangle : x \in X)$ and $\mathcal{B} = \mathsf{Span}(|p_y^*\rangle|y\rangle : y \in X)$ be vector subspaces of $\mathcal{H}$, where

$$|p_x\rangle = \textstyle\sum_{y \in X} \sqrt{p_{xy}}\,|y\rangle \quad \text{and} \quad |p_y^*\rangle = \textstyle\sum_{x \in X} \sqrt{p_{yx}^*}\,|x\rangle.$$

Definition 1 (Quantum walk). *The unitary operation $W(P) = \mathrm{ref}(\mathcal{B}) \cdot \mathrm{ref}(\mathcal{A})$ defined on $\mathcal{H}$ by is called the* quantum walk *based on the classical chain P.*

Let us give some motivations for this definition. Classical random walks do not quantize in the space of the vertices. The standard way advocated by several papers (see the survey [20]) is to extend the vertex space X by a coin space C, and define the state space of the walk as $X \times C$. Then a step of the walk is defined

as the product of two unitary operations. The first one is the *flip* operation F controlled by the vertex state, which means that for every $x \in X$, it performs a unitary coin flip F^x on the states $\{|x, c\rangle : c \in C\}$. For d-regular undirected graphs, C can be taken as the set $\{1, \ldots, d\}$, and in that case the coin flip F^x is independent from x. The second one is the *shift* operation S which is controlled by the coin state, and takes a vertex to one if its neighboring vertices. For d-regular graphs the simplest way to define it is via a labeling of the directed edges by the numbers between 1 and d such that for every $1 \leq i \leq d$, the directed edges labeled by i form a permutation. Then, if the coin state is i, the new vertex is the i^{th} neighbor according to the labeling. For general walks it is practical to take the coin space also to be X, then the state space of the walk corresponds naturally to the directed edges of the graph. In this case there is a symmetry between the two spaces, and the shift operation simply exchanges the vertices, that is $S|x, y\rangle = |y, x\rangle$, for every $x, y \in X$.

Let us pause here for a second and consider how a classical walk defined by some Markov chain P can be thought of as a walk on the directed edges of the graph (instead of the vertices). Let's think about an edge (x, u) as the state of the walk P being at x, where the previous state was u. According to this interpretation, in one step the walk on edges should move from state (x, u) to state (y, x) with probability p_{xy}. This move can be accomplished by the stochastic flip operation F controlled by the left end-point of the edge, where $F^x_{uy} = p_{xy}$ for all $x, u, y \in X$, followed by the shift S defined previously. If we define the flip F' as F but controlled by the right end-point of the edge, then it is not hard to see that $SFSF = F'F$. Therefore one can get rid of the shift operations, and two steps of the walk can be accomplished by two successive flips where the control and the target registers alternate.

Coming back to the quantization of classical walks, we thus want to find unitary coin flips which mirror the walk P, and which alternately mix the right end-point of the edges over the neighbors of the left end-point, and then the left end-point of the edges over the neighbors of the new right end-point. The reflections $\mathrm{ref}(\mathcal{A})$ and $\mathrm{ref}(\mathcal{B})$ are natural choices for that. They are also generalizations of the Grover diffusion operator [16]. Indeed, when the transition to each neighbor is equally likely, they correspond exactly to Grover diffusion. In Szegedy's original definition the alternating reflections were $\mathrm{ref}(\mathcal{A})$ and $\mathrm{ref}(\mathcal{B}')$ with $\mathcal{B}' = \mathsf{Span}(|p_y\rangle|y\rangle : y \in X)$, mirroring faithfully the classical edge based walk. The reason why the MNRS quantization chooses every second step a reflection based on the reversed walk P^* is explained now.

The eigen-spectrum of the transition matrix P plays an important role in the analysis of a classical Markov chain. Similarly, the behaviour of the quantum process $W(P)$ may be inferred from its spectral decomposition. The reflections through subspaces $\mathcal{A}$ and $\mathcal{B}$ are (real) orthogonal transformations, and so is their product $W(P)$. An orthogonal matrix may be decomposed into a direct sum of the identity, reflection through the origin, and two-dimensional rotations over orthogonal vector subspaces [17, Section 81]. These subspaces and the corresponding eigenvalues are revealed by the singular value decomposition of the

product $\Pi_{\mathcal{A}}\Pi_{\mathcal{B}}$ of the orthogonal projection operators onto the subspaces $\mathcal{A}$ and $\mathcal{B}$. Equivalently, as done by Szegedy, one can consider the singular values of the *discriminant* matrix $D(P) = (\sqrt{p_{xy}p^*_{yx}})$. Since $\sqrt{p_{xy}p^*_{yx}} = \sqrt{\pi_x}p_{xy}/\sqrt{\pi_y}$, we have

$$D(P) \quad = \quad \mathrm{diag}(\pi)^{1/2} \cdot P \cdot \mathrm{diag}(\pi)^{-1/2},$$

where $\mathrm{diag}(\pi)$ is the invertible diagonal matrix with the coordinates of the distribution π in its diagonal. Therefore $D(P)$ and P are similar, and their spectra are the same. When P is reversible then $D(P)$ is symmetric, and its singular values are equal to the absolute values of its eigenvalues. Thus, in that case we only have to study the spectrum of P.

Since the singular values of $D(P)$ all lie in the range $[0, 1]$, they can be expressed as $\cos\theta$, for some angles $\theta \in [0, \frac{\pi}{2}]$. The following theorem of Szegedy relates the singular value decomposition of $D(P)$ to the spectral decomposition of $W(P)$.

Theorem 1 (Szegedy [35]). *Let P be an irreducible Markov chain, and let $\cos\theta_1, \ldots, \cos\theta_l$ be an enumeration of those singular values (possibly repeated) of $D(P)$ that lie in the open interval $(0, 1)$. Then the exact description of the spectrum of $W(P)$ on $\mathcal{A} + \mathcal{B}$ is:*

1. *On $\mathcal{A} + \mathcal{B}$ those eigenvalues of $W(P)$ that have non-zero imaginary part are exactly $e^{\pm 2i\theta_1}, \ldots, e^{\pm 2i\theta_l}$, with the same multiplicity.*
2. *On $\mathcal{A} \cap \mathcal{B}$ the operator $W(P)$ acts as the identity* Id. *$\mathcal{A} \cap \mathcal{B}$ is spanned by the left (and right) singular vectors of $D(P)$ with singular value* 1.
3. *On $\mathcal{A} \cap \mathcal{B}^\perp$ and $\mathcal{A}^\perp \cap \mathcal{B}$ the operator $W(P)$ acts as* $-$Id. *$\mathcal{A} \cap \mathcal{B}^\perp$ (respectively, $\mathcal{A}^\perp \cap \mathcal{B}$) is spanned by the left (respectively, right) singular vectors of $D(P)$ with singular value* 0.

Let us now suppose in addition that P is ergodic and reversible. As we just said, reversibility implies that the singular values of $D(P)$ are equal to the absolute values of the eigenvalues of P. From the ergodicity it also follows that $D(P)$ has a unique singular vector with singular value 1. We have therefore the following corollary.

Corollary 1. *Let P be an ergodic and reversible Markov chain. Then, on $\mathcal{A} + \mathcal{B}$ the spectrum of $W(P)$ can be characterized as:*

$$|\pi\rangle = \sum_{x \in X} \sqrt{\pi_x}\, |x\rangle |p_x\rangle = \sum_{y \in X} \sqrt{\pi_y}\, |p^*_y\rangle |y\rangle$$

is the unique 1-eigenvector, $e^{\pm 2i\theta}$ are eigenvalues for every singular value $\cos\theta \in (0, 1)$ of $D(P)$, and all the remaining eigenvalues are -1.

The *phase gap* $\Delta(P) = \Delta$ of $W(P)$ is defined as 2θ, where θ is the smallest angle in $(0, \frac{\pi}{2}]$ such that $\cos\theta$ is a singular value of $D(P)$. This definition is motivated by the previous theorem and corollary: the angular distance of 1 from any other eigenvalue of $W(P)$ on $\mathcal{A} + \mathcal{B}$ is at least Δ. When P is ergodic and

reversible, there is a quadratic relationship between the phase gap Δ of the quantum walk $W(P)$ and the eigenvalue gap δ of the classical Markov chain P, more precisely $\Delta \geq 2\sqrt{\delta}$. Indeed, let $\delta \in (0, \frac{\pi}{2}]$ such that $\delta = 1 - \cos\theta$ and $\Delta = 2\theta$. The following (in)equalities can easily be checked: $\Delta \geq \left|1 - \mathrm{e}^{2\mathrm{i}\theta}\right| = 2\sqrt{1 - \cos^2\theta} \geq 2\sqrt{\delta}$. The origin of the quadratic speed-up due to quantum walks may be traced to this phenomenon.

4 Quantum search algorithms

As in the classical case, the quantum search algorithms look for a marked element in a finite set X. We suppose that the elements of X are coded by binary strings and that $\bar{0}$, the everywhere 0 string is in X. A data structure attached to both vertex registers is maintained during the algorithm. Again, three types of cost will be distinguished, generalizing those of the classical search. In all quantum search algorithms the overall complexity is of the order of these specific costs, which justifies their choices. The operations not involving manipulations of the data will be charged at unit cost. For the sake of simplicity, we do not formally include the data into the description of the unitary operations defining the costs. The initial state of the algorithm is explicitly related to the stationary distribution π of P.

(Quantum) Setup cost S**:** The cost for constructing the state $\sum_{x\in X}\sqrt{\pi_x}|x\rangle|\bar{0}\rangle$ with data.
(Quantum) Update cost U**:** The cost to realize any of the unitary transformations and inverses with data

$$\begin{array}{rcl} |x\rangle|\bar{0}\rangle & \mapsto & |x\rangle\sum_{y\in X}\sqrt{p_{xy}}|y\rangle, \\ |\bar{0}\rangle|y\rangle & \mapsto & \sum_{x\in X}\sqrt{p^*_{yx}}|x\rangle|y\rangle. \end{array}$$

(Quantum) Checking cost C**:** The cost to realize the unitary transformation with data, that maps $|x\rangle|y\rangle$ to $-|x\rangle|y\rangle$ if $x \in M$, and leaves it unchanged otherwise.

In the checking cost we could have included the cost of the unitary transformation which realizes a phase flip also when $y \in M$, our choice was made just for simplicity. Observe that the quantum walk $W(P)$ with data can be implemented at cost $4\mathsf{U} + 2$. Indeed, the reflection $\mathrm{ref}(\mathcal{A})$ is implemented by mapping states $|x\rangle|p_x\rangle$ to $|x\rangle|\bar{0}\rangle$, applying $\mathrm{ref}(\mathbb{C}^X \otimes |\bar{0}\rangle)$, and undoing the first transformation. In our accounting we charge unit cost for the second step since it does not depend on the database. Therefore the implementation of $\mathrm{ref}(\mathcal{A})$ is of cost $2\mathsf{U} + 1$. The reflection $\mathrm{ref}(\mathcal{B})$ may be implemented similarly.

Let us now describe how the respective algorithms of Grover, Ambainis and Szegedy are related to the classical search algorithms of Section 2. We suppose that ε, a lower bound on the proportion of marked elements is known in advance, though the results remain true even if it is not the case. Grover search (which we discuss soon in detail) is the quantum analogue of **Search Algorithm 1**.

Theorem 2 (Grover [16]). *There exists a quantum algorithm which with high probability finds a marked element, if there is any, at cost of order* $\frac{\mathsf{S}+\mathsf{C}}{\sqrt{\varepsilon}}$.

In the original application of Grover's result to unordered search there is no data structure involved, therefore $\mathsf{S}+\mathsf{C}=O(1)$, and the cost is of order $\frac{1}{\sqrt{\varepsilon}}$.

The algorithm of Ambainis is the quantum analogue of **Search Algorithm 2** in the special case of the walk on the Johnson graph and for some specific marked sets. Let us recall that for $0 < r \leq n/2$, the vertices of the Johnson graph $J(n,r)$ are the subsets of $[n]$ of size r, and there is an edge between two vertices if the size of their symmetric difference is 2. In other words, two vertices are adjacent if by deleting an element from the first one and adding a new element to it we get the second vertex. The eigenvalue gap δ of the symmetric walk on $J(n,r)$ is $n/r(n-r)=\Theta(1/r)$. If the set of marked vertices in $J(n,r)$ is either empty, or it consists of vertices that contain a fixed subset of constant size $k \leq r$ then $\varepsilon=\Omega(\frac{r^k}{n^k})$.

Theorem 3 (Ambainis [7]). *Let P be the random walk on the Johnson graph $J(n,r)$ where $r=\mathrm{o}(n)$, and let M be either empty, or the class of vertices that contain a fixed subset of constant size $k \leq r$. Then there is a quantum algorithm that finds, with high probability, the k-subset if M is not empty at cost of order* $\mathsf{S}+\frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathsf{U}+\mathsf{C})$.

Szegedy's algorithm is the quantum analogue of **Search Algorithm 3** for the class of ergodic and symmetric Markov chains. His algorithm is therefore more general than the one of Ambainis with respect to the class of Markov chains and marked sets it can deal with. Nonetheless, the approach of Ambainis has its own advantages: it is of smaller cost when C is substantially greater than U, and it also finds a marked element.

Theorem 4 (Szegedy [35]). *Let P be an ergodic and symmetric Markov chain. There exists a quantum algorithm that determines, with high probability, if M is non-empty at cost of order* $\mathsf{S}+\frac{1}{\sqrt{\delta\varepsilon}}(\mathsf{U}+\mathsf{C})$.

The MNRS algorithm is a quantum analogue of **Search Algorithm 2** for ergodic and reversible Markov chains. It generalizes the algorithms of Ambainis an Szegedy, and it combines their benefits in terms of being able to find marked elements, incurring the smaller cost of the two, and being applicable to a larger class of Markov chain.

Theorem 5 (Magniez et al. [25]). *Let P be an ergodic and reversible Markov chain, and let $\varepsilon > 0$ be a lower bound on the probability that an element chosen from the stationary distribution of P is marked whenever M is non-empty. Then, there exists a quantum algorithm which finds, with high probability, an element of M if there is any at cost of order* $\mathsf{S}+\frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}}\mathsf{U}+\mathsf{C})$.

There is an additional feature of Szegedy's algorithm which doesn't fit into the MNRS algorithmic paradigm. In fact, the quantity $\frac{1}{\sqrt{\delta\varepsilon}}$ in Theorem 4 can

be replaced by the square root of the classical hitting time [35]. The search algorithm for the 2-dimensional grid obtained this way, and the one given in [9] have smaller complexity than what follows from Theorem 5.

5 The MNRS search algorithm

We give a high level description of the MNRS search algorithm. Assume that $M \neq \emptyset$. Let $\mathcal{M} = \mathbb{C}^{M \times X}$ denote the marked subspace, that is the subspace with marked items in the first register. The purpose of the algorithm is to approximately transform the initial state $|\pi\rangle$ to the target state $|\mu\rangle$, which is the normalized projection of $|\pi\rangle$ onto $\mathcal{M}$:

$$|\mu\rangle \quad = \quad \frac{\Pi_{\mathcal{M}}|\pi\rangle}{\|\Pi_{\mathcal{M}}|\pi\rangle\|} \quad = \quad \frac{1}{\sqrt{\varepsilon}} \sum_{x \in M} \sqrt{\pi_x}\, |x\rangle |p_x\rangle,$$

where $\varepsilon = \|\Pi_{\mathcal{M}}|\pi\rangle\|^2 = \sum_{x \in M} \pi_x$ is the probability of the set M of marked states under the stationary distribution π. Let us recall that Grover search [16] solves this problem via the iterated use of the rotation $\mathrm{ref}(\pi) \cdot \mathrm{ref}(\mu^\perp)$ in the two-dimensional real subspace $\mathcal{S} = \mathsf{Span}(|\pi\rangle, |\mu\rangle)$, where $|\mu^\perp\rangle$ is the state in $\mathcal{S}$ orthogonal to $|\mu\rangle$ making some acute angle φ with $|\pi\rangle$. The angle φ is given by $\sin\varphi = \langle\mu|\pi\rangle = \sqrt{\varepsilon}$. Then $\mathrm{ref}(\pi) \cdot \mathrm{ref}(\mu^\perp)$ is a rotation by 2φ within the space $\mathcal{S}$, and therefore $\mathrm{O}(1/\varphi) = \mathrm{O}(1/\sqrt{\varepsilon})$ iterations of this rotation, starting with $|\pi\rangle$, approximates well $|\mu\rangle$. The MNRS search basically follows this idea.

Restricted to the subspace $\mathcal{S}$, the operator $\mathrm{ref}(\mu^\perp)$ is identical to $-\mathrm{ref}(\mathcal{M})$. Therefore, if the state of the algorithm remains close to the subspace $\mathcal{S}$ throughout, it can be implemented at the price of the checking cost. The reflection $\mathrm{ref}(\pi)$ is computationally harder to perform. The idea is to apply the phase estimation algorithms of Kitaev [21] and Cleve at al. [13] to $W(P)$. Corollary 1 implies that $|\pi\rangle$ is the only 1-eigenvector of $W(P)$, and all the other eigenvectors have phase at least Δ. Phase estimation approximately resolves any state $|\psi\rangle$ in $\mathcal{A}+\mathcal{B}$ along the eigenvectors of $W(P)$, and thus distinguishes $|\pi\rangle$ from all the others. Therefore it is possible to flip the phase of all states with a non-zero estimate of the phase, that is simulate the effect of the operator $\mathrm{ref}(\pi)$ in $\mathcal{A} + \mathcal{B}$. The following result of [25] resumes this discussion:

Theorem 6. *There exists a uniform family of quantum circuits $R(P)$ that uses $O(k \log(\Delta^{-1}))$ additional qubits and satisfies the following properties:*

1. *It makes $O(k\Delta^{-1})$ calls to the controlled quantum walk $\mathrm{c}{-}W(P)$ and its inverse.*
2. *$R(P)|\pi\rangle = |\pi\rangle$.*
3. *If $|\psi\rangle \in \mathcal{A} + \mathcal{B}$ is orthogonal to $|\pi\rangle$, then $\|(R(P) + \mathrm{Id})|\psi\rangle\| \leq 2^{-k}$.*

The essence of the MNRS search algorithm is the following simple procedure that satisfies the conditions of Theorem 5, but with a slightly higher complexity, of the order of $\mathsf{S} + \frac{1}{\sqrt{\varepsilon}}(\frac{1}{\sqrt{\delta}} \log \frac{1}{\sqrt{\varepsilon}} \mathsf{U} + \mathsf{C})$. Again, we suppose that ε is known.

Quantum Search(P)
1. Start from the initial state $|\pi\rangle$.
2. Repeat $O(1/\sqrt{\varepsilon})$–times:
   (a) For any basis vector $|x\rangle|y\rangle$, flip the phase if $x \in M$.
   (b) Apply circuit $R(P)$ of Theorem 6 with $k = O(\log(1/\sqrt{\varepsilon}))$.
3. Observe the first register and output if it is in M.

To see the correctness, let $|\phi_i\rangle$ be the result of i iterations of $\mathrm{ref}(\pi) \cdot \mathrm{ref}(\mu^{\perp})$ applied to $|\pi\rangle$, and let $|\psi_i\rangle$ be the result of i iterations of step (2) in **Quantum Search**(P) applied to $|\pi\rangle$. It is not hard to show by induction on i, using a hybrid argument as in [10, 36], that $\||\psi_i\rangle - |\phi_i\rangle\| \leq O(i2^{-k})$. This implies that $\||\psi_k\rangle - |\phi_k\rangle\|$ is an arbitrarily small constant when k is chosen to be $O(\log(1/\sqrt{\varepsilon}))$ and therefore the success probability is arbitrarily close to 1.

The cost of the procedure is simple to analyze. Initialization costs $\mathsf{S} + \mathsf{U}$, and in each iteration the single phase flip costs C. In the circuit $R(P)$, the controlled quantum walk and its inverse can be implemented, similarly to $W(P)$, at cost $4\mathsf{U} + 2$, simply by controlling $\mathrm{ref}(\mathbb{C}^X \otimes |\bar{0}\rangle)$ and $\mathrm{ref}(|\bar{0}\rangle \otimes \mathbb{C}^Y)$. The number of steps of the controlled quantum walk and its inverse is $O((1/\Delta)\log(1/\sqrt{\varepsilon}))$. Since $\Delta \geq 2\sqrt{\delta}$, this finishes the cost analysis. Observe that the $\log(1/\sqrt{\varepsilon})$-factor in the update cost was necessary for reducing the error of the approximate reflection operator. In [25] it is described how it can be eliminated by adapting the recursive amplitude amplification algorithm of Høyer et al. [18]

6 Applications

We give here a few examples where the quantum search algorithms, in particular the MNRS algorithm can be applied successfully. All examples will be described in the query model of computation. Here the input is given by an oracle, a query can be performed at unit cost, and all other computational steps are free. A formal description of the model can be found for example in [23] or [26]. In fact, in almost all cases, the circuit complexity of the algorithms given will be of the order of the query complexity, with additional logarithmic factors.

6.1 Grover search

As a first (and trivial) application, we observe that Grover's algorithm [16] for the unordered search problem is a special case of Theorem 5.

Unordered Search
Oracle Input: A boolean function f defined on $[n]$.
Output: An element $i \in [n]$ such that $f(i) = 1$.

Theorem 7. Unordered Search *can be solved with high probability in quantum query complexity* $O((n/k)^{1/2})$*, where* $|\{i \in [n] \ : \ f(i) = 1\}| = k$.

Proof. Consider the symmetric random walk in the complete graph on n vertices, where an element v is marked if $f(v) = 1$. The eigenvalue gap of the walk is $1 - \frac{1}{n-1}$, and the probability ε that an element is marked is k/n. There is no data structure involved in the algorithm, the setup, update and checking costs are 1.

6.2 Johnson graph based algorithms

All these examples are based on the symmetric walk in the Johnson graph $J(n, r)$, with eigenvalue gap $\Theta(1/r)$.

Element Distinctness This is the original problem for which Ambainis introduced the quantum walk based search method [7].

> Element Distinctness
> *Oracle Input:* A function f defined on $[n]$.
> *Output:* A pair of distinct elements $i, j \in [n]$ such that $f(i) = f(j)$ if there is any, otherwise reject.

Theorem 8. Element Distinctness *can be solved with high probability in quantum query complexity* $O(n^{2/3})$.

Proof. A vertex $R \subseteq [n]$ of size r is marked if there exist $i \neq j \in R$ such $f(i) = f(j)$. The probability ε that an element is marked, if there is any, is in $\Omega((r/n)^2)$. For every R, the data is defined as $\{(v, f(v)) : v \in R\}$. Then the setup cost is in $O(r)$, the update cost is $O(1)$, and the checking cost is 0. Therefore the overall cost is $O(r + n/r^{1/2})$ which is $O(n^{2/3})$ when $r = n^{2/3}$. This upper bound is tight, the $\Omega(n^{2/3})$ lower bound is due to Aaronson and Shi [1].

Matrix Product Verification This problem was studied by Buhrman and Spalek [12], and the algorithm in the query model is almost identical to the previous one.

> Matrix Product Verification
> *Oracle Input:* Three $n \times n$ matrices A, B and C.
> *Output:* Decide if $AB = C$ and in the negative case find indices i, j such that $(AB)_{ij} \neq C_{ij}$.

Theorem 9. Matrix Product Verification *can be solved with high probability in quantum query complexity* $O(n^{5/3})$.

Proof. For an $n \times n$ matrix M and a subset of indices $R \subseteq [n]$, let $M|_R$ denote the $|R| \times n$ submatrix of M corresponding to the rows restricted to R. The submatrices $M|^R$ and $M|^R_R$ are defined similarly, when the restriction concerns the columns and both the rows and columns. A vertex $R \subseteq [n]$ is marked if there exist $i, j \in R$ such that $AB_{ij} \neq C_{ij}$. The probability of being marked, if

there is such an element, is in $\Omega((r/n)^2)$. For every R, the data is defined as the set of entries in $A|_R, B|^R$ and $C|^R_R$. Then the setup cost is in $O(rn)$, the update cost is $O(n)$, and the checking cost is 0. Therefore the overall cost is $O(n^{5/3})$ when $r = n^{2/3}$. The best known lower bound is $\Omega(n^{3/2})$, and in [12] an $O(n^{5/3})$ upper bound was also proven for the time complexity with a somewhat more complicated argument.

Restricted Range Associativity The problem here is to decide if a binary operation $\circ$ is associative. The only algorithm known for the general case is Grover search, but Dörn and Thierauf [15] have proved that when the range of the operation is restricted, Theorem 5 (or Theorem 3) give a non-trivial bound. A triple (a, b, c) is called *non associative* if $(a \circ b) \circ c \neq a \circ (b \circ c)$.

Restricted Range Associativity
Oracle Input: A binary operation $\circ : [n] \times [n] \to [k]$ where $k \in O(1)$.
Output: A non associative triple (a, b, c) if there is any, otherwise reject.

Theorem 10. Restricted Range Associativity *can be solved with high probability in quantum query complexity* $O(n^{5/4})$.

Proof. We say that $R \subseteq [n]$ is marked if there exist $a, b \in R$ and $c \in [n]$ such that (a, b, c) is non associative. Therefore ε is in $\Omega((r/n)^2)$, if there is a marked element. For every $R \subseteq [n]$, the data structure is defined as $\{(a, b, a \circ b) : a, b \in R \cup [k]\}$. Then the setup cost is $O((r + k)^2 = O(r^2)$ and the update cost is $O(r + k) = O(r)$. Observe that if $b \in R$ and $c \in [n]$ are fixed, then computing $(a \circ b) \circ c$ and $a \circ (b \circ c)$ for all $a \in R$ requires at most $k+1$ queries with the help of the data structure. Thus using Grover search to find b and c, the checking cost is $O(k\sqrt{rn}) = O(\sqrt{rn})$. The overall complexity is then $O(r^2 + \frac{n}{r}(\sqrt{r}r + \sqrt{rn}))$ which is $O(n^{5/4})$ when $r = \sqrt{n}$. The best lower bound known both in the restricted range and the general case is $\Omega(n)$.

Triangle In an undirected graph G, a complete subgraph on three vertices is called a *triangle*. The algorithm of Magniez et al. [26] for finding a triangle uses the algorithm for Element Distinctness in the checking procedure.

Triangle
Oracle Input: The adjacency matrix f of a graph G on vertex set $[n]$.
Output: A triangle if there is any, otherwise reject.

Theorem 11. Triangle *can be solved with high probability in quantum query complexity* $O(n^{13/10})$.

Proof. We show how to find the edge of a triangle, if there is any, in query complexity $O(n^{13/10})$. This implies the theorem since given such an edge, Grover search finds the third vertex of the triangle with $O(n^{1/2})$ additional queries.

An element $R \subseteq [n]$ is marked if it contains a triangle edge. The probability ε that an element is marked is in $\Omega((r/n)^2)$, if there is a triangle. For every R, the data structure is the adjacency matrix of the subgraph induced by R. defined as $\{(v, f(v)) : v \in R\}$. Then the setup cost is $O(r^2)$, and the update cost is $O(r)$. The interesting part of the algorithm is the checking procedure, which is a quantum walk based search itself, and the claim is that it can be done at cost $O(\sqrt{n} \times r^{2/3})$.

To see this, let R be a set of r vertices such that the graph G restricted to R is explicitly known, and for which we would like to decide if it is marked. Observe that R is marked exactly when there exists a vertex $v \in [n]$ such that v and an edge in R form a triangle. Therefore for every vertex v, one can define a secondary search problem on R via the boolean oracle f_v, where for every $u \in R$, by definition $f_v(u) = 1$ if $\{u, v\}$ is an edge. The output of the problem is by definition positive if there is an edge $\{u, u'\}$ such that $f_v(u) = f_v(u') = 1$. To solve the problem we consider the Johnson graph $J(r, r^{2/3})$, and look for a subset which contains such an edge. In that search problem both the probability of being marked and the eigenvalue gap of the underlying Markov chain are in $\Omega(r^{-2/3})$. The data associated with a subset of R is just the values of f_v at its elements. Then the setup cost is $r^{2/3}$, the update cost is $O(1)$, and the checking cost is 0. Therefore by Theorem 5 the cost of solving a secondary search problem is in $O(r^{2/3})$. Finally the checking procedure of the original search problem consists of a Grover search for a vertex v such that the secondary search problem defined by f_v has a positive outcome. Putting things together, the problem can be solved in quantum query complexity $O(r^2 + \frac{n}{r}(\sqrt{r} \times r + \sqrt{n} \times r^{2/3}))$ which is $O(n^{13/10})$ when $r = n^{3/5}$. The best known lower bound for the problem is $\Omega(n)$.

6.3 Group commutativity

The problem here is to decide if a group multiplication is commutative in the (sub)group generated by some set of group elements. It was defined and studied in the probabilistic case by Pak [31], the quantum algorithm is due to Magniez and Nayak [23].

> Group Commutativity
> *Oracle Input:* The multiplication operation $\circ$ for a finite group whose base set contains $[n]$.
> *Output:* A non commutative couple $(i, j) \in [n] \times [n]$ if G, the (sub)group generated by $[n]$, is non-commutative, otherwise reject.

Theorem 12. Group Commutativity *can be solved with high probability in quantum query complexity* $O(n^{2/3} \log n)$.

Proof. For $0 < r < n$ let $S(n, r)$ be the set of all r-tuples of distinct elements from $[n]$. For $u = (u_1, \ldots, u_r)$ in $S(n, r)$, we set $\bar{u} = u_1 \circ \cdots \circ u_r$. We define a random walk over $S(n, r)$. Let $u = (u_1, \ldots, u_r)$ be the current vertex. Then with probability $1/2$ stay at u, and with probability $1/2$ pick uniformly random

$i \in [r]$ and $j \in [n]$. If $j = u_m$ for some m then exchange u_i and u_m, otherwise set $u_i = j$. The random walk P at the basis of the quantum algorithm is over $S(n,r) \times S(n,r)$, and it consists of two independent simultaneous copies of the above walk. The stationary distribution of P is the uniform distribution, and it is proven in [23] that its eigenvalue gap is $\Omega(1/(r \log r))$.

A vertex (u,v) is marked if $\bar{u} \circ \bar{v} \neq \bar{v} \circ \bar{u}$. It is proven again in [23] that when G is non-commutative and $r \in o(n)$, then the probability ε that an element is marked is $\Theta(r^2/n^2)$. For $u \in S(n,r)$ let T_u be the balanced binary tree with r leaves that are labeled from left to right by $u_1, \ldots, u_r$, and where each internal node is labeled by the product of the labels of its two sons. For every vertex (u,v) the data consists of (T_u, T_v). Then the setup cost is r, and the update cost is $O(\log r)$ for recomputing the leaf–root paths. The checking cost is simply 2 for querying $\bar{u} \circ \bar{v}$ and $\bar{v} \circ \bar{u}$. Therefore the query complexity to find a marked element is $O(r + \frac{n}{r}(\sqrt{r \log r} \log r + 1))$ which is $O(n^{2/3} \log n)$ when $r = n^{2/3} \log n$. Once a marked element is found, Grover search yields a non-commutative couple at cost $O(r)$. In [23] an $\Omega(n^{2/3})$ lower bound is also proven. And, it turns out that a Johnson graph based walk can be applied to this problem too [24], yielding an algorithm of complexity $O((n \log n)^{2/3})$.

Acknowledgment

I would like to thank Frédéric Magniez, Ashwin Nayak and Jérémie Roland, my coauthors in [25], for letting me to include here several ideas which were developed during that work, and for numerous helpful suggestions.

References

1. S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, pages 595–605, 2004.
2. S. Aaronson and A. Ambainis. Quantum search of spatial regions. *Theory of Computing*, 1:47–79, 2005.
3. D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani. Quantum walks on graphs. In *Proc. of the 33rd ACM Symposium on Theory of Computing*, pages 50–59, 2001.
4. R. Aleliunas, R. Karp, R. Lipton, L. Lovász, and C. Rackoff. Random Walks, Universal Traversal Sequences, and the Complexity of Maze Problems. In *Proc. of the 20th Symposium on Foundations of Computer Science*, pages 218–223, 1979.
5. A. Ambainis. Quantum walks and their algorithmic applications. *International Journal of Quantum Information*, 1(4): 507–518, 2003.
6. A. Ambainis. Quantum search algorithms. *SIGACT News*, 35(2): 22–35, 2004.
7. A. Ambainis. Quantum Walk Algorithm for Element Distinctness. *SIAM Journal on Computing*, 37:210–239, 2007.
8. A. Ambanis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous. One-dimensional quantum walks. In *Proc. of the 33rd ACM Symposium on Theory of computing*, pages 37–49, 2001.
9. A. Ambainis, J. Kempe, and A. Rivosh. Coins make quantum walks faster. In *Proc. of the 16th ACM-SIAM Symposium on Discrete Algorithms*, pages 1099–1108, 2005.

10. C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.
11. G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In Jr. Samuel J. Lomonaco and Howard E. Brandt, editors, *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *Contemporary Mathematics Series*. American Mathematical Society, pages 53–74, 2002.
12. H. Buhrman and R. Spalek. Quantum verification of matrix products. In *Proc. of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 880–889, 2006.
13. R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. *Proc. of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 454(1969):339–354, 1998.
14. T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. Second Edition The MIT Press and McGraw-Hill, 2001.
15. S. Dörn, and T. Thierauf. The Quantum Query Complexity of Algebraic Properties. In *Proc. of the 16th International Symposium on Fundamentals of Computation Theory*, pages 250–260, 2007.
16. L. Grover. A fast quantum mechanical algorithm for database search. In *Proc. of the 28th ACM Symposium on the Theory of Computing*, pages 212–219, 1996.
17. P. Halmos. *Finite-dimensional vector spaces*. Springer Verlag, 1974.
18. P. Høyer, M. Mosca, and R. de Wolf. Quantum search on bounded-error inputs. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming*, volume 2719 of *Lecture Notes in Computer Science*, pages 291–299, 2003.
19. J. Kempe. Discrete Quantum Walks Hit Exponentially Faster. In *Proc. of the International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 354–369, 2003.
20. J. Kempe. Quantum random walks – an introductory survey, *Contemporary Physics*, 44(4): 307-327, 2003.
21. A. Kitaev. Quantum measurements and the abelian stabilizer problem. *Electronic Colloquium on Computational Complexity (ECCC)*, 3, 1996.
22. D. Knuth. *Sorting and Searching*. Volume 3 of *The Art of Computer Programming*. Addison-Wesley, 1973
23. F. Magniez and A. Nayak. Quantum complexity of testing group commutativity. *Algorithmica*, 48(3):221–232, 2007.
24. F. Magniez and A. Nayak. Personal communication, 2008.
25. F. Magniez, A. Nayak, J. Roland, and M. Santha. Search via quantum walk. In *Proc. of the 39th ACM Symposium on Theory of Computing*, pages 575–584, 2007.
26. F. Magniez, M. Santha, and M. Szegedy. Quantum Algorithms for the Triangle Problem. *SIAM Journal of Computing*, 37(2):413–427, 2007.
27. D. Meyer. From quantum cellular automata to quantum lattice gases. *Journal of Statistical Physics*, 85(5-6):551–574, 1996.
28. D. Meyer. On the abscence of homogeneous scalar unitary cellular automata. *Physical Letter A*, 223(5):337–340, 1996.
29. C. Moore, and Alexander Russell. Quantum Walks on the Hypercube. In *Proc. of the 6th International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 164–178, 2002.
30. A. Nayak and A. Vishwanath. Quantum walk on the line. Technical Report quant-ph/0010117, arXiv, 2000.
31. I. Pak. Testing commutativity of a group and the power of randomization. Electronic version at http://www-math.mit.edu/ pak/research.html, 2000.

32. P. Richter. Almost uniform sampling via quantum walks. *New Journal of Physics*, Vol. 9, 72, 2007.
33. U. Schöning. A Probabilistic Algorithm for k -SAT Based on Limited Local Search and Restart. *Algorithmica* 32(4): 615–623, 2002.
34. N. Shenvi, J. Kempe, and K.B. Whaley. Quantum random-walk search algorithm. *Physical Review A*, 67(052307), 2003.
35. M. Szegedy. Quantum Speed-Up of Markov Chain Based Algorithms. In *Proc. of the 45th IEEE Symposium on Foundations of Computer Science*, pages 32–41, 2004.
36. U. Vazirani. On the power of quantum computation. *Philosophical Transactions of the Royal Society of London, Series A*, 356:1759–1768, 1998.
37. J. Watrous. Quantum simulations of classical random walks and undirected graph connectivity. *Journal of Computer and System Sciences*, 62(2):376–391, 2001.