

# Faster transport with a directed quantum walk

Stephan Hoyer\* and David A. Meyer†

*Department of Mathematics, University of California, San Diego, CA 92093*

(Dated: January 14, 2009)

We give the first example of faster transport with a quantum walk on an inherently directed graph, on the directed line with a variable number of self-loops at each vertex. These self-loops can be thought of as adding a number of small dimensions. This is a discrete time quantum walk using the Fourier transform coin, where the walk proceeds a distance  $\Theta(1)$  in constant time compared to  $\Theta(1/n)$  classically, independent of the number of these small dimensions. The analysis proceeds by reducing this walk to a walk with a two dimensional coin.

PACS numbers: 03.67.Ac

## I. INTRODUCTION

A quantum random walk is the quantum analog of a classical random walk where the walker may now exist in a superposition of locations. These quantum walks offer improvement in various measures of speed over corresponding classical versions, and they can be run efficiently on quantum computers. Accordingly, they form the basis for a family of quantum algorithms that are the accelerated version of classical randomized algorithms [1, 2], and provide one of few alternatives to algorithms based on Grover's search or Shor's factoring algorithm. Significant examples include structured search [3] and exponential speedup solving a randomized graph problem [4]. To date, all applications of quantum walks have all been on undirected graphs, but many classical random walk algorithms of interest use directed graphs, such as classes of Markov chain Monte Carlo simulations and algorithms for fast solving of satisfiability problems [5]. It would be very significant to accelerate these algorithms using quantum walks.

Quantum walks on arbitrary undirected graphs can be defined in both continuous and discrete time. In continuous time, a Hamiltonian  $H$  is given by the connectivity matrix of the graph, so unitary evolution is given by the operator  $e^{iHt}$ . In discrete time, additional degrees of freedom are generally added by associating an independent state to every distinct vertex and adjacent edge pair  $|v, e\rangle$ , where  $e$  is an edge connecting  $v$  to  $v'$  [6]. If the number of adjacent edges is constant across vertices, then this is equivalent to the direct product of vertex space  $\mathcal{H}_v$  and a "coin space"  $\mathcal{H}_c$ . In this case there are two operations associated with the walk, a coin  $C$  and shift  $S$ .  $C$  performs some unitary operation in the coin space  $\mathcal{H}_c$ , and  $S$  performs the shift  $|v, e\rangle \rightarrow |v', e\rangle$ . The evolution of the walk is given by repeated application of the operation for a single time step,  $U = S \cdot (I \otimes C)$ . In the case of general undirected graphs, a generalized coin replacing  $I \otimes C$  may have a different action at each vertex, but still operates independently on the subspaces corresponding to each vertex.

Directed quantum walks have been considered previously, but unitary evolution on arbitrary directed graphs requires reversibility that is not always available. Montanaro has shown that a discrete time directed walk may be constructed for any graph where it is possible to return to every position by associating coins with each cycle in the graph, and suggests making graphs irreversible by partial measurement of regions [7]. This is a prescription for a quantum walk but under continuous measurement it no longer corresponds to the classical walk. This walk also requires global knowledge of the graph and thus seems unlikely to be useful for algorithms, as in most cases this is equivalent to already knowing the answer to the problem the graph represents.

In contrast, following Severini [8] we identify an arbitrary quantum walk on a directed graph by associating each edge of the graph with an independent state, very similarly to the general prescription for undirected graphs. Then any pairing between incoming and outgoing vertices on a graph where each vertex has the same number of edges in and out will allow for unitary evolution. This condition on suitable graphs is clearly sufficient; it can also be shown to be necessary [8]. The coin operator  $C$  applies some unitary transformation in the subspace associated with the source of each edge at a vertex. Then the shift operator  $S$  transfers the amplitudes associated with each edge incoming to a vertex to some edge outgoing from the same vertex. Implicit in the choice of shift operator is some choice of pairings between incoming and outgoing edges. In undirected walks this is done by default, by pairing incoming edges with themselves, but in the more general case of directed walks this option is not always available.

Our objective is algorithmic speedup, but before we can expect speedup using a directed walk to yield an algorithm, precedence suggests that we should first identify a toy example to show that such speedup is possible. For undirected walks, this example is the walk along the line, where the expectation value of the position increases as  $\Theta(t)$  instead of  $\Theta(\sqrt{t})$  classically [9]. However, as yet there have been no concrete examples of quantum walks on fundamentally directed graphs that are in any sense faster than their classical equivalents. We present the first such directed walk, with an improvement in the expectation value of position along a directed line in constant time to a constant  $\Theta(1)$  from  $\Theta(1/n)$  classically, where  $n$  indicates the dimensionality of the graph.

---

\*Author to whom correspondence should be addressed; Present address: Department of Physics, University of California, Berkeley, CA 94720; Electronic address: shoyer@berkeley.edu

†Electronic address: dmeyer@math.ucsd.edu

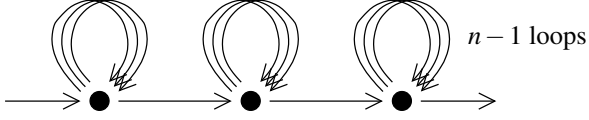


FIG. 1: Directed walk on the line with  $n-1$  self-loops at each vertex.

## II. DIRECTED WALK WITH SELF-LOOPS

Consider a directed random walk along a line with  $n-1$  self-loops at each vertex, as shown in Fig. 1. Alternatively, we can consider these self-loops as paths around an infinitesimally small  $n-1$  dimensional spatial torus  $T$  and the graph as a discretization of  $\mathbb{R} \times T$ . Initialized at a vertex  $x=0$ , classically there is a probability  $1/n$  of moving to the next vertex, so the expected position  $\langle x \rangle = t/n$  after  $t$  steps, which is  $\Theta(1/n)$ . This graph is necessarily more complicated than the base case for undirected walks, because simple classical directed walks oriented in one direction already proceed a distance  $\Theta(t)$  leaving no room for improvement, unlike  $\Theta(\sqrt{t})$  for the undirected walk. Accordingly we seek a different kind of speedup.

In the quantum version of this walk, we assign a basis vector to each edge on our graph, identifying each edge with the vertex at its base. Thus any state is a linear combination of the states

$$|x, \rightarrow\rangle, |x, 1\rangle, |x, 2\rangle, \dots, |x, n-1\rangle, \quad (1)$$

for all non-negative integer values of  $x$ , where  $\rightarrow$  indicates the edge along the line and each number indicates a distinct self-loop. Recall the  $n$ -dimensional discrete Fourier transform may be written as

$$C = \frac{1}{\sqrt{n}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{n-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{n-1} & \omega^{2(n-1)} & \dots & \omega^{(n-1)(n-1)} \end{pmatrix}, \quad (2)$$

where  $\omega = e^{2\pi i/n}$  is the  $n$ th root of unity. Then one step of our quantum walk is given by applying  $C$  as a coin operator on the coin space in the basis given by Eq. (1), and then applying the shift operator  $S$  transferring amplitudes from incoming edges at a vertex to outgoing edges. These edges are paired by the shift operator in the only non-symmetry breaking manner, by associating self-loops with themselves and pairing the incoming and outgoing paths along the line,

$$S|x, \rightarrow\rangle = |x+1, \rightarrow\rangle \quad (3a)$$

$$S|x, k\rangle = |x, k\rangle. \quad (3b)$$

We initialize this walk in the state  $|0, \rightarrow\rangle$  and repeatedly apply our time evolution operator. As with many other such proofs with quantum walks, our proof of its performance relies on reduction to the quantum walk on the line. The performance gains from this quantum walk can also be verified numerically, as shown in Fig. 2.

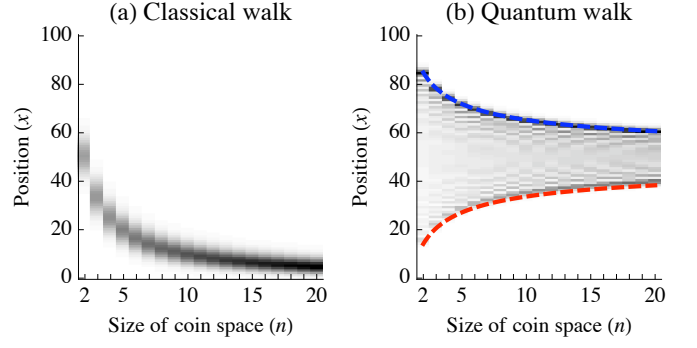


FIG. 2: (Color online) Probability of measurement at each position after 100 time steps as a function of the size of the coin space, for (a) classical and (b) quantum walks on the graph shown in Fig. 1. The boundaries of the interval for the quantum walk given by Eq. (8) are indicated by dashed lines.

**Theorem.** *This quantum walk proceeds an expected distance  $\Theta(1)$  in constant time independent of the number of self-loops.*

*Proof.* Consider the action of one iteration of the time evolution operator  $U = S \cdot (I \otimes C)$  on the initial state  $|x, \rightarrow\rangle$ ,

$$U|x, \rightarrow\rangle = \frac{1}{\sqrt{n}}|x+1, \rightarrow\rangle + \sqrt{\frac{n-1}{n}}|x, \odot\rangle, \quad (4)$$

where as the amplitude on each self-loop is the same, they are grouped into a single normalized state

$$|x, \odot\rangle = \frac{1}{\sqrt{n-1}} \sum_{k=1}^{n-1} |x, k\rangle. \quad (5)$$

Now consider the action of  $U$  on the superposition of loops state  $|x, \odot\rangle$ . It is given by

$$U|x, \odot\rangle = \sqrt{\frac{n-1}{n}}|x+1, \rightarrow\rangle - \frac{1}{\sqrt{n}}|x, \odot\rangle, \quad (6)$$

as each non-zero power of  $\omega$  in the Fourier transform is a root of the polynomial

$$x^{n-1} + \dots + x + 1 = \prod_{k=1}^{n-1} (x - \omega^k)$$

and thus for all  $k$ ,

$$\omega^k + (\omega^k)^2 + \dots + (\omega^k)^{n-1} = -1.$$

Equations (4) and (6) reduce this walk to a quantum walk along the line with the two dimensional coin

$$C' = \begin{pmatrix} \beta & \alpha \\ \alpha & -\beta \end{pmatrix}, \quad \text{where } \alpha = \sqrt{\frac{n-1}{n}}, \quad \beta = \frac{1}{\sqrt{n}}, \quad (7)$$

on the coin space given by  $\{\rightarrow, \odot\}$ . Such a walk can be easily shown to be equivalent to the standard quantum walk on the line with the coin space  $\{R, L\}$  under a simple map and thus this walk proceeds similarly in linear time. But as Bressler and Pemantle have proven directly with this coin space [10],

the form of the coin matrix  $C'$  immediately specifies the probability distribution as spread over the interval

$$\left[ \frac{1-\beta}{2}t, \frac{1+\beta}{2}t \right], \quad (8)$$

where the probability of being found outside the interval decays at least as fast as  $t^{-N}$  for any integer  $N > 0$ . Figure 2(b) shows the bounds of this interval. Thus this walk proceeds a constant distance  $\Theta(1)$  in constant time independent of the number of loops  $n-1$ , and as  $n \rightarrow \infty$  we have  $\beta \rightarrow 0$ , so  $x \sim t/2$ .  $\square$

As a caveat, the speed of this walk depends completely on the natural pairing of incoming and outgoing edges. If these edges are paired randomly at each vertex, this quantum walk slows down to the classical speed. In algorithmic terms, this means the walk requires some local sense of where the solution is in order to proceed faster. This makes the improvement in speed with this walk less remarkable, as such information would not be accessible for use in search algorithms, to date the most significant direct algorithmic use of quantum walks.

### III. FURTHER DIRECTED QUANTUM WALKS

The speedup found here is certainly different from that in the classic examples for undirected quantum walks, but it may still be of note. The scaling of speed with the number of dimensions is also not unprecedented, as seen for instance by

performance gains with quantum walks on the binary hypercube, a variation of which is the graph searched by random walk satisfiability algorithms.

Numerical investigations suggest that similar results hold even when the loops are given a finite non-trivial discretization, and faster transport is also easy to find on other directed graphs constructed with natural edge pairings. The primary obstacle to finding algorithmic applications of these directed quantum walks is most likely that graphs representing real problems are unlikely to have enough symmetry to allow such meaningful pairing of edges. Since quantum walks such as this one rely on symmetry for useful quantum interference, this challenge may remain significant.

Also remaining is the challenge of constructing useful quantum walks on directed graphs that remain fundamentally irreversible. In this case, at vertices where there are more paths in than out, some sort of decoherent process must be used to reduce the dimensionality of the space in which the walker moves. This may be framed in terms of partial measurement of a path, or more broadly as a general quantum operation. Whether quantum speed-up could still be preserved through such operations remains unclear.

### Acknowledgments

SH was partially supported by NSF Research Experience for Undergraduates Grant No. PHY-0552402. DM was partially supported by ARO Grant No. W911NF071003.

- 
- [1] A. Ambainis, *Int. J. Quant. Inform.* **1**, 507 (2003), quant-ph/0403120.
  - [2] J. Kempe, *Contemp. Phys.* **44**, 307 (2003), quant-ph/0303081.
  - [3] N. Shenvi, J. Kempe, and K. B. Whaley, *Phys. Rev. A* **67**, 052307 (2003), quant-ph/0210064.
  - [4] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, in *ACM Symposium on Theory of Computing* (2003), pp. 59–68, quant-ph/0209131.
  - [5] U. Schöning, in *IEEE Symposium on Foundations of Computer Science* (1999), pp. 410–414.
  - [6] J. Watrous, in *IEEE Conference on Computational Complexity* (1999), pp. 180–187, cs/9812012.
  - [7] A. Montanaro, *Quant. Inform. Comput.* **7** (2007), quant-ph/0504116.
  - [8] S. Severini, *SIAM J. Matrix Anal. Appl.* **25**, 295 (2003), math/0205187.
  - [9] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and J. Watrous, in *ACM Symposium on Theory of Computing* (2001), pp. 37–49.
  - [10] A. Bressler and R. Pementle, in *DMTCS Conference on the Analysis of Algorithms* (2007), pp. 403–414.