Implementing a quantum walk on a general graph on a quantum computer

Viv Kendon[1,*]

[1]*Optics Section, Blackett Laboratory, Imperial College, London, SW7 2BW, United Kingdom.*

(Dated: June 13, 2003)

A scheme for a discrete time quantum walk on a general graph of N vertices with undirected edges is given in terms of an efficient gate sequence starting from one copy of a quantum state encoding the adjacency matrix of the graph. Most of the examples of quantum walks in the literature treat graphs of fixed, small ($\lesssim \log N$) degree. To obtain a quantum speed up over classical for comparable resources it is necessary to code the position space of the quantum walk into a qubit register (or equivalent). The general discrete quantum walk method also shows an exponential saving in resources to store the description of the graph.

One of the most important tasks on the theoretical side of quantum computing is the creation and understanding of quantum algorithms. The recent presentation of two quantum algorithms based on quantum versions of random walks is particularly important in this respect, since this provides a new type of algorithm which can show an exponential speed up over classical algorithms, to add to those based on the quantum Fourier transform. Childs et al [1] have produced a scheme for a continuous time quantum walk that can find its way across a particular graph exponentially faster than any classical algorithm, while Shenvi et al [2] proved that a quantum walk can reproduce the quadratically faster search times found with Grover's algorithm for finding a marked item in an unsorted database. For an overview of the development of quantum walks for quantum computing, see Kempe [3].

One point that has not been made explicit by most authors to date is that if, as is usually the case, the answer to the problem will be obtained by measuring the position of the particle, the quantum walk should have the position space encoded into a qubit register (or equivalent [4]). This is simply because for a classical random walk algorithm, the location of the particle can be encoded in a binary string, so we will penalise ourselves by requiring exponentially more resources if we adopt unary encoding for our quantum walk. Childs *et al.* [1] explicitly perform this encoding, but without remarking on this point, since their primary motivation is to show that they can implement a *continuous* time Hamiltonian evolution efficiently on a *discrete* quantum circuit. On the other hand, this places the proposed physical implementations of a quantum walk [5–7] firmly in the realm of physics, as all of them have the position space of the walk set up in such a way that it cannot be measured as a binary encoded bit string.

Here I describe how to apply a discrete time coined quantum walk to a general graph given only by the adjacency matrix. The continuous time quantum walk as originally presented by Farhi and Gutmann [8] is already formulated to do this. If one is also given extra information about the graph, e.g., that it is of bounded degree, and the neighbouring sites can be predicted from the current location, more efficient algorithms can be constructed for both classical and quantum versions of discrete and continuous time random walks. This reduction happens in general because we can encode the description of a more regular graph in a more efficient way than the adjacency matrix.

It is possible to encode the whole adjacency matrix $\mathbf{A}$, defined by $A_{xy} = 1$ if the vertices labeled x and y are joined by end edge and zero otherwise, into a single quantum state

$$|G\rangle = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} |x, y, A_{xy}\rangle, \tag{1}$$

where x and y run through all the vertex labels $0..(N-1)$. It would be useful, therefore, if we could generate one of these states just once, and use it as input the the quantum algorithm to perform the quantum walk, using additional qubits to record the vertex position of the walk, and the trial vertex "coin" for the next step. The state $|G\rangle$ is encoded in $2n+1$ qubits, where $n = \lceil \log N \rceil$. We need another $2n$ qubits to hold the values of x, y for the quantum walk, plus another $4n$ ancilla qubits to hold intermediate results. The ancilla qubits start in state $|0\rangle$ and finish in the same state.

Figure 1 shows schematically how the quantum circuit works. To update the position of the walk, we need to perform a conditional swap on the two registers of $|\psi\rangle$ for each pair (x, y) for which $A_{xy} = 1$. In Fig. 1, the process of computing a qubit that is $|1\rangle$ if $A_{xy} = 1$ is shown schematically for a two-qubit quantum walk state $|\psi\rangle$, the extension to $2n$ qubits is obvious. The three qubit Toffoli gates flip the value of the target qubit if the other two qubits are in state 0(1), denoted by white(black) circles. The controlled swap can be done as a sequence of Fredkin gates applied to each corresponding pair of qubits. It is then necessary to "uncompute" the values in the ancilla qubits so they are restored to state $|0\rangle$ ready for the next step of the walk. This is done by reversing the gate sequence used to compare the values of (x, y) in $|\psi\rangle$ and $|G\rangle$. Even though the state $|\psi\rangle$ has been conditionally swapped, because $A_{xy} = A_{yx}$, the state $|G\rangle$ is invariant under exchange of x and y so the uncompute step still works. Thus we see that this method will only work for undirected graphs. Finally, we need to "toss the coin" to produce a new trial value for the next step. This can be done using any efficient, i.e. $O(\text{poly}(n))$ gates unitary operation.

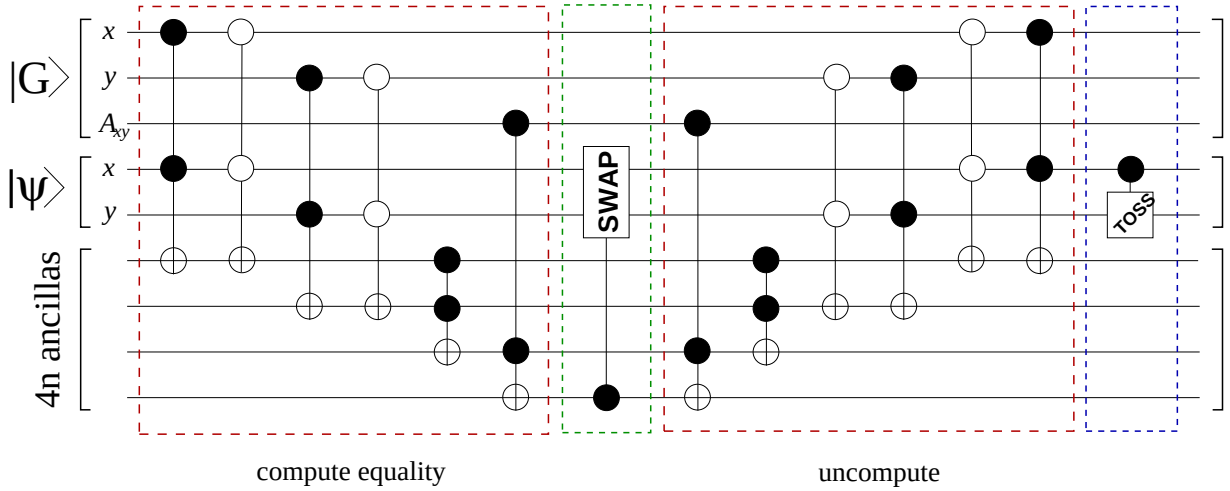

FIG. 1: Schematic circuit for one step of a discrete quantum walk. The graph is encoded in the state $|G\rangle$ and the walk position and coin state are recorded in $|\psi\rangle$. The three qubit gates are Toffoli gates conditioned on 1 (filled circles) or 0 (open circles) flipping the qubit under the cross. The conditional SWAP gate is a set of Fredkin gates on each pair of qubits, and the TOSS operation can be chosen to suit the particular application.

Given $|G\rangle$, a single step of the quantum walk algorithm can be done efficiently, i.e., in $O(\mathrm{poly}(n))$ gates per step. Each of the three qubit gates can be performed using a small number (typically 5 to 10) one and two qubit gates, see e.g., Ref. [9] for more details on construction of gates from more elementary quantum operations. The required number of three-qubit gates per step is thus $6n$ for the comparison of the values of (x, y, A_{xy}) down to one qubit, plus n for the conditional swap, plus another $6n$ to uncompute the ancillas, plus however many operations are required to toss the value of the y register. The total number of gates is thus $\gtrsim 14n$, which is linear in n, comfortably within the original prescription that "efficient" means $O(\mathrm{poly}(\log N))$. The state $|G\rangle$ is unchanged by each step of the quantum walk, but the state $|\psi\rangle$ is being entangled with it, so at the end of the walk when the position $|\psi(x)\rangle$ is measured, this destroys $|G\rangle$ as well.

Several points are notable about this implementation. First, this shows that both discrete and continuous formulations of quantum walks can treat any undirected graph (previously the discrete, coined quantum walks only applied to fixed degree graphs). Second, even before discussing whether the quantum walk itself gives advantages over classical algorithms, the storage involved to contain the description of the graph, $2n+1$ qubits, is an exponential improvement over $N(N-1)/2$ classical bits. Third, it is interesting that it turns out to depend on the graph being undirected for both the discrete and continuous time methods: there are parallels with Abelian and non-Abelian groups and the hidden subgroup problem that deserve further investigation. (See [10, 11] for some discussion of possibilities on directed graphs.)

I thank many people for useful and stimulating discussions of quantum walks, especially Peter Knight, Eugenio Roldan, John Sipe, Julia Kempe, John Watrous, Ed Farhi, Dorit Aharonov, Ben Tregenna, Will Flanagan, Rik Maile, Alex Russell, Cris Moore, Richard Cleve. This work was funded by the UK Engineering and Physical Sciences Research Council grant number GR/N2507701.

* Electronic address: Viv.Kendon@ic.ac.uk

[1] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman (2002), quant-ph/0209131.
[2] N. Shenvi, J. Kempe, and K. B. Whaley, Phys. Rev. A (2003), to appear, quant-ph/0210064.
[3] J. Kempe, Contemp. Phys. (2003), to appear., quant-ph/0303081.
[4] R. Blume-Kohout, C. M. Caves, and I. H. Deutsch, Found. Phys. (2002), to appear, quant-ph/0204157.
[5] B. C. Travaglione and G. J. Milburn, Phys. Rev. A **65**, 032310 (2002), quant-ph/0109076.
[6] B. Sanders, S. Bartlett, B. Tregenna, and P. L. Knight, Phys. Rev. A (2003), to appear, quant-ph/0207028.
[7] W. Dür, R. Raussendorf, H.-J. Briegel, and V. M. Kendon, Phys. Rev. A **66**, 05231 (2002), quant-ph/0207137.
[8] E. Farhi and S. Gutmann, Phys. Rev. A **58**, 915 (1998), quant-ph/9706062.
[9] M. A. Nielsen and I. J. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambs. UK, 2000).
[10] S. Severini, Siam J. Matrix Anal. Appl. (2002), to appear, math.CO/0205187.
[11] S. Severini (2002), math.CO/0303084.