Universal Computation by Quantum Walk

Andrew M. Childs
Department of Combinatorics and Optimization and Institute for Quantum Computing, University of Waterloo, 200 University Avenue West, Waterloo, Ontario, Canada N2L 3G1
(Received 11 June 2008; published 4 May 2009)

In some of the earliest work on quantum computing, Feynman showed how to implement universal quantum computation with a time-independent Hamiltonian. I show that this remains possible even if the Hamiltonian is restricted to be the adjacency matrix of a low-degree graph. Thus quantum walk can be regarded as a universal computational primitive, with any quantum computation encoded in some graph. The main idea is to implement quantum gates by scattering processes.

DOI: 10.1103/PhysRevLett.102.180501 PACS numbers: 03.67.Ac, 03.65.Nk, 05.40.Fb

While the first quantum algorithms were based on Fourier sampling [1], the concept of quantum walk [2,3]—a quantum mechanical analog of classical random walk—led to a new class of algorithms. By exploiting interference, quantum walks can outperform random walks at some computational tasks. For example, there is a black-box problem showing an exponential speedup of quantum walk over classical computation [4], and many quantum walk algorithms achieve polynomial speedup for problems of practical interest (e.g., [5–10]).

There are several ways to define quantum walk. Perhaps the simplest is the continuous-time quantum walk generated by the adjacency matrix A of a graph [2]. This walk takes place on a Hilbert space spanned by orthonormal basis states for each vertex of the graph, with the evolution for time t given by the unitary operator e^{-iAt}.

This Letter explores the power of quantum walk as a general model of computation. In this model, the walk takes place on an N-vertex graph for which a list of neighbors of any vertex can be computed efficiently, meaning in time poly($\log N$) (since the vertices of an N-vertex graph can be uniquely labeled using only $\lceil \log_2 N \rceil$ bits). The walk begins at one vertex, and a final vertex is measured after time $t = \text{poly}(\log N)$. I show that even a restricted version of this model, simple quantum walk on a sparse graph, is universal for quantum computation, meaning any problem that can be solved by a general-purpose quantum computer can also be solved by such a quantum walk (cf. adiabatic evolution, which is universal in a similar sense [11–14]).

This result shows that quantum walk is computationally powerful: in principle, any quantum algorithm can be recast as a quantum walk algorithm. More precisely, any m-gate quantum circuit can be simulated by a simple quantum walk on an N-vertex sparse graph, where $\log N = \text{poly}(m)$. Indeed, quantum walks and quantum circuits have essentially the same power, since a simple quantum walk on an N-vertex sparse graph can be simulated by a universal quantum computer using poly($\log N$) gates [4,15]. In particular, this means that deciding whether a quantum walk evolves from some vertex to another or not, with a $1/\text{poly}(\log N)$ gap between the probabilities of the two alternatives, is a BQP-complete promise problem. Not only does this give renewed motivation to search for quantum walk algorithms, but the specific construction may suggest new algorithmic approaches. Furthermore, it may provide tools for quantum complexity theory, as discussed further below.

In one of the earliest papers on quantum computation, Feynman constructed a Hamiltonian to implement any quantum circuit [16]. While the motivation was to give a physically reasonable description of a computing device, this can also be loosely interpreted as showing the universality of quantum walk as an abstract computational model, since the dynamics of any time-independent Hamiltonian can be viewed as a quantum walk on a weighted graph. However, this interpretation is clearest when the graph has bounded degree and its edges are unweighted, a setting we refer to as simple quantum walk on a sparse graph. In [16], the edges must have weights.

This Letter presents an alternative Hamiltonian for universal quantum computation. Here, the edges are unweighted, and the graph has maximum degree 3 (which cannot be improved). The main idea is to represent computational basis states by virtual quantum wires and to implement quantum gates by scattering off widgets attached to and connecting the wires. For an n-qubit circuit, we start with 2^n wires. Each wire is idealized as infinitely long, but can be well approximated with only poly(n) vertices. We begin at a single vertex at the far left of a quantum wire, and the output of the computation corresponds to a wire on the far right. The graph has exponentially many vertices—i.e., the Hilbert space has exponential dimension—but there is a simple rule determining the neighbors of any vertex, so the quantum walk could be efficiently simulated by a universal quantum computer. (We emphasize that vertices represent basis states, not physical objects such as qubits. Thus the construction does not directly give an architecture for a physical device, in contrast with the Feynman Hamiltonian or adiabatic quantum computing.)

We begin by briefly introducing scattering theory on graphs. First, consider an infinite line of vertices with corresponding computational basis states $|x\rangle$ for $x \in \mathbb{Z}$.

Vertex x is connected to vertices $x \pm 1$. The eigenstates of the adjacency matrix, parametrized by $k \in [-\pi, \pi)$, are the momentum states $|\tilde{k}\rangle$ with

$$\langle x|\tilde{k}\rangle = e^{ikx} \tag{1}$$

for all $x \in \mathbb{Z}$ (normalized so that $\langle\tilde{k}|\tilde{k}'\rangle = 2\pi\delta(k-k')$), with eigenvalues $2\cos k$.

Now let G be a finite graph and create an infinite graph with adjacency matrix H by attaching semi-infinite lines to M of its vertices. Label the basis states for vertices on the jth line as $|x, j\rangle$, with $x = 0$ at the vertex in the original graph and $x = 1, 2, \ldots$ along the line. On each line, an eigenstate of H must be a linear combination of states of the form (1) with momenta $\pm k$, with an eigenvalue $2\cos k$, or possibly of the same form but with $k = i\kappa$ or $k = i\kappa + \pi$ for some $\kappa > 0$, with an eigenvalue $2\cosh\kappa$ or $-2\cosh\kappa$, respectively. For each $j \in \{1, \ldots, M\}$ and each $k \in [-\pi, 0]$, there is an incoming scattering state of momentum k, denoted $|\tilde{k}, \mathrm{sc}_{\vec{j}}\rangle$, of the form

$$\langle x, j|\tilde{k}, \mathrm{sc}_{\vec{j}}\rangle = e^{-ikx} + R_j(k)e^{ikx} \tag{2}$$

$$\langle x, j'|\tilde{k}, \mathrm{sc}_{\vec{j}}\rangle = T_{j,j'}(k)e^{ikx}, \qquad j' \neq j \tag{3}$$

on the semi-infinite lines. (Since the off-diagonal elements of H are positive, whereas in the usual kinetic term $-d^2/dx^2$ they are negative, our incoming states have *negative* momentum.) The reflection coefficient $R_j(k)$, the transmission coefficients $T_{j,j'}(k)$, and the amplitudes on the vertices of G are determined by the condition $H|\tilde{k}, \mathrm{sc}_{\vec{j}}\rangle = 2\cos k|\tilde{k}, \mathrm{sc}_{\vec{j}}\rangle$. For any fixed k, these coefficients can be found by solving $|G|$ linear equations. Together with bound states $|\tilde{\kappa}, \mathrm{bd}^{\pm}\rangle$ (for discrete values of $\kappa > 0$ that can be obtained by solving $|G| - 1$ linear equations and one transcendental equation) of the form $\langle x, j|\tilde{\kappa}, \mathrm{bd}^{\pm}\rangle = B_j^{\pm}(\kappa)(\pm e^{-\kappa})^x$, the states $|\tilde{k}, \mathrm{sc}_{\vec{j}}\rangle$ form a complete set of eigenstates of H that are useful for the analysis of scattering off G.

To understand the dynamics, we expand in the basis of incoming scattering states and bound states. For evolution from vertex x on line j to vertex y on line $j' \neq j$,

$$\langle y, j'|e^{-iHt}|x, j\rangle = \int_{-\pi}^{0} e^{-2it\cos k}(T_{j,j'}e^{ik(x+y)} + T^*_{j',j}e^{-ik(x+y)})\frac{dk}{2\pi} + \sum_{\kappa,\pm} e^{\mp 2it\cosh\kappa}B_{j'}^{\pm}(\kappa)B_j^{\pm}(\kappa)^*(\pm e^{-\kappa})^{x+y}. \tag{4}$$

One can argue that the contribution from the bound states in (4) can be neglected. By the method of stationary phase, the integral over k is dominated by those values where the derivative of the phase of the integrand vanishes. The second term in the integrand can be shown to have no stationary points, and the phase of the first term is stationary for $x + y + \ell_{j,j'}(k) = v(k)t$, where

$$v(k) := \frac{d}{dk}2\cos k = -2\sin k \tag{5}$$

is the *group velocity* at momentum k and $\ell_{j,j'}(k) := \frac{d}{dk}\arg T_{j,j'}(k)$ is the *effective length* of the path through G from line j to line j'. For large $x + y$ [17],

$$|\langle y, j'|e^{-iHt}|x, j\rangle| \sim \frac{|T_{j,j'}(k^\star)|}{\sqrt{2\pi|c(k^\star)|}}, \tag{6}$$

where the phase is stationary at $k = k^\star$ and $c(k) := 2t\cos k + \frac{d^2}{dk^2}\arg T_{j,j'}(k)$.

While semi-infinite lines are convenient for analysis, they can be replaced by long but finite lines to give a finite graph [2]. The effect is small since the walk on a line has a maximum propagation speed: in (5), a maximum group velocity of 2 is obtained at $k = -\pi/2$.

We now show how to implement a universal set of quantum gates by scattering on graphs. We use the CNOT gate and two single-qubit gates that generate SU(2).

The CNOT gate is trivial to implement. This two-qubit gate swaps the basis states $|10\rangle$ and $|11\rangle$, leaving the other two states unchanged. Thus we simply exchange the appropriate wires using the widget shown in Fig. 1(a).

We can implement a phase gate by applying some phase to the $|1\rangle$ wire, leaving the $|0\rangle$ wire unchanged. To do this, we insert the widget shown in Fig. 1(b) on the $|1\rangle$ wire. Consider attaching semi-infinite lines to the terminals, and calculate the transmission coefficient for an incident wave of momentum k. The result is $T^{(b)}_{\mathrm{in,out}}(k) = 8/(8 + i\cos 2k\csc^3 k\sec k)$, whose magnitude squared is shown in Fig. 2. This widget has perfect transmission at $k = -\pi/4$, with $T^{(b)}(-\pi/4) = 1$ and $\ell^{(b)}(-\pi/4) = 1$. Relative to a single edge, this effectively introduces a phase of $e^{i\pi/4}$. Combining the widget on the $|1\rangle$ wire with a single edge for the $|0\rangle$ state, we obtain the phase gate

$$U_b := \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{7}$$

A widget that interacts two computational basis states is shown in Fig. 1(c), with its transmission probabilities shown in Fig. 2. At $k = -\pi/4$, the input amplitude is transformed into an equal superposition of output amplitudes, with none sent to the input channels. The effective

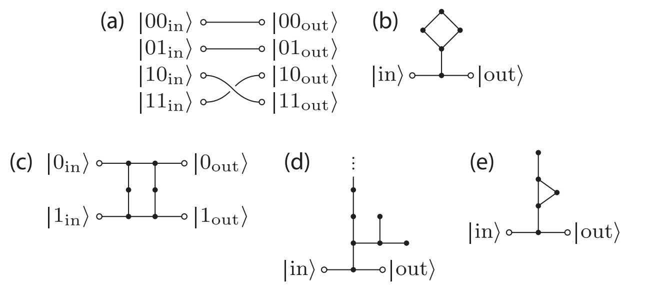

FIG. 1. Widgets used to construct a universal quantum computer. Open circles indicate vertices where previous or successive widgets can be attached. (a) CNOT gate. (b) Phase shift. (c) Basis-changing gate. (d) Momentum filter. (e) Momentum separator.

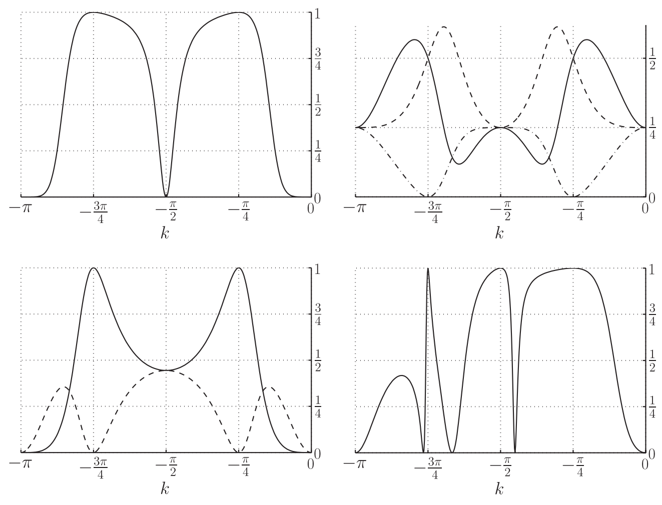

FIG. 2. Transmission probabilities. Top left: Phase shift widget (b). Top right: Basis-changing gate widget (c), with input at $|0_{\text{in}}\rangle$ and outputs at $|0_{\text{out}}\rangle$ (solid line), $|1_{\text{out}}\rangle$ (dashed line), and $|1_{\text{in}}\rangle$ (dot-dashed line). Bottom left: Filter widget (d), with input at $|\text{in}\rangle$ and outputs at $|\text{out}\rangle$ (solid line) and the semi-infinite line exiting upward (dashed line). Bottom right: Momentum separator widget (e).

lengths for forward transmission are both 2. Considering the phases of the transmission coefficients, the widget effectively performs the unitary transformation

$$U_c := -\frac{1}{\sqrt{2}}\begin{pmatrix} i & 1 \\ 1 & i \end{pmatrix}. \quad (8)$$

Since $iU_b^2 U_c U_b^2$ is the Hadamard gate, U_b and U_c generate SU(2) [18].

It is straightforward to embed these widgets in a graph representing a computation on n qubits. For a gate on j qubits, simply include its widget 2^{n-j} times, once for every setting of the $n-j$ qubits not acted on. For example, Fig. 3 shows a graph for a two-qubit circuit. Note that although the graph for an n-qubit circuit is exponentially large in n, it is sparse, and the neighbors of any vertex can be computed efficiently from the underlying circuit.

Using only the three gate widgets (a), (b), and (c), we can already construct a universal quantum computer, provided the input state is chosen appropriately. Since there is no reflection at $k = -\pi/4$, the transmission coefficients compose multiplicatively, so the concatenation of gate widgets can describe an arbitrary quantum circuit. If the input state is a wave packet consisting only of momenta close to $k = -\pi/4$, propagation through the widgets implements the circuit. But we can use a much simpler starting state, corresponding to one particular vertex of the graph.

To do this, we design a filter that only passes momenta near $k = -\pi/4$. The basic building block is shown in Fig. 1(d). Unlike the gate widgets, this widget includes a semi-infinite line, which allows undesired momentum components to be carried away.

For a single filter widget, all amplitude is transmitted forward at $k = -\pi/4, -3\pi/4$, whereas at other momenta, some amplitude exits upward and some is reflected (see Fig. 2). To filter out all but a narrow range of momenta, repeat the widget m_d times in series; this can be analyzed using a transfer matrix technique [2]. Except for k close to $-\pi/4$ or $-3\pi/4$, one eigenvalue of the transfer matrix is bounded above 1, so the transmission coefficient is exponentially small in m_d.

Unfortunately, the filter transmits undesired momenta near $k = -3\pi/4$ as well as the desirable momenta near $k = -\pi/4$. Although momentum components generically propagate at different speeds, these particular components have the same group velocity (5).

To isolate the two momenta, we use the widget shown in Fig. 1(e), whose transmission probability is pictured in Fig. 2. There is perfect transmission at $k = -\pi/4$, $-3\pi/4$. Furthermore, $\ell^{(e)}_{\text{in,out}}(-\pi/4) = 4(3 - 2\sqrt{2}) \approx 0.686$ and $\ell^{(e)}_{\text{in,out}}(-3\pi/4) = 4(3 + 2\sqrt{2}) \approx 23.3$. Since the effective length of the widget is different for these two momenta, it temporally separates them.

To simulate an arbitrary m-gate quantum circuit, we simply connect widgets for the gates. At $k = -\pi/4$, the transmission coefficients of this graph exactly implement the circuit. But a propagating wave packet comprises a range of momenta, so we must determine how close k should be to $-\pi/4$ to approximate the circuit. In turn, this determines how many filter widgets to include.

To understand widget composition, it helps to view the scattering problem in terms of 2^n input channels and 2^n output channels: define matrices $\mathcal{T}$, $\mathcal{R}$, $\bar{\mathcal{T}}$, $\bar{\mathcal{R}}$ as

$$\mathcal{T}_{j,j'} = T_{j_{\text{in}},j'_{\text{out}}} \qquad \mathcal{R}_{j,j'} = \begin{cases} R_{j_{\text{in}}} & j = j' \\ T_{j_{\text{in}},j'_{\text{in}}} & j \neq j' \end{cases} \quad (9)$$

$$\bar{\mathcal{T}}_{j,j'} = T_{j_{\text{out}},j'_{\text{in}}} \qquad \bar{\mathcal{R}}_{j,j'} = \begin{cases} R_{j_{\text{out}}} & j = j' \\ T_{j_{\text{out}},j'_{\text{out}}} & j \neq j' \end{cases} \quad (10)$$

for $j, j' \in \{0, \ldots, 2^n - 1\}$. The transmission and reflection matrices for two widgets in series are

$$\mathcal{T}_{12} = \mathcal{T}_1(1 - \mathcal{R}_2\bar{\mathcal{R}}_1)^{-1}\mathcal{T}_2 \quad (11)$$

$$\mathcal{R}_{12} = \mathcal{R}_1 + \mathcal{T}_1(1 - \mathcal{R}_2\bar{\mathcal{R}}_1)^{-1}\mathcal{R}_2\bar{\mathcal{T}}_1 \quad (12)$$

($\bar{\mathcal{T}}_{12}$ and $\bar{\mathcal{R}}_{12}$ have similar expressions). These formulas can be found by constructing scattering states for the composed widget using those for the components.

If two widgets each have little reflection, their composition also has little reflection. Suppose $\|\mathcal{R}_1\|, \|\bar{\mathcal{R}}_1\| \leq \delta_1$

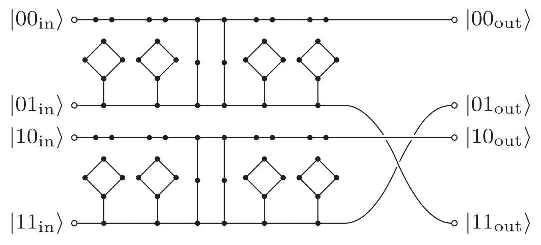
$|00_{\text{in}}\rangle$
$|01_{\text{in}}\rangle$
$|10_{\text{in}}\rangle$
$|11_{\text{in}}\rangle$
$|00_{\text{out}}\rangle$
$|01_{\text{out}}\rangle$
$|10_{\text{out}}\rangle$
$|11_{\text{out}}\rangle$

FIG. 3. Graph implementing a Hadamard gate on the second qubit followed by a CNOT gate controlled by the second qubit.

and $\|\mathcal{R}_2\|, \|\bar{\mathcal{R}}_2\| \leq \delta_2$; then $\|\mathcal{R}_{12}\| \leq \delta_1 + (1+\delta_1\delta_2)\delta_2$. Indeed, forward transmission is nearly described by the product of the two forward transmission matrices, since $\|\mathcal{T}_{12} - \mathcal{T}_1\mathcal{T}_2\| \leq \delta_1\delta_2(1+\delta_1\delta_2)$.

We apply these bounds to the transmission through m gate widgets for momenta near $k = -\pi/4$. Suppose $|k + \pi/4| = O(1/m^2)$; then $\|\mathcal{R}\| = O(1/m^2)$ for each widget. Thus the composition of all m widgets has $\|\mathcal{R}\| = O(1/m)$ for such momenta: the transmission is nearly perfect. To remove undesired momenta, precede the gate widgets by $m_d = \log\Theta(m^2)$ filter widgets. Then the filter output has exponentially small amplitude except for momentum components with $|k+\pi/4| = O(1/m^2)$ or $|k + 3\pi/4| = O(1/m^2)$.

Overall, an m-gate quantum circuit for an n-qubit unitary operation U can be simulated as follows. On input wire 0, place $m_d = \log\Theta(m^2)$ filter widgets followed by a momentum separation widget. Add widgets for the m gates in the circuit, each with 2^n inputs and 2^n outputs. Prepare the state $|x, 0_{\text{in}}\rangle$, where $x = \Theta(m^2)$ to justify the approximation (6). Run the walk for time $t \approx (x+\ell)/\sqrt{2} = O(m^2)$, where ℓ is the total effective length of all widgets and $\sqrt{2}$ is the group velocity. The 2^n input wires, 2^n output wires, and m_d filter wires can all be truncated at length greater than $2t$ without a significant effect. Finally, measure in the vertex basis. If the result is on some output wire $s \in \{0,1\}^n$ [which happens with probability $\Omega(1/m^2)$], output that s; otherwise discard the result and start over. The statistics of this simulation closely reproduce those of the original quantum circuit: by (6), the amplitude to propagate to vertex 0 on output line s is approximately $\langle s|U|0\rangle \times \Omega(1/m)$.

This construction shows that quantum walk is a universal computational primitive in the sense that any quantum circuit can be efficiently simulated by a simple quantum walk on a sparse graph. Ideas from this approach may be applicable elsewhere in quantum information processing, as follows.

One application is to quantum algorithms based on scattering. Following an early proposal along these lines [2], an algorithm for balanced binary game trees was given in [10]; this led to quantum algorithms for evaluating broad classes of formulas [19,20].

Another possible use is in quantum complexity theory. Feynman's Hamiltonian [16] was used to construct QMA- [21] and BQP-complete [22,23] problems, and the quantum walk construction might be applied to construct new complete problems with desired properties.

Finally, these ideas might inspire new quantum computer architectures. Of course, one cannot represent each vertex by a physical object; an efficient implementation must label the vertices by short bit strings and represent these with qubits. While we have not considered any such encoding, the underlying circuit provides some tensor product structure; this might permit encoding the system with a local Hamiltonian, giving a computer with no dynamic control.

I thank Shengyu Zhang for discussions that helped motivate this work, Edward Farhi and Jeffrey Goldstone for comments on a preliminary version, and Dorit Aharonov and Andrew Landahl for remarks on the degree of the graph in [16]. This work was done in part at the Caltech IQI, where the author received support from the NSF (Grant No. PHY-0456720) and the ARO (Grant No. W911NF-05-1-0294). This work was also supported in part by the ARO/DTO, MITACS, and NSERC.

[1] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).
[2] E. Farhi and S. Gutmann, Phys. Rev. A **58**, 915 (1998).
[3] J. Watrous, J. Comput. Syst. Sci. **62**, 376 (2001).
[4] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, in *Proceedings of the 35th ACM Symposium on Theory of Computing, San Diego, 2003* (ACM, New York, 2003), pp. 59–68.
[5] A. M. Childs and J. Goldstone, Phys. Rev. A **70**, 022314 (2004).
[6] A. Ambainis, SIAM J. Comput. **37**, 210 (2007).
[7] F. Magniez, M. Santha, and M. Szegedy, SIAM J. Comput. **37**, 413 (2007).
[8] H. Buhrman and R. Špalek, in *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms, Miami, 2006* (ACM, New York, 2006), pp. 880–889.
[9] F. Magniez and A. Nayak, Algorithmica **48**, 221 (2007).
[10] E. Farhi, J. Goldstone, and S. Gutmann, Theory Comput. **4**, 169 (2008).
[11] D. Aharonov, W. van Dam, J. Kempe, Z. Landau, S. Lloyd, and O. Regev, SIAM J. Comput. **37**, 166 (2007).
[12] J. Kempe and O. Regev, Quantum Inf. Comput. **3**, 258 (2003).
[13] J. Kempe, A. Kitaev, and O. Regev, SIAM J. Comput. **35**, 1070 (2006).
[14] D. Aharonov, D. Gottesman, S. Irani, and J. Kempe, in *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science, Providence, 2007* (IEEE, Washington, DC, 2007), pp. 373–383.
[15] D. Aharonov and A. Ta-Shma, in *Proceedings of the 35th ACM Symposium on Theory of Computing, San Diego, 2003* (Ref. [4]), pp. 20–29.
[16] R. P. Feynman, Opt. News **11**, 11 (1985).
[17] R. Wong, *Asymptotic Expansions of Integrals* (SIAM, Philadelphia, 2001).
[18] P. O. Boykin, T. Mor, M. Pulver, V. Roychowdhury, and F. Vatan, Inf. Proc. Lett. **75**, 101 (2000).
[19] A. Ambainis, A. M. Childs, B. W. Reichardt, R. Špalek, and S. Zhang, in *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science, Providence, 2007* (Ref. [14]), pp. 363–372.
[20] B. W. Reichardt and R. Špalek, in *Proceedings of the 40th ACM Symposium on Theory of Computing, Victoria, Canada, 2008* (ACM, New York, 2008), pp. 103–112.
[21] A. Y. Kitaev, A. H. Shen, and M. N. Vyalyi, *Classical and Quantum Computation* (AMS, Providence, 2002).
[22] P. Wocjan and S. Zhang, arXiv:quant-ph/0606179.
[23] D. Janzing and P. Wocjan, Theory Comput. **3**, 61 (2007).