Quantum random-walk search algorithm

Neil Shenvi,[1] Julia Kempe,[1,2,3] and K. Birgitta Whaley[1]
[1]*Department of Chemistry, University of California, Berkeley, California 94720*
[2]*Computer Science Division, EECS, University of California, Berkeley, California 94720*
[3]*CNRS-LRI, UMR 8623, Université de Paris-Sud, 91405 Orsay, France*
(Received 9 October 2002; published 23 May 2003)

Quantum random walks on graphs have been shown to display many interesting properties, including exponentially fast hitting times when compared with their classical counterparts. However, it is still unclear how to use these novel properties to gain an algorithmic speedup over classical algorithms. In this paper, we present a quantum search algorithm based on the quantum random-walk architecture that provides such a speedup. It will be shown that this algorithm performs an oracle search on a database of N items with $O(\sqrt{N})$ calls to the oracle, yielding a speedup similar to other quantum search algorithms. It appears that the quantum random-walk formulation has considerable flexibility, presenting interesting opportunities for development of other, possibly novel quantum algorithms.

DOI: 10.1103/PhysRevA.67.052307 PACS number(s): 03.67.Lx, 89.70.+c

I. INTRODUCTION

Recent studies of quantum random walks have suggested that they may display different behavior than their classical counterparts [1–5]. One of the promising features of these quantum random walks is that they provide an intuitive framework on which to build novel quantum algorithms. Since many classical algorithms can be formulated in terms of random walks, it is hoped that some of these may be translated into quantum algorithms which run faster than their classical counterparts. However, previous to a very recent paper by Childs *et al.* [6], there had been no quantum algorithms based on the random-walk model. In this paper we show that a quantum search algorithm can be derived from a certain kind of quantum random walk. Optimal quantum search algorithms are already well known [7–9]. The search algorithm from a quantum random walk we present here shows some differences from the established search algorithms and may possess useful properties with respect to robustness to noise and ease of physical implementation. It also provides a new direction for design of quantum algorithms from random walks, which may eventually lead to entirely new algorithms.

Current research uses two distinct models for quantum random walks, based on either discrete-time steps or on continuous-time evolution. Discrete time quantum random walks were introduced as a possible new tool for quantum algorithms generalizing discrete classical Markov chains [2]. The discrete-time walk can be thought of as a succession of unitary operations, each of which has a nonzero transition amplitude only between neighboring nodes of the graph. The relation of these to classical Markov chains provides considerable motivation for exploration of discrete random walks. Within the field of classical algorithms, the application of classical Markov chains in *classical* algorithms has been quite revolutionary, providing new approximation and optimization algorithms. By analogy, it might reasonably be hoped that similar algorithmic advances could be obtained for quantum algorithms from development of the quantum random walks. The second quantum random-walk model is the continuous-time quantum random walk, introduced in Refs. [4–6]. In the continuous-time walk, the adjacency matrix of the graph is used to construct a Hamiltonian which gives rise to a continuous-time evolution. This model differs from the discrete-time walk in that even for small times there is an (exponentially small) probability of transition to nonadjacent nodes. In this paper, we will consider the discrete-time model only.

The paper is organized as follows. Section II provides a brief introduction to discrete-time quantum random walks. Section III describes the random-walk search algorithm and provides a proof of its correctness. Section IV summarizes the similarities and differences between the random-walk search algorithm and Grover's search algorithm. Conclusions are presented in Sec. V.

Notation. Following standard computer science notation we will use the following to characterize the growth of certain functions: We will say $f(n)=O(g(n))$ if there are positive constants c and k such that $0\leqslant f(n)\leqslant cg(n)$ for $n\geqslant k$. Similarly $f(n)=\Omega(g(n))$ if $0\leqslant cg(n)\leqslant f(n)$ for constants $c,k\geqslant 0$ and $n\geqslant k$.

II. BACKGROUND

The discrete-time random walk can be described by the repeated application of a unitary evolution operator U. This operator acts on a Hilbert space $\mathcal{H}^C\otimes\mathcal{H}^S$, where $\mathcal{H}^C$ is the Hilbert space associated with a quantum coin (coin space) and $\mathcal{H}^S$ is the Hilbert space associated with the nodes of the graph. The operator U can be written as [2]

$$U=SC, \tag{1}$$

where S is a permutation matrix which performs a controlled shift based on the state of the coin space, and C is a unitary matrix which corresponds to ''flipping'' the quantum coin. We will call C the quantum coin. This operation can be visualized by analogy to a classical random walk. In each iteration of a discrete-time classical random walk on a graph, the coin is flipped. The walker then moves to an adjacent

node specified by the outcome of the coin flip. An equivalent process occurs in the quantum random walk, with the modification that the coin is a quantum coin, and can therefore exist in a superposition of states. This modification can lead to dramatic differences in behavior between the classical and quantum random walks. However, it should be noted that if the state of the coin is measured after each flip, then the quantum random walk reverts to a classical random walk (and similarly if the state of the nodes is measured after every step).

An important feature of the discrete-time quantum random walk that has significance for its use in the development of quantum algorithms is that by virtue of its definition on a quantum computer this walk will be efficiently implementable whenever its classical counterpart is efficiently implementable on a classical computer. [By efficient we mean that the walk can be simulated by a circuit with a number of gates that is polynomial in the number of bits (qubits).] This is due to the very similar structure of both these walks. To illustrate this, assume that we have an efficient way to implement the classical random walk on the underlying graph, i.e., to perform the coin flip and subsequent shift. The shift is conditional on the outcome of the coin flip (which determines the direction of the next step), i.e., we have a classical efficient circuit that performs a controlled shift on the basis states. It is straightforward [10] to translate this circuit into a quantum circuit that performs the unitary controlled shift of Eq. (1). Similarly, if there is an efficient procedure to flip the classical coin of the random walk, there will be an efficient way to implement a quantum coin. Hence implementation of the discrete-time random walk is automatically efficient if the underlying classical walk is efficiently implementable.

Note that if no measurement is made, the quantum walk is controlled by a unitary operator rather than a stochastic one. This implies that there is no limiting stationary distribution [2,11]. Nevertheless, several recent works have shown that consistent notions of mixing time can be formulated, and have shown polynomial speedup in these quantum mixing times relative to the classical analog [2,11]. Another quantity for which quantum walks have shown speedup relative to their classical analogs is the hitting time [12,13]. Under certain conditions this speedup can be exponential compared to the classical analog. We refer the reader to the recent papers [2,3,11], and [12] for some results obtained from discrete-time quantum random walks.

Our random-walk search algorithm will be based on a random walk on the n cube, i.e., the hypercube of dimension n [11,12]. The hypercube is a graph with $N=2^n$ nodes, each of which can be labeled by an n-bit binary string. Two nodes on the hypercube described by bitstrings $\vec{x}$ and $\vec{y}$ are connected by an edge if $|\vec{x}-\vec{y}|=1$, where $|\vec{x}|$ is the hamming weight of $\vec{x}$. In other words, if $\vec{x}$ and $\vec{y}$ differ by only a single-bit flip, then the two corresponding nodes on the graph are connected. Thus, each of the 2^n nodes on the n cube has degree n (i.e., it is connected to n other nodes), so the Hilbert space of the algorithm is $\mathcal{H}=\mathcal{H}^n\otimes\mathcal{H}^{2^n}$. Each state in $\mathcal{H}$ can be described by a bit string $\vec{x}$, which specifies the position on the hypercube, and a direction d, which specifies the state of the coin. The shift operator S maps a state $|d,\vec{x}\rangle$ onto the state $|d,\vec{x}\oplus\vec{e}_d\rangle$, where $\vec{e}_d$ is the dth basis vector on the hypercube. S can be written explicitly as

$$S=\sum_{d=0}^{n-1}\sum_{\vec{x}}|d,\vec{x}\oplus\vec{e}_d\rangle\langle d,\vec{x}|. \tag{2}$$

To completely specify the unitary evolution operator U, the coin operator C must also be chosen. Normally, the coin operator is chosen such that the same coin action is applied to each node on the graph. This is the case in previous studies of discrete quantum walks on the line [2,3,14] and on the hypercube [11,12]. In other words, the coin operator C can be written in a separable way as

$$C=C_0\otimes\mathcal{I}, \tag{3}$$

where C_0 is a $n\times n$ unitary operator acting on the coin space $\mathcal{H}^C$. In this case the action on the coin space $\mathcal{H}^C$ does not depend on the state of the node space $\mathcal{H}^S$. If C is separable according to Eq. (3) then the eigenstates of U are simply the tensor product of the eigenstates of an operator $C_{\vec{k}}$ on the coin space and of the Fourier modes of the hypercube (labeled by n-bit strings $\vec{k}$) [11]. One frequently chosen separable coin is Grover's ''diffusion'' operator on the coin space, given by

$$C_0=G=-\mathcal{I}+2|s^C\rangle\langle s^C|, \tag{4}$$

where $|s^C\rangle$ is the equal superposition over all n directions, i.e., $|s^C\rangle=1/\sqrt{n}\,\Sigma_{d=1}^{n}|d\rangle$ [11]. This coin operator is invariant to all permutations of the n directions, so it preserves the permutation symmetry of the hypercube. The use of the Grover diffusion operator as a coin for the hypercube was proposed in Ref. [11], where it was pointed out that this operator is the permutation invariant operator farthest away from the identity operator [11]. So, heuristically, it should provide the most efficient mixing over states, from any given initial state. The nontrivial eigenvalues and eigenvectors of U are given by [11]

$$e^{\pm i\omega_k}=1-\frac{2k}{n}\pm\frac{2i}{n}\sqrt{k(n-k)}, \tag{5}$$

$$|v_{\vec{k}}\rangle,|v_{\vec{k}}\rangle^*=\sum_{\vec{x},d}(-1)^{\vec{k}\cdot\vec{x}}\frac{2^{-n/2}}{\sqrt{2}}|d,\vec{x}\rangle \times\begin{cases}1/\sqrt{k} & \text{if } k_d=1\\ \mp i/\sqrt{n-k} & \text{if } k_d=0.\end{cases} \tag{6}$$

Note that the equal superposition over all states, $|\psi_0\rangle=|s^C\rangle\otimes|s^S\rangle$, where $|s^S\rangle$ is the equal superposition over the 2^n nodes, is an eigenvector of U with eigenvalue 1. So repeated application of U leaves the state $|\psi_0\rangle$ unchanged.

In order to create a search algorithm using the quantum random-walk architecture, we now consider a small perturbation of the unitary operator U. In the standard setting of a search algorithm we have an oracle O_f, which ''marks'' a single bitstring $\vec{x}_{target}$. More specifically the oracle com-

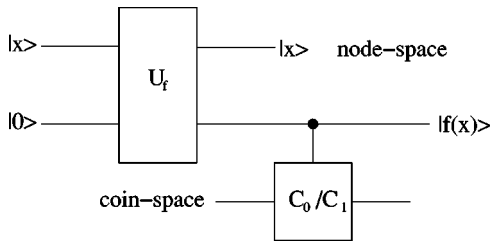

FIG. 1. Modified standard oracle that simulates the coin oracle. The standard oracle acts on the node space as $U_f:|\vec{x}\rangle\otimes|y\rangle\rightarrow|\vec{x}\rangle\otimes|y\oplus f(x)\rangle$. The controlled coin operation, denoted by C_0/C_1, applies C_0 on the coin space if the control qubit is in the state $|0\rangle$, and C_1 if it is in the state $|1\rangle$.

putes a function f such that $f(\vec{x}_{target})=1$ and $f(\vec{x})=0$ if $\vec{x}\neq\vec{x}_{target}$. The query complexity of a search algorithm is defined to be the number of queries to O_f that need to be made to find the marked string $\vec{x}_{target}$ with high probability. In the quantum case the oracle U_f is implemented via a reversible unitary operation; in the standard setting the oracle shifts the phase of the marked item. For our implementation the coin operator will take the function of the oracle. Specifically, we consider ''marking'' a single arbitrary node by applying a special coin action to that node. The oracle will act by applying a ''marking coin'' C_1 to the marked node and the original coin C_0 to the unmarked nodes, i.e., the coin action will be conditioned on the node. (Note that this modified coin is still unitary.)

This ''coin oracle'' can be easily obtained from the standard oracle of quantum search. To simulate the coin oracle we setup the standard oracle on the node space, and then add a conditional C_0 or C_1 operation, respectively, at the output. This is illustrated in Fig. 1.

Without loss of generality, we can assume that the marked node corresponds to the all-zero string $\vec{x}_{target}=\vec{0}$. Then our coin operator becomes

$$C'=C_0\otimes\mathcal{I}+(C_1-C_0)\otimes|\vec{0}\rangle\langle\vec{0}|. \tag{7}$$

The marking coin C_1 can be any $n\times n$ unitary matrix. For simplicity, we will consider here the case where $C_1=-\mathcal{I}$. Numerical studies [15] have shown that other choices for the coin C_1 yield similar results. As seen from Eq. (7) the coin operator is now a composite unitary and its action is conditioned on the node register. Our perturbed unitary evolution operator U' is given by

$$\begin{aligned} U' &= SC' \\ &= S(G\otimes\mathcal{I}-(G+\mathcal{I})\otimes|\vec{0}\rangle\langle\vec{0}|) \\ &= U-2S(|s^C\rangle\langle s^C|\otimes|\vec{0}\rangle\langle\vec{0}|). \end{aligned} \tag{8}$$

Analysis of the effects of this perturbation leads directly to the definition of the random-walk search algorithm, as is described in the following section.

III. RANDOM WALK SEARCH ALGORITHM

A. Overview of the algorithm

We define the search space of the algorithm to be the set of all n-bit binary strings, $\vec{x}=\{0,1\}^n$. We consider the function $f(\vec{x})=\{0,1\}$, such that $f(\vec{x})=1$ for exactly one input $\vec{x}_{target}$. Our goal is to find $\vec{x}_{target}$. Using the mapping of n-bit binary string to nodes on the hypercube, this search problem is then equivalent to searching for a single marked node amongst the $N=2^n$ nodes on the n cube. For purposes of the proof, we have set the marked node to be $\vec{x}_{target}=\vec{0}$, but the location of the marked node has no significance.

The random-walk search algorithm is implemented as follows.

(1) Initialize the quantum computer to the equal superposition over all states, $|\psi_0\rangle=|s^C\rangle\otimes|s^S\rangle$. This can be accomplished efficiently on the node space by applying n single-bit Hadamard operations to the $|\vec{0}\rangle$ state. A similar procedure works for the direction space.

(2) Given a coin oracle C' which applies the coin $C_0=G$ to the unmarked states and the coin $C_1=-\mathcal{I}$ to the marked state, apply the perturbed evolution operator, $U'=SC'$, $t_f=\pi/2\sqrt{2^n}$ times.

(3) Measure the state of the computer in the $|d,\vec{x}\rangle$ basis.

It is our claim that with probability $\frac{1}{2}-O(1/n)$, the outcome of the measurement will be the marked state. By repeating the algorithm a constant number of times, we can determine the marked state with an arbitrarily small degree of error. In the remainder of this section we provide a proof of this algorithm.

The general outline of the proof that we will present is the following. We need to determine the result of the operation $(U')^t$ on the initial state $|\psi_0\rangle$. To do this, we will first simplify the problem by showing that the perturbed walk on the hypercube can be collapsed to a walk on the line (Theorem 1). Next, by constructing two approximate eigenvectors of U', $|\psi_0\rangle$ and $|\psi_1\rangle$, we will show that there are exactly two eigenvalues of U' that are relevant [i.e., the initial state $|\psi_0\rangle$ has high overlap with the space spanned by the corresponding eigenvectors (see Theorem 2 and Theorem 3)]. We denote these eigenvalues by $e^{i\omega_0'}$ and $e^{-i\omega_0'}$. We will then show that the corresponding eigenvectors $|\omega_0'\rangle$ and $|-\omega_0'\rangle$ can be well approximated by linear combinations of the initial state $|\psi_0\rangle$ and the second state $|\psi_1\rangle$ (Theorem 4). As a result, our random walk search algorithm can be approximated by a two-dimensional rotation in the $|\omega_0'\rangle,|-\omega_0'\rangle$ plane away from the initial state $|\psi_0\rangle\approx 1/\sqrt{2}(|\omega_0'\rangle+|-\omega_0'\rangle)$ and towards $|\psi_1\rangle\approx i/\sqrt{2}(-|\omega_0'\rangle+|-\omega_0'\rangle)$, which constitutes a very close approximation to the target state $|\vec{x}_{target}\rangle$. Finally, we show that each application of the evolution operator U' corresponds to a rotation angle of approximately $1/\sqrt{2^{n-1}}$ Theorem (5). Hence, the search is completed after approximately $(\pi/2)\sqrt{2^{n-1}}$ steps, i.e., after $O(\sqrt{N})$ calls to the oracle, where $N=2^n$ is the number of nodes.

B. Proof of correctness

In general, analytic determination of the eigenspectrum of a large matrix is a daunting task, so we will take advantage of the symmetries inherent in U' to simplify the problem. Let us first show that the perturbed random walk on the hypercube can be collapsed onto a random walk on the line. Let P_{ij} be the permutation operator which swaps the bits i and j, in both the node space and the coin space. In other words, given a state $|d,\vec{x}\rangle$, under the permutation operator, P_{ij}, the ith and jth bits of $\vec{x}$ are swapped and the directions $d=i$ and $d=j$ are swapped. Clearly, the unperturbed evolution operator U commutes with P_{ij} since every direction in the unperturbed walk is equivalent.

Theorem 1. U' commutes with P_{ij}, i.e., the perturbed walk on the hypercube can be effectively regarded as a walk on the line.

Proof.

$$\begin{aligned} P_{ij}^{\dagger}U'P_{ij} &= P_{ij}^{\dagger}UP_{ij} - 2P_{ij}^{\dagger}S\cdot(|s^C\rangle\langle s^C|\otimes|\vec{0}\rangle\langle\vec{0}|)P_{ij} \\ &= U - \frac{2}{\sqrt{n}}\sum_{d=0}^{n-1} P_{ij}^{\dagger}|d,\vec{e_d}\rangle\langle d,0|P_{ij} \\ &= U'. \end{aligned} \tag{9}$$

So, $[U',P_{ij}]=0$. ■

Because the initial state $|\psi_0\rangle$ is an eigenvector of P_{ij} with eigenvalue 1 for all i and j, and $[U',P_{ij}]=0$, any intermediate state $|\psi_t\rangle=(U')^t|\psi_0\rangle$ must also be an eigenvector of eigenvalue 1 with respect to P_{ij}. Thus, $(U')^t$ preserves the symmetry of $|\psi_0\rangle$ with respect to bit swaps. It is therefore useful to define $2n$ basis states, $|R,0\rangle,|L,1\rangle,|R,1\rangle,\ldots,|R,n-1\rangle,|L\cdot n\rangle$, where

$$|R,x\rangle = \sqrt{\frac{1}{(n-x)\binom{n}{x}}}\sum_{|\vec{x}|=x}\sum_{x_d=0}|d,\vec{x}\rangle, \tag{10}$$

$$|L,x\rangle = \sqrt{\frac{1}{x\binom{n}{x}}}\sum_{|\vec{x}|=x}\sum_{x_d=1}|d,\vec{x}\rangle, \tag{11}$$

which are also invariant to bit swaps P_{ij}. These states span the eigenspace of eigenvalue 1 of P_{ij}. Using these basis states, we can project out all but one spatial degree of freedom and effectively reduce the random walk on the hypercube to a random walk on the line. This is illustrated in Fig. 2. The marked node corresponds now to $|R,0\rangle$. We can rewrite U, U', and $|\psi_0\rangle$ in this collapsed basis. First note that the shift operator S in this basis acts as

$$S=\sum_{x=0}^{n-1}|R,x\rangle\langle L,x+1|+|L,x+1\rangle\langle R,x| \tag{12}$$

and the unperturbed coin acts as

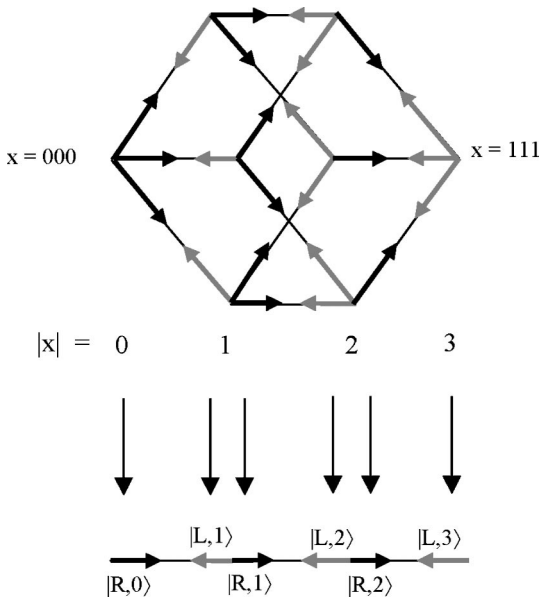

FIG. 2. Collapsing a random walk on the hypercube to a random walk on the line. The states on the hypercube are mapped to the state on the line based on their Hamming weight and the direction in which they point (see text).

$$C_0=\sum_{x=0}^{n}\begin{pmatrix}\cos\omega_x & \sin\omega_x\\ \sin\omega_x & -\cos\omega_x\end{pmatrix}\otimes|x\rangle\langle x|, \tag{13}$$

where $\cos\omega_x=1-2x/n$ and $\sin\omega_x=(2/n)\sqrt{x(n-x)}$ and where the first part acts on the space spanned by $\{|R\rangle,|L\rangle\}$ and the second part acts on the positions $\{|0\rangle,\ldots,|n\rangle\}$ on the line. Note that the coin of the collapsed walk is not homogeneous in space any more. The unitary operator U on the restricted space acts as

$$\begin{aligned} U=&\sum_{x=0}^{n-1}|R,x\rangle(-\cos\omega_{x+1}\langle L,x+1|+\sin\omega_{x+1}\langle R,x+1|) \\ &+\sum_{x=1}^{n}|L,x\rangle(\sin\omega_{x-1}\langle L,x-1|+\cos\omega_{x-1}\langle R,x-1|). \end{aligned} \tag{14}$$

Similarly,

$$U'=U+\Delta U=U-2|L,1\rangle\langle R,0|. \tag{15}$$

Note that the only difference between U and U' is in the sign of the matrix element in position $(|L,1\rangle,|R,0\rangle)$. Finally,

$$\begin{aligned} |\psi_0\rangle=\frac{1}{\sqrt{2^n}}|R,0\rangle+\frac{1}{\sqrt{2^n}}|L,n\rangle+\sum_{x=1}^{n-1}\Bigg(&\sqrt{\frac{\binom{n-1}{x-1}}{2^n}}|L,x\rangle \\ &+\sqrt{\frac{\binom{n-1}{x}}{2^n}}|R,x\rangle\Bigg). \end{aligned} \tag{16}$$

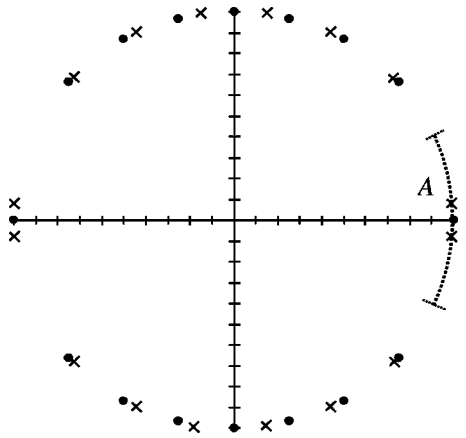

FIG. 3. The results of numerical spectral analysis of U and U' for $n=8$. The circles indicate eigenvalues of U. The crosses indicate eigenvalues of U'.

Since U and P_{ij} are mutually diagonalizable, the eigenvectors of U in the reduced space are also bit-flip invariant. Examining Eq. (5), it is clear that if we take the equal superpositions of all eigenvectors of same eigenvalue $|v_{\vec{k}}\rangle$ such that $|\vec{k}|=k$, the resulting eigenvector will be bit-swap invariant. Thus we define

$$|\omega_k\rangle=\frac{1}{\sqrt{\binom{n}{k}}}\sum_{|\vec{k}|=k}|v_{\vec{k}}\rangle, \tag{17}$$

which are the eigenvectors of U with eigenvalues $e^{i\omega_k}$ in the collapsed (symmetric) space.

Note that both U and U' are represented by real matrices; therefore, their eigenvalues and eigenvectors will come in complex-conjugate pairs.

Having determined these general properties of the perturbed matrix U', we now turn to the problem of analyzing the eigenvalue spectrum of U'. Let $\mathcal{A}$ be the arc on the unit circle containing all complex numbers of unit norm with real part greater than $1-2/(3n)$. In other words,

$$\mathcal{A}=\left\{z:\mathrm{Re}\,z>1-\frac{2}{3n},|z|=1\right\}. \tag{18}$$

Figure 3 shows the geometrical representation of $\mathcal{A}$ together with the eigenvalue spectra of the unperturbed and perturbed matrices for $n=8$. We will prove that $\mathcal{A}$ contains exactly two eigenvalues $e^{i\omega_0'}$ and $e^{-i\omega_0'}$ of U'. First, we will prove that there are *at most* two eigenvalues with real part greater than $1-2/(3n)$. Then we will show that there are *at least* two eigenvalues on $\mathcal{A}$. From these facts, it follows that there are exactly two eigenvalues of U' on $\mathcal{A}$.

Theorem 2. There are at most two eigenvalues of U' with real part greater than $1-2/(3n)$.

Proof. We will prove by contradiction. Let us assume that there are three eigenvalues, $e^{i\omega_0'}$, $e^{i\omega_1'}$, and $e^{i\omega_2'}$, with real part greater than $1-2/(3n)$. Let $|\omega_0'\rangle$, $|\omega_1'\rangle$, and $|\omega_2'\rangle$ be the corresponding eigenvectors. Then,

$$\mathrm{Re}\left(\sum_i\langle\omega_i'|U'|\omega_i'\rangle\right)=\mathrm{Re}\left(\sum_i e^{i\omega_i'}\langle\omega_i'|\omega_i'\rangle\right)>3-2/n. \tag{19}$$

Let us define Ω to be the subspace spanned by $|\omega_0'\rangle$, $|\omega_1'\rangle$, $|\omega_2'\rangle$. Then we can write Eq. (19) as the partial trace of U' over Ω,

$$\mathrm{Re}\,\mathrm{Tr}_\Omega U'>3-2/n. \tag{20}$$

Let us now define $|\psi_-\rangle=1/\sqrt{2}(|0,R\rangle-|1,L\rangle)$. We can expand the $|\omega_0'\rangle$, $|\omega_1'\rangle$, and $|\omega_2'\rangle$ in terms of $|\psi_0\rangle$, $|\psi_-\rangle$, and a residual vector,

$$\begin{aligned}|\omega_0'\rangle&=c_{00}'|\psi_0\rangle+c_{01}'|\psi_-\rangle+c_{02}'|r_0'\rangle,\\ |\omega_1'\rangle&=c_{10}'|\psi_0\rangle+c_{11}'|\psi_-\rangle+c_{11}'|r_1'\rangle,\\ |\omega_2'\rangle&=c_{20}'|\psi_0\rangle+c_{21}'|\psi_-\rangle+c_{22}'|r_2'\rangle,\end{aligned} \tag{21}$$

where $|r_i'\rangle$ is a normalized vector orthogonal to $|\psi_0\rangle$ and $|\psi_-\rangle$. We now observe that, due to the basis invariance of the trace, Eq. (20) holds for any linear combination of $|\omega_0'\rangle$, $|\omega_1'\rangle$, and $|\omega_2'\rangle$. Thus, we can construct three new orthonormal vectors, $|\alpha_0\rangle$, $|\alpha_1\rangle$, and $|\alpha_2\rangle$ by taking linear combinations of $|\omega_0'\rangle$, $|\omega_1'\rangle$, and $|\omega_2'\rangle$, such that

$$\langle\alpha_2|\psi_0\rangle=\langle\alpha_2|\psi_-\rangle=0. \tag{22}$$

In other words, we can expand $|\alpha_0\rangle$, $|\alpha_1\rangle$, and $|\alpha_2\rangle$ as

$$\begin{aligned}|\alpha_0\rangle&=c_{00}|\psi_0\rangle+c_{01}|\psi_-\rangle+c_{02}|r_0\rangle,\\ |\alpha_1\rangle&=c_{10}|\psi_0\rangle+c_{11}|\psi_-\rangle+c_{12}|r_1\rangle,\\ |\alpha_2\rangle&=|r_2\rangle.\end{aligned} \tag{23}$$

Since $|\alpha_0\rangle$, $|\alpha_1\rangle$, and $|\alpha_2\rangle$ still form a basis for Ω, from Eq. (20) it follows that

$$3-2/n<\mathrm{Re}\sum_i\langle\alpha_i|U'|\alpha_i\rangle. \tag{24}$$

Since U' is a unitary operator, we know that $\mathrm{Re}\langle\alpha_i|U'|\alpha_i\rangle\leqslant 1$ for all $|\alpha_i\rangle$. Thus, applying this inequality to the first two terms in the sum, we obtain

$$\mathrm{Re}\sum_i\langle\alpha_i|U'|\alpha_i\rangle\leqslant 2+\mathrm{Re}\langle\alpha_2|U'|\alpha_2\rangle. \tag{25}$$

Since, $U'=U+\Delta U$, we can write

$$\mathrm{Re}\langle\alpha_2|U'|\alpha_2\rangle=\mathrm{Re}\langle\alpha_2|U|\alpha_2\rangle+\mathrm{Re}\langle\alpha_2|\Delta U|\alpha_2\rangle. \tag{26}$$

Let us first consider $\langle\alpha_2|U|\alpha_2\rangle$. We can expand $|\alpha_2\rangle$ in terms of the unperturbed eigenstates, $|\alpha_2\rangle=\Sigma_j b_j|\omega_j\rangle$. So, $\mathrm{Re}\langle\alpha_2|U|\alpha_2\rangle=\Sigma_j|b_j|^2\cos\omega_j$. However, since $\langle\alpha_2|\psi_0\rangle=0$, there is no contribution from the eigenvalue with value 1. The eigenvalue with the next-largest real part is $e^{i\omega_1}=1-2/n+i(2/n)\sqrt{n-1}$. Thus,

$$\mathrm{Re}\langle\alpha_2|U|\alpha_2\rangle \leqslant 1-2/n. \tag{27}$$

Next, we consider $\langle\alpha_2|\Delta U|\alpha_2\rangle$. Let $|\psi_+\rangle = 1/\sqrt{2}(|0,R\rangle + |1,L\rangle)$. Using Eq. (15) we can express ΔU in terms of $|\psi_-\rangle$ and $|\psi_+\rangle$,

$$\Delta U = |\psi_-\rangle\langle\psi_-| + |\psi_-\rangle\langle\psi_+| - |\psi_+\rangle\langle\psi_-| - |\psi_+\rangle\langle\psi_+|. \tag{28}$$

But since $\langle\alpha_2|\psi_-\rangle = 0$ [see Eq. (22)],

$$\langle\alpha_2|\Delta U|\alpha_2\rangle = (-|\langle\psi_+|\alpha_2\rangle|^2) \leqslant 0. \tag{29}$$

Then, $\mathrm{Re}\langle\alpha_2|U'|\alpha_2\rangle \leqslant 1-2/n$. Combining Eqs. (25), (27), and (29), we obtain

$$\mathrm{Re}\sum_i \langle\alpha_i|U'|\alpha_i\rangle \leqslant 3-2/n. \tag{30}$$

Since this contradicts Eq. (24), our assumption must be false. ■

Theorem 3. There are at least two eigenvalues of U' on $\mathcal{A}$.

Proof. We will construct two approximate eigenvectors of U', $|\psi_0\rangle$ and $|\psi_1\rangle$. $|\psi_0\rangle$ is given by Eq. (16). Using Eq. (15),

$$U'|\psi_0\rangle = |\psi_0\rangle - 2/\sqrt{2^n}|L,1\rangle, \tag{31}$$

and

$$\begin{aligned}\langle\psi_0|U'|\psi_0\rangle &= \langle\psi_0|\psi_0\rangle - \langle\psi_0|L,1\rangle\langle R,0|\psi_0\rangle \\ &= 1-1/2^{n-1}.\end{aligned} \tag{32}$$

So, apart from a small residual, $|\psi_0\rangle$ is also "almost" an eigenvector of U' with eigenvalue 1. Now, we need to find a second approximate eigenvector $|\psi_1\rangle$. Let

$$|\psi_1\rangle = \left(\sum_{x=0}^{n/2-1} \frac{1}{\sqrt{2\binom{n-1}{x}}} |R,x\rangle - \frac{1}{\sqrt{2\binom{n-1}{x}}} |L,x+1\rangle \right) \Big/ c, \tag{33}$$

where c is a normalization constant,

$$c = \sqrt{\sum_{x=0}^{n/2-1} \frac{1}{\binom{n-1}{x}}}. \tag{34}$$

Using this definition and Eq. (14), we see that

$$U'|\psi_1\rangle = |\psi_1\rangle - \frac{1}{c\sqrt{2\binom{n-1}{n/2}}}(|R,n/2-1\rangle + |L,n/2+1\rangle). \tag{35}$$

Hence,

$$\langle\psi_1|U'|\psi_1\rangle = 1 - \frac{1}{2c^2\binom{n-1}{n/2}}. \tag{36}$$

Expanding Eq. (34) we find $1 < c^2 < 1+2/n$ for sufficiently large n. Thus, except for a small residual, $|\psi_1\rangle$ is "almost" an eigenvector of U' with eigenvalue 1.

Now let us verify that there is at least one eigenvalue of U' on $\mathcal{A}$. Let us assume that there are no eigenvalues of U' on $\mathcal{A}$. Then $\cos\omega_j' < 1-2/(3n)$ for all j. Then using Eq. (32),

$$\begin{aligned}1-1/2^{n-1} &= \mathrm{Re}\langle\psi_0|U'|\psi_0\rangle \\ &= \sum_j |\langle\psi_0|\omega_j'\rangle|^2 \cos\omega_j' \\ &< (1-2/(3n))\sum_j |\langle\psi_0|\omega_j'\rangle|^2 \\ &= 1-2/(3n),\end{aligned} \tag{37}$$

which is wrong for $n>3$. Hence our assumption is false and there must be at least one eigenvalue of U' on $\mathcal{A}$.

Now let us assume that there is exactly one eigenvalue of U', $e^{i\omega_0'}$, on $\mathcal{A}$. Then,

$$\begin{aligned}1-\frac{1}{2^{n-1}} &= \mathrm{Re}\langle\psi_0|U'|\psi_0\rangle \\ &= \sum_j |\langle\psi_0|\omega_j'\rangle|^2 \cos\omega_j' \\ &= |\langle\psi_0|\omega_0'\rangle|^2\cos\omega_0' + \sum_{j\neq 0} |\langle\psi_0|\omega_j'\rangle|^2\cos\omega_j' \\ &\leqslant |\langle\psi_0|\omega_0'\rangle|^2 + (1-|\langle\psi_0|\omega_0'\rangle|^2)(1-2/(3n)).\end{aligned} \tag{38}$$

Rearranging terms,

$$|\langle\psi_0|\omega_0'\rangle|^2 \geqslant 1-\frac{3n}{2^n}. \tag{39}$$

If we use $|\psi_1\rangle$ as a trial vector and follow the same arguments, we obtain the inequality

$$|\langle\psi_1|\omega_0'\rangle|^2 \geqslant 1-\frac{3n}{4c^2\binom{n-1}{n/2}}. \tag{40}$$

But since $|\psi_0\rangle$ and $|\psi_1\rangle$ are orthonormal, this leads to a contradiction, since,

$$1=\langle\omega_0'|\omega_0'\rangle\geqslant|\langle\psi_0|\omega_0'\rangle|^2+|\langle\psi_1|\omega_0'\rangle|^2$$
$$\geqslant 2-\frac{3n}{2^n}-\frac{3n}{4c^2\binom{n-1}{n/2}}, \tag{41}$$

which is not true for large n. Hence, there must be at least two eigenvalues on the arc $\mathcal{A}$. ■

As noted above, the eigenvalues and eigenvectors of U' come in complex-conjugate pairs. In particular, the two eigenvalues on $\mathcal{A}$ must be a complex-conjugate pair; let $e^{\pm i\omega_0'}$ be the two eigenvalues on $\mathcal{A}$. The corresponding eigenvectors obey $|-\omega_0'\rangle=|\omega_0'\rangle^*$ (if $e^{i\omega_0'}=e^{-i\omega_0'}=1$, then we can construct linear combinations of $|\omega_0'\rangle$ and $|-\omega_0'\rangle$ for which this statement is true). We will now show that $|\pm\omega_0'\rangle$ can be well approximated by linear combinations of $|\psi_0\rangle$ and $|\psi_1\rangle$.

Theorem 4. The two eigenvectors with eigenvalues close to 1 can be well approximated by linear combinations of the initial state $|\psi_0\rangle$ and the state $|\psi_1\rangle$, as $|\pm\omega_0'\rangle\approx 1/\sqrt{2}(|\psi_0\rangle \pm i|\psi_1\rangle)$. More precisely,

$$|\omega_0'\rangle=\sqrt{p_0}|\psi_0\rangle+\sqrt{p_1}e^{i\eta}|\psi_1\rangle+\sqrt{1-p_0-p_1}|r_0\rangle,$$
$$|-\omega_0'\rangle=\sqrt{p_0}|\psi_0\rangle+\sqrt{p_1}e^{-i\eta}|\psi_1\rangle+\sqrt{1-p_0-p_1}|r_0\rangle^*, \tag{42}$$

where $p_0=|\langle\omega_0'|\psi_0\rangle|^2=|\langle-\omega_0'|\psi_0\rangle|^2$, $p_1=|\langle\tilde{\omega}'_0|\psi_1\rangle|^2=|\langle-\tilde{\omega}_0'|\psi_1\rangle|^2$, and $|r_0\rangle$ is a normalized vector orthogonal to $|\psi_0\rangle$ and $|\psi_1\rangle$. Furthermore, $1/2\geqslant p_0\geqslant 1/2-3n/2^{n+1}$ and

$$1/2\geqslant p_1\geqslant 1/2-\frac{3n}{8c^2\binom{n-1}{n/2}},$$

with $e^{i\eta}=i+\Delta$, where

$$|\Delta|=O\left(\frac{n}{\binom{n-1}{n/2}}\right).$$

Proof. Since $|\psi_0\rangle$ and $|\psi_1\rangle$ are real vectors, $|\langle\omega_0'|\psi_0\rangle|^2=|\langle-\omega_0'|\psi_0\rangle|^2\leqslant 1/2$ and $|\langle\omega_0'|\psi_1\rangle|^2=|\langle-\omega_0'|\psi_1\rangle|^2\leqslant 1/2$. Using Eq. (32),

$$1-\frac{1}{2^{n-1}}=\mathrm{Re}\langle\psi_0|U'|\psi_0\rangle$$
$$=\sum_j\cos\omega_j'|\langle\omega_j'|\psi_0\rangle^2|$$
$$=2p_0\cos\omega_0'+\sum_{j\neq 0}\cos\omega_j'|\langle\omega_j'|\psi_0\rangle|^2$$
$$<2p_0+(1-2/(3n))(1-2p_0). \tag{43}$$

Rearranging terms, we obtain

$$p_0\geqslant 1/2-\frac{3n}{2^{n+1}}. \tag{44}$$

Using Eq. (36) and the same arguments as above we obtain

$$p_1\geqslant 1/2-\frac{3n}{8c^2\binom{n-1}{n/2}}. \tag{45}$$

Up to a global phase $|\omega_0'\rangle$ can be written as

$$|\omega_0'\rangle=|\langle\omega_0'|\psi_0\rangle||\psi_0\rangle+|\langle\omega_0'|\psi_1\rangle|e^{i\eta}|\psi_1\rangle+\sqrt{1-p_0-p_1}|r_0\rangle, \tag{46}$$

which yields Eq. (42) for $|\omega_0'\rangle$ and $|-\omega_0'\rangle$.

To estimate $e^{i\eta}$ note that since $|\omega_0'\rangle$ and $|-\omega_0'\rangle$ are eigenvectors of a unitary matrix, they must be orthogonal. Consequently,

$$0=\langle-\omega_0'|\omega_0'\rangle=p_0+p_1(e^{i\eta})^2+(1-p_0-p_1)\langle r_0^*|r_0\rangle. \tag{47}$$

Solving for $e^{i\eta}$, we obtain

$$\mathrm{Re}(e^{i\eta})^2=\frac{-p_0-(1-p_0-p_1)\mathrm{Re}\langle r_0^*|r_0\rangle}{p_1}. \tag{48}$$

Assume $\mathrm{Re}\langle r_0^*|r_0\rangle\geqslant 0$. Then, using $1/(1-x)\leqslant 1+2x$ for small x, we get

$$-1+\frac{3n}{2^n}\geqslant-\frac{p_0}{p_1}$$
$$\geqslant\mathrm{Re}(e^{i\eta})^2$$
$$\geqslant-\frac{p_0+\left(\frac{3n}{2^{n+1}}+\frac{3n}{8c^2\binom{n-1}{n/2}}\right)}{p_1}$$
$$\geqslant-1-2\left(\frac{3n}{2^{n+1}}+\frac{3n}{8c^2\binom{n-1}{n/2}}\right)$$
$$-4\left(\frac{3n}{8c^2\binom{n-1}{n/2}}\right), \tag{49}$$

which in turn implies that $e^{i\eta}=i+\Delta$ with $|\Delta|=O(\binom{n-1}{n/2})$. A similar reasoning holds if $\mathrm{Re}\langle r_0^*|r_0\rangle\leqslant 0$. ■

This means that the initial state can be approximately written as $|\psi_0\rangle\approx 1/\sqrt{2}(|\omega_0'\rangle+|-\omega_0'\rangle)$ and evolves as $(U')^t|\psi_0\rangle\approx 1/\sqrt{2}(e^{it\omega_0'}|\omega_0'\rangle+e^{-i\omega_0'}|-\omega_0'\rangle)$.

As a last ingredient we need to bound the angle ω_0'. These bounds are provided in the following final theorem.

Theorem 5. Each application of the evolution operator U' corresponds to a rotation of angle approx. $1/\sqrt{2^{n-1}}$ in the basis of the two eigenvectors $|\pm\omega_0'\rangle$. More precisely $-1/(c\sqrt{2^{n-1}})-\beta \leqslant \omega_0' \leqslant -1/(c\sqrt{2^{n-1}})+\beta$, where $\beta = O(n^{3/2}/2^n)$

Proof. We will approximate $e^{i\omega_0'}=\langle\omega_0'|U'|\omega_0'\rangle$ by $\langle\alpha|U'|\alpha\rangle$, where $|\alpha\rangle=1/\sqrt{2}(|\psi_0\rangle+e^{i\eta}|\psi_1\rangle)$. Let us first evaluate $|e^{i\omega_0'}-\langle\alpha|U'|\alpha\rangle|$. We can expand U' in terms of its eigenvectors to obtain

$$|e^{i\omega_0'}-\langle\alpha|U'|\alpha\rangle| = \left| e^{i\omega_0'} - \sum_j \left| \langle\omega_j'|\alpha\rangle\right|^2 e^{i\omega_j'}\right|. \quad (50)$$

We then note that from Eq. (42),

$$|\langle\omega_0'|\alpha\rangle|^2 = \sqrt{p_0/2}+\sqrt{p_1/2}. \quad (51)$$

So,

$$\begin{aligned}|e^{i\omega_0'}-\langle\alpha|U'|\alpha\rangle| &= \Bigg| e^{i\omega_0'} - (\sqrt{p_0/2}+\sqrt{p_1/2})e^{i\omega_0'} \\ &\quad + \sum_{|\omega_j'\rangle\neq|\omega_0'\rangle} |\langle\omega_j'|\alpha\rangle|^2 e^{i\omega_j'}\Bigg| \\ &\leqslant |e^{i\omega_0'}(1-\sqrt{p_0/2}-\sqrt{p_1/2})| \\ &\quad + \sum_{|\omega_j'\rangle\neq|\omega_0'\rangle} |\langle\omega_j'|\alpha\rangle|^2 \\ &\leqslant 2(1-\sqrt{p_0/2}-\sqrt{p_1/2}) \\ &\leqslant 2\left(\frac{3n}{2^{n+1}}+\frac{3n}{8c^2\binom{n-1}{n/2}}\right) \end{aligned} \quad (52)$$

with $\sqrt{1-x}\geqslant 1-x$ for $0\leqslant x\leqslant 1$. Using the fact that the binomial coefficients approach the Gaussian distribution for large n, such that

$$\binom{n}{x} = \sqrt{\frac{2}{\pi n}} e^{-(x-n/2)/n/2} 2^n, \quad (53)$$

we can rewrite Eq. (52) taking the leading order terms in n. Recalling that $c>1$, we obtain

$$|e^{i\omega_0'}-\langle\alpha|U'|\alpha\rangle| = O\left(\frac{n^{3/2}}{2^n}\right). \quad (54)$$

Equation (54) is an explicit formula which bounds the distance in the complex plane between the eigenvalue of interest, $e^{i\omega_0'}$, and the matrix element $\langle\alpha|U'|\alpha\rangle$. Figure 4 shows the geometric representation of Eq. (54). Note that

$$\begin{aligned}|\sin\omega_0' - \mathrm{Im}\langle\alpha|U'|\alpha\rangle| &= |\mathrm{Im}(e^{i\omega_0'}-\langle\alpha|U'|\alpha\rangle)| \\ &\leqslant |e^{i\omega_0'}-\langle\alpha|U'|\alpha\rangle|. \end{aligned} \quad (55)$$

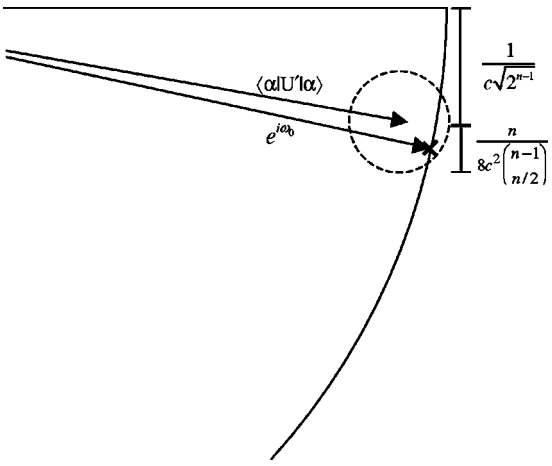

FIG. 4. Geometric representation of Theorem 5, which proves that the eigenvalue, $e^{i\omega_0'}$, must be located on a disc of radius $n/8c^2\binom{n-1}{n/2}$ centered at $\langle\alpha|U'|\alpha\rangle$. The position of the eigenvalue is denoted by a cross.

Next, we evaluate $\mathrm{Im}\langle\alpha|U'|\alpha\rangle$ using Eqs. (31) and (35),

$$\begin{aligned}\mathrm{Im}\langle\alpha|U'|\alpha\rangle &= \mathrm{Im}(e^{i\eta}\langle\psi_0|U'|\psi_1\rangle - e^{i\eta}\langle\psi_1|U'|\psi_0\rangle) \\ &= \mathrm{Im}\frac{1}{2}\Bigg(-\frac{e^{i\eta}}{c\sqrt{2\binom{n-1}{n/2}}}(\langle\psi_0|R,n/2-1\rangle \\ &\quad + \langle\psi_0|L,n/2+1\rangle) - e^{i\eta}\left(-\frac{2}{\sqrt{2^n}}\langle\psi_1|L,1\rangle\right)\Bigg) \\ &= -\mathrm{Im}\frac{e^{i\eta}}{c\sqrt{2^{n-1}}} \\ &= -\frac{1}{c\sqrt{2^{n-1}}} - O\left(\frac{n}{\sqrt{2^n\binom{n-1}{n/2}}}\right). \end{aligned} \quad (56)$$

Then, using Theorem 4, Eqs. (54) and (56), we can write

$$\left|\sin\omega_0' + \frac{1}{c\sqrt{2^{n-1}}} + O\left(\frac{n}{\sqrt{2^n\binom{n-1}{n/2}}}\right)\right| = O\left(\frac{n^{3/2}}{2^n}\right). \quad (57)$$

Using $\sin x = x + O(x^3)$ and keeping only leading order terms solving for ω_0' gives us

$$-\frac{1}{c\sqrt{2^{n-1}}} - O\left(\frac{n^{3/2}}{2^n}\right) \leqslant \omega_0' \leqslant -\frac{1}{c\sqrt{2^{n-1}}} + O\left(\frac{n^{3/2}}{2^n}\right). \quad (58)$$

■

We can now quantitatively describe the overall operation of the algorithm. Starting with initial state $|\psi_0\rangle$, we consider

the state of the computer after t applications of U'. Then using Theorem 4 we can expand $|\psi_0\rangle$ as

$$|\psi_0\rangle = \sqrt{p_0}(|\omega_0'\rangle + |-\omega_0'\rangle) + \delta|r\rangle, \tag{59}$$

where $\delta = \sqrt{1-2p_0} = O(\sqrt{n/2^n})$ and $|r\rangle$ is a residual normalized vector orthogonal to $|\omega_0'\rangle$ and $|-\omega_0'\rangle$. Now

$$\begin{aligned}(U')^t|\psi_0\rangle &= \sqrt{p_0}(e^{i\omega_0' t}|\omega_0'\rangle + e^{-i\omega_0' t}|-\omega_0'\rangle) + \delta|r'\rangle \\ &= 2p_0\cos\omega_0' t|\psi_0\rangle - 2\sqrt{p_0 p_1}(\sin\omega_0' t \\ &\quad + \mathrm{Re}\, e^{i\omega_0' t}\Delta)|\psi_1\rangle + \sqrt{1-p_0-p_1}(e^{i\omega_0' t}|r_0\rangle \\ &\quad + e^{-i\omega_0' t}|r_0^*\rangle) + \delta|r\rangle \\ &= \cos\omega_0' t|\psi_0\rangle - \sin\omega_0' t|\psi_1\rangle + O\left(\frac{n^{3/4}}{\sqrt{2^n}}\right)|\tilde{r}\rangle,\end{aligned} \tag{60}$$

where $|\tilde{r}\rangle$ is some residual normalized vector (not necessarily orthogonal to $|\psi_0\rangle$ and $|\psi_1\rangle$).

Starting with $|\psi_0\rangle$ and applying U' for $t_f = \pi/2|\omega_0'|$ steps, we approximately rotate from $|\psi_0\rangle$ to $|\psi_1\rangle$. From $|\psi_1\rangle$ we can obtain $|\vec{x}_{target}\rangle = |\vec{0}\rangle$ with high probability p using Eq. (33) and $1+1/(2n) \leqslant c^2 \leqslant 1+2/n$ for large n [following from Eq. (34)],

$$\begin{aligned}p &= \sum_d |\langle d,0|\psi_1\rangle|^2 \\ &= |\langle R,0|\psi_1\rangle|^2 \\ &= \frac{1/2}{c^2} \geqslant \frac{1/2}{1+2/n} \\ &= \frac{1}{2} - O(1/n).\end{aligned} \tag{61}$$

Finally, to obtain t_f in terms of n, we make use of the bounds on ω_0' provided by Theorem 5,

$$t_f = \frac{\pi c}{2}\sqrt{2^{n-1}}\left[1 \pm O\left(\frac{n^{3/2}}{\sqrt{2^n}}\right)\right]. \tag{62}$$

Using the inequality $1+1/(2n) \leqslant c^2 \leqslant 1+2/n$ to get $1+1/(4n) \leqslant c \leqslant 1+1/n$, we obtain

$$t_f = \frac{\pi}{2}\sqrt{2^{n-1}}\left[1 + O\left(\frac{1}{n}\right)\right]. \tag{63}$$

If we set the number of time steps to be $t_f = (\pi/2)\sqrt{2^{n-1}}$ (or the closest integer) then

$$-\sin\omega_0' t_f = \sin\frac{\pi}{2}\left[1 - O\left(\frac{1}{n}\right)\right] = 1 - O\left(\frac{1}{n^2}\right). \tag{64}$$

So the probability to measure $|\vec{x}_{target}\rangle$ after $t_f = (\pi/2)\sqrt{2^{n-1}}$ steps is still $p_{success} = 1/2 - O(1/n)$. Hence, by repeating the algorithm a constant number of times, the probability of error can be made arbitrarily small. Note the periodic nature of the evolution under U' [Eq. (60)]; this means that if we measure at $t > t_f$ the probability of success will decrease and later increase again.

In summary, we arrived at the final result that the marked state is identified after $O(\sqrt{N})$ calls to the oracle.

IV. CONNECTION TO GROVER'S ALGORITHM

The main point of this paper is to give a first algorithm in the discrete-time random-walk setting. We have shown how to realize quantum search in this scenario, without losing any of the quantum speedup obtained in Grover's search algorithm. Although the layout of our algorithm is very different from Grover's search, there are several similarities to Grover's algorithm.

Both algorithms begin in the equal superposition state over all bit strings. Both algorithms make use of the Grover diffusion operator G (sometimes known as the Grover iterate). Both algorithms can be viewed as a rotation in a two-dimensional subspace. Both algorithms use an oracle which marks the target state with a phase of -1. Both algorithms have a running time of $O(\sqrt{N})$. In both algorithms we have to measure at a specific time to obtain maximum probability of success. However, there are several important differences between the two search algorithms. In this section, we call attention to the ways in which the random-walk search algorithm is distinct from Grover's algorithm, and consider how these differences affect performance and implementation.

It is well known that Grover's algorithm can be mapped exactly onto a rotation in the two-dimensional subspace spanned by the equal-superposition state $|\psi_0\rangle$ and the marked state $|0\rangle$ [7]. Each iteration in Grover's algorithm corresponds to a rotation in this subspace. In this paper, we have shown that the random-walk search algorithm can also be viewed as a rotation in a two-dimensional subspace. However, there are two important distinctions. First, the random walk search algorithm can only be *approximately* mapped onto a two-dimensional subspace. Unlike Grover's algorithm, this mapping is not exact. Second, the two-dimensional subspace in which the random-walk search algorithm is approximately contained is spanned by $|\psi_0\rangle$ and $|\psi_1\rangle$, not by $|\psi_0\rangle$ and $|0\rangle$. Hence, the final state of the algorithm is not exactly the pure marked state $|0\rangle$ as it is in Grover's algorithm. It is a linear combination of states which is composed primarily of the marked state, but also possesses small contributions from its nearest neighbors, second-nearest neighbors, etc. Thus, the random-walk search algorithm contains traces of the underlying topology of the hypercube on which it is based.

Another crucial difference is the locality of the unitary transformations used during the algorithm. In the random-walk search algorithm the shift operator is local in the topology of the hypercube, i.e. it shifts amplitude only between the n nearest neighbors. The coin operator shifts amplitudes only on the n-dimensional coin space. So we can say that all

our operations in an iteration are n local. Compared to this the reflection operator used in Grover's algorithm is highly nonlocal.

Another difference between the two algorithms is their use of the Grover diffusion operator G. In Grover's algorithm, this operator is applied to the entire 2^n-dimensional search space (corresponding to the node space in the random-walk search algorithm). On the other hand, Grover's diffusion operator G in the random-walk algorithm is used as the quantum coin, and acts only on the n-dimensional coin space. This fact may be of practical use for certain physical implementations since many physical implementations of quantum computers contain multiple types of qubits, which have different natural gate sets. We could exploit this variety using the random-walk search algorithm by choosing the coin space to be represented by qubits on which it is convenient to implement the Grover diffusion operator. Similarly, it might be natural and easy for some physical systems to implement the shift operator rather than the gates required in Grover's search algorithm. It is ultimately the physical system that will determine which of the search algorithms is more advantageous.

Another similarity between the two algorithms is the implementation of the oracle. In Grover's algorithm, the oracle marks the target state with a phase of -1. To arrive at this random walk search algorithm, we chose the marking coin C_1 to be the $-\mathcal{I}$ coin. This choice was actually motivated because it yielded a result that was amenable to analysis, and while the emergence of Grover's algorithm appears natural in hindsight, it was not obvious at the outset. However, more generally, it is not clear whether this choice of marked coin is either optimal or unique. In fact, numerical simulations have shown us that many different types of marking coins will yield search algorithms [15]. Unfortunately, analytic treatment of the quantum random walk for more complicated coins has proven substantially more difficult than the instance analyzed here for $C_1=-\mathcal{I}$. It is an open question what (constant factor) gains might be made by using different marking coins to implement the search.

V. CONCLUSIONS

In this paper, we have shown that the random-walk search algorithm can search a list of 2^n items in time proportional to $\sqrt{2^n}$. The lower bound on a quantum search of an N-item list is known to be $\Omega(\sqrt{N})$ [9]. Thus, up to a constant factor, the random-walk search algorithm is optimal. However, although after repetition of the algorithm a constant number of times the result is arbitrarily close to the result of Grover's search, the random-walk search algorithm is not identically equivalent to Grover's algorithm. In particular, the final solution obtained by the random-walk search still retains some of the underlying character of the hypercube on which it was based, with a small admixture of states other than the solution at the marked node.

The random walk search analyzed here was based on a discrete walk on the hypercube. In general, a similar methodology can be applied to any regular graph, e.g., a two-dimensional hexagonal lattice with periodic boundary conditions, a three-dimensional rectangular lattice with periodic boundary conditions, etc. We have numerical evidence indicating that this methodology will yield quantum search algorithms when applied to other regular n-dimensional lattices. Future studies will investigate the extent of optimality of such search algorithms.

The intriguing possibility of finding novel algorithms based on the random walk also remains an open question. The results described here indicate that the random-walk search algorithm provides a suggestive framework for new algorithms. Though the optimality of Grover's algorithm precludes the construction of an improved oracle-based search algorithm based on a quantum walk, nevertheless, many other oracle problems still exist for which a quantum walk may be advantageous. For instance, the lower bound on quantum search holds only for oracles that provide ''yes/no'' information [9]. Our choice of marking coin here has a clear relation to an identifiable component of Grover's algorithm. In general, the marking coin can be an arbitrary $n\times n$ unitary matrix. The marking coin provides a intuitive means by which to introduce a large amount of information to an oracle problem. Thus, it is possible that unique coins with interesting properties may give rise to an entirely new algorithm. Overall we conclude that the quantum random walk provides a means for insight into existing quantum algorithms and offers a potentially vast source for development of new algorithms.

ACKNOWLEDGMENTS

N.S. thanks the University of California, for a Berkeleyan Fellowship. This effort is sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Laboratory, Air Force Material Command, USAF, under Contract No. F30602-01-2-0524. We also thank NSF ITR/SY Grant No. 0121555.

[1] Y. Aharonov, L. Davidovich, and N. Zagury, Phys. Rev. A **48**, 1687 (1993).
[2] D. Aharonov, A. Ambainis, J. Kempe, and U. Vazirani, in *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computation (STOC)* (ACM, New York, 2001), pp. 50–59; e-print quant-ph/0012090.
[3] A. Ambainis, E. Back, A. Nayak, A. Vishwanath, and J. Watrous, in *Proceedings of the 33rd Annual ACM Symposium on the Theory of Computation (STOC)* (ACM, New York, 2001), pp. 60–69.
[4] E. Farhi and S. Gutmann, Phys. Rev. A **58**, 915 (1998).
[5] A. Childs, E. Farhi, and S. Gutmann, Quantum Inform. Process. **1**, 35 (2002).
[6] A. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. Spielman, e-print quant-ph/0209131.
[7] L. Grover, in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computation* (ACM Press, New York, 1996), pp. 212–219.
[8] L. Grover, Phys. Rev. Lett. **79**, 325 (1997).

[9] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM J. Comput. **26**, 1510 (1997).
[10] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
[11] C. Moore and A. Russell, in *Proceedings of RANDOM, 2002*, edited by J. D. P. Rolim and P. Vadham (Springer, Cambridge, MA, 2002), pp. 164–178.
[12] J. Kempe, e-print quant-ph/0205083.
[13] T. Yamasaki, H. Kobayashi, and H. Imai, in *Proceedings of the 3rd UMC*, edited by C. Calude, M. J. Dinneen, and F. Peper (Springer, Kobe, 2002), pp. 315–330.
[14] J. Watrous, J. Comput. Syst. Sci. **62**, 376 (2001).
[15] N. Shenvi (unpublished).