

## Effects of a random noisy oracle on search algorithm complexity

Neil Shenvi, Kenneth R. Brown, and K. Birgitta Whaley

*Department of Chemistry and the Kenneth S. Pitzer Center for Theoretical Chemistry, University of California, Berkeley, California 94720, USA*

(Received 3 April 2003; published 20 November 2003)

Grover's algorithm provides a quadratic speed-up over classical algorithms for unstructured database or library searches. This paper examines the robustness of Grover's search algorithm to a random phase error in the oracle and analyzes the complexity of the search process as a function of the scaling of the oracle error with database or library size. Both the discrete- and continuous-time implementations of the search algorithm are investigated. It is shown that unless the oracle phase error scales as  $O(N^{-1/4})$ , neither the discrete- nor the continuous-time implementation of Grover's algorithm is scalably robust to this error in the absence of error correction.

DOI: 10.1103/PhysRevA.68.052313

PACS number(s): 03.67.Lx

### I. INTRODUCTION

Issues of fault tolerance and error correction are of both theoretical and practical interest in designing and implementing quantum algorithms. One approach to diminishing the effects of error focuses on error correcting codes, which can be used to correct errors once they have occurred [1,2]. Another approach which has also proven successful is that of decoherence-free subspaces, which involves mapping a computation onto a subspace that is relatively free from error [3]. This second approach is of interest because it is an example of passive, rather than active, error correction. The success of the passive approach leads to the question of whether existing quantum algorithms are inherently robust to errors, and, if they are not, whether there might exist modified implementations of these algorithms that are robust to certain errors, i.e., without requiring active error correction.

In this paper, we analyze the robustness of Grover's algorithm to error. Grover's search algorithm is the basis for a number of quantum algorithms [4–6]. Most of these algorithms involve the use of an oracle, a black box device which takes as input a quantum state in the computational basis and returns as output some function of that state. Given a perfect, noiseless oracle, Grover's search algorithm attains a quadratic speed-up over classical search algorithms. However, this speed-up is predicated upon the perfect implementation of the oracle. Although, for the purposes of analysis, the oracle is often treated as a “black box” whose inner workings are unknown, any physical implementation of Grover's algorithm must also include a physical implementation of the oracle and this may be imperfect. Therefore it is of interest to ascertain what effect the accuracy of the oracle implementation has on the overall complexity of the algorithm. We make use of this term here as it is employed in computer science terminology, namely, the complexity of an algorithm is defined as the number of computational steps required in order to achieve a predetermined fixed probability of success.

Several previous papers have studied the effects of oracle noise on Grover's algorithm, using various models [7–11]. We consider here the random phase error model, addressing it within both discrete- and continuous-time implementations of the search algorithm. The effect of random phase errors on

the discrete-time Grover's algorithm was already studied numerically in Ref. [7]. In this paper we derive analytic results for this model and present numerical evidence verifying the validity of these results. We analyze the complexity of the search algorithm as a function of the scaling of the errors, and arrive at bounds on the error that must be satisfied for a constant probability of success. In particular, we find that in order to achieve a constant success probability independent of the library size that is being searched, the oracle error must scale at most as  $N^{-1/4}$ , where  $N$  is the library size. We also analyze the effect of phase errors on a continuous-time analog of the search algorithm. This is found to be relatively straightforward to study within a density-matrix approach. We find that this leads to similar results for the complexity as a function of the scaling of the errors, namely, that there is an upper bound of  $O(N^{1/2})$  on algorithmic complexity for  $\delta \geq 1/4$  and  $O(N^{1-2\delta})$  for  $\delta \leq 1/4$ , when the error scaling goes as  $N^{-\delta}$ . We further show that these bounds are tight; in other words, the algorithmic complexity has a lower bound of  $\Omega(N^{1/2})$  for  $\delta \geq 1/4$  and  $\Omega(N^{1-2\delta})$  for  $\delta \leq 1/4$ . [Following standard computer science notation, we will use the following to characterize the growth of certain functions: We will say that  $f(n)$  is bounded above by  $g(n)$ , i.e.,  $f(n) = O(g(n))$ , if there are positive constants  $c$  and  $k$  such that  $0 \leq f(n) \leq cg(n)$  for  $n \geq k$ . Similarly,  $f(n)$  is bounded below by  $g(n)$ , i.e.,  $f(n) = \Omega(g(n))$  if  $0 \leq cg(n) \leq f(n)$  for constants  $c, k \geq 0$  and  $n \geq k$ .]

This result has practical importance in determining the amount of oracle error that is allowable as the size of the library is increased. We will show that our complexity results imply that if the library size is increased by a factor of  $k$ , the oracle error must decrease by a factor of  $k^{1/4}$  in order to attain a similar probability of success.

The remainder of this paper is organized as follows. Section II describes the random-phase error model for a noisy oracle, summarizes the Grover search algorithm in a discrete-time implementation, and then derives the effect of the oracle noise on this implementation. Section III derives the effect of oracle noise on the continuous-time formulation of Grover's algorithm proposed by Farhi and co-workers in Refs. [12,13]. We summarize and present conclusions in Sec. IV.

## II. DISCRETE-TIME SEARCH

### A. Quantum search with a phase oracle

A phase oracle is a quantum oracle that “marks” one or more computational basis states with a specified phase (usually  $-1$ ). For a function  $f(\mathbf{x}) \rightarrow \{0,1\}$ , where  $\mathbf{x}$  denotes an  $n$ -bit binary string, a phase oracle implements the following operation:

$$|\mathbf{x}\rangle \xrightarrow{O} (-1)^{f(\mathbf{x})} |\mathbf{x}\rangle. \quad (1)$$

The search problem is as follows. Given an oracle  $O$  which implements the function  $f(\mathbf{x}) \rightarrow \{0,1\}$ , find a state in the computational basis  $\mathbf{x}$  for which  $f(\mathbf{x})=1$ . Assuming a perfect, noiseless oracle (the concept of a noisy oracle is introduced below), the well-known result for the complexity of Grover’s algorithm, i.e., for the quantum search problem, is  $O(\sqrt{N/M})$ , where  $M$  is the number of states for which  $f(\mathbf{x})=1$  [14–17]. In contrast, the classical complexity of the search problem is  $O(N/M)$  [18]. Thus, the quantum algorithm provides a quadratic speed-up over the classical algorithm.

### B. Grover’s algorithm in absence of noise

The description of Grover’s algorithm in this paper follows the discussion in Ref. [19]. The search is made on the set of  $N=2^n$  states represented by all  $n$ -bit binary strings  $x \in \{0,1\}^n$ . The states are denoted by  $|x\rangle$ . Within a discrete-time computation model, Grover’s algorithm searches for marked or “target” states via repeated application of the Grover iteration operator  $G$ , which can be written as

$$G = (2|\eta\rangle\langle\eta| - I)O, \quad (2)$$

where  $|\eta\rangle = (1/\sqrt{N})\sum_{\mathbf{x}=0}^{N-1} |\mathbf{x}\rangle$  is the equal superposition over all states. From this point on, we will assume that  $M=1$  to simplify the discussion. The arguments can easily be extended to the case where  $M>1$ . Let the state  $|\tau\rangle$  be the “marked” state. Then we can write out the explicit form of the oracle as

$$O = I + (e^{i\pi} - 1)|\tau\rangle\langle\tau|. \quad (3)$$

Assuming that our registers are initialized to  $|\eta\rangle$ , it can be shown that after  $O(\sqrt{N})$  applications of  $G$ , the quantum computer is approximately in the marked state  $|\tau\rangle$  [14,15]. This result is demonstrated by noting that the search algorithm can be mapped onto a two-state subspace spanned by the two basis vectors  $|1\rangle$  and  $|2\rangle$ , where

$$|1\rangle = |\tau\rangle, \quad (4)$$

$$|2\rangle = \frac{|\eta\rangle - |\tau\rangle\langle\tau|\eta\rangle}{\sqrt{1 - |\langle\tau|\eta\rangle|^2}}. \quad (5)$$

In other words,  $|1\rangle$  is the marked state and  $|2\rangle$  is the equal superposition over all unmarked states.

When our initial state  $|\eta\rangle$  and the operator  $G$  are rewritten in the  $|1\rangle, |2\rangle$  basis,

$$|\eta\rangle = \begin{pmatrix} \sqrt{\frac{1}{N}} \\ \sqrt{\frac{N-1}{N}} \end{pmatrix}, \quad (6)$$

$$G = \begin{pmatrix} 1 - \frac{2}{N} & \frac{2\sqrt{N-1}}{N} \\ -\frac{2\sqrt{N-1}}{N} & 1 - \frac{2}{N} \end{pmatrix} = \begin{pmatrix} \cos(\Theta/2) & \sin(\Theta/2) \\ -\sin(\Theta/2) & \cos(\Theta/2) \end{pmatrix}, \quad (7)$$

where

$$\sin(\Theta/2) = \frac{2\sqrt{N-1}}{N}, \quad (8)$$

we see that the effect of repeated applications of  $G$  is to make successive rotations by  $\Theta/2$  on state  $|\eta\rangle$ . This convention for the definition of  $\Theta$  is useful because  $\Theta$  represents the angle of rotation applied by the operator  $G$  on the Bloch sphere defined for the Grover subspace spanned by the basis states  $|1\rangle$  and  $|2\rangle$ . We can therefore view Grover’s algorithm as the process of rotating our initial state  $|\eta\rangle$  onto the target state  $|\tau\rangle \equiv |1\rangle$  by a discrete sequence of small rotations. For small  $\Theta$  (i.e., large  $N$ ),  $\sin\Theta \approx \Theta$ , yielding an incremental rotation angle of approximately  $(4\sqrt{N-1})/N \approx 4/\sqrt{N}$ . Then a rotation of  $\pi$  rad on the Bloch sphere requires  $O(\sqrt{N})$  applications of  $G$ . Hence, Grover’s search algorithm requires  $O(\sqrt{N})$  calls to the oracle. The well-known lower bound for quantum search has been established as  $\Omega(\sqrt{N})$  [16]. Thus, Grover’s algorithm is an optimal search.

### C. Grover’s algorithm in presence of oracle noise

This  $O(\sqrt{N})$  bound on the running time of the search assumes that the oracle exactly implements the phase operation specified by Eq. (1). Instead, we could envision a more realistic oracle which implements Eq. (1) with some finite degree of precision. Specifically, we adopt here the concept of a noisy phase oracle introduced in Ref. [7] which implements the following operation:

$$|\mathbf{x}\rangle \xrightarrow{O} (e^{i(\pi+\epsilon)})^{f(\mathbf{x})} |\mathbf{x}\rangle, \quad (9)$$

where  $\epsilon$  is a small, random phase error ( $\epsilon \ll \pi$ ) with probability distribution  $p(\epsilon)$ . We will make the assumption that the error is zero-mean, i.e.,  $\langle\epsilon\rangle=0$ . For  $p(\epsilon)=\delta(\epsilon)$ , the noiseless oracle is recovered. In practice, the root-mean-square magnitude of  $\epsilon$  can be made small through a careful physical implementation of the oracle. However, due to the finite precision of any experimental implementation, the average magnitude of  $\epsilon$  will never be zero, i.e.,  $\epsilon_{rms} = \sqrt{\langle\epsilon^2\rangle} > 0$ .

Given this more realistic oracle model, we now investigate whether Grover’s search algorithm is still  $O(\sqrt{N})$ . The explicit form of the noisy oracle can be written as

$$O = I + (e^{i(\pi+\epsilon)} - 1)|\tau\rangle\langle\tau|. \quad (10)$$

Then Grover's iteration operator  $G$  becomes

$$G = \begin{pmatrix} \left(-1 + \frac{2}{N}\right)e^{i(\pi+\epsilon)} & \frac{2\sqrt{N-1}}{N} \\ \frac{2\sqrt{N-1}}{N}e^{i(\pi+\epsilon)} & 1 - \frac{2}{N} \end{pmatrix}, \quad (11)$$

which is clearly still unitary. To see the effects of the random error, we first note that any single-qubit unitary operator can be written as

$$U = \exp(i\alpha)R_{\hat{n}}(\Phi) = \exp(i\alpha)\left[\cos\left(\frac{\Phi}{2}\right)I - i\sin\left(\frac{\Phi}{2}\right)\hat{n}\cdot\vec{\sigma}\right], \quad (12)$$

where  $\vec{\sigma}$  are the Pauli operators. Thus, Grover's iteration operator using the noisy oracle is equivalent, up to an overall global phase factor, to a rotation  $R_{\hat{n}}(\Phi)$  on the Bloch sphere by some angle  $\Phi$  about some direction  $\hat{n}$ . We can determine the values of  $\Phi$  and  $\hat{n}$  by using Eqs. (11) and (12),

$$\cos\left(\frac{\Phi}{2}\right) = \left(1 - \frac{2}{N}\right)\cos\left(\frac{\epsilon}{2}\right), \quad (13)$$

$$\sin\left(\frac{\Phi}{2}\right)n_x = \frac{2\sqrt{N-1}}{N}\sin\left(\frac{\epsilon}{2}\right), \quad (14)$$

$$\sin\left(\frac{\Phi}{2}\right)n_y = -\frac{2\sqrt{N-1}}{N}\cos\left(\frac{\epsilon}{2}\right), \quad (15)$$

$$\sin\left(\frac{\Phi}{2}\right)n_z = -\left(1 - \frac{2}{N}\right)\sin\left(\frac{\epsilon}{2}\right). \quad (16)$$

A noiseless oracle can be recovered using these equations by setting  $\epsilon$  to zero. In that case we obtain the large- $N$  Grover's rotation angle  $\Phi \approx 4/\sqrt{N}$ , with  $n_x = n_z = 0$  and  $n_y = 1$ . Hence, the noiseless Grover's algorithm corresponds to a rotation on the Bloch sphere about the  $y$  axis. The effect of a noisy oracle is to add small, random perturbations to this rotation axis, thereby changing the pure  $y$ -axis rotation to a rotation about an axis  $\hat{n}$  which also contains nonzero  $x$  and  $z$  components.

To analyze the running time of the noisy search algorithm, we will take the limit of large  $N$ . This is a useful assumption for our current purposes since we are interested in the complexity of the search algorithm for large  $N$ . From Eqs. (13)–(16), we can express  $G$  up to a global phase factor as

$$G = \exp\left(-i\Phi\left\{\left(-1 + \frac{2}{N}\right)\sin\left(\frac{\epsilon}{2}\right)\sigma_z + \frac{2\sqrt{N-1}}{N}\left[\sin\left(\frac{\epsilon}{2}\right)\sigma_x - \cos\left(\frac{\epsilon}{2}\right)\sigma_y\right]\right\} / \sin\left(\frac{\Phi}{2}\right)\right), \quad (17)$$

where

$$\sin\left(\frac{\Phi}{2}\right) = \sqrt{1 - \left(1 - \frac{2}{N}\right)^2 \cos^2\left(\frac{\epsilon}{2}\right)}.$$

Since both  $\epsilon$  and  $N^{-1/2}$  are small parameters, we can use a double Taylor expansion of the terms in the exponent with respect to  $\epsilon$  and  $N^{-1/2}$ . Excluding second-order terms inserted and higher and higher, we obtain

$$G = \exp\left[i\left(\epsilon\sigma_z + \frac{4}{\sqrt{N}}\sigma_y\right)\right]. \quad (18)$$

Again using the fact that  $\epsilon$  and  $N^{-1/2}$  are both small parameters, we can make use of the Baker-Campbell-Hausdorff formula [20] to arrive at

$$G = \exp(i\epsilon\sigma_z)\exp\left(i\frac{4}{\sqrt{N}}\sigma_y\right) + O\left(\frac{\epsilon}{\sqrt{N}}\right) \\ \approx R_z(-\epsilon)R_y\left(-\frac{4}{\sqrt{N}}\right). \quad (19)$$

Examining Eq. (19), we observe that there are two important time scales that will be relevant to any physical implementation. The first is the time scale of the  $y$ -axis rotation,  $\sqrt{N}$ . Taking the limit as  $\epsilon \rightarrow 0$ , we see that the  $y$ -axis rotation  $R_y$  is the "driving force" of the search algorithm, in that it rotates the initial state  $|\eta\rangle$  towards the target state  $|\tau\rangle$ . The second important time scale,  $1/\epsilon$ , is the time scale of the random rotation about the  $z$  axis of the Bloch sphere,  $R_z$ . This rotation is the effect of noise and does not advance the search process. Because  $\epsilon$  is a random variable, this rotation about  $z$  will in general be different for each application of  $G$ . However,  $\epsilon$  does have a characteristic root-mean-square magnitude  $\epsilon_{rms}$ , which is determined by the probability distribution  $p(\epsilon)$ . Then the characteristic time scale of the  $z$ -axis rotation is determined by  $1/\epsilon_{rms}$ .

In general,  $\epsilon_{rms}$  and  $\sqrt{N}$  are independent parameters:  $\epsilon_{rms}$  is the phase error that is specific to the experimental implementation, while  $N$  is the size of the database. However, in reality, these two parameters will be linked for a given experimental setup because the size of the database (i.e., the number of states  $N = 2^n$  accessed by the  $n$  qubits of the computer) will affect the accuracy of the oracle implementation. Note that the oracle acts on all  $n$  qubits [see Eq. (9)]. It seems very unlikely that  $\epsilon_{rms}$  would decrease as  $N$  increases, since a greater number of qubits generally introduces more potential for noise. In the best-case scenario, it might be possible to find a physical implementation for which  $\epsilon_{rms}$  is essentially constant over a large range of values for  $N$ . To ensure that our analysis is as general as possible, we will assume here that  $\epsilon_{rms}$  scales as  $N^{-\delta}$ , where  $\delta$  is some constant that may take both positive and negative values, or zero. We can then determine what effect the scaling constant  $\delta$  has on algorithm performance. In particular, we shall determine the effect of  $\delta$  upon the complexity of the algorithm, i.e., upon the number of Grover iterations required to achieve a given probability of successful measurement of the target state  $|1\rangle$ . This will allow us to further determine the maxi-

imum allowable oracle error scaling that ensures a constant probability of success independent of the library size, for a given rate of library growth.

#### D. Dynamics and scaling of the noisy Grover search algorithm

To analyze the dynamics of the Grover iteration, we observe that any pure state on the Bloch sphere can be represented in spherical coordinates by two angles,  $\theta$  and  $\phi$ , where we define  $\theta$  to be the angle from the  $z$  axis and  $\phi$  to be the azimuthal angle. Here we have chosen our coordinates such that the target state  $|1\rangle$  is the south pole ( $\theta = \pi$ ) and the state  $|2\rangle$  is the north pole ( $\theta = 0$ ) of our Bloch sphere. Because the operator  $G$  depends upon the random variable  $\epsilon$ , each application of  $G$  induces a new probability distribution over the Bloch sphere, which we will denote by  $f(\theta, \phi)$ . In this notation, our initial state at time  $t=0$  is given by a  $\delta$  function (i.e., a pure state) centered on  $\phi=0$ , with  $\theta = \sin^{-1}(1/\sqrt{N}) \approx 1/\sqrt{N}$ .

Using this notation, we can write the probability of obtaining the state  $|1\rangle$  upon measurement after  $t$  iterations as

$$\begin{aligned} P(t) &= 1 - \frac{1}{2} \int \int \cos(\theta) f_t(\theta, \phi) \sin(\theta) d\theta d\phi \\ &= \frac{1}{2} (1 - \langle \cos(\theta) \rangle) \\ &= \frac{1}{2} (1 - \langle z \rangle). \end{aligned} \quad (20)$$

Let us define the complexity of our search algorithm to be the number of iterations required to obtain a success probability of  $2/3$ . We immediately see that if the probability of obtaining state  $|1\rangle$  upon measurement after  $t$  iterations is  $P(t)$ , then by repeating this procedure approximately  $2/3P(t)$  times, we can boost the overall success probability to  $2/3$ . It should be noted that the choice of the constant  $2/3$  is arbitrary; in general, the choice of constant will not affect the complexity of the algorithm. Thus, the overall complexity of our algorithm is  $O(t/P(t))$ . Furthermore, Eq. (20) states that in order to succeed with some desired probability  $P^*$ , our probability distribution function  $f(\theta, \phi)$  must be non-negligible when  $\theta \geq \theta^*$ , where

$$\theta^* = \cos^{-1}(1 - 2P^*). \quad (21)$$

In other words, to obtain some desired probability of success,  $P^*$ , there must be a high probability of reaching points on the Bloch sphere with polar angle greater than  $\theta^*$ .

Having established this terminology, we will now give a phenomenological description of the evolution of the probability distribution over the Bloch sphere. Specifically, we would like to know what dependence the magnitude and scaling of the error  $\epsilon_{rms}$  has on the maximum attainable polar angle.

We first rewrite the effects of the Grover iterate as a function of polar coordinates. In the polar coordinates defined above, the  $z$  rotation can be written as

$$R_z^{\epsilon} \phi \rightarrow \phi + \epsilon, \quad (22)$$

$$R_z^{\epsilon} \theta \rightarrow \theta. \quad (23)$$

If we momentarily neglect the  $y$ -axis rotations, our  $z$ -axis rotation dynamics correspond to a random walk on the variable  $\phi$  with periodic boundary conditions. To analyze the effects of the  $y$  rotation, we then take advantage of the fact that  $1/\sqrt{N}$  is a small quantity and expand in powers of  $1/\sqrt{N}$  to obtain

$$R_y^{\phi} \phi \rightarrow \phi - \sin \phi \frac{\cos \theta}{\sin \theta} \frac{4}{\sqrt{N}} + O(1/N), \quad (24)$$

$$R_y^{\phi} \theta \rightarrow \theta + \cos \phi \frac{4}{\sqrt{N}} + O(1/N). \quad (25)$$

Finally, we can write the combined effects of our noisy Grover's operator  $G$  using Eqs. (22)–(25), as

$$G \phi \rightarrow \phi - \sin \phi \frac{\cos \theta}{\sin \theta} \frac{4}{\sqrt{N}} + \epsilon, \quad (26)$$

$$G \theta \rightarrow \theta + \cos \phi \frac{4}{\sqrt{N}}, \quad (27)$$

where we have dropped terms of  $O(1/N)$ .

Having written the dynamics of the Grover operator in terms of polar coordinates on the Bloch sphere, we now consider the probability distribution of the quantum state over the Bloch sphere after  $T$  Grover iterations. We will analyze the dynamics for  $\phi \ll 1$ , and then consider when this approximation is valid. In this regime, we can approximate the Grover operator dynamics as

$$G \phi \rightarrow \phi + \epsilon, \quad (28)$$

$$G \theta \rightarrow \theta + \frac{4}{\sqrt{N}}. \quad (29)$$

It is evident that in this small  $\phi$  limit, the  $\phi$  dynamics are completely determined by  $z$ -axis rotation, which results in a random walk on the variable  $\phi$  that is controlled by the random variable  $\epsilon$ . The central limit theorem tells us that regardless of the probability distribution  $p(\epsilon)$  from which the random variable  $\epsilon$  is sampled, after an adequate number of iterations the distribution of  $\phi$  will converge to a Gaussian with width  $\epsilon_{rms} \sqrt{T}$ :

$$\phi_{rms} \propto \epsilon_{rms} \sqrt{T}, \quad (30)$$

where  $T$  is the number of iterations.

Let us now consider when this approximation is valid. Clearly, the condition that  $\phi \ll 1$  is satisfied for the initial state  $\phi_0 = 0$ . We recall that  $\epsilon_{rms}$  is assumed to scale as  $N^{-\delta}$ , where  $\delta$  is some constant. The validity of our small  $\phi$  ap-

proximation is found to be highly dependent on the scaling exponent  $\delta$ . We consider two cases:  $\delta > 1/4$  and  $\delta \leq 1/4$ .

If  $\delta > 1/4$ , we examine the probability distribution after  $T = \lambda \sqrt{N}$  iterations, where  $\lambda$  is some small constant. Inspection of Eq. (30) shows that the probability distribution of  $\phi$  after  $T$  steps will then have width  $\sqrt{\lambda} N^{1/4 - \delta}$ . Since  $N$  can certainly be made arbitrarily large, this justifies our assumption that  $\phi$  is small. Within this regime, Eq. (28) shows that the dynamics of the variable  $\theta$  are simply those of the deterministic  $y$  rotation with constant increment  $4/\sqrt{N}$ . After  $T$  iterations of this, we obtain

$$\begin{aligned} \theta_T &= \theta_0 + \frac{4T}{\sqrt{N}}, \\ &\approx \frac{4T}{\sqrt{N}}, \end{aligned} \quad (31)$$

which is the same as the large- $N$  limit of the noiseless Grover search, Eq. (9). Thus when  $\delta > 1/4$ , the error has no effect on algorithmic complexity for significantly large  $N$ , yielding a complexity of  $O(N^{1/2})$ . This bound is trivially tight since the search problem is well known to have a lower bound of  $\Omega(N^{1/2})$  [16]. For the remainder of this section, we will therefore analyze the complementary case, i.e., when  $\epsilon_{rms}$  scales as  $N^{-\delta}$  with  $\delta \leq 1/4$ , and determine the effect of the error on algorithmic complexity. We note that this latter case includes the best-case physical situation of  $\epsilon_{rms}$  independent of  $N$ , i.e.,  $\delta = 0$ .

In order to analyze the system dynamics for  $\delta \leq 1/4$ , let us examine the probability distribution after  $T$  iterations such that

$$T = \lambda / \epsilon_{rms}^2, \quad (32)$$

where  $\lambda$  is a small constant. By Eq. (30), the distribution width  $\phi_{rms}$  is proportional to  $\sqrt{\lambda}$ . Thus, by selecting a small enough constant  $\lambda$ , our approximation that  $\phi$  is small is again justified. Using Eqs. (28) and (31), we find that the  $\theta$  rotation is again essentially deterministic, and that

$$\begin{aligned} \theta_T &= \frac{4T}{\sqrt{N}} \\ &= \frac{4\lambda}{\epsilon_{rms}^2 \sqrt{N}} \\ &= O\left(\frac{1}{\epsilon_{rms}^2 \sqrt{N}}\right). \end{aligned} \quad (33)$$

Using Eq. (20), we obtain

$$P(T) = \frac{1}{2} \left[ 1 - \cos\left(\frac{\lambda}{\epsilon_{rms}^2 \sqrt{N}}\right) \right] \approx \frac{\lambda^2}{4 \epsilon_{rms}^4 N}, \quad (34)$$

where we have used the fact that  $\lambda / \epsilon_{rms}^2 \sqrt{N} \ll 1$  when  $\delta \leq 1/4$ . By proving that we can attain (essentially determinis-

tically, as described above) a polar rotation of *at least*  $4\lambda / \epsilon_{rms}^2 \sqrt{N}$  in  $T = \lambda / \epsilon_{rms}^2$  iterations, we have shown that the complexity of the noisy search algorithm is  $O(T/P(T)) = O(\epsilon_{rms}^2 N)$ . Taking the scaling of  $\epsilon_{rms}$  into account ( $\epsilon_{rms} \sim N^{-\delta}$ ) leads to the overall algorithmic complexity  $O(N^{1-2\delta})$  [21]. So for  $\delta = 1/4$  we obtain an optimal solution having the same speed-up as the noiseless quantum search. For  $0 < \delta < 1/4$ , we find a scaling intermediate between the noiseless quantum and classical search algorithms. The quantum speed-up factor decreases as  $\delta$  approaches zero and is completely lost when  $\delta = 0$ . (Note that the quantum search formally becomes slower than the classical search in the worst-case scenario when the phase errors increase with  $N$ , i.e.,  $\delta < 0$ .)

In order to show that this complexity bound is tight, we now will show that we can attain *at most* a polar rotation of  $O(1/\epsilon_{rms}^2 \sqrt{N})$  in subclassical time. Repeating the argument above following Eq. (33) will then lead to the identification of  $\Omega(N^{1-2\delta})$  as a lower bound. Thus, we find that  $O(N^{1-2\delta})$  is a tight bound on the complexity in presence of noise. Obtaining a lower bound on the algorithmic complexity is important because without it, it would be unclear whether we could further reduce our search complexity by running the noisy search algorithm for more than  $O(1/\epsilon_{rms}^2)$  iterations. The lower bound will demonstrate that after  $O(1/\epsilon_{rms}^2)$  iterations, it is impossible to achieve a superclassical enhancement in success probability.

To achieve a polar rotation greater than  $O(1/\epsilon_{rms}^2 \sqrt{N})$ , we will need to obtain a significant probability density on the region of the Bloch sphere described by  $\theta \geq c / \epsilon_{rms}^2 \sqrt{N}$ , where  $c$  is a constant. Let us therefore consider the dynamics of a state on this region of the Bloch sphere. Given a pure state with  $\phi_0 = 0$  and  $\pi/2 \geq \theta_0 \geq c / \epsilon_{rms}^2 \sqrt{N}$ , we examine the action of  $T$  iterations of the Grover algorithm on this state, where  $T = \lambda / \epsilon_{rms}^2$ . Now, instead of choosing  $\lambda$  small as previously, we select  $\lambda \gg 1$ , which ensures a large width for the long-time distribution of the azimuthal angle  $\phi$ , Eq. (30). Thus after  $T$  iterations with this large width, we obtain a completely uniform distribution on the variable  $\phi$ . We now consider what happens to the polar variable  $\theta$ . Referring to Eqs. (23) and (25), we find that after  $T$  iterations we obtain

$$\begin{aligned} |\theta_T - \theta_0| &\leq \frac{T}{\sqrt{N}} \\ &\leq \frac{\lambda}{\epsilon_{rms}^2 \sqrt{N}} \\ &= O(N^{-1/2+2\delta}). \end{aligned} \quad (35)$$

Again, provided that  $\delta \leq 1/4$ , this difference can be made arbitrarily small by selecting a large enough  $N$ . Thus, after  $T$  iterations, our probability distribution of the polar coordinate  $\theta$  is confined within an arbitrarily small window around  $\theta_0$ , but is uniformly distributed on the azimuthal coordinate  $\phi$ . Figure 1 shows a graphical representation of these dynamics.

Let us now consider the effects when the Grover operator is subsequently applied to such a probability distribution that

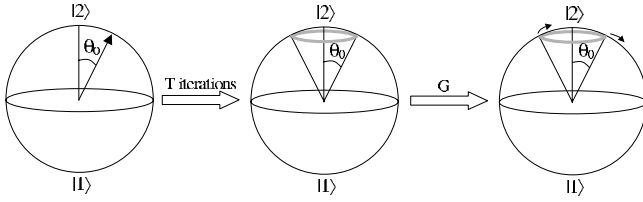


FIG. 1. Graphical representation of the noisy oracle Grover algorithm dynamics when  $\pi/2 \geq \theta \geq c/\epsilon_{rms}^2 \sqrt{N}$ . The target state  $|1\rangle$  is located at the south pole of the Bloch sphere. The north pole is state  $|2\rangle$ , Eq. (5). After  $T$  iterations, a pure state at  $\theta_0$  and  $\phi_0$  will become completely mixed with respect to the  $\phi$  variable but will be confined to an arbitrarily small window around  $\theta_0$  in the  $\theta$  variable. Subsequent applications of  $G$  will then give an equal probability of increasing or decreasing the polar angle, corresponding to a random walk in  $\theta$ .

is uniform in  $\phi$ . We will show that there exists a symmetry in the associated  $\theta$  transformations in this regime, such that one half of all azimuthal angles  $\phi$  are associated with a  $\theta$  rotation in one direction, and the other half are associated with  $\theta$  rotations of equal magnitude in the opposite direction. To reveal this symmetry, we show that the transformation  $\phi \rightarrow \phi + \pi$  on Eqs. (26) and (27) results in an exactly opposite rotation in the  $\theta$  direction to that associated with the azimuthal angle  $\phi$ . From Eq. (35), we know that the distribution of  $\theta_T$  will be arbitrarily close to  $\theta_0$ . Since we have assumed that our initial state was located in the region  $\pi/2 \geq \theta_0 \geq c/\epsilon_{rms}^2 \sqrt{N}$ , we can rewrite Eq. (26) as follows:

$$\begin{aligned} G \phi &\rightarrow \phi - \sin \phi \frac{\cos \theta}{\sin \theta} \frac{1}{\sqrt{N}} + \epsilon \\ &\approx \phi - \sin \phi \frac{\epsilon_{rms}^2 \sqrt{N}}{c} \frac{1}{\sqrt{N}} + \epsilon \\ &= \phi - \sin \phi \frac{\epsilon_{rms}^2}{c} + \epsilon. \end{aligned} \quad (36)$$

We now make use of the arbitrariness in choice of the constant  $c$ , choosing  $c \geq 1$  so that  $c \geq \sin \phi$ , making the second term arbitrarily small. Hence we obtain

$$G \phi \rightarrow \phi + \epsilon, \quad (37)$$

which demonstrates that the transformation  $\phi \rightarrow \phi + \pi$  has no net effect on the azimuthal dynamics. On the other hand, our  $\theta$  dynamics become

$$\begin{aligned} G \theta &\rightarrow \theta + \cos(\phi + \pi) \frac{4}{\sqrt{N}} + O(1/N) \\ &= \theta - \cos \phi \frac{4}{\sqrt{N}} + O(1/N). \end{aligned} \quad (38)$$

Equation (38) shows that the transformation  $\phi \rightarrow \phi + \pi$  causes the polar angle  $\theta$  to be rotated in the opposite direction, as a consequence of the change in sign of the  $\cos \phi$

factor. It follows that once  $\phi$  has reached a uniform distribution, there will be an equal probability of rotating  $\theta$  by  $4/\sqrt{N}$  in the positive and negative directions. This behavior is valid over the entire region of the Bloch sphere specified by our initial condition  $\pi/2 \geq \theta_0 \geq c/\epsilon_{rms}^2 \sqrt{N}$ . As a result, the  $\theta$  dynamics over this entire region no longer resemble a deterministic rotation towards  $\theta = \pi$ , but instead resemble a random walk on  $\theta$  with step size of approximately  $4/\sqrt{N}$ . Figure 1 shows a graphical representation of these dynamics.

Using the central limit theorem, we find that

$$\theta_{rms} \propto \sqrt{t/N}, \quad (39)$$

where  $t$  is the number of iterations after the uniform distribution in  $\phi$  has been attained, i.e., after the first  $T$  iterations. Consequently, to “move” the polar angle of the probability distribution by some small angle  $\Delta \theta$  now takes time proportional to  $\Delta \theta^2 N$ . From Eq. (20), the corresponding change in the probability of a successful measurement of the target state is

$$\Delta P \approx \frac{1}{2} \sin \Delta \theta \Delta \theta \approx \frac{1}{2} \Delta \theta^2 \propto \Delta t/N. \quad (40)$$

Thus, once we have entered the region of the Bloch sphere characterized by  $\pi/2 \geq \theta \geq c/\epsilon_{rms}^2 \sqrt{N}$ , the probability of a successful measurement further increases only at the rate  $t/N$ . This is the same as the classical result and consequently, we can attain no further speed-up over the classical algorithm once we have entered this regime. The polar angle  $O(1/\epsilon_{rms}^2 \sqrt{N})$  thus constitutes an upper bound on the rotation that can be achieved in subclassical time. Hence, by the arguments given above, our bound  $O(N^{1-2\delta})$  on the complexity of the noisy quantum search algorithm is a tight bound.

### E. Summary of discrete time noise Grover search

Using the above results, we can now give an accurate phenomenological description of the dynamics of the noisy search algorithm. We identify the critical time scale of the algorithm as  $t_{mixing} = 1/\epsilon_{rms}^2$ , which is the mixing time of the  $z$  rotation induced by the random error. For times  $t \ll t_{mixing}$ , the algorithm proceeds more or less deterministically, and the initial state is rotated towards the target state  $|1\rangle$ , attaining a polar angle of  $\theta = 1/\epsilon_{rms}^2 \sqrt{N}$ . However, for times  $t \gg t_{mixing}$ , the  $\phi$  variable becomes completely mixed and the  $\theta$  rotation, which is the driving force of the search process, has an equal probability of increasing or decreasing the polar angle of the probability distribution. As a result, the  $\theta$  dynamics also become those of a random walk, and the searching rate becomes classical. Therefore, in order to gain the maximum speed-up over the classical algorithm, the best methodology we can employ is to run the algorithm for  $t \ll t_{mixing}$ , measure, and then repeat this process to boost our overall success probability. This protocol will allow the subclassical scaling to be retained.

The effects of the scaling of  $\epsilon_{rms}$  on the complexity of the search algorithm are seen to be different according to the value of the error scaling index  $\delta$ . We saw that if  $\delta > 1/4$ , the

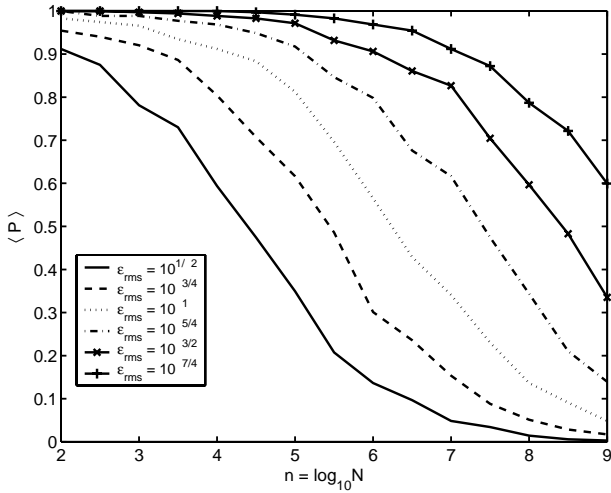


FIG. 2. Results of numerical simulations of the discrete-time implementation of Grover's algorithm with a noisy oracle. The oracle noise is determined by the probability distribution  $p(\epsilon)$  (Sec. II C).  $\langle P \rangle$  is the average success rate of 100 trials and  $N$  is the library size. The curves correspond to calculations with error magnitudes of  $\epsilon_{rms} = 10^{-0.5}, 10^{-0.75}, 10^{-1}, 10^{-1.25}, 10^{-1.5}, 10^{-1.75}$ , where the labeling goes from left to right.

search algorithm was essentially unaffected by the presence of error and the complexity is identical to the noiseless quantum search result of  $O(\sqrt{N})$ . On the other hand, if  $\delta \leq 1/4$ , then the optimal speed-up is obtained by running the algorithm for  $t = O(1/\epsilon_{rms}^2)$  iterations, achieving a maximum polar angle of  $\theta = O(1/\epsilon_{rms}^2 \sqrt{N})$  and then measuring. Expanding Eq. (20) yields the measurement success rate  $P(t) = O(1/\epsilon_{rms}^4 N)$  and a resulting algorithmic complexity of  $O(t/P(t)) = O(\epsilon_{rms}^2 N) = O(N^{1-2\delta})$ . So the optimal error scaling is given by  $\delta \geq 1/4$ , i.e., the errors scale as  $O(N^{-1/4})$ . In contrast, the most physically realistic constant error scaling  $\delta = 0$ , corresponding to a constant error over a range of  $N$  values, yields no speed-up over the classical search.

It is interesting to compare these results to the analytic results for a constant phase error oracle given in Ref. [7]. In the constant phase error model, the oracle applies the same phase  $\exp[i(\pi + \epsilon)]$  to the marked state at each iteration. It can be shown that for a constant phase error of magnitude  $\epsilon$ , the error must scale as  $\epsilon = O(N^{-1/2})$  in order to obtain a quadratic speed-up over the classical search. The differences in error accumulation between the constant and random phase error processes can be compared (for example) to the ballistic and diffusive regimes of Brownian motion, respectively. Constant phase errors tend to accumulate quickly since they are all in the same "direction." On the other hand, subsequent random phase errors can cancel each other out and thus accumulate more slowly.

Let us now consider a question of physical importance. Given an oracle with error magnitude  $\epsilon_{rms}$  and a library of size  $N$ , let us assume that we can attain some constant success rate  $P$  after  $T$  iterations. Given a larger library of size  $N' = kN$ , we wish to obtain the same success rate  $P$  after  $T' = \sqrt{k}T$  iterations. In other words, we wish to obtain the ideal quantum search complexity of  $O(\sqrt{N'})$  also for a larger

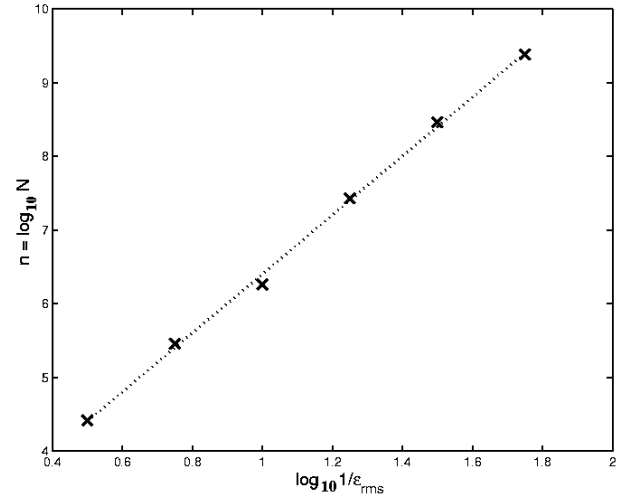


FIG. 3. Oracle error magnitude  $\epsilon_{rms}$  for given library size  $N$ , which yields an average success rate of  $P = 1/2$  for the discrete-time implementation of Grover's algorithm with a noisy oracle. The slope of the best-fit line is 4.002, which corresponds to  $\epsilon_{rms} \propto N^{-1/4}$  and an error scaling parameter  $\delta = 1/4$ .

library size. What is the maximum allowable oracle error  $\epsilon'_{rms}$  for the larger library? The answer to this question follows immediately from our previous analysis. We know that  $\epsilon_{rms}$  can scale as  $N^{-\delta}$  with  $\delta \geq 1/4$  without affecting algorithmic complexity. Thus the maximum allowable error in the oracle for the larger library is  $\epsilon'_{rms} = k^{-1/4} \epsilon_{rms}$ . As an example, if the library size is doubled, the oracle error for the larger library is required to be at most 0.84 the corresponding error for the smaller library.

To verify the validity of our analytic results, we can simulate the effects of a noisy oracle numerically. Given a library of size  $N$ , we simulated Grover's search algorithm using the procedure described in Sec. II C. For each iteration of the search algorithm, a phase error  $\epsilon$  was selected from a zero-mean Gaussian distribution with standard deviation  $\epsilon_{rms}$ . The algorithm was run for  $t = \pi\sqrt{N}/4$  iterations and the maximal probability of success attained was recorded. This process was repeated 100 times and the average success probability was calculated. Figure 2 plots the average probability of success,  $\langle P \rangle$ , versus the library size,  $n = \log_{10} N$ , for various values of  $\epsilon_{rms}$ . Figure 3 shows the value of  $\log_{10} \epsilon_{rms}$  that yields a constant success probability of  $P = 1/2$  for a given library size. As predicted by our analysis above, we find that in order to obtain a constant success rate as a function of  $n$  (or  $N$ ), the error must scale as  $\epsilon_{rms} \propto N^{-1/4}$ .

### III. EFFECT OF NOISE ON CONTINUOUS TIME ANALOG OF QUANTUM SEARCH

In this section, we analyze the effect of noise on a slightly different model for quantum search that has been proposed by Farhi and Gutmann [12]. In this model, one again starts in the symmetric superposition of all states,  $|\eta\rangle$ , and then applies the following Hamiltonian:

$$H_0 = |\eta\rangle\langle\eta| + |\tau\rangle\langle\tau| = H_\eta + H_\tau, \quad (41)$$

where  $|\tau\rangle$  is the marked state. Note that this Hamiltonian is directly related to Grover's iterate, Eqs. (2) and (3). Action of Eq. (2) amounts to simply applying the Hamiltonian  $H_\eta$  for a time  $\pi$ , followed by applying the Hamiltonian  $H_\tau$  for a time  $\pi$ . Clearly, the two operators, Eqs. (2) and (41), would be equivalent if  $H_\eta$  and  $H_\tau$  commuted. However,  $[H_\eta, H_\tau] = (1/\sqrt{N})(|\eta\rangle\langle\tau| + |\tau\rangle\langle\eta|)$ . Thus the two methods become similar for large  $N$  and are formally equivalent as  $N$  approaches infinity.

Farhi and Gutmann calculated the time evolution of the system when one starts in the equal superposition state  $|\eta\rangle$ . They found that

$$e^{-iHt}|\eta\rangle = e^{-it} \left\{ \left[ \frac{1}{\sqrt{N}} \cos\left(\frac{t}{\sqrt{N}}\right) - \sin\left(\frac{t}{\sqrt{N}}\right) \right] |\tau\rangle + \sqrt{\left(1 - \frac{1}{N}\right)} \cos\left(\frac{t}{\sqrt{N}}\right) |\eta\rangle \right\}. \quad (42)$$

They noticed that at time  $t = \pi\sqrt{N}/2$ , the initial state  $|\eta\rangle$  has evolved to the marked state  $|\tau\rangle = |1\rangle$ . The time required to evolve to the marked state scales as  $O(\sqrt{N})$ , matching the complexity of Grover's algorithm with respect to an oracle. As a result, we take the time that it takes to reach the marked state as a function of  $N$  to be the measure of the complexity of a continuous time algorithm.

As noted above, in the limit of large  $N$  the continuous time and discrete Grover's algorithm are formally equivalent. Therefore, it seems useful to also evaluate the effect of a fluctuating "oracle" in the continuous time picture. The continuous time Hamiltonian that models a discrete quantum search noisy oracle with phase fluctuation  $\epsilon$  is given by

$$H = H_\eta + (1 + \xi)H_\tau = |\eta\rangle\langle\eta| + (1 + \xi)|\tau\rangle\langle\tau|. \quad (43)$$

Here  $\xi$  is a time-dependent, Markovian stochastic variable that satisfies  $\int_0^\pi \xi dt = \epsilon$ . We shall assume that  $\epsilon$  fluctuates and that it can be described by a Gaussian distribution.

In order to evaluate the Hamiltonian and its effect on the initial state, it is simpler to transform to our orthonormal two-state bases  $|1\rangle$  and  $|2\rangle$  defined by Eqs. (4) and (5). One can then decompose the transformed Hamiltonian into the corresponding spin operators to find that

$$H = \left(1 + \frac{\xi}{2}\right) \mathbf{I} + \left(\frac{1}{N} + \frac{\xi}{2}\right) \sigma_z + \frac{1}{\sqrt{N}} \sqrt{\left(1 - \frac{1}{N}\right)} \sigma_x. \quad (44)$$

When the fluctuations in the time evolution operator  $e^{-iHt}$  are now considered, i.e., the resulting phase  $\epsilon \neq 0$ , it is useful to approach the problem by examining the evolution of the density matrix. Consider an initial density matrix

$$\rho(0) = \begin{bmatrix} a & b \\ b^* & d \end{bmatrix}. \quad (45)$$

It is well established that a fluctuating  $\sigma_z$  component of the Hamiltonian leads to dephasing. For example, applying the fluctuating perturbation  $(\xi/2)\sigma_z$  for a time  $\pi$  yields

$$\rho(t) = \exp(-i\sigma_z\pi/2)\rho(0)\exp(i\sigma_z\pi/2). \quad (46)$$

Subsequently averaging over all possible values of the resulting total phase  $\epsilon$  leads to the following evolution:

$$\int \frac{1}{\sqrt{2\pi s}} e^{-\epsilon^2/2s^2} e^{-i\epsilon\sigma_z/2} \begin{bmatrix} a & b \\ b^* & d \end{bmatrix} e^{i\epsilon^2/\sigma_z} d\epsilon = \begin{bmatrix} a & be^{-s^2/2} \\ b^*e^{-s^2/2} & d \end{bmatrix}. \quad (47)$$

Here  $s^2$  is the variance of the fluctuating phase  $\epsilon$ . Since it has been assumed that the fluctuating field  $\xi$  is Markovian, one can treat this as a dephasing term described by a decay constant  $\Gamma = s^2/2\pi$  and then use the corresponding Bloch equations [22] to calculate the evolution of the system. This Markovian approximation is valid when the fluctuation of  $\xi$  is much faster than  $1/\sqrt{N}$ . To analyze the system evolution, we therefore decompose our density matrix using the Bloch representation,

$$\rho = \frac{1}{2} \mathbf{I} + \frac{1}{2} \hat{n} \cdot \sigma. \quad (48)$$

The time evolution is calculated by solving the Bloch equations for the components of  $\hat{n}$  [22]:

$$\dot{n}_x = \frac{2}{N} n_y - \Gamma n_x, \quad (49)$$

$$\dot{n}_y = \frac{2}{\sqrt{N}} \sqrt{\left(1 - \frac{1}{N}\right)} n_z - \frac{2}{N} n_x - \Gamma n_y, \quad (50)$$

$$\dot{n}_z = -\frac{2}{\sqrt{N}} \sqrt{\left(1 - \frac{1}{N}\right)} n_y. \quad (51)$$

In order to understand the effect of the dephasing on the algorithmic complexity, we examine these equations in the limit of large  $N$  and keep only terms that are of order  $1/\sqrt{N}$ . This yields only two coupled equations,  $n_y$  and  $n_z$ :

$$\dot{n}_y = \frac{2}{\sqrt{N}} n_z - \Gamma n_y, \quad (52)$$

$$\dot{n}_z = -\frac{2}{\sqrt{N}} n_y. \quad (53)$$

Our initial density matrix is given by

$$\rho = |\eta\rangle\langle\eta| = \frac{1}{2} \mathbf{I} + \frac{1}{2} \left(-1 + \frac{2}{N}\right) \sigma_z + \frac{1}{2} \left(\frac{2\sqrt{N-1}}{N}\right) \sigma_x. \quad (54)$$

We note that the quantity  $n_z$  is the projection onto the state  $|1\rangle$  and thus provides a measure of how well the computation is proceeding. Initially, we have  $n_z = -1 + (2/N)$ , while



our computation becomes complete when  $n_z = 1$ . Solving the differential equations in the large- $N$  limit, Eq. (53), with initial conditions  $n_y(0) = 0$ ,  $n_z(0) = -1 + (2/N)$ , yields the following solution:

$$n_y(t) = \frac{2\left(-1 + \frac{2}{N}\right)}{\sqrt{\Gamma^2 N - 16}} \left( \exp\left[\frac{t}{2}(-\sqrt{\Gamma^2} - \sqrt{\Gamma^2 - 16/N})\right] + \exp\left[\frac{t}{2}(-\sqrt{\Gamma^2} + \sqrt{\Gamma^2 - 16/N})\right] \right),$$

$$n_z(t) = \frac{\left(-1 + \frac{2}{N}\right)}{2\sqrt{\Gamma^2 N - 16}} \left( (-\sqrt{\Gamma^2 N} + \sqrt{\Gamma^2 N - 16}) \times \exp\left[\frac{t}{2}(-\sqrt{\Gamma^2} - \sqrt{\Gamma^2 - 16/N})\right] + (\sqrt{\Gamma^2 N} + \sqrt{\Gamma^2 N - 16}) \times \exp\left[\frac{t}{2}(-\sqrt{\Gamma^2} + \sqrt{\Gamma^2 - 16/N})\right] \right). \quad (55)$$

Ideally, we would like to find the time at which  $n_z(t) = 1$ , or equivalently, the time when the probability of reaching the marked state is unity,

$$P(t) = \langle \tau | \rho(t) | \tau \rangle = \frac{1}{2} [1 + n_z(t)] = 1. \quad (56)$$

When measuring the complexity of Grover's algorithm, however, we need only to find the time required such that the probability  $P(t)$  of being in the marked state  $\tau$  is greater than some constant. For concreteness, we choose here the minimum time satisfying  $P(t) = \langle \tau | \rho(t) | \tau \rangle \geq 1/4$ .

To determine this time explicitly, we calculate  $P(t)$  as a function of various values of  $\Gamma$  in the limit of large  $N$ . We observe that in Eq. (55) the term  $\sqrt{\Gamma^2 - 16/N}$  is imaginary when  $\Gamma < 4/\sqrt{N}$ . Consequently, we expect that the behavior will be drastically different for the two regimes (a)  $\Gamma < 4/\sqrt{N}$  and (b)  $\Gamma > 4/\sqrt{N}$ . In order to make a direct comparison between the continuous-time behavior and the discrete-time results, we choose the dephasing constant  $\Gamma$  to scale with  $N$  in the same way as in Sec. II, namely,

$$\Gamma = \alpha N^{-2\delta\alpha} \epsilon_{rms}^2, \quad (57)$$

where  $\alpha$  is the proportionality constant. This provides the contact point of the continuous-time search algorithm with the discrete-time algorithm of Sec. II. We then see that regimes (a) and (b) correspond to the two regimes already established in Sec. II, i.e., in region (a),  $\delta \geq 1/4$ , and in region (b),  $\delta \leq 1/4$ . We note that our regions (b) and (a) both include  $\delta = 1/4$ . The boundary line between the regions occurs when  $\delta = 1/4$  and  $\alpha = 4$ .

For  $\Gamma = \alpha N^{-2\delta\alpha} < 4/\sqrt{N}$ , i.e., in region (a), we calculate from Eq. (55) that

$$P(t) = \frac{1}{2} + \frac{1}{2} \left( -1 + \frac{2}{N} \right) e^{t/2\Gamma} \left( \cos(\sqrt{16/N - \Gamma^2} t/2) + i \sqrt{\frac{\Gamma^2 N}{\Gamma^2 N - 16}} \sin(\sqrt{16/N - \Gamma^2} t/2) \right). \quad (58)$$

We now pick a time

$$t' = \frac{2\pi}{\sqrt{16/N - \Gamma^2}} \quad (59)$$

and find that

$$P(t') = \frac{1}{2} + \frac{1}{2} \left( 1 - \frac{2}{N} \right) \exp\left( -\frac{\pi\Gamma}{\sqrt{16/N - \Gamma^2}} \right). \quad (60)$$

Hence,

$$P(t') > \frac{1}{2} - O(1/N). \quad (61)$$

Since this is larger than the value corresponding to our definition of minimum time, it implies that the complexity of the search algorithm is bounded from above by  $O(t')$ . Inspection of Eq. (59) shows that  $t'$  is an increasing function of  $\Gamma$ . Therefore, in order to evaluate an upper bound for  $t'$  in the regime (a) where  $\delta \geq 1/4$ , we need to evaluate  $t'$  for the largest possible value of  $\Gamma$ . The largest value of  $\Gamma$  in this regime lies on the boundary with regime (b), namely, where  $\delta = 1/4$  and  $\alpha = 4$ . Hence, we evaluate  $t'$  as one asymptotically approaches the boundary between regions (a) and (b), for a given error scaling  $\delta$ . We choose  $\delta = 1/4$  and  $\alpha = 4 - m$ , where  $m$  is a small constant greater than zero. Substituting in Eq. (60) yields  $t' = 2\pi\sqrt{N}/m$ . Thus the minimum time will be on the order of  $O(\sqrt{N})$ , corresponding to an upper bound on the algorithmic complexity of  $O(\sqrt{N})$ . Hence for  $\delta \geq 1/4$ , in regime (a), the continuous-time search algorithm achieves its maximal algorithmic quantum speed-up. This is in agreement with the results from Sec. II, which showed that the discrete-time quantum search algorithm yields the maximum algorithmic speed-up when the root-mean-square error  $\epsilon_{rms}$  is smaller than  $N^{-1/4}$ , corresponding to  $\delta \geq 1/4$ .

To complete the continuous-time analysis, we solve for  $P(t)$  in the regime (b) where  $\delta \leq 1/4$ . Here evaluation of  $n_z$  in the limit of large  $N$  leads to

$$P(t) = \frac{1}{2} - \frac{1}{2} e^{-4t/N\Gamma}. \quad (62)$$

Setting  $P(t') = 1/4$  and solving for  $t'$  yields

$$t' = \frac{N\Gamma \ln(2)}{4} \quad (63)$$

$$= O(N^{1-2\delta}). \quad (64)$$

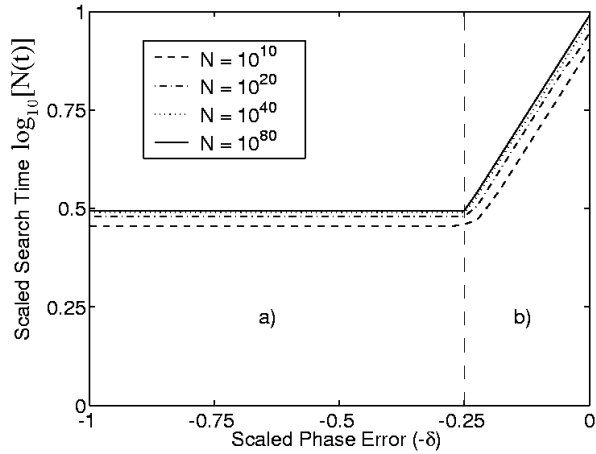


FIG. 4. Search time  $t'$  for the continuous-time search algorithm, shown as a function of the oracle phase error scaling parameter  $-\delta$ . Here  $t'$  is defined as the time it takes to achieve a success probability of  $P(t')=1/4$ . The oracle phase error is measured here by its size scaling parameter  $\delta$ , where  $\epsilon_{rms}=N^{-\delta}$  (see text). The search time shows a distinct transition between two regimes (a) and (b). In regime (a) the continuous-time algorithm for a database of size  $N$  matches the Grover bound of  $O(N^{1/2})$  for large  $N$ , i.e.,  $\log_N(t')=0.5$ . In regime (b) the quantum search speed-up is gradually lost as  $\delta$  decreases from the critical value  $1/4$  to  $0$ . At  $\delta=0$  we have constant error, independent of the database size, and the search time has now increased to equal the classical bound  $O(N)$ , i.e.,  $\log_N(t')=1$ . In the limit of large  $N$ , the scaling of the search time with the error parameter for  $\delta \geq 1/4$  [region (a)] is a constant,  $t' = O(N^{1/2})$ , while for  $\delta \leq 1/4$  [region (b)], it is  $t' = O(N^{1-2\delta})$  (see text).

We have verified our conclusions by numerically simulating Eq. (55) for various values of  $N$  and  $\Gamma$ . Figure 4 shows a logarithmic plot of the minimum time  $t'$  required to obtain a success probability of  $P(t')=1/4$ , as a function of the error scaling parameter  $\delta$ , for various values of  $N$ . It is evident that the algorithmic complexity shows a marked transition at  $\delta=1/4$ , from scaling  $O(N^{1/2})$  for  $\delta \geq 1/4$ , to  $O(N^{1-2\delta})$  for  $\delta \leq 1/4$ , as predicted by Eqs. (59) and (64), respectively. Equivalently, we can state that the continuous-time quantum search algorithm with randomized phase error achieves minimal complexity and hence maximum algorithmic speed-up when  $\epsilon_{rms} \leq N^{1/4}$  ( $\delta \geq 1/4$ ). This agrees with the results for the discrete-time algorithm.

#### IV. CONCLUSIONS

The analysis in this paper has provided a phenomenological description of the dependence of the algorithmic complexity of Grover's algorithm on the scaling of oracle phase error for a discrete quantum search, and on stochastic Hamiltonian errors leading to phase error in a continuous-time quantum search algorithm. In both the discrete- and continuous-time versions of the algorithm, it was found that if the phase error scaled with size as  $N^{-\delta}$ , then for  $\delta \leq 1/4$  the effect on the complexity of the algorithm for large  $N$  was negligible. However, if the size scaling of the error lies in the

regime  $\delta \leq 1/4$ , then for large  $N$  it was determined that there is tight bound of  $O(N^{1-2\delta})$  on the complexity of the search algorithm. In particular, this implies that in the presence of any constant (nonzero) amount of phase error in the oracle ( $\delta=0$ ), there exists some library size  $N_{max}$  above which the quantum search algorithm no longer provides a quadratic speed-up. In this case, for databases of size  $N > N_{max}$ , the search time is  $\Omega(N)$ , which is equivalent to the classical result and there is therefore no quantum speed-up. Intermediate error scaling,  $0 < \delta < 1/4$ , provides speed-up intermediate between the classical and quantum limits, respectively. These results hold for both the discrete-time and continuous-time quantum search algorithms, and assume very little about the specific form of the underlying error processes.

The complexity analysis we have made here is also important for determining the precision needed in scaling up a quantum search. For instance, let us assume that we are able to implement a quantum search for a library of size  $N$  with an oracle error of magnitude  $\epsilon_{rms}$ . Then to perform a quantum search on a library of size  $kN$  with equivalent accuracy, our results imply that we need to implement an oracle with an error of at most  $\epsilon_{rms}/k^{1/4}$ . Since this must lie in the regime  $\delta \geq 1/4$ , physically, this requires a system where the phase error decreases as a function of database size. Consequently, the precision must increase exponentially as a function of the number of qubits, putting severe demands on the physical realization. In contrast, if the error and hence the precision is constant in the system size, (e.g., a system where the natural linewidth is independent of the number of states), then there always exists a database size such that the quantum approach offers no speed-up over the classical search algorithm.

The main consequence of the nonrobustness of these forms of quantum searches to oracle noise that was demonstrated here is a practical limitation on the size of the library on which may be searched with a quadratic speed-up using quantum search algorithms without any explicit error correction. This result has significant consequences for physical implementation of quantum search algorithms, since although quantum error correction can be used to reduce the error present in the oracle, such error correction procedures can require significant resources [2]. In practice, it will therefore be necessary to balance the cost of error correction (in both spatial and temporal resources) with the extent of speed-up attained by a noisy quantum search. The analytic results presented in this paper provide a useful bound on the maximum oracle error permissible if a quadratic speed-up is to be retained. Above this maximum allowable error, we must use error correction in any physical implementation. Conversely, below this maximum error, we can be confident that error correction will not be necessary, provided that oracle phase error is the primary source of error.

As a final comment we point out that although Grover's algorithm and its continuous-time analog are not inherently robust to phase error in the oracle, it is not clear whether other implementations of quantum search may be inherently robust. Exploration of both active and passive error correc-

tion schemes for Grover's algorithm will therefore be a valuable direction for future work.

#### ACKNOWLEDGMENTS

N.S. acknowledges the University of California, Berkeley. The work of K.R.B. was supported by the Fannie and John

Hertz Foundation. K.B.W. acknowledges the Miller Foundation for Basic Research. This effort was sponsored by the Defense Advanced Research Projects Agency (DARPA) and the Air Force Laboratory, Air Force Material Command, USAF, under Grant No. F30602-01-2-0524. We also thank NSF ITR/SY Grant No. 0121555.

- 
- [1] P.W. Shor, Phys. Rev. A **52**, R2493 (1995).
  - [2] J. Preskill, Proc. R. Soc. London, Ser. A **454**, 385 (1998).
  - [3] D. Bacon, J. Kempe, D.A. Lidar, and K.B. Whaley, Phys. Rev. Lett. **85**, 1758 (2000).
  - [4] D. Abrams and C. Williams, e-print quant-ph/9908083.
  - [5] E. Novak, J. Complex. **17**, 2 (2001).
  - [6] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, e-print quant-ph/0005055.
  - [7] G.L. Long, Y.S. Li, W.L. Zhang, and C.C. Tu, Phys. Rev. A **61**, 042305 (1999).
  - [8] J. Hsieh, C.-M. Li, and D.-S. Chuu, e-print quant-ph/0209073.
  - [9] B. Pablo-Norman and M. Ruiz-Altaba, Phys. Rev. A **61**, 012301 (1999).
  - [10] D. Shapira, S. Mozes, and O. Biham, Phys. Rev. A **67**, 042301 (2003).
  - [11] J. Bae and Y. Kwon, e-print quant-ph/0211186.
  - [12] E. Farhi and S. Gutmann, Phys. Rev. A **57**, 2403 (1998).
  - [13] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, e-print quant-ph/0001106.
  - [14] L.K. Grover, in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (ACM, New York, 1996), p. 212.
  - [15] L.K. Grover, Phys. Rev. Lett. **79**, 4709 (1997).
  - [16] C.H. Bennett, E. Bernstein, G. Brassard, and U.V. Vazirani, SIAM J. Comput. **26**, 1510 (1997).
  - [17] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, Fortsch. Phys.-Prog. Phys. **46**, 493 (1998).
  - [18] D.E. Knuth, *The Art of Computer Programming* (Addison-Wesley, Reading, MA, 1973), Vol. 1.
  - [19] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
  - [20] A. Messiah, *Quantum Mechanics* (Interscience, New York, 1961).
  - [21] It should be noted that this analysis applies only in the limit of large  $N$ , which is what we mean when we talk of algorithmic complexity in the first place.
  - [22] K. Blum, *Density Matrix Theory and Applications* (Plenum Press, New York, 1981).