Efficient state preparation for a register of quantum bits

Andrei N. Soklakov and Rüdiger Schack
Department of Mathematics, Royal Holloway, University of London, Egham, Surrey TW20 0EX, United Kingdom
(Received 16 March 2005; published 9 January 2006)

We describe a quantum algorithm to prepare an arbitrary pure state of a register of a quantum computer with fidelity arbitrarily close to 1. Our algorithm is based on Grover's quantum search algorithm. For sequences of states with suitably bounded amplitudes, the algorithm requires resources that are polynomial in the number of qubits. Such sequences of states occur naturally in the problem of encoding a classical probability distribution in a quantum register.

DOI: 10.1103/PhysRevA.73.012307 PACS number(s): 03.67.Lx, 03.67.Mn

I. INTRODUCTION

In many applications of quantum computers, a quantum register, composed of a fixed number of qubits, is initially prepared in some simple standard state. This initial preparation step is followed by a sequence of quantum gate operations and measurements. There are applications of quantum computers, however, notably the task of simulating the dynamics of a physical system [1–3], that may require the initialization of a quantum register in a more general state, corresponding to the initial physical state of the simulated system. This leads naturally to the question of what quantum states can be efficiently prepared on a quantum register?

The memory of a classical computer can be easily put into any state by writing an arbitrary bit string into it. The situation for quantum computers is very different. The Hilbert space associated with a quantum register composed of as few as 100 quantum bits (qubits) is so large that it is impossible to give a classical description of a generic state vector, i.e., to list the 2^{100} complex coefficients defining it. In this sense it can be said that arbitrary pure states cannot be prepared [4].

It is, nevertheless, possible to formulate the problem of arbitrary state preparation for a register of qubits in a meaningful way. This is achieved by starting from the assumption that the state is initially defined by a set of quantum *oracles*. By assuming that the state is given in this form, we shift the focus from the problem of describing the state to the problem of the computational resources needed to actually prepare it. In other words, we address the computational complexity rather than the algorithmic complexity of the state. For the purpose of quantifying these computational resources, we simply count the number of oracle calls. We are, thereby, deliberately ignoring the internal structure of the oracles and the computational resources needed in each oracle call. The algorithm we describe here is applicable to any set of oracles, i.e., to any state. We will show that it is *efficient* for a large class of states of particular interest for the simulation of physical systems.

Let N be a positive integer. We will describe a quantum algorithm for preparing a $\lceil \log_2 N \rceil$-qubit quantum register in an approximation to the state

$$|\Psi\rangle = \sum_{x=0}^{N-1} \sqrt{p(x)} e^{2\pi i \phi(x)} |x\rangle \tag{1}$$

for arbitrary probabilities $p(x)$ and arbitrary phases $\phi(x)$. Here, and throughout the paper, $|0\rangle, |1\rangle, \ldots$ denote computational basis states. More precisely, given any small positive numbers λ and ν, our algorithm prepares the quantum register in a state $|\tilde{\Psi}\rangle$ such that, with probability greater than $1-\nu$, the fidelity obeys the bound

$$|\langle \tilde{\Psi} | \Psi \rangle| > 1 - \lambda. \tag{2}$$

To define the algorithm and to assess its efficiency for large N, we need to specify in which form the coefficients $p(x)$ and $\phi(x)$ are given. We assume that we are given classical algorithms to compute the functions $p(x)$ and $\phi(x)$ for any x. These classical algorithms are used to construct a set of quantum *oracles*. We will quantify the resources needed by our state-preparation algorithm in terms of (i) the number of oracle calls, (ii) the number of additional gate operations, and (iii) the number of auxiliary qubits needed in addition to the $\lceil \log_2 N \rceil$ register qubits.

To analyze the asymptotic, large N, behavior of our algorithm, we consider a sequence of probability functions $p_N: \{0, \ldots, N-1\} \to [0,1]$, $\Sigma_x p_N(x) = 1$, and a sequence of phase functions $\phi_N: \{0, \ldots, N-1\} \to [0,1]$, where $N = 1, 2, \ldots$. For any N, the algorithm prepares the quantum register in a state $|\tilde{\Psi}\rangle$ such that, with probability greater than $1-\nu$, the fidelity obeys the bound (2), where in the definition (1) of $|\Psi\rangle$ the functions p and ϕ are replaced by p_N and ϕ_N, respectively. Under the assumption that there exists a real number η, $0 < \eta < 1$, such that

$$p_N(x) \leqslant \frac{1}{\eta N} \quad \text{for all } N \text{ and } x, \tag{3}$$

we show that the resources needed by our state-preparation algorithm are polynomial in the number of qubits, $\log_2 N$, and the inverse parameters η^{-1}, λ^{-1}, and ν^{-1}.

An obvious example of a sequence of functions that do not satisfy the bound (3) and for which the resources required for state-preparation scale exponentially with the number of qubits is given by $p_N(x) = \delta_{xy}$ for some integer $y = y(N)$. In this case, it follows from the optimality of Grov-

er's algorithm [5,6] that the number of oracle calls needed is proportional to $\sqrt{N}$.

Sequences that do satisfy the bound (3) arise naturally in the problem of encoding a bounded probability density function f: $[0,1]\to[0,f_{\max}]$ in a state of the form

$$|\Psi_f\rangle = \mathcal{N}^{-1}\sum_{x=0}^{N-1}\sqrt{f(x/N)}x, \tag{4}$$

where $\mathcal{N}$ is a normalization factor. Grover and Rudolph have given an efficient algorithm for this problem if the function f is efficiently integrable [7]. Essentially the same algorithm was found independently by Kaye and Mosca [8], who also mention that phase factors can be introduced using the methods discussed in Ref. [9]. Recently, Rudolph [10] has found a simple nondeterministic state-preparation algorithm that is efficient for all sequences satisfying the bound (3).

For general sequences of states satisfying the bound (3), a given value of the fidelity bound λ, and assuming polynomial resources for the oracles, our algorithm is exponentially more efficient than the algorithm proposed by Ventura and Martinez [11] and later related proposals [12,13], for which the resources needed to grow like $N\log_2 N$. The use of Grover's algorithm for state preparation has been suggested by Zeng and Kuang for a special class of coherent states in the context of ion-trap quantum computers [14]. A general analysis of the state-preparation problem in the context of adiabatic quantum computation was given by Aharonov and Ta-Shma [15].

This paper is organized as follows. In Sec. II we give a full description of our algorithm. A detailed derivation is deferred to Sec. III. The algorithm depends on a number of parameters that can be freely chosen. In Sec. IV we consider a particular choice for these parameters and show that it guarantees the fidelity bound (2). We use the same choice of parameters in Sec. V to derive the worst case bounds on the time and the memory resources required by the algorithm. In Sec. VI we conclude with a brief summary.

II. ALGORITHM

Our algorithm consists of two main stages. In the first stage, the algorithm prepares the register in an approximation to the state

$$|\Psi_p\rangle = \sum_{x=0}^{N-1}\sqrt{p(x)}|x\rangle, \tag{5}$$

which differs from $|\Psi\rangle$ only in the phases $\phi(x)$. More precisely, let ϵ be the largest small parameter such that

$$\epsilon < \lambda\eta/3 \tag{6}$$

and $1/\epsilon$ is an integer. The first stage of the algorithm prepares the register in a state

$$|\Psi_{\tilde{p}}\rangle = \sum_{x=0}^{N-1}\sqrt{\tilde{p}(x)}|x\rangle, \tag{7}$$

such that, with probability greater than $1-\nu$, we have that

$$|\langle\Psi_{\tilde{p}}|\Psi_p\rangle| > 1-\lambda. \tag{8}$$

We will describe the details of the first stage below.

The second stage of the algorithm adds the phases $\phi(x)$ to the state $|\Psi_{\tilde{p}}\rangle$ resulting from the first stage. This can be done in a straightforward way as follows. We start by choosing a small parameter ϵ' such that $1/\epsilon'$ is a positive integer. We then define a list of unitary operations, $U_1,\ldots,U_{1/\epsilon'}$, on our quantum register by

$$U_k|x\rangle = \begin{cases} e^{2\pi i\epsilon'}|x\rangle & \text{if } \phi(x) > \left(k-\frac{1}{2}\right)\epsilon', \\ |x\rangle & \text{otherwise.}\end{cases} \tag{9}$$

The operators U_k are conditional phase shifts that can be realized as quantum gate sequences using the classical algorithm for computing the function $\phi(x)$ [9]. If we apply the operators U_k sequentially to the result of the first stage, we obtain

$$|\tilde{\Psi}\rangle = U_1U_2,\ldots,U_{1/\epsilon'}|\Psi_{\tilde{p}}\rangle = \sum_{x=0}^{N-1}\sqrt{\tilde{p}(x)}e^{2\pi i\tilde{\phi}(x)}|x\rangle, \tag{10}$$

where the function $\tilde{\phi}(x)$ satisfies the inequality

$$|\tilde{\phi}(x)-\phi(x)| \leqslant \epsilon'/2 \tag{11}$$

for all x. It can be shown (see Sec. IV D) that together with Eq. (8) this implies the bound

$$|\langle\tilde{\Psi}|\Psi\rangle| > 1-\lambda-\lambda', \tag{12}$$

where $\lambda' = \epsilon'^2/8$. Notice the slight abuse of the notation identifying the parameter λ in the inequality (2) with the sum $\lambda+\lambda'$ in the inequality (12).

We now proceed to a more detailed description of the first stage of the algorithm. From now on we assume that N is an integer power of 2. This can always be achieved by padding the function $p(x)$ with zeros. Given our choice of the parameter ϵ, Eq. (6), we define a list of *oracles*, $o_1,\ldots,o_{1/\epsilon}$, by

$$o_k(x) = \begin{cases} 1 & \text{if } \sqrt{p(x)} \geqslant (1-\epsilon k)/\sqrt{\eta N}, \\ 0 & \text{otherwise.}\end{cases} \tag{13}$$

We extend this definition beyond the domain of the function p by setting $o_k(x)=0$ for $x\geqslant N$. Using the classical algorithm to compute $p(x)$, one can construct quantum circuits implementing the unitary oracles

$$\hat{o}_k|x\rangle = (-1)^{o_k(x)}|x\rangle. \tag{14}$$

These circuits are efficient if the classical algorithm is efficient. The list of oracles o_k defines a new function $p'(x)$ via

$$\sqrt{p'(x)} = (1-\epsilon k)/\sqrt{\eta N} \quad \text{if } o_{k-1}(x)=0 \text{ and } o_k(x)=1, \tag{15}$$

where $o_0(x)=0$ by convention. The situation is illustrated in Fig. 1, where the x values have been permuted for clarity. Knowledge of this permutation is not required for our algorithm.

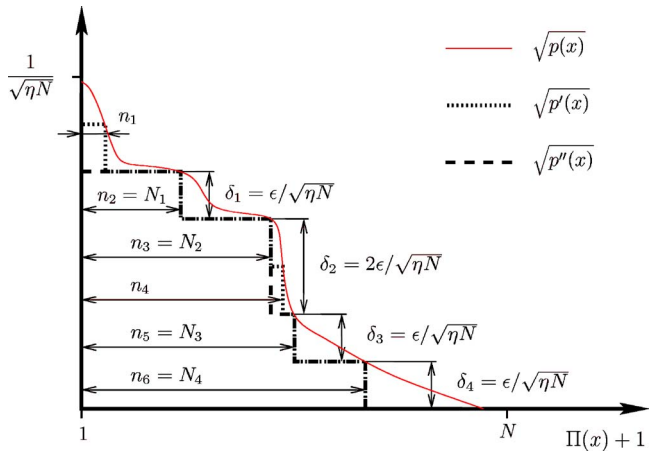

FIG. 1. (Color online) The solid line shows an example for a function $\sqrt{p(x)}$, sketched versus a permutation $\Pi(x)$, chosen so that $p(\Pi^{-1}(x)) \leqslant p(\Pi^{-1}(y))$ if $x > y$. The dotted and dashed lines show the corresponding functions $\sqrt{p'(x)}$ and $\sqrt{p''(x)}$ [see Eqs. (15) and (26)], which in this representation look like decreasing step functions. In this example, $1/\epsilon = 7$, $T = 4$, and it is assumed that $\tilde{n}_4 < \tilde{n}_3$ due to counting errors. Note that the function $p(x)$ is not required to decrease.

The essence of the first stage of the algorithm consists in using a number of Grover iterations based on the oracles $\hat{o}_k$ to prepare the register in an approximation to the state

$$|\Psi_{p'}\rangle = \mathcal{N}'^{-1} \sum_{x=0}^{N-1} \sqrt{p'(x)}|x\rangle, \tag{16}$$

where the normalization factor $\mathcal{N}'$ reflects the fact that $p'(x)$ may not be normalized. To find the number of required Grover iterations for each oracle $\hat{o}_k$, we need an estimate of the number of solutions, n_k, for each oracle, defined by

$$n_k = \sum_{x=0}^{N-1} o_k(x). \tag{17}$$

This estimate can be obtained from running the quantum counting algorithm [6] for each oracle $\hat{o}_k$. We denote the estimates obtained in this way by $\tilde{n}_k$. The accuracy of the estimate $\tilde{n}_k$ relative to n_k can be characterized by two real parameters, $0 < \nu < 1$ and $0 < \eta_c < \frac{1}{2}$, in such a way that, as a result of quantum counting, with probability greater than $1 - \nu$ we have

$$|\tilde{n}_k - n_k| < \eta_c N \quad \text{for } k = 1, \ldots, 1/\epsilon. \tag{18}$$

For each oracle $\hat{o}_k$ the resources needed to achieve the counting accuracy specified by η_c and ν depend on the actual number of solutions n_k. This dependence is important for optimizing the performance of our algorithm. In this paper, however, we present a simpler analysis assuming worst case conditions for each oracle $\hat{o}_k$. For this analysis we use a specific choice of η_c, which is given by Eq. (62).

The analysis of the algorithm is simplified if we concentrate on a subset of oracles,

$$O_k = o_{f_k} \quad \text{for } k = 1, \ldots, T, \tag{19}$$

where T and the indices $f_1, \ldots, f_T$ are determined by the construction below. We introduce a new parameter η_g [see Eq. (62) below] such that $\eta_c < \eta_g < \frac{1}{2}$. The index f_1 is defined to be the smallest integer such that

$$\tilde{n}_{f_1} \geqslant \eta_g N \quad \text{and} \quad \tilde{n}_j < \eta_g N \quad \text{for } j < f_1, \tag{20}$$

and, for $k \geqslant 2$, the index f_k is the smallest integer such that

$$\tilde{n}_{f_{k-1}} < \tilde{n}_{f_k} \quad \text{and} \quad f_{k-1} < f_k < 1/\epsilon. \tag{21}$$

The number T is the largest value of k for which these inequalities can be satisfied. The effect of Eq. (20) is to neglect narrow peaks [corresponding to small values of $\Pi(x)$ in Fig. 1]. Equation (21) makes sure that the numbers $\tilde{n}_{f_k}$ form an increasing sequence even if, due to counting errors, the estimates $\tilde{n}_k$ do not (see Fig. 1).

For $k = 1, \ldots, T$, define $\tilde{N}_k = \tilde{n}_{f_k}$,

$$N_k = \sum_{x=0}^{N-1} O_k(x), \tag{22}$$

and

$$\delta_k = \epsilon(f_{k+1} - f_k)/\sqrt{\eta N}, \tag{23}$$

where we define $f_{T+1} = 1/\epsilon$. For every oracle O_k, the value of $\tilde{N}_k$ is an approximation to the number of solutions, N_k, satisfying the bound

$$|\tilde{N}_k - N_k| = |\tilde{n}_{f_k} - n_{f_k}| < \eta_c N. \tag{24}$$

In what follows it will be convenient to introduce the notation

$$B_{s,k} = \sum_{j=s}^{k} \delta_j. \tag{25}$$

The oracles O_k define a new function $p''(x)$ via

$$\sqrt{p''(x)} = B_{k,T} \quad \text{if } O_{k-1}(x) = 0 \text{ and } O_k(x) = 1, \tag{26}$$

where $O_0(x) = 0$ by convention. The function $\sqrt{p''(\Pi^{-1}(x))}$ is a decreasing step function, with step sizes $\delta_1, \ldots, \delta_T$ which are multiples of $\epsilon/\sqrt{\eta N}$. The widths of the steps are given by the numbers N_k which are determined by the oracles (see Fig. 1).

The algorithm can now be completely described as follows. Choose a suitable (small) number, a, of *auxiliary qubits* [see Eq. (62) below], and define $L = \log_2 N + a$. For $k = 1, \ldots, T$, find the quantities

$$\tilde{\alpha}_k^2 = \frac{\left(\sum_{s=1}^{k-1} \tilde{N}_s \delta_s\right)^2 + \tilde{N}_k \left(1 - \sum_{s=1}^{k-1} (\tilde{N}_s - \tilde{N}_{s-1}) B_{s,k-1}^2\right)}{\tilde{N}_k (2^a N - \tilde{N}_k)}, \tag{27}$$

$$\tilde{\gamma}_k^{\text{fin}} = \frac{\sum_{s=1}^{k} \tilde{N}_s \delta_s}{\tilde{\alpha}_k \sqrt{2^a N \tilde{N}_k}} \quad \text{and} \quad \tilde{\gamma}_k^{\text{ini}} = \frac{\sum_{s=1}^{k-1} \tilde{N}_s \delta_s}{\tilde{\alpha}_k \sqrt{2^a N \tilde{N}_k}}, \tag{28}$$

$$\tilde{\omega}_k = \arccos\left(1 - \frac{2\tilde{N}_k}{2^a N}\right), \tag{29}$$

and

$$t_k = \left\lfloor \frac{1}{2} + \frac{1}{\tilde{\omega}_k}(\arcsin \tilde{\gamma}_k^{\text{fin}} - \arcsin \tilde{\gamma}_k^{\text{ini}}) \right\rfloor. \tag{30}$$

For $k=1,\ldots,T$, define the Grover operator

$$\hat{G}(O_k,t_k) = [(2|\Psi^0\rangle\langle\Psi^0| - \hat{I})\hat{O}_k]^{t_k}, \tag{31}$$

where

$$|\Psi^0\rangle = (2^a N)^{-1/2} \sum_{x=0}^{2^a N-1} |x\rangle, \tag{32}$$

$\hat{I}$ is the L-qubit identity operator, $\hat{O}_k|x\rangle = (-1)^{O_k(x)}|x\rangle$, and where the domain of the oracles O_k is extended to the range $0 \leqslant x < 2^a N$ by setting $O_k(x)=0$ if $x \geqslant N$.

Prepare a register of L qubits in the state $|\Psi^0\rangle$, then apply the Grover operators successively to create the state

$$|\Psi^T\rangle = \hat{G}(O_T,t_T)\cdots\hat{G}(O_1,t_1)|\Psi^0\rangle. \tag{33}$$

Now measure the a auxiliary qubits in the computational basis. If all a outcomes are 0 this stage of the algorithm successfully prepares the desired state Eq. (7).

If one of the measurements of the auxiliary qubits returns the outcome 1, then this stage of the algorithm has failed, and one has to start again by preparing the register in the state $|\Psi^0\rangle$ as in Eq. (32). Assuming the choice of parameters in Eq. (62), the probability p_{fail} that the algorithm fails in this way satisfies the bound $p_{\text{fail}} < 10\lambda$ [see Eq. (132)].

Before we provide detailed proofs of the above claims it is helpful to give a hint of how this stage of the algorithm achieves its goal. The algorithm aims at constructing the function $\sqrt{\tilde{p}(x)}$ which is close to the function $\sqrt{p''(x)}$ defined in Eq. (26). The sequence of Grover operators in Eq. (33) creates a step function that is close to $\sqrt{p''(\Pi^{-1}(x))}$ (see Fig. 1). In particular, each operator $\hat{G}(O_k,t_k)$ in Eq. (33) creates a step with the correct width N_k and a height h_k which is close to the target height δ_k. Due to a remarkable property of Grover's algorithm [16], once the features $h_1,\ldots,h_{k-1}$ have been developed they are not distorted by $\hat{G}(O_k,t_k)$ which develops h_k. In this way the algorithm proceeds building feature after feature until it constructs all T of them. At the end, because of the inherent errors, the auxiliary qubits end up having small amplitudes for nonzero values. Measuring them projects the auxiliary qubits onto the zero values with a probability that can be made arbitrarily close to 1. This also slightly changes the features h_k due to renormalization of the state after the measurement, which we take into account when we estimate the overall loss of fidelity.

III. DERIVATION OF THE ALGORITHM

In Sec. II we have already explained the second stage of our algorithm. Here we present a detailed explanation of the first stage. This section is organized as follows. In Sec. III A we review some properties of the Grover operator introducing our notation as we go along. In Sec. III B we introduce a convenient mathematical form for analyzing intermediate quantum states visited by the algorithm. And finally, in Sec. III C we derive the values (30) of the times t_k used in our algorithm.

A. Preservation of features by the Grover operator

We will be using the following result [16]. Consider an oracle O, which accepts r values of x [out of the total of $2^a N$, i.e., $\Sigma_{x=0}^{2^a N-1} O(x)=r$]. We shall call such values of x *good*, as opposed to *bad* values of x that are rejected by the oracle. Using different notation for the coefficients of good and bad states, we have that after t Grover iterations an arbitrary quantum state

$$|\Psi^{\text{ini}}\rangle = \sum_{\text{good } x} g_x^{\text{ini}}|x\rangle + \sum_{\text{bad } x} b_x^{\text{ini}}|x\rangle \tag{34}$$

is transformed into

$$|\Psi^{\text{fin}}\rangle = \hat{G}(O,t)|\Psi^{\text{ini}}\rangle = \sum_{\text{good } x} g_x^{\text{fin}}|x\rangle + \sum_{\text{bad } x} b_x^{\text{fin}}|x\rangle. \tag{35}$$

Let $\bar{g}^{\text{ini}}$ and $\bar{b}^{\text{ini}}$ be the averages of the initial amplitudes of the good and the bad states, respectively,

$$\bar{g}^{\text{ini}} = \frac{1}{r}\sum_{\text{good } x} g_x^{\text{ini}}, \quad \bar{b}^{\text{ini}} = \frac{1}{2^a N - r}\sum_{\text{bad } x} b_x^{\text{ini}}, \tag{36}$$

and similarly for the final amplitudes

$$\bar{g}^{\text{fin}} = \frac{1}{r}\sum_{\text{good } x} g_x^{\text{fin}}, \quad \bar{b}^{\text{fin}} = \frac{1}{2^a N - r}\sum_{\text{bad } x} b_x^{\text{fin}}. \tag{37}$$

Let us also define

$$\Delta g_x^{\text{ini}} = g_x^{\text{ini}} - \bar{g}^{\text{ini}}, \quad \Delta b_x^{\text{ini}} = b_x^{\text{ini}} - \bar{b}^{\text{ini}}. \tag{38}$$

In other words, Δg_x^{ini} and Δb_x^{ini} define the *features* of the initial amplitude functions g_x^{ini} and b_x^{ini} relative to their averages $\bar{g}^{\text{ini}}$ and $\bar{b}^{\text{ini}}$. Biham *et al.* have shown that the change of the amplitudes is essentially determined by the change of the averages:

$$g_x^{\text{fin}} = \bar{g}^{\text{fin}} + \Delta g_x^{\text{ini}}$$

$$b_x^{\text{fin}} = \bar{b}^{\text{fin}} + (-1)^t \Delta b_x^{\text{ini}}, \tag{39}$$

where the averages $\bar{g}^{\text{fin}}$ and $\bar{b}^{\text{fin}}$ are given as follows. Define

$$\omega = \arccos\left(1 - \frac{2r}{2^a N}\right),$$

$$\alpha = \sqrt{|\bar{b}^{\text{ini}}|^2 + |\bar{g}^{\text{ini}}|^2 \frac{r}{2^a N - r}},$$

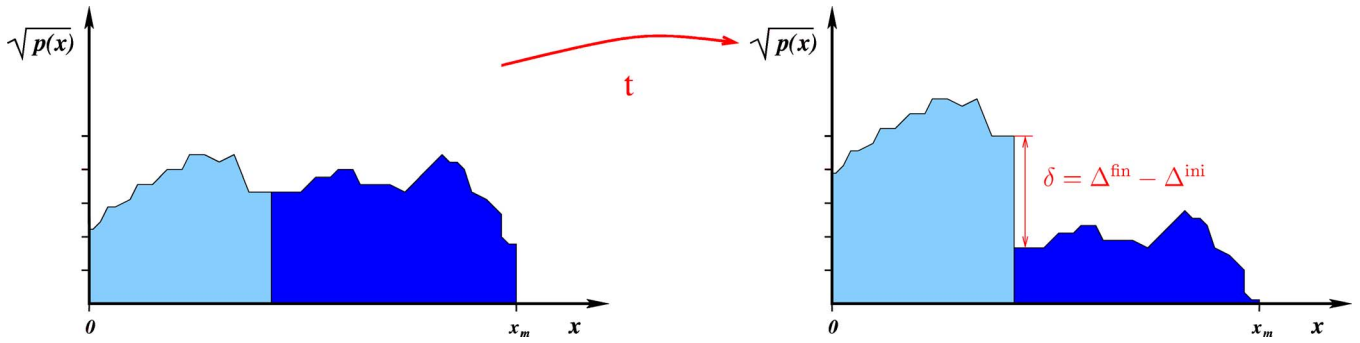

FIG. 2. (Color online) Illustration of the action of the Grover operator $\hat{G}(O,t)$ on an arbitrary amplitude function $\sqrt{p(x)}$. The light area indicates values of x that are accepted by O ("good values"), and the dark area indicates values of x that are rejected by O ("bad values").

$$\phi = \arctan\left(\frac{\bar{g}^{\text{ini}}}{\bar{b}^{\text{ini}}}\sqrt{\frac{r}{2^a N - r}}\right). \tag{40}$$

The averages are given by

$$\bar{g}^{\text{fin}} = \sqrt{\frac{2^a N - r}{r}}\alpha \sin(\omega t + \phi),$$

$$\bar{b}^{\text{fin}} = \alpha \cos(\omega t + \phi). \tag{41}$$

We shall also use the separations, Δ^{ini} and Δ^{fin}, of the averages

$$\Delta^{\text{ini}} = \bar{g}^{\text{ini}} - \bar{b}^{\text{ini}}, \quad \Delta^{\text{fin}} = \bar{g}^{\text{fin}} - \bar{b}^{\text{fin}}. \tag{42}$$

Directly from the definition we obtain

$$\Delta^{\text{fin}} = \frac{\alpha}{\sqrt{r/(2^a N)}}\sin(\omega t - \xi), \tag{43}$$

where the phase ξ can be found from Δ^{ini},

$$\xi = -\arcsin\left(\frac{\Delta^{\text{ini}}}{\alpha}\sqrt{\frac{r}{2^a N}}\right). \tag{44}$$

In our applications we will only need the case when the initial amplitude of the bad states is flat, i.e., $\Delta b_x^{\text{ini}} = 0$. In this case, the amplitude of the bad states always remains flat

$$b_x^{\text{fin}} = \bar{b}^{\text{fin}}, \tag{45}$$

This fact and the fact that the initial features Δg_x^{ini} of the amplitude of the good states are preserved,

$$g_x^{\text{fin}} - \bar{g}^{\text{fin}} = \Delta g_x^{\text{ini}}, \tag{46}$$

see (39), is crucial for understanding the rest of this paper (see Fig. 2).

B. Intermediate states

The preparation stage of our algorithm summarized in Eq. (33) gives rise to a sequence of states defined by

$$|\Psi^k\rangle = \hat{G}(O_k, t_k)|\Psi^{k-1}\rangle. \tag{47}$$

Let $\mathbb{O}_k$ be the set of solutions to the oracle O_k:

$$\mathbb{O}_k = \{x : O_k(x) = 1\}. \tag{48}$$

For $k = 2, \ldots, T$, these states can be written in the form

$$|\Psi^k\rangle = A_1^k \sum_{x\in\mathbb{O}_1} |x\rangle + A_2^k \sum_{x\in\mathbb{O}_2} |x\rangle + \cdots + A_k^k \sum_{x\in\mathbb{O}_k} |x\rangle + B^k \sum_{x\notin\mathbb{O}_k} |x\rangle, \tag{49}$$

where

$$A_j^k = B^k + \sum_{s=j}^{k} h_s. \tag{50}$$

Since B^k is real and positive, the value of B^k can be determined from the normalization condition $\langle\Psi^k|\Psi^k\rangle = 1$.

The action of the algorithm can be visualized as shown in Fig. 3, which shows the result of the first three iterations. The integers N_k [defined in Eq. (22)] are the number of good values of x according to oracle O_k. We see that each operation $\hat{G}(O_k, t_k)$ prepares a feature of height $h_k = h_k(t_k, N_1, \ldots, N_k)$. It follows from the conclusions of Sec. III A that once such a feature is developed, its height remains constant throughout the computation.

C. Values of t_k

In this subsection we show how the times t_k are related to the corresponding features h_k.

The normalization condition $\langle\Psi^k|\Psi^k\rangle = 1$ reads

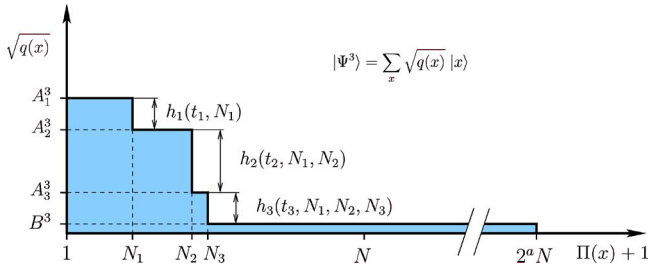

FIG. 3. (Color online) This figure shows the amplitudes for the state, $|\Psi^3\rangle$, after three iterations of the algorithm. See the text for details.

$$N_1(A_1^k)^2 + (N_2 - N_1)(A_2^k)^2 + \cdots + (N_k - N_{k-1})(A_k^k)^2 + (2^a N - N_k)(B^k)^2 = 1. \quad (51)$$

Substituting (50), this gives us a quadratic equation for B^k:

$$2^a N (B^k)^2 + 2\left(\sum_{s=1}^{k} N_s h_s\right) B^k + \sum_{s=1}^{k} (N_s - N_{s-1}) C_{s,k}^2 - 1 = 0, \quad (52)$$

where we define $N_0 = 0$ and

$$C_{s,k} = \sum_{j=s}^{k} h_j. \quad (53)$$

Solving this equation, and using the fact that $B^k \geqslant 0$, we obtain

$$B^k = -\frac{\sum_{s=1}^{k} N_s h_s}{2^a N} + \sqrt{\left(\frac{\sum_{s=1}^{k} N_s h_s}{2^a N}\right)^2 + \frac{1 - \sum_{s=1}^{k} (N_s - N_{s-1}) C_{s,k}^2}{2^a N}}. \quad (54)$$

This formula together with Eq. (50) provide an explicit expression (49) for $|\Psi^k\rangle$ in terms of the numbers $\{N_s\}_{s=1}^k$ and $\{h_s\}_{s=1}^k$.

To build the kth feature, we apply the Grover operator $\hat{G}(O_k, t_k)$ to the state $|\Psi^{k-1}\rangle$. We will now derive an expression for the integer "time" t_k in terms of the features $h_1, \ldots, h_k$ and the widths $N_1, \ldots, N_k$. Given $|\Psi^{k-1}\rangle$ as an initial state, let us define $\bar{g}_k^{\text{ini}}$ and $\bar{b}_k^{\text{ini}}$ to be the initial average amplitudes of the good and the bad states according to the oracle O_k:

$$\bar{b}_k^{\text{ini}} = B^{k-1}, \quad (55)$$

$$\bar{g}_k^{\text{ini}} = \frac{1}{N_k}[A_1^{k-1} N_1 + A_2^{k-1}(N_2 - N_1) + \cdots + A_{k-1}^{k-1}(N_{k-1} - N_{k-2}) + B^{k-1}(N_k - N_{k-1})] = \frac{\sum_{s=1}^{k-1} N_s h_s}{N_k} + B^{k-1}. \quad (56)$$

The initial separation, Δ_k^{ini}, between the good and the bad averages is therefore

$$\Delta_k^{\text{ini}} = \bar{g}_k^{\text{ini}} - \bar{b}_k^{\text{ini}} = \frac{\sum_{s=1}^{k-1} N_s h_s}{N_k}. \quad (57)$$

Observe that developing a new feature of height h_k is equivalent to increasing the initial separation Δ_k^{ini} by h_k. The final separation (after t_k steps) between the good and bad averages is therefore

$$\Delta_k^{\text{fin}} = \Delta_k^{\text{ini}} + h_k = \frac{\sum_{s=1}^{k} N_s h_s}{N_k}. \quad (58)$$

Using (57) and (58) together with (43), we, therefore, have

$$t_k = \frac{1}{\omega_k}\left[\arcsin\left(\frac{\Delta_k^{\text{fin}}}{\alpha_k}\sqrt{\frac{N_k}{2^a N}}\right) - \arcsin\left(\frac{\Delta_k^{\text{ini}}}{\alpha_k}\sqrt{\frac{N_k}{2^a N}}\right)\right], \quad (59)$$

where

$$\omega_k = \arccos\left(1 - \frac{2N_k}{2^a N}\right), \quad (60)$$

and

$$\alpha_k^2 = (\bar{b}_k^{\text{ini}})^2 + \frac{(\bar{g}_k^{\text{ini}})^2 N_k}{2^a N - N_k} = \frac{\left(\sum_{s=1}^{k-1} N_s h_s\right)^2 + N_k\left[1 - \sum_{s=1}^{k-1}(N_s - N_{s-1}) C_{s,k-1}^2\right]}{N_k(2^a N - N_k)}. \quad (61)$$

To achieve a good fidelity between the state $|\Psi_{\tilde{p}}\rangle$ that we actually prepare and our "target" state $|\Psi_p\rangle$, we want the features h_k to be as close as possible to the target values δ_k defined in Eq. (23). This motivates the formulas (27)–(30) for t_k which are obtained from the formulas (57)–(61) by (i) replacing the features h_k by the targets δ_k, (ii) replacing the widths N_k by the measured values $\tilde{N}_k$, and (iii) by rounding to the nearest integer.

IV. FIDELITY ANALYSIS

In the above description of the algorithm, we have not specified how to choose the parameters η_c, η_g, and a as a function of ϵ (or, alternatively, of the initially given parameters λ and η). The optimal choice for these parameters depends on the estimates $\tilde{n}_1, \ldots, \tilde{n}_{1/\epsilon}$ obtained in the quantum counting step. In this section we provide a rather generous worst case analysis which shows that the choice

$$\eta_c = \epsilon^5/54, \quad \eta_g = 0.99\epsilon^2, \quad a = \left(\log_2 \frac{\eta_g}{\eta_c} - 3\right) \quad (62)$$

guarantees, with probability greater than $1 - \nu$, the fidelity bound

$$|\langle \Psi_{\tilde{p}} | \Psi_p \rangle| > 1 - \lambda. \quad (63)$$

This bound is valid for arbitrary values of the $\tilde{n}_k$. In most actual applications, much larger values of the accuracy parameters ϵ, η_g, and η_c will be sufficient to guarantee this fidelity bound.

We now show that Eq. (30), for the times t_k which we motivated in the previous section, implies the fidelity bound (63) under the assumption that the parameters η_c, η_g, and a are chosen as in Eq. (62).

Our starting point will be two sets of expressions for the t_k, namely the definition of the t_k, Eqs. (27)–(30), in terms of the measured values $\tilde{N}_k$ and the target values δ_k, and Eqs. (57)–(61) above in terms of the actual values N_k and h_k. In Sec. IV A we will derive an upper bound on the error $|\delta_k - h_k|$. This bound shows how accurate our algorithm is in achieving the target height, δ_k, for the features h_k. The overall accuracy of our algorithm, however, also depends on how accurate it is in achieving the correct width of the features. This accuracy is determined by the fraction of x values for which $p'(x) \neq p''(x)$ (see Fig. 1). In Sec. IV B we obtain an upper bound on this fraction. In Sec. IV C, we derive the fidelity bound (63) and an upper bound on the probability that the algorithm fails due to a nonzero outcome of the measurement of the auxiliary qubits. And finally, in Sec. IV D, we show that the bound (11) on the phases $\tilde{\phi}(x)$ implies the overall fidelity bound (12).

A. Bound on $|\delta_k - h_k|$

It is convenient to perform the proof of the bound on $|\delta_k - h_k|$ in three steps. For this we note that t_k depend on the values of $\tilde{\gamma}_k^{\text{ini}}$, $\tilde{\gamma}_k^{\text{fin}}$ and $\tilde{\omega}_k$. In Sec. IV A 1 we determine the range of possible values for $\tilde{\gamma}_k^{\text{ini}}$ and $\tilde{\gamma}_k^{\text{fin}}$ that corresponds to the uncertainty in the measured values of $\tilde{N}_k$. Similarly in Sec. IV A 2 we determine the error range for $\tilde{\omega}_k$, and finally, in Sec. IV A 3 we complete the proof of the bound.

1. Error range for $\tilde{\gamma}_k^{\text{ini}}$ and $\tilde{\gamma}_k^{\text{fin}}$

Equation (59) provides an explicit expression for t_k in terms of $\{h_k\}$ and $\{N_k\}$:

$$t_k = F(\{h_s\},\{N_s\}). \tag{64}$$

In what follows it will be convenient to use auxiliary quantities $\{\tau_k\}$ defined as

$$\tau_k = F(\{\delta_s\},\{N_s\}). \tag{65}$$

The meaning of τ_k becomes clear in comparison with (64): τ_k are the time intervals that correspond to the target features $\{\delta_k\}$. Unlike $\{t_k\}$, $\{\tau_k\}$ are not necessarily integers.

It will be convenient to rewrite the definition of t_k given in Eqs. (27)–(30) in a slightly modified form:

$$t_k = \left\lfloor \frac{1}{2} + \frac{1}{\tilde{\omega}_k}(\arcsin \tilde{\gamma}_k^{\text{fin}} - \arcsin \tilde{\gamma}_k^{\text{ini}}) \right\rfloor, \tag{66}$$

where we use the following definitions:

$$\tilde{\omega}_k = \arccos\left(1 - \frac{2\tilde{N}_k}{2^a N}\right), \tag{67}$$

$$\tilde{\gamma}_k^{\text{fin}} = \frac{\tilde{\Delta}_k^{\text{fin}}}{\tilde{\alpha}_k}\sqrt{\frac{\tilde{N}_k}{2^a N}}, \quad \tilde{\gamma}_k^{\text{ini}} = \frac{\tilde{\Delta}_k^{\text{ini}}}{\tilde{\alpha}_k}\sqrt{\frac{\tilde{N}_k}{2^a N}}, \tag{68}$$

$$\tilde{\Delta}_k^{\text{fin}} = \frac{\sum_{s=1}^{k} \tilde{N}_s \delta_s}{\tilde{N}_k}, \quad \tilde{\Delta}_k^{\text{ini}} = \frac{\sum_{s=1}^{k-1} \tilde{N}_s \delta_s}{\tilde{N}_k}, \tag{69}$$

and

$$\tilde{\alpha}_k^2 = \frac{\left(\sum_{s=1}^{k-1} \tilde{N}_s \delta_s\right)^2 + \tilde{N}_k\left[1 - \sum_{s=1}^{k-1} (\tilde{N}_s - \tilde{N}_{s-1}) B_{s,k-1}^2\right]}{\tilde{N}_k(2^a N - \tilde{N}_k)}, \tag{70}$$

where

$$B_{s,k} = \sum_{j=s}^{k} \delta_j. \tag{71}$$

It is also convenient to define

$$\gamma_k^{\text{fin}} = \frac{\sum_{s=1}^{k} N_s \delta_s}{\alpha_k \sqrt{2^a N N_k}} \quad \text{and} \quad \gamma_k^{\text{ini}} = \frac{\sum_{s=1}^{k-1} N_s \delta_s}{\alpha_k \sqrt{2^a N N_k}}. \tag{72}$$

In this notation Eq. (65) can be rewritten as

$$\omega_k \tau_k = \arcsin \gamma_k^{\text{fin}} - \arcsin \gamma_k^{\text{ini}}. \tag{73}$$

Directly from the definitions we have

$$(\tilde{\gamma}_k^{\text{fin}})^2 = \frac{\left(1 - \frac{\tilde{N}_k}{2^a N}\right)\left(\sum_{s=1}^{k} \tilde{N}_s \delta_s\right)^2}{\left(\sum_{s=1}^{k-1} \tilde{N}_s \delta_s\right)^2 + \tilde{N}_k\left[1 - \sum_{s=1}^{k-1} (\tilde{N}_s - \tilde{N}_{s-1}) B_{s,k-1}^2\right]}. \tag{74}$$

By direct calculation we get

$$\begin{aligned}\sum_{s=1}^{k-1} (\tilde{N}_s - \tilde{N}_{s-1}) B_{s,k-1}^2 &= \sum_{s=1}^{k-1} \tilde{N}_s \left(\delta_s^2 + 2\delta_s \sum_{j=s+1}^{k} \delta_j\right) + \tilde{N}_{k-1}\delta_k^2 \\ &= \sum_{s=1}^{k-1} N_s \left(1 \pm \eta_c \frac{N}{N_s}\right)\left(\delta_s^2 + 2\delta_s \sum_{j=s+1}^{k} \delta_j\right) \\ &\quad + N_{k-1}\left(1 \pm \eta_c \frac{N}{N_{k-1}}\right)\delta_k^2. \end{aligned} \tag{75}$$

The ± notation is an abbreviation for a double inequality (see the Appendix A). Since $N_s \geqslant (\eta_g - \eta_c)N$ we obtain

$$\sum_{s=1}^{k-1} (\tilde{N}_s - \tilde{N}_{s-1}) B_{s,k-1}^2 = \left(1 \pm \frac{\eta_c}{\eta_g - \eta_c}\right) \sum_{s=1}^{k-1} (N_s - N_{s-1}) B_{s,k-1}^2, \tag{76}$$

and similarly

$$\sum_{s=1}^{k} \tilde{N}_s \delta_s = \left(1 \pm \frac{\eta_c}{\eta_g - \eta_c}\right) \sum_{s=1}^{k} N_s \delta_s. \tag{77}$$

Since $\eta_c / \eta_g < \frac{1}{10}$, we therefore have

$$\tilde{\gamma}_k^{\mathrm{fin}} = \gamma_k^{\mathrm{fin}} \frac{\left(1 \pm \frac{\eta_c}{\eta_g - \eta_c}\right)^{3/2}}{1 \pm \frac{\eta_c}{\eta_g - \eta_c}} = \gamma_k^{\mathrm{fin}}\left(1 \pm 10\frac{\eta_c}{\eta_g}\right), \quad (78)$$

and similarly

$$\tilde{\gamma}_k^{\mathrm{ini}} = \gamma_k^{\mathrm{ini}}\left(1 \pm 10\frac{\eta_c}{\eta_g}\right). \quad (79)$$

In the above formulas we have used the mean value theorem that states that for any function f that is continuous on the interval $|x| \leqslant c$, where c is some constant, and differentiable on $|x| < c$ we can write

$$f(x) = f(0) + xf'(x_0), \quad (80)$$

where $f'(x_0)$ denotes the derivative of f at some point $|x_0| < |x|$. This gives, for example, that for $|x| < \frac{1}{10}$

$$\frac{1}{1 \pm x} = 1 \pm 2x,$$

$$(1 \pm x)^{3/2} = 1 \pm 1.6x. \quad (81)$$

The error bounds (78) and (79) were obtained by simplifying the somewhat tighter but unwieldy bounds using these methods.

2. Error range for $\tilde{\omega}_k$

The aim of this subsection is to determine the ratio between $\tilde{\omega}_k$ and the true value ω_k given by Eqs. (67) and (60), respectively. Using the mean value theorem we have, by definition,

$$\begin{aligned}\tilde{\omega}_k &= \arccos\left[1 - \frac{2N_k}{2^a N}\left(1 \pm \frac{\eta_c}{\eta_g - \eta_c}\right)\right] \\ &= \arccos\left(1 - \frac{2N_k}{2^a N}\right) - \frac{1}{\sqrt{1 - \left[1 - \frac{2N_k}{2^a N}(1+u)\right]^2}} \\ &\quad \times \left(\pm \frac{2N_k}{2^a N}\frac{\eta_c}{\eta_g - \eta_c}\right), \end{aligned} \quad (82)$$

where u is a real number such that $|u| < \eta_c/\eta_g$. This implies

$$\tilde{\omega}_k = \omega_k \pm \sqrt{\frac{2N_k}{2^a N}} \frac{1}{\sqrt{1+u}\sqrt{2 - \frac{2N_k}{2^a N}(1+u)}} \frac{\eta_c}{\eta_g - \eta_c}. \quad (83)$$

Since $\eta_c/\eta_g < \frac{1}{10}$ we have

$$\frac{\eta_c}{\eta_g - \eta_c} < \frac{\eta_c}{0.9\eta_g}, \quad \sqrt{1+u} > \frac{1}{0.9\sqrt{2}}, \quad \sqrt{2 - \frac{2N_k}{2^a N}(1+u)} > 1, \quad (84)$$

and therefore

$$\tilde{\omega}_k = \omega_k \pm 2\frac{\eta_c}{\eta_g}\sqrt{\frac{N_k}{2^a N}}. \quad (85)$$

It now remains to find a bound on $\sqrt{N_k/2^a N}$ that is linear in ω_k. We have, by definition,

$$\frac{2N_k}{2^a N} = 1 - \cos\omega_k = 2\sin^2\frac{\omega_k}{2}. \quad (86)$$

Since $x^2 \geqslant \sin^2 x$ we obtain

$$\sqrt{\frac{N_k}{2^a N}} \leqslant \frac{\omega_k}{2}, \quad (87)$$

and therefore

$$\frac{\tilde{\omega}_k}{\omega_k} = 1 \pm \frac{\eta_c}{\eta_g}. \quad (88)$$

3. Proof of the bound

During the kth stage our algorithm creates a feature of height

$$h_k = \alpha_k \sqrt{\frac{2^a N}{N_k}}[\sin(\omega_k t_k - \xi_k) - \sin(-\xi_k)], \quad (89)$$

where ξ_k is some initial phase. On the other hand, the target height δ_k is

$$\delta_k = \alpha_k \sqrt{\frac{2^a N}{N_k}}[\sin(\omega_k \tau_k - \xi_k) - \sin(-\xi_k)]. \quad (90)$$

Hence

$$\begin{aligned}|h_k - \delta_k| &= \alpha_k \sqrt{\frac{2^a N}{N_k}}|\sin(\omega_k t_k - \xi_k) - \sin(\omega_k \tau_k - \xi_k)| \\ &\leqslant 2\alpha_k \sqrt{\frac{2^a N}{N_k}}\left|\sin\frac{\omega_k(t_k - \tau_k)}{2}\right| \leqslant \alpha_k \sqrt{\frac{2^a N}{N_k}}\omega_k|t_k - \tau_k|. \end{aligned} \quad (91)$$

Directly from the definition we obtain

$$\omega_k t_k = \frac{\omega_k}{\tilde{\omega}_k}(\arcsin\tilde{\gamma}_k^{\mathrm{fin}} - \arcsin\tilde{\gamma}_k^{\mathrm{ini}}) \pm \omega_k. \quad (92)$$

Using Eqs. (78), (79), and (88), using the fact that $10\eta_c/\eta_g \leqslant 1/4$ and using the inequality (A5) we have

$$\omega_k t_k = \frac{1}{1 \pm \eta_c/\eta_g}\left(\zeta_k \pm 4\sqrt{10\frac{\eta_c}{\eta_g}}\right) \pm \omega_k. \quad (93)$$

Now using (81) and the fact that $|\zeta_k| \leqslant \pi$ we derive from the above equation

$$\omega_k(t_k - \tau_k) = \pm 8\frac{\eta_c}{\eta_g} \pm \omega_k. \quad (94)$$

Since $a \geqslant 3$ [see Eqs. (62) and (6)], we have that $2N_k/(2^a N) \leqslant 1/4$, and we can, therefore, use the inequality (A6) to show that

$$\omega_k = \pm 2\sqrt{\frac{2N_k}{2^a N}}. \tag{95}$$

Since $a \leqslant \log_2(\eta_g/\eta_c) - 3$, we obtain the bound

$$\omega_k(t_k - \tau_k) = \pm 16\frac{\eta_c}{\eta_g}. \tag{96}$$

The maximum possible value of the average amplitude of "bad" states is

$$\max \bar{b} = \frac{1}{\sqrt{2^a N}}, \tag{97}$$

and the maximum possible value of the average amplitude of "good" states is

$$\max \bar{g} = \frac{1}{\sqrt{\eta N}}. \tag{98}$$

One can, therefore, write

$$\alpha_k^2 < (\max \bar{b})^2 + \frac{(\max \bar{g})^2 N_k}{2^a N - N_k} \leqslant \frac{1}{2^a N} + \frac{1}{\eta(2^a - 1)N} \leqslant \frac{1}{2^a N} + \frac{2}{\eta 2^a N} < \frac{4}{\eta 2^a N}. \tag{99}$$

Using this bound together with (96) we obtain for the error (91):

$$|h_k - \delta_k| < \frac{32}{\sqrt{\eta N(1 - \eta_c/\eta_g)}}\frac{\eta_c}{\eta_g^{3/2}}. \tag{100}$$

Substituting the parameters (62) into the right-hand side of the above inequality one can show that

$$|h_k - \delta_k| < \frac{\epsilon^2}{\sqrt{\eta N}}. \tag{101}$$

B. Number of exceptional values

When describing our algorithm in Sec. II we have introduced functions p' and p'' to distinguish two different approximations to the target function p. Namely, if p' is an approximation of p which is defined by the oracles o_k, then p'' also takes into account the fact that we may not know the exact values of n_k. In our algorithm we therefore use p'' as our target function which coincides with p' everywhere apart for a small fraction of values of x for which $p'(x) \neq p''(x)$. In this section we obtain an upper bound on this fraction which will then be used in the next section where we derive bounds on the fidelity and the failure probability.

For all j, we have $|n_j - \tilde{n}_j| \leqslant \eta_c N$. For $j < f_1$, we have $\tilde{n}_j < \eta_g N$, and hence

$$n_j < (\eta_g + \eta_c)N. \tag{102}$$

We now consider, for each $k \in \{2, \ldots, T\}$, all values of j such that $f_{k-1} \leqslant j < f_k$. For these values of j, we have

$$\tilde{n}_j \leqslant \tilde{n}_{f_{k-1}}. \tag{103}$$

Since $n_j \geqslant n_{f_{k-1}}$, we have

$$|\tilde{n}_j - \tilde{n}_{f_{k-1}}| \leqslant 2\eta_c N, \tag{104}$$

hence

$$|\tilde{n}_j - \tilde{N}_{k-1}| \leqslant 2\eta_c N, \tag{105}$$

and finally

$$|n_j - N_{k-1}| \leqslant 4\eta_c N. \tag{106}$$

Since $T \leqslant 1/\epsilon$, we find that $p''(x) \neq p'(x)$ for at most μN values, where

$$\mu = 4\eta_c/\epsilon + (\eta_g + \eta_c). \tag{107}$$

C. Fidelity bound and the failure probability

For $k = T$, Eq. (49) can be rewritten in the form

$$|\Psi^T\rangle = \sum_{x=0}^{2^a N - 1} \left(B^T + \sum_{j=1}^{T} c_j(x) h_j \right) |x\rangle, \tag{108}$$

where

$$c_j(x) = \begin{cases} 1 & \text{if } x < N_j \\ 0 & \text{otherwise.} \end{cases} \tag{109}$$

Let us define

$$d(x) = \sum_{j=1}^{T} c_j(x)(h_j - \delta_j). \tag{110}$$

Using this definition we have

$$|\langle \Psi_p | \Psi^T \rangle| = \left| \sum_{x=0}^{2^a N - 1} \sqrt{p(x)} \left(B^T + \sum_{j=1}^{T} c_j(x) h_j \right) \right| \geqslant \left| \sum_{x=0}^{N-1} \sqrt{p(x)} \times \left(B^T + \sum_{j=1}^{T} c_j(x)\delta_j \right) \right| - \left| \sum_{x=0}^{N-1} \sqrt{p(x)} d(x) \right|, \tag{111}$$

where we have used the fact that $p(x) = 0$ for $x \geqslant N$. Using (101), and since $T \leqslant 1/\epsilon$ we obtain

$$|d(x)| < \frac{\epsilon}{\sqrt{\eta N}}. \tag{112}$$

Because $\sqrt{p(x)} \leqslant 1/\sqrt{\eta N}$, this implies

$$|\langle \Psi_p | \Psi^T \rangle| \geqslant \left| \sum_{x=0}^{N-1} \sqrt{p(x)} \left(B^T + \sum_{j=1}^{T} c_j(x)\delta_j \right) \right| - \frac{\epsilon}{\eta}. \tag{113}$$

Rewriting Eq. (26) in terms of the coefficients $c_j(x)$,

$$\sqrt{p''(x)} = \sum_{j=1}^{T} c_j(x)\delta_j, \tag{114}$$

we can write

$$|\langle\Psi_p|\Psi^T\rangle| \geqslant \sum_{x=0}^{N-1} \sqrt{p(x)}\sqrt{p''(x)} - |B^T|\sqrt{\frac{N}{\eta}} - \frac{\epsilon}{\eta}. \quad (115)$$

For all x with a possible exception of at most μN values $p''(x)=p'(x)$ (see Sec. IV B). Let $\mathbb{S}_e$ be the set of exceptional values of x for which $p''(x)\neq p'(x)$. We have

$$|\langle\Psi_p|\Psi^T\rangle| \geqslant \sum_{x=0}^{N-1} \sqrt{p(x)}\sqrt{p'(x)} - \sum_{x\in\mathbb{S}_e} \sqrt{p(x)}|\sqrt{p'(x)} - \sqrt{p''(x)}| - |B^T|\sqrt{\frac{N}{\eta}} - \frac{\epsilon}{\eta}. \quad (116)$$

Since $\sqrt{p(x)}$, $\sqrt{p'(x)}$, and $\sqrt{p''(x)}$ are all bounded from the above by $1/\sqrt{\eta N}$ we obtain

$$|\langle\Psi_p|\Psi^T\rangle| \geqslant \sum_{x=0}^{N-1} \sqrt{p(x)}\sqrt{p'(x)} - |B^T|\sqrt{\frac{N}{\eta}} - \frac{\epsilon+\mu}{\eta}. \quad (117)$$

By definition of p'

$$|\sqrt{p'(x)} - \sqrt{p(x)}| \leqslant \frac{\epsilon}{\sqrt{\eta N}}, \quad (118)$$

and since p is normalized we get

$$|\langle\Psi_p|\Psi^T\rangle| \geqslant 1 - |B^T|\sqrt{\frac{N}{\eta}} - \frac{2\epsilon+\mu}{\eta}. \quad (119)$$

In order to continue we need to calculate $|B^T|$. This can be done by examining the normalization condition $\langle\Psi^T|\Psi^T\rangle = 1$. This condition reads

$$\sum_{x=0}^{2^aN-1} \left(B^T + \sum_{j=1}^{T} c_j(x)h_j\right)^2 = 1. \quad (120)$$

Using (110) and (114) this can be rewritten as

$$\sum_{x=0}^{2^aN-1} [\sqrt{p''(x)} + B^T + d(x)]^2 = 1, \quad (121)$$

or

$$\sum_{x=0}^{2^aN-1} [\sqrt{p'(x)} + B^T + d(x)]^2 - \Lambda = 1, \quad (122)$$

where

$$\Lambda = \sum_{x\in\mathbb{S}_e} \{[\sqrt{p'(x)} + B^T + d(x)]^2 - [\sqrt{p''(x)} + B^T + d(x)]^2\} = 2B^T \sum_{x\in\mathbb{S}_e} [\sqrt{p'(x)} - \sqrt{p''(x)}] + \sum_{x\in\mathbb{S}_e} [p'(x) - p''(x)] + 2\sum_{x\in\mathbb{S}_e} [\sqrt{p'(x)} - \sqrt{p''(x)}]d(x). \quad (123)$$

Let us define

$$e(x) = \sqrt{p'(x)} - \sqrt{p(x)}. \quad (124)$$

Since p is normalized, Eq. (122) gives a quadratic equation for B^T:

$$(B^T)^2 + 2UB^T + V = 0, \quad (125)$$

where

$$U = \frac{1}{2^aN}\left(\sum_{x=0}^{N-1} [\sqrt{p'(x)} + d(x)] + \sum_{x\in\mathbb{S}_e} [\sqrt{p''(x)} - \sqrt{p'(x)}]\right), \quad (126)$$

and

$$V = \frac{1}{2^aN}\sum_{x=0}^{N-1} \{2\sqrt{p(x)}[d(x)+e(x)] + [d(x)+e(x)]^2\} + \frac{1}{2^aN}\sum_{x\in\mathbb{S}_e} \{p''(x) - p'(x) + 2d(x)[\sqrt{p''(x)} - \sqrt{p'(x)}]\}. \quad (127)$$

Since $\sqrt{p'(x)}$ and $\sqrt{p''(x)}$ are bounded from above by $1/\sqrt{\eta N}$ and since $\mathbb{S}_e$ contains at most μN elements (see Sec. IV B), we obtain with the help of Eq. (112)

$$|U| \leqslant \frac{1+\epsilon+\mu}{2^a\sqrt{\eta N}}. \quad (128)$$

Similarly, since $|e(x)| \leqslant \epsilon/\sqrt{\eta N}$ we have

$$|V| \leqslant \frac{6\epsilon + 4\epsilon^2 + \mu}{2^aN\eta}. \quad (129)$$

Since $\mu < \epsilon^2$ and $2^a > 6/\epsilon^2$, see Eqs. (107) and (62), respectively, we obtain

$$|B^T| \leqslant |U| + \sqrt{U^2 + |V|} \leqslant \frac{2\epsilon^2}{\sqrt{\eta N}}. \quad (130)$$

Together with Eq. (119) this gives the lower bound on the fidelity,

$$|\langle\Psi_p|\Psi^T\rangle| \geqslant 1 - \frac{3\epsilon}{\eta}, \quad (131)$$

where we have observed that $\epsilon < \frac{1}{3}$ and used the bounds $2^a < 7/\epsilon^3$, $\mu < \epsilon^2$, which follow from our settings given in Eq. (62). The failure probability is

$$p_{\text{fail}} = (2^aN - 1)N|B^T|^2 < \frac{28\epsilon}{\eta} < 10\lambda. \quad (132)$$

D. Introduction of phases and the fidelity bound

We now show that the choice $\lambda' = \epsilon'^2/8$ together with the inequality (11), i.e., $|\tilde{\phi}(x) - \phi(x)| \leqslant \epsilon'/2$, implies the overall fidelity bound (12). The proof is straightforward.

$$|\langle\tilde{\Psi}|\Psi\rangle| = \left|\sum_x \sqrt{p(x)\tilde{p}(x)}\exp\{2\pi i[\phi(x)-\tilde{\phi}(x)]\}\right|$$
$$\geqslant \sum_x \sqrt{p(x)\tilde{p}(x)}\cos[\phi(x)-\tilde{\phi}(x)] \geqslant \sum_x \sqrt{p(x)\tilde{p}(x)}\{1$$
$$-[\phi(x)-\tilde{\phi}(x)]^2/2\} \geqslant \sum_x \sqrt{p(x)\tilde{p}(x)}(1-\epsilon'^2/8)$$
$$= |\langle\Psi_{\tilde{p}}|\Psi_p\rangle|(1-\lambda') > 1-\lambda-\lambda'. \quad (133)$$

V. RESOURCES

In this section we provide worst case scenarios for upper bounds on the resources required by the algorithm. We distinguish between the resources that are needed for the state preparation part of the algorithm (Sec. V A) and the resources that are needed by the quantum counting that precedes the actual state preparation (Sec. V B).

A. Resources needed for state preparation

1. Auxiliary qubits

From our settings (62) we obtain

$$\frac{\eta_g}{\eta_c} < 54/\epsilon^3. \quad (134)$$

We thus obtain for the number of auxiliary qubits

$$a \leqslant \log_2(\eta_g/\eta_c) - 3 < 3 + \log_2 \epsilon^{-3}. \quad (135)$$

2. Oracle calls

Here we give an upper bound on the time resources needed by the algorithm. The construction of one feature requires at most

$$\max(t_k) \leqslant \frac{2\pi}{\omega_k} \quad (136)$$

oracle calls. Using inequality (87) we can therefore write

$$\max t_k \leqslant \pi\sqrt{\frac{2^a N}{N_k}}. \quad (137)$$

From (62) we have

$$\eta_g - \eta_c > \tfrac{8}{9}\epsilon^2. \quad (138)$$

Since there are at most $1/\epsilon$ features and because $N_k \leqslant (\eta_g - \eta_c)N$ and $2^a < 8/\epsilon^3$ we, therefore, have that the total number of oracle calls, n_{oracle}, satisfies the bound

$$n_{\text{oracle}} \leqslant \frac{\max t_k}{\epsilon} < \frac{3\pi}{\epsilon^3\sqrt{\epsilon}}. \quad (139)$$

B. Resources needed for counting

1. Counting accuracy

Consider an oracle O on the set of $2^a N$ possible values of x. Using standard techniques we can count the number M of solutions of O within the absolute error ΔM

$$\Delta M < \left(\sqrt{2^a NM} + \frac{N}{2^{m-a+2}}\right)2^{-m}, \quad (140)$$

where m is the number of auxiliary qubits needed by the standard quantum counting routine [19].

We want $\Delta M < \eta_c N$, where η_c is the counting accuracy introduced earlier. This connects the desired counting accuracy η_c with the number of auxiliary qubits m,

$$\eta_c = \left(\sqrt{\frac{2^a M}{N}} + 2^{a-2}\lambda\right)\lambda, \quad (141)$$

where $\lambda = 2^{-m}$. Solving this equation for λ in the case $\lambda > 0$, we have

$$\lambda = 2^{1-a/2}(\sqrt{y+\eta_c} - \sqrt{y}), \quad (142)$$

where $y = M/N$. We see that, the larger the value of a, the larger m has to be in order to give the required counting accuracy η_c. We therefore set a to the minimum, i.e., $a=1$ (this doubles the range of x values to ensure reliable counting, see, e.g., [19]).

It is easy to check that the dependence of λ on y is monotonic. As we vary M in the range 0 to $N-1$, the corresponding values of λ vary between the limits $\sqrt{2\eta_c}$ and $\sqrt{2}(\sqrt{1+\eta_c}-1) > \eta_c/2$. It follows that the required number of auxiliary working qubits needed for counting with accuracy η_c is $m < \log_2 \eta_c^{-1}$. Thus we choose $m = \log_2 \eta_c^{-1}$. This choice guarantees the required accuracy of counting irrespective of the true value of M.

2. Counting probability

The above counting procedure does not output the correct result with probability 1. For the procedure to work correctly with probability $1-\nu$ we have to increase the number of auxiliary qubits from m to a_c which is given by

$$a_c = m + \log_2\left(2 + \frac{1}{2\nu}\right) = \log_2 \frac{1+4\nu}{2\nu\eta_c}. \quad (143)$$

The number N_{count} of oracle calls that is required by the counting procedure is

$$N_{\text{count}} = 2^{a_c} - 1. \quad (144)$$

Substituting $m = \log_2 \eta_c^{-1}$ and using Eq. (62) we obtain

$$a_c < \log_2 \frac{27(1+4\nu)}{\nu\epsilon^5}. \quad (145)$$

Since there are at most $1/\epsilon$ features the total number of oracle calls needed by the counting stage of our algorithm is bounded as

$$N_{\text{count}}^{\text{total}} \leqslant \frac{N_{\text{count}}}{\epsilon} < \frac{27(1+4\nu)}{\nu\epsilon^6}. \quad (146)$$

VI. SUMMARY AND CONCLUSIONS

In conclusion, we have described a quantum algorithm to prepare an arbitrary state of a quantum register of $\log_2 N$

qubits, provided the state is initially given in the form of a classical algorithm to compute the N complex amplitudes defining the state. For an important class of states, the algorithm is efficient in the sense of requiring numbers of oracle calls and additional gate operations that are polynomial in the number of qubits. The following table lists, for each stage of the algorithm, upper bounds on the number of oracle calls and the number of auxiliary qubits needed.

	oracle calls	auxiliary qubits
counting	$27(1+4\nu)/\nu\epsilon^6$	$\log_2 27(1+4\nu)/\nu\epsilon^5$
preparing $\lvert\Psi_{\tilde{p}}\rangle$	$3\pi/\epsilon^3\sqrt{\epsilon}$	$3+3\log_2 1/\epsilon$
preparing $\lvert\tilde{\Psi}\rangle$	$1/\epsilon'$	0

The bounds are not tight and can be improved by a more detailed error analysis. The total number of quantum gate operations depends on the implementation of the oracles. It is proportional to the number of oracle calls times a factor polynomial in $\log_2 N$ if the functions $p(x)$ and $\phi(x)$ can be efficiently computed classically.

Depending on the nature of the function $p(x)$ and the prior information about $p(x)$, the algorithm we have described in this paper can be optimized in a number of ways. For instance, the counting stage is the most expensive in terms of both oracle calls and additional qubits. If for some reason the numbers n_k characterizing the oracles are known in advance, the counting stage can be omitted, leading to considerable savings. Furthermore, in this case the fidelity bound can be guaranteed with probability 1, i.e., we can set $\nu=0$.

In some cases the algorithm can be simplified if, instead of using the oracles defined in Eq. (13), one uses oracles that return the kth bit of the expression $\sqrt{p(x)/\eta}N$. The general conclusions of the paper continue to hold for this variant of the algorithm, which we analyze in detail in Ref. [20].

Finally, by using generalizations of Grover's algorithm in which the oracles and the inversion about the mean introduce complex phase factors [17,18] it is possible to reduce the number of auxiliary qubits needed in the preparation stage of the algorithm. This leads to a reduction in the number of required oracle calls, and could also be important in implementations where the number of qubits is the main limiting factor.

ACKNOWLEDGMENT

This work was supported in part by the European Union IST-FET Project No. EDIQIP.

APPENDIX A

1. Notation for double inequalities

In this paper we have made a frequent use of the following convention. Let a, b, and c be three numbers. The notation

$$a = b \pm c \tag{A1}$$

is then understood to be equivalent to the double inequality

$$b - c \leqslant a \leqslant b + c. \tag{A2}$$

Furthermore, let g, e, and F be functions. The notation

$$h(x) = \sum_{x\in\mathcal{I}} F(g(x) \pm e(x)) \tag{A3}$$

is then equivalent to the statement that $h(x)$ can be written in the form

$$h(x) = \sum_{x\in\mathcal{I}} F(f(x)), \tag{A4}$$

where $f(x)=g(x)\pm e(x)$ for all $x\in\mathcal{I}$.

2. Trigonometric inequalities

Here we prove the following inequalities:

$$|\arcsin(x+\nu) - \arcsin(x)| \leqslant 2\sqrt{|\nu|}, \quad |\nu| \leqslant 1/4 \tag{A5}$$

and

$$|\arccos(x+\nu) - \arccos(x)| \leqslant 2\sqrt{\nu}, \quad |\nu| \leqslant 1/4. \tag{A6}$$

Consider the case $\nu\geqslant 0$, which implies that for any x

$$\arcsin(x+\nu) - \arcsin(x) \geqslant 0. \tag{A7}$$

By inspection of arcsin function we have that for $0\leqslant\nu\leqslant 1/4$ the maximum value of the difference $[\arcsin(x+\nu) - \arcsin(x)]$ is achieved for $x=-1$:

$$\max_x[\arcsin(x+\nu) - \arcsin(x)] = \arcsin(\nu-1) - \arcsin(-1). \tag{A8}$$

In the case of $y<z$ the following equality holds [21]

$$\arcsin z - \arcsin y = \arccos(\sqrt{1-y^2}\sqrt{1-z^2} + yz). \tag{A9}$$

Applying this equality to the right-hand side of (A8) we obtain

$$\arcsin(x+\nu) - \arcsin(x) \leqslant \arccos(1-\nu). \tag{A10}$$

Let us now look for a constant c such that

$$\arccos(1-\nu) \leqslant c\sqrt{\nu}. \tag{A11}$$

Since arccos is a decreasing function the above requirement is equivalent to

$$1 - \nu \geqslant \cos(c\sqrt{\nu}). \tag{A12}$$

According to the mean value theorem, there exists $u<c\sqrt{\nu}$ such that

$$\cos(c\sqrt{\nu}) = 1 - \frac{c^2\nu}{2}\cos(u), \tag{A13}$$

and therefore the requirement (A12) can be rewritten as

$$1 \leqslant \frac{c^2}{2}\cos(u). \tag{A14}$$

It is clear that this requirement is guaranteed to be satisfied if we set $c=2$. Indeed, $2\cos u>1$ for any non-negative u

$< \pi/3$ which includes all possible values of u that can correspond to $c=2$ and $\nu \leqslant 1/4$. Since $c=2$ guarantees that (A11) is satisfied, we obtain from (A10)

$$\arcsin(x+\nu) - \arcsin(x) \leqslant 2\sqrt{\nu}, \quad \nu \geqslant 0. \tag{A15}$$

The case of negative ν can be treated in an analogous fashion leading to the inequality

$$\arcsin(x) - \arcsin(x+\nu) \leqslant 2\sqrt{|\nu|}, \quad \nu \leqslant 0. \tag{A16}$$

The required inequality (A5) follows trivially. Moreover, since

$$\arcsin x + \arccos x = \pi/2, \tag{A17}$$

we also obtain (A6) as required.

[1] S. Lloyd, Science **273**, 1073 (1996).
[2] C. Zalka, Proc. R. Soc. London, Ser. A **454**, 313 (1998).
[3] S. Somaroo, C. -H. Tseng, T. F. Havel, R. Laflamme, and D. G. Cory, Phys. Rev. Lett. **82**, 5381 (1999).
[4] I. C. Percival, in *Quantum Chaos, Quantum Measurement*, edited by P. Cvitanović, I. C. Percival, and A. Wirzba (Kluwer, Dordrecht, Holland, 1992), p. 199.
[5] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
[6] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Fortschr. Phys. **46**, 493 (1998).
[7] L. Grover and T. Rudolph, e-print quant-ph/0208112.
[8] P. Kaye and M. Mosca, in *Proceedings of the International Conference on Quantum Information* (Rochester, New York, 2001); e-print quant-ph/0407102.
[9] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca, Proc. R. Soc. London, Ser. A **454**, 339 (1998).
[10] T. Rudolph (private communication).
[11] D. Ventura and T. Martinez, Found. Phys. Lett. **12**, 547 (1999).
[12] G. L. Long and Y. Sun, Phys. Rev. A **64**, 014303 (2001).
[13] M. Andrecut and M. K. Ali, Mod. Phys. Lett. B **15**, 1259 (2001).
[14] H. -S. Zeng and L. -M. Kuang, Commun. Theor. Phys. **33**, 11 (2000).
[15] D. Aharonov and A. Ta-Shma, e-print quant-ph/0301023(2003).
[16] E. Biham, O. Biham, D. Biron, M. Grassl, and D. A. Lidar, Phys. Rev. A **60**, 2742 (1999).
[17] G. L. Long, Y. S. Li, W. L. Zhang, and L. Niu, Phys. Lett. A **262**, 27 (1999).
[18] C. Zalka, quant-ph/9902049.
[19] M. Nielsen and I. C. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
[20] A. N. Soklakov and R. Schack, Opt. Spectrosc. **99**, 211 (2005).
[21] I. S. Gradsteyn and I. M. Ryzhik, *Tables of Integrals, Series, and Products* (Academic Press, New York, 1994).