

Removing measurements from quantum walks

Shenggang Ying* and Mingsheng Ying

State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China, and Centre for Quantum Computation and Intelligent Systems (QCIS), Faculty of Engineering and Information Technology, University of Technology, Sydney, NSW 2007, Australia

(Received 11 October 2012; published 31 January 2013)

Quantum walks are very useful tools in designing quantum algorithms. Amplitude amplification is a key technique to increase the success probability of a quantum-walk-based algorithm, and it is quadratically faster than classical probabilistic amplification. However, amplitude amplification only applies to quantum walks with one-shot hitting time, where no measurements except a final one are performed, and not to quantum walks with concurrent hitting time, where measurements happen or absorbing boundaries exist at each step. In this paper, we propose a procedure to modify quantum walks with concurrent hitting time by removing measurements from them. This procedure enables us to use amplitude amplification to design algorithms based on the modified quantum walks which are faster than those based on the original walks with a concurrent hitting time and more robust than those based on the corresponding walks with a one-shot hitting time.

DOI: [10.1103/PhysRevA.87.012337](https://doi.org/10.1103/PhysRevA.87.012337)

PACS number(s): 03.67.Lx, 05.40.Fb, 03.67.Ac

I. INTRODUCTION

Classical random walks are widely employed in developing probabilistic algorithms, such as the famous Schöning algorithm for k -satisfiability (k -SAT) [1]. Essentially, the success probability and time complexity (average success time) of these probabilistic algorithms are just the hitting probability and hitting time of the underlying random walks, respectively. To increase the success probability, the probability amplification technique is frequently used in the design of probabilistic algorithms: if an algorithm A of complexity $O(f(n))$ with input size n has a success probability p , then a new algorithm B of complexity $O(f(n)/p)$ with success probability $O(1)$ can be derived by repeating A $O(1/p)$ times.

A discrete-time quantum walk was defined in [2] as the quantum extension of a classical random walk, and its universality in quantum computation was examined in [3]. A class of quantum search algorithms provides typical examples of quantum-walk-based algorithms. Usually, quantum search algorithms have at least quadratic speedup over their classical counterparts; a representative example is Grover's algorithm [4], which can find the unique desired element in an unordered database of N elements in time $O(\sqrt{N})$. It was shown that quantum-walk-based search algorithms can perform as well as Grover's algorithm [5], and some even better than it in certain situations where Grover's algorithm does not work well [6]. Recently, several other quantum-walk-based algorithms rather than search have also been proposed; for example, quantum-walk algorithms for element distinctness [7] and for the triangle problem [8]. It is worth noting that in some cases, quantum walks can even be exponentially more powerful than classical random walks, that is, quantum walks will hit the target position exponentially faster than classical random walks [9,10].

As in the classical case, the success probability and hitting time of quantum walks are two key factors in the analysis of

quantum-walk-based algorithms. The methods of analyzing success probability can be relatively easily generalized from classical random walks to quantum walks. For example, a quantum type of amplification technique, namely, amplitude amplification, was introduced by Brassard *et al.* [11]: a quantum algorithm B with success probability $O(1)$ can be derived by repeating a quantum algorithm A with probability p $O(1/\sqrt{p})$ times, where no measurements except the final one are performed.

Analyzing the hitting time of quantum walks is much more difficult than in the classical case. Indeed, the notion of the hitting time of a random walk splits into two variants in the quantum setting: the one-shot hitting time and the concurrent hitting time [10,12]. Roughly speaking, the one-shot hitting time is the hitting time of quantum walks without absorbing boundaries (measurements), and the concurrent hitting time is the hitting time of quantum walks with absorbing boundaries (measurements). The hitting times with and without boundaries are the same for classical random walks, but they do not coincide in quantum walks because quantum measurement will change the state of the measured systems. Usually, a quantum-walk-based algorithm can employ an oracle-controlled coin operator to get a desired one-shot hitting time [5,6,13]. Furthermore, we are able to use amplitude amplification to improve the performance of the quantum algorithms in which only one-shot hitting time is considered. However, only classical amplification can be directly used to increase the success probability of quantum algorithms with the concurrent hitting time because the intermediate measurements will alter the amplitudes of the involved systems.

In this paper, we present a novel approach to quantum walks so that amplitude amplification technique can also be used to speed up the concurrent hitting time of quantum walks. The basic idea is as follows: for a given quantum walk \mathcal{W} , we find a new quantum walk \mathcal{W}' that employs no measurements except the final output and has the same concurrent hitting time as the original quantum walk \mathcal{W} . Then amplitude amplification can be applied to \mathcal{W}' instead of \mathcal{W} , leaving the behavior of \mathcal{W}' outside the target states the same as that of \mathcal{W} .

*yingshenggang@gmail.com

The paper is organized as follows: Sec. II contains necessary preliminaries, including quantum walks, their absorbing boundaries, and their hitting times. Our main idea of removing measurements from quantum walks is explained in Sec. III, and some applications are presented in Sec. IV. The advantages of our approach to quantum walks are further discussed in Sec. V.

II. PRELIMINARIES

In this section, we recall some necessary definitions of quantum walks and fix the notations used in the subsequent sections. For more details, we refer to [5,6,10–14].

A. Discrete-time quantum walks

Let $G = (V, E)$ be a d -regular graph, where V is the set of vertices and may be finite, say a hypercube, or infinite, say a lattice, and E is the set of edges. The Hilbert space of a quantum walk on this graph is defined as $\mathcal{H} = \mathcal{H}_p \otimes \mathcal{H}_c$, where

$$\mathcal{H}_p = \text{span}\{|x\rangle : x \in V\}$$

stands for the position space, and

$$\mathcal{H}_c = \text{span}\{|i\rangle : i = 1, \dots, d\}$$

stands for the coin space. A single step of the walk is described by the unitary evolution $U = S \cdot C$, where

$$S = \sum_{i,x} |n_i(x)\rangle\langle x| \otimes |i\rangle\langle i|,$$

$n_i(x)$ maps a node x to its i th adjacent node, $C = (I_p \otimes C_0)$, I_p is the identity operator on \mathcal{H}_p , and C_0 is a unitary operator on \mathcal{H}_c , called the coin operator. Denote the initial state $|\phi_0\rangle \in \mathcal{H}$. Then after t steps, the state of the walk is $|\phi_t\rangle = U^t |\phi_0\rangle$.

B. Absorbing boundary

The quantum walks considered in the previous subsections contain no absorbing boundaries. Now we see how the behavior of walks is changed when absorbing boundaries are inserted. An absorbing boundary is often defined by a projective measurement [14,15]. More precisely, if we set an absorbing boundary at position $v \in V$, then at each step, we do the measurement $\{P_1 = |v\rangle\langle v|, P_0 = I - P_1\}$ in the position space after the evolution operator U is applied. If the outcome is 1, this means we find the particle at position v , then the walk stops. Otherwise it continues. Let

$$\tilde{U} = (P_0 \otimes I_c)U = [(I_p - |v\rangle\langle v|) \otimes I_c]U, \quad (1)$$

where I_c is the identity operator on the coin space \mathcal{H}_c . The probability that the particle is absorbed at time t is

$$p_t = \|(|v\rangle\langle v| \otimes I_c)U\tilde{U}^{t-1}|\phi_0\rangle\|^2, \quad (2)$$

while the state left at time t becomes

$$|\tilde{\phi}_t\rangle = \tilde{U}^t |\phi_0\rangle. \quad (3)$$

Note that here $|\tilde{\phi}_t\rangle$ is not normalized and $\| |\tilde{\phi}_t\rangle \|^2$ is the probability that the outcomes of the first t measurements are all 0.

C. Hitting time

Two definitions of hitting time were introduced by Kempe in [10]. The first one is for quantum walks without absorbing boundaries.

Definition 1: One-shot hitting time. Consider a quantum walk with single-step evolution U . We say that the walk has a (T, p) one-shot $(|\phi_0\rangle, |v\rangle)$ hitting time if the probability of measuring state $|v\rangle$ at time T starting in $|\phi_0\rangle$ is not smaller than p , that is, $\| |\langle v|U^T|\phi_0\rangle|^2 \geq p$.

The second notion of hitting time is defined for quantum walks with absorbing boundaries.

Definition 2: Concurrent hitting time. Consider a quantum walk with single-step evolution U . We say that this walk has a (T, p) concurrent $(|\phi_0\rangle, |v\rangle)$ hitting time if the walk with initial state $|\phi_0\rangle$ and absorbing boundary at position $|v\rangle$ has a probability $\geq p$ of stopping at time $t \leq T$, that is, $\sum_{t \leq T} p_t \geq p$, where p_t is defined by Eq. (2).

D. Quantum-walk-based search algorithms

There are mainly two kinds of quantum-walk-based algorithms, one without measurements and the other with measurements. The basic idea to use a quantum walk without measurements to search the desired state is to replace the original coin operator C with a target-state-controlled coin operator,

$$C' = (I_p - |v\rangle\langle v|) \otimes C_0 + |v\rangle\langle v| \otimes C_1,$$

where v is the target state, and C_1 is usually taken as $-C_0$ or $-I$. The operator C' can be implemented by using an oracle (see [5]). After t steps of evolution $U' = S \cdot C'$, the state of the walk is $|\phi_t\rangle = U'^t |\phi_0\rangle$. Then we can measure the position space to get an answer with a certain probability p .

In applications, a step number T is usually preset and then the success probability depending on T can be computed. For example, for the algorithm presented in [5], we have $T = \sqrt{N}\pi/2$ and $p = 1 - O(1/n)$, where $N = 2^n$; and for the algorithm in [6], $T = O(\sqrt{N \log N})$, $p = \Omega(1/\log N)$, and thus after amplitude amplification, the target can be found with probability $O(1)$ and complexity $O(\sqrt{N \log N})$.

Another kind of quantum-walk-based algorithm performs measurements at each step to see if the desired element is found (e.g., see [10]). In a sense, the first kind is based on quantum walks with a one-shot hitting time, and the second is based on those with a concurrent hitting time.

E. Amplitude amplification

Amplitude amplification is a technique for increasing the success probability by iterations. Let us divide the position Hilbert space \mathcal{H}_p of a quantum walk into the direct sum of two subspaces: the good space

$$\mathcal{H}_1 = \text{span}\{|x\rangle : f(x) = 1\}$$

and the bad space

$$\mathcal{H}_0 = \text{span}\{|x\rangle : f(x) = 0\},$$

where f is the solution function, that is,

$$f(x) = \begin{cases} 1 & x \text{ is a desired position, i.e., a solution,} \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

We further set $S_0 = -2|0\rangle\langle 0| + I$ and $S_f = -I_{\mathcal{H}_1} + I_{\mathcal{H}_0}$. The following very useful theorem was proved in [11].

Theorem 1 Amplitude amplification. Let \mathcal{A} be a quantum algorithm without measurements, with initial state $|\phi_0\rangle = |0\rangle$ and success probability p . Let $\mathcal{D} = -\mathcal{A}S_0\mathcal{A}^{-1}S_f$ and $m = \lceil \pi/4\theta_p \rceil$, where $\sin^2(\theta_p) = p$ and $\theta_p \in [0, \pi/2]$. If we measure $\mathcal{D}^m \mathcal{A}|0\rangle$, then we can get the solution with success probability $\max\{p, 1-p\}$.

Obviously, if p is small, then $\theta_p = \Theta(\sqrt{p})$ and the iteration number $m = O(1/\sqrt{p})$.

III. REMOVING MEASUREMENTS FROM QUANTUM WALKS

As shown in Theorem 1, one can use the amplitude amplification technique to improve the performance of quantum algorithms without measurement. But amplitude amplification cannot directly apply to quantum walks with absorbing boundaries because certain measurements occur between their different steps. The aim of this section is to remove these measurements, leaving the concerned performance of the walks unchanged.

A. Design

Given a quantum walk with position space \mathcal{H}_p , coin space \mathcal{H}_c , single-step evolution U , and absorbing boundary at position $|v\rangle$, we first enlarge the Hilbert space of the quantum walk by adding the step counting space \mathcal{H}_s so that the whole state space is

$$\mathcal{H} = \mathcal{H}_s \otimes \mathcal{H}_p \otimes \mathcal{H}_c,$$

where $\mathcal{H}_s = \text{span}\{|i\rangle : 0 \leq i \leq T\}$, and T is a presumed bound of the step number. The unitary of a single step of the walk becomes

$$V = |0\rangle\langle 0| \otimes U + (I_s - |0\rangle\langle 0|) \otimes I_{p,c}, \quad (5)$$

where $I_{p,c} = I_p \otimes I_c$, and I_s, I_p , and I_c are identity operators on $\mathcal{H}_s, \mathcal{H}_p$, and \mathcal{H}_c , respectively. At each step, after V is applied, we perform an additional controlled unitary on \mathcal{H}_s :

$$W = P \otimes Q \otimes I_c + I_s \otimes (I_p - Q) \otimes I_c, \quad (6)$$

where P is a permutation matrix,

$$P = |0\rangle\langle T| + \sum_{i=0}^{T-1} |i+1\rangle\langle i|,$$

and Q is the projector onto the target subspace of \mathcal{H}_p , i.e., the good space $\mathcal{H}_1 = \text{span}\{|x\rangle : f(x) = 1\}$. The circuit for $W \cdot V$ is displayed in Fig. 1, and it can be further realized by using an oracle, as shown in Fig. 2. In Fig. 2, the oracle U_f is defined

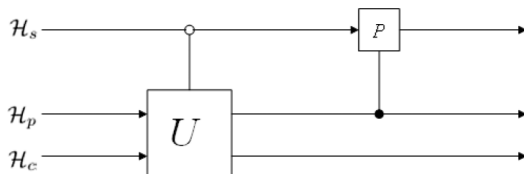


FIG. 1. Circuit of the modified walks.

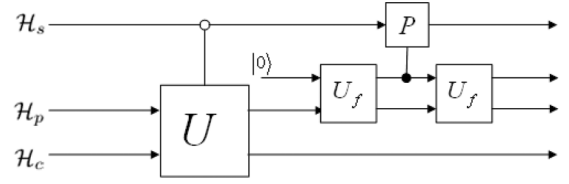


FIG. 2. Circuit of the modified walks for implementation.

by

$$U_f : |x\rangle|y\rangle \rightarrow |x\rangle|y \oplus f(x)\rangle, \quad (7)$$

as in [12], where $f(x)$ is defined by Eq. (4).

B. Evolution

Now the original walk with state space $\mathcal{H}_p \otimes \mathcal{H}_c$ and single-step unitary U is modified to a walk in Hilbert space $\mathcal{H}_s \otimes \mathcal{H}_p \otimes \mathcal{H}_c$ whose single-step evolution is WV . If the modified walk starts in initial state $|\psi_0\rangle$, then its state after t steps is $|\psi_t\rangle = (WV)^t |\psi_0\rangle$. Denote

$$\tilde{U} = (I_{p,c} - Q \otimes I_c)U = (I_{p,c} - |v\rangle\langle v| \otimes I_c)U. \quad (8)$$

It is an operator on $\mathcal{H}_p \otimes \mathcal{H}_c$.

Lemma 1. Let the initial state $|\psi_0\rangle = |0\rangle|\phi_0\rangle$, where $|\phi_0\rangle \in \mathcal{H}_p \otimes \mathcal{H}_c$. Then for any $t \leq T$, the state of the modified walk after t steps is

$$|\psi_t\rangle = |0\rangle(\tilde{U}^t |\phi_0\rangle) + \sum_{i=1}^t |i\rangle(Q \otimes I_c)U\tilde{U}^{t-i} |\phi_0\rangle. \quad (9)$$

Proof. We proceed by induction on t . First, we have the induction basis:

$$\begin{aligned} |\psi_1\rangle &= WV|0\rangle|\phi_0\rangle \\ &= W(|0\rangle\langle 0| \otimes U + (I_s - |0\rangle\langle 0|) \otimes I_{p,c})|0\rangle|\phi_0\rangle \\ &= (P \otimes Q \otimes I_c + I_s \otimes (I_p - Q) \otimes I_c)|0\rangle(U|\phi_0\rangle) \\ &= |1\rangle((Q \otimes I_c)U|\phi_0\rangle) + |0\rangle((I_{p,c} - Q \otimes I_c)U|\phi_0\rangle) \\ &= |1\rangle((Q \otimes I_c)U|\phi_0\rangle) + |0\rangle(\tilde{U}|\phi_0\rangle). \end{aligned}$$

Now we suppose that Eq. (9) is true for $t = m$, that is,

$$|\psi_m\rangle = |0\rangle(\tilde{U}^m |\phi_0\rangle) + \sum_{i=1}^m |i\rangle(Q \otimes I_c)U\tilde{U}^{m-i} |\phi_0\rangle.$$

We observe that

$$\begin{aligned} WV|i\rangle|\phi\rangle &= W(|0\rangle\langle 0| \otimes U + (I_s - |0\rangle\langle 0|) \otimes I_{p,c})|i\rangle|\phi\rangle \\ &= (P \otimes Q \otimes I_c + I_s \otimes (I_p - Q) \otimes I_c)|i\rangle|\phi\rangle \\ &= |i+1\rangle|\phi\rangle \end{aligned}$$

if $0 < i < T$ and $(Q \otimes I_c)|\phi\rangle = |\phi\rangle$. Therefore, if $m+1 \leq T$, it holds that

$$\begin{aligned} |\psi_{m+1}\rangle &= WV|\psi_m\rangle \\ &= WV(|0\rangle(\tilde{U}^m |\phi_0\rangle) + \sum_{i=1}^m |i\rangle(Q \otimes I_c)U\tilde{U}^{m-i} |\phi_0\rangle) \\ &= |1\rangle((Q \otimes I_c)U(\tilde{U}^m |\phi_0\rangle)) + |0\rangle(\tilde{U}(\tilde{U}^m |\phi_0\rangle)) \end{aligned}$$

$$\begin{aligned}
 & + \sum_{i=1}^m |i+1\rangle \langle (Q \otimes I_c) U \tilde{U}^{m-i} | \phi_0 \rangle \\
 & = |0\rangle \langle \tilde{U}^{m+1} | \phi_0 \rangle + \sum_{i=1}^{m+1} |i\rangle \langle (Q \otimes I_c) U \tilde{U}^{m+1-i} | \phi_0 \rangle.
 \end{aligned}$$

So, Eq. (9) is also true for $t = m + 1$, and we complete the proof.

Now we are able to show that outside the target subspace, the behavior of the modified walk is the same as that of the original walk with absorbing boundaries.

Proposition 1. For any $t \leq T$, we have

$$I_s \otimes (I_p - Q) \otimes I_c |\psi_t\rangle = |0\rangle |\tilde{\phi}_t\rangle,$$

where $|\tilde{\phi}_t\rangle$ is given by Eq. (3).

Proof. It can be immediately derived from Eq. (9) by a step of direct calculation.

On the other hand, the next theorem clarifies the relationship among the concurrent hitting time of the the original walk, the one-shot hitting time of the modified walk, and the concurrent hitting time of the modified walk

Theorem 2. The following three statements are equivalent to each other.

- (1) The original walk has a (T, p) concurrent $(|\phi_0\rangle, |v\rangle)$ hitting time.
- (2) The modified walk has a (T, p) one-shot $(|0\rangle|\phi_0\rangle, |v\rangle)$ hitting time.
- (3) The modified walk has a (T, p) concurrent $(|0\rangle|\phi_0\rangle, |v\rangle)$ hitting time.

Proof. Put $Q = |v\rangle\langle v|$. It suffices to show that all clauses, 1, 2, and 3, are equivalent to

$$\sum_{i \leq T} p_i \geq p. \tag{10}$$

Equivalence between Clause 1 and Eq. (10) is simply by definition.

Now we prove that Clause 2 \Leftrightarrow Eq. (10). From Lemma 1, the one-shot hitting probability of the modified walk at time T is

$$\begin{aligned}
 p\{T\} & = \|I_s \otimes Q \otimes I_c |\psi_T\rangle\|^2 \\
 & = \left\| \sum_{i=1}^T |i\rangle \langle (Q \otimes I_c) U \tilde{U}^{T-i} | \phi_0 \rangle \right\|^2 \\
 & = \sum_{i=1}^T \| \langle (Q \otimes I_c) U \tilde{U}^{T-i} | \phi_0 \rangle \|^2 \\
 & = \sum_{i=1}^T p_{T-i+1} = \sum_{i=1}^T p_i,
 \end{aligned}$$

where p_i is defined as in Eq. (2). Thus from the definition, we have that Clause 2 $\Leftrightarrow \sum_{i \leq T} p_i \geq p$.

Finally, we show that Clause 3 \Leftrightarrow Eq. (10). Since

$$(I_s \otimes (I_p - Q) \otimes I_c) W V = (I_s \otimes (I_p - Q) \otimes I_c) V$$

and $V|0\rangle|\phi\rangle = |0\rangle \otimes U|\phi\rangle$, we have

$$(I_s \otimes (I_p - Q) \otimes I_c) V|0\rangle|\phi\rangle = |0\rangle \otimes \tilde{U}|\phi\rangle.$$

Thus, by induction, it is easy to show that the probability that the particle of the modified walk is found at time t is just p_t , defined in Eq. (2), if the initial state is $|\psi_0\rangle = |0\rangle|\phi_0\rangle$, and there is an absorbing boundary at position v . Consequently, Clause 3 \Leftrightarrow Eq. (10).

Finally, we are ready to present the main result of this section, showing how the amplitude amplification technique can be employed to speed up the algorithms based on quantum walks with a concurrent hitting time. Suppose that \mathcal{W} is a quantum walk with absorbing boundaries and has a (T, p) concurrent $(|0\rangle, |v\rangle)$ hitting time. The number of evolution steps is up to T . Repeating \mathcal{W} $O(1/p)$ times will lead to a algorithm with success probability $O(1)$ and time complexity $O(T/p)$.

Theorem 3. Let \mathcal{A} denote the unitary evolution of the walk modified from \mathcal{W} . Then a new algorithm $\mathcal{B} = \mathcal{D}^m \mathcal{A}|0\rangle$ with time complexity $O(T/\sqrt{p})$ and success probability $O(1)$ can be derived, where \mathcal{D}, m are defined as in Theorem 1.

Proof. From Proposition 1 and Theorem 2, we see that the modified walk has a (T, p) one-shot hitting time, while its behavior outside the target subspace is the same as that of the original walk with absorbing boundaries. Since no measurement, excluding the final one, is performed on the modified walk, we can employ amplitude amplification to speed up the algorithm based on the modified walk, and algorithm \mathcal{B} can be derived.

IV. APPLICATIONS

In this section, we present some applications of the modified quantum walks to show how the main result from the last section works.

A. One-dimensional quantum walks with absorbing boundaries

We first consider one-dimensional quantum walks with absorbing boundaries [14,15]. Let the position space $\mathcal{H}_p = \text{span}\{|i\rangle : i \in \mathbb{Z}\}$ and the coin space $\mathcal{H}_c = \text{span}\{|R\rangle = |0\rangle, |L\rangle = |1\rangle\}$. The unitary evolution of every step is defined as $U = S \cdot (I_p \otimes C_0)$, where the coin operator

$$C_0 = \begin{pmatrix} \sqrt{a} & \sqrt{b} \\ \sqrt{b} & -\sqrt{a} \end{pmatrix}, \tag{11}$$

$a, b \in [0, 1]$, $a + b = 1$, and the shift operator

$$S = \sum |x+1\rangle\langle x| \otimes |R\rangle\langle R| + \sum |x-1\rangle\langle x| \otimes |L\rangle\langle L|. \tag{12}$$

The absorbing boundary is set at the origin, i.e., $Q = |0\rangle\langle 0|$. The eventual absorbing probability can be calculated by the method of generating function; for details, we refer to [15].

Now let Algorithm A0 stand for the algorithm directly based on this quantum walk: Algorithm A0 runs \tilde{U} again and again until the the target $|v\rangle$ is found, where \tilde{U} is obtained from U and $|v\rangle$ according to Eq. (1). On the other hand, we apply the amplitude amplification to derive a new algorithm based on the walk modified from the above walk. This new algorithm is described as follows.

Algorithm A1. We set $\mathcal{A} = (WV)^T$, where W and V are defined by Eq. (6) and Eq. (5), respectively. The one-step evolution in V is $U = S(I_p \otimes C_0)$, where C_0 is given by Eq. (11), and S by Eq. (12). We further set $S_0 = -2|\psi_0\rangle\langle\psi_0| + I$ in Theorem 1, where I is the identity operator on $\mathcal{H}_s \otimes \mathcal{H}_p \otimes \mathcal{H}_c$ and $|\psi_0\rangle$ is the initial state. Finally, the algorithm can be expressed by $\mathcal{B} = \mathcal{D}^m \mathcal{A}|\psi_0\rangle$, where \mathcal{D} and m are defined as in Theorem 1. A routine calculation yields $m = \pi/(4 \arcsin \sqrt{\text{Prob}_0})$, where Prob_0 denotes the success probability of $\mathcal{A}|0\rangle$, i.e., p in Theorem 1 and Theorem 3. Obviously, Prob_0 and m are dependent on T , $|\psi_0\rangle$, and a in C_0 , in the above algorithm.

In order to see that Algorithm A1 can speed up the success process, we compare it with Algorithm A0 as well as the following algorithm.

Algorithm A2. This algorithm runs the original walk with an absorbing boundary at the origin up to T steps. If the particle is not found, we restart the walk. To state it explicitly, we set \tilde{U} as in Eq. (1) and run \tilde{U} on the initial state $|\psi_0\rangle$ T times. Then we reset the state to be the initial state and apply \tilde{U} T times again. We repeat this process until the particle is found.

The success probabilities versus the number of oracle calls are plotted in Figs. 3–8. A part of Fig. 8 is enlarged as Fig. 9 so that it can be seen more clearly. In each figure, the red line is the success probability of Algorithm A1; in other words, it is the success probability of finding the particle at the origin, if we perform a measurement at that step. The dashed (blue) line with crosses is that of the success probability of Algorithm A2, and the dashed (green) line with circles is the success probability of Algorithm A0.

It is worth noting that the solid (red) lines are plotted every two steps, since each step of Algorithm A1 costs two oracle calls, and in contrast, every step of Algorithm A2 or Algorithm A0 only costs one oracle call. Furthermore, $\mathcal{B} = \mathcal{D}^m \mathcal{A}|\psi_0\rangle$

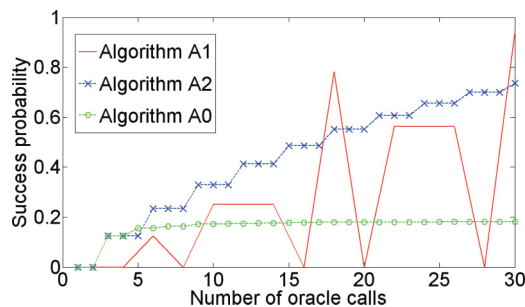


FIG. 3. (Color online) The success probability versus the number of oracle calls. The initial conditions are $a = 0.5$, $|\psi_0\rangle = |03R\rangle$, and $T = 3$. Thus the initial success probability $\text{Prob}_0 = 0.125$ and $m = 2$. The solid (red) line, dashed (blue) line with crosses, and dashed (green) line with circles are, successively, success probabilities of Algorithm A1, Algorithm A2, and Algorithm A0, respectively. The dashed (green) line with circles is plotted as a scale. The solid (red) line is indeed the success probability of finding the particle at the origin, if we perform a measurement at that step. The solid line reaches 0 at some steps because the original walk (before amplitude amplification applies to it) only reaches the origin once, i.e., $T = x$, where x is the initial position. This will occur again in some other figures. This figure shows that Algorithm A1 can perform as well as Algorithm A2 when m is small.

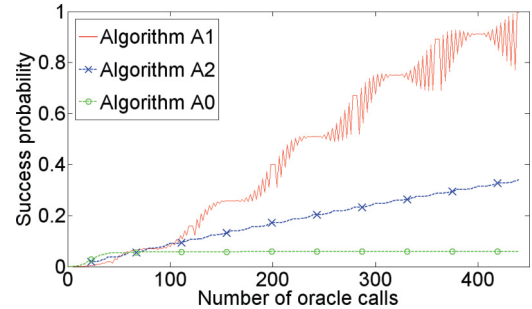


FIG. 4. (Color online) Success probability versus number of oracle calls. Initial conditions are $a = 0.01$, $|\psi_0\rangle = |03R\rangle$, and $T = 20$. Thus $\text{Prob}_0 = 0.018688$, and $m = 5$. Lines are as in Fig. 3.

employs totally $2m + 1$ times of \mathcal{A} or \mathcal{A}^{-1} . Therefore, the total number of oracle calls in Algorithm A1 is $4mT + 2T$.

From Figs. 3–8, m increases from 2 to 466, while Prob_0 decreases. We observe from these figures that Algorithm A1 works well. Whenever Prob_0 is small, Algorithm A1 is much faster. It can reach a success probability close to 1, while Algorithm A2 can only reach a success probability of $1 - (1 - \text{Prob}_0)^{4m+2} = \Theta(\sqrt{\text{Prob}_0})$. This clearly shows that the modified walk and amplitude amplification can work well together.

B. Speeding up algorithms based on quantum walks with a concurrent hitting time

If a quantum walk has a $(p(n), 1/q(n))$ concurrent hitting time, it will take $O(q(n))$ times of classical amplification to reach a success probability of $O(1)$, which leads to a total complexity of $O(p(n)q(n))$. With Theorem 2, we are able to speed up such an algorithm through amplitude amplification. Let us consider an example from [10]. Suppose that G is an n -dimensional hypercube with $N = 2^n$ nodes. The task is to travel from one corner to its opposite corner. A quantum-walk-based algorithm is as follows: the coin space is $\mathcal{H}_c = \text{span}\{|1\rangle, \dots, |n\rangle\}$, and the shift operator

$$S = \sum_x \sum_{i=1}^n |n_i(x)\rangle \langle x| \otimes |i\rangle \langle i|, \quad (13)$$

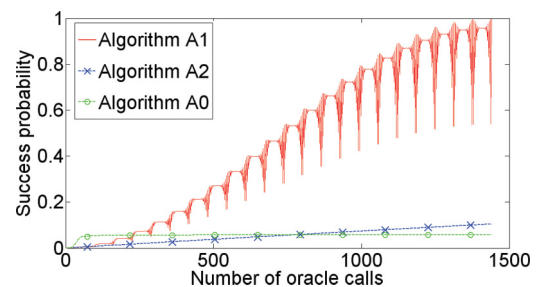


FIG. 5. (Color online) Success probability versus number of oracle calls. Initial conditions are $a = 0.01$, $|\psi_0\rangle = |04R\rangle$, and $T = 16$. Thus $\text{Prob}_0 = 0.00123$, and $m = 22$. Lines are as in Fig. 3. In this and subsequent figures, the dashed (blue) line with crosses and dashed (green) line with circles are close to the bottom. Algorithm A1 becomes considerably faster than Algorithms A2 and A0.

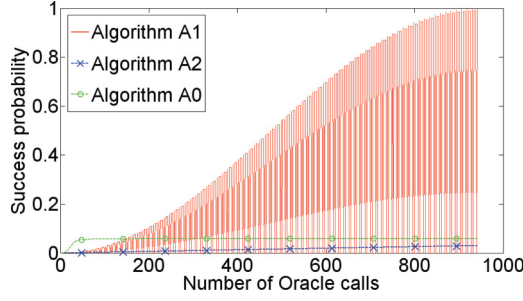


FIG. 6. (Color online) Success probability versus number of oracle calls. Initial conditions are $a = 0.01$, $|\psi_0\rangle = |03R\rangle$, and $T = 3$. Thus $\text{Prob}_0 = 9.9 \times 10^{-5}$, and $m = 78$. Lines are as in Fig. 3. Here, it is obvious, Algorithm A1 is much faster than Algorithms A2 and A0.

where $n_i(x)$ is x 's i th adjacent node, the coin operator

$$C = I_p \otimes G = I_p \otimes (D - I_c), \quad (14)$$

and D is the matrix $(d_{ij})_{n \times n}$ with $d_{ij} = 2/n$ for all $1 \leq i, j \leq n$. The walk starts in the initial state

$$|\phi_0\rangle = |\vec{0}\rangle \sum_i \frac{1}{\sqrt{n}} |i\rangle.$$

The target node is the opposite vertex $\vec{1}$ of the initial vertex $\vec{0}$. Then this walk has a $(\frac{\pi}{2}n, \Omega(\frac{1}{n \log^2 n}))$ concurrent hitting time. After the amplifications for achieving the success probability $O(1)$, the total complexity is $O(n^2 \log^2 n)$.

The above quantum-walk-based algorithm is not as good as the deterministic classical algorithm of complexity $\Theta(n^2)$ given in [9]. Using Theorem 3, however, we can derive a new quantum algorithm with complexity $O(n\sqrt{n} \log n)$ and thus it is better than the classical deterministic one. This new algorithm is based on the quantum walk modified from the above one by simply applying the procedure presented in Sec. III A with $T = \frac{\pi}{2}n$.

C. Developing a new algorithm for spatial search

Our approach can be used not only to speed up the existing algorithms, but also to design new algorithms. Let us consider the problem: Find the desired element in a database of $\sqrt{N} \times \sqrt{N}$ elements distributed in a two-dimensional grid. At each

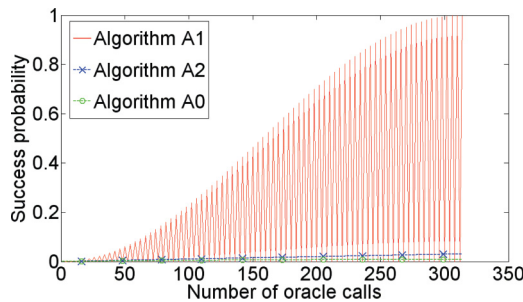


FIG. 7. (Color online) Success probability versus number of oracle calls. Initial conditions are $a = 0.9999$, $|\psi_0\rangle = |01R\rangle$, and $T = 1$. Thus $\text{Prob}_0 = 1 \times 10^{-4}$, and $m = 78$. Lines are as in Fig. 3. Here, it is obvious, Algorithm A1 is much faster than Algorithms A2 and A0.

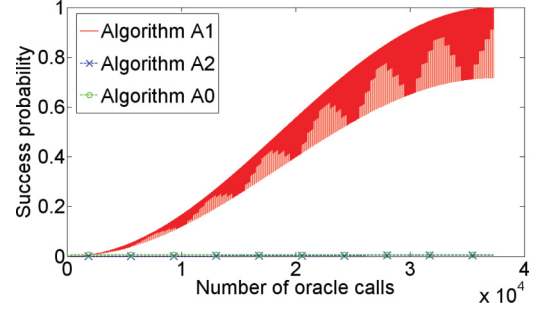


FIG. 8. (Color online) Success probability versus number of oracle calls. Initial conditions are $a = 0.0001$, $|\psi_0\rangle = |03R\rangle$, $T = 20$. Thus $\text{Prob}_0 = 2.84 \times 10^{-6}$, and $m = 466$. Lines are as in Fig. 3. The dashed (blue) line with crosses and dashed (green) line with circles are both at the bottom. This shows that Algorithm A1, derived from the modified walk and amplitude amplification, works well, and it should be a quadratic speedup.

step, we can query the current position or move to an adjacent node [6,16].

An algorithm of complexity $O(\sqrt{N} \log N)$ is presented in [6], and it is based on a quantum walk on the grid with a periodic boundary condition. The shift operator of the walk is different from that defined in Sec. II A. More precisely, let $\mathcal{H}_p = \text{span}\{|i\rangle|j\rangle : 0 \leq i, j \leq \sqrt{N} - 1\}$ with the periodic boundary conditions $|i\rangle|j\rangle = |i\rangle$ and $\mathcal{H}_c = \text{span}\{|i\rangle : i = 0, 1, 2, 3\}$. Let the unitary evolution

$$U_1 = S_{ff}C. \quad (15)$$

The shift operator S_{ff} is defined as

$$S_{ff} = (I \otimes P_{\text{swap}})S_m, \quad (16)$$

where S_m is the ordinary shift operator

$$S_m = \sum_{i,j,k} |i + x_k, j + y_k\rangle \langle i, j| \otimes |k\rangle \langle k|, \quad (17)$$

and P_{swap} is a permutation matrix,

$$P_{\text{swap}} = \begin{pmatrix} & & & 1 \\ & & 1 & \\ & 1 & & \\ 1 & & & \end{pmatrix}. \quad (18)$$

In Eq. (17), $(x_0, y_0) = (1, 0)$, $(x_1, y_1) = (0, 1)$, $(x_2, y_2) = (0, -1)$, and $(x_3, y_3) = (-1, 0)$ denote four different directions, respectively, in the grid. The coin operator is defined as

$$C = I_p \otimes G = I_p \otimes (D - I_c), \quad (19)$$

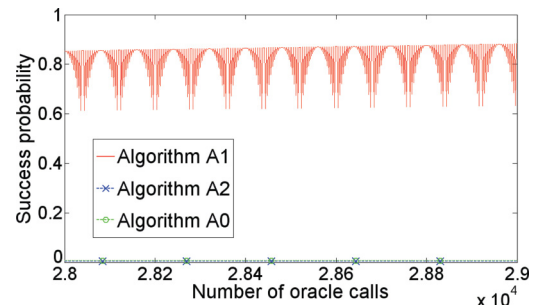


FIG. 9. (Color online) Part of Fig. 8.

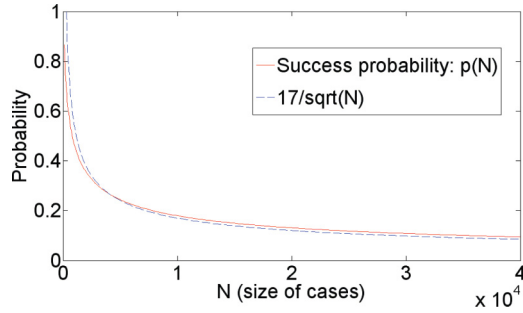


FIG. 10. (Color online) The solid (red) line is the success probability $p(N)$, and the dashed (blue) line is $17/\sqrt{N}$. Comparing these two lines, we observe that $p(N) = \Omega(1/\sqrt{N})$.

where D is the matrix $(d_{ij})_{2 \times 2}$ with $d_{ij} = \frac{1}{2}$ for $i, j = 1, 2$. Let the initial state be

$$|\Phi_0\rangle = \frac{1}{2\sqrt{N}} \sum |i, j\rangle \otimes |k\rangle. \quad (20)$$

Note that replacing $U_1 = S_{ff}C$ with $U_2 = S_m C(I_p \otimes P_{\text{swap}})$ makes no difference because we have $|\Phi_0\rangle = I \otimes P_{\text{swap}}|\Phi_0\rangle$. After $T = O(\sqrt{N} \log N)$ steps of evolution, one measures the position space. The probability of getting the desired answer is $\Omega(1/\log N)$. Thus after amplitude amplification, the quantum walk can find the target state with probability $O(1)$ in time $O(\sqrt{N} \log N)$.

Now we can derive a new algorithm simply by setting $T = 5\sqrt{N}$ and replacing the unitary U_1 in the above quantum walk with $U_2 = S_m \cdot C \cdot (I \otimes P_{\text{swap}})$. Then the new walk has a $(5\sqrt{N}, p(N))$ concurrent hitting time. It is very hard to calculate the probability $p(N)$, so we follow [13] and visualize it as the solid (red) line in Fig. 10. The dashed (blue) line is $17/\sqrt{N}$, and it is very close to the solid (red) one. This hints that $p(N) = \Omega(1/\sqrt{N})$. Applying Theorem 3, we know that the new algorithm has time complexity $O(N^{\frac{3}{4}})$ and success probability $O(1)$. Although the new algorithm is not as fast as the one developed in [6], it is faster than the classical algorithm with $\Omega(N)$. Moreover, the new algorithm has its advantages of robustness, as discussed in Sec. V D.

V. DISCUSSION

We proposed to modify a quantum walk with a concurrent hitting time by removing measurements from it so that the amplitude amplification technique can be used to design quantum algorithms based on it. Some applications of this approach were presented in Sec. IV. In this section, we further explain the advantages of our approach.

A. Deferred measurements

It is well known that all the measurements in a quantum computation can be deferred to the end of the computation [17]. However, if we apply this approach of deferring measurement to a quantum walk, then t control qubits must be added after t steps, and thus the following problems will be caused.

(1) The circuit of a single step of the modified walk will be significantly different from that of the original walk.

(2) After t steps, the complexity of the circuit will become $O(t^2)$. Thus, if the original walk has a $(p(n), 1/q(n))$ concurrent hitting time, then the total complexity of the walk after amplitude amplification will be $O(p^2(n)\sqrt{q(n)})$, which is really bad whenever $p(n) > \sqrt{q(n)}$.

Both of the above problems are serious when a quantum walk is employed in a quantum algorithm. It is obvious that these problems do not occur in the modified quantum walks defined in Sec. III A.

B. Dimension of \mathcal{H}_s

An extra Hilbert space \mathcal{H}_s is added in the modified walk. If the dimension $\dim(\mathcal{H}_s)$ of \mathcal{H}_s is too large, then the modified walk will bring some unacceptable overhead of complexity. Fortunately, in most applications the additional dimension $\dim(\mathcal{H}_s)$ in the modified walk is small. More precisely, when we consider a quantum-walk-based search algorithm, the number T of steps is usually smaller than the number N of vertices in the searching space, and $\dim(\mathcal{H}_s) = O(T)$. If this is not the case, i.e., $T > N$, a quantum algorithm is not essential because we can simply check all the N possible solutions one by one to find the answer. For example, in the algorithm in Sec. IV B, we have $\dim(\mathcal{H}_s) = O(n) = O(\log N)$, and in the second algorithm, in Sec. IV C, $\dim(\mathcal{H}_s) = O(\sqrt{N})$. Thus the space \mathcal{H}_s will bring no additional factor to the time complexity.

C. Quadratic speedup

Theorem 3 shows that amplitude amplification can be employed for designing algorithms based on quantum walks with absorbing boundaries, if we modify these walks with the procedure given in Sec. III A. Suppose that we already have a quantum walk with a $(p(n), 1/q(n))$ concurrent hitting time. The classical probability amplification enables us to design an algorithm based on this quantum walk with time complexity $O(p(n)q(n))$. Then we can modify the walk and use amplitude amplification to derive an algorithm with time complexity $O(p(n)\sqrt{q(n)})$. Only this quadratic speedup may not be good enough. Indeed, we have not found any useful algorithm based on a quantum walk with concurrent hitting, which is faster than the one based on the same quantum walk with a one-shot hitting time. Together with the robustness considered in Sec. V D, however, the quadratic speedup makes the approach of removing measurements from quantum walks really interesting.

D. Robustness

Noise in the physical implementation of quantum walks is inevitable. In this section we consider the robustness of modified quantum walks against noise and show that replacing the step evolution U of a modified quantum walk with another unitary U' approximate to U makes only a very small or even no difference to the time complexities of the algorithm based on the walk. To this end, let us first compare the two algorithms in Sec. IV C.

Let Algorithm S1 denote the first algorithm, which is based on a quantum walk without measurements. As shown in [6,13], the amplitude or probabilities of this walk at the target position are periodic. If a small error happens in the

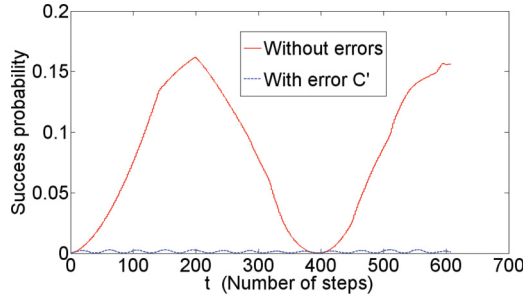


FIG. 11. (Color online) Probability at target position of Algorithm S1, before amplitude amplification. The solid (red) line is the original probability of Algorithm S1 at the target position, while the dashed (blue) line is probability of Algorithm S1 with error (C'). $N = 10000$. The horizontal axis is the number of steps, while the vertical axis is the probability. This figure shows that with error C' , the success probability of Algorithm S1 becomes very small. The success probability at the desired position of the algorithm with error is still periodic, but the period becomes very different from that without error and the peaks become very small.

coin operator C , say, C becomes $C' = I \otimes (QGQ^\dagger)$, where $Q = \text{diag}\{1, \omega, 1, 1\}$ and $\omega = e^{0.1i} \approx 0.9950 + 0.0998i$, then as shown in Fig. 11, the algorithm does not work. In fact, we see that the success probability at the desired position of the algorithm with error is still periodic, but the period becomes very different from that without error and the peaks become very small.

Additional evidence is shown in Fig. 12, where the solid (red) line is the success probability of Algorithm S1 before amplitude amplification without errors, the solid (magenta) line with crosses is the success probability after amplitude amplification without errors, the dashed (blue) line (almost at the bottom) is the success probability before amplitude amplification with error (C'), and the dashed (green) line with circles is the success probability with error (C'). The number m

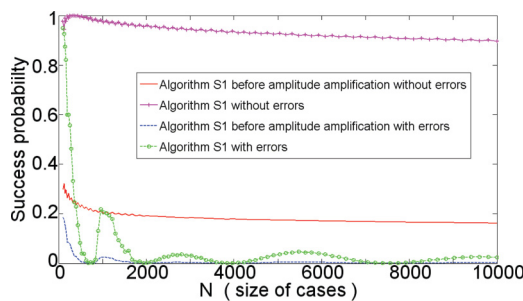


FIG. 12. (Color online) Success probabilities of different cases depending on N . The horizontal axis is the size of the cases (problems), while the vertical axis is the success probability. The solid (red) line is the success probability of Algorithm S1 before amplitude amplification without errors, the solid (magenta) line with crosses is the success probability after amplitude amplification without errors, the dashed (blue) line (almost at the bottom) is the success probability before amplitude amplification with error (C'), and the dashed (green) line with circles is the success probability with error (C'). The number m of Algorithm S1 with errors is set to be the same as the original one. Since the preset value of m is far too small, amplitude amplification only increases the success probability a little. The eventual success probability is very small and irregular and, thus, useless.

of Algorithm S1 with errors is set to be the same as the original one. This is because, before running the algorithm, we might not be able to know what kind of error would happen and thus might not be able to change m .

We can see that the success probability before amplitude amplification decreases considerably, from $\Omega(1/\log N)$ to nearly 0. In addition, the success probability after amplitude amplification decreases from $O(1)$ to nearly 0. Although amplitude amplification still increases the success probability, the following two facts makes the eventual success probability small and useless.

(1) First, the success probability before amplitude amplification [i.e., dashed (blue) line in Fig. 12] becomes nearly 0. Thus the preset value of m is far too small, and amplitude amplification is not able to enlarge the success probability to an expected value.

(2) Second, it seems that the solid (red) line and the dashed (blue) line in Fig. 12 are entirely unrelated to each other. Thus we are not able to know what the eventual success probability with error will be.

The case of the second algorithm with the modified quantum walk is quite different: the algorithm with error still works. Let Algorithm S2 denote this algorithm. Its performance is shown in Fig. 13. In this figure, the lines are the success probabilities of Algorithm S2 in different cases with different N 's, as in Fig. 12. And the (red) line with crosses is half of the solid (red) line. From this figure, we can see that the success

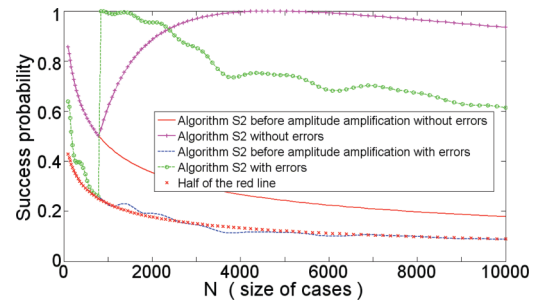


FIG. 13. (Color online) Success probability of different cases depending on N . The horizontal axis is the size of the cases (problems), while the vertical axis is the success probability. The solid (red) line is the success probability of Algorithm S2 before amplitude amplification without errors, the solid (magenta) line with crosses is the success probability after amplitude amplification without errors, the dashed (blue) line is the success probability before amplitude amplification with error (C'), and the dashed (green) line with circles is the success probability with error (C'). The number m of Algorithm S2 with errors is set to be the same as the original one. At about $N = 800$, the dashed (green) line with circles and the solid (magenta) line with pluses increase suddenly, because m increases from 0 to 1. The dashed (blue) line shows that the success probability before amplitude amplification with error C' is half the original one. Thus after amplitude amplification, it is still $\Theta(1)$. This is also shown by the dashed (green) line with circles. It will first coincide with the dashed (blue) line for $m = 0$, then become bigger than the solid (magenta) line with crosses for $m = 1$ and big initial success probability p , and bigger than half the (magenta) line with crosses for $m = 1$ and small success probability p [solid (red) line]. (Note that in this figure $m = 0$ or $m = 1$.) Therefore this error has only a small or even no influence on Algorithm S2.

probability before amplitude amplification without error is nearly half that with error. Thus after amplitude amplification, the success probability is $\Theta(1)$, and the time complexity $O(N^{\frac{3}{2}})$ still holds. This can be verified by a simple calculation from Eq. (8) in [11]. Indeed, it follows from Eq. (8) in [11] that the eventual success probability is $\sin^2((2m+1)\theta_p)$. If the initial success probability becomes $p' \sim p/2$ and m remains unchanged, $\sin^2((2m+1)\theta_p)$ will first become p' for $m=0$, then bigger than $\sin^2((2m+1)\theta_p)$ for $m=1$ and big p , bigger than 0.5 times $\sin^2((2m+1)\theta_p)$ for $m=1$ and small p , and, finally, similar to $\sin^2(\pi/2\sqrt{2}) \sim 0.8$ for very small p . Numerical evidence is also shown in Fig. 13: the dashed (green) line with circles is $\Theta(1)$, as well as the (magenta) line with crosses. Therefore, Algorithm S2 is much more robust than Algorithm S1.

It is reasonable to imagine that with some other small errors, Algorithm S2 still works well but Algorithm S1 does not. Moreover, the algorithms in [5,6,13] are very similar to Algorithm S1, and the time T to perform a measurement for them must be known exactly. Thus, if there are some errors in their evolution unitary, then these algorithms do not work. However, we can improve them with modified walks so that they become much more robust than quantum walks with one-shot hitting-time-based algorithms.

ACKNOWLEDGMENTS

We are grateful to Dr. Yuan Feng, Dr. Runyao Duan, and Dr. Tao Zhou for helpful discussions.

-
- [1] U. Schöning, in *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, FOCS'99* (IEEE Computer Society Press, New York, 1999), pp. 410–414.
- [2] Y. Aharonov, L. Davidovich, and N. Zagury, *Phys. Rev. A* **48**, 1687 (1993).
- [3] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon, *Phys. Rev. A* **81**, 042330 (2010).
- [4] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, STOC'96* (ACM Press, New York, 1996), pp. 212–219.
- [5] N. Shenvi, J. Kempe, and K. Birgitta Whaley, *Phys. Rev. A* **67**, 052307 (2003).
- [6] A. Ambainis, J. Kempe, and A. Rivosh, in *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA'05* (ACM Press, New York, 2005), pp. 1099–1108.
- [7] A. Ambainis, *SIAM J. Comput.* **37**, 210 (2007).
- [8] F. Magniez, M. Santha, and M. Szegedy, *SIAM J. Comput.* **37**, 413 (2007).
- [9] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, STOC'03* (ACM Press, New York, 2003), pp. 59–68.
- [10] J. Kempe, *Probab. Theory Relat. Fields* **133**, 215 (2005).
- [11] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, *Quantum Comput. Quantum Inform. Sci., AMS Contemp. Math Ser. Millen.* **305**, 53 (2002).
- [12] S. E. Venegas-Andraca, arXiv:1201.4780.
- [13] N. Lovett, D. Mosby, D. Stockton, and V. Kendon, arXiv:1010.4705.
- [14] A. Ambainis, E. Bach, A. Nayak, A. Vishwanath, and H. Watrous, in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing, STOC'01* (ACM Press, New York, 2001), pp. 37–49.
- [15] E. Bach, S. Coppersmith, M. P. Goldschen, R. Joynt, and J. Watrous, *J. Comput. Syst. Sci.* **69**, 562 (2004).
- [16] A. Ambainis, *SIGACT News* **35**, 22 (2004).
- [17] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).