Optimized quantum random-walk search algorithms on the hypercube

V. Potoček,[1] A. Gábris,[1,2] T. Kiss,[2] and I. Jex[1]

[1]*Department of Physics, FJFI ČVUT, Břehová 7, 115 19 Praha 1—Staré Město, Czech Republic*

[2]*Research Institute for Solid State Physics and Optics, Hungarian Academy of Sciences, P.O. Box 49, H-1525 Budapest, Hungary*

(Received 8 July 2008; revised manuscript received 16 December 2008; published 28 January 2009)

Shenvi, Kempe, and Whaley's quantum random-walk search (SKW) algorithm [Phys. Rev. A **67**, 052307 (2003)] is known to require $O(\sqrt{N})$ number of oracle queries to find the marked element, where N is the size of the search space. The overall time complexity of the SKW algorithm differs from the best achievable on a quantum computer only by a constant factor. We present improvements to the SKW algorithm which yield a significant increase in success probability, and an improvement on query complexity such that the theoretical limit of a search algorithm succeeding with probability close to one is reached. We point out which improvement can be applied if there is more than one marked element to find.

DOI: 10.1103/PhysRevA.79.012325 PACS number(s): 03.67.Ac, 03.67.Lx, 42.50.Ex, 05.40.Fb

I. INTRODUCTION

In the pioneering paper [1] Shenvi, Kempe, and Whaley (SKW) demonstrated that a useful quantum algorithm can be designed based on quantum random walks. This quantum random-walk search algorithm (the SKW algorithm) can be used to find a vertex of a hypercube that is marked by an oracle. Although the number of oracle calls needed by the SKW algorithm scales with the size of the search space similarly to the Grover search [2], its principle of operation is significantly different. Since the pioneering work a variety of quantum algorithms have been proposed utilizing quantum random walks (see, for example, [3,4]). The SKW algorithm may be divided into a quantum part, and a simple classical protocol in which the former is embedded. The quantum part is a perturbed Grover walk on a hypercube started from an equally weighted superposition of initial states and iterated for a given number of steps, to be followed by a measurement on the output state to find the marked vertex. The perturbation of the Grover coin is derived from the oracle, which is used to introduce position dependence into the coin operator. In this paper, we shall use the term SKW quantum random walk to refer to this special quantum random walk. As shown in [1] the SKW quantum random walk yields the marked vertex with probability strictly less than $1/2$; therefore, it is necessary to embed it into a classical protocol to find the marked vertex with certainty, or use an amplitude amplification scheme [5,6]. The classical protocol of the SKW algorithm is relatively simple: a measurement is made on the final state of the SKW quantum random walk, then its result is verified by querying the oracle directly. By repeating the algorithm and these two steps a sufficient number of times, we can make sure that the marked element is found with an arbitrary small failure probability. Applying an amplitude amplification scheme would provide a more efficient way for increasing the success probability; however, its use would mean departure from the quantum random-walk paradigm.

The overhead caused by repeating the quantum random walk several times, although contributing only a constant factor to the time complexity, can be a considerable source of difficulties in certain experimental scenarios. In the present paper we present modifications to the SKW algorithm which allow significant reduction of the number of necessary repetitions. We note that in two dimensions the spatial search algorithm by Ambainis, Kempe, and Rivosh [7] also yields the target vertex after one run with probability less than one, i.e., with only $\Theta(1/\sqrt{\ln N})$. Recently, Tulsi [8] has proposed improvements to this algorithm, which allow the finding of the target vertex with probability 1 after one run. The speedup in [8] has been achieved by introducing an ancilla qubit into the computational space, which is similar in spirit to our improvement modification described in Sec. III, which uses an additional coin dimension. Improvements of quantum walk-based searches have been studied also by other authors. In [9] an optimization dedicated to the scattering random-walk implementation [10–12] has been proposed, related to the findings we describe in Sec. II. In [13] the authors discussed the optimization of the quantum walk on a line by varying the coin operator parameters.

In Sec. II we prove that the final state of the SKW quantum walk consists mainly of the target vertex and its next neighbors, and present modifications to the algorithm which exploit this property. These modifications can be used to reduce the number of repetitions of the SKW quantum walk, and to reduce the number of independent verification queries to the oracle. We note that the task of verification may be problematic for certain implementations, e.g., in a spatial search implementation where a vertex being marked is a local property and not a property given by an oracle. Such additional costs have been considered in Ref. [14] in connection with quantum walks.

Based on the SKW algorithm we develop an algorithm in Sec. III that displays query complexity $1/\sqrt{2}$ of the original, thus the theoretically lowest for a search algorithm with a success probability close to 1 [15]. Our improvement is founded on the bipartite nature of the SKW quantum random walk, and we arrive at its final form after several steps. We note that some of these intermediate steps may be useful improvements in their own right, depending on the actual physical implementation.

In Sec. IV we outline the conditions under which the optimizations introduced in Sec. III can be used to find multiple marked vertices. Finally, in Sec. V we conclude our results.

II. IMPROVING SUCCESS PROBABILITY BY CONSIDERING NEXT NEIGHBORS

In this section we describe a property of the SKW quantum walk that can be used to boost the probability of finding the marked vertex by doing a proper measurement on its final state. Let $\mathcal{C}_n=(V_n,E_n)$ denote the graph of the n-dimensional hypercube. The argumentation of the present paper relies heavily on the concept of the Hamming weight and the parity of an integer, which can be easily related to each other. The Hamming weight of an integer is the number of 1's in its binary string representation $\vec{x}$, and shall be denoted by $|\vec{x}|$ in this paper. The parity of $\vec{x}$ is then simply $|\vec{x}| \bmod 2$. A related concept is the Hamming distance of two integers, say $\vec{x}$ and $\vec{y}$, that is defined as $|\vec{x}\oplus\vec{y}|$, where $\oplus$ denotes the bitwise addition modulo 2 operator. Following the notation of earlier work [1,16], the vertices V_n of the hypercube are labeled by integers $\vec{x}=0,\ldots,2^n-1$ in such a way that the Hamming distance between any two vertices connected by an edge is exactly 1. The SKW quantum walk takes place on the product Hilbert space $\mathcal{H}^{C_n}\otimes\mathcal{H}^{V_n}$ where $\mathcal{H}^{V_n}$ is the $N=2^n$-dimensional Hilbert space representing the vertices, and $\mathcal{H}^{C_n}$ is the n-dimensional space associated with the quantum coin. The propagator of the SKW quantum walk can therefore be written as

$$S=\sum_{d,\vec{x}}|d,\vec{x}\oplus\vec{e}_d\rangle\langle d,\vec{x}|, \tag{1}$$

where $\vec{e}_d=2^d$ corresponds to the edges originating from the given vertex. If the target vertex marked by the oracle $\mathcal{O}$ is denoted by $\vec{x}_{\mathrm{tg}}$, the perturbed coin operator can be written as

$$C'=C_0\otimes\mathbb{1}+(C_1-C_0)\otimes|\vec{x}_{\mathrm{tg}}\rangle\langle\vec{x}_{\mathrm{tg}}|. \tag{2}$$

For the SKW quantum walk, C_0 is usually chosen to be the n-dimensional Grover operator (also known as the Grover diffusion operator) and C_1 is chosen to be $-\mathbb{1}$. The results in this section, however, hold for any pair of inequivalent permutation invariant unitary coins. As argued in [1], due to the symmetry of the hypercube graph the vertices can always be relabeled in such a way that the marked vertex becomes $\vec{x}_{\mathrm{tg}}=0$. Since with this choice the permutation invariance of the Grover walk on the hypercube is conserved, the initial state

$$|\psi_0\rangle=\frac{1}{\sqrt{n2^n}}\sum_{d=1}^{n}\sum_{\vec{x}}|d,x\rangle \tag{3}$$

allows the reduction to a walk on a line. The basis states for this collapsed quantum walk are defined as

$$|R,x\rangle=\sqrt{\frac{1}{(n-x)\binom{n}{x}}}\sum_{|\vec{x}|=x}\sum_{x_d=0}|d,\vec{x}\rangle, \tag{4}$$

$$|L,x\rangle=\sqrt{\frac{1}{x\binom{n}{x}}}\sum_{|\vec{x}|=x}\sum_{x_d=1}|d,\vec{x}\rangle, \tag{5}$$

and the propagator becomes

$$S=\sum_{x=0}^{n-1}|R,x\rangle\langle L,x+1|+|L,x+1\rangle\langle R,x|. \tag{6}$$

The coin operator of the walk on the line acquires a strong position dependence. For example, when C_0 is the Grover coin, in the collapsed basis it becomes

$$C_0=\sum_{x=0}^{n}\begin{pmatrix}\cos\omega_x & \sin\omega_x\\ \sin\omega_x & -\cos\omega_x\end{pmatrix}\otimes|x\rangle\langle x|, \tag{7}$$

where $\cos\omega_x=1-2x/n$ and $\sin\omega_x=(2/n)\sqrt{x(n-x)}$, and the matrix is understood in the $\{|R\rangle,|L\rangle\}$ basis. The perturbed coin with $C_1=-\mathbb{1}$ can be written as

$$C'=C_0-2|R,0\rangle\langle R,0|. \tag{8}$$

It has been shown in [1] that after an optimal number of iterations the probability p_0 of obtaining the target state $|0\rangle$ in a measurement is close to $1/2$, and that the optimal number of iterations is well estimated by the nearest integer to

$$t_f=(\pi/2)\sqrt{2^{n-1}}. \tag{9}$$

This means that the final state is composed mainly of the target state, and contains smaller contributions from its next and more distant neighbors [1]. However, this statement can be refined by partitioning the SKW quantum walk into two independent quantum walks. Let $\mathcal{H}_e$ denote the subspace spanned by states $|d,\vec{x}\rangle$ such that $|\vec{x}|$ is even, and $\mathcal{H}_o$ denote the subspace spanned by the states with $|\vec{x}|$ being odd. The terms even and odd refer to a labeling where the target vertex is denoted by $\vec{x}_{\mathrm{tg}}=0$; therefore, in general, these subspaces must be defined according to the parity of $\vec{x}\oplus\vec{x}_{\mathrm{tg}}$. The two quantum walks are started in the Hilbert spaces $\mathcal{H}_e$ and $\mathcal{H}_o$, and evolve independently. In the following we shall term $\mathcal{H}_e$ the *even* subspace and $\mathcal{H}_o$ the *odd* subspace of $\mathcal{H}$. It follows from the property of the parity function that this partitioning of $\mathcal{H}$ is the same for all values of $\vec{x}_{\mathrm{tg}}$; however, the role of the two subspaces depends on the parity of $\vec{x}_{\mathrm{tg}}$. We can define the orthogonal projectors P_e and P_o that project to $\mathcal{H}_e$ and $\mathcal{H}_o$, respectively. Clearly, in the collapsed basis, the *even* subspace is spanned by the states (4) and (5) with x being even, and the *odd* subspace is spanned by those with x being odd. Since $[P_{e/o},C_0]=0$ and $[P_{e/o},C']=0$ it follows from the definition of S that

$$P_oU'=U'P_e, \tag{10a}$$

$$P_eU'=U'P_o. \tag{10b}$$

Let us introduce the (normalized) states

$$|\psi_0^{(e)}\rangle=\sqrt{2}P_e|\psi_0\rangle, \tag{11}$$

$$|\psi_0^{(o)}\rangle=\sqrt{2}P_o|\psi_0\rangle, \tag{12}$$

and express the initial state as $|\psi_0\rangle=\frac{1}{\sqrt{2}}(|\psi_0^{(e)}\rangle+|\psi_0^{(o)}\rangle)$. It can easily be seen that the action of U' on $|\psi_0^{(o)}\rangle$ simplifies to

$$U'|\psi_0^{(o)}\rangle=U|\psi_0^{(o)}\rangle=|\psi_0^{(e)}\rangle. \tag{13}$$

By successive applications of Eqs. (10) and (13) it can be shown that U' has the property

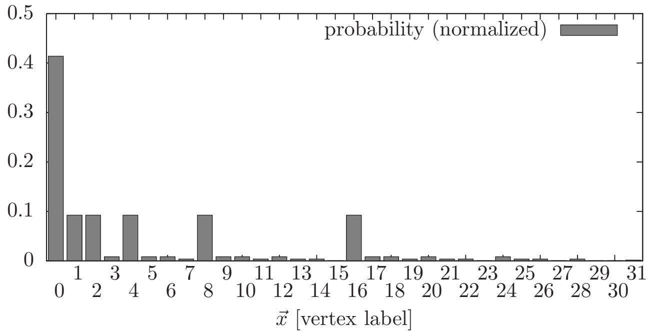

FIG. 1. The plot shows the numerically calculated probability distribution for the position of the walker after the optimal number of iterations of the SKW quantum walk in $n=5$ dimensions. In accordance with the analytic results, the probability distribution has its maximum for the marked vertex $\vec{x}_{\text{tg}}=0$ reaching a value close to $1/2$. Moreover, we observe that the nearest neighbors are also presented with high probability, and the sum of these probabilities is comparable to that of the marked vertex.

$$P_e(U')^{2r}|\psi_0\rangle = P_e(U')^{2r+1}|\psi_0\rangle \quad (r=0,1,2,\ldots), \tag{14a}$$

$$P_o(U')^{2r}|\psi_0\rangle = P_o(U')^{2r-1}|\psi_0\rangle \quad (r=1,2,3,\ldots). \tag{14b}$$

Let us express the state of the walker after t steps as

$$(U')^t|\psi_0\rangle = \sum_{x=0}^{n-1} \alpha^t_{R,x}|R,x\rangle + \sum_{x=1}^{n} \alpha^t_{L,x}|L,x\rangle, \tag{15}$$

and define $P^t_x=|\alpha^t_{L,x}|^2+|\alpha^t_{R,x}|^2$, with setting $\alpha^t_{R,n}=\alpha^t_{L,0}=0$ for convenience. The interpretation of P^t_x is clear from the definitions: P^t_0 is the probability of having the walker at the target vertex after t iterations, and P^t_1 is the total probability of finding the walker at any of the nearest neighbors of the target node. Using the above bipartition of the quantum walk it can be shown that the inequalities

$$P^{t+1}_0 \leqslant P^t_1, \tag{16a}$$

$$P^{t-1}_0 \leqslant P^t_1 \tag{16b}$$

hold for all $t>0$, from which it follows that the probabilities

$$p_0 = \sum_{d=0}^{n} |\langle d,0|\psi_f\rangle|^2, \tag{17}$$

$$p_1 = \sum_{d,|\vec{x}|=1}^{n} |\langle d,\vec{x}|\psi_f\rangle|^2 \tag{18}$$

satisfy the inequality

$$p_1 \geqslant p_0. \tag{19}$$

For the details of the calculations see Appendix A. The property (19) can also be verified on the numerical results presented in Fig. 1. Therefore, since we have $p_0=1/2-O(1/n)$, the total probability of measuring the target node or any of its direct neighbors is

$$p_c = p_0 + p_1 \geqslant 1 - O(1/n). \tag{20}$$

Since p_c is upper bounded by 1, for large n the total probability must be approaching $p_c=1$. Naturally, the question arises: can Eq. (20) be turned to our advantage? In the following we shall address this question, and answer positively.

First, let us analyze the most straightforward way of taking advantage of Eq. (20). According to the SKW protocol, the validity of the measurement outcome $\vec{x}_m$ after t_f iterations is verified using the oracle. If the verification is positive the target node is found, otherwise the result is discarded and the SKW quantum walk is repeated. However, this is unnecessary since from Eq. (20) we know that in the case of a negative answer from the oracle, the probability that $\vec{x}_m$ is a direct neighbor of $\vec{x}_{\text{tg}}$ is greater than $1-O(1/n)$. Therefore, it is sufficient to query the oracle with values from the set $\{\vec{x}_m \oplus \vec{e}_d | d=0,\ldots,n-1\}$, from which the marked element can be extracted using the simplest classical protocol by an average of $(\log_2 N)/2$ additional oracle queries.

In a scenario where the verification costs are dominating over all other costs it is crucial to perform the minimum number of necessary verification queries. One possibility could be to use amplitude amplification or another quantum based search; however, both these approaches mean a departure from the original hypercube quantum random walk.

In the following, we propose an alternative approach to reduce the number of verification queries if the coin states can also be determined. Let us set $t_{f,o}=2\lfloor t_f/2\rfloor+1$, and denote the outcome of the measurement on the coin state by d_m. Using the notations of Eq. (15), we can rewrite Eq. (14a) for the case $j=0$, and obtain $|\alpha^{t_o-1}_{R,0}|^2=|\alpha^{t_o}_{R,0}|^2=1/2-O(1/n)$. From the unitarity of the coins and the definition (6) of S it follows that we also have

$$|\alpha^{t_o}_{L,1}|^2 = |\alpha^{t_o}_{R,0}|^2 = \frac{1}{2} - O(1/n). \tag{21}$$

Note that this ensures also that we have $\alpha^{t_o}_{R,1}=O(1/n)$, which is negligible for large n. Therefore, we conclude that the final state is composed mainly of the states $|R,0\rangle$ and $|L,1\rangle = 1/\sqrt{n}\Sigma_{d=0}^{n-1}|d,\vec{e}_d\rangle$. Thus, if the measurement gives $\vec{x}_m \neq \vec{x}_{\text{tg}}$ then the target vertex can be found with $1-O(1/n)$ probability by taking $\vec{x}_{\text{tg}}=\vec{x}_m \oplus \vec{e}_{d_m}$.

In other words, if a complete measurement can be made on the coin state, the marked element can be determined with $1-O(1/n)$ probability after a single execution of the SKW algorithm and one verification query to the oracle.

III. MODIFICATION TO ATTAIN OPTIMAL QUERY COMPLEXITY

In the present section, based on the SKW algorithm we develop a search algorithm which finds the marked vertex of a hypercube using the optimal number of oracle queries. In contrast to the modifications of Sec. II, which essentially affect only the classical processing part, the improvement proposed in the present section requires a modification of the quantum walk itself.

The improvement is based on the bipartite nature of the SKW quantum walk, which implies the invariance of the *even* and *odd* subspaces under two iterations of U',

$$[P_o,U'^{2r}]=[P_e,U'^{2r}]=0 \quad (r=0,1,\ldots), \tag{22}$$

which follows from Eqs. (10). First, consider the projection of the state of the walker after $2r$ iterations onto the even subspace. In the spirit of Eq. (22) we can see that the projection of the final state corresponds to a similar projection of the initial state, which we can write as

$$U'^{2r}|\psi_0^{(e)}\rangle=\sqrt{2}P_eU'^{2r}|\psi_0\rangle. \tag{23}$$

Introducing $t_{f,e}=2\lfloor t_f/2\rfloor$ we conclude that for the probability $P_0^{(e)t_{f,e}}$ to find the marked node after $t_{f,e}$ iterations starting from the even initial state $|\psi_0^{(e)}\rangle$ the relation

$$P_0^{(e)t_{f,e}}=2P_0^{t_{f,e}}=1-O(1/n) \tag{24}$$

holds. This is an encouraging result, since it suggests that the marked element can be directly found with high probability after a single execution of the SKW algorithm without any verification queries. However, the choice $\vec{x}_{\text{tg}}=0$ is actually the result of the mapping $\vec{x}\rightarrow\vec{x}\oplus\vec{x}_{\text{tg}}$, thus we do not know in general which is the even subspace and which is the odd subspace.

The information about the parity of the marked vertex is clearly contained in the oracle. An efficient way of extracting this information is to repeat the quantum walk twice, once starting from the initial state $|\psi_0^{(e)}\rangle$ and once starting from $|\psi_0^{(o)}\rangle$. Note that it is not necessary to know which one is which, since $|\psi_0^{(e)}\rangle$ will yield $|\vec{x}_{\text{tg}}\rangle$ with nearly unit probability. Therefore, the target vertex can be identified by testing the two measurement outcomes $x_m^{(e)}$ and $x_m^{(o)}$ on the oracle.

Instead of repeating the algorithm twice, it is possible to construct another SKW quantum walk in which it is guaranteed that the marked element corresponds to a vertex with even parity. The principle of this modification is the mapping of all the vertices of the n-dimensional hypercube to the even parity vertices of an $n'=n+1$-dimensional hypercube. Since the number of even and odd vertices is equal for a hypercube in every dimension, the mapping between the original vertices and the even parity vertices of the larger hypercube can be made one to one.

In the following, we assume that the oracle is given as an operator acting on the Hilbert space $\mathcal{H}^{V_n}$ associated to the n-dimensional hypercube, and we shall construct a SKW quantum walk in $n'=n+1$ dimensions using the extended oracle acting on the Hilbert space $\mathcal{H}^{V_{n'}}$. The vertices $\vec{x}$ of the original hypercube are mapped to the even parity sites of the extended hypercube by the map

$$\vec{x}=m(\vec{x})=2\vec{x}+p(\vec{x}), \tag{25}$$

where $p(\vec{x})$ denotes the parity of $\vec{x}$. This mapping can be viewed as appending one bit to the bit string representation of the original vertex, the value of the bit being 1 for odd parity vertices, and 0 for even parity vertices. The reverse mapping simply drops the appended bit for even parity input, while the odd parity vertices of the extended hypercube do not correspond to any vertices of the original graph.

In this way the marked vertex is known to be mapped to an even parity vertex on the extended hypercube. The modification of the oracle $\mathcal{O}$ to return a positive result only for the new marked vertex is straightforward. Let the operators of the n'-dimensional extended SKW quantum walk be distinguished from the original n-dimensional one by adding a $(+)$ superscript. Therefore, the coin operators acting on $\mathcal{H}^{C_{n'}}$ are denoted by $C_0^{(+)}$ and $C_1^{(+)}$, and the propagator operator on $\mathcal{H}^{C_{n'}}\otimes\mathcal{H}^{V_{n'}}$ by $S^{(+)}$. Similarly, the perturbed coin operator is denoted by $C'^{(+)}$. With this mapping, the procedure described above can be applied very efficiently since the "good" initial state $|\psi_0^{(e)}\rangle$ is prescribed by the construction. Consequently, a single execution of the n'-dimensional SKW quantum walk is sufficient to find the marked vertex with a probability close to unity. Note that the extension to $n'=n+1$ dimensions changes the optimal number of iterations, which amounts to an increase of the query complexity by a factor of $\sqrt{2}$.

The query complexity can be reduced by noting that at every second iteration, the coin operator $C^{(+)}$ could effectively be replaced by the unperturbed coin operator $C_0^{(+)}$, thus the number of oracle queries can be reduced by $1/2$. Moreover, as shown in Appendix B, by forcing the coin operator to be $C_0^{(+)}$ for every second iteration, the equality

$$(U^{(+)}U''^{(+)})^r|\psi_0\rangle=\frac{1}{\sqrt{2}}(X+\mathbb{1})(U^{(+)}U''^{(+)})^r|\psi_0^{(e)}\rangle \tag{26}$$

holds, where X denotes the quantum NOT gate, σ_X, acting on the last qubit. Thus, an initial state of uniform superposition (3) can be used, yielding the image $(\vec{x}'_{\text{tg}})$ and the anti-image $(\vec{x}'_{\text{tg}}\oplus 1)$ of the target vertex with a total probability close to one. Therefore, by performing a measurement that ignores the last qubit we obtain the marked vertex $\vec{x}_{\text{tg}}$ with probability $1-O(1/n')$.

Using the formula (9) to calculate the query complexity, we find that the modified algorithm is completed using $t'_f=(\pi/4)\sqrt{N}$ oracle queries, which is identical to what is needed by the Grover search algorithm, and known to be the best achievable on a quantum computer for a success probability of one [15].

The storage complexity of the improved algorithm can be reduced by noting that the auxiliary qubit can be eliminated using the identities

$$[X,U^{(+)}]=[X,U''^{(+)}]=0, \tag{27}$$

$$X|\psi_0\rangle=|\psi_0\rangle. \tag{28}$$

Clearly, the reduction affects only the dimensionality of the position space, and leaves the coin space $n'=n+1$ dimensional. With some algebra, we obtain the reduced propagator from $S^{(+)}$ as

$$\tilde{S}=\sum_{\vec{x}}\left(\sum_{d=0}^{n-1}|d,\vec{x}\oplus\vec{e}_d\rangle\langle d,\vec{x}|+|n,\vec{x}\rangle\langle n,\vec{x}|\right). \tag{29}$$

Thus, the coin states $|d\rangle$ with $d<n$ become the coin states of a quantum random walk on the original n-dimensional hypercube, while the state $|n\rangle$ corresponds to a coin state in-

structing the walker to remain at the same vertex at the next iteration.

The propagator can equivalently be understood as describing a quantum random walk on a regular graph consisting of an n-dimensional hypercube having a self-loop edge attached to each of its vertices. The final version of the quantum walk for optimal search can therefore be expressed by the alternating sequence of the unitary operators

$$\tilde{U}'' = \tilde{S}C''^{(+)}, \tag{30}$$

$$\tilde{U} = \tilde{S}C_0^{(+)}, \tag{31}$$

acting on an $N=2^n$-dimensional vertex space, and an $n+1$-dimensional coin space.

IV. APPLICATIONS TO FINDING MULTIPLE MARKED VERTICES

In the present section we consider the optimization problem when the number of marked vertices is more than one. Although the SKW algorithm is guaranteed to work only when the oracle marks a single vertex, numerical calculations suggest that it can also be used to find multiple marked vertices as long as the number of marked vertices is small compared to the size of the search space.

To answer the question whether the SKW algorithm can be used to find multiple marked vertices is beyond the scope of the present paper. Instead, here we focus on the question of applicability of the improvement described in Sec. III. In the following, we shall show that the modified algorithm can be applied directly to the search for multiple marked vertices when the SKW algorithm on the extended hypercube yields sufficient results. To formalize the task of finding multiple marked vertices, let us denote the number of elements marked by the oracle by m, and their labels by $\vec{x}_{\text{tg}}^{(j)}$, such that $j=1,\ldots,m$. The coin operator of the SKW quantum walk can therefore be written as

$$C'_m = C_0 \otimes \mathbb{1} + (C_1 - C_0) \otimes \sum_{j=1}^{m} |\vec{x}_{\text{tg}}^{(j)}\rangle\langle\vec{x}_{\text{tg}}^{(j)}|, \tag{32}$$

and the unitary evolution operator as $U'_m=SC'_m$. This unitary operator is then iterated a given number of times to obtain a final state that is composed mainly of the states corresponding to the marked vertices.

For simplicity, here we consider the improved form of the SKW algorithm using the extended $n'=n+1$-dimensional hypercube with the even parity initial state. This is sufficient, since it is equivalent to all the subsequently improved forms. Clearly, by defining the n'-dimensional extension $C_m'^{(+)}$ of C'_m we arrive at the unitary evolution operator $U_m'^{(+)}$, which also obeys

$$|\psi^{(e)}(2r)\rangle = U_m'^{(+)2r}|\psi_0^{(e)}\rangle = (U_m^{(+)}U_m''^{(+)})^r|\psi_0^{(e)}\rangle, \tag{33}$$

since all the marked vertices are mapped to the even subspace. For the same reason, we have for every d the relation

$$|\langle d,\vec{x}_{\text{tg}}^{(j)}{}'|U_m'^{(+)2r}|\psi_0^{(e)}\rangle|^2 = 2|\langle d,\vec{x}_{\text{tg}}^{(j)}{}'|U_m'^{(+)2r}|\psi_0\rangle|^2, \tag{34}$$

according to the definition (11). Therefore, if the total probability of finding any of the marked vertices in the final state of the extended SKW algorithm is close to $1/2$, the modified algorithm yields them with probability close to unity.

V. CONCLUSIONS

We have proposed two alternative approaches for improving the SKW quantum random-walk search algorithm. Both improvements take advantage of the fact that the probability of success of a single run can be increased to almost 1. In the first part of the paper we have shown that the next neighbors of the target can be obtained with high probability, and that this can be exploited to reduce the number of repetitions or independent oracle queries to one or two. We note that for certain implementations, a lower number of repetitions may have a serious impact on efficiency. In the second part of the paper we have developed a two-coin quantum random-walk search algorithm on a hypercube with self-loop edges. We have pointed out that the speedup over the original SKW algorithm in terms of oracle queries is $\sqrt{2}$. This makes the algorithm equivalent to the Grover search in terms of query complexity; therefore, we present an optimal solution to the search problem if the success probability of 1 is required [15].

We have also considered the optimization problem of finding multiple marked vertices. We have shown that if the SKW quantum walk mapped to an $n+1$-dimensional hypercube yields the marked vertices with probability close to $1/2$, the algorithm in Sec. III can be applied unmodified, resulting in the same improvement as for the case of a single marked vertex.

ACKNOWLEDGMENTS

This work was supported by the Czech and Hungarian Ministries of Education (CZ-10/2007), by Grants No. MSMT LC 06002 and No. MSM 6840770039, and by the Hungarian Scientific Research Fund (Grants No. T049234 and No. T068736). The authors would like to thank Dr. E. Andersson for valuable comments and hospitality at Herriot-Watt University. Financial support by the Royal Society under Grant No. 2006/R2 IJP is gratefully acknowledged.

APPENDIX A: PROOF OF EQ. (19)

Using the notation of Eq. (15), let us consider an arbitrary $\alpha_{R,0}^{t-1}$ ($\alpha_{L,0}^{t-1}$ is set to 0 by definition). In one iteration, $\alpha_{R,0}^{t-1}$ is first transformed to some $\beta_{R,0}^{t-1}$ and $\beta_{L,0}^{t-1}$ by the coin operator C'. Upon inspecting the definition of $|R,0\rangle$ we find that due to the unitarity of the coin C' we have $|\beta_{R,0}^{t-1}|^2=|\alpha_{R,0}^{t-1}|^2=P_0^{t-1}$ and $\beta_{L,0}^{t-1}=0$. Considering the action of S, we obtain $\alpha_{L,1}^{t}=\beta_{R,0}^{t-1}$. Therefore, we can write $P_1^t=|\alpha_{R,1}^t|^2+|\alpha_{L,1}^t|^2\geqslant|\alpha_{L,1}^t|^2=|\alpha_{R,0}^{t-1}|^2=P_0^{t-1}$, which proves Eq. (16a). The second inequality can be proven along similar lines. Due to the unitarity of the coins we always have $|\beta_{R,1}^t|^2+|\beta_{L,1}^t|^2=|\alpha_{R,1}^t|^2+|\alpha_{L,1}^t|^2$, and according to the definition of S, $\alpha_{R,0}^{t+1}=\beta_{L,1}^t$ also holds.

Therefore, we can write $P_0^{t+1}=|\alpha_{R,0}^t|^2=|\beta_{L,1}^t|^2\leqslant|\beta_{R,1}^t|^2+|\beta_{L,1}^t|^2=P_1^t$, which provides Eq. (16b).

From Eqs. (14) it follows that $P_x^{2r}=P_x^{2r+1}$ if x is even, and that $P_x^{2r}=P_x^{2r-1}$ if x is odd. Combining these equalities with Eqs. (16) we obtain

$$P_1^t \geqslant P_0^t \tag{A1}$$

for every positive integer t. Equation (19) is a special case of Eq. (A1).

APPENDIX B: PROOF OF EQ. (26)

First note that $C'^{(+)}P_o=(C_0^{(+)}\otimes\mathbb{1})P_o$ holds; therefore we have

$$U'^{(+)2r}|\psi_0^{(e)}\rangle=(U^{(+)}U'^{(+)})^r|\psi_0^{(e)}\rangle, \tag{B1}$$

since Eqs. (10) hold for hypercubes in all dimensions. Moreover, we can write

$$C'^{(+)}P_e=C''^{(+)}P_e, \tag{B2}$$

by introducing

$$C''^{(+)}=[C_0^{(+)}\otimes\mathbb{1}+(C_1^{(+)}-C_0^{(+)})\otimes|\vec{x}_{\text{tg}}\rangle\langle\vec{x}_{\text{tg}}|]\otimes\mathbb{1}_2, \tag{B3}$$

where $\mathbb{1}_2$ is the identity acting on the qubit added by the extension. We can use the coin (B3) to define the unitary evolution operator $U''^{(+)}=S^{(+)}C''^{(+)}$. By considering the expression that gives the final state of the walker after $2r$ steps we find that it can be simplified to

$$U'^{(+)2r}|\psi_0^{(e)}\rangle=(U^{(+)}U''^{(+)})^r|\psi_0^{(e)}\rangle, \tag{B4}$$

by using Eqs. (B1) and (B2). The advantage of this formulation is that the oracle $\mathcal{O}$ is used on the subspace $\mathcal{H}^{V_n}$ unchanged, as can be seen in Eq. (B3). As a consequence, the coin operator $C''^{(+)}$ acts on the total Hilbert space $\mathcal{H}^{C_{n'}}\otimes\mathcal{H}^{V_{n'}}$ as if two nodes were marked which differ only in their last bits. Figure 2 illustrates the pair of marked vertices. Intuitively, this is compensated in Eq. (B4) by alternating $C''^{(+)}$ with a coin that marks no vertices at all.

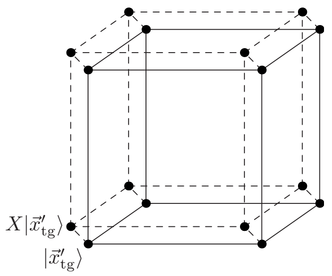
$X|\vec{x}'_{\text{tg}}\rangle$
$|\vec{x}'_{\text{tg}}\rangle$

FIG. 2. Extension of the search in $n=3$ dimensions to a hypercube in $n'=4$ dimensions without distinguishing even and odd parity vertices. The operator X denotes the application of the Pauli σ_X operation to the last qubit. The effect of operator X is switching between the image and the anti-image of a vertex.

Next, we show that we can use the uniform superposition initial state $|\psi_0\rangle$ as an initial state to the quantum walk if the iterations are carried out according to the right-hand side of Eq. (B4). Let X denote the quantum NOT gate, σ_X, acting on the last qubit. Clearly, we have $X|\psi_0^{(e)}\rangle=|\psi_0^{(o)}\rangle$, and $[X,U^{(+)}]=[X,U''^{(+)}]=0$. Thus we can rewrite the desired initial state (3) as $|\psi_0\rangle=(X+\mathbb{1})/\sqrt{2}|\psi_0^{(e)}\rangle$ and see that

$$(U^{(+)}U''^{(+)})^r|\psi_0\rangle=\frac{1}{\sqrt{2}}(X+\mathbb{1})(U^{(+)}U''^{(+)})^r|\psi_0^{(e)}\rangle \tag{B5}$$

holds. In the right-hand side we can discover Eq. (B4), which yields the state $|\vec{x}'_{\text{tg}}\rangle$ with $1-O(1/n')$ probability, where $\vec{x}'_{\text{tg}}$ is the image of $\vec{x}_{\text{tg}}$ by the map (25). This probability is distributed uniformly between the image $\vec{x}'_{\text{tg}}$ and the anti-image $\vec{x}'_{\text{tg}}\oplus 1$ due to the multiplication by $(X+\mathbb{1})/\sqrt{2}$.

[1] N. Shenvi, J. Kempe, and K. Birgitta Whaley, Phys. Rev. A **67**, 052307 (2003).

[2] L. Grover, *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)* (ACM, New York, 1996), p. 212.

[3] A. Ambainis, SIGACT News **35(2)**, 22 (2004).

[4] M. Santha, *Proceedings of the Fifth Theory and Applications of Models of Computation (TAMC08)* (Xian, LNCS 4978, 2008), pp. 31–46.

[5] G. Brassard and P. Høyer, *Proceedings of the Fifth Israeli Symposium on Theory of Computing and Systems* (IEEE Computer Society Press, Washington, D.C., 1997), pp. 12–23.

[6] L. K. Grover, Phys. Rev. Lett. **80**, 4329 (1998).

[7] A. Ambainis, J. Kempe, and A. Rivosh, *Proceedings of the 16th ACM-SIAM SODA* (Society for Industrial and Applied Mathematics, Philadelphia, 2005), pp. 1099–1108.

[8] A. Tulsi, Phys. Rev. A **78**, 012310 (2008).

[9] D. Reitzner, M. Hillery, E. Feldman, and V. Bužek, e-print arXiv:0805.1237.

[10] M. Hillery, J. Bergou, and E. Feldman, Phys. Rev. A **68**, 032314 (2003).

[11] J. Košík and V. Bužek, Phys. Rev. A **71**, 012306 (2005).

[12] A. Gábris, T. Kiss, and I. Jex, Phys. Rev. A **76**, 062315 (2007).

[13] C. M. Chandrashekar, R. Srikanth, and R. Laflamme, Phys. Rev. A **77**, 032326 (2008).

[14] F. Magniez, A. Nayak, J. Roland, and M. Santha, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing* (ACM New York, NY, USA, 2007), STOC ‘07, pp. 575–584; e-print arXiv:quant-ph/0608026.

[15] C. Zalka, Phys. Rev. A **60**, 2746 (1999).

[16] C. Moore and A. Russell, *Proceedings of RANDOM 06*,Lecture Notes in Computer Science (Springer, Berlin, 2002), Vol. 2483, pp. 164–178.