

A quantum decoding algorithm of the simplex code

Alexander Barg* Shiyu Zhou†

Abstract

The paper is devoted to the quantum implementation of the decoding of the (classical) simplex code of length n . The implementation attempts at trading off time complexity with circuit complexity of decoding. We suggest a quantum decoding algorithm that operates on a circuit of size $O(\log^2 n)$ and has time complexity $O(\log^2 n)$. It also requires an additional circuit of size $O(n)$ needed to gain bitwise access to the input vector. The best known classical parallel algorithm for this problem requires circuit size $O(n \log n)$ and time $O(\log n)$.

Keywords: linear codes, decoding, quantum computation, Reed-Muller codes.

1 Introduction

Recently it has been shown [10], [7] that quantum algorithms can solve search problems significantly faster than classical computations. These discoveries generated a large amount of research (see [11]). However, it is also noted that scaling up quantum circuit seems to be extraordinarily expensive and may present fundamental difficulties because of the decoherence problem and the precision problem in quantum computation (see e.g. [10]). Therefore, in some cases, it could be helpful to have quantum algorithms that can be implemented using circuits of small sizes while not sacrificing much in running time. With this motivation, we study in this paper the issue of trading off time complexity with circuit complexity in designing quantum algorithms for the decoding problem of binary linear codes.

The *size* of a quantum circuit is defined as the number of quantum gates (on constant number of input qubits) used in the circuit. The *circuit complexity* of a quantum algorithm is the size of the smallest quantum circuit that implements the algorithm.

Designing quantum algorithms for search problems is usually not straightforward. A question raised in [2] was whether it is possible to design quantum decoding algorithms for good (classical) codes. Here we show that the circuitry related to the decoding of the binary simplex code can be reduced significantly using quantum parallelism.

The problem that we study has a rather low classical complexity. It turns out that while the quantum approach provides a considerable reduction in the size of the computation part of the circuit, “reading the input” actually takes the most part of the circuit. Therefore, throughout the paper we consider quantum circuits consisting of two distinct components: the *input sub-circuit* and the *computation sub-circuit*. The size of a quantum circuit is the sum of the sizes of its two components. The input sub-circuit is the part of the quantum circuit that takes n input qubits and additional $\log n$ qubits as index i , and outputs the i -th qubit of the input corresponding to the index. So to speak, the

*Bell Laboratories, Lucent Technologies, 600 Mountain Avenue 2C-375, Murray Hill, NJ 07974. E-mail abarg@research.bell-labs.com.

†Previous address: Bell Laboratories, Lucent Technologies, 600 Mountain Avenue, Murray Hill, NJ 07974, USA. Current address: Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19104-6389. E-mail shiyu@central.cis.upenn.edu.

input sub-circuit provides the later computation with bit-wise access to the input. It is well known that there exists a classical circuit of size $O(n)$ that can implement this in time $O(\log n)$. Following a by now standard fact (see, e.g., §3 in [10]), this classical computation can be made reversible for only a constant factor increase in time and circuit size. Therefore, the input sub-circuit is of size $O(n)$ and takes time $O(\log n)$. The computation sub-circuit is the part of the quantum circuit that performs the actual computation given the bit-wise access to the input qubits provided by the input sub-circuit.

Let $C \subseteq \mathbf{Z}_2^n$ be a binary linear code. The decoding mapping $\delta : \mathbf{Z}_2^n \rightarrow C$ is defined as follows:

$$\delta(\mathbf{x}) = \mathbf{c} \iff d(\mathbf{x}, \mathbf{c}) = d(\mathbf{x}, C),$$

where d is the Hamming metric and $d(\mathbf{x}, C)$ is the shortest distance from \mathbf{x} to the code. If this condition is satisfied for several code vectors \mathbf{c} , the value of δ is chosen arbitrarily from them. The algorithmic decoding problem is to implement the function δ . It is known [4] that for arbitrary (linear) code this problem is NP-hard. On the other hand, as an immediate consequence of Grover's quantum search algorithm [7], the decoding of any (not necessarily linear) code of length n can be solved on a quantum computer of circuit size $O(n|C|^{1/2})$ in time $O(n|C|^{1/2})$, which is essentially optimal following a result in [3].

In this paper we take C to be the *binary simplex code*. The classical parallel algorithm decoding the binary simplex code has time complexity $O(\log n)$ and circuit complexity $O(n \log n)$ [9]. It is not known whether the $O(n \log n)$ circuit size can be reduced even at the cost of increasing the running time of the algorithm. We show, however, that there exists a quantum probabilistic algorithm that can be implemented by a circuit of size $O(n)$, where the size of the computation sub-circuit is only $O(\log^2 n)$, and runs in time $O(\log^2 n)$. However, we have to somewhat restrict the mapping. Namely, let $\mathbf{c} \in C$, let $\theta \in (0, 1)$, and define

$$D_\theta(\mathbf{c}) := \{\mathbf{x} \in \mathbf{Z}_2^n : \forall \mathbf{c}' \in C, \mathbf{c}' \neq \mathbf{c} \ d(\mathbf{c}, \mathbf{x}) - d(\mathbf{c}', \mathbf{x}) \leq -\theta n\}, \quad (1)$$

$$E_\theta(\mathbf{c}) := \{\mathbf{x} \in \mathbf{Z}_2^n : \forall \mathbf{c}' \in C, \mathbf{c}' \neq \mathbf{c} \ d(\mathbf{c}, \mathbf{x}) + d(\mathbf{c}', \mathbf{x}) \leq (1 - \theta)n\}. \quad (2)$$

Let $D_\theta(C) = \cup_{\mathbf{c} \in C} D_\theta(\mathbf{c})$ and $E_\theta(C) = \cup_{\mathbf{c} \in C} E_\theta(\mathbf{c})$.

Let δ_θ be a restriction of δ to the intersection $D_\theta(C) \cap E_\theta(C)$. The algorithm implementing δ_θ that we construct, has time complexity $O(\log^2 n)$ and the probability of decoding error n^{-c} , where the exponent depends on θ .

2 Preliminaries

Let $n = 2^m$, where m is a positive integer. We shall use both numbers $v \in \{0, 1, \dots, n-1\}$ and their binary expansions. To tell the one from the other, we denote by \vec{v} the vector of coefficients of the binary expansion of v . To any vector $\mathbf{x} \in \mathbf{Z}_2^n$ we associate a function

$$\begin{aligned} \mathbf{f}_x : \mathbf{Z}_2^m &\rightarrow \mathbf{Z}_2 \\ \vec{v} &\mapsto \mathbf{x}_v \end{aligned}$$

such that its value on \vec{v} is equal to the v th bit of \mathbf{x} . This defines a bijection between the set of Boolean functions of m arguments and \mathbf{Z}_2^n . By abuse of notation, below we use bold symbols to denote both the function and the vector of its values. For instance, $\mathbf{c}(\vec{v})$ means the v th bit of a vector $\mathbf{c} \in \mathbf{Z}_2^n$.

The functional representation of \mathbf{Z}_2^n is convenient in the study of Reed-Muller codes and their subcodes since one can define the code by specifying a subset of functions. In particular, the dual space $(\mathbf{Z}_2^m)^*$, i.e., the space of all linear functions $\{\sum_{i=1}^m u_i v_i\}$ on \mathbf{Z}_2^m , defines an $[n = 2^m, m, 2^{m-1}]$

linear code C (the simplex code extended by an all-zero column). In other words, a code vector $\mathbf{c} \in C$ has the form $(c_0, c_1, \dots, c_{n-1})$, where $c_v = \langle \vec{u}, \vec{v} \rangle$ and $\langle \cdot, \cdot \rangle$ is the standard dot product on \mathbf{Z}_2^m . The encoding procedure is given by the isomorphism $(\mathbf{Z}_2^m)^* \cong C \subset \mathbf{Z}_2^n$. The code vector corresponding to a message $\vec{u} \in (\mathbf{Z}_2^m)^*$ (the image of the covector \vec{u} under this isomorphism) is denoted by $\mathbf{c}_{\vec{u}}$. By the Plotkin bound, the covering radius of C is $\rho(C) = 2^{m-1}$.

An obvious consequence of the definition of C is as follows: for every 2 code vectors $\mathbf{c}_1 \neq \mathbf{c}_2$ we have

$$d(\mathbf{c}_1, \mathbf{c}_2) = 2^{m-1} = \frac{n}{2}. \quad (3)$$

Given any vector $\mathbf{r} \in \mathbf{Z}_2^n$, define the function $F_{\mathbf{r}} : \mathbf{Z}_2^m \rightarrow \{-1, 1\}$ by

$$F_{\mathbf{r}}(\vec{v}) = (-1)^{\mathbf{r}(\vec{v})}.$$

Definition 2.1 *The function*

$$\begin{aligned} \widehat{F}_{\mathbf{r}} : \mathbf{Z}_2^m &\rightarrow \mathbf{C} \\ \vec{u} &\mapsto \sum_{\vec{v} \in \mathbf{Z}_2^m} F_{\mathbf{r}}(\vec{v}) (-1)^{\langle \vec{u}, \vec{v} \rangle} \end{aligned}$$

is called the Walsh-Hadamard transform of the function $F_{\mathbf{r}}$ (equivalently, of the vector $\vec{u} \in \mathbf{Z}_2^m$).

Note that one can think of $\widehat{F}_{\mathbf{r}}$ as of the mapping defined on $(\mathbf{Z}_2^m)^*$. Note also that $\sum_{\vec{u}} \widehat{F}_{\mathbf{r}}^2(\vec{u}) = 2^{2m}$ by the Parseval identity.

We can also rewrite the definition of $\widehat{F}_{\mathbf{r}}$ as follows:

$$\widehat{F}_{\mathbf{r}}(\vec{u}) = \sum_{\vec{v} \in \mathbf{Z}_2^m} (-1)^{\mathbf{r}(\vec{v}) + \langle \vec{u}, \vec{v} \rangle}.$$

This implies the following proposition, which forms the basis for the decoding algorithm of C .

Proposition 2.1 *For any $\mathbf{r} \in \mathbf{Z}_2^n$ and any $\vec{u} \in (\mathbf{Z}_2^m)^*$,*

$$\widehat{F}_{\mathbf{r}}(\vec{u}) = n - 2d(\mathbf{r}, \mathbf{c}_{\vec{u}}).$$

To define a quantum-mechanical implementation of this algorithm, we shall represent vectors as states of quantum systems. An individual bit is represented by a 2-dimensional complex linear space, called qubit. The computation is built as a sequence of unitary operations composed from the two basic blocks, the quantum Walsh-Hadamard transform

$$W_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

and the conditional sign-shift transform

$$P_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

Qubits are combined into larger quantum systems, which is expressed by the direct (tensor) product of the corresponding linear spaces. Likewise, operators acting on these systems will be formed by taking tensor products of the basic operations. For instance, the Walsh-Hadamard transform on a system of m qubits is given by $W_2^m := W_2^{\otimes m}$, etc.

3 The algorithm

The algorithm that we propose depends on the θ introduced in (1)-(2), and a parameter c_0 (a positive integer), which accounts for a tradeoff between the probability of decoding error P_e and the complexity. Namely, if the algorithm is iterated $t = c_0 \ln n$ times, then $P_e = n^{-c}$, where c depends both on c_0 and θ (see Lemma 4.1 below).

Let $\mathbf{r} \in \mathbf{Z}_2^n$ be the received vector. Our decoding algorithm will most probably output the closest code vector of C if $\mathbf{r} \in D_\theta \cap E_\theta$. Note the following consequences of definitions (1)-(2). If $\mathbf{r} \in D_\theta \cap E_\theta$ and $\mathbf{c} = \delta_\theta(\mathbf{r})$ is the code vector closest to \mathbf{r} , then we can solve the inequalities with respect to $d(\mathbf{r}, \mathbf{c})$ to obtain

$$d(\mathbf{r}, \mathbf{c}) \leq n\left(\frac{1}{2} - \theta\right). \quad (4)$$

Next, by (3) we obtain, for any code vector $\mathbf{c}' \neq \mathbf{c}$,

$$d(\mathbf{r}, \mathbf{c}') \geq \frac{1}{2}n\left(\frac{1}{2} + \theta\right). \quad (5)$$

In general, we do not have a more intuitive description of the decoding region. However, it is not difficult to see that

$$\left(\bigcup_{\mathbf{c} \in C} S_{n(\frac{1}{4}-\theta)}(\mathbf{c}) \right) \subset (D_\theta \cap E_\theta),$$

where $S_u(\mathbf{c})$ is the metric ball of radius u centered at \mathbf{c} , and that the inclusion is proper.

The decoding algorithm $A(c_0, \theta)$ accepts as input a received vector \mathbf{r} and consists of the following steps.

1. Initialize the system to be in the zero-state $|\vec{0}\rangle|0\rangle$ where the first register has m qubits and the second one is a 1-qubit register.
2. Apply the transform W_2^m to the first register. Formally this corresponds to computing

$$\sigma_1 := (W_2^m \otimes I_2)(|\vec{0}\rangle|0\rangle) = W_2^m |\vec{0}\rangle I_2 |0\rangle = 2^{-\frac{m}{2}} \sum_{\vec{v} \in \mathbf{Z}_2^m} |\vec{v}\rangle |0\rangle,$$

where I_2 denotes the 2-dimensional identity transform.

3. In quantum parallelism, store the v -th bit $\mathbf{r}(\vec{v})$ of the received word \mathbf{r} in the second register, producing

$$\sigma_2 := 2^{-\frac{m}{2}} \sum_{\vec{v} \in \mathbf{Z}_2^m} |\vec{v}\rangle |\mathbf{r}(\vec{v})\rangle.$$

The unitariness of this computation has been discussed in many related works (see, e.g., [10]).

4. Apply the conditional sign-shift transform P_2 to the second register:

$$\sigma_3 := (I_2^m \otimes P_2)\sigma_2 = 2^{-\frac{m}{2}} \sum_{\vec{v} \in \mathbf{Z}_2^m} (-1)^{\mathbf{r}(\vec{v})} |\vec{v}\rangle |\mathbf{r}(\vec{v})\rangle.$$

5. Apply the inverse transform of Step (3) to the second register, producing

$$\sigma_4 := 2^{-\frac{m}{2}} \sum_{\vec{v} \in \mathbf{Z}_2^m} (-1)^{\mathbf{r}(\vec{v})} |\vec{v}\rangle |0\rangle.$$

6. Apply the transform W_2^m to the first register:

$$(W_2^m \otimes I_2)\sigma_4 = 2^{-m} \sum_{\vec{v} \in \mathbf{Z}_2^m} (-1)^{\mathbf{r}(\vec{v})} \sum_{\vec{u} \in (\mathbf{Z}_2^m)^*} (-1)^{\langle \vec{u}, \vec{v} \rangle} |\vec{u}\rangle|0\rangle = \sum_{\vec{u} \in (\mathbf{Z}_2^m)^*} \frac{\widehat{F}_{\mathbf{r}}(\vec{u})}{n} |\vec{u}\rangle|0\rangle,$$

where the last equality follows by Definition 2.1. From the Parseval identity we see that the last superposition is well defined.

7. Measure the first register, obtaining a certain vector \vec{u} .
8. Repeat Steps (1)-(7) $t = c_0 \ln n$ times. Set the decoding result to the vector \vec{u} obtained the greatest number of times by the measurements in Step (7).

Except for Steps 3,5 and 8, the amount of quantum bit operations performed in the computation is clearly $O(m) = O(\log n)$ and the computation can be implemented by circuits of size $O(\log n)$. Step 3 can be implemented using an input sub-circuit of size $O(n)$ that takes time $O(\log n)$, and so can Step 5, the inverse of Step 3. Since these steps need to be repeated $t = O(\log n)$ times (applying the same input sub-circuit), the overall time complexity is $O(\log^2 n)$. Finally we have to find the vector that appears most of the $O(\log n)$ iterations of the algorithm, i.e., to perform Step 8. This can be done by a circuit of size $O(\log^2 n)$ in time $O(\log n)$. Thus the total circuit size is $O(n)$ with computation sub-circuit of size $O(\log^2 n)$.

4 Probability of decoding error

Let $\mathbf{r} \in \mathbf{Z}_2^n$ be a vector. Suppose $\mathbf{r} \in D_\theta(C) \cup E_\theta(C)$ and let $\vec{u}_{\mathbf{r}} \in (\mathbf{Z}_2^m)^*$ be the message that corresponds to the code vector $\delta_\theta(\mathbf{r})$. As above, we denote this code vector by $\mathbf{c}_{\vec{u}_{\mathbf{r}}}$. In this section we prove that the probability P_e that the algorithm fails to find this code vector given \mathbf{r} as the input, behaves as n^{-c} .

We need Chernoff-type bounds on large deviations. The following technical result can be found, for instance, in [8], [1].

Lemma 4.1 *Let X_1, X_2, \dots, X_t be i.i.d. Bernoulli variables such that for $1 \leq i \leq t$, $\Pr[X_i = 1] = p$, where $p \in (0, 1)$. Let $X = \sum_{i=1}^t X_i$ and thus $\mu = \mathbb{E}[X] = pt$. Then for any $0 < \delta < 1$,*

$$\Pr[X < (1 - \delta)\mu] < e^{-\frac{\delta^2 \mu}{2}}.$$

This enables us to prove the following result.

Theorem 4.1 *For any vector $\mathbf{r} \in D_\theta(C) \cap E_\theta(C)$, Algorithm $A(c_0, \theta)$ fails to return $\vec{u}_{\mathbf{r}}$ with probability $P_e \leq n^{-c}$, where $c = 32c_0\theta^6 - 1$.*

Proof: The probability that \vec{u} is obtained in a measurement in Step 7 of the algorithm equals $p_{\vec{u}} = \frac{|\widehat{F}_{\mathbf{r}}(\vec{u})|^2}{n^2}$. Let $X_i(\vec{u})$, $1 \leq i \leq t$, be the Bernoulli random variables such that $X_i(\vec{u}) = 1$ according as \vec{u} is the result of the measurement in iteration i or not. Let

$$X(\vec{u}) = \sum_{i=1}^t X_i(\vec{u}).$$

Then the expected number of times that \vec{u} is obtained in t measurements is $\mathbb{E}[X(\vec{u})] = p_{\vec{u}}t$. We have

$$\begin{aligned}\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})] &= (p_{\vec{u}_{\mathbf{r}}} - p_{\vec{u}})t = \frac{t}{n^2} \left[|\widehat{F}_{\mathbf{r}}(\vec{u}_{\mathbf{r}})|^2 - |\widehat{F}_{\mathbf{r}}(\vec{u})|^2 \right] \\ &= \frac{t}{n^2} (|\widehat{F}_{\mathbf{r}}(\vec{u}_{\mathbf{r}})| - |\widehat{F}_{\mathbf{r}}(\vec{u})|)(|\widehat{F}_{\mathbf{r}}(\vec{u}_{\mathbf{r}})| + |\widehat{F}_{\mathbf{r}}(\vec{u})|).\end{aligned}\quad (6)$$

Now, since $\mathbf{c}_{\vec{u}_{\mathbf{r}}}$ is the closest code vector to \mathbf{r} and since the covering radius of C is $\frac{n}{2}$, we have

$$\widehat{F}_{\mathbf{r}}(\vec{u}_{\mathbf{r}}) = n - 2d(\mathbf{c}_{\vec{u}_{\mathbf{r}}}, \mathbf{r}) \geq 0.$$

We would like to replace the $\widehat{F}_{\mathbf{r}}$'s in (6) with distances to code vectors. In principle, we need to consider two cases depending on the sign of $\widehat{F}_{\mathbf{r}}(\vec{u})$. However, (6) is symmetric w.r.t. this sign. Therefore, we obtain for any $\vec{u} \neq \vec{u}_{\mathbf{r}}$,

$$\begin{aligned}\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})] &= \frac{4t}{n^2} (d(\mathbf{c}_{\vec{u}}, \mathbf{r}) - d(\mathbf{c}_{\vec{u}_{\mathbf{r}}}, \mathbf{r}))(n - d(\mathbf{c}_{\vec{u}_{\mathbf{r}}}, \mathbf{r}) - d(\mathbf{c}_{\vec{u}}, \mathbf{r})) \\ &\geq 4t\theta^2\end{aligned}\quad (7)$$

by (1)-(2). Now we have

$$\Pr[X(\vec{u}_{\mathbf{r}}) - X(\vec{u}) \leq 0] \leq \Pr[X(\vec{u}_{\mathbf{r}}) - X(\vec{u}) \leq (\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})]) - 4t\theta^2].$$

We will show that the right-hand side is at most $n^{-(c+1)}$. This will imply the statement of the theorem since

$$P_e \leq \sum_{\vec{u} \neq \vec{u}_{\mathbf{r}}} \Pr[X(\vec{u}_{\mathbf{r}}) - X(\vec{u}) \leq 0],$$

and there are at most n such \vec{u} in question.

$$\begin{aligned}\Pr[X(\vec{u}_{\mathbf{r}}) - X(\vec{u}) \leq (\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})]) - 4t\theta^2] \\ &\leq \Pr[X(\vec{u}_{\mathbf{r}}) - X(\vec{u}) \leq (1 - 4\theta^2)(\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})])] \\ &\leq \exp(-(4\theta^2)^2(\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})])/2) \\ &\leq \exp(-16\theta^4 \cdot 4t\theta^2/2) \\ &= n^{-32c_0\theta^6} \\ &= n^{-(c+1)},\end{aligned}$$

where the first inequality follows from $\mathbb{E}[X(\vec{u}_{\mathbf{r}})] - \mathbb{E}[X(\vec{u})] \leq t$, and the second inequality follows from Lemma 4.1. \blacksquare

Remarks. 1. Note that the 1st order Reed-Muller code $\text{RM}(1, m)$ equals $C \cup (\mathbf{1} + C)$, and therefore, the algorithm can be applied to this code as well. However, then we can only claim that decoding result is either the correct code vector or its complement.

2. Note that since any binary $[n, n - m]$ linear code can be embedded in the simplex code of length 2^m , our algorithm can be applied to the decoding of general linear codes. However, this does not seem to improve upon the application of the general search method [7].

3. In principle, the θ 's in the definitions of D_{θ} (1) and E_{θ} (2) do not have to be the same. The only change that this would incur is in the expression for c in Theorem 4.1.

5 The real case

Suppose the code C of Sections 2-4 is used for transmission over the Gaussian channel. The geometric problem is translated to the real Euclidean space by mapping code vectors onto the surface of the n -dimensional sphere of radius \sqrt{n} as follows: a vector $\mathbf{c} = (c_0, \dots, c_{n-1}) \in C$ is mapped to the vector $(-1)^{\mathbf{c}} := ((-1)^{c_i}, 0 \leq i \leq n-1)$. Such vectors are used for transmission over the Gaussian channel; on the output of the channel we receive a (generally, arbitrary) vector $\mathbf{r} \in \mathbf{R}^n$. It is known that to minimize the average error probability of decoding one has to find the code vector $\mathbf{c} \in C$ that gives the minimum to $\|\mathbf{r} - \mathbf{c}\|$ over the code. This, in turn, is equivalent to finding the maximum over $\vec{u} \in (\mathbf{Z}_2^m)^*$ of the Walsh-Hadamard transform

$$\hat{F}_{\mathbf{r}}(\vec{u}) =: \frac{1}{\|\mathbf{r}\|} \sum_{\vec{v} \in \mathbf{Z}_2^m} \mathbf{r}(\vec{v})(-1)^{\langle \vec{u}, \vec{v} \rangle} = \frac{n + \|\mathbf{r}\|^2 - \|(-1)^{\mathbf{c}_{\vec{u}}} - \mathbf{r}\|^2}{2\|\mathbf{r}\|^2}.$$

The classical complexity of the best known algorithm for doing this is $O(n \log n)$ real operations. To define a quantum implementation it is possible to extend ideas of Sections 3-4 to the real case. To obtain a meaningful bound on the probability of success we again have to restrict the domain of the decoding mapping. However, this extension encounters one problem which does not allow to claim a complexity gain over the classical algorithm. For this reason we do not include the details. The difficulty arises in the initialization step. Suppose the received vector is given as $\|\mathbf{r}\|^{-1}|\mathbf{r}\rangle = \|\mathbf{r}\|^{-1}r_0\| \mathbf{r}\|^{-1}r_1\| \dots \|\mathbf{r}\|^{-1}r_{n-1}\|$, where each coordinate $\|\mathbf{r}\|^{-1}r_k$ is approximated by an m -bit binary number. We need to perform the transformation

$$\|\mathbf{r}\|^{-1}|\mathbf{r}\rangle \rightarrow \sum_{k=0}^{2^m-1} \frac{r_k}{\|\mathbf{r}\|} |k\rangle.$$

A possible way of doing this relies on the fact that the Fourier transform on the right-hand side can be represented in the form [6]

$$\sum_{k=0}^{2^m-1} \frac{r_k}{\|\mathbf{r}\|} |k\rangle = \prod_{i=0}^m (\alpha_{i0}|0\rangle + \alpha_{i1}|1\rangle),$$

where $\alpha_{i0}^2 + \alpha_{i1}^2 = 1$ for all i . In principle, this enables us to compute the coordinates α_{i0}, α_{i1} from the known $\frac{r_k}{\|\mathbf{r}\|}$'s; however, this is a classical computation, which takes $O(n)$ operations to compute each α_{i0}, α_{i1} . Though the actual decoding performed after this stage is about as easy as in the Hamming case, the overall complexity turns out to be the same as of the classical algorithm.

Acknowledgment. We would like to thank Dorit Aharonov for telling us of [6], and Lov Grover and Francis Zane for helpful discussions on quantum computing.

References

- [1] N. Alon, J.H. Spencer, The probabilistic method. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley & Sons, Inc., New York, 1992. xvi+254 pp.
- [2] A. Barg, Complexity issues in coding theory, in: Handbook of Coding Theory, V.S. Pless and W. C. Huffman, Eds., Vol.1, Amsterdam: Elsevier (1998), pp. 649–754.

- [3] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and weaknesses of quantum computing, *SIAM J. Comput.* **26** (5) (1997), 1510–1523.
- [4] E. Berlekamp, R. J. McEliece and H. C. A. van Tilborg, On the inherent intractability of certain coding problems, *IEEE Trans. Inform Theory*, **IT-29** (3) (1978), 384–386.
- [5] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, *Fortsch.Phys.* **46** (1998) 493-506.
- [6] R. Cleve, W. van Dam, M. Nielsen, A. Tapp, Quantum entanglement and the communication complexity of the inner product function, LANL e-print quant-ph/9708019.
- [7] Lov K. Grover, A fast quantum mechanical algorithm for database search, *Proceedings, 28th Annual ACM Symposium on the Theory of Computing (STOC)*, May 1996, pp. 212-219.
- [8] R. Motwani, P. Raghavan, *Randomized algorithms*. Cambridge University Press, Cambridge, 1995. xiv+476 pp.
- [9] C. K. Rushforth, Fast Fourier-Hadamard transform of orthogonal codes, *Information and Control*, **15** (1969), 33–37.
- [10] P. W. Shor, Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer, *SIAM J. Comput.* **26** (5) (1997), 1484 - 1509
- [11] Special Section on Quantum Computation, *SIAM J. Comput.* **26** (5) (1997).