Two Efficient Quantum Search Algorithms on NMR Quantum Computers

Seiya OKUBO[1] * Tetsuro NISHINO[2] † Kazuo OHTA[2] ‡ Noboru KUNIHIRO[2] §

[1] *The Graduate School of Electro-Communications 1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585, Japan*

[2] *The University of Electro-Communications 1-5-1 Chofugaoka, Chofu, Tokyo, 182-8585, Japan*

Abstract. In this paper, we propose two efficient quantum algorithms suitable for NMR quantum computers in order to invert one-way permutation. The first algorithm employs Grover operation in order to obtain the solution with probability 1. Its complexity is smaller than Grover's algorithm on quantum Turing machines if the accuracy of observation depends on the size of problems. Though the second algorithm is slower than the first algorithm, we can run this algorithm in classical parallel manner. We also pointed out its application to codebreak of secret-key cryptosystems.

Keywords: Grover's Algorithm, Inversion of One-way Permutation, Collision Search, Secret-Key Cryptosystem, Secret-Key Search, NMR Quantum Computer.

1 Introduction

Recently, many researchers are studying how to physically implement quantum computers based on a quantum Turing machine (QTM for short). Among others, NMR (Nuclear Magnetic Resonance) offers an appealing prospect for implementation of quantum computers because of a number of reasons [1]. But, quantum computations performed on NMR is slightly different from those performed on ordinary QTMs. So, we first define bulk quantum Turing machine (BQTM for short) as a model of bulk quantum computation and introduce an accuracy of the observation of NMR computer by ε. Then, we will propose two efficient quantum search algorithms which are executed on NMR quantum computers.

A *bulk quantum Turing machine* (*BQTM* for short) is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, B, F)$, where Q is a finite set of *states*, Γ is a *tape alphabet*, $B \in \Gamma$ is a *blank symbol*, $\Sigma \subseteq \Gamma$ is an *input alphabet*, δ is a *state transition function* and is a mapping from $Q \times \Gamma \times \Gamma \times Q \times \{L, R\}$ to $\mathbf{C}$ (the set of complex numbers), $q_0 \in Q$ is an *initial state*, and $F \subseteq Q$ is a set of *final states*.

In the case of BQTM, axioms for the observation are changed as follows. If a tape cell of a BQTM is in a superposition $\alpha|0\rangle + \beta|1\rangle$, the real number $\theta = |\beta|^2 - |\alpha|^2$ will be observed. Since $|\alpha|^2 + |\beta|^2 = 1$, it follows that $-1 \leq \theta \leq 1$. We assume that this real number θ is represented by a binary number to l places of decimals with a sign, where l is a constant. For example, if $l = 5$, θ may be given by -0.00101. In the sequel, let $\varepsilon = 1/2^l$. Then, if we observe a tape cell of a BQTM in a superposition $\alpha|0\rangle + \beta|1\rangle$, we assume that a real number θ such that $|\beta|^2 - |\alpha|^2 - \varepsilon \leq \theta \leq |\beta|^2 - |\alpha|^2 + \varepsilon$ will be read with probability 1.

Furthermore, we assume that we cannot perform a partial observation in the case of BQTM. When we observe several tape cells of a BQTM, only thing we can do is to get a real number θ at each tape cell, independently. Instead, we assume that the superposition will not be destroyed for a while. That is, we assume that we can observe the tape cells in the superposition some constant times before the superposition is destroyed.

2 The First Algorithm

Here, we consider an efficient execution of Grover's quantum search algorithm [2] on NMR quantum computers, which enjoys the physical feature of NMR computer; the observation of θ is possible without destroy of state and $\theta = |\beta|^2 - |\alpha|^2 \pm \varepsilon$ where $\alpha|0\rangle + \beta|1\rangle$ and ε is the accuracy. In the sequel, we assume that there is only one solution for each instance of the search problem.

The Grover's algorithm on NMR is the same as on QTM. But the value of θ is observed on NMR computer. The algorithm is following:

Step1 Make the flat superposition: $\sum_{x=0}^{N-1} \frac{1}{\sqrt{N}} |x\rangle$

Step2 Apply j Grover's operator G : $\frac{1}{\sqrt{a}} |x_a\rangle + \sum_{X-x_a} \sqrt{\frac{1-a}{N-1}} |x\rangle$ where x_a is the solution and a is the amplitude of the solution.

Step3 Observe each cell and decide value 0 or 1 for each cell from observation. When we observe the i-th cell and obtain θ_a, set the value of the i-th cell to 0(or 1) if $\theta_a > \varepsilon$ (or $\theta_a < -\varepsilon$).

After the amplitude amplification process of the Grover's algorithm, let us assume that we obtain the superposition of the following states :

- the state (say $|1\rangle$) corresponding to the solution with amplitude $\sqrt{a}$,
- the state $|1\rangle$ with amplitude $\sqrt{\frac{1-a}{N-1} * (N/2-1)}$, and
- the state $|0\rangle$ with amplitude $\sqrt{\frac{1-a}{N-1} * (N/2)}$.

Then, the following real value θ will be obtained:

$$\begin{aligned} & a + \frac{1-a}{N-1} * (N/2-1) - \frac{1-a}{N-1} * (N/2) \\ = & \frac{aN-1}{N-1}. \end{aligned}$$

*s-okubo@ice.uec.ac.jp

†nishino@ice.uec.ac.jp

‡ota@ice.uec.ac.jp

§kunihiro@ice.uec.ac.jp

Since the real value $\theta \geq \varepsilon$ can be read, and $a \geq \frac{\varepsilon N-\varepsilon+1}{N}$, we have to have an amplitude greater than $\sqrt{\frac{\varepsilon N-\varepsilon+1}{N}}$ for the solution.

If we apply Grover operator j times, we obtain $a = \sin(2(j+1)\theta)$. Since $\theta \simeq \sin(\theta) \simeq \frac{1}{\sqrt{N}}$ and $\sin^{-1}(\sqrt{\frac{\varepsilon N-\varepsilon+1}{N}}) \simeq \sqrt{\frac{\varepsilon N-\varepsilon+1}{N}}$. , it follows that

$$j > \frac{1}{2}\sqrt{\varepsilon N - \varepsilon + 1} - 1.$$

The number of the applications of Grover's operator is required $\frac{1}{2}\sqrt{\varepsilon N-\varepsilon+1}$ on NMR computer, and this number is smaller than the number of the applications of Grover's operator on QTM with success probability $\frac{1}{2}$.

Moreover, our result means that the query complexity of computing OR of all the values of a Boolean function with probability 1 using an NMR quantum computer is only $O\left(\sqrt{\varepsilon N-\varepsilon+1}\right)$. It is shown that the lower bound of the query complexity of computing OR of all the values of a Boolean function with probability 1 using the quantum oracle setting is $\Theta(N)$. Hence BQTM may be stronger than QTM.

3 The Second Algorithm

3.1 The Proposed Algorithm

We will use the following notations: Let T_0 be the threshold values with which BQTM (NMR) can distinguish the case where $t = 0$ and the case of $t > 2^{(n-l)} = T_0$. Here t is an approximate value of the counting problem and l is a accuracy of observation.

$$\begin{aligned} &f : X \to X \quad \text{is a permutation.}\\ &y_0 \in X \quad \text{is given in advance.}\\ &(x_H, x_L) \quad \text{is a string concatenation.}\end{aligned}$$

Our perpose is to search x_0 satisfying $y_0 = f(x_0)$.

Consider the domain of f as the product set of $X = X_H \times X_L$. Guess the lower part of $x_0(x_0^L)$ as z, and introduce the reducing map, r_z from $X = X_H \times X_L$ to $X_H \times \{z\}$, where the size of $r_z^{-1}(x_1, x_2)$ can be distinguished as T_0 from 0 using NMR computers. Define a new Boolean function F using f, r_z and y_0 as follows:

$$g_z(x) = f \circ r_z(x) \text{ and } F(x) = 1 \text{ iff } g_z(x) = y_0.$$

Boolean function F can be considered as two variable function of $x = (x_H, x_L)$ where $F : X \ni x \mapsto F(x) \in \{0, 1\}$, $x = (x_H, x_L) \in X = X_H \times X_L$, $x_H \in X_H$ and $x_L \in X_L$.

Here, the sizes of two sets, X_H, X_L, are selected considering the value of T_0. Remember that T_0 satisfies the following property: NMR can distinguish two cases of "$\#F^{-1}(1) > T_0$" and "$\#F^{-1}(1) = 0$".

The proposed procedure is as follows:

Phase I We decide x_0^L in this phase.

Step1 Make a flat superposition $\sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} |x\rangle |0\rangle$.

Step2 Choose the $z \in X_L$ as a candidate of x_0^L, and define r_z by $r_z : X \ni (x_H, x_L) \mapsto (x_H, z) \in X_H \times \{z\} \subset X$. Define g_z by $g_z = f \circ r_z$, that is, $g_z(x) = f \circ r_z(x)$. Define a boolean function F using g_z by $F(x) = 1$ iff $g_z(x) = y_0$.

Step3 Write $F(x)$ in the $(n + 1)$-st cell $\sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} \frac{1}{\sqrt{N}} |x\rangle |F(x)\rangle$

Step4 Observe the $n + 1$-st cell.
If the number of $x \in X$ such as $F(x) = 1$ is more than T_0 then we get z as x_0^L and go to next step, else goto step1.

Phase II Repeat the following operations between step5 and step7 for $i = \log(\#X_H) - 1$ to 0

Step5 Make a flat superposition (initial state) $\sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} \frac{1}{\sqrt{N}} |x\rangle |0\rangle$

Step6 Write $F_i(x)$ in the $(n + 1)$-st cell, where $F_i(x) = 1$ iff $f(x_0 \cdots x_{i-1} \cdot 0 \cdot z) = y_0$

$$\sum_{x_1=0}^{1} \cdots \sum_{x_n=0}^{1} \frac{1}{\sqrt{N}} |x\rangle |F_i(x)\rangle$$

Step7 Observe the $(n + 1)$-st cell.
If the number of $x \in X$ such as $F_i(x) = 1$ is more than T_0 then we set $0 \cdot z$ to new z else we set $1 \cdot z$ to new z

Step8 Output z as answer.

3.2 Complexity

Let $T_0 = \varepsilon N$ and let a set X_L satisfy $\#X_L = T_0 = \varepsilon N$. Then $\#X_H = \frac{1}{\varepsilon}$ holds, since $\#X_H \times \#X_L = N$.

We will estimate the complexity of the proposed algorithm by the number of oracle calls of O_f and O_{r_z} , which can be implemented by standard way based on the specification of function f and r_z.

Phase I : $\#X_L = \varepsilon N$

Phase II : $\log \#X_H = -\log \varepsilon$ (Note that $-\log \varepsilon > 0$ since $0 < \varepsilon < 1$)

Total : $\varepsilon N - \log \varepsilon$

If $\varepsilon < N^{-\frac{1}{2}}$ then $n > -\log \varepsilon$ since $2^N = N > \#X_H = \frac{1}{\varepsilon}$. Thus $\varepsilon N - \log \varepsilon < N^{1-\frac{1}{2}} + n \simeq N^{\frac{1}{2}}$. As the result, this algorithm can find the solution faster than Grover's algorithm on QTM.

References

[1] Chuang, I. L., Gershenfeld, N., Kubinec, M.G., and Leung, D.W. , *Proc. R. Soc. Lond.*, Vol. A 454, pp.447-467 (1998).

[2] Grover, L. , *Proc. 28th Annual ACM Symposium on Theory of Computing* , ACM, New York, 1996, pp.212-219.