Quantum Query Complexity and the Number of Inverted States

Kazuo Iwama[†,††], Akinori Kawachi[†,††], Hiroyuki Masuda[†,††],
Rudy Raymond H.P.[†,††] and Shigeru Yamashita[†,†††]
[†]Quantum Computation and Information, ERATO,
Japan Science and Technology Corporation (JST)
Matsuo Building 2F, 406, Iseya-cho,
Kamigyou-ku, Kyoto, Kyoto 602-0873 Japan
[††]Graduate School of Informatics, Kyoto University
[†††]NTT Communication Science Laboratories
E-mail: {iwama, kawachi, hiroyuki, raymond, ger}@kuis.kyoto-u.ac.jp.

1 Introduction

Ambainis introduced a simple but a powerful lower bound technique called *quantum arguments* [1]. In this abstract we show an application of Ambainis's technique to obtain the lower bound for the *decoding problems* which are simple generalization of the quantum computations in the quantum query model [1, 2]. Although our lower bound can be seen as just a simple generalization of Ambainis's result, it relates to the lower bound of query complexity with respect to "*the number of quantum states inverted by one oracle call.*" (We can consider intuitively that a quantum oracle inverts the amplitude of some quantum basis states related to the oracle when it is called with an input quantum state where all the possible inputs are superposed. We call such basis sates "inverted states.") We also show that there exists an oracle whose query complexity is the same as our lower bound. Our results indicate that the number of inverted sates has a correlation to the query complexity.

2 Query Complexity for Decoding Problems

In the *quantum query model* [1, 2], we compute a function $f(x) : \{0,1\}^N \to \{0,1\}^n$ ($N = 2^n$) and the input bit string, say x, can be accessed only by queries to an quantum oracle (the formal definition will be given later). For our purpose mentioned in the previous section, i.e., to discuss the relation between the query complexity and the number of inverted quantum states, we extend the quantum query model to the following oracle-based quantum computation: Suppose we are given a *coding function* $C : \{0,1\}^n \to \{0,1\}^N$ and one of the codeword $C(x)$ as an input to our task. Also suppose we are only allowed to access the oracle which tells us one bit of the given $C(x)$ by one query. Then our task is to determine x. The number of oracle queries is the complexity needed to determine x. Let $C_i(x)$ be the i-th bit of the codeword $C(x)$.

By the above formulation, we can have the following theorem which can be proved in the same way as in [1].

Theorem 1. Suppose we are given a *coding function* $C : \{0,1\}^n \to \{0,1\}^N$. Let l_i be the number of x such that $C_i(x) = 1$, and $P = max_i\{l_i \cdot (N - l_i)\}$. Then any quantum algorithm that determines x with probability $1-\varepsilon$ uses $\Omega(\frac{N}{\sqrt{P}})$ oracle queries for some fixed $\varepsilon < 1/2$.

We can define the oracle transformation corresponding to an input codeword $C(x)$ as a mapping from $|i, b, z\rangle$ to $(-1)^{b \cdot C_i(x)}|i, b, z\rangle$, where i is n bits (we refer to this part as "index register."), b is one bit and z consists of all other available qubits for the algorithm. Let the number of 1's in a bit string x be $N(x)$. Then $N(C(x))$ equals to the number of inverted quantum states in the index register space by calling the quantum oracle corresponding to $C(x)$ when the index register is superposed with all the possible indices. That is, according to our definition, if we call the oracle with the index register $\sum_{j=0}^{2^n-1} \frac{1}{\sqrt{2^n}}|j\rangle$, the amplitude of a quantum state $|j\rangle$ is inverted if the $C_j(x) = 1$, and the number of such quantum states is $N(C(x))$.

Let us discuss the query complexity for the oracle (codeword $C(x)$) such that $N(C(x))$ is $O(K)$ below. Suppose $K > n$ and let us consider a coding function $F(x) : \{0,1\}^n \to \{0,1\}^N$ such that $F_i(x) = x_i$ for all $i \leq n$ and $F_i(x) = 1$ for $(n+1) \leq i \leq n + K - N(x)$, where $F_i(x)$ and x_i mean the i-th bit of $F(x)$ and x, respectively. Then, it is obvious that $N(F(x))$ is K for all x. It

is also easy to show that the query complexity is $O(n)$ since the first n bits of $C(x)$ equals to x itself. In other words, the above code $F(x)$ contains important information only in the first n bits, and thus it is enough to query the first n bits for determining x.

To have a proper discussion about the query complexity, we would like to exclude such coding functions that have much more information in some bits than other bits since the algorithm may utilize such information and the query complexity may decrease as the case of the above code $F(x)$.

If there is no prior useful information about the coding for the algorithm, the number of x such that $C_i(x) = 1$ should be $O(K)$ for all i. (Otherwise, the algorithm might know which bit is more useful to ask than the other bits.)

Therefore, we impose the following restriction on the coding.

Definition 1. Let l_i be the number of x such that $C_i(x) = 1$. Then a *proper coding function* $C(x)$ is a coding function such that $l_i = O(K)$ for all i, where $K = max_x\{N(C(x))\}$.

Intuitively, any algorithm cannot distinguish the probability (over x) between different bits that the oracle answers 1 if the coding function is proper, i.e., we only know that the probability that the oracle answers 1 is equivalently $O(K/N)$ for all bits. Note that the code $F(x)$ mentioned above does not satisfy the condition of Definition 1 since $l_i = N/2$ for $i \leq n$.

Then by Theorem 1, the following is immediate.

Theorem 2. Suppose we are given a *proper* coding function $C : \{0,1\}^n \to \{0,1\}^N$ such that $N(C(x)) = O(K)$ for $K \leq N/2$. (Without loss of generality we can assume $K \leq N/2$.) Then any quantum algorithm that determines x with probability $1 - \varepsilon$ uses $\Omega(\sqrt{\frac{N}{K}})$ oracle queries for some fixed $\varepsilon < 1/2$.

Next we show an oracle whose query complexity is the same as Theorem 2. For the purpose, we consider the following *coding function* $H^k(x)$ with parameter k, $H^k : \{0,1\}^n \to \{0,1\}^N$ such that $H^k_i(x) = 1$ if $B_j(x) = B_j(i)$ for all $j \leq n - k$ and $\bigotimes_{j=n-k+1}^{n}(B_j(x) \cdot B_j(i)) = 0$, where $B(x)$ denotes the binary representation of x and $B_i(x)$ denotes the i-th bit from the left of $B(x)$. Then there exists a quantum algorithm that determines x in our computational model with $O(\sqrt{2^{n-k}})$ queries. We show an outline of the algorithm below.

Let the $(n-k)$ bit string corresponding to the first $n-k$ bits of x be x_l. If the oracle is called with the index register $|i\rangle$ where $B_j(i)$ is fixed to 0 for each $i \geq n-k+1$, the oracle works as an Equality (EQ) oracle to find a bit string x_l in the first $n-k$ bits of the index register. Therefore, by using the oracle we can perform the Grover search for a search problem among 2^{n-k} possible solutions. Thus we can determine x_l by $O(\sqrt{2^{n-k}})$ oracle calls.

After determining x_l, if the oracle is called with the index register $|i\rangle$ where the first $n-k$ bits of i are fixed as exactly same as x_l, the oracle works as exactly opposite to an Inner Product (IP) oracle in the $n-k+1$-st to the n-th bits of the index register. Therefore, by using this oracle only once, we can determine the remaining k bits of x by the same algorithm in [3].

Concerning the coding function $H^k(x)$, it is easy to show that $N(H^k(x)) = 2^k$ or 2^{k-1} for all x, and the coding function is proper. Therefore, we have the following Theorem if we assume $K = 2^k$.

Theorem 3. There exist a proper *coding function* $H(x)$ such that $N(H(x)) = O(K)$ for all x and the query complexity to determine x is $O(\sqrt{\frac{N}{K}})$.

Accordingly, by Theorem 2 and Theorem 3, we conclude that the number of inverted states, say K in the theorems, has a strong correlation to the query complexity if we consider only the proper coding function. It should be noted that the above correlation is also true for an EQ oracle problem (Grover search) and IP problems [3]. (For an EQ oracle problem $N(C(x)) = 1$ for all x, and $N(C(x)) = N/2$ for an IP oracle problem. ($x = 0$ is an exception.) For an EQ oracle problem, the necessary queries for the problem is $\Theta(\sqrt{N})$ whereas one query is enough for an IP oracle problem.)

However, of course, only the number of quantum states inverted by one oracle call does not relate to the query complexity. Indeed we need some conditions, such as the condition for the proper coding, to have our claim. We think the condition is a necessary but not sufficient condition for excluding meaningless oracles. Therefore, we should refine the condition further. As another future work, to develop other kinds of oracles which need $O(\sqrt{N/K})$ queries might be interesting.

References

[1] A. Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of 32th ACM Symposium on Theory of Computing*, pages 636–643, 2000.

[2] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R de Wolf. Quantum lower bounds by polynomials. In *Proc. 39th Annual IEEE Symposium on Foudations of Computer Science*, pages 352–361, 1998.

[3] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, October 1997.