Report of Quantum Oracle Interrogation

Name: Jing Zexuan

Student ID: 20648303

Section: CO481/CS467/PHYS467

Abstract

This is a report of a published paper. The paper I am reviewing is *Quantum Oracle Interrogation: Getting All Information for Almost Half the Price* https://ieeexplore.ieee.org/document/743486 . The paper has two parts content. In the first part, the author shows that how $N/2+\sqrt{N}$ calls to the oracle are sufficient to guess the whole content of the oracle (a black box that is able to be encoded as a N-bit string) with probability greater than 95% given that we have a quantum computer. This exceeds the performance of classical computers which would require N calls to achieve the same task with the same success probability. As a consequence, we can conclude that any function with the N bits of the oracle as input can be calculated using $N/2+\sqrt{N}$ queries if a small probability of error is allowed(in this case, less or equal than 5%). Moreover, in this paper, the author shows that this error probability can be made arbitrary small by using $N/2+O(\sqrt{N})$ oracle queries. In the second part of the article, the discussion is turned to “approximate interrogation”. This is the case when only a certain fraction of the N oracle bits are requested. Also we can analyze that under this scenario, the quantum algorithm also outperforms the classical protocols. In the last part of the paper, an example procedure is given that could return a string where 80% of the bits are correct with N/10 queries. As a contrast, any classical protocol would need 6N/10 queries to achieve such a correctness ratio.

Introduction

- What is the problem being solved?

The black-box or oracle problems can be generalized by given a secret function (with some fixed domain size) where we have the access to query this oracle with some input and get corresponding output but the structure of this oracle is unknown, using the required amount of queries in order to decide some general property of the function. We are aiming to derive algorithms that reduce the number of queries to the oracle as much as possible in order to achieve a high efficiency. In this paper the author focuses on how do we come up with an algorithm that applies $N/2+\sqrt{N}$

queries to the oracle and are sufficient to guess the whole content of the oracle (a black box that is able to be encoded as a N-bit string) with probability greater than 95% and how quantum computing can outperform the classical protocols in terms of “approximate interrogation”.

- Why is it interesting?

In the quantum computing field, one of the key problem that computer scientists focus on is how do quantum computing protocols can differ from classical computing methods and whether we could achieve some powerful performance that classical computing cannot. Among all sorts of different problems, oracle interrogation is an important one because it is a crucial fundamental problem which could contribute to variety of applications in different areas. Back to the time when the author wrote this paper, it still remains an open question of what number of queries(the price) can we achieve to compute the oracle(get all information) with some fixed size. And of course, we are curious of how does the quantum oracle interrogation algorithm perform comparing to classical computing algorithms. Furthermore, could we sacrifice some accuracy to get an approximation about the oracle i.e. to approximate the content of the oracle with small error probability. Any solution to those problems would be meaningful for the practical application of quantum computers.

- What was known before?

Some research[1, 4, 6] about quantum computation complexity which we have already discussed in this course have revealed several lower bounds on the capability of quantum computing procedures queries with the black-box setting. For example, suppose we want to compute the representation of the black-box function F with bounded error. Let $Q_E(F)$ denotes the numbers of queries required by any quantum procedure to compute F with certainty(i.e. with probability 1) and deg(F) denotes the minimum degree of a real multi-linear polynomial P that represents F. According to corollary of theorem(BBCMW): “***The probability of outputting 1 is a real multi-linear polynomial in the input variables of degree at most 2T*** ”[1, 4], we have $Q_E(F) >= deg(F)/2$. In other words, for a quantum procedure, N/2 queries to the oracle is still necessary. Also those researches[1] have shown that for the calculation of some specific functions(the bitwise or for example) with certainty, still N queries are required.

- Outline for the report

1. I will provide some technical backgrounds in order to help us understand the paper.
2. Describing the quantum algorithm by using $N/2 + \sqrt{N}$ queries to the oracle and compute the secret function of the black-box with success probability greater than 95% given an oracle of domain size N. Then by analyzing this algorithm we can derive that an arbitrary small error probability can be achieved with lower bound $N/2 + O(\sqrt{N})$.
3. Describing a quantum procedure to approximate oracle interrogation and compare the procedure to the classical computing methods performance by discussing two example problems.

4. We draw the conclusion of this report, describe some existing open problems and recommend some future research based on this paper.

1. Preliminaries

Phase Kickback Trick

Recall from what we have discussed in the course. Generally speaking, suppose we have an unknown function value f : {0,1}->{0,1}. We would like to compute a function into the phase $(-1)^f$ by only calling the function once. By the following procedure, we can achieve such a goal. Suppose we start in the state $|\phi\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ and we could implement a controlled U_f gate and after applying such gate we get the following result states: $(-1)^f|\phi\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ .In this paper specifically, the author chooses to add (modulo 2) the value of f to the last bit. In this way, we also obtain the same effect and could compute the phase correctly.

The Hadamard Transformation

Recall the Hadamard transformation gives the following map:

$|0\rangle \rightarrow \frac{|0\rangle+|1\rangle}{\sqrt{2}}$

$|1\rangle \rightarrow \frac{|0\rangle-|1\rangle}{\sqrt{2}}$

since it is reversible, we also have $\frac{|0\rangle-|1\rangle}{\sqrt{2}} \rightarrow |1\rangle$. In more general case, we could derive Hadamard transformation in several bits. Consider a three quits system:

$$H\otimes H\otimes H|x1x2x3\rangle = \sum_{y\in\{0,1\}^3}(-1)^{x\cdot y}\frac{1}{\sqrt{8}}|y1y2y3\rangle$$

$$H\otimes H\otimes H \sum_{y\in\{0,1\}^3}(-1)^{x\cdot y}\frac{1}{\sqrt{8}}|y1y2y3\rangle = |x1x2x3\rangle$$

By mathematical induction, we could derive N bits Hadamard transformation where N is arbitrary natural number. From the above transformation, we can learn that in order to compute the secret string y it is sufficient to get a superposition together with phase values $(-1)^{x\cdot y}$ for every x. In fact, some previous research have already achieved this result before, detailed discussion can be found in [2, 3, 5, 7].
Now we have all the technical backgrounds set and could start to discuss the algorithm.

2. The Quantum Algorithm

- First, we give the algorithm:

1. Initial state preparation: Prepare a register of N + 1 qubits in the state $|\phi_{\lfloor N/2+\sqrt{N}\rfloor}\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ as the initial state.
2. Oracle calls: Apply the A_k procedure, for k = $\lfloor N/2+\sqrt{N}\rfloor$ oracle queries.
3. Hadamard transformation: Perform N Hadamard transformations to the first N qubits on the register.

4. Final observation: Observe the same first N qubits in the standard basis {|0⟩, |1⟩}. The outcome of this observation is our guess for the oracle description $\omega_1 \ldots \omega_N$. This estimation of $\omega_1 \ldots \omega_N$ will be correct for all N bits with a probability greater than 95%.

- Explanation:
In this algorithm, we prepare the state ϕ_k that is an equally weighted superposition of bit strings of size N with Hamming weight less than or equal to k and an additional qubit in state $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ attached to it:

$$|\phi_k\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{M_k}}\{\sum_{x\in\{0,1\}^N}^{|x|\leq k}|x\rangle\}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

Note M_k is the appropriate normalisation factor calculated by the number of x strings that have Hamming weight less than or equal to k:

$$M_k = \sum_{i=0}^{k}\binom{N}{i}$$

Instead of doing brute force calculation of the phase values $(-1)^{x\cdot\omega}$ for all x, we will only calculate for the strings $x_1 \ldots x_N$ whose Hamming weights(the number of ones in a bit string) are above a certain threshold k. In this way, the number of necessary oracle calls is reduced and we can obtain an outcome which is still very close to the "perfect state" (but now for ω instead of y). Just as what we mentioned in the Preliminaries previously, the value $(-1)^{x\cdot\omega}$ corresponds to the parity of a set of ω_i bits, where this set is determined by the ones in the string $x_1 \ldots x_N$. To calculate the parity we can perform a sequence of additions (modulo 2) of the relevant ω_i values. As discussed in the Preliminaries, this leads to the result that each ω_i can be obtained by a single oracle O_x query. More exactly, access to an oracle O_x implements:

$$|x\rangle|b\rangle \rightarrow |x\rangle|b\oplus(x\cdot\omega_i)\rangle$$

Therefore the Hamming weight equals the "oracle queries complexity".
Furthermore, we reduce the number of oracle calls by setting a threshold number k. We define the A_k procedure as follows:

$A_k|x\rangle|b\rangle \rightarrow |x\rangle|b\oplus(x\cdot\omega_i)\rangle$ if hamming weight |x| <= k
$A_k|x\rangle|b\rangle \rightarrow |x\rangle|b\rangle$ if hamming weight |x| > k

In other words, the algorithm performs a conditional parity calculation for hamming weight of x is less than or equal to k.
Note, we can observe that this procedure A_k requires at most k oracle calls for every string $x_1 \ldots x_N$.
Therefore apply A_k procedure to the initial state yields following result:

$$A_k|\phi_k\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{M_k}}\{\sum_{x\in\{0,1\}^N}^{|x|\leq k}(-1)^{x\cdot\omega}|x\rangle\}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

We have successfully transferred the desired information about $\omega_1 \ldots \omega_N$ to phases of the state $A_k|\phi_k\rangle$. Now the accuracy of this algorithm depends on the choice of k. If we set k to its maximum k = N, then, the algorithm would give us exactly the state |ω⟩. On the other hand the minimum value k = 0 leads to a state that does not reveal anything about |ω⟩. Any other possible value between those two values 0 and N, will yield a resulting state which is close to |ω⟩ but with some error.

- Analysis:
Consider the resulting state we get after applying the A_k procedure:

$$\frac{1}{\sqrt{M_k}}\{\sum_{x\in\{0,1\}^N}^{|x|\le k}(-1)^{x\cdot\omega}|x\rangle\}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

The probability of this state matches the target state $|\phi_N\rangle$ perfectly is:
Prob(A_k returns perfect ω) = $|\langle\phi_k|\phi_N\rangle|^2$
Since $M_k = \sum_{i=0}^{k}\binom{N}{i}$, we yield following formula:

Prob(A_k returns perfect ω) = $\frac{M_k}{2N} = \frac{1}{2N}\sum_{i=0}^{k}\binom{N}{i}$

the above equality is the reason why the algorithm also works for values of k around $N/2+\sqrt{N}$. When N is large we can assume the binomial distribution approaches the Gaussian distribution which means k has to be bigger than the average N/2 by some multiple of the standard deviation $\frac{\sqrt{N}}{2}$ of the Hamming weights over the set of bit strings. By Gauss distribution probability table:
Prob($A_{\lfloor N/2+\sqrt{N}\rfloor}$ returns perfect ω) > 95%
Thus the algorithm guess the whole content of the oracle perfectly with probability greater than 95% by $\lfloor\frac{N}{2}+\sqrt{N}\rfloor$ queries.

- Future Discussion:
Note the relation between error probability and threshold k is exponential:
Prob(k = N/2 + $\lambda\sqrt{N}$) $\approx \frac{1}{2}+\frac{1}{2}Erf(\sqrt{2}\lambda)$ = O($2^{-\lambda^2}$)
The standard approximations of the binomial distribution by the Gaussian distribution tells us as the threshold increases, the error-probability goes to zero. Thus, error probability can be made arbitrary small by using $N/2+O(\sqrt{N})$ oracle queries. Note this outperforms classical computing since we have:
Prob(classical returns perfect ω) $\le \frac{1}{2^{N-k}}$
which makes almost N queries to reach greater 95% success probability.

3. Approximate Interrogation

In this part, we ask what if we want to know only a certain fraction of the N unknown bits. Because when dealing with the real world practical problems, sometimes we do not demand an absolutely correct answer where faster approximation algorithms with high success probability are preferred. So it is worth exploring: given a threshold of k oracle queries, what is the maximum expected number of correct bits c that we can obtain via an 'approximate interrogation' procedure if we assume ω to be totally random? Based on the above algorithm, we can approach this goal efficiently. To start with, we analyze the classical circumstance.

- Classical Approximate Interrogation:
To approximate interrogation under classical situation is straightforward. Suppose we query k out of N bits, then we know k bits with certainty and we have 50%

probability to guess N - k bits correctly. Therefore the expected number of correct bits is:

$$\frac{N+k}{2} \qquad \text{(equation 1)}$$

- Quantum Approximate Interrogation:

In order to do quantum approximate interrogation, we can apply the same quantum algorithm previously discussed but with a different initial state:

$$|\phi_k^{\alpha}\rangle = \sum_{j=0}^{k} \alpha_j \cdot \frac{1}{\sqrt{\binom{N}{j}}} \sum_{x\in\{0,1\}^N}^{|x|\le k} |x\rangle \quad \text{where } \sum_j \alpha_j{}^2 = 1 \text{ (i.e. normalized)}$$

In this case the expect number of correct bits are:

$$\frac{N}{2} + \sum_{j=0}^{k-1} \alpha_j \alpha_{j+1} \sqrt{j+1}\sqrt{N-j} \qquad \text{(equation 2)}$$

It is necessary to show how could we derive equation 2:

We calculate by assuming that the unknown bit string consists of zeros only(i.e. ω = 0). This leads to that the expected number of correct bits for the algorithm equals the expected number of zeros of the observed output string y because without loss of generality we can make the assumption ω = 0 which concludes that this number will the expected number of correct bits for any ω. In this case, apply A_k to $|\varphi\rangle$ will not change the initial state:

$$A_k|\phi_k^{\alpha}\rangle = \sum_{j=0}^{k} \alpha_j \cdot \frac{1}{\sqrt{\binom{N}{j}}} \sum_{x\in\{0,1\}^N}^{|x|=j} |x\rangle$$

Afterwards, we apply N Hadamard transformation on all N qubits to get:

$$H^{\otimes N} A_k|\phi_k^{\alpha}\rangle = \sum_{j=0}^{k} \alpha_j \cdot \frac{1}{\sqrt{\binom{N}{j}}} \sum_{x\in\{0,1\}^N}^{|x|=j} H^{\otimes N}|x\rangle$$

$$= \frac{1}{\sqrt{2^N}} \sum_{y\in\{0,1\}^N} \sum_{j=0}^{k} \alpha_j \cdot \frac{1}{\sqrt{\binom{N}{j}}} \sum_{x\in\{0,1\}^N}^{|x|=j} (-1)^{y\cdot x}|x\rangle$$

The probability of observing a certain string y depends only on its Hamming weight |y|. This combines with the equation:

$$|\phi_k\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{M_k}}\{\sum_{x\in\{0,1\}^N}^{|x|\le k} |x\rangle\}\frac{|0\rangle-|1\rangle}{\sqrt{2}}$$

would give us the expression of the expected number of zeros:

$$\mathrm{E} = \sum_{t=0}^{N} t \cdot \binom{N}{t} |\langle 0^t 1^{N-t}|H^{\otimes N} A_k|\phi_k^{\alpha}\rangle|^2$$

$$= \frac{N}{2} + \sum_{j=0}^{k-1} \alpha_j \alpha_{j+1} \sqrt{j+1}\sqrt{N-j}$$

Thus this E is the expected number of correctly guessed bits for the quantum protocol as required.

With the above two equations we could make improvement to the classical methods. Next we give two examples.

- Example1 Interrogation with One Quantum Query:
Suppose in a quantum computer we only query to the oracle once, then according to equation 2 expected number of correct bits is maximised by choosing $\alpha_0 = \alpha_1 = 1/\sqrt{2}$, and equals $\frac{N}{2} + \frac{\sqrt{N}}{2}$. However, in order to achieve the same expected number of correct bits, according to equation 1, necessarily we need $\sqrt{N}$ queries. This is a very remarkable improvement, notice we obtain a O($\sqrt{N}$) speed up by quantum interrogation!

- Example2 Interrogation with Many Queries:
We first assume N is large(i.e. $\frac{\sqrt{N}}{N} \approx 0$) and k is a fraction of N such that $0 \le \frac{k}{N} \le \frac{1}{2}$.
Next, define amplitudes α as following:

$$\alpha_j = 0 \quad if\ 0 \le j \le k - \sqrt{k}$$

$$\alpha_j = \frac{1}{4\sqrt{k}} \quad if\ k - \sqrt{k} \le j \le k$$

This gives us the expected ratio of correct bits:

$$\frac{E_{quant}}{N} = \frac{1}{2} + \frac{1}{N\sqrt{K}} \sum_{j=k-\sqrt{k}}^{k-1} \sqrt{j+1}\sqrt{N-j} \approx \frac{1}{2} + \sqrt{\frac{k}{N}(1 - \frac{k}{N})}$$

Note for $\frac{1}{2} < \frac{k}{N} \le 1$ we use amplitudes where k = N/2.
Under this setting, if we calculate te expected ratio of correct bits of classical methods:

$$\frac{E_{classi}}{N} = \frac{1}{2} + \frac{k}{2N}$$

Similar to the Example 1, we can observe that the quantum algorithm performs better than the classical one, especially when k/N is small. For example, if we allow the quantum protocol to query N/10 times, then we can expect 80% of the bits to be correct. But any classical method would require 6N/10 queries to obtain the same ratio. In this example we get O($\sqrt{\frac{k}{N}}$) speed up by the quantum algorithm!
Thus, both of above examples show an improvement over the classical algorithms.

4. Conclusion

In this paper, it has been shown that for every binary function ω with domain size N, we can obtain the full description of the function with high probability while querying the oracle only $N/2 + \sqrt{N}$ times which makes a lot progress comparing to a classical computer that always requires N queries to determine the secret unction with the same success probability. Moreover, we are able to increase this probability by increasing the threshold, all within the bound $N/2 + O(\sqrt{N})$ queries. Last but not the least, the paper brings a quantum approximate interrogation algorithm that improves classical performance.
The value of research papers about the black-box complexity is reflected in many aspects. As discussed in the course, the black-box complexity is a useful tool for understanding black-box algorithmic approaches to solve a problem and it enlightens different progress in many ways:
- Contribute to efficient algorithms, such as Simon's black-box algorithm preceded Shor's algorithms that are black-box algorithms with concrete implementations of black-boxes.

- It rules out a wide class of approaches to solve NP-hard problems.

However, it is worth pointing out that generally the model of quantum computation does not guarantee any significant speed up of the existing classical algorithms. So it is necessary for us to investigate for each specific type of problem and to find out whether there is a possible improvement by using quantum algorithms or not and how much improvement can we obtain.

In addition, aiming at currently existing problems, we could recommend some future research relevant to this paper. One is that previous research about the lower bounds on parity (with bounded error) and OR (with no allowed error) for black-boxes have already shown that any quantum algorithm must use at least N/2 calls to obtain ω with bounded error. The full N queries are necessary to determine the string without error. So it is natural to explore whether the $O(\sqrt{N})$-term in the query complexity is necessary or not and is it possible to be reduced to the order of O(logN).

Since through the whole paper we are concentrating on the problems with respect to oracles with random structures(black-box model), another valuable future research on quantum computational complexity, for instance, would be investigating similar problems where oracles are structured or some properties of them have already been revealed(white-box model). This might bring us results that widen the gap between classical and quantum computation even further than what we have already achieved in this paper.

Reference:

[1] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. In Proceedings of the 39th Annual Symposium on Foundations of Computer Science (FOCS'98). IEEE, 1998. Also as preprint on the quant-ph archive, no. 9802049.

[2] E. Bernstein and U. Vazirani. Quantum complexity theory. SIAM Journal on Computing, 26(5):1411–1473, 1997.

[3] R. Cleve, A. Ekert, C. Macchiavello, and M. Mosca. Quantum algorithms revisited. Proceedings of the Royal Society of London A, 454:339–354, 1998. Also as preprint on the quant-ph archive, no. 9708016.

[4] E. Farhi, J. Goldston, S. Gutmann, and M. Sipser. A limit on the speed of quantum computation in determining parity, 1998. Preprint on the quantph archive, no. 9802045.

[5] L. Grover. Quantum computers can search arbitrarily large databases by a single query. Physical Review Letters, 79(23):4709–4712, Dec. 1997. Also as preprint on the quant-ph archive, no. 9706005.

[6] A. Nayak and F. Wu. On the quantum black-box complexity of approximating the mean and the median, 1998. Preprint on the quant-ph archive, no. 9804066.

[7] B. Terhal and J. Smolin. Single quantum querying of a database. Physical Review A, 58(3):1822–1826, Sept. 1998. Also as preprint on the quant-ph archive, no. 9705041.