Examination on Quick Quantum Oracle Interrogation

Victor Fengtu Lei
PHYS 467 Final Paper Review
Professor: Dr Michele Mosca

Apr 2024

1 Abstract

In this report, we are going to investigate a study done by Wim van Dam, on the topic of quantum oracle interrogation. An oracle of domains size N refers to a black box that provides information about an initially unknown binary string of size N. The overall objective is to investigate if a quantum algorithm could outperform its classical counterpart to retrieve the N-bit binary string under a small probability of error.

The first part of this study (section 4) shows that classically, the algorithm would need at least N queries on the oracle in order to obtain a 95% accuracy, whereas a quantum algorithm could achieve the same in $N/2+\sqrt{N}$ queries. Second part of the study (section 5) elaborates on a scenario where the constraint on error probability is relaxed to 80% accuracy, then we can find a quantum procedure that achieves using $N/10$ queries while the best classical procedure would need $6N/10$ queries under the same scenario[1].

2 Introduction

Before Wim van Dam's study on the quantum oracle interrogation there had already been proofs on the lower bounds of quantum algorithms' performance (required amount of queries) on the oracle in certain scenarios. For example if we wish to obtain the exact calculation of bitwise OR, we still require N queries on the oracle, same as in classical procedurescite[2]. Wim's study on the other hand, focuses on the upperbound of required number of queries in general to compute any function over the N bits (with small error probability). This will give us an idea of how much a quantum computer could potentially outperform the classical one when tackling any function on the domain of $\{0,1\}^N$ with a small error probability. Afterward, the study shows when classical procedures generally need N queries, a quantum computer can perform the same task with as minimal as $N/2+\sqrt{N}$ calls[1].

3 Preliminaries

Given a binary-valued function ω of domain size N. we can describe the oracle by an N-bit string $\vec{\omega} = \omega_1\omega_2...\omega_N \in \{0,1\}^N$. Now suppose we have a quantum computer and a classical computer, they both want to obtain the whole string $\vec{\omega}$ with at least 95% accuracy with as few oracle calls (which we treat as a black-box operation) to ω as possible. In this setting, we only consider the complexity of oracle calls, time or space factor will not considered. As premiliminaries, we will introduce the following tools:

One-Call Phase Kickback Trick

for some function $f \in 0,1$, we will use the following procedure the introduce a phase $(-1)^f$ to an arbitrary state $|\psi\rangle$:

$$|\psi\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} \rightarrow |\psi\rangle\frac{|0\oplus f\rangle-|1\oplus f\rangle}{\sqrt{2}} = \begin{cases} |\psi\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{if } f=0 \\ -|\psi\rangle\frac{|0\rangle-|1\rangle}{\sqrt{2}} & \text{if } f=1 \end{cases} \tag{1}$$

Inner Product Modulo 2

for binary strings $\vec{x}, \vec{y}$

$$(\vec{x},\vec{y}) = \bigoplus_{i=1}^{N}(x_i * y_i) \tag{2}$$

Hadamard Transform

$$\forall \vec{y} \in \{0,1\}^N H^{\otimes N}|\vec{y}\rangle = \frac{1}{\sqrt{2^N}}\sum_{x\in\{0,1\}^N}(-1)^{(\vec{x},\vec{y})}|\vec{x}\rangle \tag{3}$$

Since H is its own inverse, apply H again to obtain the original $\vec{y}$:

$$H^{\otimes N}(\frac{1}{\sqrt{2^N}}\sum_{x\in\{0,1\}^N}(-1)^{(\vec{x},\vec{y})}\left|\vec{x}\right\rangle) = \left|\vec{y}\right\rangle \tag{4}$$

Hamming Weight
$\forall\vec{x}\in\{0,1\}^N$, the Hamming weight of $\vec{x}$ represents the number of 1's in the string. represented by $||\vec{x}||$
bringing this concept of hamming weight together with equation (2), for since for any $x_i = 0$ the whole term of $(x_i * y_i = 0)$, and $x_i = 1$ implies $(x_i * y_i = y_i)$. Implies that equation (2) is essentially the parity of a subset (using $\vec{x}$ as characterstic vector) of $\vec{y}$.

4 Exact Interrogation Algorithm

4.1 outline

Now consider $\vec{\omega}$ for equation (3), we have shown that the inner product $(\vec{x}*\vec{\omega})$ is actually the parity of a subset of $\vec{y}$. This implies to calculate the inner product, instead of querying all entries of $\vec{\omega}$, we only need the entries of non-zero x_i, i.e, $\omega_i, s.t \quad x_i \neq 0$, this implies the hamming weight of $\vec{x}$ is essentially the querying complexity we are trying to obtain. Hence for equation (3), to reduce the querying complexity, instead of considering all possible $\vec{x}\in\{0,1\}^N$, we can instead consider only $\vec{x}, \text{with} ||\vec{x}|| \leq k$ for some relatively small positive interger k. For each such $\vec{x}$ and an arbitrary bit $b\in\{0,1\}$, we obtain the below procedure:

$$\left|\vec{x}\right\rangle\left|b\right\rangle \xrightarrow{||\vec{x}||\text{oracle calls}} \left|\vec{x}\right\rangle\left|b\oplus(\vec{x},\vec{\omega})\right\rangle \tag{5}$$

Further, we can define this operation as A_k, that operates like following:

$$A_k\left|\vec{x}\right\rangle\left|b\right\rangle = \begin{cases}\left|\vec{x}\right\rangle\left|b\oplus(\vec{x},\vec{\omega})\right\rangle & \text{if } ||\vec{x}|| \leq k \\ \left|\vec{x}\right\rangle\left|b\right\rangle & \text{if}||\vec{x}|| > k\end{cases} \tag{6}$$

With these consequences, we can now formulate our quantum algorithm for obtaining the N-bit binary string.

4.2 Quantum Algorithm

- **Initialization** Prepare a register of $N+1$ qubits, where the first N qubits consists of an equally weighted superposition on $\{0,1\}^N$ strings that contains hamming weight $\leq k$, we will late refer this as $|\psi_k\rangle$. Prepare the last qubit in the state $\frac{|0\rangle-|1\rangle}{\sqrt{2}}$.
- **Oracle Calls** Set $k = N/2 + \sqrt{N}$. Perform the A_k operation as defined in equation (6)k times.
- **Hadamard Transformation** Apply N Hadamard transform to the first N qubits.
- **Measurement** now we measure the first N qubits in the computational basis, will yield the algorithm's guess for the N bit binary string $\vec{\omega}$.

4.3 Analysis of the Quantum Algorithm

In this subsection we will investigate if the algorithm gives us what we want. We have also made these assumption during our algorithm: letting $k = N/2 + \sqrt{N}$. We will justify these assumptions will not cause the result to go beyong our desired error probability.
1. After the initialization stage, we obtain $|\psi_k\rangle \frac{|0\rangle-|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{M_k}}(\sum_{\vec{x},||\vec{x}||\leq k} |\vec{x}\rangle)\frac{|0\rangle-|1\rangle}{\sqrt{2}}$ where M_k is the appropriate normalization factor, $M_k = \sum_{i=0}^{k}\binom{N}{i}$.

2. After applying A_k to the above state on the first N qubits, we obtain:

$$A_k |\psi_k\rangle \frac{|0\rangle-|1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{M_k}}(\sum_{\vec{x},||\vec{x}||\leq k} (-1)^{(\vec{x},\vec{\omega})} |\vec{x}\rangle)\frac{|0\rangle-|1\rangle}{\sqrt{2}} = |\psi'_k\rangle \frac{|0\rangle-|1\rangle}{\sqrt{2}} \tag{7}$$

Notice that here, the state looks exactly like applying N-qubit Hadamard transformation to the state $|\vec{\omega}\rangle$ as in equation (3). Hence we could restore our guessed $|\vec{\omega}\rangle$ by applying another N qubit Hadamard transformation as in equation (4). Before applying, we know the only factor that affect our guessed $\vec{\omega}$ to differ from the actual one is that in the phase $(-1)^{(\vec{x},\vec{\omega})}$, if we set $k = N$, then it is equivalent as querying all N bits from the oracle. Hence obtaining the exact binary string.

$$|\psi'_N\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{M_N}}\Big(\sum_{\vec{x}\in\{0,1\}^N} (-1)^{(\vec{x},\vec{\omega})} |\vec{x}\rangle\Big)\frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{8}$$

While letting $k = 0$ means that we are not querying any information from the oracle as any query returns a result of 0. Hence we arrive at a conclusion: the fidelity of the algorithm depends on our choice of $k \in [0, N]$, where as $k \to N$, the fidelity increases.

3. Finally we assess the correctness of our output. We know our guessed string and the actual string are obtained by

$$|\vec{\omega'}\rangle = H^{\otimes N} |\psi'_k\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{9}$$

$$|\vec{\omega}\rangle = H^{\otimes N} |\psi'_N\rangle \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{10}$$

Because the Hadamard transform is its own adjoint and its own inverse, we have the probability of correctness to be:

$$\begin{aligned} P &= |\,\langle\vec{\omega'}|\vec{\omega}\rangle\,|^2 && (11)\\ &= |\,\langle\psi'_k|\psi'_N\rangle\,|^2 \qquad (9) && (12) \end{aligned}$$

to compute this value, by the construction of A_k in equation (6), we see that for all registers with $||\vec{x}|| \leq k$, the amplitudes and signs between $|\psi'_k\rangle, |\psi'_N\rangle$ are the same, while for all registers with $||\vec{x}|| > k$, the amplitudes of $|\psi'_k\rangle$ are zero. Hence the correctness probability is simply:

$$\begin{aligned} P &= \frac{M_k}{2^N} && (13)\\ &= \frac{1}{2^N}\sum_{i=0}^{k}\binom{N}{i} && (14) \end{aligned}$$

This implies that by picking our $k = N/2 + \sqrt{N}$, the correctness probability is indeed above 95%. justifying our assumption in the algorithm.

4.4 Classical Comparison

The correctness of guessing $\vec{\omega}$ for a classical computer can be straightforwardly formulated as:

$$P_C \leq \frac{1}{2^{N-k}} \tag{15}$$

since after querying the oracle k times, the classical computer still doesn't learn anything about $N - k$ bits of $\vec{\omega}$. Implying thatany classical procedure would need to query all N bits of $\vec{\omega}$ which requires N queries on the oracle.

5 Approximate Interrogation Algorithm

The approximate interrogation refers to the objective to know only a certain fraction of the N unknown bits instead of all N bits of the string. We will investigate when fixing the oracle query number to be k, what will be the maximum expected number of correct bits (we will call this number c), given that $\vec{\omega}$ is totally random.

5.1 Classical Approximate Interrogation

After k queries, a classical computer will learn k bits with certainty, while randomly guessing the rest $N - k$ bits, hence we could say the number of correct bits after this procedure is:

$$c_k^{clas} = \frac{N + k}{2} \tag{16}$$

5.2 Quantum Approximate Interrogation

The algorithm for Approximate Interrogation will be the same as introduced previously, only with different initial state

$$|\psi_k^\alpha\rangle = \sum_{j=0}^{k} \frac{\alpha_j}{\sqrt{\binom{N}{j}}} \sum_{\vec{x} \in \{0,1\}^N}^{||\vec{x}||=j} |\vec{x}\rangle \tag{17}$$

with normalization $\sum_j \alpha_j^2 = 1$. After the algorithm, we find the same amplitudes for $||\vec{x}|| > k$, which is 0, while for $||\vec{x}|| \leq k$, the amplitudes depend on α_j. Hence we obtain the following number of correct bits:

$$c_k^{quant} = \frac{N}{2} + \sum_{j=0}^{k-1} \alpha_j \alpha_{j+1} \sqrt{j+1}\sqrt{N-j} \tag{18}$$

Single Quantum Query

Let us consider the simplest case, and observe what best result we could get with the quantum algorithm, when we are only granted one query on the oracle, this value is optimized by choosing $\alpha_0 = \alpha_1 = 1/\sqrt{2}$, by equation (18), we obtain:

$$c_1^{quant} = \frac{N + \sqrt{N}}{2} \tag{19}$$

Compared with classical equation (16), the classical procedure will require $\sqrt{N}$ queries on the oracle to achieve the same number of correct bits.

Multiple Quantum Queries

For large N, i.e $\sqrt{N} << N$, and when $0 \leq k/N \leq 1/2$, we could define α_j as:

$$\alpha_j = \begin{cases} 0 & \text{if} \quad 0 \leq j \leq k - \sqrt{k} \\ 1/\sqrt[4]{k} & \text{if} \quad k - \sqrt{k} \leq j \leq k \end{cases} \tag{20}$$

Substitute this into equation (18) we get that:

$$c_k^{quant} = N/2 + 1/\sqrt{k} \sum_{j=k-\sqrt{k}}^{k-1} \sqrt{j+1}\sqrt{N-j} \approx N/2 + \sqrt{k(N-k)} \tag{21}$$

6 Conclusion

As the result of our investigation, we have found that although quantum procedure may not offer a significant speedup in terms of oracle querying for every problem, there are certain cases we could apply a quantum approach to provide a significant performance boost.

In the first part of our study, section 4, we have proven that given an arbitrary binary function ω with domain size N, we are able to obtain a full description of the function with over 95% accuracy in $N/2 + \sqrt{N}$ oracle queries by equation (14). While the classical procedure would need all N queries.

In the second part of our study, section 5, we fixed the number of queries made to the oracle, and investigated what is the maximum number of correct bits. We applied the same quantum procedure with an adjustment to the initial preparation of the state, and found out in equation equation (18), that the quantum procedure would be able to obtain $\frac{N+\sqrt{N}}{2}$ correct bits while it takes at least $\sqrt{N}$ queries for the classical approach to achieve the same number. While shown in equation (21), that the quantum procedure still holds an advantage in the objective value with multiple queries on the oracle compared to the classical procedure untill all N queries are used, and both procedure returns all N bits exactly.

Throughout the investigation, we always assume the function ω to be completely random. The author Wim Van Dam suggests under a ”white-box” model (structured oracle), with existing knowledge of the model, there are possibilities that certain quantum procedure could be even more advantageous over the classical ones than what we have discussed for the black-box model. [1] [2]

The potential application of this algorithm could be furthered to other oracle related problems such as symmetric oracle problems[3], vector interpolation [4], and matrix-vector products [5]

References

[1] W. van Dam, “Quantum oracle interrogation: getting all information for almost half the price,” in *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*. IEEE Comput. Soc, 1998, p. 362–367. [Online]. Available: http://dx.doi.org/10.1109/SFCS.1998.743486

[2] 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.

[3] D. Copeland and J. Pommersheim, “Quantum query complexity of symmetric oracle problems,” *CoRR*, vol. abs/1812.09428, 2018. [Online]. Available: http://arxiv.org/abs/1812.09428

[4] S. Decoppet, “Optimal quantum algorithm for vector interpolation,” 2022. [Online]. Available: https://arxiv.org/abs/2212.03939

[5] A. M. Childs, S.-H. Hung, and T. Li, “Quantum query complexity with matrix-vector products.” Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. [Online]. Available: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2021.55