Oracles with Costs

Shelby Kimmel[*], Cedric Lin[†], and Han-Hsuan Lin[‡]

1 Introduction

The standard oracle model is a powerful paradigm for understanding quantum computers. Tools such as the adversary semidefinite program [10, 11], learning graphs [5, 6], and the polynomial method [4] allow us to accurately characterize the quantum query complexity of many problems of interest.

However, the oracle model does not capture the full power or challenges of quantum computing. For example, problems such as k-SAT do not fit easily into the oracle model. Additionally, while the query complexity of the hidden sub-group problem is known to be polynomial [9], for some non-abelian groups there is no efficient algorithm.

In this paper, we describe a variation of the oracle model. We have access to two oracles, rather than a single oracle[1], but one oracle is more expensive to use. In the standard oracle model, the objective is to evaluate a function using the oracle as few times as possible. In this new model, which we call the Oracle Model with Costs, the goal is to evaluate a function in the cheapest way possible.

To motivate this model, we consider the problem of k-SAT on n bits. We would like to find an assignment $x \in \{0,1\}^n$ such that all clauses are satisfied. Consider an algorithm for k-SAT that runs two types of tests on a possible assignment x. The two tests are:

1. Check whether all clauses are satisfied.
2. Check whether some subset of the clauses are satisfied.

These two tests can be implemented as unitaries $\mathcal{O}_1, \mathcal{O}_2$ that act on the Hilbert space $\mathbb{C}^{2^n} \otimes \mathbb{C}^2 \otimes \mathbb{C}^d$ (for some $d \in \mathbb{Z}^+$) in the following way:

$$\mathcal{O}_i |x\rangle|y\rangle|0\rangle = |x\rangle|y \oplus f_i(x)\rangle|0\rangle. \tag{1}$$

Here $f_i(x) = 1$ if assignment x passes Test i and $f_i(x) = 0$ otherwise. Because Test 1 involves checking all of the clauses, rather than some subset of the clauses, $\mathcal{O}_1$ will potentially take a longer time to apply than $\mathcal{O}_2$.

This situation can easily be recast into an oracle problem, which we call Searching with Two Oracles (STO). In STO, one is given two oracles, $\mathcal{O}_1$ and $\mathcal{O}_2$. $\mathcal{O}_1$ identifies the marked item, if it exists, but is costly to use. $\mathcal{O}_2$ is inexpensive to use, but provides less information: it identifies some

[*] Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland, shelbyk@umd.edu. (Part of this work completed while at MIT.)

[†] Center for Theoretical Physics, Massachusetts Institute of Technology, cedricl@mit.edu.

[‡] Center for Theoretical Physics, Massachusetts Institute of Technology, hanmas@mit.edu.

[1] The model can easily be extended to more than two oracles, but for simplicity, we limit ourselves to two.

subset of items that will include the marked item, if it exists. An algorithm for STO that finds the marked item while incurring the smallest cost can be applied to the problem of k-SAT, potentially improving on naive approaches that rely only on $\mathcal{O}_1$.

Normally k-SAT is not thought of as an oracle problem, because the clauses contain more information than can easily be incorporated into a single oracle. However, with multiple oracles, the information in the clauses can be incorporated into different oracles. Introducing costs adds further subtlety. We develop some tools for understanding the Oracle Model with Costs. We hope this model will provide new insight into problems previously thought beyond the tools of query algorithms.

This work falls broadly into the class of oracle problems with advice, in which access to a single oracle is supplemented with some extra information that can come in the form of another oracle or classical information, e.g. [12, 13]. Previous works [3, 12] have considered multiple oracles, but not with costs. Furthermore, the additional advice oracles considered in these works tend to be somewhat unnatural, and are tailored to the specific problems considered. The addition of costs is related to work by Ambainis, in which he considered a single oracle that has different costs for querying different items [2].

2 Results

We use the problem of STO as a test-bed for tools and ideas that can hopefully be applied to more complex problems. More formally, we give the definition of STO:

Definition 1 (Search with Two Oracles (STO))**.** *Let N and M be known positive integers and let $S \subseteq \{0, \cdots, N-1\}$ be an unknown set. There might or might not exist a special item i^*. If i^* exists, then one is promised that $i^* \in S$ and $|S| = M$. If i^* doesn't exist, the size of S is arbitrary. Let $\mathcal{O}^*$ and $\mathcal{O}^S$ be two unitaries acting on the standard basis states $|i\rangle$ for $i \in \{0, \cdots, N-1\}$ as follows:*

$$\mathcal{O}^*|i\rangle = \begin{cases} -|i\rangle & \textit{if } i = i^* \\ |i\rangle & \textit{if } i \neq i^* \textit{ or } i^* \textit{ doesn't exist.} \end{cases} \qquad \mathcal{O}^S|i\rangle = \begin{cases} -|i\rangle & \textit{if } i \in S \\ |i\rangle & \textit{if } i \notin S. \end{cases} \tag{2}$$

Given oracle access to $\mathcal{O}^*$ and $\mathcal{O}^S$, (where for simplicity the oracles can not be used in superposition), one must determine with bounded error the value of i^*, or determine that i^* does not exist.

Let $\mathcal{O}^*$ have cost c^* and $\mathcal{O}^S$ have cost c^S. Among algorithms that solve STO, we want to find the algorithm that has the smallest cost, where if q^* is the number of times $\mathcal{O}^*$ is used, and q^S is the number of times $\mathcal{O}^S$ is used, the cost of an algorithm is

$$c^*q^* + c^Sq^S. \tag{3}$$

Two classical deterministic algorithms for STO are

1. Classical Algorithm 1 (CA1): Query each element with $\mathcal{O}^*$, to test whether the element is marked. This has cost c^*N.
2. Classical Algorithm 2 (CA2): Query each element with $\mathcal{O}^S$. This will identify all elements of S. Then test the elements of S with $\mathcal{O}^*$ to determine whether any of the elements of S are marked. This has cost $c^SN + c^*M$.

It can be shown that the optimal deterministic algorithm is either CA1 or CA2; thus for a deterministic classical computer the cost is exactly

$$\min\left(c^S N + c^* M, c^* N\right). \tag{4}$$

We can quantize CA1 and CA2 using amplitude amplification [7]. The quantized versions of CA1 and CA2 have costs $O\left(c^*\sqrt{N}\right)$ and $O\left(c^S\sqrt{N} + c^*\sqrt{M}\right)$ respectively, giving a quantum algorithm with cost

$$O\left(\min\left(c^S\sqrt{N} + c^*\sqrt{M}, c^*\sqrt{N}\right)\right). \tag{5}$$

In the classical case, the optimal algorithm is always CA1 or CA2. One might guess that in the quantum case, the optimal algorithm is always one of the two quantized versions of these classical algorithms. However, we construct a quantum algorithm that achieves lower query cost than the simple quantized versions of these classical algorithms. The quantum problem thus has a richer structure than that of the corresponding classical problem.

Additionally, we prove lower bounds on the cost of STO. By integrating multiple oracles into the framework of the adversary method [1, 14] we show that the bound of Eq. 5 is asymptotically tight, i.e. the cost of STO is

$$\Theta\left(\min\left(c^S\sqrt{N} + c^*\sqrt{M}, c^*\sqrt{N}\right)\right). \tag{6}$$

We were furthermore curious whether our algorithm is exactly optimal. By exactly optimal, we mean that the upper and lower bounds on the cost are equal, not just asymptotically tight up to a multiplicative constant. Grover's algorithm is known to be exactly optimal [15]. Our algorithm is a 2-level nesting of Grover's algorithm, i.e. we perform amplitude amplification on an algorithm that is itself a Grover search algorithm. If we can prove exact optimality of our algorithm, it would provide strong evidence that extensions of Grover's algorithm, like amplitude amplification, are also exactly optimal.

The adversarial lower bounds we prove are insufficient to show our algorithm is exactly optimal (even when we include ideas from exact Grover lower bounds [8]). However, we do give some evidence that our algorithm is indeed exactly optimal. In particular, we consider algorithms in which we cannot perform arbitrary operations, but instead are limited to the unitaries $\mathcal{O}^*$, $\mathcal{O}^S$, and $\mathcal{G}$, where

$$\mathcal{G} = 2|\psi\rangle\langle\psi| - \mathbb{I} \tag{7}$$

for $|\psi\rangle = \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle$. When limited to these three unitaries (and given no ancilla qubits), we prove that our algorithm is exactly optimal in terms of cost (up to a multiplicative factor that is asymptotically 1). Our proof uses ideas from both adversary methods (we create a non-standard progress function) and the geometric picture of Grover's algorithm. It is an open question whether these techniques could be combined with more standard lower-bounding methods to extend our result to allow arbitrary operations.

References

[1] Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proc. 32nd ACM STOC*, pages 636–643. ACM, 2000.

[2] Andris Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47(3):786–807, 2010.

[3] Andris Ambainis, Ansis Rosmanis, and Dominique Unruh. Quantum attacks on classical proof systems - the hardness of quantum rewinding. *arXiv preprint arXiv:1404.6898*, 2014.

[4] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001.

[5] Aleksandrs Belovs. Learning-graph-based quantum algorithm for k-distinctness. In *Proc. IEEE 53rd FOCS*, pages 207–216. IEEE Computer Society, 2012.

[6] Aleksandrs Belovs and Ansis Rosmanis. On the power of non-adaptive learning graphs. *Computational Complexity*, 23(2):323–354, 2014.

[7] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055*, 2000.

[8] Catalin Dohotaru and Peter Høyer. Exact quantum lower bound for grover's problem. *arXiv preprint arXiv:0810.3647*, 2008.

[9] Mark Ettinger, Peter Høyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43–48, 2004.

[10] Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proc. 39th ACM STOC*, pages 526–535, 2007.

[11] Troy Lee, Rajat Mittal, Ben W Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proc. 52nd IEEE FOCS*, pages 344–353. IEEE, 2011.

[12] Ashley Montanaro. Quantum search with advice. In *Theory of Quantum Computation, Communication, and Cryptography*, pages 77–93. Springer, 2011.

[13] Aran Nayebi, Scott Aaronson, Aleksandrs Belovs, and Luca Trevisan. Quantum lower bound for inverting a permutation with advice. *arXiv preprint arXiv:1408.3193*, 2014.

[14] Ben W Reichardt. Reflections for quantum query algorithms. In *Proc. 22nd ACM-SIAM SODA*, pages 560–569. SIAM, 2011.

[15] Christof Zalka. Grover's quantum searching algorithm is optimal. *Physical Review A*, 60(4):2746, 1999.