Upper bounds on quantum query complexity inspired by the Elitzur-Vaidman bomb tester

Cedric Yen-Yu Lin*[1] and Han-Hsuan Lin†[1]

[1] *Center for Theoretical Physics, Massachusetts Institute of Technology, Cambridge, MA, USA*

This work is available online at arXiv.org as arXiv:1410.0932 [quant-ph]

1 Introduction

Quantum query complexity is an important method of understanding the power of quantum computers. In this model we are given a black-box containing a boolean string $x = x_1 \cdots x_N$, and we would like to calculate some function $f(x)$ with as few accesses to the black-box as possible. It is often easier to give bounds on the query complexity than to the time complexity of a problem, and insights from the former often prove useful in understanding the power and limitations of quantum computers. One famous example is Grover's algorithm for unstructured search [1]; by casting this problem into the query model it was shown that $\Theta(\sqrt{N})$ queries was required [2], proving that Grover's algorithm is optimal.

Several methods have been proposed to bound the quantum query complexity. Upper bounds are almost always proven by finding better query algorithms. Some general methods of constructing quantum algorithms have been proposed, such as quantum walks [3, 4, 5, 6] and learning graphs [7]. For lower bounds, the main methods are the polynomial method [8] and adversary method [9]. In particular, adversary lower bounds have been shown to be tight [10, 11, 12], but calculating such a tight bound seems difficult in general.

To improve our understanding of quantum query complexity, we introduce a new oracle model, which we call the *bomb oracle*. This model is inspired by the concept of *interaction free measurements*, illustrated vividly by the Elitzur-Vaidman bomb testing problem [13], in which a property of a system can be measured without disturbing the system significantly. Like the quantum oracle model, in the bomb oracle model we want to evaluate a function $f(x)$ on a black-box boolean string $x = x_1 \cdots x_N$ while querying the oracle as few times as possible. In this model, however, the bomb oracle is a controlled quantum oracle with the extra requirement that the algorithm fails if the controlled query returns a 1. This seemingly impossible task can be tackled in a fashion similar to the Elitzur-Vaidman bomb tester [14].

Our main result is that the bomb query complexity, $B(f)$, is characterized by the square of the quantum query complexity $Q(f)$: $B(f) = \Theta(Q(f)^2)$.

This characterization allows us to give *nonconstructive* upper bounds to the quantum query complexity for some problems. It is sometimes easy to design a bomb query algorithm by adapting a classical algorithm. By our main result, this gives an upper bound on the quantum query complexity.

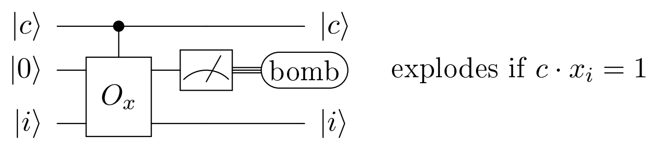
*cedricl@mit.edu

†hanmas@mit.edu

We provide a general method for doing so, and inspired by this method we give a corresponding explicit quantum algorithm. Using this method, we were able to give an $O(n^{3/2})$ algorithm for the single-source shortest paths (SSSP) problem in an unweighted graph with n vertices, beating the best-known $O(n^{3/2}\sqrt{\log n})$ algorithm [15]. A more striking example is our $O(n^{7/4})$ algorithm for maximum bipartite matching; in this case the best-known upper bound was the trivial $O(n^2)$.

2 Model

We define the *bomb query model* as follows: we want to compute a function $f(x)$ using a quantum circuit, where access to the hidden query string x is not provided through the usual quantum oracle O_x, but rather through a bomb oracle, shown in the following circuit:



(1)

In this circuit O_x is the traditional quantum oracle: $O_x|0, i\rangle = |x_i, i\rangle$. There are however three differences between the bomb oracle and the usual quantum oracle O_x:

- We allow an extra control bit c to control the oracle O_x. (This modification on its own would not change the query complexity.)
- The input to the record register *must* be $|0\rangle$ before the application of controlled-O_x; after the application of controlled-O_x it will contain $|c \cdot x_i\rangle$.
- After the application of controlled-O_x, the record register is immediately measured. If a 1 is measured (corresponding to $c \cdot x_i = 1$), the algorithm fails. We say *the bomb has exploded.*

We define the *bomb query complexity* $B_\epsilon(f)$ to be the minimum number of times the bomb oracle shown above needs to be applied in an algorithm such that the following hold for all input string x:

- The bomb explodes with probability at most ϵ.
- The probability that the bomb outputs the wrong answer is bounded by a constant (say 0.01).

3 Main Result

Let $Q(f)$ be the bounded-error quantum complexity. Our main result is the following:

$$B_\epsilon(f) = \Theta\left(\frac{Q(f)^2}{\epsilon}\right). \tag{2}$$

We prove the $B_\epsilon(f) = O(Q(f)^2/\epsilon)$ upper bound by mimicking the solution of the Elitzur-Vaidman problem [14]: we simulate each quantum query with a gadget using $O(Q(f)/\epsilon)$ bomb queries. By utilizing the quantum Zeno effect, the gadget simulates a quantum query with $O(\epsilon/Q(f))$ error and probability of explosion. This allows us to simulate a quantum algorithm with $O(Q(f)^2/\epsilon)$ bomb queries while keeping the probability of explosion and the error constant-sized.

We prove the $B_\epsilon(f) = \Omega(Q(f)^2/\epsilon)$ lower bound through a novel adaption of the adversary method to bomb query complexity. We show that the bomb query complexity is $\Omega(\mathrm{Adv}^{\pm}(f))^2/\epsilon$, where $\mathrm{Adv}^{\pm}(f)$ is the adversary bound with general weights [16]. Since the general adversary method is tight for quantum query complexity, i.e. $\mathrm{Adv}^{\pm}(f) = \Theta(Q(f))$ [10, 11, 12], this shows that $B_\epsilon(f) = \Omega(Q(f)^2/\epsilon)$.

4 Applications

Inspired by our characterization of bomb query complexity, we have the following result (stated informally):

> Suppose there is a classical algorithm that computes $f(x)$ in T queries, and the algorithm guesses the result of each query (0 or 1), making no more than an expected G mistakes for all x. Then there is an explicit quantum algorithm using $O(\sqrt{TG})$ queries.

This result is inspired by the easy construction of bomb query algorithms for certain functions. Take, for example, the OR function: decide whether the string x is the all-zero string 0^N or not. A simple classical algorithm would be to simply check each bit of x one-by-one until we find a 1; this takes at most $T = N$ queries. At each query, the algorithm could guess that the query result is 1; since the algorithm ends when a 1 is found, there is at most $G = 1$ wrong guess. Therefore $Q(OR) = O(\sqrt{TG}) = O(\sqrt{N})$. We have thus proved the existence of Grover's algorithm.

We were able to make this upper bound *constructive*, by constructing an explicit quantum algorithm tht makes $O(\sqrt{TG})$ queries. This algorithm is very similar to Kothari's algorithm for oracle identification [17]. Roughly speaking, the quantum algorithm takes the T-query classical algorithm and uses quantum search to squentially find the G mistakes.

It turns out that this approach can be used to improve the upper bounds of several graph problems in the adjacency matrix model. For example consider the following problem: given an unweighted directed graph G with n vertices, find all shortest paths from a fixed vertex $v \in G$ to all other vertices $w \in G$ (single source shortest paths). By analyzing the classical breadth-first search algorithm, we obtain $Q(f) = O(n^{3/2})$ (beating the best known upper bound of $O(n^{3/2}\sqrt{\log n})$ [15]). Another example is finding a maximum matching (a maximum set of edges that do not share vertices) in a bipartite graph with n vertices; by analyzing the classical Hopcroft-Karp algorithm [18] we see that this takes no more than $O(n^{7/4})$ quantum queries to the adjacency matrix. (The best known upper bound is the trivial $O(n^2)$, although the time complexity of this problem was studied in [19, 20].)

Finally, we hope that the bomb query complexity model can help us understand the relationship between the classical randomized query complexity, $R(f)$, and the quantum query complexity $Q(f)$. It is known [8] that for all total functions f, $R(f) = O(Q(f)^6)$; however, there is a long-standing conjecture that actually $R(f) = O(Q(f)^2)$. In light of our results, this conjecture is equivalent to the conjecture that $R(f) = O(\epsilon B_\epsilon(f))$. Further study on the relationship between bomb and classical randomized complexity may therefore shed light on the limitations of quantum computation.

References

[1] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing (STOC)*. May, 1996. arXiv:quant-ph/9605043.

[2] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, "Strengths and weaknesses of quantum computing," *SIAM Journal on Computing* **26** no. 5, (1997) 1510–1523, arXiv:quant-ph/9701001.

[3] A. Ambainis, "Quantum walk algorithm for element distinctness," *SIAM Journal on Computing* **37** no. 1, (2007) 210–239, arXiv:quant-ph/0311001.

[4] M. Szegedy, "Quantum speed-up of Markov chain based algorithms," in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*. 2004.

[5] F. Magniez, A. Nayak, J. Roland, and M. Santha, "Search via quantum walk," *SIAM Journal on Computing* **40** no. 1, (2011) 142–164, arXiv:quant-ph/0608026.

[6] S. Jeffery, R. Kothari, and F. Magniez, "Nested quantum walks with quantum data structures," arXiv:1210.1199 [quant-ph].

[7] A. Belovs, "Span programs for functions with constant-sized 1-certificates," arXiv:1105.4024 [quant-ph].

[8] 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)*, p. 352. 1998. arXiv:quant-ph/9802049.

[9] A. Ambainis, "Quantum lower bounds by quantum arguments," in *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pp. 636–643. 2000. arXiv:quant-ph/0002066.

[10] B. W. Reichardt, "Span programs and quantum query complexity: The general adversary bound is nearly tight for every boolean function," in *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 544–551. 2009. arXiv:0904.2759 [quant-ph].

[11] B. W. Reichardt, "Reflections for quantum query algorithms," in *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 560–569. 2011. arXiv:1005.1601 [quant-ph].

[12] T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy, "Quantum query complexity of state conversion," in *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 344–353. 2011. arXiv:1011.3020 [quant-ph].

[13] A. C. Elitzur and L. Vaidman, "Quantum mechanical interaction-free measurements," *Foundations of Physics* **23** no. 7, (July, 1993) 987–997, arXiv:hep-th/9305002.

[14] P. Kwiat, H. Weinfurter, T. Herzog, A. Zeilinger, and M. A. Kasevich, "Interaction-free measurement," *Physical Review Letters* **74** no. 24, (1995) 4763.

[15] B. Furrow, "A panoply of quantum algorithms," *Quantum Information and Computation* **8** no. 8, (September, 2008) 834–859, arXiv:quant-ph/0606127.

[16] P. Høyer, T. Lee, and R. Špalek, "Negative weights make adversaries stronger," *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC)* (2007) 526–535, arXiv:quant-ph/0611054.

[17] R. Kothari, "An optimal quantum algorithm for the oracle identification problem," in *Proceedings of the 31st International Symposium on Theoretical Aspects of Computer Science (STACS)*, E. W. Mayr and N. Portier, eds., vol. 25 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 482–493. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2014. arXiv:1311.7685 [quant-ph].

[18] J. E. Hopcroft and R. M. Karp, "An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs," *SIAM Journal on Computing* **2** no. 4, (1973) 225–231.

[19] A. Ambainis and R. Špalek, "Quantum algorithms for matching and network flows," in *Lecture Notes in Computer Science*, vol. 3884, pp. 172–183. Springer, 2006. arXiv:quant-ph/0508205.

[20] S. Dörn, "Quantum algorithms for matching problems," *Theory of Computing Systems* **45** no. 3, (October, 2009) 613–628.