On the Power of Quantum Oracles

Elham Kashefi[1,2], Adrian Kent, Vlatko Vedral[1], Konrad Banaszek[1,2]
Trusted E-Services Laboratory
HP Laboratories Bristol
HPL-2001-315 (R.1)
April 10th, 2002*

quantum computing, algorithms, query complexity, search problems

A standard quantum oracle S_f for a function $f: Z_N \rightarrow Z_N$ is defined to act on two input states and return two outputs, with inputs $|i\rangle$ and $|j\rangle$ ($i, j \in Z_N$) returning outputs $|i\rangle$ and $|j \oplus f(i)\rangle$. For general f, this is the simplest invertible quantum map that allows the evaluation of any $f(i)$ with one call. However, if f is known to be a one-one, a simpler oracle, M_f, which returns $|f(i)\rangle$ given $|i\rangle$, shares these properties.

We consider the relative strengths of these oracles. We define a simple promise problem which minimal quantum oracles can solve exponentially faster than classical oracles, via an algorithm which does not extend to standard quantum oracles. We then prove our main result: two invocations of M_f suffice to construct S_f, while $\Theta(\sqrt{N})$ invocations of S_f are required to construct M_f.

* Internal Accession Date Only Approved for External Publication?
1 Optics Section, The Blackett Laboratory, Imperial College, London SW7 2BZ, England,
2 Centre for Quantum Computation, Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, England

On the Power of Quantum Oracles

Elham Kashefi*†, Adrian Kent^S, Vlatko Vedral* and Konrad Banaszek*†

*Optics Section, The Blackett Laboratory, Imperial College, London SW7 2BZ, England

† Centre for Quantum Computation, Clarendon Laboratory, University of Oxford, Parks Road, Oxford OX1 3PU, England

^{S} Hewlett-Packard Laboratories, Filton Road, Stoke Gifford, Bristol BS34 8QZ, England

(January 25, 2002)

A standard quantum oracle S_f for a function $f: Z_N \to Z_N$ is defined to act on two input states and return two outputs, with inputs $|i\rangle$ and $|j\rangle$ ($i, j \in Z_N$) returning outputs $|i\rangle$ and $|j \oplus f(i)\rangle$. For general f, this is the simplest invertible quantum map that allows the evaluation of any $f(i)$ with one call. However, if f is known to be a one-one, a simpler oracle, M_f, which returns $|f(i)\rangle$ given $|i\rangle$, shares these properties. We consider the relative strengths of these oracles. We define a simple promise problem which minimal quantum oracles can solve exponentially faster than classical oracles, via an algorithm which does not extend to standard quantum oracles. We then prove our main result: two invocations of M_f suffice to construct S_f, while $\Theta(\sqrt{N})$ invocations of S_f are required to construct M_f.

Recent years have witnessed an explosion of interest in quantum computation, as it has become clearer that quantum algorithms are more efficient than classical for a variety of tasks. [8,11,9,3]. One important way of comparing the efficiencies is by analysing *query complexity*, which measures the number of invocations of an "oracle" — which may be a standard circuit implementing a useful sub-routine, a physical device, or a purely theoretical construct — needed to complete a task. A number of general results show the limitations and advantages of quantum computers using the query complexity models [2,12].

In this letter we compare the query complexity analysis of quantum algorithms given two different ways of representing a permutation in terms of a black box quantum oracle. We begin with a short discussion of graph isomorphism problems, which motivates the rest of the paper.

Suppose we are given two graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, represented as sets of vertices and edges in some standard notation. The graph isomorphism (GI) problem is to determine whether G_1 and G_2 are isomorphic: that is, whether there is a bijection $f : V_1 \to V_2$ such that $(f(u), f(v)) \in E_2$ if and only if $(u, v) \in E_1$. (We assume $|V_1| = |V_2|$, else the problem is trivial.) Like factorisation, GI is a problem which is not NP-complete but for which no deterministic general case polynomial time classical algorithm is known. Unlike factorisation, no polynomial time quantum algorithm for GI is currently known either.

We are interested in a restricted version (NAGI) of GI, in which it is given that G_1 and G_2 are non-automorphic: i.e., they have no non-trivial automorphisms. So far as we are aware, no polynomial time classical or quantum algorithms are known for NAGI either. The following observations suggest a possible line of attack in the quantum case.

First, for any non-automorphic graph $G = (V, E)$, we can define a unitary map O_G that takes permutations ρ of V as inputs and outputs the permuted graph $\rho(G) = (\rho(V), \rho(E))$, with some standard ordering (e.g. alphabetical) of the vertices and edges, in some standard computational basis representations. That is, writing $|V| = N$, for any $\rho \in S_N$, O_G maps $|\rho\rangle$ to $|\rho(G)\rangle$. Consider a pair (G_1, G_2) of non-automorphic graphs. Given circuits implementing O_{G_1}, O_{G_2}, we could input copies of the state $\frac{1}{\sqrt{N!}} \sum_{\rho \in S_N} |\rho\rangle$ to each circuit, and compare the outputs $|\psi_i\rangle = \sum_{\rho \in S_N} |\rho(G_i)\rangle$. Now, if the graphs are isomorphic, these outputs are equal; if not, they are orthogonal. These two cases can be distinguished with arbitrarily high confidence in polynomial time (see below), so this would solve the problem.

Unfortunately, our algorithm for NAGI requires constructing circuits for the O_{G_i}, which could be at least as hard as solving the original problem. On the other hand, it is easy to devise a circuit, S_G, which takes two inputs, $|\rho\rangle$ and a blank set of states $|0\rangle$, and outputs $|\rho\rangle$ and $|\rho(G)\rangle$. Since S_G and O_G implement apparently similar tasks, one might hope to find a way of constructing O_G from a network involving a small number of copies of S_G. Such a construction would solve NAGI. Alternatively, one might hope to prove such a construction is impossible, and so definitively close off this particular line of attack.

Thus motivated, we translate this into an abstract problem in query complexity. Our oracles here are specified quantum circuits which implement prescribed maps. Since a quantum circuit can be reversed gate by gate, this means that having access to an oracle O also implies access to an oracle O^{-1} implementing the inverse unitary map. We thus define the query complexity of an algorithm involving an oracle O_f associated to a function f to be the number of copies of O_f and/or O_f^{-1} required to implement the algorithm in a circuit that, apart from the oracles, is independent of f.

Note that in the alternative model in which oracles are taken to be black boxes supplied by a third party, one should not assume that O^{-1} can easily be constructed from O, as we know no way of efficiently reversing the operation of an unknown physical evolution.

The following oracles are defined for a general function $f : \{0,1\}^m \to \{0,1\}^n$:

- the *phase* oracle: $P_f : |x\rangle|b\rangle \to e^{2\pi i f(x)\cdot b/2^n}|x\rangle|b\rangle$.
- the *standard* oracle: $S_f : |x\rangle|b\rangle \to |x\rangle|b \oplus f(x)\rangle$.

Here x and b are strings of m and n bits respectively, $|x\rangle$ and $|b\rangle$ are the corresponding computational basis states, and $\oplus$ is addition modulo 2^n. The oracles P_f and S_f are equivalent in power: each can be constructed by a quantum circuit containing just one copy of the other.

For the rest of the paper we take $m = n$ and suppose we know f is a permutation on the set $\{0,1\}^n$. There is then a simpler invertible quantum map associated to f:

- the *minimal* oracle: $M_f : |x\rangle \to |f(x)\rangle$.

We can model NAGI, and illustrate the different behaviors of the standard and minimal oracles, by a promise problem. Suppose we are given two permutations, α and β, of Z_N, and a subset S of Z_N, and are promised that the images $\alpha(S)$ and $\beta(S)$ are either identical or disjoint. The problem is to determine which. (This problem has been considered in a different context by Buhrman et al [5].)

For simplicity we take $N = 2^n$, where n is an integer. We represent elements $x \in Z_N$ by computational basis states of n qubits in the standard way, and write $|S\rangle = \sum_{x\in S}|x\rangle$.

Figure 1 gives a quantum network with minimal oracles that identifies disjoint images with probability 1/2.

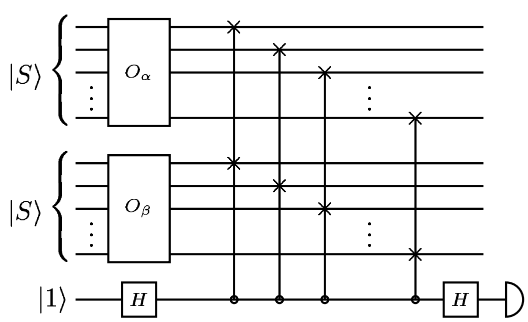

FIG. 1. A quantum circuit for the permutation promise problem.

Let $A = \{\alpha(x)|x \in S\}$ and $B = \{\beta(x)|x \in S\}$, one query to each oracles M_α and M_β creates superpositions of $\sum_{i\in A}|i\rangle$ and $\sum_{j\in B}|j\rangle$. The state of computation before applying the controlled gates is:

$$\sum_{i\in A, j\in B} |i\rangle|j\rangle \otimes \big(|0\rangle - |1\rangle\big)$$

After controlled swap gates, the state becomes:

$$\Big(\sum_{i\in A,j\in B}|i\rangle|j\rangle\Big)|0\rangle - \Big(\sum_{i\in A,j\in B}|j\rangle|i\rangle\Big)|1\rangle\,.$$

The final Hadamard gate on the ancilla qubit gives:

$$\begin{aligned}&\big(\textstyle\sum_{i\in A,j\in B}|i\rangle|j\rangle - \sum_{i\in A,j\in B}|j\rangle|i\rangle\big)|0\rangle\\ +&\big(\textstyle\sum_{i\in A,j\in B}|i\rangle|j\rangle + \sum_{i\in A,j\in B}|j\rangle|i\rangle\big)|1\rangle\end{aligned}$$

A $|0\rangle$ outcome shows unambiguously that the images are disjoint. A $|1\rangle$ outcome is generated with probability 1 if the images are identical, and with probability 1/2 if the images are disjoint. Repeating the computation K times allows one to exponentially improve the confidence of the result. If after K trials we get $|0\rangle$ at least once, we know for certain that $\alpha(S) \neq \beta(S)$. If all the outcomes were $|1\rangle$ then, on the Bayesian hypothesis that the two possibilities were initially equally likely, concluding that $\alpha(S) = \beta(S)$ has an error probability $p_K = \frac{1}{1+2^K}$. Note that p_K is independent of the problem size and decreases exponentially with the number of repetitions.

Clearly, a naive adaptation of the algorithm to standard oracles does not work. Replacing M_α and M_β by S_α and S_β, and replacing the inputs by $|S\rangle \otimes |0\rangle$, results in output states which are orthogonal if the images are disjoint, but also in general very nearly orthogonal if the images are identical. Applying a symmetric projection as above thus almost always fails to distinguish the cases.

This example suggests that minimal oracles may be rather more powerful than standard oracles. To establish a more precise version of this hypothesis, we examine how good each oracle is at simulating the other. One way round turns out to be simple. We can construct S_f from M_f and $(M_f)^{-1} = M_{f^{-1}}$ as follows:

$$S_f = (M_{f^{-1}} \otimes I) \circ A \circ (M_f \otimes I)$$

where $\circ$ represents the composition of operations (or the concatenation of networks) and the modulo N adder A is defined by $A : |a\rangle \otimes |b\rangle \to |a\rangle \otimes |a \oplus b\rangle$.

So a standard oracle can easily be simulated given a minimal oracle, using just two invocations, one of M_f and one of $(M_f)^{-1}$. However, as we now show, the converse is not true. Simulating a minimal oracle M_f requires exponentially many uses of the standard oracle S_f.

First, consider the standard oracle $S_{f^{-1}}$ which maps a basis state $|y\rangle|b\rangle$ to $|y\rangle|b\oplus f^{-1}(y)\rangle$. Since $S_{f^{-1}} : |y\rangle|0\rangle \to |y\rangle|f^{-1}(y)\rangle$, simulating it allows us to solve the search problem of identifying $|f^{-1}(y)\rangle$ from a database of N

elements. It is known that, using Grover's search algorithm, one can simulate $S_{f^{-1}}$ with $O(\sqrt{N})$ invocations of S_f [4] . In the following we explain one possible way of doing that.

Prepare the state $|y\rangle|0\rangle|0\rangle|0\rangle$, where the first three registers consist of n qubits and the last register is a single qubit. Apply Hadamard transformations on the second register to get $|\phi_1\rangle = |y\rangle \sum_{x\in Z_N} |x\rangle|0\rangle|0\rangle$. Invoking S_f on the second and third registers now gives

$$|y\rangle(\sum_{x\in Z_N} |x\rangle|f(x)\rangle)|0\rangle.$$

Using CNOT gates, compare the first and third registers and put the result in the fourth, obtaining

$$\Big(|y\rangle \sum_{x\in Z_N, x\neq f^{-1}(y)} |x\rangle|f(x)\rangle|0\rangle\Big) + \Big(|y\rangle||f^{-1}(y)\rangle|y\rangle|1\rangle\Big).$$

Now apply $(S_f)^{-1}$ on the second and third registers, obtaining

$$\Big(|y\rangle \sum_{x\in Z_N, x\neq f^{-1}(y)} |x\rangle|0\rangle|0\rangle\Big) + \Big(|y\rangle|f^{-1}(y)\rangle|0\rangle|1\rangle\Big).$$

Taken together, these operations leave the first and third registers unchanged, while their action on the second and fourth defines an oracle for the search problem. Applying Grover's algorithm [9] to this oracle, we obtain the state $|y\rangle|f^{-1}(y)\rangle$ after $O(\sqrt{N})$ invocations.

Lemma 1 The inverse oracle $S_{f^{-1}}$ can be simulated by a network that uses $\Theta(\sqrt{N})$ invocations of S_f and $(S_f)^{-1}$.

Proof The upper bound of $O(\sqrt{N})$ is implied by the Grover-based algorithm just discussed. Ambainis [1] has shown that $\Omega(\sqrt{N})$ invocations of the standard oracle S_f are required to invert a general permutation f. His elegant proof (Theorem 6 of [1]) easily extends to establish the same lower bound even if invocations of ${S_f}^{-1}$ are also allowed. QED.

Given S_f and $S_{f^{-1}}$, Bennett has shown how to simulate M_f within classical reversible computation [6] (also see below). Now we can establish our main result:

Lemma 2 The minimal oracle M_f can be simulated by a network that uses $\Theta(\sqrt{N})$ invocations of S_f and $(S_f)^{-1}$.

Proof Given S_f and $S_{f^{-1}}$, we can simulate M_f as follows:

$$M_f \otimes I = (S_{f^{-1}})^{-1} \otimes S \otimes S_f\,,$$

where the swap gate S is defined by $S : |a\rangle \otimes |b\rangle \to |b\rangle \otimes |a\rangle$. From Lemma 1, $S_{f^{-1}}$ needs $\Theta(\sqrt{N})$ invocations of S_f and $(S_f)^{-1}$. Therefore we get the upper bound of $O(\sqrt{N})$ for simulation of M_f.

However this is the optimal simulation. For suppose there were a network which simulates M_f with less than $\Omega(\sqrt{N})$ queries. The reversed network simulates $M_{f^{-1}}$. From these two, by our earlier results, we can construct a network that simulates $S_{f^{-1}}$ with less than $\Omega(\sqrt{N})$ queries, which contradicts Lemma 1. QED.

In summary, constructing a minimal oracle requires exponentially many invocations of a standard oracle. We can thus indeed definitively exclude the attack on the non-automorphic graph isomorphism problem which motivated our discussion. We have not, however, been able to exclude the possibility of directly constructing a polynomial size network defining an M_f oracle for any given $1-1$ function f, which would lead to a polynomial time solution of NAGI. Another interesting open question is the general relationship between the complexity classes BQP^{M_f} and BQP^{S_f}.

Acknowledgments. We thank Charles Bennett for helpful discussions and for drawing our attention to Refs. [6]. E. K. thanks Mike Mosca for useful discussions and Waterloo University for hospitality. This work was supported by EPSRC, the European grants EQUIP and QAIP and the QUIPROCONE grant.

[1] A. Ambainis, Quantum lower bounds by quantum arguments, *Proc. 32nd ACM Symp. Theor. Comput.*, 2000.

[2] R. Beals, H. Buhram, R. Cleve, M. Mosca, and R. Wolf, Quantum lower bounds by polynomials, *FOCS*, 352, 1998.

[3] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, Strengths and weaknesses of quantum computing, *SIAM Journal on Computing*, **26**, 1510, 1997.

[4] M. Boyer, G. Brassard, P. Hoyer and A. Tapp, Tight bounds on quantum searching, *Fortsch. Phys. – Prog. Phys*, **46**, 493, 1998.

[5] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum fingerprinting, quant-ph/0102001, 2001.

[6] C. Bennett, Logical Reversibility of Computations, *IBM Journal of Research and Development*, **17**, 525, 1973.

[7] R. Cleve, An introduction to quantum complexity theory, In C. Macchiavello, G.M. Palma, , and A. Zeilinger, editors, *Collected Papers on Quantum Computation and Quantum Information Theory*, World Scientific, 1999.

[8] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, In *Proceedings of the Royal Society of London*, **A400**, 97, 1985.

[9] L.K. Grover, A fast quantum mechanical algorithm for database search, In *Proceedings of STOC'96 – Symposium on the Theory of Computing*, 212, Philadelphia, Pennsylvania, 1996.

[10] P.W. Shor, Algorithms for quantum computation: Discrete logarithms and factoring, In *Proceedings of FOCS'94 – Symposium on Foundations of Computer Science*, 124, 1994.

[11] W. van Dam, Quantum oracle interrogation: Getting all information for almost half the price, In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS'98)*, 362, quant-ph/9805006, 1998.

[12] V. Vedral, A. Barenco, and A. Ekert, Quantum networks for elementary arithmetic operations, *Phys. Rev. A*, **54**, 147, 1996.

[13] C. Zalka, Grover's quantum searching algorithm is optimal, *Phys. Rev. A*, **60**, 2746, 1999.