3 August 2001
revised 18 February 2002

QUANTUM ALGORITHMS FOR HIGHLY STRUCTURED SEARCH PROBLEMS

Markus Hunziker[1,2] **and David A. Meyer**[1]

[1]*Project in Geometry and Physics, Department of Mathematics*
University of California/San Diego, La Jolla, CA 92093-0112
{hunziker,dmeyer}@math.ucsd.edu

[2]*Present address: Department of Mathematics*
University of Georgia, Athens, GA 30602-7403
hunziker@math.uga.edu

ABSTRACT

We consider the problem of identifying a base k string given an oracle which returns information about the number of correct components in a query, specifically, the Hamming distance between the query and the solution, modulo $r = \max\{2, 6-k\}$. Classically this problem requires $\Omega(n \log_r k)$ queries. For $k \in \{2,3,4\}$ we construct quantum algorithms requiring only a single quantum query. For $k > 4$, we show that $O(\sqrt{k})$ quantum queries suffice. In both case, the quantum algorithms are optimal.

2001 Physics and Astronomy Classification Scheme: 03.67.Lx.
2000 American Mathematical Society Subject Classification: 81P68, 68Q17.
Key Words: quantum search, query complexity.

1. Introduction

It is widely believed that the $O(n^2 \log n \log\log n)$ complexity of Shor's quantum factoring algorithm [1] is less than the complexity of the best possible classical factoring algorithm—based on the $O\big(\exp(cn^{1/3}\log^{2/3} n)\big)$ complexity of the best *known* classical algorithm [2]—and hence that quantum computing may provide efficient solutions to classically hard problems. The only *proved* separations between optimal classical and quantum solutions, however, are in terms of *query* complexity, *i.e.*, relative to oracles.

The original result of this type is the Deutsch-Jozsa algorithm [3]: Let

$$\mathcal{DJ}^n = \{f : \{0,1\}^n \to \{0,1\} \mid f(x) = \text{const. or } |f^{-1}(0)| = |f^{-1}(1)|\}.$$

Given an oracle which responds to a query $x \in \{0,1\}^n$ by evaluating $f(x)$ for some $f \in \mathcal{DJ}^n$, the improved [4,5] Deutsch-Jozsa algorithm requires only a single quantum query to determine if f is a constant function, while in the worst case, no classical algorithm can do so with fewer than $2^{n-1}+1$ queries.

Building on this result, Bernstein and Vazirani considered a parameterized family of functions [6]: Let

$$\mathcal{BV}^n = \{f_a : \{0,1\}^n \to \{0,1\} \mid a \in \{0,1\}^n \text{ and } f_a(x) = x \cdot a \bmod 2\}.$$

Given an oracle which responds to a query $x \in \{0,1\}^n$ by evaluating $f_a(x)$ for some $f_a \in \mathcal{BV}^n$, the improved [4,5] Bernstein-Vazirani algorithm identifies a with a single quantum query, rather than the n required classically. This can be thought of as a *search* problem: an n bit number a is to be found by querying the oracle which responds with the parity of $x \cdot a$, where x is the query.

The most familiar quantum search algorithm is due to Grover [7]. This is also best discussed as an oracle problem [8,5,9]: Let

$$\mathcal{G}^n = \{g_a : \{0,1\}^n \to \{0,1\} \mid a \in \{0,1\}^n \text{ and } g_a(x) = \delta_{xa}\},$$

where $\delta_{xa} = 1$ if $x = a$ and 0 otherwise. Given an oracle which responds to a query $x \in \{0,1\}^n$ by evaluating $g_a(x)$ for some $g_a \in \mathcal{G}^n$, Grover's algorithm identifies a with probability bounded above $\frac{1}{2}$ using $O(2^{n/2})$ quantum queries rather than the $\Omega(2^n)$ queries required classically. Here the 'naïve' Grover oracle only responds that a query is correct, or not—so both the classical and quantum algorithms require more queries than the corresponding algorithms for the 'sophisticated' [10] Bernstein-Vazirani oracle.

More precisely, the response of the Bernstein-Vazirani oracle depends on the structure of the search problem as a product of bits; it computes the number of bits in the query which are 1s and are correct, *i.e.*, the same as in the hidden bit string a. The Grover oracle ignores this structure, responding only that the whole bit string of the query is the same as a, or not. In this paper we consider a new class of structured search problems, *i.e.*, families

of oracles whose responses depend on the product structure of the space being searched, and prove that there are separations between the quantum and classical query complexities of these problems. We are interested specifically in *highly* structured search problems, *i.e.*, problems for which all the factors in the search space have the same dimension. This can be contrasted with the situations considered by Farhi and Gutmann [11], by Grover [12], and by Cerf, Grover and Williams [13] in which the search space decomposes, but not necessarily into isomorphic factors. We exploit the additional structure to develop different—and more efficient—quantum algorithms than apply in these less structured problems. Hogg has also considered highly structured search problems—in the context of k-SAT [14]—and for $k = 1$ constructs an algorithm similar to our Algorithm A.

In the next section we define the first of this class of problems and show how it differs from the Bernstein-Vazirani problem. Using the two bit response of the oracle, we present a single query quantum algorithm in §3. In §4 we generalize the problem, and the quantum algorithm, to cases where the structure of the search problem is a product of 3- or 4-dimensional factors, and in §5 to higher dimensional factors for which more than one query is required. We conclude with a brief discussion in §6.

2. The problem

The Bernstein-Vazirani algorithm solves a highly structured search problem using information about (the parity of) the number of 1s in query bit strings which are correct. Our goal is to find efficient quantum algorithms which use information about the number of factors in the query which are correct, no matter what their values are. Thus we define

$$\mathcal{H}_2^n = \{h_a : \{0,1\}^n \to \{0,1,2,3\} \mid a \in \{0,1\}^n \text{ and } h_a(x) = \mathrm{dist}(x,a) \bmod 4\},$$

where $\mathrm{dist}(x,a)$ is the Hamming distance between a and x, *i.e.*, the number of bits at which they differ. The subscript 2 indicates that the answer a is a bit string—its tensor factors have dimension 2—and the superscript n indicates its length. Given an oracle which responds to a query $x \in \{0,1\}^n$ by evaluating $h_a(x)$ for some $h_a \in \mathcal{H}_2^n$, our quantum algorithm will identify a with a single query.

Notice that unlike the Bernstein-Vazirani oracle, the response is returned mod 4, rather than mod 2. It is easy to see that with a mod 2 response there is *no* solution to the problem, classically or quantum mechanically:

LEMMA 1. *Let* $a, a' \in \{0,1\}^n$. *If* $\mathrm{dist}(a,a') \equiv 0 \bmod 2$ *then for all* $x \in \{0,1\}^n$, $h_a(x) = h_{a'}(x)$.

Proof. The bits at which a and a' agree contribute equally to $h_a(x)$ and $h_{a'}(x)$. There are also an even number of bits at which a and a' differ. Each contributes 1 to either $h_a(x)$ or to $h_{a'}(x)$, but not to both. ∎

Thus the responses of oracles hiding a and a' are identical when $\mathrm{dist}(a,a') \equiv 0 \bmod 2$, which means the oracles are indistinguishable by any classical or quantum mechanical algorithm.

In the next section we show that when the response is returned mod 4, however, the problem is solvable. The fact that our oracle returns a two bit response not only distinguishes it from the Bernstein-Vazirani oracle, but also means that no quantum algorithm for weighing matrix problems, considered recently by van Dam [15], can solve this problem. Classically, of course, the same kind of information-theoretic argument which provides lower bounds on the number of queries necessary to identify elements of $\mathcal{DJ}^n$, $\mathcal{BV}^n$ or $\mathcal{G}^n$ shows that any classical algorithm requires $\Omega(n/2)$ queries to identify an element of $\mathcal{H}_2^n$.

3. The algorithm

To describe a quantum algorithm which solves the problem posed in §2 we must introduce notation for a few standard unitary transformations. Let $F_d : \mathbb{C}^d \to \mathbb{C}^d$ be the d-dimensional discrete Fourier transform. In a basis $\{|0\rangle, \ldots, |d-1\rangle\}$,

$$F_d : |q\rangle \mapsto \frac{1}{\sqrt{d}} \sum_{p=0}^{d-1} e^{2\pi i p q/d} |p\rangle,$$

and the inverse transform is

$$F_d^{-1} : |p\rangle \mapsto \frac{1}{\sqrt{d}} \sum_{q=0}^{d-1} e^{-2\pi i p q/d} |q\rangle.$$

Also, let $T_d : \mathbb{C}^d \to \mathbb{C}^d$ denote the one step shift operator

$$T_d : |q\rangle \mapsto |q \oplus 1\rangle,$$

where $\oplus$ denotes addition mod d. When $d = 2$, $F_2 = \left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)/\sqrt{2}$, the 'Hadamard transform' H, and $T_2 = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$, the Pauli matrix σ_x.

The 'data structure' for the algorithm we will describe is $(\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^4$, query and response 'registers'. The oracle acts by the unitary transformation O_{h_a} defined by linear extension from

$$|x, b\rangle \mapsto |x, b \oplus h_a(x)\rangle = I_{2^n} \otimes T_4^{h_a(x)} |x, b\rangle. \tag{1}$$

This is exactly analogous to the action of the oracles in the Deutsch-Jozsa, Bernstein-Vazirani and Grover algorithms [16]; only the possible functions $h_a \in \mathcal{H}_2^n$ and the size of the response register, 4- rather than 2-dimensional, differ.

THEOREM 2. *An element of $\mathcal{H}_2^n$ can be identified with probability* 1 *using only a single quantum query.*

Proof. We give an explicit algorithm:

Algorithm A.

1. Initialize the state to $|0 \ldots 0\rangle \otimes |0\rangle \in (\mathbb{C}^2)^{\otimes n} \otimes \mathbb{C}^4$.

2. Apply the unitary transformation $F_2^{\otimes n} \otimes (F_4 T_4)$. Since

$$F_4 T_4 |0\rangle = F_4 |1\rangle = \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle)$$

and $F_2|0\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, this produces the state

$$\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} |x\rangle \otimes \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle).$$

3. Call the oracle, *i.e.*, apply O_{h_a}. Since

$$T_4 \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle) = -i\frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle),$$

this produces the state

$$\frac{1}{2^{n/2}} \sum_{x=0}^{2^n-1} (-i)^{h_a(x)} |x\rangle \otimes \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle).$$

4. Apply the unitary transformation $\left(\left(\begin{smallmatrix} 1 & i \\ i & 1 \end{smallmatrix}\right)/\sqrt{2}\right)^{\otimes n} \otimes I_4$. This produces the state

$$\frac{1}{2^n} \sum_{x,y=0}^{2^n-1} (i)^{h_x(y)} (-i)^{h_a(x)} |y\rangle \otimes \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle) = |a\rangle \otimes \frac{1}{2}(|0\rangle + i|1\rangle - |2\rangle - i|3\rangle).$$

5. Measure the query register to obtain a with probability 1. ∎

We saw in Lemma 1 that no algorithm could solve the problem if the oracle responds with the Hamming distance modulo 2, and we have just proved that there is a single query quantum algorithm if the Hamming distance is returned modulo 4. In fact, the oracle *must* return this information for a phase-kickback algorithm [5] like this one to succeed: We chose the single qubit operator $\left(\begin{smallmatrix} 1 & \alpha \\ \alpha & 1 \end{smallmatrix}\right)/\sqrt{2}$ in step 4 of Algorithm A to contribute a factor proportional to a fixed phase α when the corresponding bits in x and y differ, and to contribute a factor proportional to 1 when they are the same. For this operator to be unitary, we must have $\alpha + \bar{\alpha} = 0$, which means that it can only cancel a kicked-back phase of i (or equivalently, $-i$). Hence we must have a 4-dimensional response register.

4. Generalization to dimensions 3 and 4

We can generalize to the problem of identifying a string which is the base k representation of a non-negative integer, given an oracle which responds according to the number of correct components in a query string. To construct algorithms analogous to Algorithm A, we require $k \times k$ unitary operators U_k such that

$$\langle i | U_k | j \rangle = \frac{1}{\sqrt{k}} \begin{cases} 1 & \text{if } i = j; \\ \alpha_k & \text{if } i \neq j,\ |\alpha_k| = 1. \end{cases} \tag{2}$$

U_k is unitary iff $\alpha_k + \bar{\alpha}_k = 2 - k$, which has solutions $\alpha_2 = i$, $\alpha_3 = \omega$ (ω a nontrivial cube root of 1), $\alpha_4 = -1$, and none for $k > 4$. Thus for $k \geq 2$, we define $r = \max\{2, 6 - k\}$ and let

$$\mathcal{H}_k^n = \{h_a : \{0, \ldots, k-1\}^n \to \{0, \ldots, r-1\} \mid a \in \{0, \ldots, k-1\}^n \text{ and } h_a(x) = \operatorname{dist}(x, a) \bmod r\},$$

where $\operatorname{dist}(x, a)$ is the generalized Hamming distance between a and x, *i.e.*, the number of components at which they differ. This specializes to our previous definition for $\mathcal{H}_2^n$ and preserves the sense of the original problem since $h_a(x)$ can still be written as n less the number of correct components in the query. The base k oracle acts as before, where the response register in (1) is now r-dimensional. Just as in §2, this implies information theoretic lower bounds on the number of queries required classically: $\Omega(n \log_r k)$. For $k = 3$ or 4 we can do much better quantum mechanically:

THEOREM 3. *An element of $\mathcal{H}_k^n$, for $k \in \{2, 3, 4\}$, can be identified with probability* 1 *using only a single quantum query.*

Proof. We give an explicit algorithm by generalizing Algorithm A:

Algorithm B.
1. Initialize the state to $|0 \ldots 0\rangle \otimes |0\rangle \in (\mathbb{C}^k)^{\otimes n} \otimes \mathbb{C}^r$.
2. Apply the unitary transformation $F_k^{\otimes n} \otimes (F_r T_r)$.
3. Call the oracle, *i.e.*, apply O_{h_a}.
4. Apply the unitary transformation $U_k^{\otimes n} \otimes I_r$.
5. Measure the query register.

Calculations parallel to those in the proof of Theorem 2 verify that the measurement in step 5 obtains a with probability 1. ∎

5. Higher dimensions

For $k > 4$, there is no unitary U_k as defined by (2), so we cannot give an exact single query quantum algorithm analogous to the one which works for $k \in \{2, 3, 4\}$. Notice, however, that

$$\operatorname{dist}(x, a) = \sum_{i=0}^{n-1} \operatorname{dist}(x_i, a_i) = n - \sum_{i=0}^{n-1} \delta_{x_i a_i},$$

where $x = \sum k^i x_i$ and $a = \sum k^i a_i$ for $i \in \{0, \ldots, n-1\}$. Thus for $k \geq 2$, if we apply steps 1–3 of Algorithm B, the state is

$$\frac{1}{k^{n/2}} \sum_{x=0}^{k^n-1} (-1)^{h_a(x)} |x\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) = \frac{(-1)^n}{k^{n/2}} \sum_{x=0}^{k^n-1} \bigotimes_{i=0}^{n-1} (-1)^{\delta_{x_i a_i}} |x_i\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$$
$$= (-1)^n \bigotimes_{i=0}^{n-1} \frac{1}{\sqrt{k}} \sum_{x_i=0}^{k-1} (-1)^{\delta_{x_i a_i}} |x_i\rangle \otimes \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

This is the tensor product of n k-dimensional factors, the i^{th} in the state it would be had a Grover oracle evaluating the function $g_{a_i} \in \mathcal{G}^k$ been called, tensor the state $(|0\rangle - |1\rangle)/\sqrt{2}$. Then, when $k = 4$, step 4 of Algorithm B multiplies each of the first n tensor factors by

$$U_4 = \frac{1}{2}\begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & -1 & 1 & -1 \\ -1 & -1 & -1 & 1 \end{pmatrix} = I_4 - 2F_4|0\rangle\langle 0|F_4^{-1}.$$

This is reflection about the hyperplane orthogonal to $(|0\rangle + |1\rangle + |2\rangle + |3\rangle)/2$, which is the second half of the iteration step in Grover's algorithm [7,17,18,19], applied to identify $g_a \in \mathcal{G}^4$.

With this as motivation, we introduce $k \times k$ unitary operators V_k such that

$$\langle i\,|V_k\,|j\rangle = \begin{cases} 1 - \frac{2}{k} & \text{if } i = j; \\ -\frac{2}{k} & \text{if } i \neq j. \end{cases}$$

These are a natural extension to higher dimensions of the U_k, in the sense that they have one value on the diagonal and another off it. Furthermore, $V_4 = U_4$, and in general

$$V_k = I_k - 2F_k|0\rangle\langle 0|F_k^{-1},$$

i.e., reflection in the hyperplane orthogonal to $(|0\rangle + \cdots + |k-1\rangle)/\sqrt{k} = F_k|0\rangle$, again the second half of the iteration step in Grover's algorithm [7,17,18,19], applied to identify $g_a \in \mathcal{G}^k$. Let $\theta = \csc^{-1}(\sqrt{k})$ be the angle between $F_k|0\rangle$ and each basis state. Then for $k \geq 2$ we have:

Algorithm C.

1. Initialize the state to $|0\ldots0\rangle \otimes |0\rangle \in (\mathbb{C}^k)^{\otimes n} \otimes \mathbb{C}^2$.
2. Apply the unitary transformation $F_k^{\otimes n} \otimes (F_2 \circ T_2)$.
3. Repeat $\left\lfloor \frac{1}{2}\left(\frac{\pi}{2\theta} - 1\right) \right\rceil$ (where $\lfloor\cdot\rceil$ denotes 'closest integer to') times:
  a. Call the oracle, *i.e.*, apply O_{h_a}.
  b. Apply the unitary transformation $V_k^{\otimes n} \otimes I_2$.
4. Measure the query register.

THEOREM 4. *Algorithm C identifies an element of $\mathcal{H}_k^n$ with probability at least $\frac{1}{2} + \epsilon$ $(0 < \epsilon \leq \frac{1}{2})$ for $n \leq -k\ln(\frac{1}{2} + \epsilon)$, using $\left\lfloor \frac{1}{2}\left(\frac{\pi}{2\theta} - 1\right) \right\rceil \sim \left\lfloor \frac{\pi}{4}\sqrt{k} \right\rceil$ quantum queries.*

Proof. As we explained in the previous paragraph, Algorithm C simultaneously implements Grover's algorithm for each of the n components in the query register. Since these components are k-dimensional, after $\left\lfloor \frac{1}{2}\left(\frac{\pi}{2\theta} - 1\right) \right\rceil$ iterations, the state of each factor is within an angle $\theta = \csc^{-1}(\sqrt{k})$ of the solution vector $|a_i\rangle$ in that factor. Measurement of each factor identifies a_i correctly with probability at worst $1 - \frac{1}{k}$, independently. The probability that all n factors are correct is at worst $(1 - \frac{1}{k})^n$; for this to be bounded above $\frac{1}{2}$, n must be no more than $-k\ln(\frac{1}{2} + \epsilon)$. ∎

6. Discussion

We have defined a set of problems in which the objective is to identify a hidden number using base k queries and feedback about the number of components which are correct. Algorithm B solves the problem exactly for $k \in \{2, 3, 4\}$ with a single quantum query, using the Hamming distance modulo $r = \max\{2, 6 - k\}$. Algorithm C solves the problem with probability bounded above $\frac{1}{2}$ for $k \geq 2$, using $O(\sqrt{k})$ queries, but only for numbers with no more than $n = O(k)$ components. Classically these problems require $\Omega(n \log_r k)$ queries, so each algorithm gives a quantum improvement. For $k \in \{2, 3, 4\}$, Algorithm B clearly gives a best possible solution. For $k > 4$, Algorithm C must also be a best possible solution since an algorithm requiring fewer queries could be applied to a single component and would outperform Grover's algorithm, which is known to be optimal [8,4,9]. To eliminate the bound on the length of the string, n, step 3 of Algorithm C can be replaced by one of the generalizations of Grover's algorithm which identifies a solution with probability 1 [9,20,21]. Such a modification of Algorithm C identifies an element of $\mathcal{H}_k^n$ with $O(\sqrt{k})$ queries, for *any* n. Finally, we remark that recursive versions of the $k \in \{2, 3, 4\}$ problems can be defined and solved analogously to the recursive parity problem of Bernstein and Vazirani [6], to increase the quantum from classical separation.

Acknowledgements

This work was supported in part by the National Security Agency (NSA) and Advanced Research and Development Activity (ARDA) under Army Research Office (ARO) contract number DAAG55-98-1-0376, and also by the Defense Advanced Research Projects Agency (DARPA) under DARPA/SSC contract number N66001-00-C-8040, subcontracted through Orincon Corporation.

References

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring", in S. Goldwasser, ed., *Proceedings of the 35th Symposium on Foundations of Computer Science*, Santa Fe, NM, 20–22 November 1994 (Los Alamitos, CA: IEEE Computer Society Press 1994) 124–134;
P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer", *SIAM J. Comput.* **26** (1997) 1484–1509.

[2] A. K. Lenstra, H. W. Lenstra, Jr., M. S. Manasse and J. M. Pollard, "The number field sieve", in *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, Baltimore, MD, 14–16 May 1990 (New York: ACM Press 1990) 564–572;
A. K. Lenstra and H. W. Lenstra, Jr., eds., *The Development of the Number Field Sieve*, Lecture Notes in Mathematics, vol. 1554 (New York: Springer-Verlag 1993).

[3] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation", *Proc. Roy. Soc. Lond. A* **439** (1992) 553–558.

[4] C. H. Bennett, E. Bernstein, G. Brassard and U. Vazirani, "Strengths and weaknesses of quantum computing", *SIAM J. Comput.* **26** (1997) 1510–1523.

[5] R. Cleve, A. Ekert, C. Macchiavello and M. Mosca, "Quantum algorithms revisited",

Proc. Roy. Soc. Lond. A **454** (1998) 339–354.

[6] E. Bernstein and U. Vazirani, "Quantum complexity theory", in *Proceedings of the 25th ACM Symposium on Theory of Computing*, San Diego, CA, 16–18 May 1993 (New York: ACM Press 1993) 11–20;
E. Bernstein and U. Vazirani, "Quantum complexity theory", *SIAM J. Comput.* **26** (1997) 1411–1473.

[7] L. K. Grover, "A fast quantum mechanical algorithm for database search", in *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, Philadelphia, PA, 22–24 May 1996 (New York: ACM 1996) 212–219.

[8] M. Boyer, G. Brassard, P. Høyer and A. Tapp, "Tight bounds on quantum searching", *Fortsch. Phys.* **46** (1998) 493–506.

[9] C. Zalka, "Grover's quantum searching algorithm is optimal", *Phys. Rev. A* **60** (1999) 2746–2751.

[10] D. A. Meyer, "Sophisticated quantum search without entanglement", *Phys. Rev. Lett.* **85** (2000) 2014–2017.

[11] E. Farhi and S. Gutmann, "Quantum mechanical square root speedup in a structured search problem", `quant-ph/9711035`.

[12] L. K. Grover, "Quantum search on structured problems", *Chaos, Solitons & Fractals* **10** (1999) 1695–1705.

[13] N. J. Cerf, L. K. Grover and C. P. Williams, "Nested quantum search and structured problems", *Phys. Rev. A* **61** (2000) 032303/1–14.

[14] T. Hogg, "Highly structured searches with quantum computers", *Phys. Rev. Lett.* **80** (1998) 2473–2476.

[15] W. van Dam, "Quantum algorithms for weighing matrices and quadratic residues", `quant-ph/0008059`.

[16] D. A. Meyer, "Quantum games and quantum algorithms", `quant-ph/0004092`; to appear in the AMS *Contemporary Mathematics* volume: *Quantum Computation and Quantum Information Science*.

[17] E. Farhi and S. Gutmann, "Analog analogue of a digital quantum computation", *Phys. Rev. A* **57** (1998) 2403–2406.

[18] R. Jozsa, "Searching in Grover's algorithm", `quant-ph/9901021`.

[19] G. Brassard, P. Høyer, M. Mosca and A. Tapp, "Quantum amplitude amplification and estimation", `quant-ph/0005055`.

[20] P. Høyer, "Arbitrary phases in quantum amplitude amplification", *Phys. Rev. A* **62** (2000) 052304/1–5.

[21] G. L. Long, "Grover algorithm with zero theoretical failure rate", `quant-ph/0106071`.