Tight Bounds on Quantum Searching ⋆

Michel Boyer [a], Gilles Brassard [a,1]
Peter Høyer [b,2] and Alain Tapp [a,3]

[a] *Département IRO, Université de Montréal*
C.P. 6128, succursale centre-ville, Montréal (Québec), Canada H3C 3J7.
email: {boyer,brassard,tappa}@iro.umontreal.ca.

[b] *Department of Mathematics and Computer Science, Odense University*
Campusvej 55, DK–5230 Odense M, Denmark. email: u2pi@imada.ou.dk.

We provide a tight analysis of Grover's algorithm for quantum database searching. We give a simple closed-form formula for the probability of success after any given number of iterations of the algorithm. This allows us to determine the number of iterations necessary to achieve almost certainty of finding the answer. Furthermore, we analyze the behaviour of the algorithm when the element to be found appears more than once in the table and we provide a new algorithm to find such an element even when the number of solutions is not known ahead of time. Finally, we provide a lower bound on the efficiency of any possible quantum database searching algorithm and we show that Grover's algorithm comes within 2.62% of being optimal.

Key words: Quantum computation. Searching. Lower bound. Oracle. Soufflés.
PACS: 03.65.Bz, 89.80.+h, 89.70.+c, 02.70.-c

⋆ This research was presented at the Fourth Workshop on Physics and Computation, Boston, 23 November 1996.
[1] Supported in part by Canada's NSERC and Québec's FCAR.
[2] Supported in part by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT). Current address: Département IRO, Université de Montréal.
[3] Supported in part by a postgraduate fellowship from Canada's NSERC.

1 Introduction

Let $X_N = \{0, 1, \ldots, N-1\}$ for some integer N and consider an arbitrary function $F : X_N \rightarrow \{0, 1\}$. The goal is to find some $i \in X_N$ such that $F(i) = 1$, provided such an i exists. If F is given as a black box—the only knowledge you can gain about F is in asking for its value on arbitrary points of its domain—and if there is a unique solution, no classical algorithm (deterministic or probabilistic) can expect to achieve a probability of success better than 50% without asking for the value of F on roughly $N/2$ points. Throughout this paper we assume for simplicity that each evaluation of F takes unit time. Grover [1] has discovered an algorithm for the *quantum* computer that can solve this problem in expected time in $O(\sqrt{N}\,)$, provided there is a unique solution. He also remarked that a result in [2] implies that his algorithm is optimal, up to an unspecified multiplicative constant, among all possible quantum algorithms.

In this paper we provide a tight analysis of Grover's algorithm. In particular we give a simple closed-form formula for the probability of success after any given number of iterations. This allows us to determine the number of iterations necessary to achieve almost certainty of finding the answer, as well as an upper bound on the probability of failure. More significantly, we analyze the behaviour of the algorithm when there is an arbitrary number of solutions. An algorithm follows immediately to solve the problem in a time in $O(\sqrt{N/t}\,)$ when it is known that there are exactly t solutions. Moreover we provide an algorithm capable of solving the problem in a time in $O(\sqrt{N/t}\,)$ even if the number t of solutions is not known in advance. We also generalize Grover's algorithm to the case N is not a power of 2. Finally, we refine the argument of [2] to show that Grover's algorithm is within 2.62% of being optimal.

To motivate this work, here are three simple applications for Grover's algorithm. Assume you have a large table $T[0\,..\,N-1]$ in which you would like to find some element y. More precisely, you wish to find an integer i such that $0 \leq i < N$ and $T[i] = y$, provided such an i exists. This *database searching problem* can obviously be solved in a time in $O(\log N)$ if the table is sorted, but no classical algorithm can succeed in the general case with probability better than 50%, say, without probing more than half the entries of T. Grover's algorithm solves this problem in a time in $O(\sqrt{N}\,)$ on the quantum computer by using $F(i) = 1$ if and only if $T[i] = y$. An exciting cryptographic application is that Grover's algorithm can be used to crack the widely used Data Encryption Standard (DES) [3] under a known plaintext attack. Given a matching pair (m, c) of plaintext and ciphertext, consider function $F : \{0, 1\}^{56} \rightarrow \{0, 1\}$ defined by $F(k) = 1$ if and only if $\text{DES}_k(m) = c$. Provided there is a unique solution, the required key k can be found after roughly 185 million expected calls to a quantum DES device [4]. Thus quantum computing makes single-key DES totally insecure. For yet another application, consider a Boolean formula on n variables. You would like to determine if the formula is satisfiable. There may exist an efficient classical algorithm for this problem but none are known. (This is equivalent to the famous $\mathbf{P} \stackrel{?}{=} \mathbf{NP}$ open question in theoretical computer science [5]). In this case Grover's algorithm solves the problem in a time in $O(2^{n/2})$, which is better than the time in $O(2^n)$ required by the obvious classical algorithm, but not good enough to imply that $\mathbf{NP} \subseteq \mathbf{BQP}$ [2].

2 Overview of Grover's Algorithm

Grover's algorithm consists of an initialization followed by a number of identical iterations, a final measurement, and a classical test. For every $F : X_N \to \{0, 1\}$, let S_F be the conditional phase shift transform defined by

$$S_F|i\rangle \; = \; \begin{cases} -|i\rangle \text{ if } F(i) = 1 \\ \;\; |i\rangle \text{ otherwise.} \end{cases}$$

Let S_0 denote S_{F_0}, where $F_0(i) = 1$ if and only if $i = 0$.

Assume for the moment that $N = 2^n$ is a power of 2 and consider any integer $j \in X_N$ as a bit string of length n. Define $i \cdot j$ as the number of 1 in the bitwise AND of i and j. Let W be the Walsh–Hadamard transform defined by

$$W|j\rangle \; = \; \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (-1)^{i \cdot j} |i\rangle.$$

This is efficiently implemented [6] by applying the simple unitary transformation $\frac{1}{\sqrt{2}}\left(\begin{smallmatrix} 1 & 1 \\ 1 & -1 \end{smallmatrix}\right)$ independently to each qubit of $|j\rangle$. Now we can define one *Grover iteration* as the unitary transformation

$$G_F = -W S_0 W S_F. \tag{1}$$

Grover's algorithm first creates a state $|\Psi\rangle = W|0\rangle$. Then G_F is applied to $|\Psi\rangle$ some number m of times. (One primary purpose of this paper is to determine the optimal choice for m.) Finally, the state $|\Psi\rangle$ is measured, which yields some classical value i. The algorithm *succeeds* if and only if $F(i) = 1$.

Let us now assume we are given a quantum black box Q_F for computing F. This will usually come as a unitary transformation that sends state $|i, b\rangle$ to $|i, b \oplus F(i)\rangle$, where $|b\rangle$ is a single qubit and $\oplus$ denotes the exclusive–or. The obvious approach to implementing S_F as a unitary transformation requires two applications of Q_F: if P is the conditional phase-shift defined by $P|i, b\rangle = (-1)^b|i, b\rangle$ then $(S_F|i\rangle)|0\rangle$ can be computed as $Q_F P Q_F|i, 0\rangle$.

However, it follows from Lemma 5.5 in [7] that S_F can be implemented using a single application of Q_F. For this, it suffices to note that

$$(S_F|i\rangle)|\Delta\rangle = Q_F(|i\rangle|\Delta\rangle)$$

where $|\Delta\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$.

The Walsh–Hadamard transform W is well-defined only if N is a power of 2. However, this assumption on N can be removed by observing that G_F is just one of many transforms that can be used as iteration in Grover's algorithm. Let W' be any unitary transform satisfying

$$W'|0\rangle \;=\; \frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle. \tag{2}$$

Then one may easily verify that the transform $G'_F = W' S_0 W'^{\dagger} S_F$ works just as well. (The minus sign in eq. (1) was clearly unnecessary although it makes the analysis easier.) Any transform W' satisfying eq. (2) can thus be used in Grover's algorithm. When N is a power of 2, the Walsh–Hadamard transform is indeed the simplest possible choice for W'. When N is not a power of 2, the approximate Fourier transform given by Kitaev [8] can be used.

3 Finding a Unique Solution

Assume for now that there is a unique i_0 such that $F(i_0) = 1$. For any real numbers k and ℓ such that $k^2 + (N-1)\ell^2 = 1$, define the state of a quantum register

$$|\Psi(k,\ell)\rangle \;=\; k|i_0\rangle + \sum_{i \neq i_0} \ell|i\rangle$$

where the sum is over all $i \neq i_0$ such that $0 \leq i < N$.

The heart of Grover's algorithm is the iteration described in the previous section. A simple calculation—see Grover's original article [1] for details—shows that each iteration efficiently transforms $|\Psi(k,\ell)\rangle$ into

$$\left|\Psi\left(\tfrac{N-2}{N}k + \tfrac{2(N-1)}{N}\ell, \tfrac{N-2}{N}\ell - \tfrac{2}{N}k\right)\right\rangle.$$

It follows that the j–th iteration produces state $|\Psi_j\rangle = |\Psi(k_j, \ell_j)\rangle$ where

$$k_{j+1} = \tfrac{N-2}{N}k_j + \tfrac{2(N-1)}{N}\ell_j \quad \text{and} \quad \ell_{j+1} = \tfrac{N-2}{N}\ell_j - \tfrac{2}{N}k_j \tag{3}$$

with initial conditions $k_0 = l_0 = 1/\sqrt{N}$.

In his paper, Grover proves that there exists a number m less than $\sqrt{2N}$ such that k_m^2, the probability of success after m iterations, is at least 50%. This is correct, but one must be careful in using his algorithm because the probability of success does not increase monotonically with the number of iterations. By the time you have performed $\sqrt{2N}$ iterations, the probability of success has dropped down to less than 9.5% and it becomes vanishingly small after about 11% more iterations before it picks up again. This shows that it is not sufficient to know the existence of m in order to apply the algorithm: its explicit value is needed.

The key to a tighter analysis of Grover's algorithm is an explicit closed-form formula for k_j and ℓ_j. This can be obtained by standard techniques—and a little sweat—from recurrence (3). Let angle θ be defined so that $\sin^2\theta = 1/N$ and $0 < \theta \leq \pi/2$. It is straightforward to verify by mathematical induction that

$$k_j = \sin((2j+1)\theta) \quad \text{and} \quad \ell_j = \tfrac{1}{\sqrt{N-1}} \cos((2j+1)\theta). \tag{4}$$

It follows from eq. (4) that $k_m = 1$ when $(2m+1)\theta = \pi/2$, which happens when $m = (\pi - 2\theta)/4\theta$. Of course, we must perform an *integer* number of iterations but it will be shown in the next section that the probability of failure is no more than $1/N$ if we iterate $\lfloor \pi/4\theta \rfloor$ times. This is essentially $\frac{\pi}{4}\sqrt{N}$ iterations when N is large because $\theta \approx \sin\theta = 1/\sqrt{N}$ when θ is small. It is sufficient to perform half this number of iterations, approximately $\frac{\pi}{8}\sqrt{N}$, if we are satisfied with a 50% probability of success, as Grover considered in his original paper [1]. We shall prove in Section 7 that this is optimal within a few percent because any quantum algorithm that solves the search problem with a 50% probability of success must evaluate F at least $(\sin\frac{\pi}{8})\sqrt{N}$ times and $\frac{\pi}{8} \approx 1.026 \sin\frac{\pi}{8}$. One must know when to stop, however: if we work twice as hard as we would need to succeed with almost certainty, that is we apply approximately $\frac{\pi}{2}\sqrt{N}$ iterations of Grover's algorithm, we fail with near certainty!

4 The Case of Multiple Solutions

Let us now consider the case when there are t different values of i such that $F(i) = 1$. We are interested in finding an arbitrary solution. Grover briefly considers this problem [1], but he provides no details concerning the efficiency of his method.

We assume in this section that the number t of solutions is known and that it is not zero. Let $A = \{i \mid F(i) = 1\}$ and $B = \{i \mid F(i) = 0\}$. For any real numbers

k and ℓ such that $tk^2 + (N-t)\ell^2 = 1$, redefine

$$|\Psi(k,\ell)\rangle \;=\; \sum_{i\in A} k|i\rangle + \sum_{i\in B} \ell|i\rangle .$$

A straightforward analysis of Grover's algorithm shows that one iteration transforms $|\Psi(k,\ell)\rangle$ into

$$\left|\Psi\left(\tfrac{N-2t}{N}k + \tfrac{2(N-t)}{N}\ell,\, \tfrac{N-2t}{N}\ell - \tfrac{2t}{N}k\right)\right\rangle .$$

This gives rise to a recurrence similar to (3), whose solution is that the state $|\Psi(k_j,\ell_j)\rangle$ after j iterations is given by

$$k_j = \tfrac{1}{\sqrt{t}}\sin((2j+1)\theta) \quad \text{and} \quad \ell_j = \tfrac{1}{\sqrt{N-t}}\cos((2j+1)\theta) \tag{5}$$

where the angle θ is so that $\sin^2\theta = t/N$ and $0 < \theta \le \pi/2$.

The probability of obtaining a solution is maximized when ℓ_m is as close to 0 as possible. We would have $\ell_{\tilde{m}} = 0$ when $\tilde{m} = (\pi - 2\theta)/4\theta$ if that were an integer. Let $m = \lfloor \pi/4\theta \rfloor$. Note that $|m - \tilde{m}| \le \frac{1}{2}$. It follows that $|(2m+1)\theta - (2\tilde{m}+1)\theta| \le \theta$. But $(2\tilde{m}+1)\theta = \pi/2$ by definition of $\tilde{m}$. Therefore $|\cos((2m+1)\theta)| \le |\sin\theta|$. We conclude that the probability of failure after exactly m iterations is

$$(N-t)\ell_m^2 \;=\; \cos^2((2m+1)\theta) \;\le\; \sin^2\theta \;=\; t/N.$$

This is negligible when $t \ll N$.

Note that this algorithm runs in a time in $O(\sqrt{N/t}\,)$ since $\theta \ge \sin\theta = \sqrt{t/N}$ and therefore

$$m \;\le\; \frac{\pi}{4\theta} \;\le\; \frac{\pi}{4}\sqrt{\frac{N}{t}}.$$

A slight improvement is possible in terms of the expected time if we stop short of m iterations, observe the register, and start all over again in case of failure. The expected number of iterations before success with this strategy is $E(j) = j/tk_j^2$ if we stop after j iterations since our probability of success at that point is tk_j^2. Setting the derivative of $E(j)$ to 0, we find that the optimal number of iterations is given by the j so that $4\theta j = \tan((2j+1)\theta)$. The solution to this equation is very close to $j = z/4\theta$ when $t \ll N$, where $z \approx 2.33112$ is such that $z = \tan(z/2)$. It follows that the optimal number of iterations is close to $0.58278\sqrt{N/t}$ when $t \ll N$ and the probability of success is close to $\sin^2(z/2) \approx 0.84458$. Therefore, the expected number of iterations before success if we restart the process in case of failure is roughly $(z/(4\sin^2(z/2)))\sqrt{N/t} \approx 0.69003\sqrt{N/t}$, which is about 88% of $\frac{\pi}{4}\sqrt{N/t}$, the number of iterations after which success is almost certain. For a numerical example, consider $N = 2^{20}$ and $t = 1$. In this case, we achieve almost certainty of success after 804 iterations. If, instead, we stop at 596 iterations, the probability of success is only 0.84420 but the expected number of iterations before success if we restart the process in case of failure is $596/0.8442 \approx 706$, which is indeed better than 804.

5 The Case $t = N/4$

An interesting special case occurs when $t = N/4$. Of course, even a classical probabilistic computer can solve this problem efficiently, with high probability, but not quite as efficiently as a quantum computer. Here $\sin^2\theta = t/N = 1/4$ and therefore $\theta = \pi/6$. It follows that $\ell_1 = \frac{1}{\sqrt{N-t}}\cos(3\theta) = 0$. In other words, a solution is found *with certainty* after a single iteration. In terms of the number of times F has to be evaluated, this is essentially four times more efficient than the expected performance of the best possible classical probabilistic algorithm when N is large. Furthermore, the quantum algorithm becomes *exponentially* better than any possible classical algorithm if we compare worst-case performances, taking the worst possible coin flips in the case of a probabilistic algorithm. This is somewhat reminiscent of the Deutsch–Jozsa algorithm [6].

6 Unknown Number of Solutions

A more challenging situation occurs when the number of solutions is not known ahead of time. If we decide to iterate $\frac{\pi}{4}\sqrt{N}$ times, which would give almost certainty of finding a solution if there were only one, the probability of success would be vanishingly small should the number of solutions be in fact 4 times a small perfect square. For example we saw that we are almost certain to find a unique solution among 2^{20} possibilities if we iterate 804 times. The same number of iterations would yield a solution with probability less than one in a million should there be 4 solutions! To find a solution efficiently when their number is unknown, we need the following lemmas, the first of which is easily proved by mathematical induction using straightforward algebra.

Lemma 1 *For any positive integer m and real number α such that* $\sin\alpha \neq 0$,

$$\sum_{j=0}^{m-1} \cos((2j+1)\alpha) \;=\; \frac{\sin(2m\alpha)}{2\sin\alpha}.$$

Lemma 2 *Let t be the (unknown) number of solutions and assume that $0 < t < N$. Let angle θ be so that $\sin^2\theta = t/N$ and $0 < \theta < \pi/2$. Let m be an arbitrary positive integer. Let j be an integer chosen at random according to the uniform distribution between 0 and $m-1$. If we observe the register after applying j iterations of Grover's algorithm starting from the initial state, the probability P_m of obtaining a solution is given by*

$$P_m \;=\; \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}.$$

In particular $P_m \geq 1/4$ when $m \geq 1/\sin(2\theta)$.

PROOF. The probability of success if we perform j iterations of Grover's algorithm is $tk_j^2 = \sin^2((2j+1)\theta)$. It follows that the average success probability when $0 \leq j < m$ is chosen randomly is

$$\begin{aligned} P_m &= \sum_{j=0}^{m-1} \frac{1}{m}\sin^2((2j+1)\theta) \\ &= \frac{1}{2m}\sum_{j=0}^{m-1} 1 - \cos((2j+1)2\theta) \;=\; \frac{1}{2} - \frac{\sin(4m\theta)}{4m\sin(2\theta)}. \end{aligned}$$

If $m \geq 1/\sin(2\theta)$ then

$$\frac{\sin(4m\theta)}{4m\sin(2\theta)} \;\leq\; \frac{1}{4m\sin(2\theta)} \;\leq\; \frac{1}{4}.$$

The conclusion follows. $\square$

We are now ready to describe the algorithm for finding a solution when the number t of solutions is unknown. For simplicity we assume at first that $1 \leq t \leq 3N/4$.

(i) Initialize $m = 1$ and set $\lambda = 8/7$.
(Any value of λ strictly between 1 and 4/3 would do.)
(ii) Choose an integer j uniformly at random such that $0 \leq j < m$.
(iii) Apply j iterations of Grover's algorithm starting from initial state

$$|\Psi_0\rangle \;=\; W|0\rangle \;=\; \frac{1}{\sqrt{N}}\sum_i |i\rangle.$$

(iv) Observe the register and let i be the outcome.
(v) If $F(i) = 1$, the problem is solved: **exit**.
(vi) Otherwise, set m to $\min(\lambda m, \sqrt{N})$ and go back to step (ii).

Theorem 3 *This algorithm finds a solution in expected time in $O(\sqrt{N/t}\,)$.*

PROOF. Let angle θ be so that $\sin^2\theta = t/N$ and $0 < \theta < \pi/2$,

$$m_0 \;=\; 1/\sin(2\theta) \;=\; \frac{N}{2\sqrt{(N-t)t}}$$

and $s_0 = \lceil \log_\lambda m_0 \rceil$. Note that $m_0 \le \sqrt{N/t}$ because $t \le 3N/4$.

We shall estimate the expected number of times that a Grover iteration is performed before a solution is found: the total time needed is clearly in the order of that number since we assumed that F can be evaluated in unit time. On the s–th time round the main loop, the value of m is $\min(\sqrt{N}, \lambda^{s-1})$ and the expected number of Grover iterations is less than half that value since j is chosen randomly between 0 and $m-1$. Note that $m < m_0$ for the first s_0 times round the main loop, whereas $m \ge m_0$ afterwards. We say that the algorithm reaches the *critical stage* when $m \ge m_0$ for the first time, which may never happen of course if success comes earlier.

The expected total number of Grover iterations needed to reach the critical stage, if it is reached, is at most

$$\tfrac{1}{2}\sum_{s=1}^{s_0} \lambda^{s-1} \;<\; \frac{1}{2}\frac{\lambda}{\lambda-1}m_0 \;=\; 4m_0 .$$

Thus, if the algorithm succeeds before reaching the critical stage, it does so in a time in $O(m_0)$, which is in $O(\sqrt{N/t}\,)$ as required.

If the critical stage is reached then every time round the main loop from this point on will succeed with probability at least $1/4$ by virtue of Lemma 2 since $m \ge 1/\sin(2\theta)$. Therefore, $\frac{1}{2}\lambda^{s_0}$ expected iterations will be performed at round $s = s_0 + 1$. This will succeed with probability at least $1/4$. With complementary probability at most $3/4$, at least one more trip round the loop will be necessary, requiring $\frac{1}{2}\lambda^{s_0+1}$ additional expected iterations. Again, this will succeed with probability at least $1/4$. With probability at most $(3/4)^2$, at least one more trip will be required, costing another $\frac{1}{2}\lambda^{s_0+2}$ expected iterations, and so on. Summing up, the expected number of Grover iterations needed to succeed once the critical stage has been reached is less than

$$\tfrac{1}{2}\sum_{u=0}^{\infty} \left(\tfrac{3}{4}\right)^u \lambda^{s_0+u} \;<\; \frac{2\lambda}{4-3\lambda}m_0 \;=\; 4m_0 .$$

The total expected number of Grover iterations, whether or not the critical stage is reached, is therefore less than $8m_0$ and thus the total expected time is in $O(\sqrt{N/t}\,)$ provided $0 < t \leq 3N/4$. Note that $8m_0 \approx 4\sqrt{N/t}$ when $t \ll N$, which is less than six times the expected number of iterations that we would have needed had we known the value of t ahead of time. The case $t > 3N/4$ can be disposed of in constant expected time by classical sampling. The case $t = 0$ is handled by an appropriate time-out in the above algorithm, which allows us to claim in a time in $O(\sqrt{N}\,)$ that there are no solutions when this is the case, with an arbitrarily small probability of failure when in fact there is a solution. $\square$

7 An Improved Lower Bound

Drawing on general results from [2], Grover points out in [1] that any algorithm for quantum database searching must take a time at least proportional to $\sqrt{N}$ to succeed with nonnegligible probability when there is a unique solution. In this section we prove that if the function F having t solutions is used as a black box in any quantum algorithm Q that makes less than $(\sin\frac{\pi}{8})\sqrt{\lfloor N/t \rfloor} - 1$ calls to F then, averaging over all such possible F, the probability that Q succeeds cannot be better than 50%. Obviously, it follows that, for any $t < N$ and any quantum algorithm that makes less than $(\sin\frac{\pi}{8})\sqrt{\lfloor N/t \rfloor} - 1$ calls to F, there exists an F that has t solutions, yet the algorithm's probability of success does not exceed 50%. This proves that Grover's algorithm comes within 2.62% of being optimal when the number of solutions is known in advance since it follows from Section 4 that *it* needs to call F only about $\frac{\pi}{8}\sqrt{N/t}$ times to succeed with probability better than 50%.

After reading an early version of this paper, Grover noticed that our lower bound would *not* apply if we were interested in the *expected* (rather than worst-case) number of calls to F necessary to succeed with probability at least 50%. A better algorithm in terms of the expected number of calls to F consists in first tossing a biased coin. With probability 40%, do nothing—and fail for sure. With probability 60%, apply $0.58278\sqrt{N/t}$ iterations of Grover's algorithm before looking at the quantum state: this will succeed with probability roughly 84.458%, as we saw in Section 4. The total expected number of iterations—and thus of calls to F—is $60\% \times 0.58278\sqrt{N/t} < 0.35\sqrt{N/t}$, which is less than $(\sin\frac{\pi}{8})\sqrt{N/t}$, yet the expected success probability is $60\% \times 84.458\%$, which is better than 50%. Nevertheless this approach *never* yields success unless F is evaluated more than $(\sin\frac{\pi}{8})\sqrt{N/t}$ times, which is why our lower bound is not contradicted by this example.

To capture the notion that F is a black box, we consider that it is given as an *oracle*. All matrices and vectors in this section are finite and complex-valued. The norm of vector $\mathbf{a}$ is denoted $\|\mathbf{a}\|$. The norm of a complex number c is denoted $|c|$.

We restate a basic fact on complex-valued vectors.

Proposition 4 *For all normalized vectors* $\mathbf{a}$ *and* $\mathbf{b}$*, and all complex scalars* α *and* β*,*

$$\|\alpha\mathbf{a} - \beta\mathbf{b}\|^2 \;\geq\; |\alpha|^2 + |\beta|^2 - 2|\alpha||\beta|.$$

The following proposition is a consequence of Chebyshev's summation inequality.

Proposition 5 *For all set of complex numbers,* $\{x_i\}_{i=0}^{r-1}$*,*

$$\left(\sum_{i=0}^{r-1} |x_i|\right)^2 \;\leq\; r\sum_{i=0}^{r-1} |x_i|^2.$$

Lemma 6 *Let* S *be any set of* N *strings, and* $\mathcal{C}$ *be any configuration space. Let* $|\phi_0\rangle$ *be any superposition, and*

$$|\phi_r\rangle = U_r \ldots U_2 U_1 |\phi_0\rangle$$

any sequence of r *unitary transforms. Let* $\{f_i\}_{i=0}^{r}$ *be any set of partial functions from* $\mathcal{C}$ *into* S*. For any* $y \in S$*, let*

$$|\phi'_r\rangle = U'_r \ldots U'_2 U'_1 |\phi_0\rangle$$

be any sequence of r *unitary transforms where for all* $i = 1, \ldots, r$*,*

$$U'_i|c\rangle = U_i|c\rangle \quad \textit{if} \quad f_{i-1}(|c\rangle) \neq y.$$

Set $|\phi'_0\rangle = |\phi_0\rangle$*, and for all* $i = 1, \ldots, r$*, set* $|\phi_i\rangle = U_i|\phi_{i-1}\rangle$ *and* $|\phi'_i\rangle = U'_i|\phi'_{i-1}\rangle$*. For all* $i = 0, 1, \ldots, r$*, set* $|\phi_i\rangle = \alpha_{i,y}|\phi_{i,y}\rangle + \alpha_{i,\overline{y}}|\phi_{i,\overline{y}}\rangle$*, where* $|\phi_{i,y}\rangle$ *(resp.* $|\phi_{i,\overline{y}}\rangle$*) is a normalized superposition of configurations where* f_i *equals (resp. does not equal)* y*. Denote* $|\phi'_i\rangle$ *similarly.*

Then the following holds:

(1)	$\||\phi'_r\rangle - |\phi_r\rangle\| \;\leq\; 2\sum_{i=0}^{r-1} |\alpha_{i,y}|$	*for all* $y \in S$.
(2)	$(1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|) \;\leq\; \||\phi'_r\rangle - |\phi_r\rangle\|^2$	*for all* $y \in S$.
(3)	$N - \sqrt{N} - \sum_{y\in S} |\alpha'_{r,\overline{y}}| \;\leq\; 2r^2$.	

PROOF. We divide the proof into three parts.

Proof of (1): For all $y \in S$ and all $i = 1, \ldots, r$ we have

$$
\begin{aligned}
U_i'|\phi_{i-1}\rangle &= U_i'\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle + \alpha_{i-1,\overline{y}}|\phi_{i-1,\overline{y}}\rangle\right)\\
&= U_i'\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right) + U_i\left(\alpha_{i-1,\overline{y}}|\phi_{i-1,\overline{y}}\rangle\right)\\
&= U_i'\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right) - U_i\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right) + U_i|\phi_{i-1}\rangle\\
&= |\phi_i\rangle + (U_i' - U_i)\left(\alpha_{i-1,y}|\phi_{i-1,y}\rangle\right).
\end{aligned}
$$

Hence, by mathematical induction on i,

$$
|\phi_i'\rangle \;=\; U_i'\dots U_1'|\phi_0\rangle \;=\; |\phi_i\rangle + \sum_{j=1}^{i}(U_i'\dots U_{j+1}')(U_j' - U_j)\left(\alpha_{j-1,y}|\phi_{j-1,y}\rangle\right),
$$

so,

$$
\begin{aligned}
\||\phi_i'\rangle - |\phi_i\rangle\| &= \|\sum_{j=1}^{i}(U_i'\dots U_{j+1}')(U_j' - U_j)\left(\alpha_{j-1,y}|\phi_{j-1,y}\rangle\right)\|\\
&\leq 2\sum_{j=1}^{i}|\alpha_{j-1,y}|,
\end{aligned}
$$

and (1) follows.

Proof of (2): The inequality follows from:

$$
\begin{aligned}
\||\phi_r'\rangle - |\phi_r\rangle\|^2 &= \|(\alpha_{r,y}'|\phi_{r,y}'\rangle + \alpha_{r,\overline{y}}'|\phi_{r,\overline{y}}'\rangle) - (\alpha_{r,y}|\phi_{r,y}\rangle + \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle)\|^2\\
&= \|(\alpha_{r,y}'|\phi_{r,y}'\rangle - \alpha_{r,y}|\phi_{r,y}\rangle) + (\alpha_{r,\overline{y}}'|\phi_{r,\overline{y}}'\rangle - \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle)\|^2\\
&= \|\alpha_{r,y}'|\phi_{r,y}'\rangle - \alpha_{r,y}|\phi_{r,y}\rangle\|^2 + \|\alpha_{r,\overline{y}}'|\phi_{r,\overline{y}}'\rangle - \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle\|^2\\
&\geq (\,|\alpha_{r,y}'|^2 + |\alpha_{r,y}|^2 - 2|\alpha_{r,y}'||\alpha_{r,y}|\,)\\
&\quad + (\,|\alpha_{r,\overline{y}}'|^2 + |\alpha_{r,\overline{y}}|^2 - 2|\alpha_{r,\overline{y}}'||\alpha_{r,\overline{y}}|\,)\\
&= 2(1 - |\alpha_{r,y}'||\alpha_{r,y}| - |\alpha_{r,\overline{y}}'||\alpha_{r,\overline{y}}|)\\
&\geq 2(1 - |\alpha_{r,y}| - |\alpha_{r,\overline{y}}'|),
\end{aligned}
$$

where the two inequalities follow from proposition 4 and the fact that the norm of any scalar is at most 1.

Proof of (3): By (2), (1), and proposition 5,

$$
1 - |\alpha_{r,y}| - |\alpha_{r,\overline{y}}'| \;\leq\; \frac{1}{2}\||\phi_r'\rangle - |\phi_r\rangle\|^2 \;\leq\; 2\left(\sum_{i=0}^{r-1}|\alpha_{i,y}|\right)^2 \;\leq\; 2r\sum_{i=0}^{r-1}|\alpha_{i,y}|^2.
$$

Thus,

$$\begin{aligned}
\sum_{y\in S}\Big(1-|\alpha_{r,y}|-|\alpha'_{r,\overline{y}}|\Big) &\le \sum_{y\in S}\left(2r\sum_{i=0}^{r-1}|\alpha_{i,y}|^2\right)\\
&= 2r\sum_{i=0}^{r-1}\left(\sum_{y\in S}|\alpha_{i,y}|^2\right) \;\le\; 2r^2.
\end{aligned}$$

Since

$$\begin{aligned}
\sum_{y\in S}\Big(1-|\alpha_{r,y}|-|\alpha'_{r,\overline{y}}|\Big) &= N-\sum_{y\in S}|\alpha_{r,y}|-\sum_{y\in S}|\alpha'_{r,\overline{y}}|\\
&\ge N-\sqrt{N}\left(\sum_{y\in S}|\alpha_{r,y}|^2\right)^{1/2}-\sum_{y\in S}|\alpha'_{r,\overline{y}}|\\
&= N-\sqrt{N}-\sum_{y\in S}|\alpha'_{r,\overline{y}}|,
\end{aligned}$$

we have

$$N-\sqrt{N}-\sum_{y\in S}|\alpha'_{r,\overline{y}}| \;\le\; \sum_{y\in S}(1-|\alpha_{r,y}|-|\alpha'_{r,\overline{y}}|) \;\le\; 2r^2,$$

and (3) follows. □

Theorem 7 *Let S be any set of N strings, and M be any oracle quantum machine with bounded error probability. Let $y \in S$ be a randomly and uniformly chosen element from S. Let F be the oracle such that $F(x)=1$ if and only if $x=y$. Then the average number of times M must query F in order to determine y with probability at least 50% is at least $\left\lfloor (\sin\frac{\pi}{8})\sqrt{N}\right\rfloor$, where the average is taken over all possible y.*

PROOF. Let S be any set of N strings and $\mathcal{C}$ be any configuration space. Let $|\psi_0\rangle$ be any superposition of configurations, and M any bounded-error oracle quantum machine. Given any oracle $F^\star$, assume that we run $M^{F^\star}$ for s steps, and assume that M queries r times its oracle $F^\star$ during the computation. Since we will only run M using oracle $F^\star$ with $F^\star(x)=0$ if $x \notin S$, without loss of generality, assume that M never queries $F^\star$ on strings not in S.

First, consider the case that we run M using the trivial oracle: let F be the oracle such that $F(x)=0$ for all $x \in S$, and let

$$|\psi_s\rangle = A_s \ldots A_1|\psi_0\rangle \tag{6}$$

be the unitary transformation corresponding to the computation of M using oracle F.

For all $i = 1, \ldots, r$, let q_i be the time stamp for M's i–th query, and set $q_{r+1} = s + 1$. Then eq. (6) can also be written as

$$|\phi_r\rangle = U_r \ldots U_1 |\phi_0\rangle \tag{7}$$

where $|\phi_0\rangle = A_{q_1-1} \ldots A_1 |\psi_0\rangle$, and for all $i = 1, \ldots, r$, $U_i = A_{q_{i+1}-1} \ldots A_{q_i}$ and $|\phi_i\rangle = U_i |\phi_{i-1}\rangle$. At the i–th query some configurations will query F, some will not. For all $i = 0, \ldots, r-1$, set $f_i(|c\rangle) = x$ if $|c\rangle$ queries F on x at the $(i+1)$–st query.

Now, consider what happens if we flip one of the oracle bits: Given any $y \in S$, let F' be the oracle such that $F'(x) = 1$ if and only if $x = y$. Then the computation of $M^{F'}$ corresponds to the unitary transformation

$$|\phi'_r\rangle = U'_r \ldots U'_1 |\phi_0\rangle$$

where $U'_i|c\rangle = U_i|c\rangle$ if $f_{i-1}(|c\rangle) \neq y$.

At the end of the computation of $M^{F'}$, we measure the superposition $|\phi'_r\rangle$ in order to determine the unknown y. For each configuration $|c\rangle \in \mathcal{C}$, set $f_r(|c\rangle) = x$ if, by measuring $|c\rangle$, M answers that x is the unknown y.

Set $|\phi'_r\rangle = \alpha'_{r,y}|\phi'_{r,y}\rangle + \alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle$ where $|\phi'_{r,y}\rangle$ (resp. $|\phi'_{r,\overline{y}}\rangle$) is the normalized superposition of configurations where f_r equals (resp. does not equal) y. Then $|\alpha'_{r,y}|^2$ is the probability that $M^{F'}$ correctly determines y. Since, by assumption, this probability is at least 50%,

$$|\alpha'_{r,\overline{y}}| \leq \frac{1}{\sqrt{2}} \quad \text{for all} \quad y \in S. \tag{8}$$

Furthermore, by Lemma 6,

$$N - \sqrt{N} - \sum_{y \in S} |\alpha'_{r,\overline{y}}| \ \leq\ 2r^2.$$

Hence, by eq. (8)

$$2r^2 \ \geq\ N - \sqrt{N} - \frac{1}{\sqrt{2}}N \ =\ \left(1 - \frac{1}{\sqrt{2}}\right) N - \sqrt{N}.$$

It follows by straightforward algebra that

$$r \ \geq\ \frac{\sqrt{2-\sqrt{2}}}{2}\sqrt{N} - 1 \ =\ (\sin\tfrac{\pi}{8})\sqrt{N} - 1 \tag{9}$$

provided $N \geq 15$. But eq. 9 holds nevertheless for all N because the oracle must be queried at least once to succeed with probability at least 50% when $N > 2$, and therefore $r \geq 1 \geq (\sin\frac{\pi}{8})\sqrt{N} - 1$ holds as required for

$2 < N < 15$. In addition, the equation holds vacuously when $N \leq 2$ since $r \geq 0 \geq (\sin \frac{\pi}{8})\sqrt{N} - 1$ in that case. The theorem follows directly from the generality of eq. 9. $\square$

Theorem 7 gives a lower bound for finding a unique solution using a bounded-error quantum machine. However, in most applications we would expect that there will be more than one solution. Furthermore, we might even not know if there is a solution at all. Let t be the number of solutions. For the case $t \geq 1$, we have the following theorem.

Theorem 8 *Let S be any set of N strings, and M be any bounded-error oracle quantum machine. Let $A \subseteq S$ be a randomly and uniformly chosen subset of S of size t, $t \geq 1$. Let F be the oracle such that $F(x) = 1$ if and only if $x \in A$. Then the average number of times M must query F in order to determine some member $y \in A$ with probability at least 50% is at least $\left\lfloor (\sin \frac{\pi}{8})\sqrt{\lfloor N/t \rfloor} \right\rfloor$, where the average is taken over all possible A of size t.*

The proof of this theorem is almost identical to the proof of Lemma 6 and Theorem 7. In Lemma 6, eqs. (1) and (2) now hold for all subsets of t strings. Hence, by choosing a largest number of such disjoint subsets from S, say R of cardinality $N_t = \lfloor N/t \rfloor$, in the proof of (3), we obtain

$$N_t - \sqrt{N_t} - \sum_{X_i \in R} |\alpha'_{r,\overline{X_i}}| \;\leq\; 2r^2.$$

The remaining part of the proof is the same as the proof of Theorem 7, only with obvious and minor changes.

8 Conclusions and Future Directions

We have provided a tight analysis of Grover's quantum search algorithm and proved that it comes to within a few percent of being optimal in terms of the number of times the function must be evaluated when it is provided as a black box (or an oracle). Moreover, we showed how to apply the algorithm even when the number of solutions is unknown ahead of time. It would be interesting to determine if in fact Grover's algorithm is exactly optimal or whether it is possible to improve it slightly. Also, a lower bound on the *expected* number of function evaluations required to find the solution by any quantum algorithm would be useful. How would it compare with our upper bound $0.69003\sqrt{N}$?

Grover's algorithm and the ideas presented in this paper can be extended in several directions, which we are currently investigating and will be the topic of a subsequent paper. In particular, Grover's algorithm can be thought of in a more general setting than quantum searching. Each iteration of the algorithm can be used to amplify the amplitude of a desired state. From this perspective, Grover's algorithm is really an *amplitude amplification* process.

It would be silly to use Grover's algorithm directly to solve most **NP**–complete problems because there are classical heuristics that would go faster on almost all instances. We are currently investigating the extent by which these heuristics can be sped up on a quantum computer by way of amplitude amplification. In many cases, we can combine the classical heuristics with amplitude amplification to allow quadratic speed up compared to the best classical heuristics available, but we do not yet know how general this phenomenon is. Similarly, more efficient quantum algorithms might exist for specific **NP**–complete problems if the structure of the problem is exploited. Furthermore, we are investigating how to use ideas from Grover's algorithm to solve problems higher than **NP** in the polynomial-time hierarchy.

Assume $F : X_N \to \{0,1\}$ is as in our paper but our goal is to determine the number t of $i \in X_N$ such that $F(i) = 1$ rather than finding a specific one. In light of the theory of **#P**–completeness, this is thought to be a harder problem for classical computers. Combining Grover's algorithm with some ideas from Shor's quantum factoring algorithm [9], we have preliminary results that indicate the possibility of solving this *quantum counting* problem with high probability in a time in $O(t\sqrt{N}\,)$ without need for a large supply of auxiliary quantum memory. If we are satisfied with an approximate answer, a time in $O(\sqrt{N}\,)$ provides an answer whose absolute error is bounded by $\sqrt{t}$ with high probability, and a time in $O(\sqrt{N/t}\,)$ suffices to count with small expected relative error.

We presented in the Section 1 an application of Grover's algorithm to the cryptanalysis of secret-key cryptosystems such as the DES. Can quantum computing be used in more subtle ways for cryptanalytical purposes, for instance when double or triple-key encipherment is used? What is the best way to use quantum searching for finding collisions in a cryptographic hash function?

Acknowledgement

We are grateful to Richard Cleve for telling us how to implement one iteration of Grover's algorithm with a single function evaluation, and to Lov Grover for pointing out that our lower bound would not apply to the *expected* number of function evaluations to succeed with a given probability. The third author would like to thank Edmund Christiansen for helpful discussions concerning recursion equations, and Joan Boyar for helpful discussions in general.

References

[1] GROVER, Lov K., "A fast quantum mechanical algorithm for database search", *Proceedings of 28th Annual ACM Symposium on Theory of Computing*, 1996, pp. 212–219.

[2] BENNETT, Charles H., Ethan BERNSTEIN, Gilles BRASSARD and Umesh VAZIRANI, "Strengths and weaknesses of quantum computing", to appear *SIAM Journal on Computing.*

[3] NATIONAL BUREAU OF STANDARDS, "Data Encryption Standard", *Federal Information Processing Standard*, U. S. Department of Commerce, FIPS PUB 46, Washington, DC, 1977.

[4] BRASSARD, Gilles, "Searching a quantum phone book", *Science*, in press, 1997.

[5] GAREY, Michael R. and David S. JOHNSON, *Computers and Intractability: A Guide to the Theory of* **NP**–*completeness*, W. H. Freeman, 1979.

[6] DEUTSCH, David and Richard JOZSA, "Rapid solution of problems by quantum computation", *Proceedings of the Royal Society, London*, Vol. A439, 1992, pp. 553–558.

[7] BARENCO, Adriano, Charles H. BENNETT, Richard CLEVE, David P. DIVINCENZO, Norman MARGOLUS, Peter SHOR, Tycho SLEATOR, John A. SMOLIN and Harald WEINFURTER, "Elementary gates for quantum computation", *Physical Review A*, Vol. 52, 1995, pp. 3457–3467.

[8] KITAEV, A. Yu., "Quantum measurements and the Abelian stabilizer problem", manuscript, 1995. Available on Los Alamos e-Print archive (`http://xxx.lanl.gov`) as quant-ph/9511026.

[9] SHOR, Peter W., "Algorithms for quantum computation: Discrete logarithms and factoring", *Proceedings of 35th Annual IEEE Symposium on Foundations of Computer Science*, 1994, pp. 124–134.