PhysComp96
Full paper
Draft, 10 May 1996

Tight bounds on quantum searching

Michel Boyer
Université de Montréal*
Gilles Brassard, FRSC†
Université de Montréal
Peter Høyer‡
Odense University§
Alain Tapp¶
Université de Montréal

quant-ph/9605034 23 May 1996

We provide a tight analysis of Grover's recent 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 analyse 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. Using techniques from Shor's quantum factoring algorithm in addition to Grover's approach, we introduce a new technique for approximate quantum counting, which allows to estimate the number of solutions. Finally we provide a lower bound on the efficiency of any possible quantum database searching algorithm and we show that Grover's algorithm nearly comes within a factor 2 of being optimal in terms of the number of probes required in the table.

1 Introduction

Assume you have a large table $T[0\,..\,N-1]$ in which you would like to find some element x. More precisely, you wish to find an integer i such that $0 \leq i < N$ and $T[i] = x$, provided such an i exists. This problem can obviously be solved in a time in $O(\log N)$ if the table is sorted, but no classical algorithm (deterministic or probabilistic) can succeed in the general case—when the elements of T are in an arbitrary order—with probability better than $1/2$, say, without probing more than half the entries of T. Grover [4] has recently discovered an algorithm for the *quantum* computer that can solve this problem in expected time in $O(\sqrt{N})$. He also remarked that a result in [1] implies that his algorithm is optimal, up to a 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 analyse the behaviour of the algorithm when the element to be found appears more than once in the table. 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. We also 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. Bringing ideas from Shor's quantum factorization algorithm [6] into Grover's algorithm, we sketch a new quantum algorithm capable of approximately *counting* the number of solutions. We also generalize Grover's algorithm in the case N is not a power of 2. Finally, we refine the argument of [1] to show that Grover's algorithm could not be improved to require much less than half the number of table lookups that it currently makes when a 50% probability of success is desired.

2 Finding a unique solution

Assume for now that there is a unique i_0 such that $T[i_0] = x$. 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 $0 \leq i < N$. (We shall never need complex amplitudes in this paper, except in §7.)

*Département IRO, C.P. 6128, succursale centre–ville, Montréal, Canada H3C 3J7. {boyer,brassard,tappa}@iro.umontreal.ca

†Supported in part by NSERC and FCAR

‡Supported in part by the ESPRIT Long Term Research Programme of the EU under project number 20244 (ALCOM-IT).

§Department of Mathematics and Computer Science, Odense University, Campusvej 55, DK–5230 Odense M, Denmark. u2pi@imada.ou.dk

¶Supported in part by NSERC

The heart of Grover's algorithm is a process, henceforth called an *iteration*, that efficiently transforms $|\Psi(k,\ell)\rangle$ into $|\Psi(\frac{N-2}{N}k+\frac{2(N-1)}{N}\ell,\frac{N-2}{N}\ell-\frac{2}{N}k)\rangle$. Although we review the iteration process in §6—where we call it G—we refer the reader to Grover's original article [4] for a more complete description and the proof that it performs as required. Grover's algorithm begins by creating an equal superposition

$$|\Psi_0\rangle = |\Psi(1/\sqrt{N},1/\sqrt{N}\,)\rangle = \sum_{i=0}^{N-1}\frac{1}{\sqrt{N}}\,|i\rangle$$

of all possible values for i, $0\le i<N$. Then some number m of iterations are performed. It is clear from the above discussion that the effect of the j–th iteration is to produce state $|\Psi_j\rangle$=$|\Psi(k_j,\ell_j)\rangle$ where $k_0=l_0=1/\sqrt{N}$ and

$$\left.\begin{array}{rcl} k_{j+1} &=& \frac{N-2}{N}k_j+\frac{2(N-1)}{N}\ell_j \\ \ell_{j+1} &=& \frac{N-2}{N}\ell_j-\frac{2}{N}k_j \end{array}\right\} \qquad (1)$$

Finally, state $|\Psi_m\rangle$ is observed, yielding some value i. The algorithm *succeeds* if and only if $T[i]=x$.

In his paper, Grover proves that there exists a number m less than $\sqrt{2N}$ such that the probability of success after m iterations is at least $1/2$. 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 in practice: 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 (1). Let angle θ be defined so that $\sin^2\theta=1/N$. It is straightforward to verify by mathematical induction that

$$\left.\begin{array}{rcl} k_j &=& \sin((2j+1)\theta) \\ \ell_j &=& \frac{1}{\sqrt{N-1}}\cos((2j+1)\theta) \end{array}\right\} \qquad (2)$$

It follows from equation (2) 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 very close to $\frac{\pi}{4}\sqrt{N}$ 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 content with a 50% probability of success, as Grover considered in his original paper [4]. 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 achieve a negligible probability of *success*!

3 The case of multiple solutions

Let us now consider the case when there are t solutions to the problem, that is there are t different values of i such that $T[i]=x$. We are interested in finding an arbitrary solution. Grover briefly considers this setting [4], but he provides no details concerning the efficiency of his method.

We assume in this section that the value of t is known. Let $A=\{i\,|\,T[i]=x\}$ and $B=\{i\,|\,T[i]\ne x\}$. 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

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

This gives rise to a recurrence similar to (1), whose solution is that the state $|\Psi(k_j,\ell_j)\rangle$ after j iterations is given by

$$\left.\begin{array}{rcl} k_j &=& \frac{1}{\sqrt{t}}\sin((2j+1)\theta) \\ \ell_j &=& \frac{1}{\sqrt{N-t}}\cos((2j+1)\theta) \end{array}\right\} \qquad (3)$$

where the angle θ is chosen so that $\sin^2\theta=t/N$.

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 $|m-\tilde{m}|\le 1/2$. 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 tells us that the optimal number of iterations is given by the j so that $4\theta j = \tan((2j+1)\theta)$.

We have not solved this equation exactly but it is very close to $z = \tan(z/2)$ with $z = 4\theta j$ when the optimal j is large, which happens when $t \ll N$. The solution for z is approximately 2.33112. 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 the case $N = 2^{20}$ 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 0.8442 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.

3.1 The case $t = N/4$

An interesting special case occurs when $t = N/4$. Of course, even a classical computer can find a solution efficiently in this case, with high probability, but not quite as efficiently as a quantum computer. Here $\sin^2\theta = t/N = 1/4$ and therefore $\theta = \pi/6$. This implies 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. Because one iteration of Grover's algorithm requires two table look-ups (including one for uncomputation purposes—see §7), this is twice as efficient (in terms of table look-ups) than the expected performance of the obvious classical probabilistic algorithm—and that's best possible classically. 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 [3].

4 Unknown number of solutions

A much more interesting case 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 above 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! In order to find a solution efficiently when their number is unknown, we need the following lemmas, the first of which is proved by straightforward algebra.

Lemma 1 *For any real numbers α and β, and any positive integer m,*

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

In particular, when $\alpha = \beta$,

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

Lemma 2 *Let t be the (unknown) number of solutions and let θ be such that $\sin^2\theta = t/N$. 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 $|\Psi_0\rangle = \sum_i \frac{1}{\sqrt{N}}|i\rangle$, the probability of obtaining a solution is exactly*

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

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. ∎

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 $1 \leq t \leq 3N/4$.

1. Initialize $m = 1$ and set $\lambda = 6/5$.
   (Any value of λ strictly between 1 and 4/3 would do.)

2. choose j uniformly at random among the nonnegative integers smaller than m.

3. Apply j iterations of Grover's algorithm starting from initial state $|\Psi_0\rangle = \sum_i \frac{1}{\sqrt{N}}|i\rangle$.

4. Observe the register: let i be the outcome.

5. If $T[i] = x$, the problem is solved: **exit**.

6. Otherwise, set m to $\min(\lambda m, \sqrt{N}\,)$ and go back to step 2.

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

Proof. Let θ be the angle so that $\sin^2\theta = t/N$. Let

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

(recall that we assumed $t \leq 3N/4$).

We shall estimate the expected number of times that a Grover iteration is performed: the total time needed is clearly in the order of that number. On the s–th time round the main loop, the value of m is λ^{s-1} and the expected number of Grover iterations is less than half that value since j is chosen randomly so that $0 \leq j < m$. We say that the algorithm reaches the *critical stage* if it goes through the main loop more than $\lceil \log_\lambda m_0 \rceil$ times. The value of m will exceed m_0 if and when the algorithm reaches that stage.

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

$$\frac{1}{2}\sum_{s=1}^{\lceil \log_\lambda m_0\rceil} \lambda^{s-1} < \frac{1}{2}\frac{\lambda}{\lambda-1}m_0 = 3m_0\,.$$

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 $m \geq 1/\sin(2\theta)$. It follows that the expected number of Grover iterations needed to succeed once the critical stage has been reached is upper-bounded by

$$\frac{1}{2}\sum_{u=0}^{\infty}\frac{3^u}{4^{u+1}}\lambda^{u+\lceil \log_\lambda m_0\rceil} < \frac{\lambda}{8-6\lambda}m_0 = \frac{3}{2}m_0\,.$$

The total expected number of Grover iterations, in case the critical stage is reached, is therefore upper-bounded by $\frac{9}{2}m_0$ and thus the total expected time is in $O(\sqrt{N/t}\,)$ provided $0 < t \leq 3N/4$. Note that $\frac{9}{2}m_0 \approx \frac{9}{4}\sqrt{N/t}$ when $t \ll N$, which is less than four 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 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. ∎

5 Quantum counting

We are currently investigating the power of quantum computers in approximately *counting* the number t of solutions, rather than merely finding one. For this, we use techniques inspired by Shor's celebrated quantum factorization algorithm [6] and combine them with Grover's algorithm. Here we sketch the basic ideas, leaving the details—many of which still have to be worked out—to a further paper [2].

Let k_j and ℓ_j be as in equation (3) and recall that $A = \{i \,|\, T[i] = x\}$ and $B = \{i \,|\, T[i] \neq x\}$. The key observation is that the value of θ, and therefore that of t, can be inferred directly from the *period* of the function that sends j onto k_j. This period can be estimated from sampling in a discrete Fourier transform of the function. In order to profit from the ability of quantum computers to compute Fourier transforms, though, we must first create a state in which the amplitude of $|j\rangle$ is proportional to k_j for values of j ranging over several periods.

Let P be a power of 2, arbitrary for the moment, and let $f = P\theta/\pi$ be the number of periods of k_j when j spans the range from 0 to $P-1$. (In general f need not be an integer.) Create state

$$|\Psi_0\rangle = \sum_{j=0}^{P-1}\sum_{i=0}^{N-1}\frac{1}{\sqrt{PN}}|j,i\rangle\,.$$

Then apply to $|\Psi_0\rangle$ a transformation that sends $|j\rangle|\Psi\rangle$ to $|j\rangle\, G^j|\Psi\rangle$, where G is the Grover iteration. This takes a time proportional to P, resulting in the state

$$\sum_{j=0}^{P-1}\left[\frac{1}{\sqrt{P}}|j\rangle\left(\sum_{i\in A}k_j|i\rangle + \sum_{i\in B}l_j|i\rangle\right)\right]\,.$$

Now, observe the second part of the register. Assume without loss of generality that some element from A is obtained. (There are no essential differences if instead an element from B is obtained since k_j and ℓ_j have exactly the same period.) At this point, the first part of the register has collapsed to state

$$\sum_{j=0}^{P-1}k_j|j\rangle$$

up to renormalization. If we apply a quantum discrete Fourier transform to this state [6] (*not* what Grover calls the quantum Fourier transform in [4]!), and if f is large enough, the amplitude of all values of j becomes vanishingly small, except for values very close to f or $P-f$. Finally, we observe the register. With high probability, this yields an excellent approximation $\tilde{f}$ on f, from which we estimate

$$\tilde{\theta} = \frac{\tilde{f}\pi}{P} \quad \text{and} \quad \tilde{t} = N \sin^2 \tilde{\theta}\,.$$

To evaluate the accuracy of $\tilde{t}$, we assume that $|f-\tilde{f}| < 1$, which happens with reasonable probability provided f is sufficiently large—see [2] for details. It follows that $|\theta - \tilde{\theta}| < \pi/P$ and therefore $|\sin\theta - \sin\tilde{\theta}| < \pi/P$ as well. From $\tilde{t} = N\sin^2\tilde{\theta}$, $t = N\sin^2\theta$ and $\sin\theta = \sqrt{t/N}$, we derive

$$|t - \tilde{t}| < \frac{2\pi}{P}\sqrt{tN} + \frac{\pi^2}{P^2}N\,. \tag{4}$$

Recall that the running time of the algorithm is proportional to P. This parameter allows us to balance the desired accuracy of the approximation with the running time required to achieve it. Let c be a constant.

- ◇ If we take $P = c\sqrt{N}$, the error in our estimate of t is bounded by $\frac{2\pi}{c}\sqrt{t} + \frac{\pi^2}{c^2}$ provided $|f-\tilde{f}| < 1$. This is reminiscent of finding the answer up to a few standard deviations.

- ◇ If we are satisfied with keeping small the *relative* error, we run the algorithm on successive powers of 2 for P until $\tilde{f}$ becomes reasonably large. This will happen when $P = c\sqrt{N/t}$. After a total time proportional to $\sqrt{N/t}$, this yields an estimate for t that is likely to be within a factor $(1+\pi/c)^2$ of the correct answer.

- ◇ If we want the *absolute* error to be probably bounded by a constant, we apply the algorithm once with $P = c\sqrt{N}$ in order to estimate t. Then, we run it again, but with $P = c\sqrt{\tilde{t}N}$. According to equation (4), and pretending $P = c\sqrt{tN}$ for simplicity, the resulting error in our second estimate of t is likely to be bounded by $\frac{2\pi}{c} + \frac{\pi^2}{c^2 t}$. In particular, we get the *exact* answer, provided $|f-\tilde{f}| < 1$, if we take $c \geq 14$ since $\frac{2\pi}{c} + \frac{\pi^2}{c^2 t} < 1/2$ in that case. (Note that successive applications of Grover's algorithm in which we strike out the solutions as they are found will also provide an exact count with high probability in a time in $O(\sqrt{tN}\,)$, but at an enormous cost in terms of additional memory—see [2].)

- ◇ Finally, we have a variation on this technique that gives the *exact* answer in a time in $O(\sqrt{N}\,)$ with a vanishingly small probability of error provided the number of solutions is a small perfect square.

We defer the details to [2].

6 Implementation considerations

Grover's algorithm consists of a number of iterations followed by a measurement. In his original article [4] Grover shows that the unitary transform G, defined below, efficiently implements what we called an iteration in §2.

For every $A \subset \mathbb{Z}_N$, let S_A be the conditional phase shift transform given by

$$S_A|i\rangle = \begin{cases} -|i\rangle & \text{if } i \in A \\ |i\rangle & \text{otherwise.} \end{cases}$$

For every $i \in \mathbb{Z}_N$, denote $S_{\{i\}}$ by S_i. Let T be the Walsh-Hadamard transform

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

where $i \cdot j$ denotes the bitwise dot product of the two strings i and j. Then the transform G is given by

$$G = -TS_0TS_{i_0}\,.$$

Grover considers only the case when N is a power of 2 since the transform T is well-defined only in this case. However, the assumption on N can be removed by observing that G is just one of many transforms that efficiently implements an iteration. Let T' be any unitary transform satisfying

$$T'|0\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle. \tag{5}$$

Then one may easily verify that the transform $T'S_0T'^{-1}S_{i_0}$ works just as well, and, more interestingly, that

$$T'S_0T'^{-1}S_A$$

implements the general iteration analysed in §3. Any transform T' satisfying (5) can thus be used in the algorithm.

When N is a power of 2, the Walsh-Hadamard transform is indeed the simplest possible choice for T'. When N is not a power of two, the approximate Fourier transform given by Kitaev [5] can be used.

7 An improved lower bound

Drawing on general results from [1], Grover points out that any algorithm for quantum database searching must take a time at least proportional to $\sqrt{N}$ when there is

a unique solution. Here we refine and generalize this result by giving an explicit lower bound on the number of table lookups required by any quantum algorithm as a function of the number of solutions. This lower bound is only a few percent smaller than the number of iterations required by Grover's algorithm when the number of solutions is known in advance. Unfortunately, each iteration of Grover's algorithm requires *two* table lookups because $T[i]$ must first be fetched (to decide on potential phase shift) and then it must be erased (to allow interference to take place) by a process often referred to as uncomputation. Therefore, we merely prove that Grover's algorithm is roughly within a factor 2 of being optimal in terms of the number of table lookups.

We rephrase the problem in terms of an oracle O defined so that $O(i) = 1$ whenever i is a solution. All matrices and vectors in this section are finite and complex-valued. Let the *inner product* $\langle \mathbf{a}, \mathbf{b} \rangle$ of two vectors $\mathbf{a}$ and $\mathbf{b}$ be defined as $\sum_i a_i^\star b_i$, where $c^\star$ denotes the complex conjugate of c. The norm of $\mathbf{a}$ is denoted $\|\mathbf{a}\|$. The absolute value 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 inequalities.

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 configurationspace. 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* $i = 1, \ldots, r$*,*

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

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* $|\phi_{i,y}\rangle$ $(|\phi_{i,\overline{y}}\rangle)$ *is the normalized superposition of configurations where* f_i *(does not) equals* 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. $2\{1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|\} \leq \| |\phi'_r\rangle - |\phi_r\rangle \|^2$ *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

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

Hence, by induction on i,

$$\begin{aligned}
&|\phi'_i\rangle = U'_i \ldots U'_1|\phi_0\rangle\\
&= |\phi_i\rangle + \textstyle\sum_{j=1}^{i}(U'_i \ldots U'_{j+1})(U'_j - U_j)\left(\alpha_{j-1,y}|\phi_{j-1,y}\rangle\right),
\end{aligned}$$

so,

$$\begin{aligned}
&\| |\phi'_i\rangle - |\phi_i\rangle \|\\
&= \| \textstyle\sum_{j=1}^{i}(U'_i \ldots U'_{j+1})(U'_j - U_j)\left(\alpha_{j-1,y}|\phi_{j-1,y}\rangle\right) \|\\
&\leq 2\textstyle\sum_{j=1}^{i} |\alpha_{j-1,y}|,
\end{aligned}$$

and (1) follows.
Proof of (2): The identity follows from:

$$\begin{aligned}
&\| |\phi'_r\rangle - |\phi_r\rangle \|\\
&= \|(\alpha'_{r,y}|\phi'_{r,y}\rangle + \alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle)\\
&\quad - (\alpha_{r,y}|\phi_{r,y}\rangle + \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle)\|\\
&= \|(\alpha'_{r,y}|\phi'_{r,y}\rangle - \alpha_{r,y}|\phi_{r,y}\rangle)\\
&\quad + (\alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle - \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle)\|\\
&= \left\{\|\alpha'_{r,y}|\phi'_{r,y}\rangle - \alpha_{r,y}|\phi_{r,y}\rangle\|^2\right.\\
&\quad \left. + \|\alpha'_{r,\overline{y}}|\phi'_{r,\overline{y}}\rangle - \alpha_{r,\overline{y}}|\phi_{r,\overline{y}}\rangle\|^2\right\}^{1/2}\\
&\geq \left\{(|\alpha'_{r,y}|^2 + |\alpha_{r,y}|^2 - 2|\alpha'_{r,y}||\alpha_{r,y}|)\right.\\
&\quad \left. + (|\alpha'_{r,\overline{y}}|^2 + |\alpha_{r,\overline{y}}|^2 - 2|\alpha'_{r,\overline{y}}||\alpha_{r,\overline{y}}|)\right\}^{1/2}\\
&= \left\{2 - 2(|\alpha'_{r,y}||\alpha_{r,y}| + |\alpha'_{r,\overline{y}}||\alpha_{r,\overline{y}}|)\right\}^{1/2}\\
&= \sqrt{2}\left\{1 - |\alpha'_{r,y}||\alpha_{r,y}| - |\alpha'_{r,\overline{y}}||\alpha_{r,\overline{y}}|\right\}^{1/2}\\
&\geq \sqrt{2}\left\{1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}|\right\}^{1/2},
\end{aligned}$$

where the two inequalities follow from proposition 4 and the fact that the absolute value of any scalar is at most one.
Proof of (3): By (2), (1), and proposition 5,

$$\begin{aligned}
1 - |\alpha_{r,y}| - |\alpha'_{r,\overline{y}}| &\leq \frac{1}{2}\| |\phi'_r\rangle - |\phi_r\rangle \|^2\\
&\leq 2\left(\sum_{i=0}^{r-1} |\alpha_{i,y}|\right)^2 \leq 2r\sum_{i=0}^{r-1} |\alpha_{i,y}|^2.
\end{aligned}$$

Thus,

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

Since,

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

we have

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

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 $y\in_R S$ be a randomly and uniformly chosen element from S. Put O to be the oracle where $O(x)=1$ if and only if $x=y$. Then the expected number of times M must query O in order to determine y with probability at least $1/2$ is at least $\lfloor(\sin(\pi/8))\sqrt{N}\rfloor$.*

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

First, consider the case that we run M using the trivial oracle: let O be the oracle where $O(x)=0$ 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 O.

For all $i=1,\ldots,r$, set q_i to be the timestamp for M's i'th query, and set $q_{r+1}=s+1$. Then (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 O, some will not. For all $i=0,\ldots,r-1$, set $f_i(|c\rangle)=x$ if $|c\rangle$ queries x at the $i+1$'th query.

Now, consider what happens if we flip one bit of the oracle bits: Given any $y\in S$, let O' be the oracle where $O'(x)=1$ if and only if $x=y$. Then the computation of $M^{O'}$ 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 $M^{O'}$, 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 $|\phi'_{r,y}\rangle$ ($|\phi'_{r,\overline{y}}\rangle$) is the normalized superposition of configurations where f_r (does not) equals y. Then $|\alpha'_{r,y}|^2$ is the probability that $M^{O'}$ correctly determines y. Since, by assumption, this probability is at least $1/2$,

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

Furthermore, by Lemma 6,

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

Hence,

$$\begin{aligned}2r^2 &\geq N-\sqrt{N}-\sum_{y\in S}|\alpha'_{r,\overline{y}}|\\ &\geq N-\sqrt{N}-\frac{1}{\sqrt{2}}N\\ &= (1-\frac{1}{\sqrt{2}})N-\sqrt{N},\end{aligned}$$

so

$$\begin{aligned}r &\geq \left\{(2-\sqrt{2})\frac{N}{4}-\frac{\sqrt{N}}{2}\right\}^{1/2}\\ &= \left\{2-\sqrt{2}-\frac{2}{\sqrt{N}}\right\}^{1/2}\frac{\sqrt{N}}{2}\\ &> \left\{\sqrt{2-\sqrt{2}}-\frac{2}{\sqrt{N}}\right\}\frac{\sqrt{N}}{2}\\ &= \frac{\sqrt{2-\sqrt{2}}}{2}\sqrt{N}-1\\ &= (\sin(\pi/8))\sqrt{N}-1,\end{aligned}$$

which proves the theorem. ∎

Theorem 7 gives a lower bound for finding a unique feasible $y\in S$ using a bounded-error quantum machine. However, in most applications we would expect that there will be more than one feasible y, say t such y's. Furthermore, we might even not know if there is a feasible y or not. For the case $t\geq 1$, we have:

Theorem 8 *Let S be any set of N strings, and M be any bounded-error oracle quantum machine. Let $A\subset_R S$ be a randomly and uniformly chosen subset of S of size t, $t\geq 1$. Put O to be the oracle where $O(x)=1$ if and only if $x\in A$. Then the expected number of times M must query O in order to determine some member $y\in A$ with probability at least $1/2$ is at least $\lfloor(\sin(\pi/8))\sqrt{\lfloor N/t\rfloor}\rfloor$.*

The proof of this theorem is almost identical to the proof of Lemma 6 and Theorem 7. In Lemma 6, equations (1) and (2) now hold for all subsets of t strings. Hence, by choosing a largest number of such disjoint subsets from S, say $T = \{X_1, \ldots, X_{N_t}\}$ where $N_t = \lfloor N/t \rfloor$, in the proof of (3), we obtain

$$N_t - \sqrt{N_t} - \sum_{X_i \in T} |\alpha'_{r,\overline{X_i}}| \leq 2r^2.$$

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

Acknowledgements

We are grateful to Umesh and Vijay Vazirani for discussions concerning classical approximate counting. 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] Bennett, Charles H., Ethan Bernstein, Gilles Brassard and Umesh Vazirani, "Strengths and weaknesses of quantum computing", manuscript (1995).

[2] Brassard, Gilles and Alain Tapp, "Approximate quantum counting", in preparation (1996).

[3] Deutsch, David and Richard Jozsa, "Rapid solution of problems by quantum computation", *Proceedings of the Royal Society, London* **A439** (1992), 553–558.

[4] Grover, Lov K., "A fast quantum mechanical algorithm for database search", *Proceedings of the 28th Annual ACM Symposium on Theory of Computing* (1996).

[5] Kitaev, A. Yu., "Quantum measurements and the Abelian stabilizer problem", manuscript quant-ph/9511-026 (1995).

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