Quantum Amplitude Amplification and Estimation

Gilles Brassard [*] Peter Høyer [†]
Michele Mosca [‡] Alain Tapp [§]

2 May 2000

Abstract

Consider a Boolean function $\chi : X \rightarrow \{0,1\}$ that partitions set X between its *good* and *bad* elements, where x is good if $\chi(x) = 1$ and bad otherwise. Consider also a quantum algorithm $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sum_{x\in X} \alpha_x |x\rangle$ is a quantum superposition of the elements of X, and let a denote the probability that a good element is produced if $\mathcal{A}|0\rangle$ is measured. If we repeat the process of running $\mathcal{A}$, measuring the output, and using χ to check the validity of the result, we shall expect to repeat $1/a$ times on the average before a solution is found. *Amplitude amplification* is a process that allows to find a good x after an expected number of applications of $\mathcal{A}$ and its inverse which is proportional to $1/\sqrt{a}$, assuming algorithm $\mathcal{A}$ makes no measurements.

[*] Département IRO, Université de Montréal, C.P. 6128, succursale centre-ville, Montréal (Québec), Canada H3C 3J7. email: brassard@iro.umontreal.ca. Supported in part by Canada's NSERC and Québec's FCAR.

[†] BRICS, Department of Computer Science, University of Aarhus, Ny Munkegade, Bldg. 540, DK-8000 Aarhus C, Denmark. email: hoyer@brics.dk. Part of this work was done while at Département IRO, Université de Montréal. Basic Research in Computer Science is supported by the Danish National Research Foundation.

[‡] CACR, Department of C&O, Faculty of Mathematics, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1. email: mmosca@cacr.math.uwaterloo.ca. Most of this work was done while at Centre for Quantum Computation, Clarendon Laboratory, University of Oxford. Supported in part by Canada's NSERC and UK's CESG.

[§] CACR, email: atapp@cacr.math.uwaterloo.ca. Most of this work was done while at Département IRO, Université de Montréal. Supported in part by a postdoctoral fellowship from Canada's NSERC.

This is a generalization of Grover's searching algorithm in which $\mathcal{A}$ was restricted to producing an equal superposition of all members of X and we had a promise that a single x existed such that $\chi(x) = 1$. Our algorithm works whether or not the value of a is known ahead of time. In case the value of a is known, we can find a good x after a number of applications of $\mathcal{A}$ and its inverse which is proportional to $1/\sqrt{a}$ even in the worst case. We show that this quadratic speedup can also be obtained for a large family of search problems for which good classical heuristics exist. Finally, as our main result, we combine ideas from Grover's and Shor's quantum algorithms to perform *amplitude estimation*, a process that allows to estimate the value of a. We apply amplitude estimation to the problem of *approximate counting*, in which we wish to estimate the number of $x \in X$ such that $\chi(x) = 1$. We obtain optimal quantum algorithms in a variety of settings.

Keywords: *Quantum computation. Searching. Counting. Lower bound.*

1 Introduction

Quantum computing is a field at the junction of theoretical modern physics and theoretical computer science. Practical experiments involving a few quantum bits have been successfully performed, and much progress has been achieved in quantum information theory, quantum error correction and fault tolerant quantum computation. Although we are still far from having desktop quantum computers in our offices, the quantum computational paradigm could soon be more than mere theoretical exercise.

The discovery by Peter Shor [15] of a polynomial-time quantum algorithm for factoring and computing discrete logarithms was a major milestone in the history of quantum computing. Another significant result is Lov Grover's quantum search algorithm [8, 9]. Grover's algorithm does not solve **NP**–complete problems in polynomial time, but the wide range of its applications more than compensates for this.

In this paper, we generalize Grover's algorithm in a variety of directions. Consider a problem that is characterized by a Boolean function $\chi(x, y)$ in the sense that y is a good solution to instance x if and only if $\chi(x, y) = 1$. (There could be more than one good solution to a given instance.) If we have a probabilistic algorithm $\mathcal{P}$ that outputs a guess $\mathcal{P}(x)$ on input x, we can call $\mathcal{P}$ and χ repeatedly until a solution to instance x is found. If $\chi(x, \mathcal{P}(x)) = 1$

with probability $p_x > 0$, we expect to repeat this process $1/p_x$ times on the average. Consider now the case when we have a quantum algorithm $\mathcal{A}$ instead of the probabilistic algorithm. Assume $\mathcal{A}$ makes no measurements: instead of a classical answer, it produces quantum superposition $|\Psi_x\rangle$ when run on input x. Let a_x denote the probability that $|\Psi_x\rangle$, *if measured*, would be a good solution. If we repeat the process of running $\mathcal{A}$ on x, measuring the output, and using χ to check the validity of the result, we shall expect to repeat $1/a_x$ times on the average before a solution is found. This is no better than the classical probabilistic paradigm.

In Section 2, we describe a more efficient approach to this problem, which we call amplitude amplification. Intuitively, the probabilistic paradigm increases the probability of success roughly by a constant on each iteration; by contrast, amplitude amplification increases the *amplitude* of success roughly by a constant on each iteration. Because amplitudes correspond to square roots of probabilities, it suffices to repeat the amplitude amplification process approximately $1/\sqrt{a_x}$ times to achieve success with overwhelming probability. For simplicity, we assume in the rest of this paper that there is a single instance for which we seek a good solution, which allows us to dispense with input x, but the generalization to the paradigm outlined above is straightforward. Grover's original database searching quantum algorithm is a special case of this process, in which χ is given by a function $f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$ for which we are promised that there exists a unique x_0 such that $f(x_0) = 1$. If we use the Fourier transform as quantum algorithm $\mathcal{A}$—or more simply the Walsh–Hadamard transform in case N is a power of 2—an equal superposition of all possible x's is produced, whose success probability would be $1/N$ if measured. Classical repetition would succeed after an expected number N of evaluations of f. Amplitude amplification corresponds to Grover's algorithm: it succeeds after approximately $\sqrt{N}$ evaluations of the function.

We generalize this result further to the case when the probability of success a of algorithm $\mathcal{A}$ is not known ahead of time: it remains sufficient to evaluate $\mathcal{A}$ and χ an expected number of times that is proportional to $1/\sqrt{a}$. Moreover, in the case a is known ahead of time, we give two different techniques that are guaranteed to find a good solution after a number of iterations that is proportional to $1/\sqrt{a}$ in the worst case.

It can be proven that Grover's algorithm goes quadratically faster than any possible classical algorithm when function f is given as a black box. However, it is usually the case in practice that information is known about f

that allows us to solve the problem much more efficiently than by exhaustive search. The use of classical *heuristics*, in particular, will often yield a solution significantly more efficiently than straight quantum amplitude amplification would. In Section 3, we consider a broad class of classical heuristics and show how to apply amplitude amplification to obtain quadratic speedup compared to any such heuristic.

Finally, Section 4 addresses the question of estimating the success probability a of quantum algorithm $\mathcal{A}$. We call this process *amplitude estimation*. As a special case of our main result (Theorem 12), an estimate for a is obtained after any number M of iterations which is within $2\pi\sqrt{a(1-a)}/M + \pi^2/M^2$ of the correct value with probability at least $8/\pi^2$, where one iteration consists of running algorithm $\mathcal{A}$ once forwards and once backwards, and of computing function χ once. As an application of this technique, we show how to approximately count the number of x such that $f(x) = 1$ given a function $f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$. If the correct answer is $t > 0$, it suffices to compute the function $\sqrt{N}$ times to obtain an estimate roughly within $\sqrt{t}$ of the correct answer. A number of evaluations of f proportional to $\frac{1}{\varepsilon}\sqrt{N/t}$ yields a result that is likely to be within εt of the correct answer. (We can do slightly better in case ε is not fixed.) If it is known ahead of time that the correct answer is either $t = 0$ or $t = t_0$ for some fixed t_0, we can determine which is the case with certainty using a number of evaluations of f proportional to $\sqrt{N/t_0}$. If we have no prior knowledge about t, the exact count can be obtained with high probability after a number of evaluations of f that is proportional to $\sqrt{t(N-t)}$ when $0 < t < N$ and $\sqrt{N}$ otherwise. Most of these results are optimal.

We assume in this paper that the reader is familiar with basic notions of quantum computing.

2 Quantum amplitude amplification

Suppose we have a classical randomized algorithm that succeeds with some probability p. If we repeat the algorithm, say, j times, then our probability of success increases to roughly jp (assuming $jp \ll 1$). Intuitively, we can think of this strategy as each additional run of the given algorithm boosting the probability of success by an additive amount of roughly p.

A quantum analogue of boosting the probability of success would be to boost the *amplitude* of being in a certain subspace of a Hilbert space. The general concept of amplifying the amplitude of a subspace was discovered by Brassard and Høyer [4] as a generalization of the boosting technique applied by Grover in his original quantum searching paper [8]. Following [4] and [3], we refer to their idea as *amplitude amplification* and detail the ingredients below.

Let $\mathcal{H}$ denote the Hilbert space representing the state space of a quantum system. Every Boolean function $\chi : \mathbb{Z} \to \{0,1\}$ induces a partition of $\mathcal{H}$ into a direct sum of two subspaces, a good subspace and a bad subspace. The *good subspace* is the subspace spanned by the set of basis states $|x\rangle \in \mathcal{H}$ for which $\chi(x) = 1$, and the *bad subspace* is its orthogonal complement in $\mathcal{H}$. We say that the elements of the good subspace are *good*, and that the elements of the bad subspace are *bad*.

Every pure state $|\Upsilon\rangle$ in $\mathcal{H}$ has a unique decomposition as $|\Upsilon\rangle = |\Upsilon_1\rangle + |\Upsilon_0\rangle$, where $|\Upsilon_1\rangle$ denotes the projection onto the good subspace, and $|\Upsilon_0\rangle$ denotes the projection onto the bad subspace. Let $a_\Upsilon = \langle\Upsilon_1|\Upsilon_1\rangle$ denote the probability that measuring $|\Upsilon\rangle$ produces a good state, and similarly, let $b_\Upsilon = \langle\Upsilon_0|\Upsilon_0\rangle$. Since $|\Upsilon_1\rangle$ and $|\Upsilon_0\rangle$ are orthogonal, we have $a_\Upsilon + b_\Upsilon = 1$.

Let $\mathcal{A}$ be any quantum algorithm that acts on $\mathcal{H}$ and uses no measurements. Let $|\Psi\rangle = \mathcal{A}|0\rangle$ denote the state obtained by applying $\mathcal{A}$ to the initial zero state. The amplification process is realized by repeatedly applying the following unitary operator [4] on the state $|\Psi\rangle$,

$$\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi) = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_\chi. \tag{1}$$

Here, the operator $\mathbf{S}_\chi$ conditionally changes the sign of the amplitudes of the good states,

$$|x\rangle \longmapsto \begin{cases} -|x\rangle & \text{if } \chi(x) = 1 \\ |x\rangle & \text{if } \chi(x) = 0, \end{cases}$$

while the operator $\mathbf{S}_0$ changes the sign of the amplitude if and only if the state is the zero state $|0\rangle$. The operator $\mathbf{Q}$ is well-defined since we assume that $\mathcal{A}$ uses no measurements and, therefore, $\mathcal{A}$ has an inverse.

The usefulness of operator $\mathbf{Q}$ stems from its simple action on the subspace $\mathcal{H}_\Psi$ spanned by the vectors $|\Psi_1\rangle$ and $|\Psi_0\rangle$.

Lemma 1 *We have that*

$$\begin{aligned} \mathbf{Q}|\Psi_1\rangle &= (1-2a)|\Psi_1\rangle - 2a|\Psi_0\rangle \\ \mathbf{Q}|\Psi_0\rangle &= 2(1-a)|\Psi_1\rangle + (1-2a)|\Psi_0\rangle, \end{aligned}$$

where $a = \langle \Psi_1 | \Psi_1 \rangle$.

It follows that the subspace $\mathcal{H}_\Psi$ is stable under the action of $\mathbf{Q}$, a property that was first observed by Brassard and Høyer [4] and rediscovered by Grover [10].

Suppose $0 < a < 1$. Then $\mathcal{H}_\Psi$ is a subspace of dimension 2, and otherwise $\mathcal{H}_\Psi$ has dimension 1. The action of $\mathbf{Q}$ on $\mathcal{H}_\Psi$ is also realized by the operator

$$\mathbf{U}_\Psi \mathbf{U}_{\Psi_0}, \tag{2}$$

which is composed of 2 reflections. The first operator, $\mathbf{U}_{\Psi_0} = \mathbf{I} - \frac{2}{1-a} |\Psi_0\rangle\langle\Psi_0|$, implements a reflection through the ray spanned by the vector $|\Psi_0\rangle$, while the second operator $\mathbf{U}_\Psi = \mathbf{I} - 2|\Psi\rangle\langle\Psi|$ implements a reflection through the ray spanned by the vector $|\Psi\rangle$.

Consider the orthogonal complement $\mathcal{H}_\Psi^\perp$ of $\mathcal{H}_\Psi$ in $\mathcal{H}$. Since the operator $\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}$ acts as the identity on $\mathcal{H}_\Psi^\perp$, operator $\mathbf{Q}$ acts as $-\mathbf{S}_\chi$ on $\mathcal{H}_\Psi^\perp$. Thus, $\mathbf{Q}^2$ acts as the identity on $\mathcal{H}_\Psi^\perp$, and every eigenvector of $\mathbf{Q}$ in $\mathcal{H}_\Psi^\perp$ has eigenvalue $+1$ or -1. It follows that to understand the action of $\mathbf{Q}$ on an arbitrary initial vector $|\Upsilon\rangle$ in $\mathcal{H}$, it suffices to consider the action of $\mathbf{Q}$ on the projection of $|\Upsilon\rangle$ onto $\mathcal{H}_\Psi$.

Since operator $\mathbf{Q}$ is unitary, the subspace $\mathcal{H}_\Psi$ has an orthonormal basis consisting of two eigenvectors of $\mathbf{Q}$,

$$|\Psi_\pm\rangle = \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{a}} |\Psi_1\rangle \pm \frac{\imath}{\sqrt{1-a}} |\Psi_0\rangle \right), \tag{3}$$

provided $0 < a < 1$, where $\imath = \sqrt{-1}$ denotes the principal square root of -1. The corresponding eigenvalues are

$$\lambda_\pm = e^{\pm \imath 2 \theta_a}, \tag{4}$$

where the angle θ_a is defined so that

$$\sin^2(\theta_a) = a = \langle \Psi_1 | \Psi_1 \rangle \tag{5}$$

and $0 \le \theta_a \le \pi/2$.

We use operator $\mathbf{Q}$ to boost the success probability a of the quantum algorithm $\mathcal{A}$. First, express $|\Psi\rangle = \mathcal{A}|0\rangle$ in the eigenvector basis,

$$\mathcal{A}|0\rangle = |\Psi\rangle = \frac{-\imath}{\sqrt{2}} \left(e^{\imath \theta_a} |\Psi_+\rangle - e^{-\imath \theta_a} |\Psi_-\rangle \right). \tag{6}$$

It is now immediate that after j applications of operator $\mathbf{Q}$, the state is

$$\mathbf{Q}^j|\Psi\rangle = \frac{-\imath}{\sqrt{2}}\left(e^{(2j+1)\imath\theta_a}|\Psi_+\rangle - e^{-(2j+1)\imath\theta_a}|\Psi_-\rangle\right) \tag{7}$$

$$= \frac{1}{\sqrt{a}}\sin((2j+1)\theta_a)\,|\Psi_1\rangle + \frac{1}{\sqrt{1-a}}\cos((2j+1)\theta_a)\,|\Psi_0\rangle. \tag{8}$$

It follows that if $0 < a < 1$ and if we compute $\mathbf{Q}^m|\Psi\rangle$ for some integer $m \geq 0$, then a final measurement will produce a good state with probability equal to $\sin^2((2m+1)\theta_a)$.

If the initial success probability a is either 0 or 1, then the subspace $\mathcal{H}_\Psi$ spanned by $|\Psi_1\rangle$ and $|\Psi_0\rangle$ has dimension 1 only, but the conclusion remains the same: If we measure the system after m rounds of amplitude amplification, then the outcome is good with probability $\sin^2((2m+1)\theta_a)$, where the angle θ_a is defined so that Equation 5 is satisfied and so that $0 \leq \theta_a \leq \pi/2$.

Therefore, assuming $a > 0$, to obtain a high probability of success, we want to choose integer m such that $\sin^2((2m+1)\theta_a)$ is close to 1. Unfortunately, our ability to choose m wisely depends on our knowledge about θ_a, which itself depends on a. The two extreme cases are when we know the exact value of a, and when we have no prior knowledge about a whatsoever.

Suppose the value of a is known. If $a > 0$, then by letting $m = \lfloor \pi/4\theta_a \rfloor$, we have that $\sin^2((2m+1)\theta_a) \geq 1 - a$, as shown in [3]. The next theorem is immediate.

Theorem 2 (Quadratic speedup) *Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0,1\}$ be any Boolean function. Let a the initial success probability of $\mathcal{A}$. Suppose $a > 0$, and set $m = \lfloor \pi/4\theta_a \rfloor$, where θ_a is defined so that $\sin^2(\theta_a) = a$ and $0 < \theta_a \leq \pi/2$. Then, if we compute $\mathbf{Q}^m\mathcal{A}\,|0\rangle$ and measure the system, the outcome is good with probability at least* $\max(1-a, a)$.

Note that any implementation of algorithm $\mathbf{Q}^m\mathcal{A}|0\rangle$ requires that the value of a is known so that the value of m can be computed. We refer to Theorem 2 as a quadratic speedup, or the square-root running-time result. The reason for this is that if an algorithm $\mathcal{A}$ has success probability $a > 0$, then after an expected number of $1/a$ applications of $\mathcal{A}$, we will find a good solution. Applying the above theorem reduces this to an expected number of at most $(2m+1)/\max(1-a, a) \in \Theta(\frac{1}{\sqrt{a}})$ applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$.

As an application of Theorem 2, consider the search problem [9] in which we are given a Boolean function $f : \{0, 1, \dots, N-1\} \to \{0, 1\}$ satisfying the promise that there exists a unique $x_0 \in \{0, 1, \dots, N-1\}$ on which f takes value 1, and we are asked to find x_0. If f is given as a black box, then on a classical computer, we need to evaluate f on an expected number of roughly half the elements of the domain in order to determine x_0.

By contrast, Grover [9] discovered a quantum algorithm that only requires an expected number of evaluations of f in the order of $\sqrt{N}$. In terms of amplitude amplification, Grover's algorithm reads as follows: Let $\chi = f$, and let $\mathcal{A} = \mathbf{W}$ be the Walsh-Hadamard transform on n qubits that maps the initial zero state $|0\rangle$ to $\frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle$, an equally-weighted superposition of all $N = 2^n$ elements in the domain of f. Then the operator $\mathbf{Q} = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_\chi$ is equal to the iterate $-\mathbf{W}\mathbf{S}_0\mathbf{W}\mathbf{S}_f$ applied by Grover in his searching paper [9]. The initial success probability a of $\mathcal{A}$ is exactly $1/N$, and if we measure after $m = \lfloor \pi/4\theta_a \rfloor$ iterations of $\mathbf{Q}$, the probability of measuring x_0 is lower bounded by $1 - 1/N$ [3].

Now, suppose that the value of a is not known. In Section 4, we discuss techniques for finding an estimate of a, whereafter one then can apply a weakened version of Theorem 2 in which the exact value of a is replaced by an estimate of it. Another idea is to try to find a good solution without prior computation of an estimate of a. Within that approach, by adapting the ideas in Section 6 in [3] we can still obtain a quadratic speedup.

Theorem 3 (Quadratic speedup without knowing a) *There exists a quantum algorithm* **QSearch** *with the following property. Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0, 1\}$ be any Boolean function. Let a denote the initial success probability of $\mathcal{A}$. Algorithm* **QSearch** *finds a good solution using an expected number of applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ which are in $\Theta(\frac{1}{\sqrt{a}})$ if $a > 0$, and otherwise runs forever.*

The algorithm in the above theorem utilizes the given quantum algorithm $\mathcal{A}$ as a subroutine and the operator $\mathbf{Q}$. The complete algorithm is as follows:

Algorithm(QSearch($\mathcal{A}, \chi$))

1. Set $l = 0$ and let c be any constant such that $1 < c < 2$.

2. Increase l by 1 and set $M = \lceil c^l \rceil$.

3. Apply $\mathcal{A}$ on the initial state $|0\rangle$, and measure the system. If the outcome $|z\rangle$ is good, that is, if $\chi(z) = 1$, then output z and stop.

4. Initialize a register of appropriate size to the state $\mathcal{A}|0\rangle$.

5. Pick an integer j between 1 and M uniformly at random.

6. Apply $\mathbf{Q}^j$ to the register, where $\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi)$.

7. Measure the register. If the outcome $|z\rangle$ is good, then output z and stop. Otherwise, go to step 2.

The intuition behind this algorithm is as follows. In a 2-dimensional real vector space, if we pick a unit vector $(x, y) = (\cos(\cdot), \sin(\cdot))$ uniformly at random then the expected value of y^2 is $1/2$. Consider Equation 8. If we pick j at random between 1 and M for some integer M such that $M\theta_a$ is larger than, say, 100π, then we have a good approximation to a random unit vector, and we will succeed with probability close to $1/2$.

To turn this intuition into an algorithm, the only obstacle left is that we do not know the value of θ_a, and hence do not know an appropriate value for M. However, we can overcome this by using exponentially increasing values of M, an idea similar to the one used in "exponential searching" (which is a term that does not refer to the running time of the method, but rather to an exponentially increasing growth of the size of the search space).

The correctness of algorithm **QSearch** is immediate and thus to prove the theorem, it suffices to show that the expected number of applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ is in the order of $1/\sqrt{a}$. This can be proven by essentially the same techniques applied in the proof of Theorem 3 in [3] and we therefore only give a very brief sketch of the proof.

On the one hand, if the initial success probability a is at least $3/4$, then step 3 ensures that we soon will measure a good solution. On the other hand, if $0 < a < 3/4$ then, for any given value of M, the probability of measuring a good solution in step 7 is lower bounded by

$$\frac{1}{2}\left(1 - \frac{1}{2M\sqrt{a}}\right). \tag{9}$$

Let $c_0 > 0$ be such that $c = 2(1 - c_0)$ and let $M_0 = 1/(2c_0\sqrt{a}\,)$. The expected number of applications of $\mathcal{A}$ is upper bounded by $T_1 + T_2$, where T_1 denotes the maximum number of applications of $\mathcal{A}$ the algorithm uses while $M < M_0$, and where T_2 denotes the expected number of applications of $\mathcal{A}$ the algorithm uses while $M \geq M_0$. Clearly $T_1 \in O(M_0) = O(\frac{1}{\sqrt{a}})$ and we now show that $T_2 \in O(\frac{1}{\sqrt{a}})$ as well.

For all $M \geq M_0$, the measurement in step 7 yields a good solution with probability at least $\frac{1}{2}(1 - c_0)$, and hence it fails to yield a good solution with probability at most $p_0 = \frac{1}{2}(1 + c_0)$. Thus for all $i \geq 0$, with probability at most p_0^i, we have that $M \geq M_0 c^i$ at some point after step 2 while running the algorithm. Hence T_2 is at most on the order of $\sum_{i \geq 0} M_0 (cp_0)^i$ which is in $O(M_0)$ since $cp_0 < 1$. The total expected number of applications of $\mathcal{A}$ is thus in $O(M_0)$, which is $O(\frac{1}{\sqrt{a}})$.

For the lower bound, if M were in $o\big(\frac{1}{\sqrt{a}}\big)$, then the probability that we measure a good solution in step 7 would be vanishingly small. This completes our sketch of the proof of Theorem 3.

2.1 Quantum de-randomization when the success probability is known

We now consider the situation where the success probability a of the quantum algorithm $\mathcal{A}$ is known. If $a = 0$ or $a = 1$, then amplitude amplification will not change the success probability, so in the rest of this section, we assume that $0 < a < 1$. Theorem 2 allows us to boost the probability of success to at least $\max(1 - a, a)$. A natural question to ask is whether it is possible to improve this to certainty, still given the value of a. It turns out that the answer is positive. This is unlike classical computers, where no such general de-randomization technique is known. We now describe 2 optimal methods for obtaining this, but other approaches are possible.

The first method is by applying amplitude amplification, not on the original algorithm $\mathcal{A}$, but on a slightly modified version of it. By Equation 8, if we measure the state $\mathbf{Q}^m \mathcal{A}|0\rangle$, then the outcome is good with probability $\sin^2((2m + 1)\theta_a)$. In particular, if $\tilde{m} = \pi/4\theta_a - 1/2$ happens to be an integer, then we would succeed with certainty after $\tilde{m}$ applications of $\mathbf{Q}$. In general, $\overline{m} = \lceil \tilde{m} \rceil$ iterations is a fraction of 1 iteration too many, but we can compensate for that by choosing $\overline{\theta_a} = \pi/(4\overline{m} + 2)$, an angle slightly smaller than θ_a. Any quantum algorithm that succeeds with probability $\overline{a}$ such that $\sin^2(\overline{\theta_a}) = \overline{a}$, will succeed with certainty after $\overline{m}$ iterations of amplitude amplification. Given $\mathcal{A}$ and its initial success probability a, it is easy to construct a new quantum algorithm that succeeds with probability $\overline{a} \leq a$:

Let $\mathcal{B}$ denote the quantum algorithm that takes a single qubit in the initial state $|0\rangle$ and rotates it to the superposition $\sqrt{1-\overline{a}/a}\,|0\rangle + \sqrt{\overline{a}/a}\,|1\rangle$. Apply both $\mathcal{A}$ and $\mathcal{B}$, and define a good solution as one in which $\mathcal{A}$ produces a good solution, and the outcome of $\mathcal{B}$ is the state $|1\rangle$. Theorem 4 follows.

Theorem 4 (Quadratic speedup with known a) *Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0,1\}$ be any Boolean function. There exists a quantum algorithm that given the initial success probability $a > 0$ of $\mathcal{A}$, finds a good solution with certainty using a number of applications of $\mathcal{A}$ and $\mathcal{A}^{-1}$ which is in $\Theta(\frac{1}{\sqrt{a}})$ in the worst case.*

The second method to obtain success probability 1 requires a generalization of operator $\mathbf{Q}$. Given angles $0 \leq \phi, \varphi < 2\pi$, redefine $\mathbf{Q}$ as follows,

$$\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi) = -\mathcal{A}\mathbf{S}_0(\phi)\mathcal{A}^{-1}\mathbf{S}_\chi(\varphi). \tag{10}$$

Here, the operator $\mathbf{S}_\chi(\varphi)$ is the natural generalization of the $\mathbf{S}_\chi$ operator,

$$|x\rangle \;\longmapsto\; \begin{cases} e^{\imath\varphi}|x\rangle & \text{if } \chi(x) = 1 \\ |x\rangle & \text{if } \chi(x) = 0. \end{cases}$$

Similarly, the operator $\mathbf{S}_0(\phi)$ multiplies the amplitude by a factor of $e^{\imath\phi}$ if and only if the state is the zero state $|0\rangle$. The action of operator $\mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$ is also realized by applying an operator that is composed of two pseudo-reflections: the operator $\mathcal{A}\mathbf{S}_0(\phi)\mathcal{A}^{-1}$ and the operator $-\mathbf{S}_\chi(\varphi)$.

The next lemma shows that the subspace $\mathcal{H}_\Psi$ spanned by $|\Psi_1\rangle$ and $|\Psi_0\rangle$ is stable under the action of $\mathbf{Q}$, just as in the special case $\mathbf{Q}(\mathcal{A}, \chi, \pi, \pi)$ studied above.

Lemma 5 *Let $\mathbf{Q} = \mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$. Then*

$$\begin{aligned} \mathbf{Q}|\Psi_1\rangle &= e^{\imath\varphi}((1-e^{\imath\phi})a - 1)|\Psi_1\rangle + e^{\imath\varphi}(1-e^{\imath\phi})a|\Psi_0\rangle \\ \mathbf{Q}|\Psi_0\rangle &= (1-e^{\imath\phi})(1-a)|\Psi_1\rangle - ((1-e^{\imath\phi})a + e^{\imath\phi})|\Psi_0\rangle, \end{aligned}$$

where $a = \langle\Psi_1|\Psi_1\rangle$.

Let $\tilde{m} = \pi/4\theta_a - 1/2$, and suppose that $\tilde{m}$ is not an integer. In the second method to obtain a good solution with certainty, we also apply $\lceil\tilde{m}\rceil$ iterations of amplitude amplification, but now we slow down the speed of the very last iteration only, as opposed to of all iterations as in the first method. For the case $\tilde{m} < 1$, this second method has also been suggested by Chi and Kim [6]. We start by applying the operator $\mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$ with $\phi = \varphi = \pi$ a number of $\lfloor \tilde{m} \rfloor$ times to the initial state $|\Psi\rangle = \mathcal{A}|0\rangle$. By Equation 8, this produces the superposition

$$\frac{1}{\sqrt{a}} \sin\big((2\lfloor \tilde{m} \rfloor + 1)\theta_a\big)\, |\Psi_1\rangle + \frac{1}{\sqrt{1-a}} \cos\big((2\lfloor \tilde{m} \rfloor + 1)\theta_a\big)\, |\Psi_0\rangle.$$

Then, we apply operator $\mathbf{Q}$ one more time, but now using angles ϕ and φ, both between 0 and 2π, satisfying

$$\begin{aligned} & e^{\imath\varphi}(1 - e^{\imath\phi})\sqrt{a} \sin\big((2\lfloor \tilde{m} \rfloor + 1)\theta_a\big) \\ & \qquad\qquad = ((1 - e^{\imath\phi})a + e^{\imath\phi}) \frac{1}{\sqrt{1-a}} \cos\big((2\lfloor \tilde{m} \rfloor + 1)\theta_a\big). \qquad (11) \end{aligned}$$

By Lemma 5, this ensures that the resulting superposition has inner product zero with $|\Psi_0\rangle$, and thus a subsequent measurement will yield a good solution with certainty.

The problem of choosing $\phi, \varphi \in \mathbb{R}$ such that Equation 11 holds is equivalent to requiring that

$$\cot\big((2\lfloor \tilde{m} \rfloor + 1)\theta_a\big) = e^{\imath\varphi} \sin(2\theta_a)\big(-\cos(2\theta_a) + \imath \cot(\phi/2)\big)^{-1}. \qquad (12)$$

By appropriate choices of ϕ and φ, the right hand side of Equation 12 can be made equal to any nonzero complex number of norm at most $\tan(2\theta_a)$. Thus, since the left hand side of this equation is equal to some real number smaller than $\tan(2\theta_a)$, there exist $\phi, \varphi \in \mathbb{R}$ such that Equation 12 is satisfied, and hence also such that the expression in Equation 11 vanishes. In conclusion, applying $\mathbf{Q}(\mathcal{A}, \chi, \phi, \varphi)$ with such $\phi, \varphi \in \mathbb{R}$ at the very last iteration allows us to measure a good solution with certainty.

3 Heuristics

As explained in the previous section, using the amplitude amplification technique to search for a solution to a search problem, one obtains a quadratic speedup compared to a brute force search. For many problems, however, good heuristics are known for which the expected running time, when applied to a "real-life" problem, is in $o(\sqrt{N})$, where N is the size of the search space. This fact would make amplitude amplification much less useful unless a quantum computer is somehow able to take advantage of these classical heuristics. In this section we concentrate on a large family of classical heuristics that can be applied to search problems. We show how these heuristics can be incorporated into the general amplitude amplification process.

By a heuristic, we mean a probabilistic algorithm, running in polynomial time, that outputs what one is searching for with some non-negligible probability.

Suppose we have a family $\mathcal{F}$ of functions such that each $f \in \mathcal{F}$ is of the form $f : X \rightarrow \{0,1\}$. For a given function f we seek an input $x \in X$ such that $f(x) = 1$. A *heuristic* is a function $G : \mathcal{F} \times R \rightarrow X$, for an appropriate finite set R. The heuristic G uses a random seed $r \in R$ to generate a guess for an x such that $f(x) = 1$. For every function $f \in \mathcal{F}$, let $t_f = |\{x \in X \mid f(x) = 1\}|$, the number of good inputs x, and let $h_f = |\{r \in R \mid f(G(f,r)) = 1\}|$, the number of good seeds. We say that the heuristic is *efficient* for a given f if $h_f/|R| > t_f/|X|$, that is, if using G and a random seed to generate inputs to f succeeds with a higher probability than directly guessing inputs to f uniformly at random. The heuristic is *good* in general if

$$\mathrm{E}_{\mathcal{F}}\left(\frac{h_f}{|R|}\right) \; > \; \mathrm{E}_{\mathcal{F}}\left(\frac{t_f}{|X|}\right) \; .$$

Here $\mathrm{E}_{\mathcal{F}}$ denotes the expectation over all f according to some fixed distribution. Note that for some f, h_f might be small but repeated uses of the heuristic, with seeds uniformly chosen in R, will increase the probability of finding a solution.

Theorem 6 *Let $\mathcal{F} \subseteq \{f \mid f : X \rightarrow \{0,1\}\}$ be a family of Boolean functions and $\mathcal{D}$ be a probability distribution over $\mathcal{F}$. If on a classical computer, using heuristic $G : \mathcal{F} \times R \rightarrow X$, one finds $x_0 \in X$ such that $f(x_0) = 1$ for random f taken from distribution D in expected time T then using a quantum computer, a solution can be found in expected time in $O(\sqrt{T}\,)$.*

Proof A simple solution to this problem is to embed the classical heuristic G into the function used in the algorithm **QSearch**. Let $\chi(r) = f(G(f,r))$ and $x = G(f, \mathbf{QSearch}(\mathbf{W}, \chi))$, so that $f(x) = 1$. By Theorem 3, for each function $f \in \mathcal{F}$, we have an expected running time in $\Theta(\sqrt{|R|/h_f}\,)$. Let P_f denote the probability that f occurs. Then $\sum_{f \in \mathcal{F}} P_f = 1$, and we have that the expected running time is in the order of $\sum_{f \in \mathcal{F}} \sqrt{|R|/h_f}\, P_f$, which can be rewritten as

$$\sum_{f \in \mathcal{F}} \sqrt{\frac{|R|}{h_f} P_f} \sqrt{P_f} \leq \left(\sum_{f \in \mathcal{F}} \frac{|R|}{h_f} P_f \right)^{1/2} \left(\sum_{f \in \mathcal{F}} P_f \right)^{1/2} = \left(\sum_{f \in \mathcal{F}} \frac{|R|}{h_f} P_f \right)^{1/2}$$

by Cauchy–Schwarz's inequality. □

An alternative way to prove Theorem 6 is to incorporate the heuristic into the operator $\mathcal{A}$ and do a minor modification to f. Let $\mathcal{A}$ be the quantum implementation of G. It is required that the operator $\mathcal{A}$ be unitary, but clearly in general the classical heuristic does not need to be reversible. As usual in quantum algorithms one will need first to modify the heuristic $G : \mathcal{F} \times R \to X$ to make it reversible, which can be done efficiently using standard techniques [2]. We obtain a reversible function $G'_f : R \times \mathbf{0} \to R \times X$. Let $\mathcal{A}$ be the natural unitary operation implementing G'_f and let us modify χ (the good set membership function) to consider only the second part of the register, that is $\chi((r, x)) = 1$ if and only if $f(x) = 1$. We then have that $a = h_f/|R|$ and by Theorem 3, for each function $f \in \mathcal{F}$, we have an expected running time in $\Theta(\sqrt{|R|/h_f}\,)$. The rest of the reasoning is similar. This alternative technique shows, using a simple example, the usefulness of the general scheme of amplitude amplification described in the preceding section, although it is clear that from a computational point of view this is strictly equivalent to the technique given in the earlier proof of the theorem.

4 Quantum amplitude estimation

Section 2 dealt in a very general way with combinatorial search problems, namely, given a Boolean function $f : X \to \{0, 1\}$ find an $x \in X$ such that $f(x) = 1$. In this section, we deal with the related problem of estimating $t = |\{x \in X \mid f(x) = 1\}|$, the number of inputs on which f takes the value 1.

We can describe this counting problem in terms of amplitude estimation. Using the notation of Section 2, given a unitary transformation $\mathcal{A}$ and a Boolean function χ, let $|\Psi\rangle = \mathcal{A}|0\rangle$. Write $|\Psi\rangle = |\Psi_1\rangle + |\Psi_0\rangle$ as a superposition of the good and bad components of $|\Psi\rangle$. Then *amplitude estimation* is the problem of estimating $a = \langle \Psi_1 | \Psi_1 \rangle$, the probability that a measurement of $|\Psi\rangle$ yields a good state.

The problem of estimating $t = |\{x \in X \mid f(x) = 1\}|$ can be formulated in these terms as follows. For simplicity, we take $X = \{0, 1, \ldots, N-1\}$. If N is a power of 2, then we set $\chi = f$ and $\mathcal{A} = \mathbf{W}$. If N is not a power of 2, we set $\chi = f$ and $\mathcal{A} = \mathbf{F}_N$, the quantum Fourier transform which, for every integer $M \geq 1$, is defined by

$$\mathbf{F}_M \ : \ |x\rangle \ \longmapsto \ \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{2\pi \imath xy/M} |y\rangle \qquad (0 \leq x < M). \tag{13}$$

Then in both cases we have $a = t/N$, and thus an estimate for a directly translates into an estimate for t.

To estimate a, we make good use of the properties of operator $\mathbf{Q} = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_f$. By Equation 8 in Section 2, we have that the amplitudes of $|\Psi_1\rangle$ and $|\Psi_0\rangle$ as functions of the number of applications of $\mathbf{Q}$, are sinusoidal functions, both of period $\frac{\pi}{\theta_a}$. Recall that $0 \leq \theta_a \leq \pi/2$ and $a = \sin^2(\theta_a)$, and thus an estimate for θ_a also gives an estimate for a.

To estimate this period, it is a natural approach [5] to apply Fourier analysis like Shor [15] does for a classical function in his factoring algorithm. This approach can also be viewed as an eigenvalue estimation [12, 7] and is best analysed in the basis of eigenvectors of the operator at hand [13]. By Equation 4, the eigenvalues of $\mathbf{Q}$ on the subspace spanned by $|\Psi_1\rangle$ and $|\Psi_0\rangle$ are $\lambda_+ = e^{\imath 2\theta_a}$ and $\lambda_- = e^{-\imath 2\theta_a}$. Thus we can estimate a simply by estimating one of these two eigenvalues. Errors in our estimate $\tilde{\theta}_a$ for θ_a translate into errors in our estimate $\tilde{a} = \sin^2(\tilde{\theta}_a)$ for a, as described in the next lemma.

Lemma 7 *Let* $a = \sin^2(\theta_a)$ *and* $\tilde{a} = \sin^2(\tilde{\theta}_a)$ *with* $0 \leq \theta_a, \tilde{\theta}_a \leq 2\pi$ *then*

$$\left|\tilde{\theta}_a - \theta_a\right| \leq \varepsilon \ \Rightarrow \ |\tilde{a} - a| \leq 2\varepsilon\sqrt{a(1-a)} + \varepsilon^2 \,.$$

Proof For $\varepsilon \geq 0$, using standard trigonometric identities, we obtain

$$\begin{aligned} \sin^2(\theta_a + \varepsilon) - \sin^2(\theta_a) &= \sqrt{a(1-a)}\sin(2\varepsilon) + (1-2a)\sin^2(\varepsilon) \text{ and} \\ \sin^2(\theta_a) - \sin^2(\theta_a - \varepsilon) &= \sqrt{a(1-a)}\sin(2\varepsilon) + (2a-1)\sin^2(\varepsilon). \end{aligned}$$

The inequality follows directly. □

We want to estimate one of the eigenvalues of $\mathbf{Q}$. For this purpose, we utilize the following operator Λ. For any positive integer M and any unitary operator $\mathbf{U}$, the operator $\Lambda_M(\mathbf{U})$ is defined by

$$|j\rangle|y\rangle \ \longmapsto \ |j\rangle(\mathbf{U}^j|y\rangle) \qquad (0 \leq j < M). \tag{14}$$

Note that if $|\Phi\rangle$ is an eigenvector of $\mathbf{U}$ with eigenvalue $e^{2\pi\imath\omega}$, then $\Lambda_M(\mathbf{U})$ maps $|j\rangle|\Phi\rangle$ to $e^{2\pi\imath\omega j}|j\rangle|\Phi\rangle$.

Definition 8 *For any integer $M > 0$ and real number $0 \leq \omega < 1$, let*

$$|\mathcal{S}_M(\omega)\rangle = \frac{1}{\sqrt{M}} \sum_{y=0}^{M-1} e^{2\pi\imath\omega y}\,|y\rangle.$$

We then have, for all $0 \leq x \leq M-1$

$$\mathbf{F}_M\,|x\rangle = |\mathcal{S}_M(x/M)\rangle.$$

The state $|\mathcal{S}_M(\omega)\rangle$ encodes the angle $2\pi\omega$ ($0 \leq \omega < 1$) in the phases of an equally weighted superposition of all basis states. Different angles have different encodings, and the overlap between $|\mathcal{S}_M(\omega_0)\rangle$ and $|\mathcal{S}_M(\omega_1)\rangle$ is a measure for the distance between the two angles ω_0 and ω_1.

Definition 9 *For any two real numbers $\omega_0, \omega_1 \in \mathbb{R}$, let $d(\omega_0,\omega_1) = \min_{z\in\mathbb{Z}}\{|z+\omega_1-\omega_0|\}$.*

Thus $2\pi d(\omega_0,\omega_1)$ is the length of the shortest arc on the unit circle going from $e^{2\pi\imath\omega_0}$ to $e^{2\pi\imath\omega_1}$.

Lemma 10 *For $0 \leq \omega_0 < 1$ and $0 \leq \omega_1 < 1$ let $\Delta = d(\omega_0,\omega_1)$. If $\Delta = 0$ we have $|\langle\mathcal{S}_M(\omega_0)|\mathcal{S}_M(\omega_1)\rangle|^2 = 1$. Otherwise*

$$|\langle\mathcal{S}_M(\omega_0)|\mathcal{S}_M(\omega_1)\rangle|^2 = \frac{\sin^2(M\Delta\pi)}{M^2\sin^2(\Delta\pi)}.$$

Proof

$$\begin{aligned} |\langle\mathcal{S}_M(\omega_0)|\mathcal{S}_M(\omega_1)\rangle|^2 &= \left|\left(\frac{1}{\sqrt{M}}\sum_{y=0}^{M-1} e^{-2\pi\imath\omega_0 y}\langle y|\right)\left(\frac{1}{\sqrt{M}}\sum_{y=0}^{M-1} e^{2\pi\imath\omega_1 y}|y\rangle\right)\right|^2 \\ &= \frac{1}{M^2}\left|\sum_{y=0}^{M-1} e^{2\pi\imath\Delta y}\right|^2 = \frac{\sin^2(M\Delta\pi)}{M^2\sin^2(\Delta\pi)}. \end{aligned}$$

□

Consider the problem of estimating ω where $0 \leq \omega < 1$, given the state $|\mathcal{S}_M(\omega)\rangle$. If $\omega = x/M$ for some integer $0 \leq x < M$, then $\mathbf{F}_M^{-1}|\mathcal{S}_M(x/M)\rangle = |x\rangle$ by definition, and thus we have a perfect phase estimator. If $M\omega$ is not an integer, then observing $\mathbf{F}_M^{-1}|\mathcal{S}_M(\omega)\rangle$ still provides a good estimation of ω, as shown in the following theorem.

Theorem 11 *Let X be the discrete random variable corresponding to the classical result of measuring $\mathbf{F}_M^{-1}|\mathcal{S}_M(\omega)\rangle$ in the computational basis. If $M\omega$ is an integer then* $\mathrm{Prob}(X = M\omega) = 1$. *Otherwise, letting $\Delta = d(\omega, x/M)$,*

$$\mathrm{Prob}(X = x) \;=\; \frac{\sin^2(M\Delta\pi)}{M^2\sin^2(\Delta\pi)} \;\leq\; \frac{1}{(2M\Delta)^2}\,.$$

For any $k > 1$ we also have

$$\mathrm{Prob}\left(d(X/M, \omega) \leq k/M\right) \;\geq\; 1 - \frac{1}{2(k-1)}$$

and, in the case $k = 1$ and $M > 2$,

$$\mathrm{Prob}\left(d(X/M, \omega) \leq 1/M\right) \;\geq\; \frac{8}{\pi^2}\,.$$

Proof Clearly

$$\begin{aligned}\mathrm{Prob}(X = x) &= \left|\langle x|\mathbf{F}^{-1}|\mathcal{S}_M(\omega)\rangle\right|^2 \\ &= \left|(\mathbf{F}|x\rangle)^{\dagger}|\mathcal{S}_M(\omega)\rangle\right|^2 \\ &= |\langle\mathcal{S}_M(x/M)|\mathcal{S}_M(\omega)\rangle|^2\end{aligned}$$

thus using Lemma 10 we directly obtain the first part of the theorem. We use this fact to prove the next part of the theorem.

$$\begin{aligned}\mathrm{Prob}\left(d(X/M, \omega) \leq k/M\right) &= 1 - \mathrm{Prob}(d(X/M, \omega) > k/M) \\ &\geq 1 - 2\sum_{j=k}^{\infty} \frac{1}{4M^2(\frac{j}{M})^2} \\ &\geq 1 - \frac{1}{2(k-1)}.\end{aligned}$$

For the last part, we use the fact that for $M > 2$, the given expression attains its minimum at $\Delta = 1/(2M)$ in the range $0 \leq \Delta \leq 1/M$.

$$\begin{aligned}\mathrm{Prob}\left(d(X/M,\omega) \leq 1/M\right) &= \mathrm{Prob}(X = \lfloor M\omega \rfloor) + \mathrm{Prob}(X = \lceil M\omega \rceil) \\ &= \frac{\sin^2(M\Delta\pi)}{M^2\sin^2(\Delta\pi)} + \frac{\sin^2(M(\frac{1}{M}-\Delta)\pi)}{M^2\sin^2((\frac{1}{M}-\Delta)\pi)} \\ &\geq \frac{8}{\pi^2}.\end{aligned}$$

□

The following algorithm computes an estimate for a, via an estimate for θ_a.

Algorithm(Est_Amp$(\mathcal{A}, \chi, M)$ **)**

1. Initialize two registers of appropriate sizes to the state $|0\rangle\mathcal{A}|0\rangle$.
2. Apply $\mathbf{F}_M$ to the first register.
3. Apply $\Lambda_M(\mathbf{Q})$ where $\mathbf{Q} = -\mathcal{A}\mathbf{S}_0\mathcal{A}^{-1}\mathbf{S}_\chi$.
4. Apply $\mathbf{F}_M^{-1}$ to the first register.
5. Measure the first register and denote the outcome $|y\rangle$.
6. Output $\tilde{a} = \sin^2(\pi\frac{y}{M})$.

Steps 1 to 5 are illustrated on Figure 1. This algorithm can also be summarized, following the approach in [11], as the unitary transformation

$$\Big((\mathbf{F}_M^{-1} \otimes \mathbf{I})\ \Lambda_M(\mathbf{Q})\ (\mathbf{F}_M \otimes \mathbf{I})\Big)$$

applied on state $|0\rangle\mathcal{A}|0\rangle$, followed by a measurement of the first register and classical post-processing of the outcome. In practice, we could choose M to be a power of 2, which would allow us to use a Walsh–Hadamard transform instead of a Fourier transform in step 2.

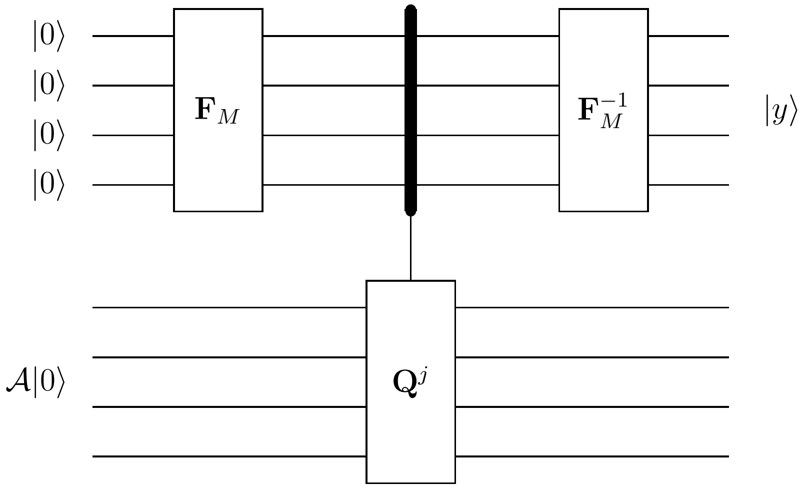

Figure 1: Quantum circuit for amplitude estimation.

Theorem 12 (Amplitude Estimation) *For any positive integer k, the algorithm* $\textbf{Est\_Amp}(\mathcal{A}, \chi, M)$ *outputs $\tilde{a}$ $(0 \leq \tilde{a} \leq 1)$ such that*

$$|\tilde{a} - a| \;\leq\; 2\pi k \frac{\sqrt{a(1-a)}}{M} + k^2 \frac{\pi^2}{M^2}$$

with probability at least $\frac{8}{\pi^2}$ when $k = 1$ and with probability greater than $1 - \frac{1}{2(k-1)}$ for $k \geq 2$. It uses exactly M evaluations of f. If $a = 0$ then $\tilde{a} = 0$ with certainty, and if $a = 1$ and M is even, then $\tilde{a} = 1$ with certainty.

Proof After step 1, by Equation 6, we have state

$$|0\rangle\mathcal{A}|0\rangle \;=\; \frac{-\imath}{\sqrt{2}}|0\rangle \left(e^{\imath\theta_a}|\Psi_+\rangle \,-\, e^{-\imath\theta_a}|\Psi_-\rangle\right).$$

After step 2, ignoring global phase, we have

$$\frac{1}{\sqrt{2M}} \sum_{j=0}^{M-1} |j\rangle \left(e^{\imath\theta_a}|\Psi_+\rangle \,-\, e^{-\imath\theta_a}|\Psi_-\rangle\right)$$

and after applying $\Lambda_M(\mathbf{Q})$ we have

$$\begin{aligned}
&\frac{1}{\sqrt{2M}}\sum_{j=0}^{M-1}|j\rangle\left(e^{\imath\theta_a}e^{2\imath j\theta_a}|\Psi_+\rangle \;-\; e^{-\imath\theta_a}e^{-2\imath j\theta_a}|\Psi_-\rangle\right)\\
=\;&\frac{e^{\imath\theta_a}}{\sqrt{2M}}\sum_{j=0}^{M-1}e^{2\imath j\theta_a}|j\rangle|\Psi_+\rangle \;-\; \frac{e^{-\imath\theta_a}}{\sqrt{2M}}\sum_{j=0}^{M-1}e^{-2\imath j\theta_a}|j\rangle|\Psi_-\rangle\\
=\;&\frac{e^{\imath\theta_a}}{\sqrt{2}}|\mathcal{S}_M(\tfrac{\theta_a}{\pi})\rangle|\Psi_+\rangle \;-\; \frac{e^{-\imath\theta_a}}{\sqrt{2}}|\mathcal{S}_M(1-\tfrac{\theta_a}{\pi})\rangle|\Psi_-\rangle.
\end{aligned}$$

We then apply $\mathbf{F}_M^{-1}$ to the first register and measure it in the computational basis.

The rest of the proof follows from Theorem 11. Tracing out the second register in the eigenvector basis, we see that the first register is in an equally weighted mixture of $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$ and $\mathbf{F}_M^{-1}|\mathcal{S}_M(1-\frac{\theta_a}{\pi})\rangle$. Thus the measured value $|y\rangle$ is the result of measuring either the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$ or the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(1-\frac{\theta_a}{\pi})\rangle$. The probability of measuring $|y\rangle$ given the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(1-\frac{\theta_a}{\pi})\rangle$ is equal to the probability of measuring $|M-y\rangle$ given the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$. Since $\sin^2\left(\pi\frac{(M-y)}{M}\right) = \sin^2\left(\pi\frac{y}{M}\right)$, we can assume we measured $|y\rangle$ given the state $\mathbf{F}_M^{-1}|\mathcal{S}_M(\frac{\theta_a}{\pi})\rangle$ and $\tilde{\theta}_a = \pi\frac{y}{M}$ estimates θ_a as described in Theorem 11. Thus we obtain bounds on $d(\tilde{\theta}_a,\theta_a)$ that translate, using Lemma 7, into the appropriate bounds on $|\tilde{a}-a|$. □

A straightforward application of this algorithm is to approximately count the number of solutions t to $f(x)=1$. To do this we simply set $\mathcal{A}=\mathbf{W}$ if N is a power of 2, or in general $\mathcal{A}=\mathbf{F}_N$ or any other transformation that maps $|0\rangle$ to $\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle$. Setting $\chi=f$, we then have $a=\langle\Psi_1|\Psi_1\rangle=t/N$, which suggests the following algorithm.

Algorithm(Count(f,M) **)**

1. Output $t' = N \times$ **Est_Amp**$(\mathbf{F}_N, f, M)$.

By Theorem 12, we obtain the following.

Theorem 13 (Counting) *For any positive integers M and k, and any Boolean function $f : \{0, 1, \dots , N-1\} \to \{0, 1\}$, the algorithm* $\mathbf{Count}(f, M)$ *outputs an estimate t' to $t = |f^{-1}(1)|$ such that*

$$\bigl|t' - t\bigr| \;\le\; 2\pi k \frac{\sqrt{t(N-t)}}{M} + \pi^2 k^2 \frac{N}{M^2}$$

with probability at least $8/\pi^2$ when $k = 1$, and with probability greater than $1 - \frac{1}{2(k-1)}$ for $k \ge 2$. If $t = 0$ then $t' = 0$ with certainty, and if $t = N$ and M is even, then $t' = N$ with certainty.

Note that $\mathbf{Count}(f, M)$ outputs a real number. In the following counting algorithms we will wish to output an integer, and therefore we will round off the output of **Count** to an integer. To assure that the rounding off can be done efficiently[1] we will round off to an integer $\tilde{t}$ satisfying $\bigl|\tilde{t} - \mathbf{Count}(f, M)\bigr| \le \frac{2}{3}$.

If we want to estimate t within a few standard deviations, we can apply algorithm **Count** with $M = \lceil \sqrt{N}\, \rceil$.

Corollary 14 *Given a Boolean function $f : \{0, 1, \dots , N-1\} \to \{0, 1\}$ with t defined as above, rounding off the output of* $\mathbf{Count}\bigl(f, \lceil \sqrt{N}\, \rceil\bigr)$ *gives an estimate $\tilde{t}$ such that*

$$\bigl|\tilde{t} - t\bigr| \;<\; 2\pi \sqrt{\frac{t(N-t)}{N}} + 11 \tag{15}$$

with probability at least $8/\pi^2$ and requires exactly $\lceil \sqrt{N}\, \rceil$ evaluations of f.

We now look at the case of estimating t with some relative error, also referred to as *approximately counting t with accuracy ε*. For this we require the following crucial observation about the output t' of algorithm $\mathbf{Count}(f, L)$. Namely t' is likely to be equal to zero if and only if $L \in o(\sqrt{N/t})$. Thus, we can find a rough estimate of $\sqrt{N/t}$ simply by running algorithm $\mathbf{Count}(f, L)$ with exponentially increasing values of L until we obtain a non-zero output. Having this rough estimate L of $\sqrt{N/t}$ we can then apply Theorem 13 with M in the order of $\frac{1}{\varepsilon}L$ to find an estimate $\tilde{t}$ of t with the required accuracy. The precise algorithm is as follows.

[1] For example, if $t' + \frac{1}{2}$ is super-exponentially close to an integer n we may not be able to decide efficiently if t' is closer to n or $n - 1$.

Algorithm(Basic_Approx_Count(f, ε) **)**

```
1. Start with $\ell = 0$.
2. Increase $\ell$ by 1.
3. Set $t' = \mathbf{Count}(f, 2^\ell)$.
4. If $t' = 0$ and $2^\ell < 2\sqrt{N}$ then go to step 2.
5. Set $M = \left\lceil \frac{20\pi^2}{\varepsilon} 2^\ell \right\rceil$.
6. Set $t' = \mathbf{Count}(f, M)$.
7. Output an integer $\tilde{t}$ satisfying $\left|\tilde{t} - t'\right| \leq \frac{2}{3}$.
```

Theorem 15 *Given a Boolean function f with N and t defined as above, and any $0 < \varepsilon \leq 1$,* **Basic_Approx_Count**(f, ε) *outputs an estimate $\tilde{t}$ such that*

$$\left|\tilde{t} - t\right| \;\leq\; \varepsilon t$$

with probability at least $\frac{2}{3}$, using an expected number of evaluations of f which is in $\Theta\left(\frac{1}{\varepsilon}\sqrt{N/t}\right)$. If $t = 0$, the algorithm outputs $\tilde{t} = t$ with certainty and f is evaluated a number of times in $\Theta\left(\sqrt{N}\right)$.

Proof When $t = 0$, the analysis is straightforward. For $t > 0$, let θ denote $\theta_{t/N}$ and $m = \left\lfloor \log_2(\frac{1}{5\theta}) \right\rfloor$. From Theorem 11 we have that the probability that step 3 outputs $\mathbf{Count}(f, 2^\ell) = 0$ for $\ell = 1, 2, \ldots, m$ is

$$\prod_{\ell=1}^{m} \frac{\sin^2(2^\ell\theta)}{2^{2\ell}\sin^2(\theta)} \;\geq\; \prod_{\ell=1}^{m} \cos^2(2^\ell\theta) \;=\; \frac{\sin^2(2^{m+1}\theta)}{2^{2m}\sin^2(2\theta)} \;\geq\; \cos^2\left(\tfrac{2}{5}\right).$$

The previous inequalities are obtained by using the fact that $\sin(M\theta) \geq M\sin(\theta)\cos(M\theta)$ for any $M \geq 0$ and $0 \leq M\theta < \frac{\pi}{2}$, which can be readily seen by considering the Taylor expansion of $\tan(x)$ at $x = M\theta$.

Now assuming step 3 has outputted 0 at least m times (note that $2^m \leq \frac{1}{5\theta} \leq \frac{1}{5}\sqrt{N/t} < 2\sqrt{N}$), after step 5 we have $M \geq \frac{20\pi^2}{\varepsilon} 2^{m+1} \geq \frac{4\pi^2}{\varepsilon\theta}$ and by Theorem 13 (and the fact that $\theta \leq \frac{\pi}{2}\sin(\theta) = \frac{\pi}{2}\sqrt{t/N}$) the probability that **Count**(f, M) outputs an integer t' satisfying $|t' - t| \leq \frac{\varepsilon}{4}t + \frac{\varepsilon^2}{64}t$ is at least $8/\pi^2$. Let us suppose this is the case. If $\varepsilon t < 1$, then $|\tilde{t} - t| < 1$ and, since $\tilde{t}$ and t are both integers, we must have $t = \tilde{t}$. If $\varepsilon t \geq 1$, then rounding off t' to $\tilde{t}$ introduces an error of at most $\frac{2}{3} \leq \frac{2\varepsilon}{3}t$, making the total error at most $\frac{\varepsilon}{4}t + \frac{\varepsilon^2}{64}t + \frac{2\varepsilon}{3}t < \varepsilon t$. Therefore the overall probability of outputting an estimate with error at most εt is at least $\cos^2\left(\frac{2}{5}\right) \times (8/\pi^2) > \frac{2}{3}$.

To upper bound the number of applications of f, note that by Theorem 13, for any integer $L \geq 18\pi\sqrt{N/t}$, the probability that **Count**(f, L) outputs 0 is less than 1/4. Thus the expected value of M at step 6 is in $\Theta(\frac{1}{\varepsilon}\sqrt{N/t})$. □

We remark that in algorithm **Basic_Approx_Count**, we could alternatively to steps 1 to 4 use algorithm **QSearch** of Section 2, provided we have **QSearch** also output its final value of M. In this case, we would use (a multiple of) that value as our rough estimate of $\sqrt{N/t}$, instead of using the final value of 2^ℓ found in step 4 of **Basic_Approx_Count**.

Algorithm **Basic_Approx_Count** is optimal for any fixed ε, but not in general. In Appendix A we give an optimal algorithm, while we now present two simple optimal algorithms for counting the number of solutions exactly. That is, we now consider the problem of determining the exact value of $t = |f^{-1}(-1)|$. In the special case that we are given a nonzero integer t_0 and promised that either $t = 0$ or $t = t_0$, then we can determine which is the case with certainty using a number of evaluations of f in $O(\sqrt{N/t_0})$. This is an easy corollary of Theorem 4 and we state it without proof.

Theorem 16 *Let $f : \{0, 1, \ldots, N-1\} \to \{0, 1\}$ be a given Boolean function such that the cardinality of the preimage of 1 is either 0 or t_0. Then there exists a quantum algorithm that determines with certainty which is the case using a number of evaluations of f which is in $\Theta\big(\sqrt{N/t_0}\big)$, and in the latter case, also outputs a random element of $f^{-1}(1)$.*

For the general case in which we do not have any prior knowledge about t, we offer the following algorithm.

Algorithm(Exact_Count(f))

```
1. Set $t'_1 = \mathbf{Count}\big(f, \lceil 14\pi\sqrt{N}\,\rceil\big)$ and $t'_2 = \mathbf{Count}\big(f, \lceil 14\pi\sqrt{N}\,\rceil\big)$.

2. Let $M_i = \lceil 30\sqrt{(t'_i + 1)(N - t'_i + 1)}\,\rceil$ for $i = 1, 2$.
```

3. Set $M = \min\{M_1, M_2\}$.

4. Set $t' = \mathbf{Count}(f, M)$.

5. Output an integer $\tilde{t}$ satisfying $\left|\tilde{t} - t'\right| \leq \frac{2}{3}$.

The main idea of this algorithm is the same as that of algorithm **Basic_Approx_Count**. First we find a rough estimate t'_r of t, and then we run algorithm $\mathbf{Count}(f, M)$ with a value of M that depends on t'_r. By Theorem 13, if we set M to be in the order of $\sqrt{t'_r(N - t'_r)}$, then the output $t' = \mathbf{Count}(f, M)$ is likely to be so that $|t' - t| < \frac{1}{3}$, in which case $\tilde{t} = t$.

Theorem 17 *Given a Boolean function f with N and t defined as above, algorithm* **Exact_Count** *requires an expected number of evaluations of f which is in $\Theta(\sqrt{(t+1)(N-t+1)}\,)$ and outputs an estimate $\tilde{t}$ which equals t with probability at least $\frac{2}{3}$ using space only linear in* $\log(N)$.

Proof Apply Theorem 13 with $k = 7$. For each $i = 1, 2$, with probability greater than $\frac{11}{12}$, outcome t'_i satisfies $\left|t'_i - t\right| < \sqrt{\frac{t(N-t)}{N}} + 1/4$, in which case we also have that $\sqrt{t(N-t)} \leq \frac{\sqrt{2}}{30} M_i$. Thus, with probability greater than $\left(\frac{11}{12}\right)^2$, we have

$$\frac{\sqrt{t(N-t)}}{M} \;\leq\; \frac{\sqrt{2}}{30}.$$

Suppose this is the case. Then by Theorem 13, with probability at least $8/\pi^2$,

$$|t' - t| \;\leq\; \frac{2\pi\sqrt{2}}{30} + \frac{4\pi^2}{30^2} \;<\; \frac{1}{3}$$

and consequently

$$|\tilde{t} - t| < 1.$$

Hence, with probability at least $\left(\frac{11}{12}\right)^2 \times 8/\pi^2 > \frac{2}{3}$, we have $\tilde{t} = t$.

The number of applications of f is $2\lceil 14\pi\sqrt{N}\,\rceil + M$. Consider the expected value of M_i for $i = 1, 2$. Since

$$\sqrt{(t'_i + 1)(N - t'_i + 1)} \;\leq\; \sqrt{(t+1)(N-t+1)} + \sqrt{N|t'_i - t|}$$

for any $0 \leq t'_i, t \leq N$, we just need to upper bound the expected value of $\sqrt{N|t'_i - t|}$. By Theorem 13, for any $k \geq 2$,

$$|t'_i - t| \ \leq \ k\sqrt{\frac{t(N-t)}{N}} + k^2$$

with probability at least $1 - \frac{1}{k}$. Hence M_i is less than

$$30(1+k)\left(\sqrt{(t+1)(N-t+1)} + \sqrt{N}\right) + 1 \tag{16}$$

with probability at least $1 - \frac{1}{k}$.

In particular, the minimum of M_1 and M_2 is greater than the expression given in Equation 16 with probability at most $\frac{1}{k^2}$. Since any positive random variable Z satisfying $\text{Prob}(Z > k) \leq \frac{1}{k^2}$ has expectation upper bounded by a constant, the expected value of M is in $O\big(\sqrt{(t+1)(N-t+1)}\big)$. □

It follows from Theorem 4.10 of [1] that any quantum algorithm capable of deciding with high probability whether or not a function $f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$ is such that $|f^{-1}(1)| \leq t$, given some $0 < t < N$, must query f a number of times which is at least in $\Omega\big(\sqrt{(t+1)(N-t+1)}\big)$ times. Therefore, our exact counting algorithm is optimal up to a constant factor.

Note also that successive applications of Grover's algorithm in which we strike out the solutions as they are found will also provide an algorithm to perform exact counting. In order to obtain a constant probability of success, if the algorithm fails to return a new element, one must do more than a constant number of trials. In particular, repeating until we get $\log(N)$ failures will provide an overall constant probability of success. Unfortunately, the number of applications of f is then in $O\big(\sqrt{tN} + \log(N)\sqrt{N/t}\big)$ and the cost in terms of additional quantum memory is prohibitive, that is in $\Theta(t)$.

5 Concluding remarks

Let $f : \{0, 1, \ldots, N-1\} \rightarrow \{0, 1\}$ be a function provided as a black box, in the sense that the only knowledge available about f is given by evaluating it on arbitrary points in its domain. We are interested in the number of times that f must be evaluated to achieve certain goals, and this number is our measure of efficiency. Grover's algorithm can find the x_0 such that $f(x_0) = 1$ quadratically faster in the expected sense than the best possible classical algorithm provided the solution is known to be unique [8, 9]. We have generalized Grover's algorithm in several directions.

- ◇ The quadratic speedup remains when the solution is not unique, even if the number of solutions is not known ahead of time.

- ◇ If the number of solutions is known (and nonzero), we can find one quadratically faster in the worst case than would be possible classically even in the expected case.

- ◇ If the number t of solutions is known to be either 0 or t_0, we can tell which is the case with certainty, and exhibit a solution if $t > 0$, in a time in $O(\sqrt{N/t_0}\,)$ in the worst case. By contrast, the best classical algorithm would need $N - t_0 + 1$ queries in the worst case. This is much better than a quadratic speedup when t_0 is large.

- ◇ The quadratic speedup remains in a variety of settings that are not constrained to the black-box model: even if additional information about f can be used to design efficient classical heuristics, we can still find solutions quadratically faster on a quantum computer, provided the heuristic falls under the broad scope of our technique.

- ◇ We give efficient quantum algorithms to estimate the number of solutions in a variety of error models. In all cases, our quantum algorithms are proven optimal, up to a multiplicative constant, among all possible quantum algorithms. In most cases, our quantum algorithms are known to be quadratically faster than the best possible classical algorithm. In the case of counting the number of solutions up to relative error ε, our optimal quantum algorithm is quadratically faster than the best known classical algorithm for fixed ε, but in fact it is better than that when ε is not a constant. Since we do not believe that a super-quadratic quantum improvement for a non-promise black-box problem is possible, we conjecture that there exists a classical algorithm that uses a number of queries in $O(\min\{M^2, N\})$, where $M = \sqrt{\frac{N}{\lfloor \varepsilon t \rfloor + 1}} + \frac{\sqrt{t(N-t)}}{\lfloor \varepsilon t \rfloor + 1}$ is proportional to the number of queries required by our optimal quantum algorithm. This conjecture is further supported by the fact that we can easily find a good estimate for M^2, without prior knowledge of t, using a number of classical queries in $O(\frac{1}{\varepsilon} + \frac{N}{t+1})$.

- ◇ We can amplify efficiently the success probability not only of classical search algorithms, but also of quantum algorithms. More precisely, if a quantum algorithm can output an x that has probability $a > 0$ of being such that $f(x) = 1$, then a solution can be found after evaluating f an expected number of time in $O(1/\sqrt{a}\,)$. If the value of a is known, a solution can be found after evaluating f a number of time in $O(1/\sqrt{a}\,)$ even in the worst case. We call this process *amplitude amplification.* Again, this is quadratically faster than would be possible if the quantum search algorithm were available as a black box to a classical algorithm.

- ◇ Finally, we provide a general technique, known as *amplitude estimation*, to estimate efficiently the success probability a of quantum search algorithms. This is the natural quantum generalization of the above-mentioned technique to estimate the number of classical solutions to the equation $f(x) = 1$.

The following table summarizes the number of applications of the given function f in the quantum algorithms presented in this paper. The table also compares the quantum complexities with the classical complexities of these problems, when the latter are known. Any lower bounds indicated (implicit in the use of the "Θ" notation) correspond to those in the black-box model of computation. In the case of the efficiency of quantum counting with accuracy ε, we refer to the algorithm given below in the Appendix.

Problem	Quantum Complexity	Classical Complexity
Decision	$\Theta(\sqrt{N/(t+1)}\,)$	$\Theta(N/(t+1))$
Searching	$\Theta(\sqrt{N/(t+1)}\,)$	$\Theta(N/(t+1))$
Counting with error $\sqrt{t}$	$\Theta(\sqrt{N}\,)$	
Counting with accuracy ε	$\Theta\left(\sqrt{\frac{N}{\lfloor \varepsilon t \rfloor + 1}} + \frac{\sqrt{t(N-t)}}{\lfloor \varepsilon t \rfloor + 1}\right)$	$O(\frac{1}{\varepsilon^2} N/(t+1))$
Exact counting	$\Theta\big(\sqrt{(t+1)(N-t+1)}\,\big)$	$\Theta(N)$

We leave as open the problem of finding a quantum algorithm that exploits the structure of some searching or counting problem in a genuinely quantum way. By this, we mean in a way that is not equivalent to applying amplitude amplification or amplitude estimation to a classical heuristic. Note that Shor's factoring algorithm does this in the different context of integer factorization.

Acknowledgements

We are grateful to Joan Boyar, Harry Buhrman, Artur Ekert, Ashwin Nayak, Jeff Shallitt, Barbara Terhal and Ronald de Wolf for helpful discussions.

A Tight Algorithm for Approximate Counting

Here we combine the ideas of algorithms **Basic_Approx_Count** and **Exact_Count** to obtain an optimal algorithm for approximately counting. That this algorithm is optimal follows readily from Corollary 1.2 and Theorem 1.13 of Nayak and Wu [14].

Theorem 18 *Given a Boolean function f with N and t defined as above, and any ε such that $\frac{1}{3N} < \varepsilon \leq 1$, the following algorithm* **Approx_Count**(f, ε) *outputs an estimate $\tilde{t}$ such that*

$$\left|\tilde{t} - t\right| \;\leq\; \varepsilon t$$

with probability at least $\frac{2}{3}$, using an expected number of evaluations of f in the order of

$$S = \sqrt{\frac{N}{\lfloor \varepsilon t \rfloor + 1}} + \frac{\sqrt{t(N-t)}}{\lfloor \varepsilon t \rfloor + 1}.$$

If $t = 0$ or $t = N$, the algorithm outputs $\tilde{t} = t$ with certainty.

We assume that $\varepsilon N > 1/3$, since otherwise approximately counting with accuracy ε reduces to exact counting. Set

$$S' = \min\left\{ \frac{1}{\sqrt{\varepsilon}} \sqrt{\frac{N}{t}} \left(1 + \sqrt{\frac{N-t}{\varepsilon N}} \right),\ \sqrt{(t+1)(N-t+1)} \right\} \tag{17}$$

and note that $S' \in \Theta(S)$ where S is defined as in Theorem 18. The algorithm works by finding approximate values for each of the different terms in Equation 17. The general outline of the algorithm is as follows.

Algorithm(Approx_Count(f, ε) **)**

1. Find integer L_1 approximating $\sqrt{N/(t+1)}$.
2. Find integer L_2 approximating $\sqrt{(N-t)/(\varepsilon N)}$.
3. Set $M_1 = \frac{1}{\sqrt{\varepsilon}} L_1(1+L_2)$.
4. If $M_1 > \sqrt{N}$ then find integer M_2 approximating $\sqrt{(t+1)(N-t+1)}$. If $M_1 \le \sqrt{N}$ then set $M_2 = \infty$.
5. Set $M = \min\{M_1, M_2\}$.
6. Set $t' =$ **Count**$(f, \lceil 10\pi M \rceil)$.
7. Output an integer $\tilde{t}$ satisfying $\left|\tilde{t} - t'\right| \le \frac{2}{3}$.

Proof To find L_1, we run steps 1 to 4 of algorithm **Basic_Approx_Count** and then set $L_1 = \lceil 9\pi \times 2^l \rceil$. A proof analogous to that of Theorem 15 gives that

- $L_1 > \sqrt{N/(t+1)}$ with probability at least 0.95, and
- the expected value of L_1 is in $\Theta\big(\sqrt{N/(t+1)}\,\big)$.

This requires a number of evaluations of f which is in $\Theta(L_1)$, and thus, the expected number of evaluations of f so far is in $O(S')$.

In step 2, for some constant c to be determined below, we use $2\lceil \frac{c}{\sqrt{\varepsilon}} \rceil$ evaluations of f to find integer L_2 satisfying

- $L_2 > \sqrt{(N-t)/(\varepsilon N)}$ with probability at least 0.95, and
- the expected value of L_2 is in $O\big(\sqrt{(N-t+1)/(\varepsilon N)}\,\big)$.

Since $N - t = |f^{-1}(0)|$, finding such L_2 boils down to estimating, with accuracy in $\Theta(\sqrt{\varepsilon}\,)$, the square root of the probability that f takes the value 0 on a random point in its domain. Or equivalently, the probability that $\neg f$ takes the value 1, where $\neg f = 1 - f$. Suppose for some constant c, we run **Count**$(\neg f, \lceil \frac{c}{\sqrt{\varepsilon}} \rceil)$ twice with outputs $\tilde{r}_1$ and $\tilde{r}_2$. By Theorem 13, each output $\tilde{r}_i$ $(i = 1, 2)$ satisfies that

$$\left| \sqrt{\frac{\tilde{r}_i}{\varepsilon N}} - \sqrt{\frac{N-t}{\varepsilon N}} \right| \leq \sqrt{\frac{2\pi k}{c}} \sqrt[4]{\frac{N-t}{\varepsilon N}} + \frac{\pi k}{c}$$

with probability at least $1 - \frac{1}{2(k-1)}$ for every $k \geq 2$. It follows that $\tilde{r} = \min\left\{ \sqrt{\tilde{r}_1/(\varepsilon N)}, \sqrt{\tilde{r}_2/(\varepsilon N)} \right\}$ has expected value in $O\big(\sqrt{(N-t+1)/(\varepsilon N)}\,\big)$. Setting $k = 21$, $c = 8\pi k$, and $L_2 = \lceil 2\tilde{r} \rceil + 1$, ensures that L_2 satisfies the two properties mentioned above. The number of evaluations of f in step 2 is in $\Theta(\frac{1}{\sqrt{\varepsilon}})$ which is in $O(S')$.

In step 3, we set $M_1 = \frac{1}{\sqrt{\varepsilon}} L_1(1 + L_2)$. Note that

- $M_1 > \frac{1}{\sqrt{\varepsilon}} \sqrt{\frac{N}{t+1}} \left(1 + \sqrt{\frac{N-t}{\varepsilon N}}\right)$ with probability at least 0.95^2, and
- the expected value of M_1 is in the order of $\frac{1}{\sqrt{\varepsilon}} \sqrt{\frac{N}{t+1}} \left(1 + \sqrt{\frac{N-t+1}{\varepsilon N}}\right)$.

In step 4, analogously to algorithm **Exact_Count**, a number of evaluations of f in $\Theta(\sqrt{N})$ suffices to find an integer M_2 such that

- $M_2 > \sqrt{(t+1)(N-t+1)}$ with probability at least 0.95, and
- the expected value of M_2 is in $\Theta\big(\sqrt{(t+1)(N-t+1)}\,\big)$.

Fortunately, since $\sqrt{(t+1)(N-t+1)} \geq \sqrt{N}$, we shall only need M_2 if $M_1 > \sqrt{N}$. We obtain that, after step 5,

- M is greater than

  $$\min\left\{ \frac{1}{\sqrt{\varepsilon}} \sqrt{\frac{N}{t+1}} \left(1 + \sqrt{\frac{N-t}{\varepsilon N}}\right),\ \sqrt{(t+1)(N-t+1)} \right\}$$

  with probability at least $0.95^3 > 0.85$, and
- the expected value of M is in $O(S')$.

To derive this latter statement, we use the fact that the expected value of the minimum of two random variables is at most the minimum of their expectation.

Finally, by Theorem 13, applying algorithm $\mathbf{Count}(f, \lceil 10\pi M \rceil)$ given such an M, produces an estimate t' of t such that $|t' - t| \leq \frac{\varepsilon t}{3}$ (which implies that $|\tilde{t} - t| \leq \varepsilon t$) with probability at least $8/\pi^2$. Hence our overall success probability is at least $0.85 \times 8/\pi^2 > 2/3$, and the expected number of evaluations of f is in $O(S')$. □

References

[1] Beals, Robert, Harry Buhrman, Richard Cleve, Michele Mosca and Ronald de Wolf, "Quantum lower bounds by polynomials", *Proceedings of 39th Annual Symposium on Foundations of Computer Science*, November 1998, pp. 352 – 361.

[2] Bennett, Charles H., "Notes on the history of reversible computation", *IBM Journal of Research and Development*, 1988, Vol. 32, pp. 16 – 23.

[3] Boyer, Michel, Gilles Brassard, Peter Høyer and Alain Tapp, "Tight bounds on quantum searching", *Fortschritte Der Physik*, special issue on quantum computing and quantum cryptography, 1998, Vol. 46, pp. 493 – 505.

[4] Brassard, Gilles and Peter Høyer, "An exact quantum polynomial-time algorithm for Simon's problem", *Proceedings of Fifth Israeli Symposium on Theory of Computing and Systems*, IEEE Computer Society Press, June 1997, pp. 12 – 23.

[5] Brassard, Gilles, Peter Høyer and Alain Tapp, "Quantum counting", *Proceedings of 25th International Colloquium on Automata, Languages, and Programming*, Lecture Notes in Computer Science, Vol. 1443, Springer-Verlag, July 1998, pp. 820 – 831.

[6] Chi, Dong–Pyo and Jinsoo Kim, "Quantum database searching by a single query", Lecture at *First NASA International Conference on Quantum Computing and Quantum Communications*, Palm Springs, February 1998.

[7] Cleve, Richard, Artur Ekert, Chiara Macchiavello and Michele Mosca, “Quantum algorithms revisited”, *Proceedings of the Royal Society, London*, Vol. A354, 1998, pp. 339 – 354.

[8] Grover, Lov K., “A fast quantum mechanical algorithm for database search”, *Proceedings of 28th Annual ACM Symposium on Theory of Computing*, May 1996, pp. 212 – 219.

[9] Grover, Lov K., “Quantum mechanics helps in searching for a needle in a haystack”, *Physical Review Letters*, Vol. 79, July 1997, pp. 325 – 328.

[10] Grover, Lov K., “Quantum computers can search rapidly by using almost any transformation”, *Physical Review Letters*, Vol. 80, May 1998, pp. 4329 – 4332.

[11] Høyer, Peter, “Conjugated operators in quantum algorithms”, *Physical Review A*, Vol. 59, May 1999, pp. 3280 – 3289.

[12] Kitaev, A. Yu., “Quantum measurements and the Abelian stabilizer problem”, November 1995. Available at Los Alamos e-Print archive as <http://arXiv.org/abs/quant-ph/9511026>.

[13] Mosca, Michele, “Quantum searching and counting by eigenvector analysis”, *Proceedings of Randomized Algorithms*, Satellite Workshop of 23rd International Symposium on Mathematical Foundations of Computer Science, Brno, Czech Republic, August 1998, pp. 90 – 100.

[14] Nayak, Ashwin and Felix Wu, “The quantum query complexity of approximating the median and related statistics”, *Proceedings of 31st Annual ACM Symposium on Theory of Computing*, May 1999, pp. 384 – 393.

[15] Shor, Peter W., “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer”, *SIAM Journal on Computing*, Vol. 26, October 1997, pp. 1484 – 1509.