How fast can quantum annealers count?

Itay Hen

Department of Physics, University of California, Santa Cruz, CA 95064, USA
E-mail: itayhe@physics.ucsc.edu

Received 8 November 2013, revised 3 April 2014
Accepted for publication 23 April 2014
Published 27 May 2014

Abstract
We outline an algorithm for the quantum counting problem using adiabatic quantum computation (AQC). We show that the mechanism of quantum-adiabatic evolution may be utilized toward estimating the number of solutions to a problem, and not only to find them. Using local adiabatic evolution, a process in which the adiabatic procedure is performed at a variable rate, the problem of counting the number of marked items in an unstructured database is solved quadratically faster than the corresponding classical algorithm. The above algorithm provides further evidence for the potentially powerful capabilities of AQC as a paradigm for more efficient problem solving on a quantum computer, and may be used as the basis for solving more sophisticated problems.

Keywords: adiabatic quantum computing, quantum annealers, Grover’s search algorithm, quantum counting
PACS numbers: 03.67.Ac, 03.67.Lx, 89.70.+c

1. Introduction

The paradigm of adiabatic quantum computation (AQC), proposed by Farhi et al [1] about a decade ago is a simple yet intriguing approach to problem solving on a quantum computer. Unlike the leading paradigm of circuit-based quantum computing, AQC is an analogue, time-continuous method which does not require the design and use of quantum gates. As such, it can in many ways be thought of as a simpler and perhaps more profound method for performing quantum computations while also being easier to design and implement experimentally [2–5].

The potential advantages of AQC over the usual circuit-based and other computational paradigms have generated substantial research, and intensive efforts have been made toward finding computational problems that are solved faster on an adiabatic quantum computer (henceforth, a quantum annealer) than on a classical computer (see, e.g., [6–15] and references therein). One of the most important results in that context, which establishes the universality and quantum power of the AQC paradigm is the existence of a theorem stating that adiabatic quantum computing is polynomially equivalent to circuit model quantum computation [16, 17].
Despite the above equivalence, it is not yet understood in which cases AQC is as efficient as circuit model quantum computation (or even potentially polynomially faster). It would therefore be interesting to look at problems for which circuit model algorithms are known, and to see whether equally-efficient adiabatic algorithms can be found. To date, however, there are almost no examples to show that AQC algorithms are similar in complexity to circuit-model algorithms. One notable exception is the adiabatic Grover’s search algorithm devised by Roland and Cerf [18], demonstrating that the quadratic speedup over classical algorithms first shown by Grover [19] using circuit-based computation, can also be gained with adiabatic quantum algorithms (we discuss this algorithm in more detail later).

In what follows, we consider another problem for which, much like Grover’s search problem, there is a circuit-model algorithm that is quadratically faster than its classical counterpart. This is the problem of ‘quantum counting’ [20, 21] in which one has to estimate the number of occurrences of a specific item in a database.

The problem of counting the number of solutions to an optimization problem (equivalently, the ground states of physical systems) is known to play a key role in the understanding of the physical properties of correlated classical as well as quantum systems (see, e.g., [22] and references therein). The computational difficulty of solving this problem is the focus of ongoing studies in computer science as well [22, 23].

Here, we show that the quantum counting problem, where one has to estimate the number of marked items in an unsorted ‘blackbox’ database, may also be solved on a quantum annealer and, quadratically faster than the best corresponding classical algorithms via the use of local adiabatic evolution (LAE) [18]. We shall see that the complexity of the proposed method is on par with the analogous circuit-based algorithm and so is quadratically faster than the corresponding classical algorithm.

The paper is organized as follows. In the next section we briefly discuss the principles of the quantum adiabatic algorithm (QAA) that is the heart of AQC. We then describe in some detail the concept of LAE in section 3. In section 4, we analyze the dynamical equations of the QAA for the adiabatic Grover’s algorithm. These are later used in sections 5 and 6 to devise a QAA for solving the problem of quantum counting with quadratic speedup. Finally, in section 7, we conclude with a few comments.

2. Quantum adiabatic algorithm

We first describe the QAA which provides the general approach for solving problems by casting them into the form of optimization problems [1].

Within the framework of the QAA approach, the solution to an optimization problem is encoded in the ground state of a blackbox Hamiltonian $\hat{H}_p$. To find the solution, the QAA prescribes the following. As a first step, the system is prepared in the ground state of another Hamiltonian $\hat{H}_d$, commonly referred to as the driver Hamiltonian. The driver Hamiltonian is chosen such that it does not commute with the problem Hamiltonian and has a ground state that is fairly easy to prepare. As a next step, the Hamiltonian of the system is slowly modified from $\hat{H}_d$ to $\hat{H}_p$, using the linear interpolation, i.e.,

$$\hat{H}(s) = s\hat{H}_p + (1 - s)\hat{H}_d,$$  

where $s(t)$ is a parameter varying smoothly with time, from 0 at $t = 0$ to 1 at the end of the algorithm, at $t = T$. If this process is done slowly enough, the adiabatic theorem of quantum mechanics (see, e.g., [24] and [25]) ensures that the system will stay close to the ground state of the instantaneous Hamiltonian throughout the evolution, so that the system evolves to a
state that is close to the ground state of $\hat{H}_p$. At this point, measuring the state will give the solution of the original problem with high probability.

In the simple case where $s(t)$ varies from 0 to 1 at a constant rate, the runtime $T$ must be chosen to be large enough so that the adiabatic approximation holds: this condition determines the efficiency, or complexity, of the QAA. A condition\(^1\) on $T$ can be given in terms of the instantaneous eigenstates $\{|m\rangle\}$ and eigenvalues $\{E_m\}$ of the Hamiltonian $H(s)$, as $[7, 30]$

$$T > \frac{1}{\epsilon} \max_s V_{01}(s),$$

where $\epsilon$ is a small number, inversely proportional to the running time of the algorithm, that controls the expected probability for failure of the algorithm, $g(s)$ is the first excitation gap $E_1(s) - E_0(s)$ and $V_{01}(s) = |\langle 0|d\hat{H}/ds|1\rangle|$ (in our units $\hbar = 1$).

3. Local adiabatic evolution

In the framework of LAE, the adiabatic parameter $s(t)$ is varied not at a constant rate but rather at a variable rate, slowing down in the vicinity of the minimum gap and speeding-up in places where the gap is large [18]. The procedure is based on formulating a ‘local’ Landau–Zener condition for each value of the adiabatic parameter $s$. This condition can be written as:

$$\frac{|ds|}{dt} \leq \frac{g^2(s)}{V_{01}(s)},$$

where $g(s), V_{01}(s)$ and $\epsilon$ are as in (2).

When applicable, LAE can provide a significant speedup in calculation times [18, 31, 32]. A well-known example to demonstrate this, was first considered by Roland and Cerf [18]. In this example, an adiabatic algorithm for solving Grover’s search problem [19] which describes a search for a marked item in an unstructured database, was found. More specifically, Roland and Cerf showed that while the application of the adiabatic theorem with a linear $s(t)$ results in a running time that is of order $N$, where $N$ is the number of items in the database, a carefully-chosen variable rate of $s(t)$ yields a running time that scales with $\sqrt{N}$, i.e., a quadratic speedup is gained, similarly to the original result by Grover, that was found for the circuit-based model.

LAE however can only be efficiently used in cases where one has proper knowledge of the exact behavior of the gap and relevant matrix elements of the system for the problem at hand. This is normally not the case. The amenability of Grover’s search problem to LAE stems from the fact that the gap of the system (as well as the relevant matrix element) can be calculated explicitly beforehand, even if the specific ground state is not known. With a proper choice of problem and driver Hamiltonians (which are discussed next), the gap in the Grover search problem is given by the simple expression:

$$g(\eta; s) = [1 - 4(1 - \eta)s(1 - s)]^{1/2},$$

where $\eta = M/N$ is the ratio of the number of solutions $M$ to the size of the database $N$. Moreover, the matrix element $|\langle 0|d\hat{H}/ds|1\rangle|$ can also be calculated in a straightforward manner to give:

$$V_{01}(s) = |\langle 0|d\hat{H}/ds|1\rangle| = \frac{\sqrt{\eta(1 - \eta)}}{g(\eta; s)}.$$

\(^1\) We note that the conditions given here are in general neither necessary nor sufficient. More rigorous conditions may be found in, e.g., [26–29]. The ‘traditional’ condition stated here is however sufficient for adiabatic search algorithms such as the one discussed here [28].
The condition for LAE for the Grover problem is thus simply given by:

\[ \left| \frac{ds}{dt} \right| \leq \epsilon \frac{g^2(\eta; s)}{\sqrt{\eta(1-\eta)}}. \] (6)

A less strict inequality, using the fact that \( V_{01}(s) < 1 \), yields the more convenient bound of [18]:

\[ \left| \frac{ds}{dt} \right| \leq \epsilon g^2(\eta; s). \] (7)

Integrating in time either of the two equations above between \( t = 0 \) (also, \( s = 0 \)) and \( t = T \) (\( s = 1 \)) yields in the limit of \( N \to \infty \):

\[ T \propto \frac{1}{\epsilon} \sqrt{\frac{N}{M}} = \frac{1}{\epsilon \sqrt{\eta}} \] (8)

i.e., a quadratic speedup over the corresponding classical algorithms, a result also shared by Grover’s original circuit-based algorithm [19].

4. Analysis of the adiabatic Grover problem

An important property of the gap of Grover’s search problem, (4), may be formulated as follows.

\[ g(\eta_1; s) \leq g(\eta_2; s) \quad \text{for any} \quad \eta_1 \leq \eta_2, \] (9)

where \( \eta_1 \) and \( \eta_2 \) are two arbitrary ratios. The above property leads to the interesting (and somewhat surprising) attractive feature of LAE for the Grover search problem: if one runs the algorithm as if the ratio of solutions is \( \eta^* \) even though the actual ratio of solutions \( \eta \) may not be known, then one still obtains at the end of the adiabatic run a state that is very close the symmetric superposition of all solution states, as long as \( \eta^* \leq \eta \). This is because choosing the adiabatic profile with \( \eta^* \), namely,

\[ \frac{ds}{dt} = \epsilon g^2(\eta^*; s), \] (10)

also satisfies the inequality (7) for any \( \eta^* \leq \eta \). For example, a simple strategy for finding a solution state without knowing the number of solutions would be to run the algorithm with a profile function corresponding to \( M^* = 1 \) (i.e., \( \eta^* = 1/N \)). This will ensure a solution state at the end of the evolution with a (suboptimal) runtime of \( T \propto \frac{1}{\epsilon} \sqrt{N} \) (other, perhaps more efficient, strategies for finding a solution state without knowing \( M \) may also be found).

The above feature of the adiabatic procedure is to be contrasted with the circuit-based Grover’s search algorithm, in which case one must know in advance the number of solutions in order to obtain a solution state with high probability. In the circuit-based case, the number of solutions \( M \) determines the number of ‘Grover iterations’ needed for the algorithm to end up in a solution state. Too many or too few iterations will drastically lower the probability of finding a solution. For that purpose, the quantum counting algorithm [20] has been devised, with the goal of estimating the number of solutions \( M \) prior to the application of Grover’s search algorithm.

In the gate mode, the probability of success is periodic in the number of Grover iterations and it is this periodicity that has been utilized to construct the quantum counting algorithm. As discussed above, the probability of finding a solution in the adiabatic case is not periodic in the runtime of algorithm, so the techniques of the original quantum counting algorithm cannot, unfortunately, be used. However, in what follows we show that LAE, with the aid of the unique properties of the adiabatic Grover’s problem presented above, may be used to
solve the quantum counting problem adiabatically. The method suggested here, is comprised of repeated applications of the adiabatic algorithm for Grover’s search problem of Roland and Cerf [18] but with varying, carefully-chosen, adiabatic profiles.

We therefore first analyze in what follows the dynamics of the adiabatic algorithm for Grover’s search problem (a similar derivation is briefly discussed in [18]). Specifically, we derive here the equations describing an unsorted-database search with an unknown number (equivalently, ratio) of solutions $M$ (or $\eta$) using an adiabatic profile $s(t)$ with a ratio $\eta_*$. We then calculate the probability of a solution state at $s = 1$, in two limiting cases.

4.1. Dynamics of the adiabatic Grover’s algorithm

Consider the problem Hamiltonian for the Grover problem [18]:

$$\hat{H}_p = 1 - \sum_{m \in M} |m\rangle \langle m|.$$  \hspace{1cm} (11)

Here the $M$ states $m \in M$ are eigenstates of the computational basis with eigenvalue zero (all other states have eigenvalue one). The problem Hamiltonian is therefore a diagonal matrix of size $N$ with $M$ zero entries and $(N - M)$ entries of one along the diagonal. The driver Hamiltonian is simply given by:

$$\hat{H}_d = -|\phi\rangle \langle \phi|.$$  \hspace{1cm} (12)

where $|\phi\rangle$ is the fully-symmetric state:

$$|\phi\rangle = \frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle.$$  \hspace{1cm} (13)

Obviously, the ground state of the driver Hamiltonian, i.e., the state that the system is initially prepared in, is

$$|\psi(t = 0)\rangle = |\phi\rangle.$$  \hspace{1cm} (14)

The Schrödinger equation

$$i\frac{d}{dt}\psi(t) = \hat{H}\psi(t)$$ \hspace{1cm} (15)

with $|\psi(t)\rangle = (c_0(t), c_1(t), \ldots, c_i(t), \ldots, c_N(t))$ simplifies considerably due to symmetry considerations: the state of the system has only two distinct amplitudes. The amplitudes $c_i(t)$ that correspond to ‘solution’ states and those that correspond to ‘non-solution’ states. Let us denote the former set by $a(t)$ and that of the latter by $b(t)$. The Schrödinger equation thus becomes the two coupled equations:

$$ia = -(1 - s)[\eta a + (1 - \eta)b]$$ \hspace{1cm} (16a)

$$ib = -(1 - s)[\eta a + (1 - \eta)b] + sb.$$ \hspace{1cm} (16b)

Note that $N$, the size of the problem and $M$, the number of solutions appear in the equations only as the ratio $\eta = M/N$.

The time-dependence of $a$ and $b$ is easily transformed into $s$ dependence (as there is a one-to-one correspondence between the two in our case). The time-derivative transforms in the usual way: $d/dt = (ds/dt) \times d/ds$, and the two coupled equations now become:

$$ieg^2(\eta_*; s)a' = -(1 - s)[\eta a + (1 - \eta)b]$$ \hspace{1cm} (17a)

$$ieg^2(\eta_*; s)b' = -(1 - s)[\eta a + (1 - \eta)b] + sb.$$ \hspace{1cm} (17b)
Here, the prime symbol (') stands for differentiation with respect to the new independent variable $s$ and we have used (10) for $ds/dt$ where the ratio $\eta_s$ may in general be different than $\eta$. As already noted earlier, a choice of $\eta_s \ll \eta$ results in a true adiabatic evolution and is expected to yield at the end of the QAA, a state that is very close to the equal superpositions of all solution states.

In addition, the adiabatic process dictates the initial conditions
\begin{equation}
    a(s = 0) = b(s = 0) = 1,
\end{equation}
which correspond to the fully-symmetric state (i.e., an equal superposition of all possible computational-basis states). The specific value of 1 chosen here leads to the simple normalization condition
\begin{equation}
    \eta|a|^2 + (1 - \eta)|b|^2 = 1,
\end{equation}
that, similarly to the equations of motion, does not explicitly depend on $N$ or $M$ but only on their ratio.

### 4.2. End probabilities

At the end of the evolution, at $s = 1$, the final state of the system is a linear combination of the equal superposition of solutions states, namely, of $|\psi_{\text{sol}}\rangle \propto \sum_{m \in \mathcal{M}} |m\rangle$ and the equal superposition of the non-solution states $|\psi_{\text{no-sol}}\rangle \propto \sum_{m \notin \mathcal{M}} |m\rangle$. The closer the evolution is to being truly adiabatic, the larger the amplitude of the solution-states superposition will be. While the exact values of the two complex components at $s = 1$ cannot be computed analytically, one can obtain an expression for the probability of a solution state, i.e., the squared overlap between the final state at $s = 1$ and the optimal solution $|\psi_{\text{sol}}\rangle$:
\begin{equation}
    P_{\text{sol}} = |\langle \psi(s = 1) | \psi_{\text{sol}} \rangle|^2 = \lim_{s \to 1} \frac{\eta |a(s)|^2}{\eta |a(s)|^2 + (1 - \eta) |b(s)|^2},
\end{equation}
in two important limiting cases. One is the limit of small $\epsilon$, i.e., when the allowed error in the final state goes to zero. In the $\epsilon \to 0$ limit, the two equations are of the WKB type [33], and have a series solution of the form
\begin{equation}
    a(s) = e^{\text{Re}[\sigma'(s)]} \text{ with } \sigma'(s) = \sum_{k=0}^{\infty} (i \epsilon)^k \int_s^1 a_k(s') \, ds',
\end{equation}
where the functions $a_k(\cdot)$ are real-valued. The above representation can be directly plugged into the coupled equations of motion. The first of the equations (17a) can be solved for $b(s)$ to give:
\begin{equation}
    b(s) = -e^{\text{Re}[\sigma(s)]} \left( 1 - \eta \frac{\eta_s^2 (\eta_s s) - 1}{1 - s} \text{Im}[\sigma'(s)] \right)
\end{equation}
and the probability of a solution state simplifies to:
\begin{equation}
    P_{\text{sol}} = \lim_{s \to 1} \frac{\eta |a(s)|^2}{\eta |a(s)|^2 + (1 - \eta) |b(s)|^2} = \lim_{s \to 1} \frac{\eta}{\eta + (1 - \eta) r(s)^2},
\end{equation}
where we have defined:
\begin{equation}
    r(s)^2 = |b(s)/a(s)|^2 = \left( \frac{\eta(1 - s) - \epsilon g^2(\eta_s s) \text{ Re}[\sigma'(s)]}{(1 - s)(1 - \eta)} \right)^2 + \left( \frac{\epsilon g^2(\eta_s s) \text{ Im}[\sigma'(s)]}{(1 - s)(1 - \eta)} \right)^2.
\end{equation}
Replacing \( \sigma(s) \) with its series expansion, turns the second equation (17b) into a series of equations for the various orders of \( \epsilon \). To second-order, we may keep only \( \alpha_0(\cdot) \) and \( \alpha_1(\cdot) \), at which stage solving for these and taking the \( s \to 1 \) limit, yields:

\[
P_{\text{sol}}(\epsilon \to 0) = 1 - \epsilon^2 \eta(1 - \eta) + O(\epsilon^4). \tag{25}
\]

A lengthier and more tedious calculation in which terms up to fourth-order of \( \epsilon \) are kept, yields the fourth-order correction \( \epsilon^4 \eta(1 - \eta)(4 - 77\eta + 105\eta^2 + 64\eta^3 - 112\eta^4 + 16\eta^5) \).

To leading order in \( \epsilon \), the probability to find a solution does not depend on value of \( \eta_\ast \), as long as \( \epsilon \) is small enough. As expected, when \( \epsilon \) goes to 0 (i.e., when no error is allowed), the probability of the system to be in the equal superposition of solution states at the end of the QAA approaches 1 (in this case, the running time of the algorithm goes to infinity).

The more interesting limit however, is the one in which \( \epsilon \) is small but kept fixed while \( N \), the dimension of the Hilbert space (alternatively, the size of the database) goes to infinity. This limit translates to the limit where the ratios \( \eta = M/N \) and \( \eta_\ast = M_\ast/N \) are small\(^2\), i.e., \( M, M_\ast \ll N \). In this case, the probability of success is completely determined by the Landau–Zener transition in the immediate vicinity of the minimum gap, at \( s = 1/2 \). There, the gap can easily be calculated to be \( g(\eta; s = 1/2) = \sqrt{\eta} \). In addition, the Landau–Zener sweep rate is determined on the other hand by \( dg/ds = dg/d\eta \times d\eta/ds = 2 \times \epsilon g^2(\eta_\ast, s = 1/2) = 2\epsilon \eta_\ast \). The Landau–Zener formula therefore immediately yields:

\[
P_{\text{sol}}(N \to \infty) \approx 1 - e^{-\frac{\pi}{\pi\eta\sqrt{\epsilon}}} = 1 - e^{-\frac{\pi}{\pi\eta\sqrt{\epsilon}}} \tag{26}
\]

As expected, the probability of obtaining a solution increases with decreasing \( \epsilon \) and also, as discussed in the previous section, with decreasing \( M_\ast \). Most importantly, in the \( N \to \infty \) limit, the above probability has a well established \( N \)-independent value. The above probability will become useful in what follows when we address the problem of quantum counting.

5. Quantum counting: zero versus nonzero number of solutions

Before addressing the full-complexity of the general quantum counting problem, let us first consider the special case of finding whether or not the number of solutions \( M \) (number of zero entries in the problem Hamiltonian) is zero. Circuit-based algorithms give a running time that is on the order of \( O(\sqrt{N}) \) [20, 21], that is, a quadratic speedup over the analogous classical algorithm.

To solve this problem quantum-adiabatically, one executes the QAA algorithm with the adiabatic profile function \( s(t) \) that obeys \( ds/dt = \epsilon g^2(\eta = 1/N, s) \), i.e., a profile that is suitable for solving the case of any nonzero number of solutions with quantum speedup.

Now, if the number of solutions \( M \) is zero, then the problem Hamiltonian will have no effect on the state of the system, and the state will remain the initial state \( |\phi\rangle \) throughout the adiabatic evolution, except for an undetectable global phase. A measurement of the \( z \)-magnetization at the end of the run will produce a random configuration with energy 1.

If on the other hand the number of solutions is nonzero, the Grover LAE procedure ensures us that at the end of each of the run, at \( s = 1 \), the final state of the system will be very close to \( |\psi_{\text{sol}}\rangle \), the equal superposition of zero-energy states with a probability greater than \( P_{\text{sol}} = 1 - e^{-\frac{\pi}{\pi\eta\sqrt{\epsilon}}} \) (in the \( N \to \infty \) limit), and a measurement along the \( z \)-direction will produce a zero-energy state with this probability.

We conclude then, that one execution of the adiabatic algorithm with runtime \( T \propto \sqrt{N}/\epsilon \) is sufficient for establishing whether or not the number of solutions is zero.

\(^2\) We note that in the case where the number of solutions \( M \) is of order \( N \), the quantum counting problem becomes less interesting, as the classical probability for finding a solution, \( P_{\text{sol}}^{\text{(classical)}} = M/N \), becomes significantly large, in which case no quantum speedup is gained.
The general case of estimating the number of solutions $M$ can be analyzed with the aid of the probability $P_{\text{sol}}$ obtained in (26). Let us first briefly analyze the classical analogue of that same problem. In the classical case, random sampling of items in the database will give $P_{\text{sol}} = M/N = \eta$ for each picked item. The number of successes after $k$ trials will therefore be distributed binomially with the unbiased mean of $\hat{\eta}_{\text{(classical)}} = \eta$ and a sampling error of:

$$\Delta \hat{\eta}_{\text{(classical)}} \sim \sqrt{\frac{\eta(1-\eta)}{k}} \approx \sqrt{\frac{\eta}{k}}.$$  

(27)

In terms of the number of solutions $M$, the error in $M$ is thus

$$\Delta M_{\text{(classical)}} \sim \sqrt{\frac{MN}{k}},$$

i.e., the number of trials $k$ required for estimating the number of solutions is

$$k \sim \frac{NM}{\Delta M^2}. \tag{28}$$

Specifically, for an error estimate of $\Delta M \sim \sqrt{M}$, the number of trials $k$ required, has to scale linearly with the dimension of the database $N$.

In the quantum-adiabatic case, the question thus becomes: how many runs $k$ of the adiabatic Grover’s search algorithm (and with which adiabatic profiles), each run producing a solution state with probability $P_{\text{sol}}$ [as given by (26)], are necessary to yield an unbiased estimate of $M$?

We answer this question by proposing the following two-step algorithm: first, we find a suitable value of $M_{\ast}$, according to which the adiabatic paths for subsequent QAA runs will be constructed. As a second step, we calculate how many QAA runs, with that chosen value of $M_{\ast}$ are required to estimate $M$ for a given error rate $^3$.

The first step of the algorithm is designed to find an optimal value for $M_{\ast}$ that will determine the adiabatic profile to be used in the second step. Note that $M_{\ast}$ should be on the order of the unknown $M$. Choosing a value for $M_{\ast}$ that is too large (i.e., $M_{\ast} \gg M/\epsilon$) would correspond to probabilities that are very close to zero, leading to very poor statistics that would in turn not suffice for the determination of $M$ in an accurate manner. Choosing a value for $M_{\ast}$ that is too small (i.e., $M_{\ast} \ll M/\epsilon$) would lead to unnecessarily-long running times for the adiabatic runs and would correspond to probabilities that are very close to one, leading again to very poor statistics. An optimal choice for $M_{\ast}$ would thus be a value that would yield a significantly-large probability for obtaining a solution state (one reasonable choice could be, e.g., $1 - P_{\text{sol}} \approx 0.1$), which implies $M_{\ast} \sim M/\epsilon$.

Since $M$ is unknown, a way to find a value of $M_{\ast}$ that would yield a given probability, is to employ a simple noisy binary search algorithm [34–36]. Starting with geometrically diminishing values of $M_{\ast}$, namely, $M_{\ast} = N/2, N/4, \ldots$, the QAA should be executed multiple times for each value, such that for each tested value $M_{\ast}$, the average ratio of solutions is computed (note that since $M \ll N$ the ratios will be zero at first). When $M_{\ast}$ comes close to the vicinity of $M$ and the obtained ratios of solutions are no longer strictly zero, the trial $M_{\ast}$ values will need to be adjusted (i.e., increased or decreased) according to rules of binary search, until an approximate value of the chosen probability of solutions, $P_{\text{sol}}$ is obtained. Note that in order

$^3$ Once could certainly imagine a more elaborate algorithm in which $M_{\ast}$ is suitably adjusted after every run or a constant number of runs, depending on the estimate of $M$ at that point. While such an adaptive algorithm is expect to scale somewhat better than the algorithm we propose here, its complexity is harder to analyze analytically.
to obtain a certain probability of solutions $P_{\text{sol}}$ with a square-root error accuracy, only a fixed (i.e., an $M$-independent) number of QAA executions is required for each tested value of $M_*$ (as the number of ‘solution’ outcomes is distributed binomially). For a binary search such as the one outlined above, the number of trial $M_*$ values required is logarithmic in $N$, where the complexity of each trial run is $O(\sqrt{N/M_*})$ [as is given in (8)]. The overall complexity of this first step is therefore simply $O(\sqrt{N/M})$.

Having found an optimal value for $M_*$, the second step of the algorithm is initiated, consisting of the execution of multiple QAA runs with the chosen value of $M_*$ (note that the chosen value obeys $M_* \sim M/\epsilon$). A straightforward calculation analogous to the one performed for the classical case above, following (27), yields, in the quantum-adiabatic case, the number of trials:

$$k \propto \frac{16e^2 \epsilon^2 M_*^2 \left(e^{\frac{2}{M_*}} - 1\right)}{\pi^2 (\Delta M)^2} \approx \frac{4e^2 M_* M}{\epsilon \pi (\Delta M)^2}.$$  

(30)

where the last approximation is justified by the smallness of the probability $1 - P_{\text{sol}}$. Since the complexity of each QAA run is $O(1/\epsilon \sqrt{N/M_*})$, the overall complexity of the algorithm will be $O(k/\epsilon \sqrt{N/M_*}) = O((M/\Delta M)^2 \sqrt{N/\epsilon M})$. For an error that scales with the square root of the number of solutions, i.e., $\Delta M \sim \sqrt{M}$, the complexity of the algorithm will be $O(\sqrt{NM}/\epsilon)$, i.e., the quadratic speedup over the classical analogue is obtained, as the complexity only scales with the square root of the size of the database. Comparing to the corresponding quantum circuit algorithm [20], we find that while both algorithms scale similarly with the size of the database, the scaling of $M$ is somewhat different. For example, for an $O(\sqrt{M})$ error, the circuit-based algorithm yields an $O(\sqrt{N})$ complexity, i.e., it is $\sqrt{M}$ faster than the QAA proposed here. Since, however, we are considering here the case of small number of solutions, this difference plays only a minor role.

A similar comparison for the case where the error is required to scale linearly with the number of solutions, i.e., $\Delta M \sim M$, the adiabatic algorithm above yields a complexity of $O(\sqrt{NM}/\epsilon M)$, which in this case is identical to the circuit-based scaling [20]. For the convenience of the reader, we have added a table comparing the complexities of the classical, quantum-adiabatic and quantum circuit algorithms for different error estimates (see Table 1).

7. Summary and conclusions

We have given a prescription for solving the quantum counting problem in which the number of occurrences of a specific item within an unsorted database is to be estimated. Using local adiabatic evolution on Grover-type problems, the algorithm was shown to perform quadratically faster than the corresponding classical algorithm and on par with the circuit-based result. We have thus shown that quantum-adiabatic evolution may be utilized toward counting the number of solutions (or ground states) to a given problem and is not restricted to only finding those.

| Error scaling | Classical algorithm | Circuit quantum algorithm [20] | Quantum-adiabatic algorithm |
|---------------|---------------------|--------------------------------|----------------------------|
| $\Delta M \sim O(1)$ | $O(NM)$ | $O(\sqrt{NM})$ | $O(M\sqrt{NM})$ |
| $\Delta M \sim O(\sqrt{M})$ | $O(N)$ | $O(\sqrt{N})$ | $O(\sqrt{NM})$ |
| $\Delta M \sim O(M)$ | $O(N/M)$ | $O(\sqrt{N/M})$ | $O(\sqrt{NM})$ |

Table 1. Complexities of the classical, quantum circuit and quantum-adiabatic counting algorithms for different error estimations. Here, $N$ is the size of the database, $M$ is the number of solutions and $\Delta M$ is the error estimation.
We note here, that similarly to the original Grover’s algorithm, the problem Hamiltonian of the quantum counting algorithm presented here, is given as an oracle, and the quadratic speedup exhibited here holds only relative to it.

It is known that the gate-model quantum counting algorithm, may be viewed as a special case of a quantum phase estimation algorithm [21], in which one is required to estimate an eigenvalue of an implementable unitary operator. It would be interesting to see whether the algorithm outlined above can be used as the basis for a similar adiabatic calculation. Since however in the adiabatic case one deals with Hermitian operators, as opposed to the unitary operators of the circuit model, it is unclear how this could be done. Nonetheless, it is worth noting that the fully-symmetric initial wave function $|\phi\rangle$ may be re-expressed in the form [21]:

$$|\phi\rangle = \sqrt{\eta} |\alpha\rangle + \sqrt{1-\eta} |\beta\rangle,$$

where we have defined:

$$|\alpha\rangle = \frac{1}{\sqrt{N\eta}} \sum_{m \in \mathcal{M}} |m\rangle,$$

$$|\beta\rangle = \frac{1}{\sqrt{N(1-\eta)}} \sum_{m \not\in \mathcal{M}} |m\rangle.$$

Replacing $\eta = M/N$ with the appropriate angle of $\sin^2(\theta/2)$, we can rewrite the wave function as:

$$|\phi\rangle = \sin(\theta/2) |\alpha\rangle + \cos(\theta/2) |\beta\rangle,$$

which reveals that estimating $M$ is equivalent to estimating an ‘angle’ $\theta$ that is associated with the problem Hamiltonian, which could possibly be linked to a phase of a unitary operator.

The importance of the above results, aside from the implications resulting from having a quadratically-faster quantum-adiabatic algorithm for this specific problem, lies in the fact that it is one of very few examples that illustrate the power and potential encompassed in adiabatic quantum computation, providing an adiabatic quantum algorithm that is as efficient as its circuit-model counterpart.

Acknowledgments

We thank Peter Young, Eleanor Rieffel and Jérémie Roland for useful comments and discussions. We acknowledge partial support by the National Security Agency (NSA) under Army Research Office (ARO) contract number W911NF-09-1-0391, and in part by the National Science Foundation under grant no. DMR-0906366.

References

[1] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem Science 292 472 (A longer version of the paper appeared in arXiv:quant-ph/0104129)

[2] Vandersypen L M K, Steffen M, Breyta G, Yannoni C S, Sherwood M H and Chuang I L 2001 Experimental realization of Shor’s quantum factoring algorithm using nuclear magnetic resonance Nature 414 883–7

[3] Johnson M W et al 2011 Quantum annealing with manufactured spins Nature 473 194–8

[4] Bian Z, Chudak F, Macready W G, Clark L and Gaitan F 2012 Experimental determination of Ramsey numbers with quantum annealing arXiv:1201.1842

[5] Berkley A J et al 2013 Tunneling spectroscopy using a probe qubit Phys. Rev. B 87 020502(R)
[6] Hogg T 2003 Adiabatic quantum computing for random satisfiability problems Phys. Rev. A 67 022314
[7] Farhi E, Goldstone J and Gutmann S 2002 Quantum adiabatic algorithms versus simulated annealing arXiv:quant-ph/0201031
[8] Farhi E, Goldstone J, Gutmann S and Nagaj D 2008 How to make the quantum adiabatic algorithm fail Int. J. Quantum Inform. 6 503
[9] Altshuler B, Krovi H and Roland J 2009 Adiabatic quantum computing for random satisfiability problems Phys. Rev. A 67 022314
[10] Knysh S and Smelyanskiy V N 2011 On the relevance of avoided crossings away from quantum critical point to the complexity of quantum adiabatic algorithm arXiv:1005.3011
[11] Young A P, Knysh S and Smelyanskiy V N 2010 First order phase transition in the quantum adiabatic algorithm Phys. Rev. Lett. 104 020502
[12] Hen I and Young A P 2011 Exponential complexity of the quantum adiabatic algorithm for certain satisfiability problems Phys. Rev. E 84 061152
[13] Hen I 2012 Excitation gap from optimized correlation functions in quantum Monte Carlo simulations Phys. Rev. E 85 036705
[14] Hen I 2014 Period finding with adiabatic quantum computation Europhys. Lett. 105 50005
[15] Aharonov D, van Dam W, Kempe J, Landau Z, Lloyd S and Regev O 2007 Adiabatic quantum computation is equivalent to standard quantum computation SIAM J. Comput. 37 166
[16] Mivel A, Lidar D A and Mitchell M 2007 Simple proof of equivalence between adiabatic quantum computation and the circuit model Phys. Rev. Lett. 99 070502
[17] Roland J and Cerf N J 2002 Quantum search by local adiabatic evolution Phys. Rev. A 65 042308
[18] Grover L K 1997 Quantum mechanics helps in searching for a needle in a haystack Phys. Rev. Lett. 79 325
[19] Brassard G, Hoyer P and Tapp A 1998 Quantum counting Automata, Languages and Programming (Lecture Notes in Computer Science vol 1443) (Berlin: Springer) pp 820–31
[20] Nielsen M A and Chuang I L 2000 Quantum Computation and Quantum Information (Cambridge: Cambridge University Press)
[21] Brown B, Flammia S T and Schuch N 2011 Computational difficulty of computing the density of states Phys. Rev. Lett. 107 040501
[22] Matsuda Y, Nishimori H and Katzgraber H G 2009 Ground-state statistics from annealing algorithms: quantum versus classical approaches New J. Phys. 11 073021
[23] Kato T 1951 On the adiabatic theorem of quantum mechanics J. Phys. Soc. Japan 5 435
[24] Messiah A 1962 Quantum Mechanics (Amsterdam: North-Holland)
[25] Tong D M, Singh K, Kwek L C and Oh C H 2005 Quantitative conditions do not guarantee the validity of the adiabatic approximation Phys. Rev. Lett. 95 110407
[26] Schaller G, Mostame S and Schützhold R 2006 General error estimate for adiabatic quantum computing Phys. Rev. A 73 062307
[27] Jansen S, Ruskai M B and Seiler R 2007 Bounds for the adiabatic approximation with applications to quantum computation J. Math. Phys. 47 102111
[28] Amin M H S 2009 Consistency of the adiabatic theorem Phys. Rev. Lett. 102 220401
[29] Wannier G H 1965 Probability of violation of the Ehrenfest principle Physics I 251
[30] Krovi H, Magniez F, Ozols M and Roland J 2010 Finding is as easy as detecting for quantum walks Automata, Languages and Programming (Lecture Notes in Computer Science vol 6198) ed S Abramsky, C Gavoille, C Kirchner, Friedhelm Meyer auf der Heide and Paul Spirakis (Berlin: Springer) pp 540–551

11