Experimental Implementation of an Adiabatic Quantum Optimization Algorithm

Matthias Steffen,[1,2,*] Wim van Dam,[3,4] Tad Hogg,[3] Greg Breyta,[5] and Isaac Chuang[1]

[1]*Center for Bits and Atoms–MIT, Cambridge, Massachusetts 02139*
[2]*Solid State and Photonics Laboratory, Stanford University, Stanford, California 94305-4075*
[3]*HP Labs, Palo Alto, California 94304-1126*
[4]*MSRI, Berkeley, California 94720-5070*
[5]*IBM Almaden Research Center, San Jose, California 95120*

(Received 1 October 2002; published 14 February 2003)

We report the realization of a nuclear magnetic resonance computer with three quantum bits that simulates an adiabatic quantum optimization algorithm. Adiabatic quantum algorithms offer new insight into how quantum resources can be used to solve hard problems. This experiment uses a particularly well-suited three quantum bit molecule and was made possible by introducing a technique that encodes general instances of the given optimization problem into an easily applicable Hamiltonian. Our results indicate an optimal run time of the adiabatic algorithm that agrees well with the prediction of a simple decoherence model.

DOI: 10.1103/PhysRevLett.90.067903 PACS numbers: 03.67.Lx, 03.65.Yz, 76.60.–k

Since the discovery of the algorithms of Shor [1] and Grover [2], the quest of finding new quantum algorithms proved a formidable challenge. Recently, however, a novel algorithm was proposed, using adiabatic evolution [3,4]. Despite the uncertainty in its scaling behavior, this algorithm remains a remarkable discovery because it offers new insights into the potential usefulness of quantum resources for computational tasks.

Experimental realizations of quantum algorithms in the past demonstrated Grover's search algorithm, the Deutsch-Jozsa algorithm, order finding, and Shor's algorithm [5,6]. Recently, Hogg's algorithm was implemented using only one computational step [7]; however, a demonstration of an adiabatic quantum algorithm thus far has remained beyond reach.

Here, we provide the first experimental implementation of an adiabatic quantum optimization algorithm using three qubits and nuclear magnetic resonance (NMR) techniques [8]. NMR techniques are especially attractive because several tens of qubits may be accessible, which is precisely the range that could be crucial in determining the scaling behavior of adiabatic quantum algorithms [9]. Compared to earlier implementations of search problems [5,10], this experiment is a full implementation of a true optimization problem which does not require a black box function or ancilla bits.

This experiment was made possible by overcoming two experimental challenges. First, an adiabatic evolution requires a smoothly varying Hamiltonian over time, but the terms of the available Hamiltonian in our system cannot be smoothly varied and may even have fixed values. We developed a method to approximately smoothly vary a Hamiltonian despite the given restrictions by extending NMR average Hamiltonian techniques [11]. Second, general instances of the optimization algorithm may require the application of Hamiltonians that are not easily accessible. We developed methods to implement general instances of a well-known classical NP-complete (nondeterministic, polynomial time) optimization problem given a fixed natural system Hamiltonian.

We provide a concrete procedure detailing these methods. We then apply the results to the Maximum Cut (MAXCUT) [12] optimization problem. Our experiments indicate there exists an optimal total running time which can be predicted using a decoherence model based on independent stochastic relaxation of the spins.

An adiabatic quantum algorithm evolves the quantum state with a slowly varying, time-dependent Hamiltonian. Suppose we are given some time-dependent Hamiltonian $H(t)$, where $0 \leq t \leq T$, and at $t = 0$ we start in the ground state of $H(0)$. By varying $H(t)$ slowly, the quantum system remains in the ground state of $H(t)$ for all $0 \leq t \leq T$ provided the lowest two energy eigenvalues of $H(t)$ are never degenerate [13]. Now suppose we can encode an optimization problem into $H(T)$. Then the state of the quantum system at time $t = T$ represents the solution to the optimization problem [3]. The total run time T of the adiabatic algorithm scales as $g_{\min}^{-2}$, where $g_{\min}$ is the minimum separation between the lowest two energy eigenvalues of $H(t)$ [3,14]. The scaling behavior of $g_{\min}$ will ultimately determine the success of adiabatic quantum algorithms. Classical simulations of this scaling behavior are hard due to the exponentially growing size of Hilbert space. In contrast, sufficiently large quantum computers could simulate this behavior efficiently.

Smoothly varying some time-dependent Hamiltonian appears straightforward but contrasts with the traditional picture of discrete unitary operations including fault tolerant quantum circuit constructions [15]. Fortunately, we can approximate a smoothly varying Hamiltonian using methods of quantum simulations [16] and recast adiabatic evolution in terms of unitary operations.

Discretizing a continuous Hamiltonian is a straightforward process and changes the run time T of the adiabatic

algorithm only polynomially [14]. For simplicity, let the discrete time Hamiltonian $H[m]$ be a linear interpolation from some beginning Hamiltonian $H[0] = H_b$ to some final problem Hamiltonian $H[M] = H_p$ such that $H[M] = (m/M)H_p + (1 - m/M)H_b$. The unitary evolution of the discrete algorithm can be written as

$$U = \prod_m U_m = \prod_m e^{-i[(1-m/M)H_b + (m/M)H_p]\Delta t}, \qquad (1)$$

where $\Delta t = T/(M + 1)$, and $M + 1$ is the total number of discretization steps. The adiabatic limit is achieved when both $T, M \to \infty$ and $\Delta t \to 0$.

Full control over the strength of H_b and H_p is needed to implement Eq. (1). However, this may not necessarily be a realistic experimental assumption. We will next show how the discrete time adiabatic algorithm can still be implemented when H_b and H_p cannot both be applied simultaneously *and* when they are both fixed in strength.

When both H_b and H_p are fixed, we can approximate U_m to second order by using the Trotter formula $\exp[(A + B)\Delta t] = \exp(A\Delta t/2)\exp(B\Delta t)\exp(A\Delta t/2) + \mathcal{O}(\Delta t^2)$ [16]. Higher order approximations can be constructed if more accuracy is required.

Now suppose H_b and H_p are both constant. Since any unitary matrix is generated by an action $-iH\Delta t$, we can increase the effect of a constant Hamiltonian H by lengthening the time Δt. Thus, we can implicitly increase the strength of H_b and H_p even when they are constant by simply increasing the time during which they are applied.

This technique also allows cases when the accessible Hamiltonians are not of the required strength, for example, when we are given $H_b' = gH_b$ and $H_p' = hH_p$ but still wish to implement H_b and H_p. Using all of the described techniques, we can now write U_m as

$$U_m \approx e^{-iH_b'[(1-m/M)\Delta t/2g]} \circ e^{-iH_p'[(m/M)\Delta t/h]}, \qquad (2)$$

where $A \circ B = ABA$. Each discretization step is of length $(1 - m/M)\Delta t/g + (m/M)\Delta t/h$, which is not constant when $g \neq h$. As an illustration consider Fig. 1(a).

We choose $\Delta t = T/(M + 1)$ to be constant as we vary the number of discretization steps $M + 1$. This way, the total run time T increases with $M + 1$, allowing us to test the behavior of the algorithm when approaching one of the conditions for the adiabatic limit. Even when the discrete approximation is not close to the adiabatic limit, the implemented algorithm can often find solutions using relatively few steps but lacks the guaranteed performance of the adiabatic theorem [17].

Adiabatic evolution has been proposed to solve general optimization problems, including *NP*-complete ones. In this general setting, the algorithm can depend on the existence of a black box function or the usage of large amounts of workspace. Our goal here is to optimize a hard natural problem in a way that avoids these difficulties. We will first describe which problem we chose and later explain why it does not require ancilla qubits.

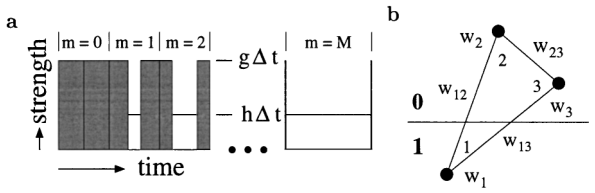

FIG. 1. (a) Illustration of Eq. (2). The shaded and clear boxes denote the strength and duration of the Hamiltonians H_b and H_p, respectively. (b) Illustration of a graph consisting of three nodes and three edges. The edges carry weights w_{12}, w_{13}, and w_{23}. When $\min(w_{ij}) = w_{23}$ as indicated by the length of the edges, the MAXCUT corresponds to the drawn cut. The solution is therefore $s = 100$ and also $s = 011$ due to symmetry. This symmetry can be broken by assigning the weights w_1, w_2, and w_3 to the nodes.

We found the MAXCUT problem to be a well-suited problem to demonstrate an adiabatic quantum algorithm because it allows a variety of interesting test cases. It also appears in the study of spin glasses [18], among others. The decision variant of the MAXCUT problem is part of the core *NP*-complete problems [12], and even the approximation within a factor of 1.0624 of the perfect solution is *NP* complete [19].

The MAXCUT problem can be understood as follows. A *cut* is defined as the partitioning of an undirected n-node graph with edge weights into two sets. We define the payoff as the sum of weights of edges crossing the cut. The maximum cut is a cut that maximizes this payoff. By assigning either $s_i = 0$ or $s_i = 1$ to each node i, depending on its location with respect to the cut, the MAXCUT problem can be restated as finding the n-bit number s that maximizes the payoff. An extension of the MAXCUT problem is to let the nodes themselves carry weights, which can be regarded as the nodes having a preference on their location. As an illustration consider a graph with three nodes as drawn in Fig. 1(b).

The payoff as a function of the cut defined by s is

$$P(s) = \sum_i w_i s_i + \sum_{i,j} s_i(1 - s_j)w_{ij}, \qquad (3)$$

where w_{ij} are the edge weights, w_i denotes the node weights, and s_i is the value of the ith bit of s.

The smallest meaningful test case of the MAXCUT problem requires three nodes and admits a variety of interesting cases by varying w_i and w_{ij}. We aimed at two goals when choosing a representative set of weights. First, we wanted the minimum energy gap $g_{\min}$ to be smaller than the one for a three-qubit adiabatic Grover search. Second, we wanted a resulting energy landscape with both a global and local maximum such that a greedy classical search would incorrectly find the local maximum half the time [20]. These goals are met by the choice

$w_1 = w_2 = w_3 = 2$, $w_{12} = 2$, $w_{13} = 1$, and $w_{23} = 3$. The payoff function for this set of weights is $P(s) = [0\ 6\ 7\ 7\ 5\ \mathbf{9}\ 8\ 6]$, where $s = [000\ 001\ 010\ 011\ 100\ 101\ 110\ 111]$. The global maximum lies at $s = 101$ so the answer on the quantum computer following measurement should be $|101\rangle$, and not at the local maximum $s = 110$

In the quantum setting, this payoff function $P(s)$ can be encoded into the Hamiltonian H_p by rewriting Eq. (3) using Pauli matrices:

$$H_p = \sum_i w_i(I - \sigma_{zi})/2 + \sum_{i<j} w_{ij}(I - \sigma_{zi}\sigma_{zj})/2, \quad (4)$$

where I is the $2^n \times 2^n$ identity matrix and σ_{zi} is the Pauli Z matrix on spin i. The identity matrices in the equation above only lead to an overall phase which cannot be observed and, hence, they can be ignored. The diagonal values of Eq. (4) are equal to $P(s)$. Because of the direct encoding of $P(s)$ into H_p, no black box function or ancilla qubits are required, which makes this a full implementation of an optimization problem.

Similar to Eq. (4), the natural Hamiltonian of n weakly coupled spin-1/2 nuclei subject to a static magnetic field B_0 is well approximated by [21]

$$\mathcal{H} = -\sum_i \omega_i\sigma_{zi}/2 + \sum_{i<j} \pi J_{ij}\sigma_{zi}\sigma_{zj}/2 + \mathcal{H}_{\text{env}}, \quad (5)$$

where the first term represents the Larmor precession of each spin i about $-B_0$, and ω_i is its Larmor frequency. The second term describes the scalar spin-spin coupling of strength J_{ij} between spins i and j. The last term represents coupling to the environment, causing decoherence. Note the resemblances between $\mathcal{H}$ and H_p.

Despite the similarities, the spin-spin couplings of Eq. (5) are generally different from a randomly chosen set of weights. Therefore, we require a procedure to turn the fixed J_{ij} into any specified weights w_{ij}. This is achieved using refocusing schemes that are typically used to turn on only one of the couplings while turning all others off [21].

We have modified a refocusing scheme to effectively change the couplings to any arbitrary value. Consider the pulse sequence drawn in Fig. 2. Based on this scheme, we can derive the underconstrained system $(\alpha + \beta - \gamma - \delta)J_{12} = w_{12}$, $(\alpha - \beta - \gamma + \delta)J_{13} = w_{13}$, and $(\alpha - \beta + \gamma - \delta)J_{23} = w_{23}$, which can be solved for positive α, β, γ, and δ such that $J_{ij} \rightarrow w_{ij}$.

The single weights w_i are implemented by introducing a reference frame for each spin i which rotates about $-B_0$ at frequency $(w_i - w_i)/2$. In order to apply the single qubit rotations of our refocusing scheme on resonance, we apply the reference frequency shift only during the delay segment α, which we can always choose to be a positive value. Thus, H_p is implemented by applying the refocusing scheme from Fig. 2 while going off resonance during the delay segment α.

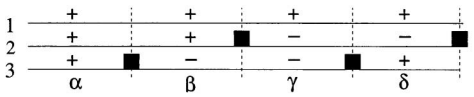

FIG. 2. Refocusing scheme to effectively change J_{ij} into w_{ij}. The horizontal lines denote qubits 1, 2, and 3 and time goes from left to right. The black rectangles represent 180° rotations. The delay segments are of length α, β, γ, and δ. When all segments are of equal length, all couplings are effectively turned off [22] because $\sigma_{xi}e^{-i\sigma_{zi}\sigma_{zj}t}\sigma_{xi} = e^{i\sigma_{zi}\sigma_{zj}t}$. In our experiment, $\alpha = 0.42$ ms, $\beta = 0$ ms, $\gamma = 4$ ms, and $\delta = 2.9$ ms in the last slice $M + 1$. The rf pulses that implement Hb' perform 33.75° rotations on the qubits in the first slice.

A full implementation of an adiabatic algorithm also requires a proper choice of H_b. We choose $H_b = \sum_i \sigma_{xi}$ for several reasons. First, its highest two excited states are nondegenerate. Second, it can be easily generated using single qubit rotations. Third, its highest excited state is created from a pure state with all qubits in the $|0\rangle$ state by applying a Hadamard gate on all qubits (we require the initial state to be the *highest excited* state of H_b because we are optimizing for the *maximum* value of H_p).

The full adiabatic quantum algorithm is now implemented by first creating the highest excited state of H_b. We then apply $M + 1$ unitary matrices as given by Eq. (2) and illustrated by Fig. 1(a). Accordingly, from slice to slice, we decrease the time during which H_b is active while increasing the time during which H_p is active. Finally, we measure the quantum system and read out the answer.

We selected ^{13}C-labeled $CHFBr_2$ for our experiments [10]. The Hamiltonian of the ^{1}H-^{19}F-^{13}C system is of the form of Eq. (5) with measured couplings $J_{HC} = 224$ Hz, $J_{HF} = 50$ Hz, and $J_{FC} = -311$ Hz. Experiments were carried out at MIT using an 11.7 Tesla Oxford Instruments magnet and a Varian Unity Inova spectrometer with a triple resonance (H-F-X) probe from Nalorac.

The experiments were performed at room temperature at which the thermal equilibrium state is highly mixed and cannot be turned into the required initial state by just unitary transforms. We thus first created an approximate effective pure state as in Ref. [10] by summing over three temporal labeling experiments.

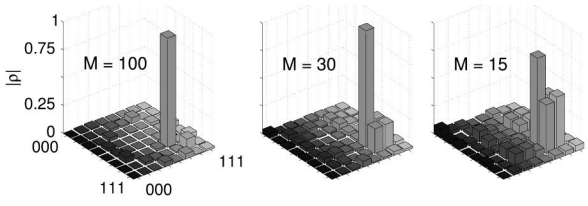

FIG. 3. Plot of the absolute value of the deviation density matrix for $M = 100$ ($T = 374$ ms), $M = 30$ ($T = 115$ ms), and $M = 15$ ($T = 59.2$ ms), adjusted by an identity portion such that the minimum diagonal value equals zero. The scale is arbitrary but the same for each plot.

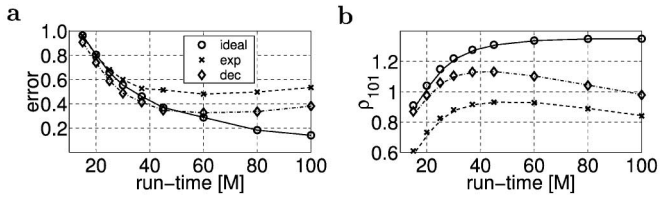

FIG. 4. Experimental performance of the adiabatic algorithm. (a) Plot of the error as a function of M. The error measure is the trace distance $D(\rho, \sigma) = |\rho - \sigma|/2$, where σ is the traceless deviation density matrix for $M = 400$, approximating $M \to \infty$, and ρ equals the ideal expected ($\bigcirc$), the experimentally obtained ($\times$), or the ideal expected traceless deviation density matrix with decoherence effects ($\diamond$) [6]. The minimum error occurs at about $M = 60$ indicating an optimal run time of the algorithm. (b) A similar observation can be made when plotting $|101\rangle\langle 101|$ as a function of M.

In our experiments, we actually implemented $0.5H_p$ and $0.5887H_b$ instead of H_p and H_b. This ensures that the error due to the second order Trotter approximation is sufficiently small. We also choose g so the applied rf field does not heat the sample, and $g \gg h$ so J_{ij} can be ignored when applying H_b. All of these choices result in a total experimental time that is within the shortest T_2 decoherence time [10]. We reconstructed the traceless deviation density matrices upon completion of the experiments using quantum state tomography [10].

We executed this algorithm for several M [with w_i and w_{ij} as listed above Eq. (4)]. Since we chose Δt to be constant, this meant increasing the run time T of the algorithm. The reconstructed deviation density matrices are shown in Fig. 3. The plots clearly display the expected pure state $|101\rangle$. The local maximum at $s = 110$ has a decreasingly small probability of being measured for increasing M. Simulations using Eq. (2) show that this optimization algorithm performs better for increasing M. We wanted to verify whether this is indeed true experimentally.

For this purpose, we estimate the error of our obtained deviation density matrices compared with the ideal case of $M = \infty$. Figure 4(a) plots the trace distance as a function of M, using the same arbitrary scale as in Fig. 3. From the plot, we observe there exists an optimal run time of the algorithm, corresponding to 0.226 s in our experiment. This optimal run time is in good agreement with the prediction of a previously developed simple decoherence model [6]. Predicting the impact of decoherence has already provided invaluable insight into estimating errors in previous experiments [6], and we believe continued effort towards understanding decoherence will greatly benefit experimental investigations of quantum systems.

In conclusion, we have provided the first experimental demonstration of an adiabatic quantum optimization algorithm. We show a concrete procedure turning a continuous time adiabatic quantum algorithm into a discrete time version, even when certain restrictions apply to the accessible Hamiltonians. Our results indicate that there exists an optimal run time of the algorithm which can be roughly predicted using a simple decoherence model. We believe this implementation opens the door to a variety of interesting experimental demonstrations and investigations of adiabatic quantum algorithms.

We wish to thank A. Childs, A. Landahl, and E. for useful discussions. This work was supported by the NSF Grant No. CCR-0122419, the DARPA QuIST program, and the HP/MSRI grant.

*Electronic address: msteffen@snowmass.stanford.edu

[1] P. Shor, in *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (IEEE, Los Alamitos, CA, 1994), p. 124.
[2] L. K. Grover, Phys. Rev. Lett. **79**, 325 (1997).
[3] E. Farhi *et al.*, quant-ph/0001106.
[4] T. Hogg, Phys. Rev. A **61**, 052311 (2000).
[5] D. G. Cory *et al.*, Fort. Phys. **48**, 875 (2000); L. M. K. Vandersypen, Ph.D. thesis, Stanford University, 2001, and references therein.
[6] L. M. K. Vandersypen *et al.*, Nature (London) **414**, 883 (2001).
[7] X. Peng *et al.*, Phys. Rev. A **65**, 042315 (2002).
[8] N. Gershenfeld and I. L. Chuang, Science **275**, 350 (1997); D. Cory, A. F. Fahmy, and T. F. Havel, Proc. Natl. Acad. Sci. U.S.A. **94**, 1634 (1997).
[9] E. Farhi *et al.*, Science **292**, 472 (2001).
[10] L. M. K. Vandersypen *et al.*, Appl. Phys. Lett. **76**, 646 (2000).
[11] W. W. Rhim, A. Pines, and J. S. Waugh, Phys. Rev. Lett. **25**, 218 (1970).
[12] M. R. Garey, D. S. Johnson, and L. Stockmeyer, Theor. Comput. Sci. **1**, 237 (1976).
[13] A. Messiah, *Quantum Mechanics* (Wiley, New York, 1976).
[14] W. van Dam, M. Mosca, and U. Vazirani, in *Proceedings of the 42nd Annual Symposium on FOCS* (IEEE, Las Vegas, NV, 2001), p. 279.
[15] A. Aharonov and M. Ben-Or, in *Proceedings of the 29th Annual ACM STOC* (ACM, New York, 1997), p. 176.
[16] H. F. Trotter, Pacific J. Math. **8**, 887 (1958).
[17] T. Hogg, quant-ph/0206059.
[18] F. Barahona, J. Phys. A Math. Gen. **18**, L673 (1985).
[19] G. Ausiello *et al.*, *Complexity and Approximation. Combinatorial Optimization Problems and Their Approximability Properties* (Springer-Verlag, Berlin, 1999).
[20] A greedy search is done by first choosing a random node configuration s, and then repeatedly moving to a new configuration s' which differs from the previous configuration by only one node value s'_i and which also has the highest payoff, until the payoff is maximized.
[21] R. Ernst, *Principle of Nuclear Magnetic Resonance in One and Two Dimensions* (Oxford University Press, New York, 1994).
[22] D. W. Leung *et al.*, Phys. Rev. A **61**, 042310 (2000).