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 Chuang1

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

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.

Since the discovery of Shor’s1 and Grover’s2 algorithms, the quest of finding new quantum algorithms proved a formidable challenge. Recently however, a novel algorithm was proposed, using adiabatic evolution3,4. 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 algorithm5,6, the Deutsch-Jozsa algorithm5,7,8, order-finding9,10, and Shor’s algorithm11. Recently, Hogg’s algorithm was implemented using only one computational step12, 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) techniques13,14. 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 algorithms15. Compared to earlier implementations of search problems5,6, 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 techniques16. 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 optimization problem given a fixed natural system Hamiltonian.

We provide a concrete procedure detailing these methods. We then apply the results to our optimization problem which is known as Maximum Cut or MAXCUT17. Our experimental results indicate there exists an optimal total running time which can be predicted using a decoherence model that is based on independent stochastic relaxation of the spins.

The evolution of a quantum state during an adiabatic quantum algorithm is determined by 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 degenerate18. 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. The total run-time \( T \) of the adiabatic algorithm scales as \( g_{\text{min}}^2 \) where \( g_{\text{min}} \) is the minimum separation between the lowest two energy eigenvalues of \( H(t) \). It is the scaling behavior of \( g_{\text{min}} \) that 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 it contrasts with the traditional picture of discrete unitary operations including fault tolerant quantum circuit constructions19. Fortunately, we can approximate a smoothly varying Hamiltonian using methods of quantum simulations20 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 polynomially18. For simplicity, let the discrete time Hamiltonian \( H[m] \) be a linear interpolation from some beginning Hamiltonian \( H[0] = H_0 \) to some final problem Hamiltonian \( H[M] = H_p \) such that

\[
H[m] = (m/M)H_p + (1 - m/M)H_0.
\]

The unitary evolu-
tion of the discrete algorithm can be written as:

\[ U = \prod_{m} U_m = \prod_{m} e^{-i((1-m/M)H_0+(m/M)H_p)\Delta t} \]  

(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 \( H_0 \) and \( H_p \) is needed to implement Eq. \( \text{1} \). 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 \( H_0 \) and \( H_p \) cannot both be applied simultaneously and when they are both fixed in strength.

When both \( H_0 \) and \( H_p \) are fixed, we can approximate \( U_m \) to second order by using the Trotter formula

\[ \exp((A+B)\Delta t) = \exp(A\Delta t/2)\exp(B\Delta t)\exp(A\Delta t/2) + \mathcal{O}(\Delta t^2) \]  

(2)

Higher order approximations can be constructed if more accuracy is required.

Now suppose \( H_0 \) and \( H_p \) are both constant. Since any unitary matrix is generated by an action \( -i\hbar \Delta t \), we can increase the effect of a constant Hamiltonian by lengthening the time \( \Delta t \). Thus, we can implicitly increase the strength of \( H_0 \) 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 \( H_0' = gH_0 \) and \( H_p' = hH_p \) but still wish to implement \( H_0 \) and \( H_p \). Using all of the described techniques, we can now write \( U_m \) as:

\[ U_m \approx e^{-iH_0'[1-m/M]\Delta t/2g} \circ e^{-iH_p'[m/M]\Delta t/h} \]  

(3)

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.

In this experiment 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.

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 on explain why it does not require ancilla qubits.

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 has applications in the study of spin glasses and VLSI design, among others. The decision variant of the MAXCUT problem is part of the core NP-complete problems and even the approximation within a factor of 1.0024 of the perfect solution is NP-complete.

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.

The payoff as a function of the cut defined by \( s \) is given by

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

(3)

where \( w_{ij} \) are the edge weights, \( w_i \) denotes the preference of the nodes to be on the 1 side of the cut, and \( s_i \) is the value of the \( i \)-th bit of \( s \), for \( 0 \leq s \leq 2^n - 1 \).

The smallest meaningful test case of the MAXCUT problem requires 3 nodes and admits a variety of interesting cases by varying \( w_{12}, w_{13} \), and \( w_{23} \). When \( \min(w_{12}) = w_{23} \) 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.
this set of weights is \( P(s) = [0 \ 6 \ 7 \ 7 \ 5 \ 9 \ 8 \ 6] \) 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. (4) using Pauli matrices:

\[
H_p = 
\sum_i w_i (I - \sigma_{z_i})/2 + \sum_{i<j} w_{ij} (I - \sigma_{z_i} \sigma_{z_j})/2 \tag{4}
\]

where \( I \) is the \( 2^n \times 2^n \) identity matrix and \( \sigma_{z_i} \) 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 \( \Delta \)

\[
\mathcal{H} = -\sum_{i<j} \pi J_{ij} \sigma_{z_i} \sigma_{z_j} / 2 + \mathcal{H}_{\text{env}} \tag{5}
\]

where the first term represents the Larmor precession of each spin \( i \) about \(-B_0\), and \( \omega_i \) 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. (4) 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.

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 under-constrained system \((\alpha + \beta - \gamma - \delta)J_{12} = w_{12}, (\alpha - \beta - \gamma + \delta)J_{13} = w_{13}, (\alpha - \beta + \gamma - \delta)J_{23} = w_{23}\), which can be solved for positive \( \alpha, \beta, \gamma, \) and \( \delta \) 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 \((\omega_i - w_i)/2\). 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 \( \alpha \), 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 \( \alpha \).

A full implementation of an adiabatic algorithm also requires a proper choice of \( H_b \). We choose \( H_b = \sum_i \sigma_{x_i} \) for several reasons. First, its highest two excited states are non-degenerate. Second, it can be easily generated using single qubit rotations, and 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. (4) and illustrated by Fig. 1d. 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 \(^{13}\text{C}\)-labeled CHFBr2 for our experiments. The Hamiltonian of the \(^{1}H-^{19}\text{F}\)\(^{13}\text{C}\) system is of the form of Eq. (4) with measured couplings \( J_{HF} = 224 \text{ Hz} \), \( J_{HC} = 50 \text{ Hz} \), and \( J_{FC} = -311 \text{ Hz} \). The interaction with the Br nuclei is averaged out, contributing only to \( \mathcal{H}_{\text{env}} \). 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. 4 by summing over three temporal labeling experiments.

In our experiments, we actually implemented 0.5\( H_p \) and 0.5887\( H_b \) instead of \( H_p \) and \( H_b \). This ensures that the error due to the 2nd order Trotter approximation is bounded by \( \sqrt{\sum_i |e_i|^2} < 0.0356 \) where \( e_i \) is the contribution of the \( i \)-th undesired Pauli matrix. 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 \( T_2^\ast \) decoherence time \( 9 \). We recon-
structured the traceless deviation density matrices upon completion of the experiments using quantum state tomography.

We executed this algorithm for several $M$ (with $w_i$ and $w_{ij}$ as listed above Eq. 4). Since we chose $\Delta t$ 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 even if the scale is arbitrary but the same for each plot.

For this purpose, we estimate the error of our obtained deviation density matrices compared with the ideal case of $M = \infty$. Fig. 1 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 seconds in our experiment. This optimal run time is in good agreement with the prediction of a previously developed simple decoherence model (11). Predicting the impact of decoherence has already provided invaluable insight into estimating errors in previous experiments (11), 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.

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.

We wish to thank A. Childs, A. Landahl, and E. Farhi for useful discussions. This work was supported by the NSF grant CCR-0122419, the DARPA QuIST program, the HP/MSRI postdoc fellowship.

\* Electronic address: msteffen@snowmass.stanford.edu

[1] P. Shor, in Proceedings of the 35th Annual Symposium on Foundations of Computer Science (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] J. Jones, M. Mosca, and R. H. Hansen, Nature 393, 344 (1998).
[6] L. M. K. Vandersypen et al., Appl. Phys. Lett. 76, 646 (2000).
[7] I. L. Chuang et al., Nature 393, 143 (1998).
[8] R. Marx et al., Phys. Rev. A 62, 012310 (2000).
[9] J. Jones and M. Mosca, J. Chem. Phys. 109, 1648 (1998).
[10] L. M. K. Vandersypen et al., Phys. Rev. Lett. 85, 5452 (2000).
[11] L. M. K. Vandersypen et al., Nature 414, 883 (2001).
[12] X. Peng et al., Phys. Rev. A 65, 042315 (2002).
[13] N. Gershenfeld and I. L. Chuang, Science 275, 350 (1997).
[14] D. Cory, A. F. Fahmy, and T. F. Havel, Proc. Natl. Acad. Sci. U.S.A. 94, 1634 (1997).
[15] E. Farhi and J. Goldstone, Science 292, 472 (2001).
[16] W. W. Rhim, A. Pines, and J. S. Waugh, Phys. Rev. Lett. 25, 218 (1970).
[17] M. R. Garey, D. S. Johnson, and L. Stockmeyer, Theoretical Computer Science 1, 237 (1976).
[18] A. Messiah, Quantum Mechanics (Wiley, 1976).
[19] W. van Dam, M. Mosca, and U. Vazirani, in Proceedings of the 42nd Annual Symposium on FOCS (2001), p. 279.
[20] A. Aharonov and M. Ben-Or, in Proceedings of the 29th Annual ACM Symposium on the Theory of Computing (1997), p. 176.
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] H. F. Trotter, Pacific J. Math. \textbf{8}, 887 (1958).
[22] T. Hogg, quant-ph/0206059.
[23] F. Barahona, J. Phys. Math: Math. Gen. \textbf{18}, L673 (1985).
[24] J. D. Cho et al., IEEE Trans. Comp. \textbf{47}, 1253 (1998).
[25] G. Ausiello et al., \textit{Complexity and Approximation. Combinatorial optimization problems and their approximability properties} (Springer Verlag, 1999).
[26] 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.
[27] R. Ernst, \textit{Principle of Nuclear Magnetic Resonance in One and two Dimensions} (Oxford University Press, 1994).
[28] D. W. Leung et al., Phys. Rev. A \textbf{61}, 042310 (2000).