

# Adiabatic quantum computing for random satisfiability problems

Tad Hogg

HP Labs, Palo Alto, California 94304

(Received 10 June 2002; revised manuscript received 20 November 2002; published 28 February 2003)

The discrete formulation of adiabatic quantum computing is compared with other search methods, classical and quantum, for random satisfiability (SAT) problems. With the number of steps growing only as the cube of the number of variables, the adiabatic method gives solution probabilities close to 1 for problem sizes feasible to evaluate via simulation on current computers. However, for these sizes the minimum energy gaps of most instances are fairly large, so the good performance scaling seen for small problems may not reflect asymptotic behavior where costs are dominated by tiny gaps. Moreover, the resulting search costs are much higher than for other methods. Variants of the quantum algorithm that do not match the adiabatic limit give lower costs, on average, and slower growth than the conventional GSAT heuristic method.

DOI: 10.1103/PhysRevA.67.022314

PACS number(s): 03.67.Lx

## I. INTRODUCTION

Quantum computers [1–4] can rapidly evaluate all search states of nondeterministic polynomial (NP) problems [5], but appear unlikely to give short worst-case solution times [6]. Of more practical interest is whether their average performance improves on conventional heuristics.

Adiabatic quantum computing, using a slowly changing time-dependent Hamiltonian, appears to give polynomial average cost growth for some NP combinatorial search problems [7]. These observations, while encouraging, are limited to small problems for which other methods, both conventional and quantum, can have even lower costs. Furthermore, although adiabatic methods apparently show exponential cost scaling for set partitioning [8] and finding the ground state of spin glasses [9], the typical performance of adiabatic quantum computing for large NP search problems remains an open question. Thus it is of interest to compare the adiabatic method with other techniques for NP problems having a well-studied class of hard instances.

This paper provides such a comparison for  $k$ -satisfiability ( $k$ -SAT), consisting of  $n$  Boolean variables and  $m$  clauses. A clause is a logical OR of  $k$  variables, each of which may be negated. A solution is an assignment, i.e., a value, true or false, for each variable, satisfying all the clauses. An example 2-SAT instance with three variables and two clauses is  $v_1$  OR (NOT  $v_2$ ) and  $v_2$  OR  $v_3$ , which has four solutions, e.g.,  $v_1 = v_2 = \text{false}$  and  $v_3 = \text{true}$ . For a given instance, let the cost  $c(s)$  of an assignment  $s$  be the number of clauses it does not satisfy.

For  $k \geq 3$ ,  $k$ -SAT is NP-complete [5], i.e., among the most difficult NP problems in the worst case. For average behavior we use the random  $k$ -SAT ensemble, in which the  $m$  clauses are selected uniformly at random. That is, for each clause, a set of  $k$  variables is selected randomly, and each selected variable is negated with probability 1/2. The algorithms we consider are probabilistic, so cannot definitively determine that no solution exists. Thus we use soluble instances: after random generation, we solve the instances with an exhaustive conventional method and only retain those with a solution. This ensemble has a high concentration of hard instances near a phase transition in search difficulty

[10–13]. For 3-SAT, we generate instances near this transition by using  $\mu \equiv m/n = 4.25$ , although, for those  $n$  not divisible by 4, half the samples had  $m = \lfloor 4.25n \rfloor$  and half had  $m$  larger by 1.

The remainder of this paper describes several quantum search algorithms in the context of satisfiability problems, and then compares their behavior.

## II. ALGORITHMS

The adiabatic technique [7] is based on two Hamiltonians  $H^{(0)}$  and  $H^{(c)}$ . The first is selected to have a known ground state, while the ground states of  $H^{(c)}$  correspond to the solutions of the problem instance to be solved. The algorithm continuously evolves the state of the quantum computer using  $H(f) = (1-f)H^{(0)} + fH^{(c)}$  with  $f$  ranging from 0 to 1. Under suitable conditions, i.e., with a nonzero gap between relevant eigenvalues of  $H(f)$ , the adiabatic theorem guarantees that, with sufficiently slow changes in  $f$ , the evolution maps the ground state of  $H^{(0)}$  into a ground state of  $H^{(c)}$ , so a subsequent measurement gives a solution. The choices of  $H^{(0)}$  and  $H^{(c)}$  and how  $f$  varies as a function of time are somewhat arbitrary.

In matrix form, one Hamiltonian with minimal-cost assignments as ground states is  $H_{r,s}^{(c)} = c(s)\delta_{r,s}$ , for assignments  $r$  and  $s$ , where  $\delta_{r,s}$  is 1 if  $r=s$  and 0 otherwise. This Hamiltonian introduces a phase factor in the amplitude of assignment  $s$  depending on its associated cost  $c(s)$ .

For  $H^{(0)}$ , we introduce a non-negative weight  $w_i$  for variable  $i$ , let  $\omega \equiv \sum_{i=1}^n w_i$ , and take

$$H_{r,s}^{(0)} = \begin{cases} \omega/2 & \text{if } r=s, \\ -w_i/2 & \text{if } r \text{ and } s \text{ differ only for variable } i, \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

This Hamiltonian can be implemented with elementary quantum gates by use of the Walsh-Hadamard transform  $W$ , with elements  $W_{r,s} = 2^{-n/2}(-1)^{r \cdot s}$  (treating the states  $r$  and  $s$  as vectors of bits so their dot product counts the number of variables assigned the value 1 in both states). Specifically,  $H^{(0)} = WDW$  where  $D$  is a diagonal matrix with the value for

state  $r$  given by the weighted sum of the bits:  $\sum_{i=1}^n w_i r_i$  with  $r_i$  representing the value of the  $i$ th bit of  $r$ . In particular, if all the weights equal 1,  $D_{r,r}$  just counts the number of bits equal to 1.

The adiabatic method is a continuous process. To compare with other algorithms, we use the algorithmically equivalent discrete formulation [14,15] acting on the amplitude vector initially in the ground state of  $H^{(0)}$ , i.e.,  $\psi_s^{(0)} = 2^{-n/2}$ . This formulation consists of  $j$  steps and a parameter  $\Delta$ . Step  $h$  is a matrix multiplication:

$$\psi^{(h)} = e^{-i\tau(f)H^{(0)}\Delta} e^{-i\rho(f)H^{(c)}\Delta} \psi^{(h-1)} \quad (2)$$

with the mixing phase function  $\tau(f) = 1 - f$ , cost phase function  $\rho(f) = f$ , and taking  $\hbar = 1$ . After these steps, the probability of finding a solution is  $P_{\text{soln}} = \sum_s |\psi^{(j)}|^2$ , with the sum over all solutions  $s$ .

As a simple choice for the evolution, we take  $f$  to vary linearly from 0 to 1. We exclude the steps with  $f=0$  and 1 since they have no effect on  $P_{\text{soln}}$ . Specifically, we take  $f = h/(j+1)$  for step  $h$ , ranging from 1 to  $j$ .

The expected number of steps required to find a solution is  $C = j/P_{\text{soln}}$ , providing a commonly used proxy for the computational cost of discrete methods, pending further study of clock rates for the underlying gate operations and the ability of compilers to eliminate redundant operations. As also observed with conventional heuristics, the cost distribution for random  $k$ -SAT is highly skewed, so a few instances dominate the mean cost. Instead, we use the median cost to indicate typical behavior. The time for the continuous formulation is  $T = j\Delta$ , so the adiabatic limit is  $j\Delta \rightarrow \infty$ . By contrast, in the discrete formulation,  $\Delta$  parametrizes the operators of Eq. (2) rather than determining the time required to perform them.

Equation (2) follows the continuous evolution  $\psi^{(h+1)} \approx e^{-iH(f)\Delta} \psi^{(h)}$  when  $\Delta \|H\| \rightarrow 0$ , which holds when  $\Delta \ll 1/n$  [14,15]. This last condition uses the fact that the norm  $\|H\|$  is the largest eigenvalue of  $H$ , which is  $O(n)$  since we consider  $k$ -SAT problems with  $m \propto n$ . As a specific choice, we use  $\Delta = 1/\sqrt{j}$ . Other scaling choices  $\Delta = 1/j^\alpha$  with  $0 < \alpha < 1$  give qualitatively similar behaviors to those reported here while maintaining correspondence with the continuous evolution for sufficiently large  $j$ .

The *unweighted*  $H^{(0)}$  uses equal weights:  $w_i = 1$  so  $\omega = n$ . Alternatively,  $w_i$  can be the number of times variable  $i$  appears in a clause [7], as also used by some conventional heuristics to adjust the importance of changes in each variable. This choice gives  $\omega = mk$ . By matching  $H^{(0)}$  to the problem instance, one might expect such weights to improve performance. Instead, for random 3-SAT these weights give *higher* costs  $C$ , requiring about  $n$  times as many steps to achieve the same  $P_{\text{soln}}$  as the unweighted choice. If instead these weights are normalized so their average value is 1, the performance is about the same as in the unweighted case, but still slightly worse. In light of these observations, we use the unweighted  $H^{(0)}$  in this paper.

We compare the adiabatic limit with two other methods, summarized in Table I. First, for the *discrete adiabatic* case we take  $\Delta$  independent of  $n$  and  $j$ , violating the condition

TABLE I. Summary of quantum search algorithms using problem structure. The heuristic method requires finding appropriate choices for the phase functions to give good performance and for the number of steps  $j$  to increase with problem size  $n$ . The adiabatic methods require sufficiently large values of  $T = j\Delta$ . A constant value for a parameter in this table means it is taken to be independent of  $n$  and  $j$ .

Algorithm	Parameter		Phase function
	$T$	$\Delta$	
Adiabatic	$T \rightarrow \infty$	$\Delta \rightarrow 0$	$\rho(0) = 0 = \tau(1)$
Discrete adiabatic	$T \rightarrow \infty$	Constant	$\rho(0) = 0 = \tau(1)$
Heuristic	Constant	$\Delta \rightarrow 0$	Suitable $\rho, \tau$

$\Delta n \rightarrow 0$  so Eq. (2) no longer closely approximates the continuous evolution and does not necessarily give  $P_{\text{soln}} \rightarrow 1$  as  $j \rightarrow \infty$ . In this case, a discrete version of the adiabatic theorem, described in the Appendix, ensures that  $P_{\text{soln}}$  is close to 1 if  $\Delta$  is not too large.

Second, the *heuristic* method, studied previously [16,17], has  $\Delta = 1/j$  and forms for  $\tau(f)$  and  $\rho(f)$  that do not range between 0 and 1. Instead, these phase functions must be selected appropriately to give good performance. Identifying such choices and characterizing their performance are major issues for this algorithm, although mean-field approximations based on a few problem parameters, e.g., the ratio  $m/n$  for  $k$ -SAT, can give reasonably good choices. This method does not correspond to the adiabatic limit:  $P_{\text{soln}}$  has a limit less than 1 as  $j \rightarrow \infty$ .

For all these techniques, the expected cost  $C = j/P_{\text{soln}}$  is minimized for intermediate values of  $j$  rather than taking  $j \rightarrow \infty$  as used in the limits listed in Table I. Identifying the parameters and phase functions  $\rho(f)$  and  $\tau(f)$  giving minimal cost for a given problem instance depends on details of the search space structure unlikely to be available prior to solving that instance. However, as described below, taking  $j$  to grow only as a fairly small power of  $n$  provides relatively modest costs, on average, for problem sizes feasible to simulate.

### III. BEHAVIOR

For the adiabatic method, Fig. 1 shows the median  $P_{\text{soln}}$  for various growth rates of the number of steps.  $P_{\text{soln}} \rightarrow 1$  as  $j$  increases. At least for  $n \leq 20$ ,  $P_{\text{soln}} \approx 1$  when  $j = n^3$ , so median costs are  $O(n^3)$ , a substantial improvement over all known classical methods if it continues for larger  $n$ . However, for smaller powers of  $n$ ,  $P_{\text{soln}}$  values decrease, but this is only evident for  $j = n^2$  for  $n > 20$ . This raises the possibility of such a decline, at somewhat larger  $n$ , for larger  $j$  as well. Provided such a decline only leads to  $P_{\text{soln}}$  decreasing as a power of  $n$ , corresponding to a straight line on the log-log plot of Fig. 1, median costs would still only grow as a power of  $n$ . The remainder of this section describes the algorithm behaviors in more detail.

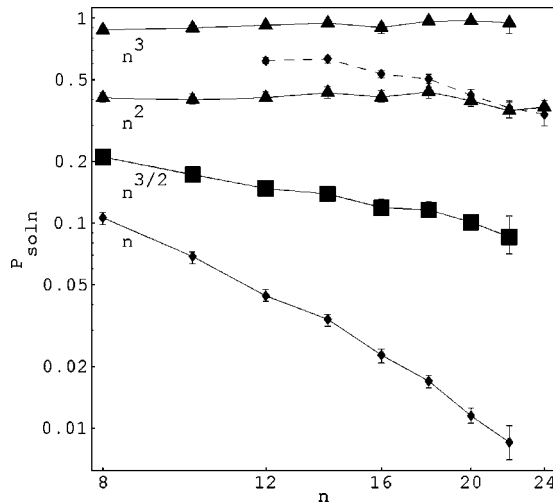


FIG. 1. Log-log plot of median  $P_{\text{soln}}$  for the adiabatic method vs  $n$  with the number of steps  $j$  equal to  $n$ , the integer nearest  $n^{3/2}$ ,  $n^2$ , and  $n^3$  (solid curves, from bottom to top, respectively). We use  $\Delta = 1/\sqrt{j}$ . For comparison, the dashed curve shows  $P_{\text{soln}}$  for the heuristic method using at most  $n$  steps. The error bars show the 95% confidence intervals [18] [p. 124] of the medians estimated from the random sample of instances. The same instances were solved with each method. We use 1000 instances for each  $n$  up to 20, and 500 for larger  $n$ , except only 100 for  $j = n^3$  for  $n \geq 16$ .

**A. Energy gap**

Asymptotically, the adiabatic method’s cost is dominated by the growth of  $1/G^2$  where  $G = \min_f g(f)$  and  $g(f)$  is the energy gap in  $H(f)$ , i.e., the difference between the ground state eigenvalue and the smallest higher eigenvalue corresponding to a nonsolution. Evaluation using sparse matrix techniques [19] for  $n \leq 20$  gives the median  $G$  in the range 0.3–0.5, as illustrated for one instance in Fig. 2, and, more significantly, it does not decrease over this range of  $n$ . This minimum is not much smaller than other values of  $g(f)$ . Hence, unlike for large  $n$ , the cost is not dominated by the minimum gap size and so the values of Fig. 1 may not reflect asymptotic scaling.

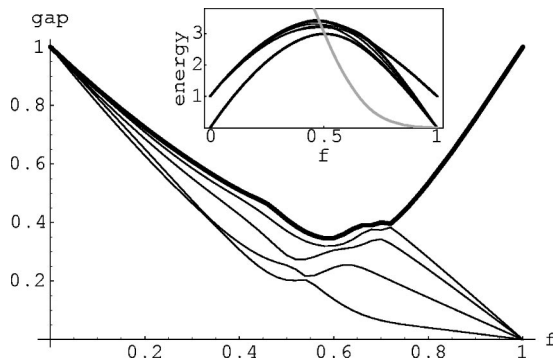


FIG. 2. Difference between eigenvalues of the lowest five excited states and the ground state vs  $f$  for an instance with  $n = 20$ ,  $m = 85$ , and five solutions. The inset shows the actual eigenvalues, with the gray curve showing the expected cost  $\langle c \rangle$  in the ground state.

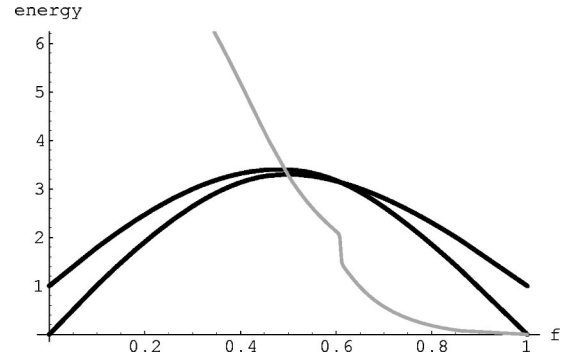


FIG. 3. Lowest two eigenvalues vs  $f$  for an instance with  $n = 20$ ,  $m = 85$ , one solution, and a particularly small minimum gap. The gray curve shows the expected cost  $\langle c \rangle$  in the ground state, equal to  $m/2^k = 10.625$  at  $f = 0$ . Note the abrupt drop at the location of the minimum gap. The ground state for  $f = 1$  is the solution whose cost is zero, so  $\langle c \rangle \rightarrow 0$  as  $f \rightarrow 1$ .

By contrast, Fig. 3 illustrates the behavior of an instance with a small minimum gap. One characterization of the eigenstates of  $H(f)$  is their expected cost, i.e.,  $\langle c \rangle_a = \sum_s c(s) |\phi_s^{(a)}(f)|^2$  where  $\phi^{(a)}(f)$  is the  $a$ th eigenvector of  $H(f)$ . In particular, for  $a = 1$  this gives the expected cost in the ground state, which we denote simply as  $\langle c \rangle$ . The expected cost in the ground state drops rapidly at the minimum gap location, in contrast to the smooth behavior for instances with larger gaps (as, for example, in Fig. 2). We thus see a difference in behavior of the ground state for instances with small gaps, presumably representative of typical behavior for larger  $n$ , and the behavior of more typical instances for  $n \approx 20$ .

With the adiabatic method and  $T$  sufficiently large, the actual state of the quantum computer after step  $h$ ,  $\psi^{(h)}$ , closely approximates the ground state eigenvector  $\phi^{(1)}$ , up to an irrelevant overall phase. Thus the computation will also show the jump in expected cost.

Detailed quantitative comparison of the typical behaviors due to small minimum gaps and conventional heuristics requires larger problem sizes. Nevertheless, we can gain some insight from instances with small gaps for  $n \approx 20$ , which tend to have high costs for both the quantum methods and conventional heuristics, such as GSAT [20], even when restricting comparison to problems with the same numbers of variables and solutions. For the instance shown in Fig. 3, GSAT trials readily reach states with one or two conflicts, but have a relatively low chance of finding the solution. This behavior, typical of conventional heuristics [21], corresponds to the abrupt drop in  $\langle c \rangle$  of Fig. 3. Thus finding assignments with costs below this value dominates the running time of both the quantum and conventional methods. These observations suggest that small energy gaps characterize hard problems more generally than just for the adiabatic method, which may provide useful insights into the nature of search along with quantities such as the backbone (i.e., variables with the same values in all solutions [13]).

Simple problems or algorithms ignoring problem structure allow determining the gap for large  $n$  [14,15,22]. This is difficult for random SAT. For instance, although random  $k$ -SAT corresponds to random costs for  $H^{(c)}$  and the extreme

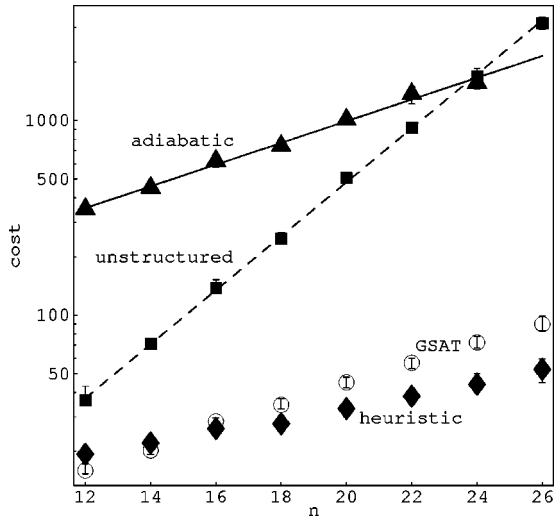


FIG. 4. Logarithmic plot of median search cost vs  $n$  for the heuristic (diamond) and unstructured (box) searches, GSAT with restarts after  $n$  steps (circle) and the adiabatic search with  $j=n^2$  (triangle). The values are based on the same instances as in Fig. 1. The lines are exponential fits to the unstructured (dashed) and adiabatic (solid) methods.

eigenvalues of random matrices can be determined when elements are chosen independently [23,24], the costs of nearby states for SAT instances are highly correlated since they likely conflict with many of the same clauses. Alternatively, upper [25] and lower [26] bounds for eigenvalues can be based on classes of trial vectors. For instance, vectors whose components for state  $s$  depend only on  $c(s)$  give fairly close upper bounds for the ground state of random 3-SAT, on average, as well as a mean-field approximation for the heuristic method [16]. However, simple lower bounds for higher energy states are below the upper bound for the ground state for some values of  $f$ , and so do not give useful estimates for  $G$ . Furthermore, typical soluble instances have exponentially many solutions (although still an exponentially small fraction of all states). Thus a full analysis of performance based on energy values must also consider the behavior of the many eigenvalues corresponding to solutions, which can be complicated, as illustrated in Fig. 2.

### B. Search cost

Even if  $n \leq 20$  does not identify asymptotic behavior, this range of feasible simulations allows comparing algorithm costs. Such comparisons are particularly relevant for quantum computer implementations with relatively few qubits and limited coherence times which are thus limited to small problems and few steps. Figure 4 compares the median values of the expected search costs  $C$ . For the adiabatic method, using  $j=n^3$  gives large costs, far higher than those of conventional heuristics and other quantum methods. Using just enough steps to achieve moderate values of  $P_{\text{soln}}$  reduces cost [7], e.g.,  $j=n^2$ . Alternatively, for each  $n$ , testing various  $j$  on a small sample of instances indicates the number of steps required to achieve a fixed value of  $P_{\text{soln}}$ , e.g.,  $1/8$ . In our case, the latter approach has median costs about 20%

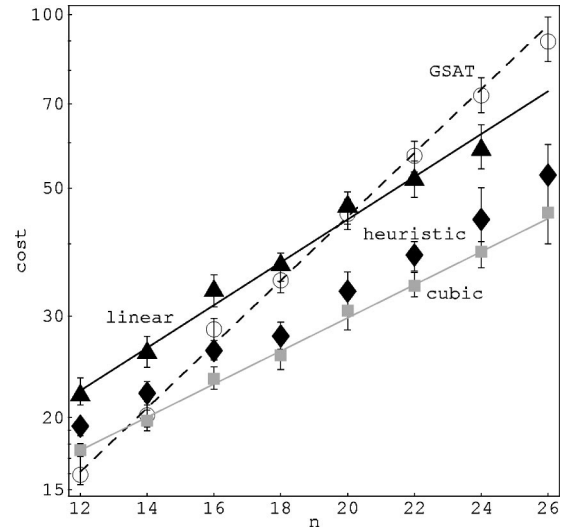


FIG. 5. Logarithmic plot of median search cost vs  $n$  for GSAT (circle), the heuristic method (diamond), both of which are also shown in Fig. 4, and two versions of the discrete adiabatic method:  $\Delta=1.2$  with linear phase functions (triangle) and the cubic polynomial variation with  $f$  (gray box) described in the text. The lines are exponential fits to GSAT (dashed) and the two discrete adiabatic quantum methods. The figure uses the same instances as Fig. 4.

lower than the former, but with the same cost growth rate. Because this improvement is minor compared to the differences with other algorithms shown in the figure, and to avoid the additional variability due to estimating  $j$  from a sample of instances, we simply take  $j=n^2$  to illustrate the adiabatic method.

The figure also shows Grover's unstructured search [27] (without prior knowledge of the number of solutions [28]) and the conventional heuristic GSAT [20]. The unstructured search cost grows as  $e^{0.32n}$ . The exponential fit to the adiabatic method is  $e^{0.13n}$ . This fit gives a residual about one-half as large as that from a power-law fit. The growth rate is about the same as that of GSAT.

Figure 4 shows that the heuristic search, using at most  $n$  steps, gives low costs due to its fairly high values for  $P_{\text{soln}}$  shown in Fig. 1. The constant  $\Delta$  scaling for the discrete adiabatic method also gives large  $P_{\text{soln}}$  values for  $j \approx n$ . Thus both  $\Delta=1/j$  and  $\Delta$  independent of  $j$  make better use of quantum coherence in the discrete formulation than the continuous adiabatic limit (with  $1/j \ll \Delta \ll 1/n$ ) for hard random 3-SAT. These behaviors are shown in Fig. 5.

Because these quantum methods and GSAT consist of a series of independent trials, they can be combined with amplitude amplification to give an additional quadratic performance improvement [29]. However, this is only a significant benefit when  $P_{\text{soln}}$  is fairly small, which is not the case for the heuristic and GSAT methods for these problem sizes.

For the adiabatic method, taking  $\rho(f)$  and  $\tau(f)$  in Eq. (2) to vary according to  $g(f)^2$  reduces costs [15,22]. This concentrates steps at values of  $f$  close to the minimum gap. While  $g(f)$  is costly to evaluate for SAT instances, using average values of  $g(f)$  based on a sample of instances gives some benefit. For example, for  $j=n^2$ ,  $P_{\text{soln}}$  increases from



around 0.4 shown in Fig. 1 to a range of 0.5–0.6 but this does not appear to reduce the cost’s growth rate.

Similar improvement occurs with constant  $\Delta$ . Optimizing  $\tau$  and  $\rho$  separately for each step on a sample of instances gives values close to a cubic polynomial in  $f$ . Restricting attention to such polynomials, for a set of 100  $n=12$  instances the best performance was with  $\Delta=1.31275$ ,  $\rho(f)=p(f)$ , and  $\tau(f)=1-p(f)$ , where  $p(f)=1.92708f-2.66179f^2+1.73471f^3$ . This cubic is similar to the functional form optimizing the adiabatic method for unstructured search [15,22]. Figure 5 shows the resulting cost reduction. Hence, tuning the algorithm to the problem ensemble is beneficial as also suggested by a mean-field analysis of the heuristic search [16].

The simulations also show that these quantum algorithms have a large performance variance among instances with given  $n$  and  $m$ , and no single choice for  $\rho$  and  $\tau$  is best for all problem instances. Thus portfolios [30] combining a variety of such choices can give further improvements.

#### IV. CONCLUSION

In summary, for random SAT, the adiabatic method improves on unstructured search and provides a general technique to exploit readily computed properties of hard search problems through the choice of Hamiltonians. However nonadiabatic-limit algorithms require fewer steps, comparable to GSAT, and appear to have slower cost growth. As a caveat, small energy gaps appear to be associated with instances difficult to solve with both quantum and classical methods. Thus the simulation results presented here, based on fairly small problem sizes for which most instances have fairly large energy gaps, may not reveal the asymptotic scaling of the typical search cost for hard random SAT problems. Evaluating the behavior of these algorithms and, more generally, identifying better ways to use state costs in quantum algorithms remain open questions.

Quantum computers with only a moderate number of qubits could test algorithms beyond the range of simulators, and hence provide useful insights even if the problem sizes are still readily solved by conventional heuristics. Such studies could help address the question of whether, with suitable tuning based on readily evaluated average properties of search states, the ability to operate on the entire search space allows quantum computers to effectively exploit weak correlations among state costs in ways classical machines cannot.

#### ACKNOWLEDGMENTS

I have benefitted from discussions with Rob Schreiber and Wim van Dam.

#### APPENDIX: DISCRETE ADIABATIC BEHAVIOR

When  $\Delta$  is held constant, the steps of Eq. (2) do not approximate the continuous evolution induced by  $H(f)$ , and hence  $\psi^{(h)}$  does not closely follow the ground state of  $H(f)$  when  $T \rightarrow \infty$ . Nevertheless,  $\psi^{(h)}$  does closely follow an eigenstate of the unitary operator involved in Eq. (2). This

discrete version of the adiabatic theorem ensures good performance of the algorithm provided the continuous change in the eigenvector takes the initial ground state into the final one, rather than into some other eigenvector.

#### 1. The discrete adiabatic limit

Consider a smoothly changing sequence of unitary matrices  $U(f)$  defined for  $0 \leq f \leq 1$  and vectors  $\psi^{(h+1)} = U(f)\psi^{(h)}$  with  $f=h/j$  for  $h=0, \dots, j-1$ . Let  $e^{-i\theta_r(f)}$  and  $\hat{e}_r(f)$  be the  $r^{\text{th}}$  eigenvalue and (normalized) eigenvector of  $U(f)$ .

We start with  $\psi^{(0)}$  equal to the eigenvector  $\hat{e}_1(0)$  of  $U(0)$ , which we assume to be nondegenerate for simplicity. Provided the difference between eigenvalues is bounded away from zero, for sufficiently large  $j$ ,  $\psi^{(j)}$  will be close to an eigenvector of  $U(1)$ . To see this let  $\epsilon=1/j$  and expand  $\psi^{(h)} = \sum_r c_r(f) \Lambda_r(f) \hat{e}_r(f)$  in the eigenbasis of  $U(f)$  where

$$\Lambda_r(h/j) \equiv \exp\left(-i \sum_{k=0}^{h-1} \theta_r(k/j)\right).$$

First order perturbation theory gives the change in the  $c_r$  values during one step to be  $O(\epsilon)$ . After  $j$  steps, it might appear that these changes could build up to  $O(\epsilon j) = O(1)$ . However, this is not the case due to the rapid variation in phases when  $j$  is large. Specifically, the changes in coefficients for  $r \neq 1$  are

$$\frac{dc_r}{df} = P_{1,r}(f) \Phi_r(f), \quad (\text{A1})$$

where  $P_{s,r}(h/j) \equiv e^{-ij\Theta_{s,r}(f)}$ ,

$$\Theta_{s,r}(f) \equiv \frac{1}{j} \sum_{k=0}^{h-1} [\theta_s(k/j) - \theta_r(k/j)],$$

and

$$\Phi_r \equiv \frac{\langle r | dU/df | 1 \rangle}{e^{-i\theta_r} - e^{-i\theta_1}}.$$

Since  $c_r(0)=0$ , Eq. (A1) gives

$$c_r(f) = \int_0^f e^{-ij\Theta_{1,r}(\kappa)} \Phi_r(\kappa) d\kappa.$$

As  $j$  increases, the integrand oscillates increasingly rapidly so the integral goes to zero as  $j \rightarrow \infty$  by applying the Riemann-Lebesgue lemma, since  $d\Theta_{1,r}/df = \theta_1 - \theta_r$  is non-zero and  $|\Phi_r(f)|$  is bounded for all  $f$  and  $r \neq 1$ , by the assumption of no level crossing. Hence  $c_r(f) \rightarrow 0$  so  $\psi^{(j)}$  approaches  $\hat{e}_1(1)$ , up to an overall phase factor, as  $j \rightarrow \infty$ .

#### 2. An example

An important caveat in applying this result to quantum algorithms is that, while  $j \rightarrow \infty$  suffices to ensure  $\psi^{(h)}$  closely follows the evolution of an eigenvector of  $U(f)$ , this evolu-

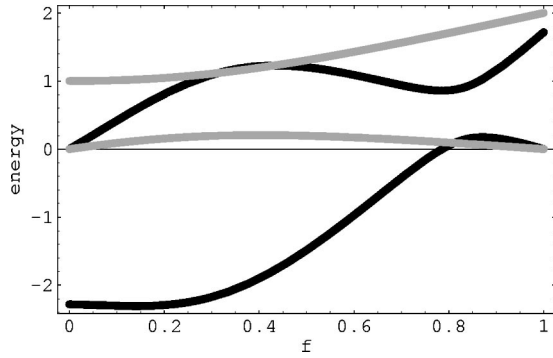


FIG. 6. Energy values  $\theta_r(f)$  corresponding to the two eigenvalues of  $U(f)$  vs  $f$  for  $\Delta=1$  (gray) and 4 (black). The values are defined only up to a multiple of  $2\pi$ , and we take  $-\pi < \theta \leq \pi$ . The ground states of  $H^{(0)}$  and  $H^{(c)}$  correspond to  $\theta(0)=0$  and  $\theta(1)=0$ , respectively. The values for  $\Delta=1$  are close to those of the combined Hamiltonian  $H(f)=(1-f)H^{(0)}+fH^{(c)}$ . However, the  $\Delta=4$  values do not remain close to those of  $H(f)\Delta$ .

tion may not lead to the desired eigenvector of  $U(1)$ , i.e., corresponding to solutions to the search problem. This is because the eigenvalues of  $U(f)$  lie on the unit circle in the complex plane and can “wrap around” as  $\Delta$  increases. Hence, in addition to ensuring that the eigenvalue gap does not get too small, good performance also requires selecting appropriate  $\Delta$ . Alternatively, one could start from a different eigenvector of  $U(0)$ , which would be useful if one could determine which eigenvector maps to the solutions.

One guarantee of avoiding this problem is that *none* of the eigenvalues of  $U(f)$  wrap around the unit circle, i.e.,  $\Delta\|H\|\rightarrow 0$ , corresponding to the continuous adiabatic limit. Simulations show that performance remains good for moderate values of  $j$  even if  $\Delta$  does not go to zero, provided  $\Delta$  is below some threshold value. For hard random 3-SAT prob-

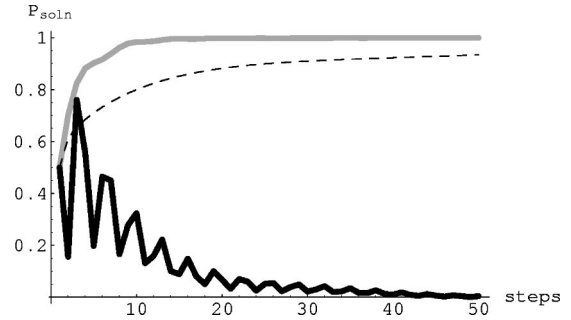


FIG. 7.  $P_{\text{soln}}$  vs  $j$  for  $\Delta=1$  (gray) and 4 (black). For comparison, the dashed curve uses  $\Delta=1/\sqrt{j}$  corresponding to the continuous adiabatic limit.

lems with  $j \propto n$ , this threshold appears to be somewhat larger than 1.

To illustrate these remarks consider the  $n=1$  example

$$H^{(0)} = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad H^{(c)} = \begin{pmatrix} 0 & 0 \\ 0 & 2 \end{pmatrix},$$

so  $U(f) = e^{-iH^{(0)}(1-f)\Delta} e^{-iH^{(c)}f\Delta}$ . Figure 6 shows the behavior of the two eigenvalues of  $U(f)$  for two values of  $\Delta$ . For  $\Delta=4$  the initial ground state eigenvector, with eigenvalue 1, evolves into the second eigenvector of  $U(1)$  rather than the eigenvector corresponding to the ground state of  $H^{(c)}$ .

Figure 7 shows the consequence of this behavior: when  $\Delta$  is too large,  $\psi^{(h)}$  follows the evolving eigenvector to the wrong state when  $f=1$ , giving  $P_{\text{soln}} \rightarrow 0$  as  $j \rightarrow \infty$ . As another observation from this figure,  $P_{\text{soln}}(j)$  exhibits oscillations (although they are quite small for  $\Delta=1$ ). With appropriate phase choices, these oscillations can be quite large, allowing  $P_{\text{soln}}$  to approach 1 with only a modest number of steps even when  $P_{\text{soln}}$  approaches 0 for larger  $j$ . This observation is the basis of the heuristic method.

- 
- [1] D. Deutsch, Proc. R. Soc. London, Ser. A **400**, 97 (1985).  
 [2] D. P. DiVincenzo, Science **270**, 255 (1995).  
 [3] R. P. Feynman, *Feynman Lectures on Computation* (Addison-Wesley, Reading, MA, 1996).  
 [4] A. Steane, Rep. Prog. Phys. **61**, 117 (1998).  
 [5] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness* (Freeman, San Francisco, 1979).  
 [6] C. H. Bennett, E. Bernstein, G. Brassard, and U. V. Vazirani, SIAM J. Comput. **26**, 1510 (1997).  
 [7] E. Farhi *et al.*, Science **292**, 472 (2001).  
 [8] V. N. Smelyanskiy, U. V. Toussaint, and D. A. Timucin, e-print quant-ph/0112143.  
 [9] G. E. Santoro *et al.*, Science **295**, 2427 (2002).  
 [10] P. Cheeseman, B. Kanefsky, and W. M. Taylor, in *Proceedings of IJCAI91*, edited by J. Mylopoulos and R. Reiter (Morgan Kaufmann, San Mateo, CA, 1991), pp. 331–337.  
 [11] S. Kirkpatrick and B. Selman, Science **264**, 1297 (1994).  
 [12] *Frontiers in Problem Solving: Phase Transitions and Complexity*, edited by T. Hogg, B. A. Huberman, and C. P. Williams, special issue of Artif. Intell. **81**, (1996).  
 [13] R. Monasson, R. Zecchina, S. Kirkpatrick, B. Selman, and L. Troyansky, Nature (London) **400**, 133 (1999).  
 [14] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Technical Report No. MIT-CTP-2936, MIT, 2000 (unpublished).  
 [15] W. van Dam, M. Mosca, and U. Vazirani, in *Proceedings of the 42nd Annual Symposium on Foundations of Computer Science (FOCS2001)*, edited by M. Naor (IEEE, New York, 2001), pp. 279–287.  
 [16] T. Hogg, Phys. Rev. A **61**, 052311 (2000).  
 [17] T. Hogg, e-print quant-ph/0104048.  
 [18] G. W. Snedecor and W. G. Cochran, *Statistical Methods*, 6th ed. (Iowa State University Press, Ames, IA, 1967).  
 [19] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods* (SIAM, Philadelphia, PA, 1998).  
 [20] B. Selman, H. Levesque, and D. Mitchell, in *Proceedings of the 10th National Conference on Artificial Intelligence (AAAI92)*, edited by P. Rosenbloom and P. Szolovits (AAAI Press, Menlo Park, CA, 1992), pp. 440–446.

- [21] J. Frank, P. Cheeseman, and J. Stutz, *J. Artif. Intell. Res.* **7**, 249 (1997).
- [22] J. Roland and N. J. Cerf, *Phys. Rev. A* **65**, 042308 (2002).
- [23] S. F. Edwards and R. C. Jones, *J. Phys. A* **9**, 1595 (1976).
- [24] Z. Furedi and K. Komlos, *Combinatorica* **1**, 233 (1981).
- [25] J. K. L. MacDonald, *Phys. Rev.* **43**, 830 (1933).
- [26] P.-O. Lowdin, *Phys. Rev.* **139**, A357 (1965).
- [27] L. K. Grover, *Phys. Rev. Lett.* **79**, 325 (1997).
- [28] M. Boyer, G. Brassard, P. Hoyer, and A. Tapp, in *Proceedings of the Workshop on Physics and Computation (PhysComp96)*, edited by T. Toffoli *et al.* (New England Complex Systems Institute, Cambridge, MA, 1996), pp. 36–43.
- [29] G. Brassard, P. Hoyer, and A. Tapp, in *Proceedings of 25th International Colloquium on Automata, Languages, and Programming (ICALP98)*, edited by K. Larsen (Springer, Berlin, 1998), pp. 820–831.
- [30] S. M. Maurer, T. Hogg, and B. A. Huberman, *Phys. Rev. Lett.* **87**, 257901 (2001).