Ising formulations of many NP problems

Andrew Lucas *

Lyman Laboratory of Physics, Department of Physics, Harvard University, Cambridge, MA, USA

*Correspondence:
Andrew Lucas, Lyman Laboratory of Physics, Department of Physics, Harvard University, 17 Oxford St., Cambridge, MA 02138, USA
e-mail: lucas@fas.harvard.edu

We provide Ising formulations for many NP-complete and NP-hard problems, including all of Karp's 21 NP-complete problems. This collects and extends mappings to the Ising model from partitioning, covering, and satisfiability. In each case, the required number of spins is at most cubic in the size of the problem. This work may be useful in designing adiabatic quantum optimization algorithms.

Keywords: spin glasses, complexity theory, adiabatic quantum computation, NP, algorithms

1. INTRODUCTION

1.1. QUANTUM ADIABATIC OPTIMIZATION

Recently, there has been much interest in the possibility of using adiabatic quantum optimization (AQO) to solve NP-complete and NP-hard problems [1, 2] [1]. This is due to the following trick: suppose we have a quantum Hamiltonian H_P whose ground state encodes the solution to a problem of interest, and another Hamiltonian H_0, whose ground state is "easy" (both to find and to prepare in an experimental setup). Then, if we prepare a quantum system to be in the ground state of H_0, and then adiabatically change the Hamiltonian for a time T according to

$$H(t) = \left(1 - \frac{t}{T}\right) H_0 + \frac{t}{T} H_P, \qquad (1)$$

then if T is large enough, and H_0 and H_P do not commute, the quantum system will remain in the ground state for all times, by the adiabatic theorem of quantum mechanics. At time T, measuring the quantum state will return a solution of our problem.

There has been debate about whether or not these algorithms would actually be useful: i.e., whether an adiabatic quantum optimizer would run any faster than classical algorithms [3–9], due to the fact that if the problem has size N, one typically finds

$$T = O\left[\exp\left(\alpha N^\beta\right)\right], \qquad (2)$$

in order for the system to remain in the ground state, for positive coefficients α and β, as $N \to \infty$. This is a consequence of the requirement that exponentially small energy gaps between the ground state of $H(t)$ and the first excited state, at some intermediate time, not lead to Landau–Zener transitions into

excited states [5] [2]. While it is unlikely that NP-complete problems can be solved in polynomial time by AQO, the coefficients α, β may be smaller than known classical algorithms, so there is still a possibility that an AQO algorithm may be more efficient than classical algorithms, on some classes of problems.

There has been substantial experimental progress toward building a device capable of running such algorithms [11–13], when the Hamiltonian H_P may be written as the quantum version of an Ising spin glass. A classical Ising model can be written as a quadratic function of a set of N spins $s_i = \pm 1$:

$$H(s_1, \ldots, s_N) = -\sum_{i<j} J_{ij} s_i s_j - \sum_{i=1}^{N} h_i s_i. \qquad (3)$$

The quantum version of this Hamiltonian is simply

$$H_P = H\left(\sigma_1^z, \ldots, \sigma_N^z\right) \qquad (4)$$

where σ_i^z is a Pauli matrix (a 2×2 matrix, whose cousin $(1 + \sigma_i^z)/2$ has eigenvectors $|0, 1\rangle$ with eigenvalues 0, 1) acting on the ith qubit in a Hilbert space of N qubits $\{|+\rangle, |-\rangle\}^{\otimes N}$, and J_{ij} and h_i are real numbers. We then choose H_0 to consist of transverse magnetic fields [11]:

$$H_0 = -h_0 \sum_{i=1}^{N} \sigma_i^x, \qquad (5)$$

so that the ground state of H_0 is an equal superposition of all possible states in the eigenbasis of H_P [equivalent to the eigenbasis of the set of operators σ_i^z $(i = 1, \ldots, N)$]. This means that one

---

[1] In this paper, when a generic statement is true for both NP-complete and NP-hard problems, we will refer to these problems as NP problems. Formally this can be misleading as P is contained in NP, but for ease of notation we will simply write NP.

[2] If one is only interested in approximate solutions (for example, finding a state whose energy per site is optimal, in the thermodynamic ($N \to \infty$) limit, as opposed to finding the exact ground state), one expects $T = O(N^\gamma)$ [5, 10].

does not expect any level crossings [3]. For more work discussing the choice of H, see Whitfield et al. [14]. Also, note that this class of Hamiltonians is not believed to be sufficient to build a universal adiabatic quantum computer [15]—at all times, $H(t)$ belongs to a special class of Hamiltonians called *stoquastic* Hamiltonians [16].

1.2. ISING SPIN GLASSES

Ising spin glasses[4] are known to be NP-hard problems for classical computers [17], so it is natural to suspect intimate connections with all other NP problems. For the purposes of this paper, an NP-complete problem is always a decision problem with a yes or no answer (does the ground state of H have energy ≤ 0?), whereas an NP-hard problem is an optimization problem (what is the ground state energy of H?). The class of NP-complete problems includes a variety of notoriously hard problems, and has thus attracted much interest over the last 40 years [18, 19]. Mathematically, because the decision form of the Ising model is NP-complete, there exists a polynomial time mapping to any other NP-complete problem.

Analogies between the statistical physics of Ising spin glasses and NP problems have been frequently studied in the past [20–22], and have been used to construct simulated annealing algorithms [23] which have been quite fruitful in approximate algorithms for problems on classical computers. These connections have suggested a *physical* understanding of the emergence of hardness in these problems via a complex energy landscape with many local minima [24]. Conversely, computational hardness of solving glassy problems has implications for the difficulty of the solutions to important scientific problems ranging from polymer folding [25, 26] to memory [27] to collective decision making in economics and social sciences [28, 29]. Problems of practical scientific interest have already been encoded and solved (in simple instances) on experimental devices using Ising Hamiltonians [30–35].

Finally, we note that Ising glasses often go by the name QUBO (quadratic unconstrained binary optimization), in the more mathematical literature [36, 37]. Useful tricks have been developed to fix the values of some spins immediately [38] and to decompose large QUBO problems into smaller ones [39].

1.3. THE GOAL OF THIS PAPER

Mathematically, the fact that a problem is NP-complete means we can find a mapping to the decision form of the Ising model with a polynomial number of steps. This mapping can be re-interpreted as a pseudo-Boolean optimization problem [37]. As the constructions of these pseudo-Boolean optimization problems (or "*p*-spin glasses") often lead to three-body or higher interactions in H (e.g.,

terms of the form $s_1 s_2 s_3$), we then conclude by using "gadgets" to reduce the problem to an Ising spin glass, by introducing a polynomial number of auxiliary spins which help to enforce the three-body interaction by multiple two-body interactions ($s_1 s_2$) [40, 41]. As such, we can get from any NP-complete problem to the Hamiltonian of an Ising spin glass, whose decision problem (does the ground state have energy ≤ 0?) solves the NP-complete problem of interest. Classical gadgets are useful for many problems in physics as the physical energy (Hamiltonian) contains three-body interactions, but they are also useful for writing down many algorithms in other fields (e.g., integer factorization [42]).

However, for generic problems, this is a very inefficient procedure, as the power of the polynomial can grow quite rapidly. As such, the typical NP-complete problem (of size N) studied in the context of Ising glasses is very straightforward to write as a glass with N spins (such as number partitioning, or satisfiability). The primary purpose of this paper is to present constructions of Ising Hamiltonians for problems where finding a choice of Hamiltonian is a bit subtle; for pedagogical purposes, we will also provide a review of some of the simple maps from partitioning and satisfiability to an Ising spin glass. In particular, we will describe how "all of the famous NP problems"[5] Karp [18], Garey and Johnson [19] can be written down as Ising models with a polynomial number of spins which scales no faster than N^3. For most of this paper, we will find it no more difficult to solve the NP-hard optimization problem vs. the NP-complete decision problem, and as such we will usually focus on the optimization problems. The techniques employed in this paper, which are rare elsewhere in the quantum computation literature, are primarily of a few flavors, which roughly correspond to the tackling the following issues: minimax optimization problems, problems with inequalities as constraints (for example, $n \geq 1$, as opposed to $n = 1$), and problems which ask global questions about graphs. The methods we use to phrase these problems as Ising glasses generalize very naturally.

1.4. WHAT PROBLEMS ARE EASY (TO EMBED) ON EXPERIMENTAL AQO DEVICES?

We hope that the reader may be inspired, after reading this paper, to think about solving some of these classical computing problems, or others like them, on experimental devices implementing AQO. Toward this end, the reader should look for three things in the implementations in this paper. The first is the number of spins required to encode the problem. In some instances, the "logical spins/bits" (the spins which are required to encode a solution of the problem) are the only spins required; but in general, we may require auxiliary "ancilla spins/bits", which are required to enforce constraints in the problem. Sometimes, the number of ancilla bits required can be quite large, and can be the dominant fraction of the spins in the Hamiltonian. Another thing to watch out for is the possibility that large separations of energy scales are required: e.g., the ratio of couplings J_{12}/J_{23} in some Ising glass is proportional to N, the size of the problem being studied. A final thing to note is whether or not the graph must be highly connected: does the typical degree of vertices on the *Ising embedding graph*

---

[3]This is due to the fact that the eigenbases of H_0 and H_P are very different, and one has to tune (in our case) two parameters of a 2×2 Hermitian matrix to find a matrix with degenerate eigenvalues (the identity matrix). We only have one, t, and thus do not expect any degeneracies [5].

[4]In this paper, we will casually refer to the Ising models we are constructing as "glasses", as they can be on general graphs and have both positive and negative couplings J_{ij}. There are various mathematical definitions for a spin glass, none of which seem to capture properly the physical essence of a glass on all problems. We will be liberal with our use of the word glass, and refer to any NP problem, formulated as an Ising model, as a glass.

[5]No offense to anyone whose problems have been left out.

(not the graph associated with the NP problem) scale linearly with N?

It is probably evident why we do not want too many ancilla bits—this simply means we can only encode smaller problems on the same size piece of hardware. It is a bit more subtle to understand why complete graphs, or separations of energy scales, are problematic. It is probable that the successful experimental implementations of AQO with the most qubits are on devices generated by DWave Systems [11–13] [6]. As such, we now discuss the ease with which these Hamiltonians can be encoded onto such a device. These devices may only encode problems via a "chimera" graph. The primary problem with Hamiltonians on a complete graph is that it is inefficient [43, 44] to embed complete graphs onto the chimera graph. A primary difficulty is demonstrated by the following simple case: a node v in the complete graph must be mapped two a pair of nodes u and w on the chimera graph, with the coupling J_{uw} large compared to other scales in the problem, to ensure that $s_u = s_w$ (so these nodes effectively act as one spin). A second problem is that some of the Hamiltonians require separations of energy scales. However, in practice, these devices may only encode couplings constants of $1, \ldots, 16$, due to experimental uncertainties [11–13]. This means that it is unlikely that, for very connected graphs, one may successfully encode any H with a separation of energy scales. A final challenge is that sometimes couplings or qubits are broken—at this early stage in the hardware development, optimal algorithms have embeddings which are insensitive to this possibility [45].

2. PARTITIONING PROBLEMS

The first class of problems we will study are partitioning problems, which (as the name suggests) are problems about dividing a set into two subsets. These maps are celebrated in the spin glass community [24], as they helped physicists realize the possibility of using spin glass technology to understand computational hardness in random ensembles of computing problems. For completeness, we review these mappings here, and present a new one based on similar ideas (the clique problem).

2.1. NUMBER PARTITIONING

Number partitioning asks the following: given a set of N positive numbers $S = \{n_1, \ldots, n_N\}$, is there a partition of this set of numbers into two disjoint subsets R and $S - R$, such that the sum of the elements in both sets is the same? For example, can one divide a set of assets with values $n_1, \ldots, n_N$, fairly between two people? This problem is known to be NP-complete [18]. This can be phrased trivially as an Ising model as follows. Let n_i ($i = 1, \ldots, N = |S|$) describe the numbers in set S, and let

$$H = A \left(\sum_{i=1}^{N} n_i s_i \right)^2 \tag{6}$$

be an energy function, where $s_i = \pm 1$ is an Ising spin variable. Here $A > 0$ is some positive constant. Typically, such constants are scaled to 1 in the literature, but for simplicity we will retain them, since in many formulations a separation of energy scales will prove useful, and retaining each scale can make it easier to follow conceptually. Classical studies of this problem are slightly easier if the square above is replaced with absolute value [24].

It is clear that if there is a solution to the Ising model with $H = 0$, then there is a configuration of spins where the sum of the n_i for the $+1$ spins is the same for the sum of the n_i for the -1 spins. Thus, if the ground state energy is $H = 0$, there is a solution to the number partitioning problem.

This Ising glass has *degeneracies*—i.e., there are always at least two different solutions to the problem. This can be seen by noting that if s_i^* denotes a solution to the problem, then $-s_i^*$ is also a solution. Physically, this corresponds to the fact that we do not care which set is labeled as $\pm$. In the spin glass literature, the change $s_i \to -s_i$, which does not change the form of H, is often (rather loosely) called a *gauge transformation*. The existence of a gauge transformation which leaves the couplings unchanged (as there are no linear terms) implies that all energy levels of H are degenerate. It is possible that there are $2m$ ground states (with $m > 1$). This means that there are m physically distinct solutions to the computational problem. We only need to find one of them to be happy with our adiabatic quantum algorithm. We can remove this double degeneracy by fixing $s_1 = 1$. This also allows us to remove one spin: now only $s_2, \ldots, s_N$ are included on the graph, and s_1 serves as an effective magnetic field. So in general, we require $N - 1$ spins, which live on a complete graph, to encode this problem.

If the ground state has $H > 0$, we know that there are no solutions to the partitioning problem, but the ground state we do find is (one of) the best possible solutions, in the sense that it minimizes the mismatch. Minimizing this mismatch is an NP-hard problem, and we see that we do not require any more fancy footwork to solve the optimization problem—the same Hamiltonian does the trick.

2.2. GRAPH PARTITIONING

Graph partitioning is the original [20] example of a map between the physics of Ising spin glasses and NP-complete problems. Let us consider an undirected graph $G = (V, E)$. with an even number $N = |V|$ of vertices. We ask: what is a partition of the set V into two subsets of equal size $N/2$ such that the number of edges connecting the two subsets is minimized? This problem has many applications: finding these partitions can allow us to run some graph algorithms in parallel on the two partitions, and then make some modifications due to the few connecting edges at the end [39]. Graph partitioning is known to be an NP-hard problem; the corresponding decision problem (are there less than k edges connecting the two sets?) is NP-complete [18]. We will place an Ising spin $s_v = \pm 1$ on each vertex $v \in V$ on the graph, and we will let $+1$ and -1 denote the vertex being in either the $+$ set or the $-$ set. We solve this with an energy functional consisting of two components:

$$H = H_A + H_B \tag{7}$$

where

$$H_A = A \left(\sum_{n=1}^{N} s_i \right)^2 \qquad (8)$$

is an energy which provides a penalty if the number of elements in the + set is not equal to the number in the − set, and

$$H_B = B \sum_{(uv) \in E} \frac{1 - s_u s_v}{2} \qquad (9)$$

is a term which provides an energy penalty B for each time that an edge connects vertices from different subsets. If $B > 0$, then we wish to minimize the number of edges between the two subsets; if $B < 0$, we will choose to maximize this number. Should we choose $B < 0$, we must ensure that B is small enough so that it is never favorable to violate the constraint of H_A in order to minimize energy. To determine a rather simple lower bound on A, we ask the question: what is the minimum value of ΔH_B—the change in the energy contributed by the B-term—if we violate the A constraint once. It is easy to see that the penalty for violating the A-constraint is $\Delta H_A \geq 4A$. The best gain we can get by flipping a spin is to gain an energy of $B \min(\Delta, N/2)$, where Δ is the maximal degree of G[7]. We conclude

$$\frac{A}{B} \geq \frac{\min(2\Delta, N)}{8}. \qquad (10)$$

N spins on a complete graph are required to encode this problem.

This Hamiltonian is invariant under the same gauge transformation $s_i \rightarrow -s_i$. We conclude that we can always remove one spin by fixing a single vertex to be in the + set.

We have written H in a slightly different form than the original [20], which employed a constraint on the space of solutions to the problem, that

$$\sum_{i=1}^{N} s_i = 0. \qquad (11)$$

We will want none of our formulations to do this (i.e., we wish to solve the unconstrained optimization problem), as the experimental hardware that is being built for quantum optimization can only solve unconstrained problems. Instead, we encode constraint equations by making penalty Hamiltonians which raise the energy of a state which violates them.

2.3. CLIQUES

A clique of size K in an undirected graph $G = (V, E)$ is a subset $W \subseteq V$ of the vertices, of size $|W| = K$, such that the subgraph (W, E_W) (where E_W is the edge set E restricted to edges between nodes in W) is a complete graph—i.e., all possible $K(K-1)/2$

---

[7]The reason we can use $N/2$ in this formula instead of N has to do with the fact that we are "perturbing" a solution where $H_A = 0$. Due to the fact that the H_A constraint is very penalizing if it is violated by having many spins in the same partition, it is easy to see that cases where an energy gain of $(N-1)B$ can be obtained by flipping a spin are very energetically penalized, and not relevant to the discussion.

edges in the graph are present, because every vertex in the clique has an edge to every other vertex in the clique. Cliques in social networks can be useful as they are "communities of friends"; finding anomalously large cliques is also a key sign that there is structure in a graph which may appear to otherwise be random [46]. The NP-complete decision problem of whether or not a clique of size K exists [18] can be written as an Ising-like model, as follows. We place a spin variable $s_v = \pm 1$ on each vertex $v \in V$ of the graph. In general, in this paper, for a spin variable s_α, we will define the binary bit variable

$$x_\alpha \equiv \frac{s_\alpha + 1}{2}. \qquad (12)$$

It will typically be more convenient to phrase the energies in terms of this variable x_α, as it will be for this problem. Note that any energy functional which was quadratic in s_v will remain quadratic in x_v, and vice versa, so we are free to use either variable. We then choose

$$H = A \left(K - \sum_v x_v \right)^2 + B \left[\frac{K(K-1)}{2} - \sum_{(uv) \in E} x_u x_v \right] \qquad (13)$$

where $A, B > 0$ are positive constants. We want the ground state of this Hamiltonian is $H = 0$ if and only if a clique of size K exists. It is easy to see that $H = 0$ if there is a clique of size K. However, we wish to now show that $H \neq 0$ for any other solution. It is easy to see that if there are n x_vs which are 1, that the minimum possible value of H is

$$H_{\min}(n) = A(n-K)^2 + B \frac{K(K-1) - n(n-1)}{2}$$

$$= (n-K) \left[A(n-K) - B \frac{n+K-1}{2} \right]. \qquad (14)$$

The most "dangerous" possible value of $n = 1 + K$. We can easily see that so long as $A > KB$, $H_{\min}(K+1) > 0$. We finally note that, given a ground state solution, it is of course easy to read off from the x_v which K nodes form a clique. N spins on a complete graph are required to solve this problem.

A quantum algorithm for this NP-complete problem can be made slightly more efficient so long as the initial state can be carefully prepared [47].

The NP-hard version of the clique problem asks us to find (one of) the *largest* cliques in a graph. We can modify the above Hamiltonian to account for this, by adding an extra variable y_i ($i = 2, \ldots, \Delta$), which is 1 if the largest clique has size i, and 0 otherwise. Let $H = H_A + H_B + H_C$ where

$$H_A = A \left(1 - \sum_{i=2}^{N} y_i \right)^2 + A \left(\sum_{i=2}^{n} n y_n - \sum_v x_v \right)^2 \qquad (15)$$

and

$$H_B = B \left[\frac{1}{2} \left(\sum_{i=2}^{N} n y_n \right) \left(-1 + \sum_{i=2}^{N} n y_n \right) - \sum_{(uv) \in E} x_u x_v \right]. \qquad (16)$$

We want cliques to satisfy $H_A = H_B = 0$, and to be the only ground states. The Hamiltonian above satisfies this if A/B is large enough so the constraints of $H_A = 0$ are always satisfied—we can see this by noting that the first term of H_A forces us to pick only one value of $y_i = n$, and the second term fixes us to choose n vertices. Then $H_B = 0$ ensures that we have a clique. Similarly to the discussion above, we see that the absence of negative energy states requires $A > NB$. If the maximal degree of the graph is Δ, this can be simplified to $A > \Delta B$. Now that we know that all ground states are cliques[8], we have to find the state with the smallest value of y_n. This can be obtained by choosing

$$H = -C \sum_v x_v, \tag{17}$$

where $C > 0$ is some constant. If C is small enough, then the ground state energy is $H = -CK$, where K is the size of the largest clique in the graph. To determine an upper bound on C, so that we solve the cliques problem (as opposed to some other problem), we need to make sure that it is never favorable to color an extra vertex, at the expense of mildly violating the H_A constraint. The penalty for coloring one extra vertex, given $y_i = n$, is at minimum $A - nB - C$. We conclude that we must choose

$$C < A - nB. \tag{18}$$

So, for example, we could take $A = (\Delta + 2)B$, and $B = C$.

2.4. REDUCING *N* TO LOG*N* SPINS IN SOME CONSTRAINTS
There is a trick which can be used to dramatically reduce the number of extra y_i spins which must be added, in the NP-hard version of the clique problem above [48]. In general, this trick is usable throughout this paper, as we will see similar constructions of auxiliary y bits appearing repeatedly.

We know that we want to encode a variable which can take the values $2, \ldots, N$ (or Δ, if we know the maximal degree of the graph—the argument is identical either way). For simplicity, suppose we wish to encode the values $1, \ldots, N$ (this is a negligible difference, in the large N limit). Define an integer M so that

$$2^M \leq N < 2^{M+1}. \tag{19}$$

Alternatively, $M = \lfloor \log N \rfloor$—in this paper, the base 2 is implied in the logarithm. In this case, we only need $M + 1$ binary variables: $y_0, \ldots, y_M$, instead of N binary variables, $y_1, \ldots, y_N$, to encode a variable which can take N values. It is easy to check that replacing

$$\sum_{n=1}^{N} n y_n \rightarrow \sum_{n=0}^{M-1} 2^n y_n + \left(N + 1 - 2^M\right) y_M \tag{20}$$

solves the same clique problem, without loss of generality. (This is true in general for all of our NP problems.) If $N \neq 2^{M+1} - 1$, the ground state may be degenerate, as the summation of ys to a given integer is not always unique. When actually encoding these

---

[8] The ground state has $H = 0$ so long as the edge set is non-empty: any connected pair of edges is a clique of size 2.

---

problems for computational purposes, of course, this trick should be used, but for pedagogy and simplicity we will not write it out explicitly for the remainder of the paper.

Using this trick, we note that solving the NP-hard version of the cliques problem requires $N + 1 + \lfloor \log \Delta \rfloor$ spins.

3. BINARY INTEGER LINEAR PROGRAMMING
Let $x_1, \ldots, x_N$ be N binary variables, which we arrange into a vector $\mathbf{x}$. The binary integer linear programming (ILP) problem asks: what is the largest value of $\mathbf{c} \cdot \mathbf{x}$, for some vector $\mathbf{c}$, given a constraint

$$\mathsf{S}\mathbf{x} = \mathbf{b} \tag{21}$$

with S an $m \times N$ matrix and $\mathbf{b}$ a vector with m components. This is NP-hard [18], with a corresponding NP-complete decision problem. Many problems can be posed as ILP: e.g., a supplier who wants to maximize profit, given regulatory constraints [48].

The Ising Hamiltonian corresponding to this problem can be constructed as follows. Let $H = H_A + H_B$ where

$$H_A = A \sum_{j=1}^{m} \left[b_j - \sum_{i=1}^{N} S_{ji} x_i \right]^2 \tag{22}$$

and $A > 0$ is a constant. The ground states of $H_A = 0$ enforce (if such a ground state exists, of course!) the constraint that $\mathsf{S}\mathbf{x} = \mathbf{b}$. Then we set

$$H_B = -B \sum_{i=1}^{N} c_i x_i. \tag{23}$$

with $B \ll A$ another positive constant.

To find constraints on the required ratio A/B, we proceed similarly to before. For simplicity, let us assume that the constraint Equation (21) can be satisfied for some choice of $\mathbf{x}$. For such a choice, the largest possible value of $-\Delta H_B$ is, in principle, $B\mathcal{C}$, where

$$\mathcal{C} = \sum_{i=1}^{N} \max(c_i, 0). \tag{24}$$

The smallest possible value of ΔH_A is related to the properties of the matrix S, and would occur if we only violate a single constraint, and violate that constraint by the smallest possible amount, given by

$$\mathcal{S} \equiv \min_{\sigma_i \in \{0,1\}, \, j} \left(\max \left[1, \frac{1}{2} \sum_i (-1)^{\sigma_i} S_{ji} \right] \right). \tag{25}$$

This bound could be made better if we know more specific properties of S and/or $\mathbf{b}$. We conclude

$$\frac{A}{B} \geq \frac{\mathcal{C}}{\mathcal{S}}. \tag{26}$$

If the coefficients c_i and S_{ij} are O(1) integers, we have $\mathcal{C} \leq N \max(c_i)$, and $\mathcal{S} \geq 1$, so we conclude $A/B \gtrsim N$.

4. COVERING AND PACKING PROBLEMS

In this section, we discuss another simple class of mappings from NP problems to Ising models: "covering" and "packing" problems. These problems can often be thought of as asking: how can I pick elements out of a set (such as vertices out of a graph's vertex set) so that they "cover" the graph in some simple way (e.g., removing them makes the edge set empty). In this class of problems, there are constraints which must be exactly satisfied. Many of the problems described below are often discussed in the literature, but again we review them here for completeness. We conclude the section with the minimal maximal matching problem, which is a slightly more involved problem that has not been discussed in the AQO literature before.

These are, by far, the most popular class of problems discussed in the AQO literature. As we mentioned in the introduction, this is because this is the *only* class of NP problems (discussed in this paper) for which it is easy to embed the problem via a graph which is not complete (or close to complete).

4.1. EXACT COVER

The exact cover problem goes as follows: consider a set $U = \{1, \ldots, n\}$, and subsets $V_i \subseteq U$ ($i = 1, \ldots, N$) such that

$$U = \bigcup_i V_i. \tag{27}$$

The question is: is there a subset of the set of sets $\{V_i\}$, called R, such that the elements of R are disjoint sets, and the union of the elements of R is U? This problem was described in Choi [49] but for simplicity, we repeat it here. This decision problem is NP-complete [18]. The Hamiltonian we use is

$$H_A = A \sum_{\alpha=1}^{n} \left(1 - \sum_{i:\alpha \in V_i} x_i\right)^2. \tag{28}$$

In the above Hamiltonian α denotes the elements of U, while i denotes the subsets V_i. $H_A = 0$ precisely when every element is included exactly one time, which implies that the unions are disjoint. The existence of a ground state of energy $H = 0$ corresponds to the existence of a solution to the exact cover problem. If this ground state is degenerate, there are multiple solutions. N spins are required.

It is also straightforward to extend this, and find the *smallest* exact cover (this makes the problem NP-hard). This is done by simply adding a second energy scale: $H = H_A + H_B$, with H_A given above, and

$$H_B = B \sum_i x_i. \tag{29}$$

The ground state of this model will be mB, where m is the smallest number of subsets required. To find the ratio A/B required to encode the correct problem, we note that the worst case scenario is that there are a very small number of subsets with a single common element, whose union is U. To ensure this does not happen, one can set $A > nB$[9].

4.2. SET PACKING

Let us consider the same setup as the previous problem, but now ask a different question: what is the largest number of subsets V_i which are all disjoint? This is called the set packing problem; this optimization problem is NP-hard [18]. To do this, we use $H = H_A + H_B$:

$$H_A = A \sum_{i,j: V_i \cap V_j \neq \emptyset} x_i x_j, \tag{30}$$

which is minimized only when all subsets are disjoint. Then, we use

$$H_B = -B \sum_i x_i \tag{31}$$

which simply counts the number of sets we included. Choosing $B < A$ ensures that it is never favorable to violate the constraint H_A (since there will always be a penalty of at least A per extra set included) [4].

Note that an isomorphic formulation of this problem, in the context of graph theory is as follows: let us consider the sets to be encoded in an undirected graph $G = (V, E)$, where each set V_i maps to a vertex $i \in V$. An edge $ij \in E$ exists when $V_i \cap V_j$ is non-empty. It is straightforward to see that if we replace

$$H_A = A \sum_{ij \in E} x_i x_j \tag{32}$$

that the question of what is the maximal number of vertices which may be "colored" ($x_i = 1$) such that no two colored vertices are connected by an edge, is exactly equivalent to the set packing problem described above. This version is called the maximal independent set (MIS) problem.

4.3. VERTEX COVER

Given an undirected graph $G = (V, E)$, what is the smallest number of vertices that can be "colored" such that every edge is incident to a colored vertex? This is NP-hard; the decision form is NP-complete [18]. Let x_v be a binary variable on each vertex, which is 1 if it is colored, and 0 if it is not colored. Our Hamiltonian will be $H = H_A + H_B$. The constraint that every edge has at least colored vertex is encoded in H_A:

$$H_A = A \sum_{uv \in E} (1 - x_u)(1 - x_v). \tag{33}$$

Then, we want to minimize the number of colored vertices with H_B:

$$H_B = B \sum_v x_v \tag{34}$$

---

[9]The example where $V = \{\{1, 2\}, \{3\}, \ldots, \{n\}, \{2, \ldots, n\}\}$ shows that to leading order in n, this bound is optimal.

Choose $B < A$, as if we uncolor any vertex that ruins the solution, at least one edge will no longer connect to a colored vertex. The number of spins required is $|V|$, the size of the vertex set.

4.4. SATISFIABILITY

Satisfiability is one of the most famous NP-complete problems [18]. Every satisfiability problem can be written as a so-called 3SAT problem in conjunctive normal form (and this algorithm takes only polynomial steps/time) and so we will focus for simplicity on this case. In this case, we ask whether

$$\Psi = C_1 \wedge C_2 \cdots \wedge C_m \qquad (35)$$

can take on the value of true—i.e., every C_i for $1 \le i \le m$ is true, where the form of each C_i is:

$$C_i = y_{i_1} \vee y_{i_2} \vee y_{i_3} \qquad (36)$$

Here y_{i_1}, y_{i_2}, and y_{i_3} are selected from another set of Boolean variables: $x_1, \ldots, x_N, \bar{x}_1, \ldots, \bar{x}_N$. This is a very brief description of satisfiability; physicists who are unfamiliar with this problem should read appropriate chapters of Mézard and Montanari [24].

There is a well-known reduction of 3SAT to MIS [49] which we reproduce here, for completeness. Consider solving the set packing problem on a graph G with $3m$ nodes, which we construct as follows. For each clause C_i, we add 3 nodes to the graph, and connect each node to the other 3. After this step, if there is a y_1 and y_2 such that $y_1 = \bar{y}_2$, then we also add an edge between these two nodes. Solving MIS on this graph, and asking whether the solution has exactly m nodes, is equivalent to solving the 3SAT problem. This can be seen as follows: if a solution to the 3SAT problem exists, only one element of each clause needs to be true—if more are true, that is also acceptable, but we must have that one is true, so let us choose to color the vertex corresponding to the variable which is true. However, we may also not choose to have both x_1 be true and $\bar{x}_1$ be true, so we are required to connect all such points with an edge. Since the graph is made up of m connected triangles, the only way to color m vertices if each vertex is in a distinct triangle, so there must be an element of each clause C_i which is true.

Note that we can solve an NP-hard version of this problem (if we have to violate some clauses, what is the fewest number?), by solving the optimization version of the MIS problem.

4.5. MINIMAL MAXIMAL MATCHING

The minimal maximal (minimax) matching problem on a graph is defined as follows: let $G = (V, E)$ denote an undirected graph, and let $C \subseteq E$ be a proposed "coloring". The constraints on C are as follows: for each edge in C, let us color the two vertices it connects: i.e., let $D = \bigcup_{e \in C} \partial e$. We will then demand that: no two edges in C share a vertex (if $e_1, e_2 \in C$, $\partial e_1 \cap \partial e_2 = \emptyset$) and that if $u, v \in D$, that $(uv) \notin E$. This is NP-hard; the decision problem is NP-complete [19]. This is minimal in that we cannot add any more edges to C (coloring any appropriate vertices) without violating the first constraint, and maximal in the sense that the trivial empty set solution is not allowed—we must include all edges between uncolored vertices.

Note that, from this point on in this paper, we have not found any of the Ising formulations of this paper in the literature.

We will use the spins on the graph to model whether or not an edge is colored. Let us use the binary variable x_e to denote whether or not an edge is colored; thus, the number of spins is $|E| = \mathrm{O}(\Delta N)$, the size of the edge set; as before, Δ represents the maximal degree. To encode this problem, we use a series of three Hamiltonians:

$$H = H_A + H_B + H_C. \qquad (37)$$

The first and largest term, H_A, will impose the constraint that no vertex has two colored edges. This can be done by setting

$$H_A = A \sum_v \sum_{\{e_1, e_2\} \subset \partial v} x_{e_1} x_{e_2}. \qquad (38)$$

Here $A > 0$ is a positive energy, and ∂v corresponds to the subset of E of edges which connect to v. Thus the ground states consist of $H_A = 0$; if $H_A > 0$, it is because there is a vertex where two of its edges are colored.

We also can define, for states with $H_A = 0$, the variable

$$y_v \equiv \begin{cases} 1 & v \text{ has a colored edge} \\ 0 & v \text{ has no colored edges} \end{cases} = \sum_{e \in \partial v} x_e. \qquad (39)$$

We stress that this definition is only valid for states with $H_A = 0$, since in these states each vertex has either 0 or 1 colored edges. We then define the energy H_B, such that solutions to the minimax coloring problem also have $H_B = 0$. Since we have already constrained the number of colored edges per vertex, we choose H_B to raise the energy of all solutions where there exists a possible edge which can be colored, yet still not violate the coloring condition, out of the ground state. To do this, we can sum over all edges in the graph, and check whether or not the edge connects two vertices, neither of which are colored:

$$H_B = B \sum_{e = (uv)} (1 - y_u)(1 - y_v). \qquad (40)$$

Note that since, $1 - y_v$ can be negative, we must choose $B > 0$ to be small enough. To bound B, we note that the only problem (a negative term in H_B) comes when $y_u = 0$, $y_v > 1$, and $(uv) \in E$. Suppose that m of v's neighbors have $y_u = 0$. Then, the contributions to H_A and H_B associated to node v are given by

$$H_v = A \frac{y_v(y_v - 1)}{2} - B(y_u - 1)m. \qquad (41)$$

Note that $m + y_u \le k$, if k is the degree of node v. Putting all of this together, we conclude that if Δ is the maximal degree in the graph, because the worst case scenario is $y_v = 2$, $m = \Delta - 2$, if we pick

$$A > (\Delta - 2)B, \qquad (42)$$

then it is never favorable to have any $y_v > 1$. This will ensure that a ground state of $H_A + H_B$ will have $H_A = H_B = 0$: i.e., states which do not violate the minimax constraints.

Now, given the states where $H_A = H_B = 0$, we now want the ground state of $H_A + H_B + H_C$ to be the state where the fewest number of edges are colored. To do this, we simply let

$$H_C = C \sum_e x_e \tag{43}$$

count the number of colored edges. Here C is an energy scale chosen to be small enough such that it is never energetically favorable to violate the constraints imposed by either the H_A or H_B terms: one requires $C < B$, since there is an energy penalty of B associated to each edge which could be colored, but isn't. The term with the smallest H_C has the smallest number of edges, and is clearly the solution to the minimax problem. Each ground state of this spin model is equivalent to a solution of the minimax problem.

5. PROBLEMS WITH INEQUALITIES

We now turn to NP problems whose formulations as Ising models are more subtle, due to the fact that constraints involve inequalities as opposed to equalities. These constraints can be re-written as constraints only involving equalities by an expansion of the number of spins.

As with partitioning problems, we will find that these Hamiltonians require embedding highly connected graphs onto a quantum device. This may limit their usability on current hardware.

5.1. SET COVER
Consider a set $U = \{1, \ldots, n\}$, with sets $V_i \subseteq U$ $(i = 1, \ldots, N)$ such that

$$U = \bigcup_{i=1}^{N} V_\alpha. \tag{44}$$

The set covering problem is to find the smallest possible number of V_is, such that the union of them is equal to U. This is a generalization of the exact covering problem, where we do not care if some $\alpha \in U$ shows up in multiple sets V_i; finding the smallest number of sets which "cover" U is NP-hard [18].

Let us denote x_i to be a binary variable which is 1 if set i is included, and 0 if set i is not included. Let us then denote $x_{\alpha,m}$ to be a binary variable which is 1 if the number of V_is which include element α is $m \geq 1$, and 0 otherwise. Set $H = H_A + H_B$. Our first energy imposes the constraints that exactly one $x_{\alpha,m}$ must be 1, since each element of U must be included a fixed number of times, and that the number of times that we claimed α was included is in fact equal to the number of V_i we have included, with α as an element:

$$H_A = A \sum_{\alpha=1}^{n} \left(1 - \sum_{m=1}^{N} x_{\alpha,m} \right)^2$$

$$+ A \sum_{\alpha=1}^{n} \left(\sum_{m=1}^{N} m x_{\alpha,m} - \sum_{i:\alpha \in V_i} x_i \right)^2. \tag{45}$$

Finally, we minimize over the number of V_αs included:

$$H_B = B \sum_{i=1}^{N} x_i, \tag{46}$$

with $0 < B < A$ required to never violate the constraints of H_A (the worst case is that one set must be included to obtain one element of U; the change in H if we include this last element is $B - A$, which must be negative).

Let $M \leq N$ be the maximal number of sets which contain any given element of U; then N x_is are required, and $n\lfloor 1 + \log M \rfloor$ spins are required (using the trick described earlier) for the $x_{\alpha,m}$ spins; the total number is therefore $N + n\lfloor 1 + \log M \rfloor$ spins.

5.2. KNAPSACK WITH INTEGER WEIGHTS
The knapsack problem is the following problem: we have a list of N objects, labeled by indices α, with the weight of each object given by w_α, and its value given by c_α, and we have a knapsack which can only carry weight W. If x_α is a binary variable denoting whether (1) or not (0) object α is contained in the knapsack, the total weight in the knapsack is

$$\mathcal{W} = \sum_{\alpha=1}^{N} w_\alpha x_\alpha \tag{47}$$

and the total cost is

$$\mathcal{C} = \sum_{\alpha=1}^{N} c_\alpha x_\alpha. \tag{48}$$

The NP-hard [18] knapsack problem asks us to maximize $\mathcal{C}$ subject to the constraint that $\mathcal{W} \leq W$. It has a huge variety of applications, particularly in economics and finance [50].

Let y_n for $1 \leq n \leq W$ denote a binary variable which is 1 if the final weight of the knapsack is n, and 0 otherwise. Our solution consists of letting $H = H_A + H_B$, with

$$H_A = A \left(1 - \sum_{n=1}^{W} y_n \right)^2 + A \left(\sum_{n=1}^{W} n y_n - \sum_{\alpha} w_\alpha x_\alpha \right)^2 \tag{49}$$

which enforces that the weight can only take on one value and that the weight of the objects in the knapsack equals the value we claimed it did, and finally

$$H_B = -B \sum_{\alpha} c_\alpha x_\alpha. \tag{50}$$

As we require that it is not possible to find a solution where H_A is weakly violated at the expense of H_B becoming more negative, we require $0 < B \max(c_\alpha) < A$ (adding one item to the knapsack, which makes it too heavy, is not allowed). The number of spins required is (using the log trick) $N + \lfloor 1 + \log W \rfloor$.

6. COLORING PROBLEMS
We now turn to coloring problems. Naively, coloring problems are often best phrased as Potts models [51], where the spins can

take on more than two values, but these classical Potts models can be converted to classical Ising models with an expansion of the number of spins. This simple trick forms the basis for our solutions to this class of problems.

6.1. GRAPH COLORING

Given an undirected graph $G = (V, E)$, and a set of n colors, is it possible to color each vertex in the graph with a specific color, such that no edge connects two vertices of the same color? This is one of the more famous NP-complete [18] problems, as one can think of it as the generalization of the problem of how many colors are needed to color a map, such that no two countries which share a border have the same color. Of course, in this special case[10], one can prove that there is always a coloring for $n \geq 4$ [52, 53]. This problem is called the graph coloring problem.

Our solution consists of the following: we denote $x_{v,i}$ to be a binary variable which is 1 if vertex v is colored with color i, and 0 otherwise. The energy is

$$H = A \sum_v \left(1 - \sum_{i=1}^n x_{v,i} \right)^2 + A \sum_{(uv) \in E} \sum_{i=1}^n x_{u,i} x_{v,i}. \qquad (51)$$

The first term enforces the constraint that each vertex has exactly one color, and provides an energy penalty each time this is violated, and the second term gives an energy penalty each time an edge connects two vertices of the same color. If there is a ground state of this model with $H = 0$, then there is a solution to the coloring problem on this graph with n colors. We can also read off the color of each node (in one such coloring scheme) by looking at which xs are 1. Note that the number of spins can be slightly reduced, since there is a permutation symmetry among colorings, by choosing a specific node in the graph to have the color 1, and one of its neighbors to have the color 2, for example. The total number of spins required is thus nN.

6.2. CLIQUE COVER

The clique cover problem, for an undirected graph $G = (V, E)$, is the following: given n colors, we assign a distinct color to each vertex of the graph. Let $W_1, \ldots, W_n$ be the subsets of V corresponding to each color, and $E_{W_1}, \ldots, E_{W_n}$ the edge set restricted to edges between vertices in the W_i sets. The clique cover problem asks whether or not (W_i, E_{W_i}) is a complete graph for each W_i (i.e., does each set of colored vertices form a clique?). This problem is known to be NP-complete [18].

Our solution is very similar to the graph coloring problem. Again, we employ the same binary variables as for graph coloring, and use a Hamiltonian very similar to the cliques problem:

$$H = A \sum_v \left(1 - \sum_{i=1}^n x_{v,i} \right)^2 + B \sum_{i=1}^n \left[\frac{1}{2} \left(-1 + \sum_v x_{v,i} \right) \right.$$
$$\left. \sum_v x_{v,i} - \sum_{(uv) \in E} x_{u,i} x_{v,i} \right]. \qquad (52)$$

___

[10]The graphs are *planar*—the vertices can be realized by points on $\mathbb{R}^2$, and the edges as line segments between them, such that no two line segments intersect (except at a vertex).

The first term enforces the constraint that each vertex has exactly one color by giving an energy penalty each time this constraint is violated. In the second term, since the sum over v of $x_{v,i}$ counts the number of nodes with color i, the first sum counts highest possible number of edges that could exist with color i. The second term then checks if, in fact, this number of edges does in fact exist. Thus $H = 0$ if and only if the clique cover problem is solved by the given coloring. If a ground state exists with $H = 0$, there is a solution to the clique covering problem. The discussion on the required ratio A/B to encode the correct solution is analogous to the discussion for the cliques problem. The total number of spins required is nN.

6.3. JOB SEQUENCING WITH INTEGER LENGTHS

The job sequencing problem is as follows: we are given a list of N jobs for m computer clusters. Each job i has length L_i. How can each job be assigned to a computer in the cluster such that, if the set of jobs on cluster α is V_α, then the length of that cluster, defined as

$$M_\alpha \equiv \sum_{i \in V_\alpha} L_i, \qquad (53)$$

are chosen such that $\max(M_\alpha)$ is minimized? Essentially, this means that if we run all of the jobs simultaneously, all jobs will have finished running in the shortest amount of time. This is NP-hard [18], and there is a decision version [is $\max(M_\alpha) \leq M_0$?] which is NP-complete. We assume that $L_i \in \mathbb{N}$.

To do this, we will begin by demanding that without loss of generality, $M_1 \geq M_\alpha$ for any α. Introduce the variables $x_{i,\alpha}$ which are 1 if job i is added to computer α, and 0 otherwise, and the variables $y_{n,\alpha}$ for $\alpha \neq 1$ and $n \geq 0$, which is 1 if the difference $M_1 - M_\alpha = n$. Then the Hamiltonian

$$H_A = A \sum_{i=1}^N \left(1 - \sum_\alpha x_{i,\alpha} \right)^2$$
$$+ A \sum_{\alpha=1}^m \left(\sum_{n=1}^{\mathcal{M}} n y_{n,\alpha} + \sum_i L_i (x_{i,\alpha} - x_{i,1}) \right)^2 \qquad (54)$$

encodes that each job can be given to exactly one computer, and that no computer can have a longer total length than computer 1. The number $\mathcal{M}$ must be chosen by the user, and is related to the number of auxiliary spins required to adequately impose the length constraints that $M_1 \geq M_\alpha$: in the worst case, it is given by $N \max(L_i)$. To find the minimal maximal length M_1, we just use

$$H_B = B \sum_i L_i x_{i,1}. \qquad (55)$$

Similarly to finding bounds on A/B for the knapsack problem, for this Hamiltonian to encode the solution to the problem, we require (in the worst case) $0 < B \max(L_i) < A$. Using the log trick, the number of spins required here is $mN + (m-1) \lfloor 1 + \log \mathcal{M} \rfloor$.

7. HAMILTONIAN CYCLES

In this section, we describe the solution to the (undirected or directed) Hamiltonian cycles problem, and subsequently the traveling salesman problem, which for the Ising spin glass formulation, is a trivial extension.

7.1. HAMILTONIAN CYCLES AND PATHS

Let $G = (V, E)$, and $N = |V|$. The graph can either be directed or undirected; our method of solution will not change. The Hamiltonian path problem is as follows: starting at some node in the graph, can one travel along an edge, visiting other nodes in the graph, such that one can reach every single node in the graph without ever returning to the same node twice? The Hamiltonian cycles problem asks that, in addition, the traveler can return to the starting point from the last node he visits. Hamiltonian cycles is a generalization of the famous Königsberg bridge problem [24], and is NP-complete [18].

Without loss of generality, let us label the vertices $1, \dots, N$, and take the edge set (uv) to be directed—i.e., the order uv matters. It is trivial to extend to undirected graphs, by just considering a directed graph with (vu) added to the edge set whenever (uv) is added to the edge set. Our solution will use N^2 bits $x_{v,i}$, where v represents the vertex and i represents its order in a prospective cycle. Our energy will have three components. The first two things we require are that every vertex can only appear once in a cycle, and that there must be a jth node in the cycle for each j. Finally, for the nodes in our prospective ordering, if $x_{u,j}$ and $x_{v,j+1}$ are both 1, then there should be an energy penalty if $(uv) \notin E$. Note that $N + 1$ should be read as 1, in the expressions below, if we are solving the cycles problem. These are encoded in the Hamiltonian:

$$H = A \sum_{v=1}^{n} \left(1 - \sum_{j=1}^{N} x_{v,j}\right)^2 + A \sum_{j=1}^{n} \left(1 - \sum_{v=1}^{N} x_{v,j}\right)^2$$
$$+ A \sum_{(uv) \notin E} \sum_{j=1}^{N} x_{u,j} x_{v,j+1}. \qquad (56)$$

$A > 0$ is a constant. It is clear that a ground state of this system has $H = 0$ only if we have an ordering of vertices where each vertex is only included once, and adjacent vertices in the cycle have edges on the graph—i.e., we have a Hamiltonian cycle.

To solve the Hamiltonian path problem instead, restrict the last sum over j above from 1 to $N - 1$; we do not care about whether or not the first and last nodes are also connected. N^2 spins are required to solve this problem.

It is straightforward to slightly reduce the size of the state space for the Hamiltonian cycles problem as follows: it is clear that node 1 must always be included in a Hamiltonian cycle, and without loss of generality we can set $x_{1,i} = \delta_{1,i}$: this just means that the overall ordering of the cycle is chosen so that node 1 comes first. This reduces the number of spins to $(N - 1)^2$.

7.2. TRAVELING SALESMAN

The traveling salesman problem for a graph $G = (V, E)$, where each edge uv in the graph has a weight W_{uv} associated to it, is to find the Hamiltonian cycle such that the sum of the weights of each edge in the cycle is minimized. Typically, the traveling salesman problem assumes a complete graph, but we have the technology developed to solve it on a more arbitrary graph. The decision problem (does a path of total weight $\leq W$ exist?) is NP-complete [18].

To solve this problem, we use $H = H_A + H_B$, with H_A the Hamiltonian given for the directed (or undirected) Hamiltonian cycles problem. We then simply add

$$H_B = B \sum_{(uv) \in E} W_{uv} \sum_{j=1}^{N} x_{u,j} x_{v,j+1}. \qquad (57)$$

with B small enough that it is never favorable to violate the constraints of H_A; one such constraint is $0 < B \max(W_{uv}) < A$ (we assume in complete generality $W_{uv} \geq 0$ for each $(uv) \in E$).[11] If the traveling salesman does not have to return to his starting position, we can restrict the sum over j from 1 to $N - 1$, as before. As with Hamiltonian cycles, $(N - 1)^2$ spins are required, as we may fix node 1 to appear first in the cycle.

8. TREE PROBLEMS

The most subtle NP problems to solve with Ising models are problems which require finding connected tree subgraphs of larger graphs.[12] Because determining whether a subgraph is a tree requires global information about the connectivity of a graph, we will rely on similar tricks to what we used to write down Hamiltonian cycles as an Ising model.

8.1. MINIMAL SPANNING TREE WITH A MAXIMAL DEGREE CONSTRAINT

The minimal spanning tree problem is the following: given an undirected graph $G = (V, E)$, where each edge $(uv) \in E$ is associated with a cost c_{uv}, what is the tree $T \subseteq G$, which contains all vertices, such that the cost of T, defined as

$$c(T) \equiv \sum_{(uv) \in E_T} c_{uv}, \qquad (58)$$

is minimized (if such a tree exists)? Without loss of generality, we take $c_{uv} > 0$ in this subsection (a positive constant can always be added to each c_{uv} ensure that the smallest value of c_{uv} is strictly positive, without changing the trees T which solve the problem). We will also add a degree constraint, that each degree in T be $\leq \Delta$. This makes the problem NP-hard, with a corresponding NP-complete decision problem [18].

To solve this problem, we place a binary variable y_e on each edge to determine whether or not that edge is included in T:

---

[11]One can also encode graph structure by assuming a complete graph (this allows one to neglect the third term in H_A), but choosing the weights of the "non-existent" edges to obey $W_{uv \notin E} \geq N \max(W_{uv \in E})$. As W_{uv} is not defined if $(uv) \notin E$, these are in fact two identical interpretations.

[12]A tree is a graph with no cycles. A cycle is set of vertices $v_1, \dots, v_n$ with $(v_1 v_2), \dots, (v_{n-1} v_n), (v_n v_1) \in E$. It is easy to check that if (V, E) is a tree, $|E| = |V| - 1$.

$$y_e \equiv \begin{cases} 1 & e \in E_T \\ 0 & \text{otherwise} \end{cases}. \tag{59}$$

We also place a large number of binary variables $x_{v,i}$ on each vertex, and $x_{uv,i}, x_{vu,i}$ on edge (uv) (these are distinct spins): the number $i = 0, 1, \ldots, N/2$ will be used to keep track of the depth a node in the tree, and if $x_{uv} = 1$, it means that u is closer to the root than v, and if $x_{vu} = 1$ it means that v is closer to the root. Finally, we use another variable $z_{v,i}$ $(i = 1, \ldots \Delta)$ to count the number of degrees of each node. We now use energy $H = H_A + H_B$, where the terms in H_A are used to impose the constraints that: there is exactly one root to the tree, each vertex has a depth, each bond has a depth, and its two vertices must be at different heights, the tree is connected (i.e., exactly one edge to a non-root vertex comes from a vertex at lower depth), each node can have at most Δ edges, and each edge at depth i points between a node at depth $i - 1$ and i, respectively:

$$H_A = A \left(1 - \sum_v x_{v,0} \right)^2 + A \sum_v \left(1 - \sum_i x_{v,i} \right)^2$$

$$+ A \sum_{uv \in E} \left(y_{uv} - \sum_i (x_{uv,i} + x_{vu,i}) \right)^2$$

$$+ A \sum_v \sum_{i=1}^{N/2} \left(x_{v,i} - \sum_{u:(uv) \in E} x_{uv,i} \right)^2$$

$$+ A \sum_v \left(\sum_{j=1}^{\Delta} j z_{v,j} - \sum_{u:(uv) \in E} \sum_i (x_{uv,i} + x_{vu,i}) \right)^2$$

$$+ A \sum_{(uv),(vu) \in E} \sum_{i=1}^{N/2} x_{uv,i} (2 - x_{u,i-1} - x_{v,i}) \tag{60}$$

The ground states with $H_A = 0$ are trees which include every vertex. In the last term in the sum, remember that $x_{uv,i}$ and $x_{vu,i}$ are both spins that are included for each edge; the notation in the summation is meant to remind us of this. We then add

$$H_B = B \sum_{uv,vu \in E} \sum_{i=1}^{N/2} c_{uv} x_{uv,i}. \tag{61}$$

In order to solve the correct problem, we need to make sure that we never remove any $x_{uv,i}$ from H_B in order to have a more negative total H. As each constraint in H_A contributes an energy $\geq A$ if it is violated, we conclude that setting $0 < B \max(c_{uv}) < A$ is sufficient. The minimum of E will find the minimal spanning tree, subject to the degree constraint.

The number of spins required is $|V|(\lfloor |V| + 1 \rfloor + 2)/2 + |E|(|V| + 1) + |V|\lfloor 1 + \log \Delta \rfloor$. The maximal possible number of edges on any graph is $|E| = O(|V|^2)$, so this Ising formulation may require a cubic number of spins in the size of the vertex set.

8.2. STEINER TREES

The NP-hard [18] Steiner tree problem is somewhat similar to the problem above: given our costs c_{uv}, we want to find a minimal spanning tree for a subset $U \subset V$ of the vertices (i.e., a tree such that the sum of c_{uv}s along all included edges is minimal). We no longer impose degree constraints; the problem turns out to be "hard" already, as we now allow for the possibility of not including nodes which are not in U.

To solve this by finding the ground state of an Ising model, we use the same Hamiltonian as for the minimal spanning tree, except we add binary variables y_v for $v \notin U$ which determine whether or not a node v is included in the tree. We use the Hamiltonian $H = H_A + H_B$, where H_A enforces constraints similarly to in the previous case:

$$H_A = A \left(1 - \sum_v x_{v,0} \right)^2 + A \sum_{v \in U} \left(1 - \sum_i x_{v,i} \right)^2$$

$$+ A \sum_{v \notin U} \left(y_v - \sum_i x_{v,i} \right)^2$$

$$+ A \sum_v \sum_{i=1}^{N/2} \left(x_{v,i} - \sum_{(uv) \in E} x_{uv,i} \right)^2$$

$$+ A \sum_{uv,vu \in E} \sum_{i=1}^{N/2} x_{uv,i} (2 - x_{u,i-1} - x_{v,i})$$

$$+ A \sum_{uv \in E} \left(y_{uv} - \sum_i (x_{uv,i} + x_{vu,i}) \right)^2 \tag{62}$$

We then use H_B from the previous model to determine the minimum weight tree; the same constraints on A/B apply. The number of spins is $|V|(\lfloor |V| + 1 \rfloor + 4 + 2|E|)/2 + |E|$.

8.3. DIRECTED FEEDBACK VERTEX SET

A feedback vertex set for a directed graph $G = (V, E)$ is a subset $F \subset V$ such that the subgraph $(V - F, \partial(V - F))$ is acyclic (has no cycles). We will refer to F as the feedback set. Solving a decision problem for whether or not a feedback set exists for $|F| \leq k$ is NP-complete [18]. We solve the optimization problem of finding the smallest size of the feedback set first for a directed graph—the extension to an undirected graph will be a bit more involved.

Before solving this problem, it will help to prove two lemmas. The first lemma is quite simple: there exists a node in a directed acyclic graph which is not the end point of any edges. Suppose that for each vertex, there was an edge that ends on that vertex. Then pick an arbitrary vertex, pick any edge ending on that vertex, and follow that edge in reverse to the starting vertex. Repeat this process more than N times, and a simple counting argument implies that we must have visited the same node more than once, at least once. Thus, we have traversed a cycle in reverse, which contradicts our assumption.

The second lemma is as follows: a directed graph $G = (V, E)$ is acyclic if and only if there is a height function $h : V \to \mathbb{N}$ such that if $uv \in E$, $h(u) < h(v)$: i.e., every edge points from a node at

lower height to one at higher height. That height function existence implies acyclic is easiest to prove using the contrapositive: suppose that a graph is cyclic. Then on a cycle of edges, we have

$$0 < \sum [h(u_{i+1}) - h(u_i)] = h(u_1) - h(u_n) + h(u_n)$$
$$- h(u_{n-1}) + \cdots - h(u_1) = 0 \tag{63}$$

is a contradiction. To prove that an acyclic graph has a height function, we construct one recursively. Using our first lemma, we know that there exists a vertex u with only outgoing edges, so let us call $h(u) = 1$. For any other vertex, we will call the height of that vertex $h(v) = 1 + h'(v)$, where $h'(v)$ is found by repeating this process on the graph with node u removed (which must also be acyclic). It is clear this process will terminate and assign exactly one node height i for each integer $1 \leq i \leq |V|$.

We can now exploit this lemma to write down an Ising spin formulation of this problem. We place a binary variable y_v on each vertex, which is 0 if v is part of the feedback set, and 1 otherwise. We then place a binary variable $x_{v,i}$ on each vertex, which is 1 if vertex v is at height i. So far the heights i are arbitrary, and the requirement that a height function be valid will be imposed by the energy. The energy functional we use is $H = H_A + H_B$ where

$$H_A = A \sum_v \left(y_v - \sum_i x_{v,i} \right)^2 + A \sum_{uv \in E} \sum_{i \geq j} x_{u,i} x_{v,j}. \tag{64}$$

The first term ensures that if a vertex is not part of the feedback set, it has a well-defined height; the second term ensures that an edge only connects a node with lower height to a node at higher height. We then find the smallest possible feedback set by adding

$$H_B = B \sum_v (1 - y_v). \tag{65}$$

In order to solve the correct problem, we cannot add too few nodes to the feedback set. If we set $y_v = 1$ for a node which should be part of the feedback set, we find an energy penalty of A from H_A, and a gain of B from H_B. We conclude that $B < A$ is sufficient to ensure we solve the correct problem. We see that $|V|(|V| + 1)$ spins are required.

8.4. UNDIRECTED FEEDBACK VERTEX SET
The extension to undirected graphs requires a bit more care. In this case, we have to be careful because there is no *a priori* distinction on whether the height of one end of an edge is smaller or larger than the other—this makes the problem much more involved, at first sight. Furthermore, it is not true that a directed acyclic graph is acyclic if the orientation of edges is ignored. However, for an undirected graph, we also know that a feedback vertex set must reduce the graph to trees, although there is no longer a requirement that these trees are connected (this is called a forest). With this in mind, we find that the problem is actually extremely similar to minimal spanning tree, but without degree constraints or connectivity constraints. The new subtlety, however, is that we cannot remove edges.

To solve this problem, we do the following: introduce a binary variable $x_{v,i}$, which is 1 if v is a vertex in any tree (anywhere in the forest) at depth i, and 0 otherwise. However, to account for the fact that we may remove vertices, we will allow for $y_v = 1$ if v is part of the feedback vertex set, and 0 otherwise. We do a similar thing for edges: we consider $x_{uv,i}, x_{vu,i}$ to be defined as before when $i > 0$. We also define the variables y_{uv}, y_{vu}, which we take to be 1 when the ending node of the "directed" edge is in the feedback vertex set. Now, we can write down a very similar energy to the minimal spanning tree:

$$H_A = A \sum_v \left(1 - y_v - \sum_i x_{v,i} \right)^2$$
$$+ A \sum_{uv \in E} \left(1 - \sum_i (x_{uv,i} + x_{vu,i} + y_{uv} + y_{vu}) \right)^2$$
$$+ A \sum_{uv \in E} (y_{uv} - y_v)^2 + A \sum_v \sum_{i > 0} \left(x_{v,i} - \sum_{u:uv \in, E} x_{uv,i} \right)^2$$
$$+ A \sum_{uv, vu \in E} \sum_{i > 0} x_{uv,i} (2 - x_{u,i-1} - x_{v,i}) \tag{66}$$

The changes are as follows: we no longer constrain only 1 node to be the root, or constrain the degree of a vertex—however, we have to add a new term to ensure that edges are only ignored in the tree constraint if they point to a node in the feedback set. We then add

$$H_B = B \sum_v y_v \tag{67}$$

with $B < A$ required so that the A constraints are never violated. This counts the number of nodes in the feedback set, so thus H is minimized when H_B is smallest—i.e., we have to remove the fewest number of nodes. The number of spins required is $(|E| + |V|)\lceil (|V| + 3)/2 \rceil$ [13].

The recent paper [54] has a more efficient implementation of a mapping, for use in understanding random ensembles of this problem by the replica method. Unfortunately, this technique is not efficient for AQO; the Hamiltonian contains N-body terms.

8.5. FEEDBACK EDGE SET
For a directed graph, the feedback edge set problem is to find the smallest set of edges $F \subset E$ such that $(V, E - F)$ is a directed acyclic graph. It is known to be NP-hard [18][14]. Our solution will be somewhat similar to the directed feedback vertex set. We place a binary variable y_{uv} on each edge, which is 1 if $uv \notin F$, and define $x_{uv,i}$ to be 1 if both $y_{uv} = 1$ and the height of node u is i. We also add a binary variable $x_{v,i}$, as for the feedback vertex set. Our constraint energy must then enforce that: each vertex and included edge has a well-defined height, and that each edge points from a

---

[13]This follows from the fact that the sum over is is $\lceil (|V| + 1)/2 \rceil$; then we account for the ys.

[14]It is in P if the graph is undirected however.

lower height to a higher height:

$$H_A = A \sum_v \left(1 - \sum_i x_{v,i}\right)^2 + A \sum_{uv \in E} \left(y_{uv} - \sum_i x_{uv,i}\right)^2$$

$$+ A \sum_{uv} \sum_i x_{uv,i} \left(2 - x_{u,i} - \sum_{j > i} x_{v,j}\right). \quad (68)$$

We then use

$$H_B = B \sum_{uv \in E} (1 - y_{uv}) \quad (69)$$

to count the number of edges in F—it is minimized when this number is smallest. As before, one needs $B < A$ to encode the correct problem. The number of spins required is $|E| + |V|$ ($|V| + |E|$).

9. GRAPH ISOMORPHISMS

Graphs G_1 and G_2, with N vertices each, are isomorphic if there is a labeling of vertices $1, \ldots, N$ in each graph such that the adjacency matrices for the graphs is identical. More carefully: any graph $G = (V, E)$, with vertices labeled as $1, \ldots, N$, has an $N \times N$ adjacency matrix A with

$$A_{ij} = \begin{cases} 1 & (ij) \in E, \\ 0 & (ij) \notin E. \end{cases}, \quad (70)$$

which contains all information about the edge set E. Let $\mathsf{A}_{1,2}$ be the adjacency matrices of graphs $G_{1,2}$. If there is a permutation matrix P such that $\mathsf{A}_2 = \mathsf{P}^\mathsf{T} \mathsf{A}_1 \mathsf{P}$, then we say $G_{1,2}$ are isomorphic.

The question of whether two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic is believed to be hard, but its classification into a complexity class is still a mystery [55]. Since it is (in practice) a hard problem, let us nonetheless describe an Ising formulation for it. An isomorphism is only possible if $|V_1| = |V_2| \equiv N$, so we will restrict ourselves to this case, and without loss of generality, we label the vertices of G_1 with $1, \ldots, N$.

We write this as an Ising model as follows. Let us describe a proposed isomorphism through binary variables $x_{v,i}$ which is 1 if vertex v in G_2 gets mapped to vertex i in G_1. The energy

$$H_A = A \sum_v \left(1 - \sum_i x_{v,i}\right)^2 + A \sum_i \left(1 - \sum_v x_{v,i}\right)^2 \quad (71)$$

ensures that this map is bijective. We then use an energy

$$H_B = B \sum_{ij \notin E_1} \sum_{uv \in E_2} x_{u,i} x_{v,j} + B \sum_{ij \in E_1} \sum_{uv \notin E_2} x_{u,i} x_{v,j} \quad (72)$$

to penalize a bad mapping: i.e., an edge that is not in G_1 is in G_2, or an edge that is in G_1 is not in G_2. As usual, assume $A, B > 0$.

If the ground state of this Hamiltonian has $H = 0$, there is an isomorphism. N^2 spins are required.

An approximate algorithm that uses quantum annealing to distinguish between non-isomorphic graphs via the spectra of graph-dependent Hamiltonians was presented in Hen and Young [56].

10. CONCLUSION

The focus of research into AQO has essentially been on NP-complete/hard problems, because the Ising model is NP-hard, and because computer scientists have struggled to find efficient ways of solving these problems. In this paper, we have presented strategies for mapping a wide variety of NP problems to Ising spin glasses, exemplified by a demonstration of a glass for each of Karp's 21 NP-complete problems. It is an open question the extent to which AQO will help provide efficient solutions for these problems, whether these solutions are exact or approximate.

However, physicists are interested in building a universal quantum computer which is capable of solving much more than just Ising models. As an example, a universal quantum computer would also reduce the time for searching an unsorted list of N items from $O(N)$ to $O(\sqrt{N})$ [57]. This would be incredibly useful for many practical applications, despite the fact that searching is an easy linear time algorithm. Analogously, it may be the case that there exists a family of "easy" problems which AQO can solve in polynomial time, yet more efficiently than a classical polynomial time algorithm. This statement may even be true with Ising-implementing AQO hardware, although if so it is not obvious.

It is certainly the case that an AQO-implementing device can be used to solve easy problems. Consider the simple problem of finding the largest integer in a list $n_1, \ldots, n_N$ (this is the searching algorithm that a universal quantum computer can perform efficiently). Introducing binary variables x_i for $i = 1, \ldots, N$, the Ising model

$$H = A \left(1 - \sum_i x_i\right)^2 - B \sum_i n_i x_i \quad (73)$$

for $A > B \max(n_i)$ solves this problem. In fact, this problem looks somewhat like an instance of the random field Ising model on a complete graph, and yet this has a very simple $O(N)$ classical algorithm. It would surely take longer to program this algorithm into a quantum device than to solve the problem itself.

The above example demonstrates that sometimes the "hardness" of a problem can be deceptive—one can phrase something that is easy in a way which makes it seem hard. It is worth discussing more closely the hardness of NP problems, because it turns out that sometimes, NP problems can be easier than they first appear. To be NP-complete but not P (if P $\neq$ NP) one only needs a small family of instances of the problem to be unsolvable in polynomial time by a deterministic algorithm. However, typical instances may not be so hard. Many popular NP problems can almost surely be solved exactly in polynomial time on

large random instances [58, 59] [15], and there exist randomized algorithms for some NP problems which can get arbitrarily close to a solution with arbitrarily low failure probability in polynomial time [60, 61] (though multiplicative coefficients or polynomial exponents must diverge as the failure probability and/or error on determining the ground state tends to zero, if P ≠ NP). In addition, popular algorithms in P, like matrix decomposition, may serve as the bottlenecks of *practical* computations, and should not be thought of as "easy". Typical instances approach the asymptotic bounds on worst-case runtimes, in contrast to the case for some NP problems; many recent developments focus on randomized algorithms [62–64].

The Hamiltonians of this paper may be deceptively "hard"—this can mean that they involve too many spins. Another possibility is that these Hamiltonians have small spectral gaps, and that alternative choices have much larger spectral gaps—this is a question we have not addressed at all in this paper. Studying how to simplify quantum algorithms, and more importantly increase energy gaps (and thus reduce T), even by constant factors, is a much needed endeavor.

MATERIALS AND METHODS

This paper discusses theoretical results; no materials are needed. The methodology used was discussed throughout the paper.

REFERENCES

1. Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A, Preda D. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* (2001) **292**:472. doi: 10.1126/science.1057726

2. Das A, Chakrabarti BK. *Colloquium:* quantum annealing and analog quantum computation. *Rev Mod Phy.* (2008) **80**:1061. doi: 10.1103/RevModPhys.80.1061

3. Altshuler B, Krovi H, Roland J. Anderson localization makes adiabatic quantum optimization fail. *Proc Natl Acad Sci USA.* (2010) **107**:12446. doi: 10.1073/pnas.1002116107

4. Dickson NG, Amin MHS. Does adiabatic quantum optimization fail for NP-complete problems?. *Phys Rev Lett.* (2011) **106**:050502. doi: 10.1103/PhysRevLett.106.050502

5. Bapst V, Foini L, Krzakala F, Semerjian G, Zamponi F. The quantum adiabatic algorithm applied to random optimization problems: the quantum spin glass perspective. *Phys Rep.* (2013) **523**:127. doi: 10.1016/j.physrep.2012.10.002

6. Farhi E, Goldstone J, Gosset D, Gutmann S, Shor P. Unstructured randomness, small gaps and localization. *Quantum Comput Inform.* (2011) **11**:840. Available online at: http://arxiv.org/abs/1010.0009

7. Farhi E, Gosset D, Hen I, Sandvik AW, Shor P, Young AP, et al. The performance of the quantum adiabatic algorithm on random instances of two optimization problems on regular hypergraphs. *Phys Rev.* (2012) **A86**:052334. doi: 10.1103/PhysRevA.86.052334

8. Hen I, Young AP. Exponential complexity of the quantum adiabatic algorithm for certain satisfiability problems. *Phys Rev.* (2011) **E84**:061152. doi: 10.1103/PhysRevE.84.061152

9. Jorg T, Krzakala F, Semerjian G, Zamponi F. First-order transitions and the performance of quantum algorithms in random optimization problems. *Phys Rev Lett.* (2010) **104**:207206. doi: 10.1103/PhysRevLett.104.207206

10. Santoro GE, Martonak R, Tosatti E, Car R. Theory of quantum annealing of an Ising spin glass. *Science* (2002) **295**:2427. doi: 10.1126/science.1068774

11. Boixo S, Albash T, Spedalieri FM, Chancellor N, Lidar DA. Experimental signature of programmable quantum annealing. *Nat Commun.* (2013) **4**:3067. doi: 10.1038/ncomms3067

12. Boixo S, Rønnow TF, Isakov SV, Wang Z, Wecker D, Lidar DA, et al. Quantum annealing with more than one hundred qubits (2013). Available online at: http://arxiv.org/abs/1304.4595

13. Johnson MW, Amin MHS, Gildert S, Lanting T, Hamze F, Dickson N, et al. Quantum annealing with manufactured spins. *Nature* (2011) **473**:194. doi: 10.1038/nature10012

14. Whitfield JD, Faccin M, Biamonte JD. Ground state spin logic. *Europhys Lett.* (2012) **99**:57004. doi: 10.1209/0295-5075/99/57004

15. Biamonte JD, Love PJ. Realizable Hamiltonians for universal adiabatic quantum computers. *Phys Rev.* (2008) **A78**:012352. doi: 10.1103/PhysRevA.78.012352

16. Bravyi S, DiVincenzo DP, Oliveira RI, Terhal BM. The complexity of stoquastic local Hamiltonian problems. *Quantum Inform Comput.* (2008) **8**:0361. Available online at: http://arxiv.org/abs/quant-ph/0606140

17. Barahona F. On the computational complexity of Ising spin glass models. *J Phys.* (1982) **A15**:3241.

18. Karp RM. Reducibility among combinatorial problems. In: Miller RE, Thatcher JW, and Bohlinger JD, editors. *Complexity of Computer Computations*. New York, NY: Plenum Press (1972), p. 85. doi: 10.1007/978-1-4684-2001-2_9

19. Garey MR, Johnson DS. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA : W.H. Freeman (1979).

20. Fu Y, Anderson PW. Application of statistical mechanics to NP-complete problems in combinatorial optimisation. *J Phys.* (1986) **A19**:1605.

21. Mézard M, Parisi G, Virasoro M. *Spin Glass Theory and Beyond*. Singapore: World Scientific (1987).

22. Hartmann AK, Weigt M. *Phase Transitions in Combinatorial Optimization Problems: Basics, Algorithms and Statistical Mechanics*. Weinheim: Wiley-VCH (2006).

23. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science* (1983) **220**:671. doi: 10.1126/science.220.4598.671

24. Mézard M, Montanari A. *Information, Physics and Computation*. Oxford: Oxford University Press (2009). doi: 10.1093/acprof:oso/9780198570837.001.0001

25. Bryngelson JD, Wolynes PG. Spin glasses and the statistical mechanics of protein folding. *Proc Natl Acad Sci USA.* (1987) **84**:7524. doi: 10.1073/pnas.84.21.7524

26. Berger B, Leighton T. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. *J Comput Biol.* (1998) **5**:27. doi: 10.1089/cmb.1998.5.27

27. Hopfield JJ. Neural networks and physical systems with emergent collective computational abilities. *Proc Natl Acad Sci USA.* (1982) **79**:2554. doi: 10.1073/pnas.79.8.2554

28. Bouchaud JP. Crises and collective socio-economic phenomena: simple models and challenges. *J Stat Phys.* (2013) **151**:567. doi: 10.1007/s10955-012-0687-3

29. Lucas A, Lee CH. Multistable binary decision making on networks. *Phys Rev.* (2013) **E87**:032806. doi: 10.1103/PhysRevE.87.032806. Available online at: http://arxiv.org/abs/1210.6044

30. Xu N, Zhu J, Lu D, Zhou X, Peng X, Du J. Quantum factorization of 143 on a dipolar-coupling nuclear magnetic resonance system. *Phys Rev Lett.* (2012) **108**:130501 [*Erratum* (2012) **109**:269902E]. doi: 10.1103/PhysRevLett.108.130501

31. Bian Z, Chudak F, Macready WG, Clark L, Gaitan F. Experimental determination of Ramsey numbers. *Phys Rev Lett.* (2013) **111**:130505. doi: 10.1103/PhysRevLett.111.130505

---

[15]One has to be careful with the phrases "random" and "typical", as this immediately implies a probability distribution over a space of problem instances. This probability distribution may place vanishingly small probability on a set of relevant instances for any given application. For the simple probability distributions used in these papers, it is highly non-trivial that most instances turn out to be solvable in polynomial time.

32. Perdomo-Ortiz A, Dickson N, Drew-Brook M, Rose G, Aspuru-Guzik A. Finding low-energy conformations of lattice protein models by quantum annealing. *Sci Rep.* (2012) **2**:571. doi: 10.1038/srep00571

33. Babbush R, Perdomo-Ortiz A, O'Gorman B, Macready W, Aspuru-Guzik A. Construction of energy functions for lattice heteropolymer models: a case study in constraint satisfaction programming and adiabatic quantum optimization (2013). Available online at: http://arxiv.org/abs/1211.3422

34. Neven H, Rose G, Macready WG. Image recognition with an adiabatic quantum computer I. Mapping to quadratic unconstrained binary optimization (2008). Available online at: http://arxiv.org/abs/0804.4457

35. Denchev V, Ding N, Vishwanathan SVN, Neven H. "Robust classification with adiabatic quantum optimization." *Proceedings of the 29th International Conference on Machine Learning.* (Edinburgh), p. 863 (2012).

36. Boros E, Hammer PL. The max-cut problem and quadratic 0-1 optimization; polyhedral aspects, relaxations and bounds. *Annal Operat Res.* (1991) **33**:151. doi: 10.1007/BF02115753

37. Boros E, Hammer PL. Pseudo-Boolean optimization. *Discrete Appl Math.* (2002) **123**:155. doi: 10.1016/S0166-218X(01)00341-9

38. Boros E, Hammer PL, Tavares G. Preprocessing of unconstrained quadratic binary optimization. *RUTCOR Research Report* 10-2006 (2006).

39. Billionnet A, Jaumard B. A decomposition method for minimizing quadratic pseudo-Boolean functions. *Operat Res Lett.* (1989) **8**:161. doi: 10.1016/0167-6377(89)90043-6

40. Biamonte JD. Non-perturbative *k*-body to two-body commuting conversion Hamiltonians and embedding problem instances into Ising spins. *Phys Rev.* (2008) **A77**:052331. doi: 10.1103/PhysRevA.77.052331

41. Babbush R, O'Gorman B, Aspuru-Guzik A. Resource efficient gadgets for compiling adiabatic quantum optimization problems. *Annal. Phys.* (2013) **525**:877. doi: 10.1002/andp.201300120

42. Peng X, Liao Z, Xu N, Qin G, Zhou X, Suter D, et al. A quantum adiabatic algorithm for factorization and its experimental implementation. *Phys Rev Lett.* (2008) **101**:220405. doi: 10.1103/PhysRevLett.101.220405

43. Choi V. Minor-embedding in adiabatic quantum computation: I the parameter setting problem. *Quantum Inform Process.* (2008) **7**:193. doi: 10.1007/s11128-008-0082-9

44. Choi V. Minor-embedding in adiabatic quantum computation: II Minor-universal graph design. *Quantum Inform Process.* (2011) **10**:343. doi: 10.1007/s11128-010-0200-3

45. Klymko C, Sullivan BD, Humble TS. Adiabatic quantum programming: minor embedding with hard faults. *Quantum Inform Process.* Available online at: http://arxiv.org/abs/1210.8395

46. Alon N, Krivelevich M, Sudakov B. Finding a large hidden clique in a random graph. *Random Struct Algorith.* (1998) **13**:457. doi: 10.1002/(SICI)1098-2418(199810/12)13:3/4<457::AID-RSA14>3.3.CO;2-K

47. Childs AM, Farhi E, Goldstone J, Gutmann S. Finding cliques by quantum adiabatic evolution. *Quantum Inform Comput.* (2002) **2**:181. Available online at: http://arxiv.org/abs/quant-ph/0012104

48. Schrijver A. *Theory of Integer and Linear Programming.* Chichester, NY: Wiley (1998).

49. Choi V. Adiabatic quantum algorithms for the NP-complete maximum-weight independent set, exact cover and 3SAT problems (2010). Available online at: http://arxiv.org/abs/1004.2226

50. Kellerer H, Pferschy U. *Knapsack Problems.* Berlin: Springer (2004). doi: 10.1007/978-3-540-24777-7

51. Wu FY. The Potts model. *Rev Mod Phys.* (1982) **54**:1. doi: 10.1103/RevModPhys.54.235

52. Appel K, Haken W. Every planar map is four colorable I: discharging. *Illinois J Math.* (1977) **21**:429.

53. Appel K, Haken W, Koch J. Every planar map is four colorable II Reducibility. *Illinois J Math.* (1977) **21**:491.

54. Zhou HJ. Spin glass approach to the feedback vertex set problem. *Eur Phys J* (2013) **B86**:455. doi: 10.1140/epjb/e2013-40690-1

55. Johnson DS. The NP-completeness column. *ACM Trans Algorith.* (2005) **1**:160. doi: 10.1145/1077464.1077476

56. Hen I, Young AP. Solving the graph isomorphism problem with a quantum annealer. *Phys Rev.* (2012) **A86**:042310. doi: 10.1103/PhysRevA.86.042310

57. Nielsen MA, Chuang IA. *Quantum Computation and Quantum Information.* Cambridge: Cambridge University Press (2000).

58. Beier R, Vöcking B. "Random knapsack in expected polynomial time." *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing,* (San Diego, CA), p. 232 (2004).

59. Krivelevich M, Vilenchik D. "Solving random satisfiable 3CNF formulas in expected polynomial time." *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms* (Miami, FL), p. 454 (2006). doi: 10.1145/1109557.1109608

60. Dyer M, Frieze A, Kannan R. A random polynomial-time algorithm for approximating the volume of convex bodies. *J ACM.* (1991) **38**:1. doi: 10.1145/102782.102783

61. Vazirani VV. *Approximation Algorithms.* Berlin: Springer (2003). doi: 10.1007/978-3-662-04565-7

62. Liberty E, Woolfe F, Martinsson PG, Rokhlin V, Tygert M. Randomized algorithms for the low-rank approximation of matrices. *Proc Natl Acad Sci USA.* (2007) **104**:20167. doi: 10.1073/pnas.0709640104

63. Halko N, Martinsson P, Tropp JA. Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* (2011) **53**:217. doi: 10.1137/090771806

64. Lucas A, Stalzer M, Feo J. Parallel implementation of a fast randomized algorithm for the decomposition of low rank matrices. *Parallel Process Lett.* (accepted). Available online at: http://arxiv.org/abs/1205.3830