SCIENTIFIC REP🝔RTS

OPEN

Solving Set Cover with Pairs Problem using Quantum Annealing

Yudong Cao[1], Shuxian Jiang[1], Debbie Perouli[2] & Sabre Kais[3,4]

Here we consider using quantum annealing to solve Set Cover with Pairs (SCP), an NP-hard combinatorial optimization problem that plays an important role in networking, computational biology, and biochemistry. We show an explicit construction of Ising Hamiltonians whose ground states encode the solution of SCP instances. We numerically simulate the time-dependent Schrödinger equation in order to test the performance of quantum annealing for random instances and compare with that of simulated annealing. We also discuss explicit embedding strategies for realizing our Hamiltonian construction on the D-wave type restricted Ising Hamiltonian based on Chimera graphs. Our embedding on the Chimera graph preserves the structure of the original SCP instance and in particular, the embedding for general complete bipartite graphs and logical disjunctions may be of broader use than that the specific problem we deal with.

Quantum annealing (QA) uses the principles of quantum mechanics for solving unconstrained optimization problems[1–4]. Since the initial proposal of QA, there has been much interest in the search for practical problems where it can be advantageous with respect to classical algorithms[4–33], particularly simulated annealing (SA)[34–36]. Extensive theoretical, numerical and experimental efforts have been dedicated to studying the performance of quantum annealing on problems such as satisfiability[37–39], exact cover[3,39], max independent set[39], max clique[40], integer factorization[41], graph isomorphism[42,43], ramsey number[44], binary classification[45,46], unstructured search[47] and search engine ranking[48]. Many of these approaches[3,37,38,40–46] recast the computational problem at hand into a problem of finding the ground state of a quantum Ising spin glass model, which is NP-complete to solve in the worst case[49,50].

The computational difficulty of Ising spin glass has not only given the quantum Ising Hamiltonians the versatility for efficiently encoding many problems in NP[50], but also motivated physical realization of QA using systems described by the quantum Ising model[6,7,9]. The notion of adiabatic quantum computing (AQC)[3,37,51], which can be regarded as a particular class of QA, has further established QA in the context of quantum computation (In this work we will use the terms quantum annealing and adiabatic quantum computing synonymously). Although it is believed that even universal quantum computers cannot solve NP-hard problems efficiently in general[52], there has been evidence in experimental quantum Ising systems that suggests quantum speedup over classical computation due to quantum tunneling[53,54]. It is then of great interest to explore more regimes where quantum annealing could offer a speedup compared with simulated annealing.

Here we consider a variant of Set Cover (SC) called Set Cover with Pairs (SCP). SC is one of Karp's 21 NP-complete problems[55] and SCP was first introduced[56] as a generalization of SC. Instead of requiring each element to be covered by a single object as in SC, the SCP problem is to find a minimum subset of objects so that each element is covered by *at least one pair* of objects. We will present its formal definition in the Preliminaries section. SCP and its variants arise in a wide variety of contexts including Internet traffic monitoring and content distribution[57], computational biology[58,59], and biochemistry[60]. On classical computers, the SCP problem is at least as hard as SC. Specifically, its difficulty on classical computers can be manifested in the results by Breslau *et al.*[57], which showed that no polynomial time algorithm can approximately solve Disjoint-Path Facility Location, a special case of SCP, on n objects to within a factor that is $2^{\log^{1-\varepsilon} n}$ for any $\varepsilon > 0$. Due to its complexity, various heuristics[56] and local search algorithms[60] have been proposed.

In this paper we explore using quantum annealing based on Ising spin glass to solve SCP. We start by reducing SCP to finding the ground state of Ising spin glass, via integer linear programming (Theorem 1). We then simulate

[1]Department of Computer Science, Purdue University, West Lafayette, IN 47906, USA. [2]Department of Mathematics, Statistics and Computer Science, Marquette University, Milwaukee, WI 53233, USA. [3]Department of Chemistry, Physics and Birck Nanotechnology Center, Purdue University, West Lafayette, IN 47906, USA. [4]Qatar Energy and Environment Research Institute (QEERI), HBKU, Doha, Qatar. Correspondence and requests for materials should be addressed to Y.C. (email: cao23@purdue.edu) or S.J. (email: jiang97@purdue.edu)

the adiabatic evolution of the time dependent transverse Ising Hamiltonian $H(s) = (1 - s)H_0 + sH_1$ which interpolates linearly between an initial Hamiltonian H_0 of independent spins in uniform transverse field and a final Hamiltonian H_1 that encodes an SCP instance. For randomly generated SCP instances that lead to Ising Hamiltonian constructions of up to 19 spins, we explicitly simulate the time dependent Schrödinger equation. We compute the minimum evolution time that each instance needed to accomplish 25% success probability. For benchmark purpose we also use simulate annealing to solve the instances and compare its performance with that of adiabatic evolution. Results show that the median time for yielding 25% success probablity scales as $O(2^{0.33M})$ for quantum annealing and $O(2^{0.21M})$ for simulated annealing, observing no general quantum speedup. However, the performance of quantum annealing appears to have wider range of variance from instance to instance than simulated annealing, casting hope that perhaps certain subsets of the instance could yield a quantum advantage over the classical algorithms.

Aside from the theoretical and numerical studies, we also consider the potential implementation our Hamiltonian construction on the large-scale Ising spin systems manufactured by D-Wave Systems[6,7,9,14]. Benchmarking the efficiency of QA is currently of significant interest. An important issue that needs to be addressed in such benchmarks is that the physical implementation of the algorithm could be affected by instance-specific features. This is manifested in the embedding[61,62] of the Ising Hamiltonian construction onto the specific topology of the hardware (the Chimera graph[21,61,63]). Here we present a general embedding of SCP instances onto a Chimera graph that preserves the original structure of the instances and requires less qubits than the usual approach by complete graph embedding. This allows for efficient physical implementations that are untainted by ad hoc constructions that are specific to individual instances.

Preliminaries

Set Cover with Pairs.

Given a *ground set* U and a collection S of subsets of U, which we call the *cover set*. Each element in S has a non-negative weight, the Set Cover (SC) problem asks to find a minimum weight subset of S that covers all elements in U. Define *cover function* as $Q: S \mapsto 2^U$ where $\forall s \in S$, $Q(s)$ is the set of all elements in U covered by s. Then SC can be formulated as finding a minimum weight $S' \subseteq S$ such that $Q(S') = \bigcup_{s' \in S'} Q(s') = U$. Set Cover with Pairs (SCP) can be considered as a generalization of SC in the sense that if we define the cover function such that $\forall i, j \in S$, $i \neq j$, $Q(i, j)$ is the set of elements in U covered by the pair $\{i, j\}$, then SCP asks to find a minimum subset $A \subseteq S$ such that $Q(A) = \bigcup_{\{i,j\} \in S} Q(i, j) = U$. Here we restrict to cases where each element of S has unit weight.

A *graph* $G(V, E)$ is a set of vertices V connected by a set of edges E. A *bipartite graph* is defined as a graph whose set of vertices V can be partitioned into two disjoint sets V_1 and V_2 such that no two vertices within the same set are adjacent. We formally define SCP as the following.

Definition 1. (*Set Cover with Pairs*) *Let U and S be disjoint sets of elements and $V = U \cup S$. Given a bipartite graph $G(V, E)$ between U and S with E being the set of all edges, find a subset $A \subseteq S$ such that*:

1. $\forall c_i \in U, \exists\, a_1^{(i)}, a_2^{(i)} \in A$ *such that* $(a_1^{(i)}, c_i) \in E$ *and* $(a_2^{(i)}, c_i) \in E$. *In other words, c_i is covered by the pair* $\{a_1^{(i)}, a_2^{(i)}\}$.
2. *The size of the set, $|A|$, is minimized.*

We use the notation SCP(G, U, S) *to refer to a problem instance with $|U| = n$, $|S| = m$ and the connectivity between U and S determined by G.*

Quantum annealing, adiabatic quantum computing.

In this paper we use QA as a heuristic method to solve the SCP problem. QA was proposed[2] for solving optimization problems using quantum fluctuations, known as quantum tunneling, to escape local minima and discover the lowest energy state. Farhi *et al.*[3] provide the framework for using Adiabatic Quantum Computation (AQC), which is closely related to QA, as a quantum paradigm to solve NP-hard optimization problems. The first step of the framework is to define a Hamiltonian H_P whose ground state corresponds to the solution of the combinatorial optimization problem. Then, we initialize a system in the ground state of some beginning Hamiltonian H_B that is easy to solve, and perform the adiabatic evolution $H(s) = (1 - s)H_B + sH_P$. Here $s \in [0, 1]$ is a time parameter. In this paper we only consider time-dependent function $s(t) = t/T$ for total evolution time T, but in general it could be any general functions that satisfy $s(0) = 0$ and $s(T) = 1$. The adiabatic evolution is governed by the Schrödinger equation

$$i\frac{d}{dt}|\psi(t)\rangle = H(s(t))|\psi(t)\rangle \tag{1}$$

where $|\psi(t)\rangle$ is the state of the system at any time $t \in [0, T]$. Let $\pi_i(s)$ be the i-th instantaneous eigenstate of $H(s)$. In other words, let $H(s)\pi_i(s) = E_i(s)\pi_i(s)$ for any s. In particular, let $|\pi_0(s)\rangle$ be the instantaneous ground state of $H(s)$.

According to the adiabatic theorem[64], for s varying sufficiently slow from 0 to 1, the state of the system $|\psi(t)\rangle$ will remain close to the true ground state $|\pi_0(s(t))\rangle$. At the end of the evolution the system is roughly in the ground state of H_P, which encodes the optimal solution to the problem. If the ground state of H_P is NP-complete to find (for instance consider the case for Ising spin glass[49]), then the adiabatic evolution $H(s)$ could be used as a heuristic for solving the problem.

An important issue associated with AQC is that the adiabatic evolution needs to be slow enough to avoid exciting the system out of its ground state at any point. In order to estimate the scaling of the minimum runtime T needed for the adiabatic computation, criteria based on the minimum gap between the ground state and the first excited state of $H(s)$ is often used. However, here we do not use the minimum gap as an intermediate for estimating the runtime scaling, but instead numerically integrate the time dependent Schrödinger equation (1).

Quantum Ising model with transverse field. The Hamiltonian for an Ising spin glass on N spins can be written as

$$H = \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i<j}^{N} J_{ij} \sigma_i^z \sigma_j^z \tag{2}$$

where $\sigma_i^z = \mathbb{I}^{\otimes(i-1)} \otimes \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \otimes \mathbb{I}^{\otimes(n-i)}$ acts on the i-th spin with $\mathbb{I}$ being a 2×2 identity matrix. h_i, J_{ij} are coefficients. The Hamiltonian is diagonal in the basis $\{|s\rangle \in \mathbb{C}^{2^N} | \mathbf{s} \in \{0, 1\}^N\}$ in the Hilbert space $\mathscr{H}$. In particular $\sigma^z|0\rangle = |0\rangle$ and $\sigma^z|1\rangle = -|1\rangle$. We formally define the problem of finding the ground state of an N-qubit Ising Hamiltonian in the following.

Definition 2. (*Ising Hamiltonian*) *Given the Hamiltonian H in equation (2), find a quantum state $|\mathbf{s}\rangle \in \mathscr{H}$, where $\mathscr{H}$ is 2^N-dimensional, such that the energy $E(\mathbf{s}) = \langle \mathbf{s}|H|\mathbf{s}\rangle$ is minimized. We use the notation* ISING $(\mathbf{h}, \mathbf{J})$ *to refer to the problem instance where $\mathbf{h} = (h_1, h_2, \cdots, h_N)^T$ and $\mathbf{J} \in \mathbb{R}^{N \times N}$ is a matrix such that the ij-th and the ji-th elements are equal to $J_{ij}/2$. The diagonal elements of $\mathbf{J}$ are 0. Hence $E(\mathbf{s}) = \mathbf{h}^T\mathbf{p}(\mathbf{s}) + \mathbf{p}(\mathbf{s})^T\mathbf{J}\mathbf{p}(\mathbf{s})$ where $\mathbf{p}(\mathbf{s}) = 1 - 2\mathbf{s} \in \{-1, 1\}^N$.*

In this paper, we construct Ising Hamiltonians whose ground state encodes the solution to an arbitrary instance of the SCP problem. The physical system used for quantum annealing that we assume is identical to that of D-Wave[6,7,9,14], namely Ising spin glass with transverse field

$$H = \sum_{i=1}^{N} \Delta_i \sigma_i^x + \sum_{i=1}^{N} h_i \sigma_i^z + \sum_{i<j}^{N} J_{ij} \sigma_i^z \sigma_j^z \tag{3}$$

where $\sigma_i^x = \mathbb{I}^{\otimes(i-1)} \otimes \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \otimes \mathbb{I}^{\otimes(n-i)}$ acts on the i-th spin. The beginning Hamiltonian H_B has its h_i, $j_{ij} = 0$ for all i, j and the final Hamiltonian H_P has $\Delta_i = 0$ for all i while h_i and J_{ij} depend on the problem instance at hand. We will elaborate on assigning h_i and J_{ij} coefficients in H_P in Theorem 1.

Graph minor embedding. The interactions described by the transverse Ising Hamiltonian in equation (3) are not restricted by any constrains. However, in practice the topology of interactions is always constrained to the connectivity that the hardware permits. Therefore in order to physically implement an arbitrary transverse Ising Hamiltonian, one must address the problem of embedding the Hamiltonian into the logical fabric of the hardware[61,62]. For convenience we define the *interaction graph* of an Ising Hamiltonian H of the form in equation (2) as a graph $G_H(V_H, E_H)$ such that each spin i maps to a distinctive element v_i in V_H and there is an edge between v_i and v_j iff $J_{ij} \neq 0$. This definition also applies to the transverse Ising system described in equation (3). We use the term *hardware graph* to refer to a graph whose vertices represent the qubits in the hardware and the edges describe the allowed set of couplings in the hardware.

In Section Set Cover with Pairs we defined bipartite graphs. Here we define a *complete bipartite graph* $K_{m,n}$ as a bipartite graph where $|V_1| = m$, $|V_2| = n$ and each vertex in V_1 is connected with each vertex in V_2. A graph $H(W, F)$ is a *subgraph* of $G(V, E)$ if $W \subseteq V$ and $F \subseteq E$. It is possible that the interaction graph of the desired Ising Hamiltonian is a subgraph of the hardware connectivity graph. In this case the embedding problem can be solved by *subgraph embedding*, which we define as the following.

Definition 3. *A subgraph embedding of $G(V, E)$ into $G'(V', E')$ is a mapping $f: V \mapsto V'$ such that each vertex in V is mapped to a unique vertex in V' and if $(u, v) \in E$ then $(f(u), f(v)) \in E'$.*

In more general cases, for an arbitrary Ising Hamiltonian, a subgraph embedding may not be obtainable and we will need to embed the interaction graph into the hardware as a graph *minor*. Before we define minor embedding rigorously, recall that a graph is *connected* if for any pair of vertices u and v there is a path from u to v. A *tree* is a connected graph which does not contain any simple cycles as subgraphs. T is a *subtree* of G if T is a subgraph of G and T is a tree. We then define minor embedding as the following.

Definition 4. *A minor embedding of $G(V, E)$ in $G'(V', E')$ is defined by a mapping $\phi: V \mapsto V'$ such that each vertex $v \in V$ is mapped to a connected subtree T_v of G' and if $(u, v) \in E$ then there exist $i_u, i_v \in V'$ such that $i_u \in T_u$, $i_v \in T_v$ and $(i_u, i_v) \in E'$.*

If such a mapping ϕ exists between G and G', we say G is a *minor* of G' and we use $G \leq_m G'$ to denote such relationship. Our goal is to take the interaction graph G_H of our Ising Hamiltonian construction and construct the mapping ϕ that embeds G_H into the hardware graph as a minor.
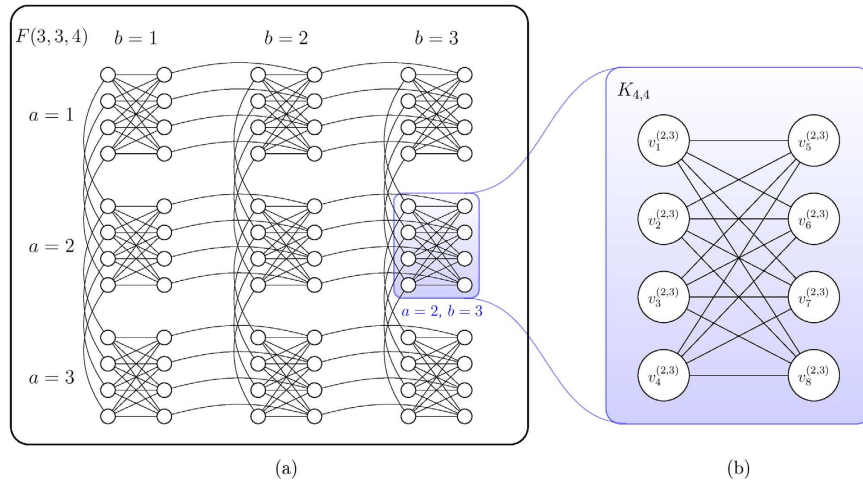
Figure 1. The Chimera graph that represents the qubit connectivity of D-Wave hardware. (**a**) Example of a 3×3 grid of $K_{4,4}$ cells, denoted as $F(3, 3, 4)$. (**b**) Labelling of nodes within a particular cell on the a-th row and b-th column. Here we use the cell on the 2nd row and 3rd column as an example.

Chimera graphs.   Here we specifically consider the embedding our construction into a particular type of hardware graphs used by D-Wave devices[44,65] called the *Chimera graphs*. The basic components of this graph are 8-spin unit cells[6] whose interactions form a $K_{4,4}$. The $K_{4,4}$ unit cells are tiled together and the 4 nodes on the left half of $K_{4,4}$ are connected to their counterparts in the cells above and below. The 4 nodes on the right half of $K_{4,4}$ are connected to their counterparts in the cells left and right. Furthermore, we define $F(p, q, c)$ as a Chimera graph formed by an $p \times q$ grid of $K_{c,c}$ cells. Figure 1a shows $F(3, 4)$ as an example. Note that any $K_{m,n}$ with $m, n \leq c$ can be trivially embedded in $F(p, q, c)$ with any $p, q \geq 1$ via subgraph embedding. However, it is not clear *a priori* how to embed $K_{m,n}$ with $m > c$ or $n > c$ onto a Chimera graph, other than using the general embedding of an $(m + n)$-node complete graph and consider $K_{m,n}$ as a subgraph. This costs $O((m + n)^2)$ qubits in general and one may lose the intuitive structure of a bipartite graph in the embedding. One of the building blocks of our embedding for our Ising Hamiltonian construction (Section Embedding on quantum hardware) is an alternative embedding strategy for mapping any $K_{m,n}$ onto $F(\lceil n/c \rceil, \lceil m/c \rceil, c)$ as a graph minor. Our embedding costs $O(mn)$ qubits and preserves the structure of the bipartite graph.

Quantum annealing for solving SCP
From an arbitrary SCP instance to an Ising Hamiltonian construction.
SCP is NP-complete most simply because Set Cover (SC) is a special case of SCP[56] and a solution to SCP is clearly efficiently verifiable. Since SC is NP-complete itself, any SCP instance can be rewritten as an instance of SC with polynomial overhead. The Ising Hamiltonian construction for Set Cover is explicitly known[39,50]. Hence it is natural to consider using the chain of reductions from SCP to SC and then from SC to ISING (Definition 2). If we recast each SCP(G, U, S) with $|S| = m$ into an SC instance with a cover set of size $O(m^2)$. Using the construction by Lucas[50] we have an Ising Hamiltonian

$$H = H_A + H_B = A \sum_{\alpha=1}^{n} \left(1 - \sum_{i:\alpha \in V_i} x_i \right)^2 + B \sum_{i=1}^{N} x_i \tag{4}$$

where V_i is the i-th cover set in the SC instance. Since the cover set $\{V_i\}$ is possibly of size up to $O(m^2)$, this leads to the Ising Hamiltonian in equation (4) costing $O(nm^2)$ qubits.

Here we present an alternative Ising Hamiltonian construction for encoding the solution to any SCP instance. We state the result precisely as Theorem 1 below. The qubit cost of our construction is comparable to that of Lucas. However, in Section Embedding on quantum hardware we argue that our construction affords more advantages in terms of embedding.

Theorem 1. *Given an instance of the Set Cover with Pairs Problem* SCP(G, U, S) *as in Definition* 1, *there exists an efficient* (*classical*) *algorithm that computes an instance of the Ising Hamiltonian ground state problem* ISING$(\mathbf{h}, \mathbf{J})$ *with* $\mathbf{h} \in \mathbb{R}^M$ *and* $\mathbf{J} \in \mathbb{R}^{M \times M}$ *where the number of qubits involved in the Hamiltonian is* $M = O(nm^2)$ *with* $n = |U|$ *and* $m = |S|$.

Proof. First, we recast an SCP instance to an instance of integer programming, which is NP-hard in the worst case. Then, we convert the integer programming problem to an instance of the ISING problem. Recall Definition 1 of an SCP(G, U, S) instance, where $G(V, E)$ is a graph on the vertices $V = U \cup S$. For each pair $f_i, f_j \in S$ define a set $Q_{ij} = \{c_k \in U | (f_i, c_k) \in E$ and $(f_j, c_k) \in E\}$. The problem can be recast as an integer program by

4

$$\min \quad \sum_{f_i \in S} s_i \tag{LP}$$

$$\text{s. t.} \quad \sum_{c_k \in Q_{ij}} t_{ij} \geq 1 \quad \forall \, c_k \in U \tag{LP.1}$$

$$t_{ij} \leq s_i \text{ and } t_{ij} \leq s_j \;\; \forall f_i \neq f_j, \text{ where } f_i, f_j \in S \text{ and } i < j \tag{LP.2}$$

$$s_i, t_{ij} \in \{0, 1\} \;\; \forall f_i \neq f_j, \text{ where } f_i, f_j \in S \tag{LP.3}$$

We have introduced the variable s_i to indicate whether f_i is chosen for the cover $A \subseteq S$ ($s_i = 1$ means that f_i is chosen, otherwise $s_i = 0$). We have also introduced the auxiliary variable t_{ij} to indicate whether f_i and f_j are *both* chosen. Hence, constraint LP.1 ensures that each element $c_k \in U$ is covered by at least one pair in S. LP.2 ensures that a pair of elements in S cannot cover any $c_k \in U$ unless both elements are chosen.

To convert the integer program to an ISING instance, we first convert the constraints into expressions of logical operations. LP.1 can be rewritten as

$$\bigvee_{c_k \in Q_{ij}} t_{ij}^{(k)} = 1, \quad \forall \, c_k \in U \tag{5}$$

LP.2 can be translated to a truth table for the binary operation involving t_{ij} and $s_i (s_j)$ where only the entry $\{s_i = 0, t_{ij} = 1\}(s_j = 0, t_{ij} = 1)$ evaluates to 0 and the other three entries evaluate to 1. Using the following Hamiltonians we could translate the logic operations $\vee$, $\wedge$ and $\leq$ into the ground states of Ising model, see ref. 66 for more details.

$$\begin{aligned}
H_\vee(s_1, s_2, s_*) &= \frac{1}{4}(3\mathbb{I} - \sigma_1^z - \sigma_2^z + 2\sigma_*^z + \sigma_1^z \sigma_2^z - 2\sigma_1^z \sigma_*^z - 2\sigma_2^z \sigma_*^z) \\
H_\wedge(s_1, s_2, s_*) &= \frac{1}{4}(4\mathbb{I} + \sigma_1^z + \sigma_2^z - 2\sigma_*^z + 2\sigma_1^z \sigma_2^z - 3\sigma_1^z \sigma_*^z - 3\sigma_2^z \sigma_*^z) \\
H_\leq(s_1, s_2) &= \frac{1}{4}(\mathbb{I} - \sigma_1^z + \sigma_2^z - \sigma_1^z \sigma_2^z).
\end{aligned} \tag{6}$$

Note that $H_\leq(s_1, s_2)$ is essentially $|10\rangle\langle10|_{s_1 s_2}$. In other words we are penalizing the only 2-bit string $s_1 s_2$ that violates the constraint $s_1 \leq s_2$. The ground state subspace of $H_\vee$ is spanned by $\{|s_1 s_2 s_*\rangle \,|\, s_1 \vee s_2 = s_*, s_1, s_2, s_* \in \{0, 1\}\}$. Similarly, the ground state subspace of $H_\wedge$ is spanned by $\{|s_1 s_2 s_*\rangle \,|\, s_1 \wedge s_2 = s_*\}$ and that of $H_\leq$ spanned by $\{|s_1 s_2\rangle \,|\, s_1 \leq s_2\}$.

By linearly combining the above constraint Hamiltonians, we can enforce multiple constraints to hold at the same time. For example, the statement $s_1 \vee s_2 \wedge s_3 = 1$ can be decomposed as simultaneously ensuring $s_1 \vee s_2 = y$, $y \wedge s_3 = z$, and $z = 1$. In other words we have used auxiliary variables y and z to transform the constraint $s_1 \vee s_2 \wedge s_3 = 1$, which involves a clause $s_1 \vee s_2 \wedge s_3$ of three variables, to a set of constraints involving only clauses of two variables. Then, the Ising Hamiltonian $H = H_\vee(s_1, s_2, y) + H_\wedge(y, s_3, z) + |0\rangle\langle0|_z$ has its ground state spanned by states $|s_1 s_2 s_3 y z\rangle$ with $s_1, s_2,$ and s_3 satisfying $s_1 \vee s_2 \wedge s_3 = 1$. The third term in H ensures that $z = 1$ by penalizing states with $|z\rangle = |1\rangle$.

Therefore, we can translate (5) to an Ising Hamiltonian. For a fixed k, the constraint (5) takes the form of $t_1^{(k)} \vee t_2^{(k)} \vee \cdots \vee t_{N_k}^{(k)} = 1$ where each $t_j^{(k)} \in \{0, 1\}$ and $N_k \leq \frac{1}{2}m(m - 1) = O(m^2)$. Similarly to the example above, we introduce $N_k - 1$ auxiliary variables $x_1^{(k)}, x_2^{(k)}, \cdots, x_{N_k - 1}^{(k)}$ such that

$$x_j^{(k)} = \begin{cases} t_1^{(k)} \vee t_2^{(k)} & j = 1 \\ x_{j-1}^{(k)} \vee t_{j+1}^{(k)} & j = 2, \cdots, N_k - 1 \end{cases} \tag{7}$$

Thus, $x_{N_k-1}^{(k)} = t_1^{(k)} \vee t_2^{(k)} \vee \cdots \vee t_{N_k}^{(k)}$. In order to ensure the first constrain holds, it is needed to ensure that $x_{N_k-1}^{(k)} = 1$. Then we could write down the corresponding Ising Hamiltonian for the constraint as

$$H_k = H_\vee(t_1^{(k)}, t_2^{(k)}, x_1^{(k)}) + \sum_{j=2}^{N_k-1} H_\vee(x_{j-1}^{(k)}, t_{j+1}^{(k)}, x_j^{(k)}) + |0\rangle\langle0|_{x_{N_k-1}^{(k)}}. \tag{8}$$

The last term is meant to make sure that $x_{N_k-1}^{(k)} = 1$ in the ground state of H_k. Therefore the Hamiltonian whose ground state subspace is spanned by all states that obey both of the constraints in the integer program (5) can be written as
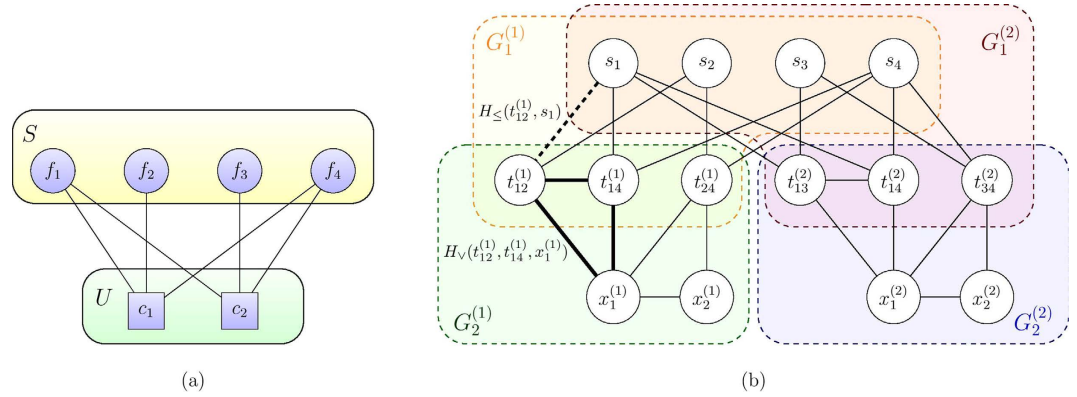
Figure 2. Example of converting an SCP instance to Ising Hamiltonian. (**a**) The SCP instance. Here $S = \{f_1, f_2, f_3, f_4\}$ and $U = \{c_1, c_2\}$. The solution is the set $A = \{f_1, f_4\}$. The circles represent the covering set elements S and the squares are the ground elements U. (**b**) The interaction of Ising instance H_{SCP} converted from the SCP instance in (**a**). Every node corresponds to a qubit. The s_i's are the output bits that correspond to the covering set elements S. The others are auxiliary variables. Every edge represents an interaction term between the corresponding spins. Here we do not show the 1-local terms in our construction of H_{SCP} (for example the terms in H_{targ} for enforcing the minimization of the target function). The bold dashed black line exemplifies the edges between the $t_{ij}^{(k)}$ nodes and the s_i nodes, which come from the constraints $t_{ij}^{(k)} \leq s_i$ and $t_{ij}^{(k)} \leq s_j$ for each pair $\{f_i, f_j\}$ that covers c_k. Each of the inequality constraints is enforced by a $H_{\leq}$ term in (6). The bold triangle exemplifies the $H_{\vee}$ constraints in (6) that are used to enforce the logical relationship between the $t_{ij}^{(k)}$ variables and the auxiliary variables as shown in (7). The areas marked by $G_1^{(1)}, G_1^{(2)}$ etc outline the structure of the Ising Hamiltonian that is relevant in the discussion of hardware embedding.

$$
\begin{aligned}
H_{\text{cons}} &= \sum_{c_k \in U} H_k + H_{\leq} \\
H_{\leq} &= \sum_{i,j: f_i, f_j \in S} (H_{\leq}(t_{ij}, s_i) + H_{\leq}(t_{ij}, s_j)).
\end{aligned}
\tag{9}
$$

The target function $\sum_{f_i \in S} s_i$ which we seek to minimize can be directly mapped to an Ising Hamiltonian $H_{\text{targ}} = \sum_{f_i \in S} |1\rangle\langle 1|_{s_i} = \sum_{f_i \in S} \frac{1}{2}(1 - \sigma_{s_i})$. This is because we would like to essentially minimize the number of 1's in the set of s_i values and penalize choices with more 1's. Therefore the final Hamiltonian whose ground state contains the solution to the original SCP instance becomes

$$
H_{\text{SCP}} = \alpha H_{\text{targ}} + H_{\text{cons}}
\tag{10}
$$

for some weight factor α.

We now estimate the overhead for the mapping. H_{targ} acts on $|S| = m$ qubits. In H_{cons}, $H_{\leq}$ acts on $O(m^2)$ qubits, since there are $O(m^2)$ variables t_{ij}. Each H_k in H_{cons} requires $N_k = O(m^2)$ qubits. There are in total $|U| = n$ of the H_k terms, which gives $O(nm^2)$ qubits in total. $\square$

Example. Consider the SCP instance shown in Fig. 2a. With the mapping presented in Theorem 1, we arrive at an Ising instance ISING (**h, J**) where $\alpha = 1/4$ in (10) and **h, J** are presented in Supplementary Material Details of the example SCP instance. The ground state subspace of the Hamiltonian in (2) with h_i and J_{ij} coefficients defined above, restricted to the s_i elements is spanned by $[|\psi\rangle = |s_1 s_2 \cdots x_2^{(2)}\rangle$ such that $|s_1 s_2 s_3 s_4\rangle = |1001\rangle]$. This corresponds to $A = \{f_1, f_4\}$, the solution to the SCP instance. Figure 2b illustrates the interaction graph of the spins in the Ising Hamiltonian that corresponds to the SCP instance.

Numerical simulation of quantum annealing. In order to test the time complexity of using quantum annealing to solve SCP instances via the construction in Theorem 1, we generate random instances of SCP that lead to Ising Hamiltonian H_{SCP} of $N = 3, 4, \cdots, 19$ spins. In Definition 1 we use a bipartite graph between the ground state U of size n and the cover set S of size m to describe an SCP instance. For fixed n and m, there are in total 2^{mn} such possible bipartite graphs (if we consider each bipartite graph as a subgraph of $K_{m,n}$ and count the cardinality of the power set of the edges of $K_{m,n}$. Therefore to generate random bipartite graphs we only need to flip mn fair coins to uniformly choose from all possibile bipartite graphs between U and S. However, we would like to exclude the bipartite graphs where some element of S is not connected to any element in U. These "dummy nodes" are not pertinent to the computational problem at hand and should be removed from consideration before converting the SCP instance to an Ising Hamiltonian H_{SCP}. We thus use a scheme for generating random instances of SCP *without* dummy nodes as described in Algorithm 1. Under the constraint that no dummy element in S is allowed, there are in total $(2^n - 1)^m$ possible bipartite graphs. In Supplementary Material Proof of correctness for

Algorithm 1 we rigorously show that Algorithm 1 indeed samples uniformly among the $(2^n - 1)^m$ possible "dummy-free" bipartite graphs.

Algorithm 1 Algorithm for generating a random SCP (G, U, S) without dummy elements in the cover set

Input: The ground set U and the cover set S

Procedure:

1:	Initialize the output graph $G \leftarrow \varnothing$;
2:	**for all** $s \in S$ **do**
3:	**for all** $u \in U$ **do**
4:	With probability 1/2, add edge (s, u) to G;
5:	**end for**
6:	**if** s is still unattached to any element in U **then**
7:	Repeat steps 3 through 5 until s is attached to some element in U.
8:	**end if**
9:	**end for**
10:	**return** G.

For each randomly generated instance from Algorithm 1 we construct an Ising Hamiltonian H_{SCP} according to Theorem 1. We then perform a numerical simulation of the time dependent Schrödinger equation (1) from time $t = 0$ to $t = T$ with time step $\Delta t = 1$ and the time dependent Hamiltonian defined as

$$
\begin{aligned}
H(s(t)) &= \left(1 - \frac{t}{T}\right)H_B + \frac{t}{T}H_{SCP} \\
H_B &= \sum_{i=1}^{M}\sigma_i^x
\end{aligned}
\tag{11}
$$

where H_{SCP} is defined in equation (10). Here because of the construction of H_{SCP}, our total Hamiltonian $H(s(t))$ acts not only on the spins $\mathbf{s} \in \{0, 1\}^m$ indicating our choice of elements in the cover set S, but also auxiliary variables $t_{ij}^{(k)}$ and $x_i^{(k)}$, for which we use $\mathbf{t}$ and $\mathbf{x}$ to denote their respective collections. Our initial state is the ground state of H_B, namely

$$
|\psi(0)\rangle = \frac{1}{2^M}\sum_{\mathbf{s},\mathbf{t},\mathbf{x}\in\{0,1\}^M}|\mathbf{s}, \mathbf{t}, \mathbf{x}\rangle.
\tag{12}
$$

To obtain the final state $|\psi(T)\rangle$ where T is some positive integer, we use the ode45 subroutine of MATLAB under default settings to numerically integrate Schrödinger equation to obtain $|\psi(1)\rangle$ from $|\psi(0)\rangle$, and then use $|\psi(1)\rangle$ as an initial state to obtain $|\psi(2)\rangle$ in the same fashion, and so on. We define the success probability p as a function of the total annealing time T as $p(T) = \|\Pi|\psi(T)\rangle\|_2$ where Π is a projector onto the subspace spanned by states with s being a solution of the original SCP instance. Using binary search we determine the minimum time $T^\star$ to achieve $p(T^*) \geq 0.25$ for each instance of SCP. Figure 3 shows the distribution of $T^\star$ for SCP instances that lead to Ising Hamiltonians H_{SCP} of the same sizes, as well as how the median annealing time scales as a function of number of spins M. Results show that for instances with M up to 19, the median annealing time scales roughly as $O(2^{0.31M})$.

Numerical experiment with Simulated Annealing. Simulated annealing, first introduced three decades ago[67], has been widely used as a heuristic for handling hard combinatorial optimization problems. It is especially of interest as a benchmark for quantum annealing[34–36] because of similarities between the two algorithms. While quantum annealing employs quantum tunneling to escape from local minima, simulated annealing relies on thermal excitation to avoid being trapped in local minima. The general procedure we adopt for simulated annealing to approach the ground state of an Ising spin glass can be summarized as the following[68]:

1. Repeat R times the following:

   (a) Initialize $\mathbf{s} \leftarrow s_0$ randomly;
   (b) Perform S times the following: (let $i = 0, 1, \cdots, S - 1$ index the steps)

      (i) Set the temperature $T_i \leftarrow \tau(i)$;
      (ii) Perform a *sweep* on $\mathbf{s}_i$ to obtain $\mathbf{s}'$; (a sweep is a sequence of steps each of which randomly selects a spin and flips its state, so that on average each spin is flipped once during a sweep)
      (iii) With probability $\exp\left(\frac{E(\mathbf{s}') - E(\mathbf{s})}{T_i}\right)$, let $\mathbf{s}_{i+1} = \mathbf{s}'$. Otherwise let $\mathbf{s}_{i+1} \leftarrow \mathbf{s}_i$.
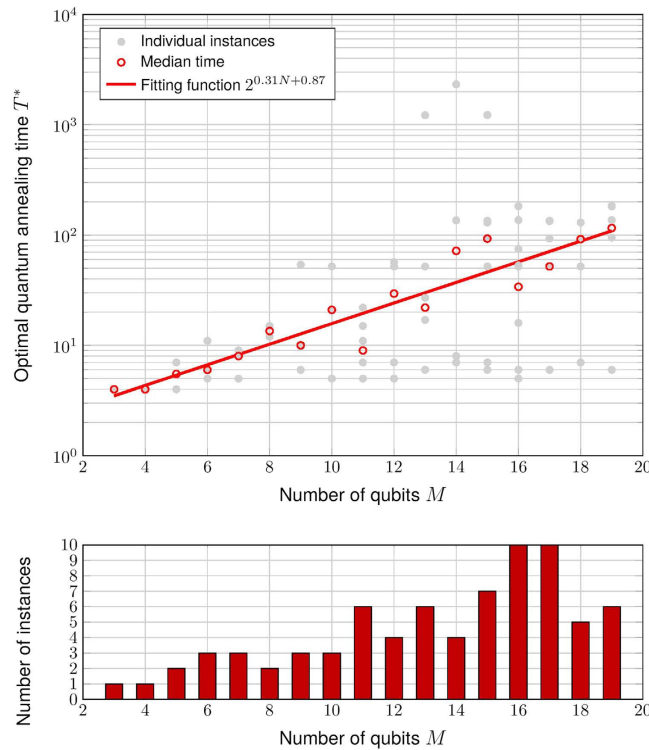
2. Return $\mathbf{s}_S$ as the answer.

Figure 3. Plot of the optimal quantum annealing time *T** versus the number of spins involved in the construction of H_{SCP}. Here we fit the logarithm of median T^* with a straight line. The size M of our Ising systems ranges from 3 to 19. From the fitting function we observe that the annealing time scales as roughly $O(2^{0.31M})$. We also provide on the bottom plot the number of instances for each M.

For the purpose of comparison we also used simulated annealing to solve the same set of instances generated by Algorithm 1 for testing quantum annealing. The program implementation that we use is built by Isakov *et al.*[68], which is a highly optimized implementation of simulated annealing with care taken to exploit the structures of the interaction graph, such as being bipartite and of bounded degree. Here we use the program's most basic realization of single-spin code for general interactions with magnetic field on an interaction graph of any degree.

As mentioned by Isakov *et al.*, to improve the solution returned by simulated annealing, one could increase either the number of sweeps S or number of repetitions R in the implementation, or both of them. However, note that the total annealing time is proportional to the product $S \cdot R$ and there is a trade-off between S and R. For a fixed number of sweeps S let the success probability (*i.e.* the fraction of $\mathbf{s}_i$ that is satisfactory) be $w(S)$. In order to achieve a constant success probability p (say 25%, which is what we use here), we need at least $R = \lceil \log(1-p)/\log(1-w(S)) \rceil$ repetitions. Hence the total time of simulated annealing can be written as

$$T(S) = \left\lceil \frac{\log(1-p)}{\log(1-w(S))} \right\rceil \cdot S. \tag{13}$$

In general $w(S)$ increases as S increases, leading to a decrease in R. We numerically investigate this with an Ising system of $N = 17$ spins generated from an SCP instance via the construction in Theorem 1. We plot the annealing time T versus S in Fig. 4a. For each SCP instance with the number of spin M we compute the optimal S^* such that $T^* = T(S^*)$ is the optimized runtime (Fig. 4a). We further explore how the optimal runtime T^* scales as a function of the number of spins M. As shown in Fig. 4b, a linear fit on a semilog plot shows that roughly $T^* = O(2^{0.21M})$.

The units of time used for both Fig. 4a,b are arbitrary and thus do not support a point-to-point comparison. But the scaling difference seems apparent. For quantum annealing we restrict to systems of at most 19 spins due to computational limitations faced in representing the full Ising Hamiltonian when numerically integrating the time-dependent Schrödinger equation (1).

Although there is no quantum speedup observed in terms of median runtime over all randomly generated instances of the same size, we notice that for a fixed number of spins M the performances of both quantum annealing and simulated annealing are sensitive to the specific instance of Ising Hamiltonian H_{SCP} than simulated annealing. This can be seen by considering at the same time the quantum annealing results in Fig. 3 and the test results for simulated annealing shown in Fig. 4b. One could then speculate that perhaps by focusing on a specific subset of SCP instances could yield a quantum advantage.
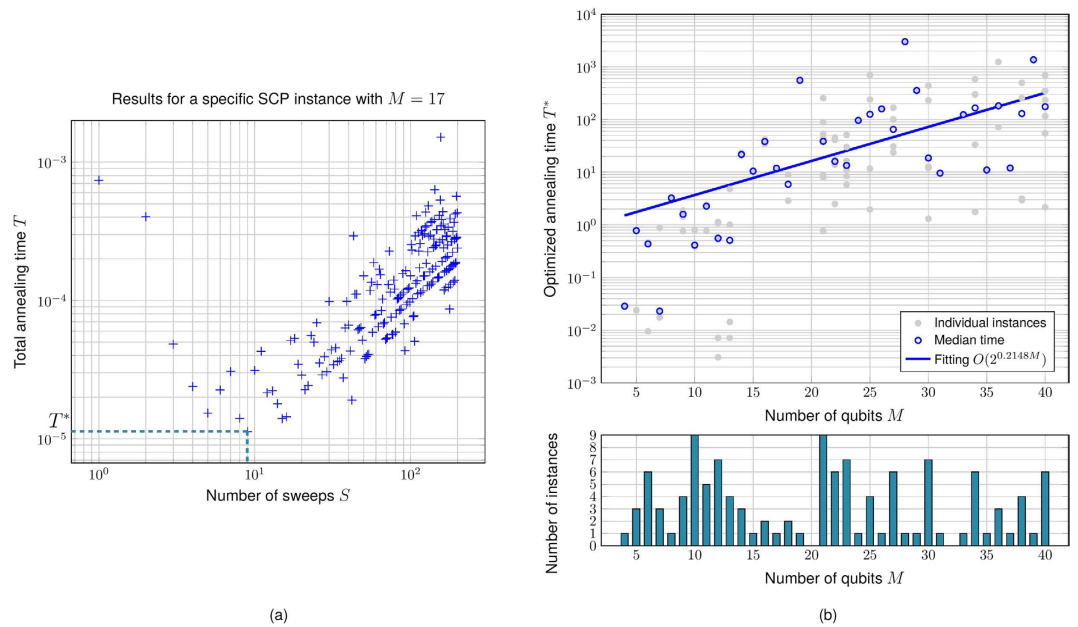
Figure 4. (**a**) Plot of annealing time T versus number of sweeps S using the simulated annealing implementation[68] on an Ising Hamiltonians of 17 spins constructed from an SCP instance. We use the default settings for all parameters other than S and R. Also we mark the optimal runtime T_*. (**b**) Plot of optimized annealing time T^* versus the number of spins involved in the Ising Hamiltonian HSCP corresponding to randomly generated SCP instances according to Algorithm 1. We also provide on the bottom plot the number of instances for each M.

Embedding on quantum hardware

In this section we deal with the physical realization of quantum annealing for solving SCP instances using D-Wave type hardwares. There are mainly two aspects[62,69] of this effort: 1) The *embedding problem*[62], namely embedding the interaction graph of the Ising Hamiltonian construction H_{SCP} as a graph minor of a Chimera graph (refer to Section Graph minor embedding for definitions of the graph terminologies). 2) The *parameter setting problem*[69], namely assigning the strengths of the couplings and local magnetic fields for embedded graph on the hardware, in a way that minimizes the energy scaling (or control precision) required for implementing the embedding. Here we focus on the former issue.

We start with an observation on the structures of H_{SCP}. For any instance SCP(G, U, S) according to Definition 1, the interaction graph $I_{SCP(G,U,S)}$ of the corresponding Ising Hamiltonian H_{SCP} can be regarded as a union of n sub-graphs, namely $I_{SCP(G,U,S)} = G^{(1)} \cup G^{(2)} \cup \cdots \cup G^{(n)}$. Each subgraph $G^{(i)}$ is associated with an element of the ground set $c_i \in U$ as in Fig. 2a. Each $G^{(i)}$ could be further partitioned into two parts, $G_1^{(i)}$ and $G_2^{(i)}$. For any k, $G_1^{(k)}$ is a bipartite graph between $\{s_i\}_{i=1}^m$ and $\{t_{ij}^{(k)}|f_i, f_j \in S$ cover $c_k \in U\}$. $G_2^{(k)}$ essentially describes the interaction between the auxiliary variables $t_{ij}^{(k)}$ and $x_i^{(k)}$ as described in equation (7). In Fig. 2b we illustrate such partition using the example from Fig. 2b. Our goal is then to show constructively that $I_{SCP}(G,U,S) \leq_m F(f_1, f_2, c)$ for some f_1, f_2 that depend on m, n and $c = 4$, which describes the Chimera graph realized by D-Wave hardware (Fig. 1a).

It is known[61] that one could embed a complete graph on $cm + 1$ nodes onto Chimera graph $F(m, m, c)$. Since any n-node graph is a subgraph of the n-node complete graph, in principle any n-node graph can be embedded onto Chimera graphs of size $O(n^2)$ using the complete graph embedding. A downside of this approach is that it may fail to embed many graphs that are in fact embeddable[61]. Also, using embeddings based on complete graph embeddings will likely lose the intuition on the structure of the original graph. For graphs with specific structures, such as bipartite graphs one may be able to find an embedding that is also in some sense structured. We show in the following Lemma an embedding for any complete bipartite graph $K_{p,q}$ onto a Chimera graph. The ability to do so enables us to embed any bipartite graph onto a Chimera graph.

Lemma 1. *For any positive integers p, q and c, $K_{p,q} \leq_m F(\lceil q/c \rceil, \lceil p/c \rceil, c)$.*

Proof. By the definition of graph minor embedding in Section Graph minor embedding, it suffices to construct a mapping $\phi_{p,q}: K_{p,q} \mapsto F(\lceil q/c \rceil, \lceil p/c \rceil, c)$ where each v in $F_{p,q}$ is mapped to a tree T_v in $F(\lceil q/c \rceil, \lceil p/c \rceil, c)$ and each edge $e = (u, v)$ in $K_{p,q}$ is mapped to an edge (i_u, i_v) with $i_u \in T_u$ and $i_v \in T_v$.

Let $i = 1, 2, \cdots, p$ label the nodes on one side of $K_{p,q}$ and $j' = 1, 2, \cdots, q$ label the nodes in the other. Using the labelling scheme on the nodes of Chimera graphs introduced in Section Chimera graphs and Fig. 1b, we define our mapping $\phi_{p,q}$ as
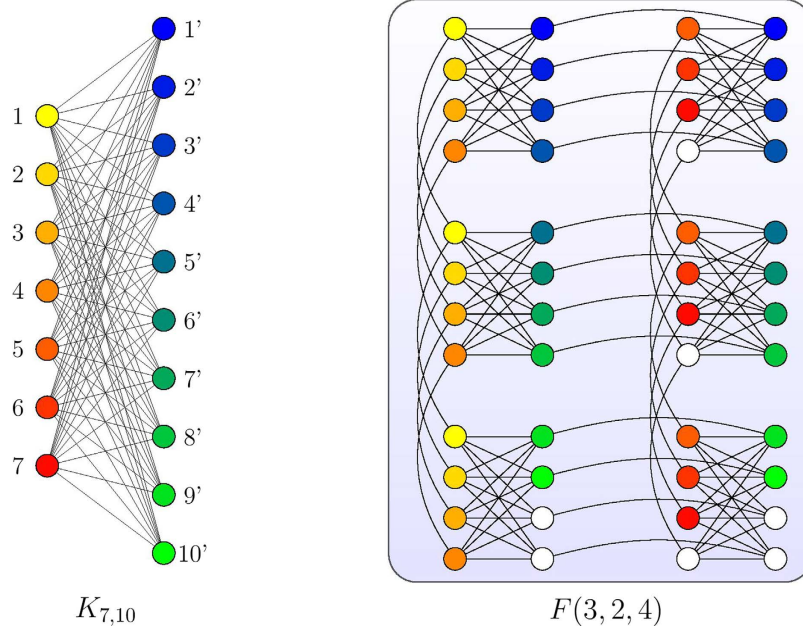
Figure 5. An example showing the embedding scheme outlined in Lemma 1. The nodes and the trees mapped from the nodes are marked with the same colors.

$$
\begin{aligned}
\phi_{p,q}(i) &= \{v_{i \bmod c}^{(t, \lceil i/c \rceil)} | t = 1, \cdots, \lceil q/c \rceil\} \\
\phi_{p,q}(j') &= \left\{ v_{c+(j' \bmod c)}^{(\lceil j'/c \rceil, t)} \middle| t = 1, \cdots, \lceil p/c \rceil \right\}. \\
\phi_{p,q}(i, j') &= \left(v_{i \bmod c}^{(\lceil i/c \rceil, \lceil j'/c \rceil)}, v_{c+(j' \bmod c)}^{(\lceil i/c \rceil, \lceil j'/c \rceil)} \right)
\end{aligned}
\tag{14}
$$

where $\phi_{p,q}(u, v)$ maps an edge (u, v) in $K_{p,q}$ to the Chimera graph. If we choose the edges in the Chimera graph properly, it could be checked that $\phi_{p,q}(K_{p,q})$ is a subgraph of $F(\lceil q/c \rceil, \lceil p/c \rceil, c)$. □

In Fig. 5 we show an example of embedding $K_{7,10}$ into $F(3, 2, 4)$. A natural corollary of Lemma 1 is that any bipartite graph between p and q nodes can be minor embedded in $F(\lceil q/c \rceil, \lceil p/c \rceil, c)$. We are then prepared to handle embedding the $G_1^{(i)}$ parts of the interaction graphs of H_{SCP}, which are but bipartite graphs (see Fig. 2b for example).

We then proceed to treat the $G_2^{(i)}$ parts of the interaction graph. The connectivity of $G_2^{(k)}$ is completely specified by (7). To describe such connectivity we define a family of graph $L_n(V_n, E_n)$ as $V_n = T_n \cup X_{n-1}$ where $T_n = \{t_1, t_2, \cdots, t_n\}$ and $X_{n-1} = \{x_1, x_2, \cdots, x_{n-1}\}$ are two disjoint sets of nodes, the former representing the intermediate variables $t_{ij}^{(k)}$ and the latter representing the x_k variables in equation (7). The set of edges takes the form

$$
E_n = \{(t_1, t_2), (t_1, x_1), (t_2, x_1)\} \cup \left(\bigcup_{i=2}^{n-1} \{(x_{i-1}, x_i), (x_{i-1}, t_{i+1}), (x_i, t_{i+1})\} \right).
\tag{15}
$$

In Fig. 6 we show an example of L_{10}. For any $k = 1, 2, \cdots, |U|$, let r_k be the number of pairs $f_i, f_j \in S$ that cover k. Then $G_2^{(k)} = L_{r_k}$. Hence in order to show that we could embed any $G_2^{(i)}$ onto a Chimera graph, it suffices to show that we can embed any L_n onto a Chimera graph. We show this in the following Lemma for $c = 4$.

Lemma 2. *For any positive integer n, $L_n \leq_m F(\lceil 2n/c \rceil, 2, c)$ where we restrict to $c = 4$.*

Proof. Similar to Lemma 1, we construct a mapping $\mu_n: L_n \mapsto F(\lceil 2n/c \rceil, 2, c)$ where we fix $c = 4$. Following the notation for nodes in L_n in Fig. 6 and the notation for nodes in $F(p, q, c)$ in Fig. 1b, we construct μ as

$$
\begin{aligned}
\mu_n(t_i) &= \left\{ v_{(2i-1) \bmod c}^{\left(\lceil \frac{2i-1}{c} \rceil, 1 \right)}, v_{c+[(2i-1) \bmod c]}^{\left(\lceil \frac{2i-1}{c} \rceil, 1 \right)} \right\} \cup \xi_t(t_i) \\
\mu_n(x_i) &= \left\{ v_{(2i) \bmod c}^{\left(\lceil \frac{2i}{c} \rceil, 1 \right)}, v_{c+[(2i) \bmod c]}^{\left(\lceil \frac{2i}{c} \rceil, 1 \right)}, v_{c+[(2i) \bmod c]}^{\left(\lceil \frac{2i}{c} \rceil, 2 \right)} \right\} \cup \xi_x(x_i)
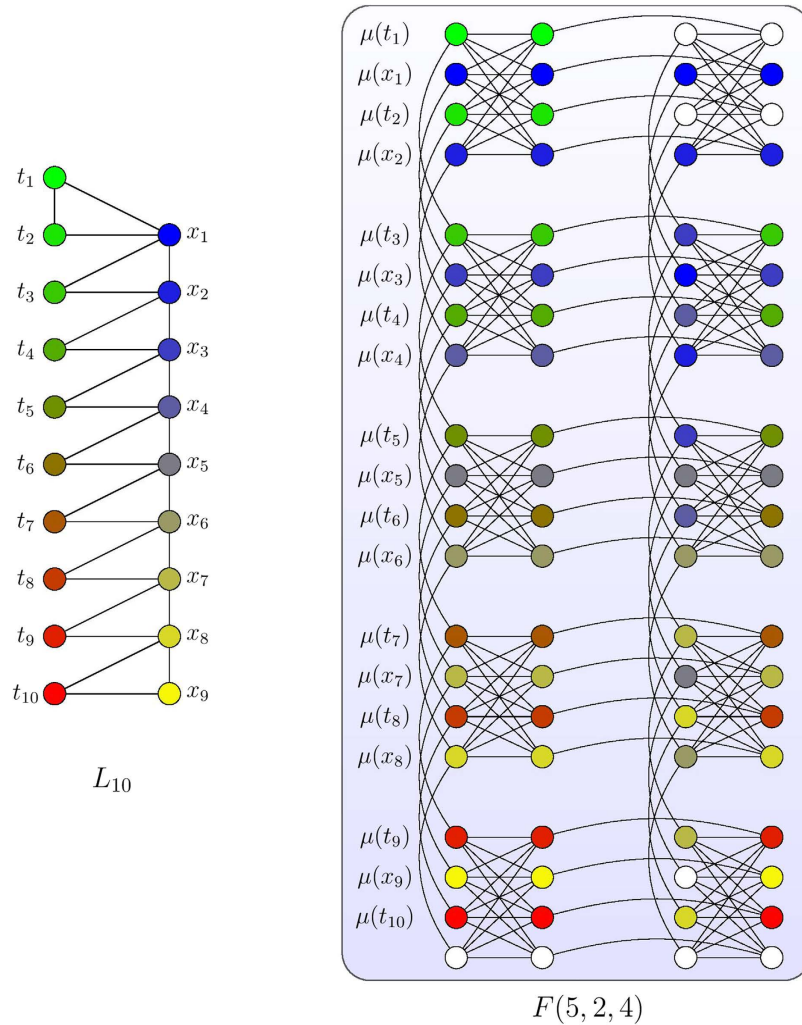\end{aligned}
\tag{16}
$$

Figure 6. An example of embedding L_{10} onto $F(5, 2, 4)$. Each color in the left diagram represents a node u in L_{10} and the nodes of the same color in the right diagram shows $\mu_{10}(u)$.

where ξ_t and ξ_x are defined as

$$\xi_t(t_i) = \begin{cases} \varnothing & \text{if } i = 1, 2 \\ \left\{ v^{\left(\left\lceil\frac{2i-1}{c}\right\rceil, 2\right)}_{c+[(2i-1) \bmod c]} \right\} & \text{otherwise.} \end{cases} \tag{17}$$

$$\xi_x(x_i) = \begin{cases} \left\{ v^{\left(\left\lceil\frac{2i}{c}\right\rceil, 2\right)}_{(2i) \bmod c}, v^{\left(\left\lceil\frac{2i}{c}\right\rceil+1, 2\right)}_{(2i) \bmod c} \right\} & \text{if } \lceil i/2 \rceil \bmod 2 = 1 \text{ and } 2i + 4 < 2n - 1 \\ \left\{ v^{\left(\left\lceil\frac{2i}{c}\right\rceil, 2\right)}_{[(2i) \bmod c]-1}, v^{\left(\left\lceil\frac{2i}{c}\right\rceil+1, 2\right)}_{[(2i) \bmod c]-1} \right\} & \text{if } \lceil i/2 \rceil \bmod 2 = 0 \text{ and } 2i + 4 < 2n - 1 \\ \varnothing & \text{otherwise.} \end{cases} \tag{18}$$

With the vertex mapping μ_n, a mapping of edges in L_n onto the Chimera graph $F(\lceil 2n/c \rceil, 2, c)$ is easy to find.

In Fig. 6 we show an example of embedding L_{10} onto $F(5, 2, 4)$. We could then proceed to embed the interaction graph $I_{SCP(G,U,S)}$, such as the one shown in Fig. 2b, in a Chimera graph. Specifically, we state the following theorem.

Theorem 2. *For any instance* SCP(G, U, S) *with* $|U| = n$ *and* $|S| = m$, $I_{SCP(G,U,S)} \leq_m F(f_1, f_2, c)$ *where* $f_1 = O(nm^2)$, $f_2 = O(m)$ *and* $c = 4$ *is a constant.*
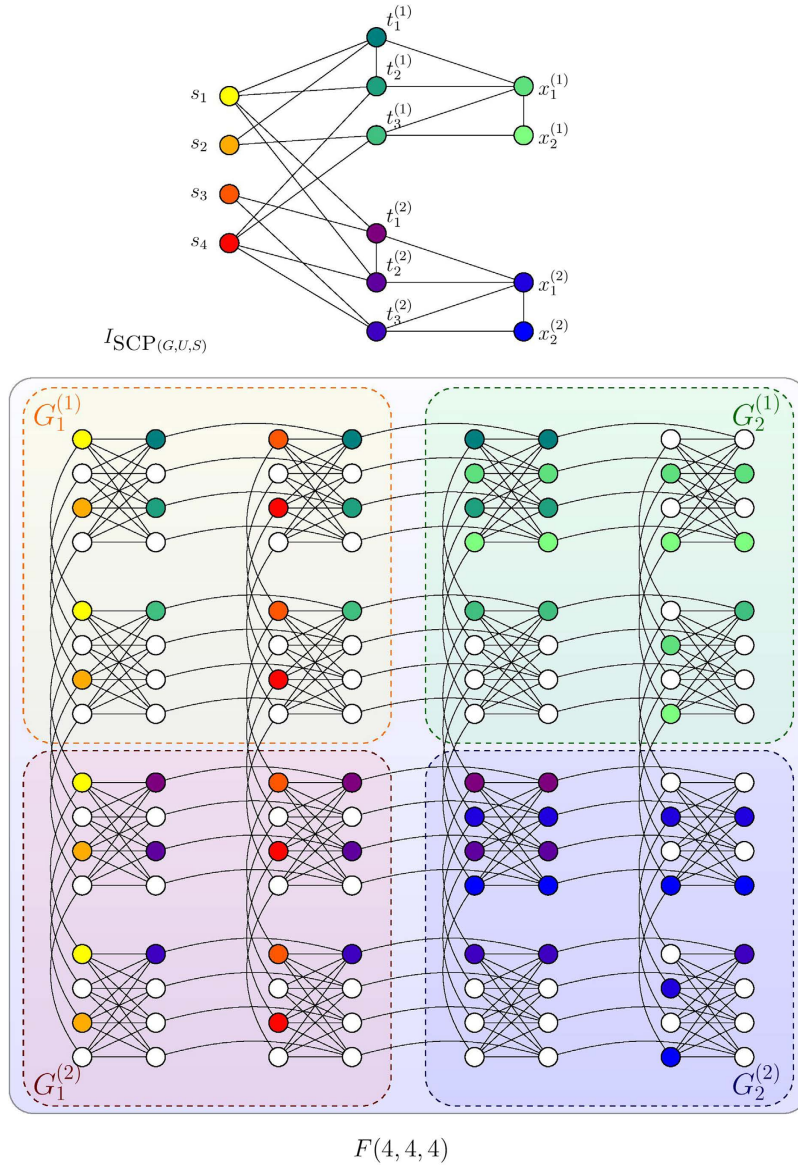
Figure 7. Embedding the interaction graph of the example physical system in Fig. 2b onto $F(4, 4, 4)$. Note that the structure of Fig. 2 is preserved on the Chimera graph.

Proof. Our embedding combines ideas from Lemma 1 and 2. We modify the mapping $\phi_{p,q}$ constructed in Lemma 1 to produce a new mapping $\theta_{p,q}$ that produces more spacing between the embedded nodes (see for example $G_1^{(1)}$ and $G_1^{(2)}$ in Fig. 7):

$$
\begin{aligned}
\theta_{p,q}(i) &= \left\{ v_{(2i-1) \bmod c}^{(t, \lceil (2i-1)/c \rceil)} \Big| t = 1, \cdots, \lceil q/c \rceil \right\} \\
\theta_{p,q}(j') &= \left\{ v_{c+(2j'-1 \bmod c)}^{(\lceil (2j'-1)/c \rceil, t)} \Big| t = 1, \cdots, \lceil p/c \rceil \right\}
\end{aligned}
\tag{19}
$$

Let $\mu_n^{(r,s)}$ denote a mapping μ described in Lemma 2 that maps the upper left node (Fig. 6) t_1 to $v_1^{(r,s)}$ instead of $v_1^{(1,1)}$. The rest of the mapping then proceeds from $v_1^{(r,s)}$. In other words, $\mu_n^{(r,s)}$ is the mapping μ that is shifted by $p-1$ cells to the right and $q-1$ cells below. Trivially $\mu_n^{(1,1)} = \mu$. Similarly we define $\theta_{p,q}^{(r,s)}$ as the shifted embedding under $\theta_{p,q}$ where $\theta_{p,q}^{(r,s)}(1) = v_1^{(r,s)}$. Recall that for any ground set element $c_k \in U$, r_k is the number of pairs in S that covers c_k. We could then specify the embedding from $I_{\text{SCP}(G,U,S)}$ onto $F(f_1, f_2, c)$ as

$$\Phi(s_i) = \bigcup_{j=1}^{n} \theta_{m,r_i}^{(1+d_j,1)}(i), \text{ where } i = 1, \cdots, m$$

$$\Phi(t_i^{(j)}) = \theta_{m,r_i}^{(1+d_j,1)}(t_i) \bigcup \mu_{r_j}^{(1+d_j,1+\lceil 2m/c \rceil)}(t_i), \text{ where } i = 1, \cdots, r_j \text{ and } j = 1, \cdots, n$$

$$\Phi(x_i^{(j)}) = \mu_{r_j}^{(1+d_j,1+\lceil 2m/c \rceil)}(t_i), \text{ where } i = 1, \cdots, r_j - 1 \text{ and } j = 1, \cdots, n \tag{20}$$

where $d_j = \sum_{k=1}^{j-1} \lceil 2r_k/c \rceil$ is the total number of rows of cells occupied by the embedded graphs for handling the ground elements c_1 through c_{j-1}. In total $\Phi(I_{SCP(G,U,S)})$ will occupy $f_1 = \sum_{k=1}^{n} \lceil 2r_k/c \rceil \leq n \binom{m}{2} \cdot 2/c = O(nm^2)$ rows and $f_2 = \lceil 2m/c \rceil + 2 = O(m)$ columns. $\square$

In Fig. 7 we show an embedding $I_{SCP(G,U,S)}$ of the example instance in Fig. 2 onto $F(4, 4, 4)$. Note that our embedding preserves the original structure of the interaction graph as shown in Fig. 2b. Furthermore, note that the interaction graph $I_{SCP(G,U,S)}$ has $M = O(nm^2)$ nodes. Using the complete graph embedding requires $O(M^2) = O(n^2m^4)$ qubits. For the same reason, the construction of Ising Hamiltonian described in equation (4) is likely going to cost $O(nm^4)$ in the worst case of embeding in a Chimera graph since the interaction graph of the Hamiltonian could involve complete graphs of size $O(m^2)$ due to the square term H_A. By comparison our embedding costs $f_1 f_2 \cdot 2c = O(nm^3)$ qubits and preserves the structure of the original instance, which affords slightly more advantage for scalable physical implementations.

Discussion

Our interest in SCP is largely motivated by its important applications in various areas[57–60]. We have shown a complete pipeline of reductions that converts an arbitrary SCP instance to an interaction graph on a D-Wave type hardware based on Chimera graphs, in a way that preserves the structure of the instance throughout (Figs 2b and 7) and is more qubit efficient than the usual approach by complete graph embedding. Although no quantum speedup is observed at this stage based on comparison of median annealing times, the large variance of runtimes observed in Fig. 3a from instance to instance might suggest that specific subsets of instances could provide quantum speedup. Of course, a clearer understanding of the performance of quantum annealing on solving SCP could only be brought forth by both scaling up the numerical simulation of the quantum annealing process to include instances with larger number of spins and actual experimental implementation of the quantum annealing process. Both of them are of interest to us in our future work.

References

1. Finnila, A. B., Gomez, M. A., Sebenik, C., Stenson, C. & Doll, J. D. Quantum annealing: A new method for minimizing multidimensional functions. *Chemical Physics Letter* **219,** 343–348 (1994).
2. Kadowaki, T. & Nishimori, H. Quantum annealing in the transverse ising model. *Physical Review E* **58,** 5355 (1998).
3. Farhi, E. *et al.* A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science* **292,** 472–475 (2001).
4. Das, A. & Chakrabarti, B. K. *Quantum annealing and related optimization methods* vol. 679 (Springer Science & Business Media, 2005).
5. Das, A. & Chakrabarti, B. K. Quantum annealing and analog quantum computation. *Review of Modern Physics* **80,** 1061 (2008).
6. Harris, R. *et al.* Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Physical Review B* **82,** 024511 (2010).
7. Johnson, M. W. *et al.* Quantum annealing with manufactured spins. *Nature* **473,** 194–198 (2011).
8. Bapst, V., Foini, L., Krzakala, F., Semerjian, G. & Zamponi, F. The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective. *Physics Reports* **523,** 127–205 (2013).
9. Dickson, N. G. *et al.* Thermally assisted quantum annealing of a 16-qubit problem. *Nature Communications* **4,** 1903 (2013).
10. Boixo, S. *et al.* Evidence for quantum annealing with more than one hundred qubits. *Nature Physics* **10,** 218–224 (2014).
11. McGeoch, C. C. & Wang, C. Experimental evaluation of an adiabiatic quantum system for combinatorial optimization. In *Proceedings of the ACM International Conference on Computing Frontiers* (New York, USA, 2013).
12. Dash, S. A note on QUBO instances defined on Chimera graphs. ArXiv:1306.1202 [math.OC] (2013).
13. Boixo, S., Albash, T., Spedalieri, F. M., Chancellor, N. & Lidar, D. A. Experimental signature of programmable quantum annealing. *Nature Communications* **4,** 3067 (2013).
14. Lanting, T. *et al.* Entanglement in a quantum annealing processor. *Physical Review X* **4,** 021041 (2014).
15. Santra, S., Quiroz, G., Steeg, G. V. & Lidar, D. MAX 2-SAT with up to 108 qubits. *New Journal of Physics* **16,** 045006 (2014).
16. Rønnow, T. F. *et al.* Defining and detecting quantum speedup. *Science* **345,** 420 (2014).
17. Vinci, W. *et al.* Hearing the shape of the ising model with a programmable superconducting-flux annealer. *Scientific Report* **4,** 5703 (2014).
18. Shin, S. W., Smith, G., Smolin, J. A. & Vazirani, U. How "quantum" is the D-Wave machine? ArXiv:1401.7087 [quant-ph] (2014).
19. Albash, T., Vinci, W., Mishra, A., Warburton, P. A. & Lidar, D. A. Consistency tests of classical and quantum models for a quantum annealer. *Physical Review A* **91,** 042314 (2015).
20. McGeoch, C. In *Adiabatic Quantum Computation and Quantum Annealing*: Theory and Practice. 2014 (Morgan & Claypool).
21. Venturelli, D. *et al.* Quantum optimization of fully-connected spin glasses. *Physical Review X* **5,** 031040 (2015).
22. Vinci, W., Albash, T., Paz-Silva, G., Hen, I. & Lidar, D. A. Quantum annealing correction with minor embedding. *Physical Review A* **92,** 042310 (2015).
23. Albash, T., Rønnow, T. F., Troyer, M. & Lidar, D. A. Reexamining classical and quantum models for the D-Wave One processor. *The European Physical Journal Special Topics* **224,** 111 (2015).
24. King, A. D. & McGeoch, C. C. Algorithm engineering for a quantum annealing platform. ArXiv:1410.2628 [cs.DS] (2014).
25. Crowley, P. J. D., Duric, T., Vinci, W., Warburton, P. A. & Green, A. G. Quantum and classical in adiabatic computation. *Physical Review A* **90,** 042317 (2014).
26. Hen, I. *et al.* Probing for quantum speedup in spin glass problems with planted solutions. *Physical Review A* **92,** 042325 (2015).
27. Steiger, D. S., Rønnow, T. F. & Troyer, M. Heavy tails in the distribution of time-to-solution for classical and quantum annealing. *Physical Review Letters* **115,** 230501 (2015).

28. Bauer, B., Wang, L., Pižorn, I. & Troyer, M. Entanglement as a resource in adiabatic quantum optimization (2015).
29. Albash, T., Hen, I., Spedalieri, F. M. & Lidar, D. A. Reexamination of the evidence for entanglement in the d-wave processor. *Physical Review A* **92,** 062328 (2015).
30. Katzgraber, H. G., Hamze, F., Zhu, Z., Ochoa, A. J. & Munoz-Bauza, H. Seeking quantum speedup through spin glasses: The good, the bad, and the ugly. *Physical Review X* **5,** 031026 (2015).
31. Chancellor, N., Szoke, S., Vinci, W., Aeppli, G. & Warburton, P. A. Maximum-entropy inference with a programmable annealer. *Scientific Report* 22318 (2016).
32. Perdomo-Ortiz, A., O'Gorman, B., Fluegemann, J., Biswas, R. & Smelyanskiy, V. N. Determination and correction of persistent biases in quantum annealers arXiv:1503.05679 [quant–ph] (2015).
33. Vinci, W., Albash, T. & Lidar, D. A. Nested quantum annealing correction arXiv:1511.07084 [quant–ph] (2015).
34. Farhi, E., Goldstone, J. & Gutmann, S. Quantum adiabatic evolution algorithms versus simulated annealing. *MIT-CTP-3228* (2002).
35. Santoro, G. E., Martoňák, R., Tosatti, E. & Car, R. Theory of quantum annealing of an Ising spin glass. *Science* **295,** 2427–2430 (2002).
36. Heim, B., Rønnow, T. F., Isakov, S. V. & Troyer, M. Quantum versus classical annealing of ising spin glasses. *Science* **348,** 215–217 (2014).
37. Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. Quantum computation by adiabatic evolution. *MIT-CTP-2936* (2000).
38. Farhi, E., Goldstone, J. & Gutmann, S. A numerical study of the performance of a quantum adiabatic evolution algorithm for satisfiability. *MIT-CTP-3006* (2000).
39. Choi, V. Adiabatic quantum algorithms for the NP-complete Maximum-Weight Independent set, Exact Cover and 3SAT problems. ArXiv:1004.2226 (2010).
40. Childs, A. M., Farhi, E., Goldstone, J. & Gutmann, S. Finding cliques by quantum adiabatic evolution. *Quantum Information and Computation* **2,** MIT-CTP #3067 (2002).
41. Peng, X. *et al.* Quantum adiabatic algorithm for factorization and its experimental implementation. *Physical Review Letters* **101,** 220405 (2008).
42. Hen, I. & Young, A. P. Solving the graph-isomorphism problem with a quantum annealer. *Physical Review A* **86,** 042310 (2012).
43. Gaitan, F. & Clark, L. Graph isomorphism and adiabatic quantum computing. *Physical Review A* **89,** 022342 (2014).
44. Bian, Z., Chudak, F., Macready, W. G., Clark, L. & Gaitan, F. Experimental determination of ramsey numbers. *Physical Review Letters* **111,** 130505 (2013).
45. Neven, H., Denchev, V. S., Rose, G. & Macready, W. G. Training a binary classifier with the quantum adiabatic algorithm. ArXiv:0811.0416 (2008).
46. Denchev, V. S., Ding, N., Vishwanathan, S. & Neven, H. Robust classification with adiabatic quantum optimization. ArXiv:1205.1148 (2012).
47. Roland, J. & Cerf, N. J. Quantum search by local adiabatic evolution. *Physical Review A* **65,** 042308 (2002).
48. Garnerone, S., Zanardi, P. & Lidar, D. A. Adiabatic quantum algorithm for search engine ranking. *Physical Review Letters* **108,** 230506 (2012).
49. Barahona, F. On the computational complexity of ising spin glass models. *Journal of Physics A*: *Mathematical and General* **15,** 3241, URL http://stacks.iop.org/0305-4470/15/i=10/a=028 (1982).
50. Lucas, A. Ising formulations of many NP problems. ArXiv:1302.5843 (2013).
51. Crosson, E., Farhi, E., Lin, C. Y.-Y., Lin, H.-H. & Shor, P. Different strategies for optimization using the quantum adiabatic algorithm. ArXiv:1401.7320 (2014).
52. Aaronson, S. BQP and the polynomial hierarchy. *Proceedings of the forty-second ACM Symposium on Theory of Computing* (*STOC*) 141–150 (2010).
53. Nagaj, D., Somma, R. D. & Kieferova, M. Quantum Speedup by Quantun Annealing. *Phys. Rev. Lett.* **109,** 050501 (2012).
54. Denchev, V. S. *et al.* What is the computational value of finite range tunneling? ArXiv:1512.02206 (2015).
55. Karp, R. M. Reducibility among Combinatorial Problems: Proceedings of a symposium on the Complexity of Computer Computations 85–103 (1972).
56. Hassin, R. & Segev, D. The Set Cover with Pairs Problem. *Lecture Notes in Computer Science* **3821,** 164–176 (2005).
57. Breslau, L. *et al.* Disjoint-path Facility Location: Theory and Practice. *Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments* (*ALENEX*) 60–74 (2011).
58. Lancia, G., Pinotti, C. M. & Rizzi, R. Haplotyping populations by pure parsimony: Complexity, exact and approximation algorithms. *INFORMS Journal on Computing* **16,** 348–359 (2004).
59. Wang, I.-L. & Yang, H.-E. Haplotyping populations by pure parsimony based on compatible genotypes and greedy heuristics. *Applied Mathematics and computation* **217,** 9798–9809 (2011).
60. Gonçalves, L. B., de Lima Martins, S., Ochi, L. S. & Subramanian, A. Exact and heuristic approaches for the set cover with pairs problem. *Optimization Letters* **6,** 641–653 (2012).
61. Klymko, C., Sullivan, B. D. & Humble, T. S. Adiabatic quantum programming: Minor embedding with hard faults. ArXiv:1210.8395 [quant-ph] (2012).
62. Choi, V. Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design. *Quantum Information Processing* **10,** 343–353 (2011).
63. Bian, Z. *et al.* Discrete optimization using quantum annealing on sparse ising models. *Frontiers in Physics* **2** (2014).
64. Messiah, A. *Quantum Mechanics* Volume 2 (North-Holland Publishing Company, 1962).
65. Perdomo-Ortiz, A., Dickson, N., Drew-Brook, M., Rose, G. & Aspuru-Guzik, A. Finding low-energy conformations of lattice protein models by quantum annealing. *Scientific Reports* **2** (2012).
66. Biamonte, J. D. Nonperturbative k-body to two-body commuting conversion hamiltonians and embedding problem instances into ising spins. *Physical Review A* **77,** 052331 (2008).
67. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220,** 671–680 (1983).
68. Isakov, S. V., Zintchenko, I. N., Ronnow, T. F. & Troyer, M. Optimised simulated annealing for ising spin glasses. *Computer Physics Communications*, **192** 265–271 (2015).
69. Choi, V. Minor-embedding in adiabatic quantum computation: I. The Parameter setting problem. *Quantum Information Processing* **7,** 193–209 (2008).

Acknowledgements

Author Contributions

All authors contributed to the initial idea of the study. Y.C. and S.J. showed the theoretical results. S.J., D.P. and S.K. designed the numerical tests and analyzed the results. All authors participated in the preparation of the manuscript.

Additional Information