

## 4-Clique network minor embedding for quantum annealers

Elijah Pelofske<sup>\*</sup>*Los Alamos National Laboratory, CCS-3 Information Sciences*
 (Received 22 September 2023; revised 18 January 2024; accepted 13 February 2024; published 13 March 2024)

Quantum annealing is a quantum algorithm for computing solutions to combinatorial optimization problems. This study proposes a method for minor embedding optimization problems onto sparse quantum annealing hardware graphs called 4-clique network minor embedding. This method is in contrast to the standard minor embedding technique of using a path of linearly connected qubits in order to represent a logical variable state. The 4-clique minor embedding is possible on Pegasus graph connectivity, which is the native hardware graph for some of the current D-Wave quantum annealers. The Pegasus hardware graph contains many cliques of size 4, making it possible to form a graph composed entirely of paths of connected 4-cliques on which a problem can be minor-embedded. The 4-clique chains come at the cost of additional qubit usage on the hardware graph, but they allow for stronger coupling within each chain, thereby increasing chain integrity, reducing chain breaks, and allow for greater usage of the available energy scale for programming logical problem coefficients on current quantum annealers. The 4-clique minor embedding technique is compared with the standard linear path minor embedding with experiments on two D-Wave quantum annealing processors with Pegasus hardware graphs. We show proof-of-concept experiments where the 4-clique minor embeddings can use weak chain strengths while successfully carrying out the computation of minimizing random all-to-all spin glass problem instances.

DOI: [10.1103/PhysRevApplied.21.034023](https://doi.org/10.1103/PhysRevApplied.21.034023)

### I. INTRODUCTION

Quantum annealing is a type of analog quantum computation which is effectively a relaxed version of adiabatic quantum computing. Quantum annealing is designed to solve combinatorial optimization problems by using quantum fluctuations in order to minimize an encoded problem Hamiltonian [1–8]. This process of computation works by starting the system in the easy-to-prepare ground state of a Hamiltonian, and then slowly transitioning the system into a second Hamiltonian of which we wish to find the ground state (generally this second Hamiltonian corresponds to a problem of which we do not know the ground state because it is difficult to compute). In the transverse-field Ising model version of quantum annealing, the system is put into an initial uniform superposition that is the ground state of the Hamiltonian:

$$H_{\text{initial}} = \sum_i^n \sigma_i^x, \quad (1)$$

---

<sup>\*</sup>epelofske@lanl.gov

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

where  $\sigma_i^x$  is the Pauli matrix for each qubit at index  $i$ . The user-programmed problem Hamiltonian is then turned on over the course of the anneal:

$$H(t) = A(t)H_{\text{initial}} + B(t)H_{\text{ising}}. \quad (2)$$

Combined,  $A(t)$  and  $B(t)$  define the *anneal schedules*. Typically, at  $t = 0$  the  $A(t)$  term is dominant as the system is prepared in the ground state of  $H_{\text{initial}}$ , and therefore the qubits are put into an initial superposition, and at the end of the anneal  $B(t)$  (the problem Hamiltonian) is dominant. The *annealing time* over which these schedules are applied can be varied, and in the case of physically implemented quantum annealers the possible annealing time is constrained by the hardware. At the end of the anneal, the qubit states are read out as classical bits, which correspond to the variable states. Those samples are intended to be low-energy solutions to the problem Ising  $H_{\text{ising}}$  that the user has specified. For D-Wave quantum annealers, the user can program the anneal schedule—specifically the ratio between  $A(s)$  and  $B(s)$  (known as the anneal fraction) for each point in time during the anneal. The classical problem Hamiltonian, composed of single- and two-body terms, is defined as

$$H_{\text{ising}} = \sum_i^n h_i \sigma_i^z + \sum_{i < j}^n J_{ij} \sigma_i^z \sigma_j^z. \quad (3)$$

The classical problem Hamiltonian is equivalent to a large class of combinatorial optimization problems. Specifically, for optimization problems where the decision variables are discrete, the variable states for Ising models can be either  $+1$  or  $-1$ . There are formulations known for converting many NP-hard problems into discrete combinatorial optimization problems of this form, thereby allowing quantum annealers to sample low-energy classical solutions of the programmed optimization problems. The D-Wave company has created a number of quantum annealing processors using superconducting flux qubits; these devices have been applied to a wide range of problems [4,9,10]. Quantum annealing has experimentally been shown to be competitive for good heuristic sampling of combinatorial optimization problems [11–14] and simulation of frustrated magnetic systems [15–17]. Although the size (e.g., the number of qubits) has been increasing as hardware development improves, the hardware of these quantum annealers is relatively sparsely connected, thus limiting what problem connectivity graphs can be sampled on these devices.

Minor embedding is the only mechanism that allows logical problems with structure that does not directly match the underlying hardware graph to be programmed onto the hardware [18–25]. In minor embedding for quantum annealing, each variable on the logical problem graph can be represented by a collection of physical qubits which are linked together ferromagnetically. The ferromagnetic coupling attempts to ensure that the variables representing each logical qubit are in agreement as to what the logical variable state is; there is an energy penalty for a qubit not having the same spin state as its neighbors. The standard minor embedding that is used creates a linear path of physically linked qubits, typically a linear nearest neighbors (LNN) graph, which form this ferromagnetically bound logical variable. These linear path groups of qubits are typically referred to as *chains*.

The essential idea of computing graph minors, in particular for embedding a problem onto a fixed hardware graph, is that the minor does not need to be a linear path; it can be effectively any graph structure that is embeddable onto the target graph. The constraint is that the minor embeddings ideally should require as few additional hardware qubits (and couplers) as possible to fit more problems, or larger problems, onto the hardware chip. Therefore, the linear paths are used so as to reduce the overhead of using additional physical qubits, so that each logical variable can be routed so that the required logical quadratic variable interactions can occur.

In this paper we propose a new method of minor embedding which we will refer to as *4-clique network minor embedding*. This method is based on a property of one of the D-Wave quantum annealing device connectivities, called Pegasus [24,26–28], which (while still quite sparse) contains a large number of 4-cliques throughout its

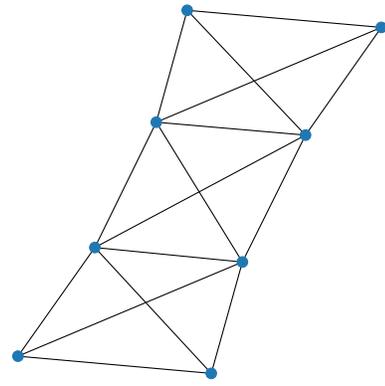


FIG. 1. A path of 4-cliques.

hardware lattice. The first D-Wave devices that were manufactured had a connectivity graph called Chimera [24,29], which is sparser than Pegasus. The newest generation of D-Wave quantum annealing hardware has a graph connectivity called Zephyr [28,30]. With the 4-cliques in the Pegasus hardware graph, it is possible to form connected paths of 4-cliques (see Fig. 1) from *4-clique chains* in order to create minor embeddings onto Pegasus chip hardware. While the 4-clique minor embedding uses more qubits, it allows significantly more ferromagnetic couplers to be programmed in each chain, which reinforces the integrity of each chain more than a linear path connectivity. With linear path embeddings the measured qubits in a chain often disagree on their logical spin state, especially for very long chains. Chains that have measured spins which are not the same are referred to as *broken chains*. Therefore, 4-clique embeddings allow more ferromagnetic chain break penalty weights to be used for each logical variable, thereby ideally reducing the number of chain breaks. The intuitive reasoning is that with a greater number of ferromagnetic couplers enforcing the state of each logical variable, there will be fewer chain breaks and therefore smaller chain strengths can be used.

One of the problems in general with minor embedding is that the strong ferromagnetic couplers can dominate the programmable energy range on the chip; current D-Wave devices have a set range of physical weights that the user can program to specify the problem Hamiltonian. There is limited precision when encoding these weights (approximately two decimal places), and therefore it is important to use as much of the physical programmable weight range as possible for the problem coefficients (instead of using those weights for encoding a minor embedding). Adding in the strong ferromagnetic chain couplers reduces the effective range that can be used to program the logical problem weights. The 4-clique network minor embedding is therefore useful for a second reason: by increasing the number of ferromagnetic couplers per chain, the (relative)

magnitude of the ferromagnetic couplers can be reduced compared to linear path embeddings which could allow for a larger effective programming weight range to be used for programming the weights of the logical problem.

We note that a related idea is Quantum annealing correction [12,31–37], where the states of problem variables are reinforced using ferromagnetic couplings to a common penalty qubit.

Section II describes the 4-clique graph construction in detail, along with creating some example minor embeddings. Section III describes the quantum annealing experimental results on the 4-clique minor embeddings compared to linear path minor embeddings when executed on D-Wave quantum annealers. Section IV concludes with what the results show with regard to the effectiveness of 4-clique minor embedding and future research questions. The figures in this paper were generated using Matplotlib [38,39], NetworkX [40], and D-Wave NetworkX [41] in Python 3. Data associated with this paper, including raw D-Wave measurements and minor embeddings, are available as a public dataset [42].

## II. METHODS

Section II A describes the 4-clique graph construction on a Pegasus graph, and how the minor embedding process works using this 4-clique network. Section II B describes the implementation of 4-clique minor embeddings on quantum annealing hardware; specifically, the problem instances that are used to compare 4-clique and equivalent linear path minor embeddings are described, along with the D-Wave parameter settings used for the experiments.

### A. 4-clique network minor embedding

Algorithm 1 constructs a network of 4-clique paths from a hardware graph connectivity. Algorithm 1 assumes that the hardware graph contains at least one clique of size 4, otherwise it does not contract any edges in the graph. For example, Algorithm 1 performs no

contractions to a Chimera graph since the maximum clique of a Chimera lattice is 2. The largest connected component (there are other smaller components which are not connected to the main graph) of the contracted 4-clique graphs of Advantage\_system6.1 and Advantage\_system4.1 are shown using different layout algorithms in Figure 2. Note that the resulting contracted 4-clique graph from Algorithm 1 may not be connected, and may not have large connected components. The set of connected components, and thus the largest connected component, was computed using NetworkX [40,43]. Once this contracted 4-clique graph of a target hardware graph has been created, standard minor embedding algorithms such as Minorminer [44,45] can be applied to create a minor embedding composed of chains of 4-cliques, such as in Fig. 1. For the purpose of constructing the 4-clique graph from the contracted clique graph provided by Algorithm 1, one can take the subgraph of the device hardware graph induced by the separated nodes  $n_x$ ,  $n_y$ , given by the name of the contracted nodes for all of the contracted nodes in the 4-clique chain. Table I shows a hardware summary in terms of qubits and couplers for the two Pegasus hardware graph D-Wave devices that are used for analyzing and implementing example spin glass problems, embedded using 4-clique network minor embeddings (and compared to the linear path minor embeddings). The largest connected component of the contracted 4-clique network for Advantage\_system4.1 uses a proportion of 0.878 of the available hardware qubits, and a proportion of 0.623 of the available hardware couplers. The largest connected component of the contracted 4-clique network for Advantage\_system6.1 uses a proportion of 0.877 of the available hardware qubits, and a proportion of 0.622 of the available hardware couplers.

To make two variable interactions possible in the 4-clique minor embedding, since all chains are already part of a 4-clique network, every 4-clique chain can be connected to another 4-clique chain by four physical couplers for each single coupler that was computed in the

ALGORITHM 1. Contract hardware graph to a 4-clique network.

- 
- 1: **Input:** Hardware graph  $G$
  - 2: **4-cliques**  $\leftarrow$  Compute all cliques of size 4 in  $G$
  - 3: **for**  $K \in$  4-cliques **do**
  - 4:   If any of the nodes in  $K$  have already been contracted in a previous iteration, skip this iteration
  - 5:   Randomly choose two of the nodes  $n_1, n_2$  from  $K$
  - 6:   Contract edge  $(n_1, n_2)$  to form a node called  $n_1n_2$
  - 7:   Choose the remaining two nodes  $n_3, n_4$  from  $K$
  - 8:   Contract edge  $(n_3, n_4)$  to form a node called  $n_3n_4$
  - 9:   Remove any self edges that may have been generated from these two edge contractions
  - 10: **end for**
  - 11: Remove all nodes and edges in  $G$  which were not formed by edge contraction. Specifically, remove all nodes that are not named with the form  $n_xn_y$  and remove all edges that were not formed by edge contractions (e.g. remove all edges that do not have the form of  $(n_{x_1}n_{y_1}, n_{x_2}n_{y_2})$  )
  - 12: **Return:**  $G$
-

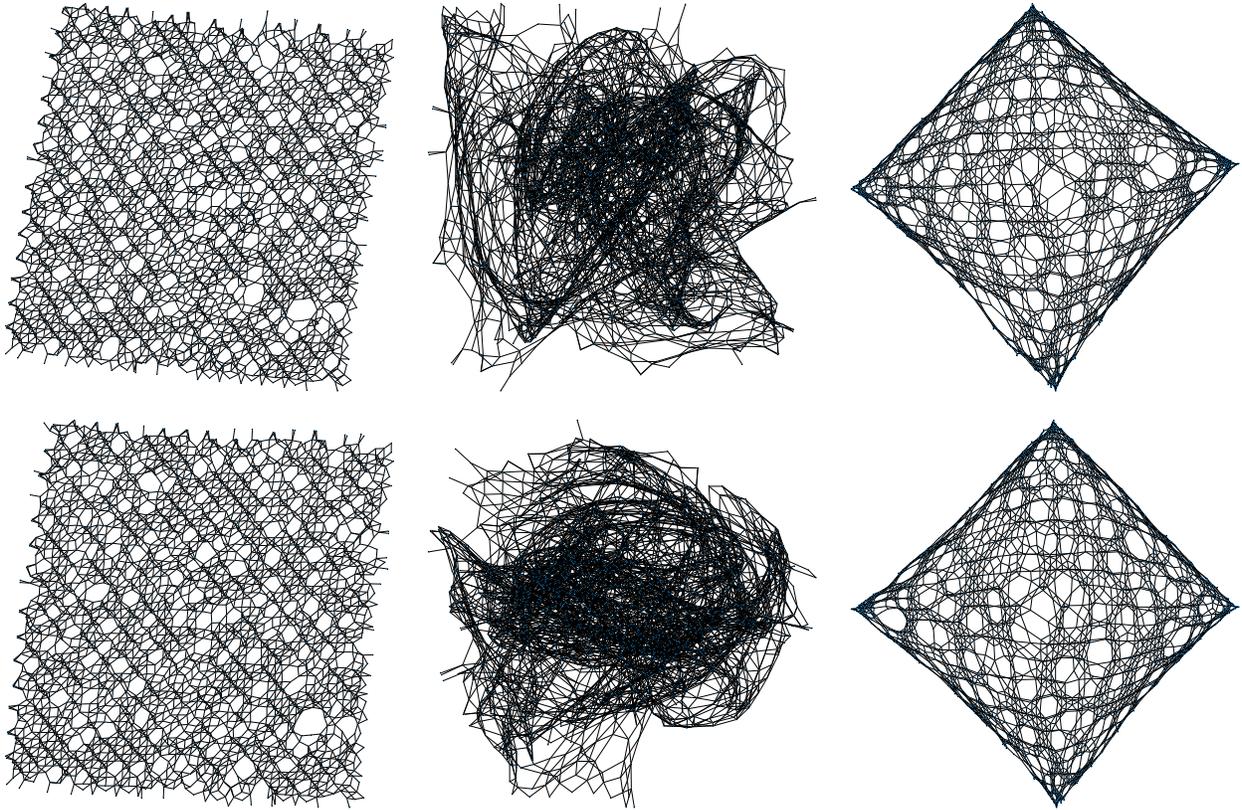


FIG. 2. Contracted 4-clique graph of `Advantage_system4.1` with Kamada-Kawai layout (top left), and spectral layout (top right). Contracted 4-clique graph of `Advantage_system6.1` with Kamada-Kawai layout (bottom left), spring layout (bottom middle), and spectral layout (bottom right). The contracted 4-clique graph of `Advantage_system4.1` has 2471 nodes and 6270 edges. The contracted 4-clique graph of `Advantage_system6.1` has 2463 nodes and 6245 edges. As defined by Algorithm 1, an edge in a contracted 4-clique graph represents four edges in the underlying hardware graph, and a node represents two physical qubits in the hardware graph. These contracted clique graphs are quite sparse; the maximum clique of both of these graphs is 2. When the clique contraction is performed on the hardware graphs, there are many small unconnected components that are generated. These graphs show only the largest connected component since it is the one that can be used for large minor embeddings.

minor embedding for the contracted clique graph. Note that although a linear path minor embedding is usually computed by forming a single path of connected qubits, it is possible for the minor embedding algorithm to use some branching if required, and therefore sometimes the standard minor embeddings are not strictly linear path (although almost always they do create linear path chains).

To provide a direct comparison between a 4-clique embedding and a linear path embedding, we can take any minor embedding of a problem connectivity with the target of the contracted clique graphs (e.g., in Fig. 2), and

separate out a linear path by taking one of the two variables in each node pair  $n_x, n_y$ . This means that we can compute a minor embedding of a problem on a contracted 4-clique graph, and then compute a linear path embedding of the same path length (but half the number of qubits in each chain) for the purpose of directly comparing the two embeddings since they have high hardware overlap.

In linear path minor embedding chains, the node degrees are usually either 2, for nodes within the chain, or 1 for nodes at the end of the chain assuming the chain is strictly LNN. The minor embedding may utilize branching of a

TABLE I. D-Wave quantum annealing processor hardware summary. Note that each of these devices has some hardware defects which cause the available hardware (qubits and couplers) to be smaller than the ideal Pegasus  $P_{16}$  graph lattice structure.

D-Wave QPU chip id	Topology name	Available qubits	Available couplers	Annealing time (min, max) [ $\mu$ s]
<code>Advantage_system4.1</code>	Pegasus $P_{16}$	5627	40 279	(0.5, 2000)
<code>Advantage_system6.1</code>	Pegasus $P_{16}$	5616	40 135	(0.5, 2000)

linear path, resulting in node degrees of 3, for example. As shown by Fig. 1, in 4-clique chains the node degrees are 5 for nodes within the chain and 3 for nodes at the end of the chains (again, assuming the 4-clique path is linear, and not branching).

The largest all-to-all minor embeddings which could be constructed, using heuristic minor embedding algorithms, on the contracted 4-clique networks for both `Advantage_system6.1` and `Advantage_system4.1` are 32 node cliques. Example 4-clique random minor embeddings for  $K_{32}$  graphs are shown in Fig. 5, overlaid onto the Pegasus hardware graph. Table II in Appendix B details all of the computed minor embedding chain length statistics for the 4-clique minor embeddings for  $N = 3, \dots, 32$ .

D-Wave quantum annealers currently have three distinct hardware connectivities, called Chimera, Pegasus, and Zephyr. Chimera graphs are too sparse for the 4-clique minor embedding scheme (the maximum clique of a Chimera graph is 2). Pegasus works perfectly for this idea; in fact its highly connected 4-cliques motivated this idea. Zephyr also has cliques of size 4, but Zephyr cannot form a large fully connected 4-clique network; instead the 4-clique contractions result in disconnected subgraphs. The contracted clique graph of a Zephyr  $Z_{16}$  graph is shown in Appendix A. This means that large 4-clique minor embeddings could not be created using Zephyr hardware, whereas for Pegasus larger 4-clique minor embeddings can be created.

## B. Implementation on quantum annealing hardware

An important point when implementing 4-clique minor embeddings is that the number of qubits used is likely significantly more than for an equivalent linear embedding. Therefore, as when minor-embedding very large problems with linear path chains, it is important to consider whether the uniform problem coefficient spreading causes the programmed weights to fall below the machine precision. If this is the case, for very large minor embeddings, it may be necessary to create nonuniform problem weight encoding on the chains. The number of couplers used to actually encode quadratic terms could also be varied; in the 4-clique embedding four couplers can be used, and those weights could be distributed in nonuniform ways (e.g., the weights could be placed entirely on one coupler). For the experimental results shown in Sec. III, uniform weight distributions are used for both the linear and the quadratic terms.

Section III reports experimental energy results from using both the equivalent linear minor embedding and the 4-clique network minor embedding. The linear minor embedding is constructed from the 4-clique minor embedding by taking only one linear path down the 4-clique chain—and thereby using exactly one half of the qubits

as the equivalent 4-clique minor embedding. This conversion from a 4-clique minor embedding to the linear path minor embedding is shown as side-by-side comparisons in Figures 3 and 4. The chain strength is the primary parameter of interest when comparing these minor embeddings because that chain strength will be applied to both the linear path minor embeddings and the 4-clique minor embeddings. The relevant question is whether there is a difference between the two embeddings when the chain strength is the same. Importantly, the comparison with the linear path minor embeddings is a fair comparison of the impact of varying chain strength since the same region of the hardware graph is used. Note, however, that optimized linear path minor embeddings could have significantly shorter chain lengths and thus perform better for smaller problem instances where chain breaks and incoherent quantum annealing are dominant sources of error.

When executing the problem instances on the D-Wave quantum annealers, autoscaling is left on. Autoscaling is a back-end parameter which, if left on, will scale the provided problem coefficients into the maximum energy scale that is possible on the quantum annealer, for both the linear and quadratic terms. Each parameter combination of the quantum annealer uses exactly 1000 anneals. The annealing time is varied for the purpose of observing what effect it has on the results of the computation. The parameter `programming_thermalization` was set to 0  $\mu$ s, and all other programming parameters were set to their default values.

Another critical component of using minor embeddings is how to handle cases where the chains do not agree on the logical variable state (i.e., the chain is *broken*). There are simple chain break resolution methods such as majority vote which can classically repair the chain with postprocessing. With the aim of illustrating the chain break frequency, along with solution quality in the D-Wave results reported in Sec. III, no classical postprocessing is used. In particular, any sample which contains a broken chain is set to have an energy value of 0, meaning that the overall measurement statistics still have 1000 data points per setting.

The problem instances we will consider are random spin glasses, defined on an all-to-all connected graph  $G = (V, E)$  of size  $N$ :

$$C(z) = \sum_{v \in V} w_v z_v + \sum_{(i,j) \in E} w_{ij} z_i z_j, \quad (4)$$

where  $w_{ij}$  and  $w_v$  denote random coefficients, chosen uniformly at random from  $\{+1, -1\}$ , making these problems effectively discrete-coefficient Sherrington-Kirkpatrick models [46] with local fields. The goal is to find the vector of variables  $z = [z_0, z_1, \dots, z_N]$  such that the cost function in Eq. (4) is minimized, where the decision variables  $z_i$  are spins (+1 or -1). This cost function evaluation, as it is

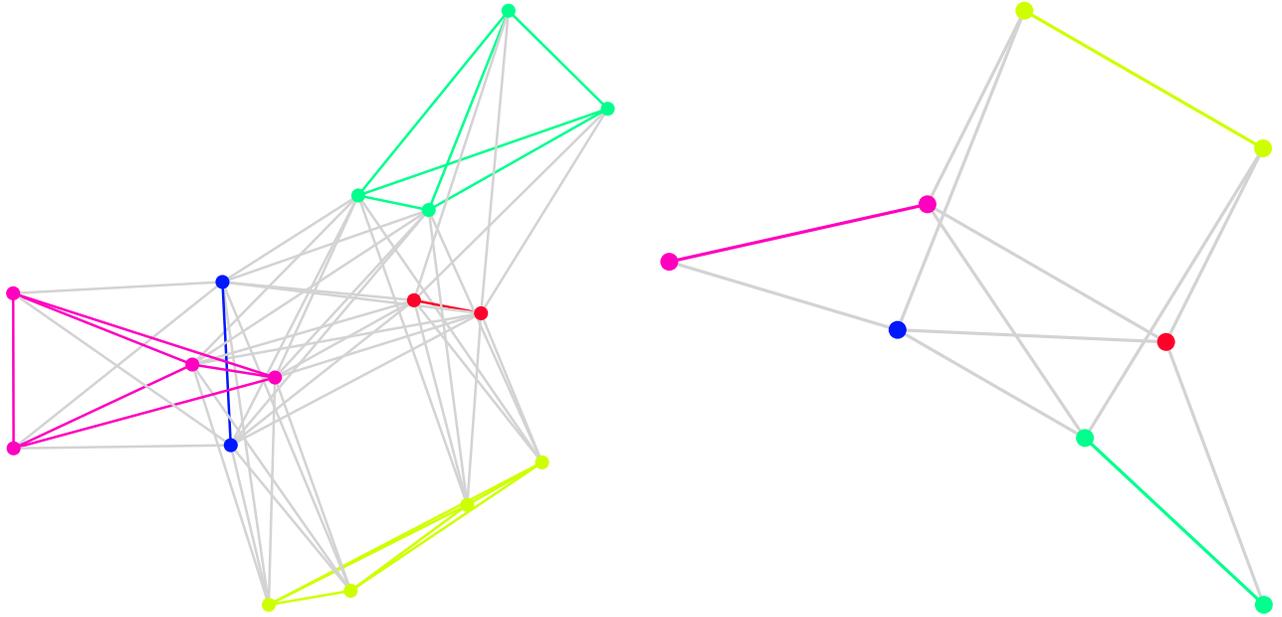


FIG. 3. Minor embedding of a  $K_5$  clique onto the target graph connectivity of Pegasus, using a 4-clique minor embedding (left) and an equivalent linear path minor embedding (right). Nodes in the graph are physical qubits, and edges are physical couplers. Gray edges represent the physical couplers onto which the problem-specific coefficients would be encoded. Colored edges and nodes denote the minor embeddings; each chain (comprised of qubits and couplers) is uniquely colored. Notice that the 4-clique minor embedding by default uses chains of length 2 as the smallest possible chains to encode a logical variable; these together form a single node pair  $n_x, n_y$  that are two of the variables in a 4-clique in the hardware. The linear path embedding by contrast does use chains of length 1 (meaning there is no chain). Qubits and couplers not used by the minor embedding are not shown. Because Pegasus natively has cliques of size 4, it makes no sense to actually use a minor embedding of size 4. Therefore, this specific minor embedding diagram is purely for the purposes of describing the 4-clique minor embedding algorithm.

an Ising model, is referred to as the *energy* for that given set of variable assignments. For each problem size  $N$  that is tested, a new problem instance is generated with new random coefficients.

### III. RESULTS

This section analyzes experimental results from executing random spin glasses on `Advantage_system4.1` and `Advantage_system6.1`.

Figure 6 shows a side-by-side comparison of 4-clique and linear path minor embeddings on  $N = 8$  random problem instances, using relatively small chain strengths of 1.1 and 1.01. It is evident that there is very little difference between the energy distributions for the two minor embeddings—both minor embeddings have very low chain break rates and both have converged to the same optimal solution. Note that this proof of concept with this eight-variable problem instance would perform better on the hardware if encoded using optimized small chain length linear path minor embeddings compared to the 4-clique network minor embedding given how small the problem instance is. We will probe larger problem instances

next so as to see how more complicated 4-clique minor embeddings perform.

Figure 7 shows the same comparison as Fig. 6, except that the problem size was increased to  $N = 10$ . In these plots, there is now a clear difference between the 4-clique and the comparable linear path minor embeddings. At these very low chain strengths, the greater connectivity of the 4-clique path minor embedding allowed the computation to remain stable and find low-energy solutions. By contrast, the linear path minor embedding has an extremely high chain break frequency and therefore the computations are not as robust at finding low-energy solutions. This shows the 4-clique minor embedding is able to utilize a smaller proportion of the available programmable coefficient range on the hardware by being able to carry out the computation with a chain strength of only 1.01 (note that the hardware option of autoscaling is turned on), compared to the equivalent minor embedding technique. There are a few outlier instances where the linear path minor embedding is able to sample the optimal solution, but only when the chain strength was set to 1.1.

Figure 8 continues the trend observed in Fig. 7 where the 4-clique minor embedding is able to sample low-energy solutions with minimal chain breaks compared

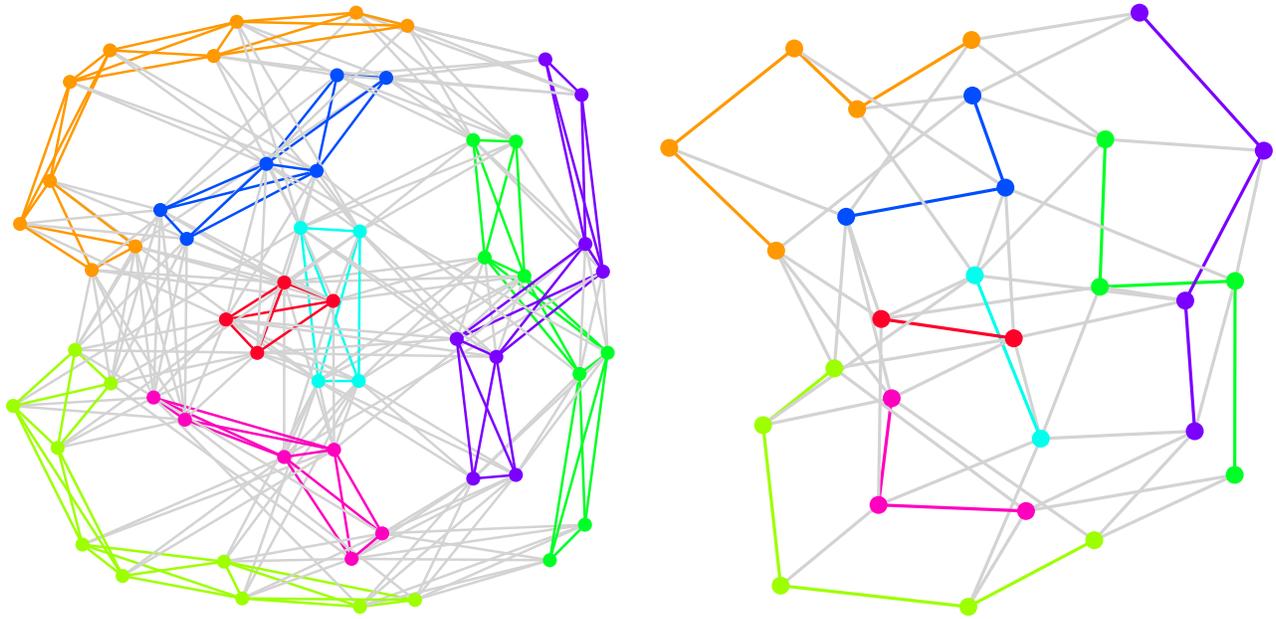


FIG. 4. Minor embedding of a  $K_8$  clique using a 4-clique minor embedding (left) and an equivalent linear path minor embedding (right) on a Pegasus graph. Gray edges represent the physical couplers onto which the problem-specific coefficients would be encoded. Colored edges and nodes denote the minor embeddings; each chain (comprised of qubits and couplers) is uniquely colored. As in Fig. 3, the parts of the Pegasus graph that are not used by the minor embeddings are not shown.

to the equivalent linear path minor embedding for an  $N = 15$  problem instance. Notably, all of the samples for the linear path minor embedding had broken chains

and therefore clearly performed worse than the 4-clique minor embedding. This is a critical observation that the 4-clique network minor embedding was able to obtain

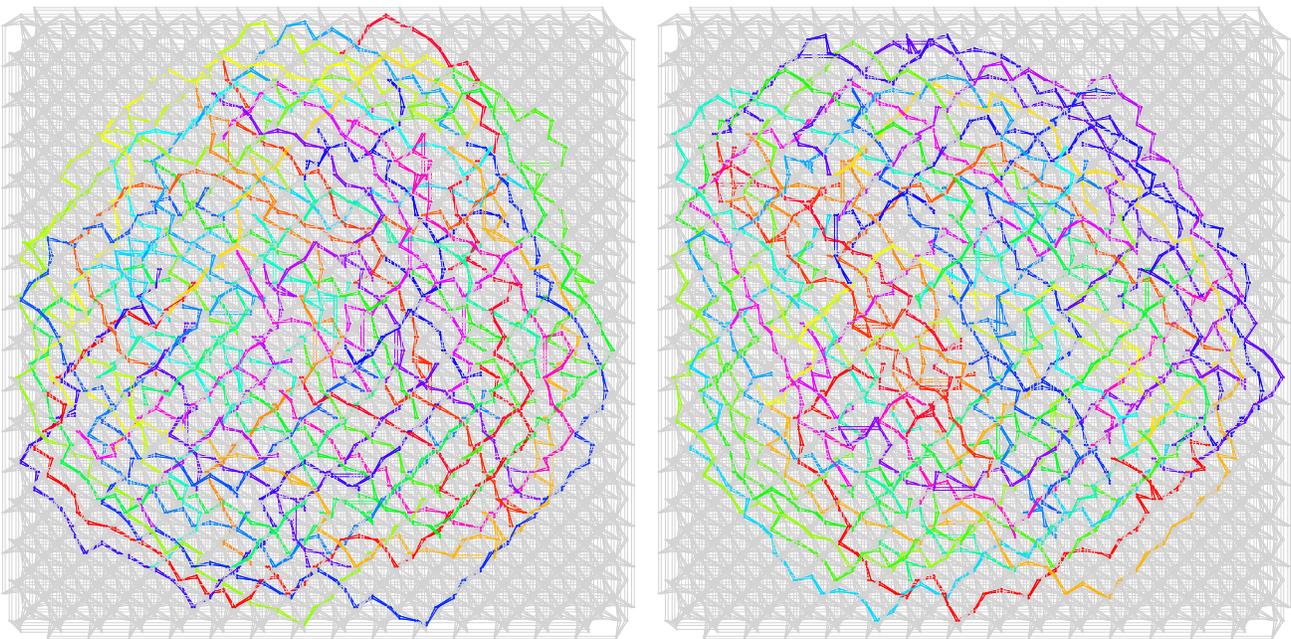


FIG. 5.  $K_{32}$  4-clique minor embedding on the  $P_{16}$  Pegasus hardware graphs of Advantage\_system4.1 (left) and Advantage\_system6.1 (right). Each of the 32 chains is uniquely colored. These are the largest all-to-all 4-clique minor embeddings that could be computed using Minorminer in a reasonable amount of time.

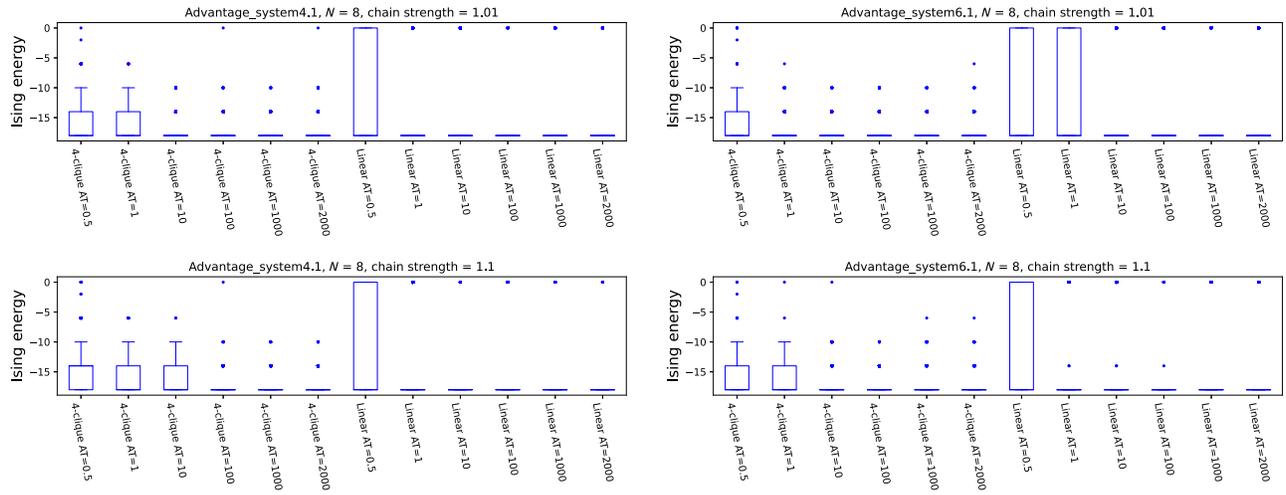


FIG. 6. Sampling a single  $N = 8$  variable random spin glass instance using Advantage\_system4.1 (left column) and Advantage\_system6.1 (right column). Chain strength of 1.01 (top row) and 1.1 (bottom row). Each plot is comprised of the spectrum of energy results from the 4-clique minor embedding in the left-hand portion, and the corresponding equivalent linear path minor embedding energy spectrum plots in the right-hand portion. Each set of data, for each combination of annealing time chain strength and minor embedding, is comprised of exactly 1000 samples. Annealing times (AT) of 0.5, 1, 10, 100, 1000, and 2000  $\mu\text{s}$  are used so as to evaluate how the two minor embeddings compare over different annealing times. Any outlier energy data points are represented as small blue dots, which may overlap each other.

low-energy samples of the Ising model using the comparatively extremely small chain strength of 1.01, leaving much of the available programmable coupler energy scale available for encoding of the coefficients of the original Ising model instead of dedicating that energy scale to ferromagnetic chains.

Figure 9 shows the energy results for an  $N = 20$  problem instance. Because of the increased chain lengths

(see Table II), the small chain strengths of 1.01 and 1.1 do not work well for this problem size. Therefore, this plot also includes results for a chain strength of 1.4. Although the chain strength needed to be increased to see good low-energy state sampling with the 4-clique minor embedding, it is still the case that the 4-clique embedding performed better than the linear path minor embedding, especially with respect to chain break frequency. At this very long

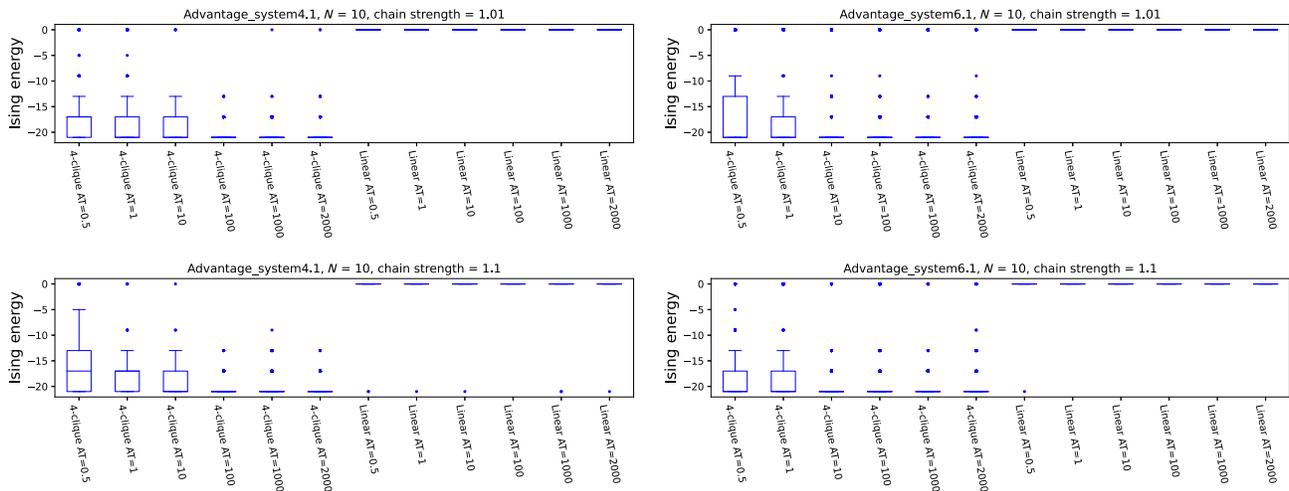


FIG. 7.  $N = 10$  variable random spin glass sample energies from Advantage\_system4.1 (left column) and Advantage\_system6.1 (right column). Chain strength of 1.01 (top row) and 1.1 (bottom row). Each plot is comprised of the distribution of energy results from the 4-clique minor embedding in the left-hand portion, and the corresponding equivalent linear path minor embedding energy spectrum plots in the right-hand portion. Annealing times of 0.5, 1, 10, 100, 1000, and 2000  $\mu\text{s}$  are used so as to evaluate how the two minor embeddings compare over different annealing times.

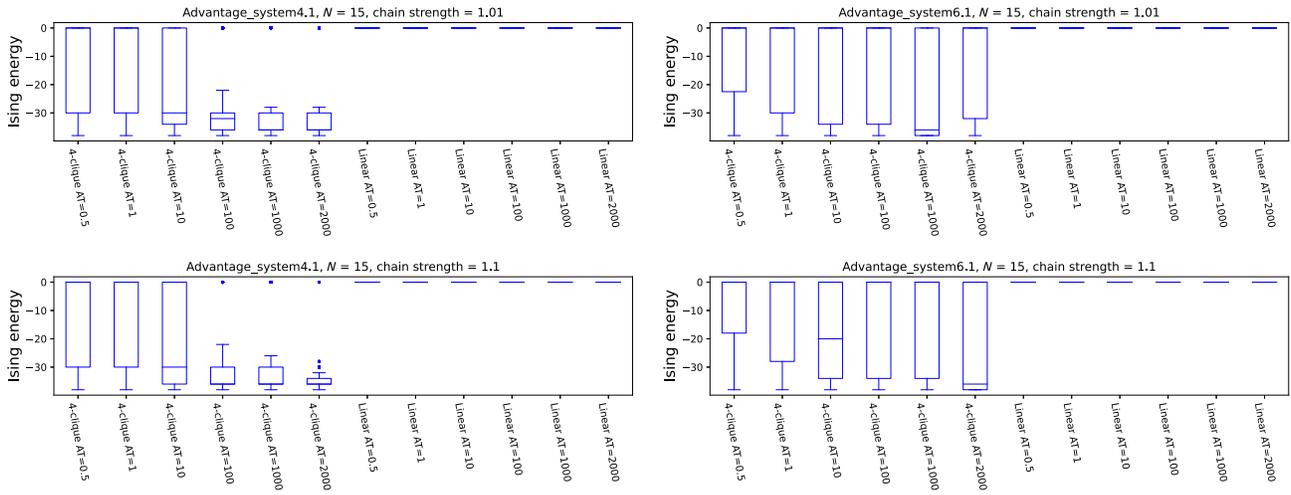


FIG. 8.  $N = 15$  variable random spin glass energy distributions from Advantage\_system4.1 (left column) and Advantage\_system6.1 (right column). Chain strength of 1.01 (top row) and 1.1 (bottom row). Each plot is comprised of the spectrum of energy results from the 4-clique minor embedding in the left-hand portion, and the corresponding equivalent linear path minor embedding energy spectrum plots in the right-hand portion. Annealing times of 0.5, 1, 10, 100, 1000, and 2000  $\mu\text{s}$  are used to evaluate how the two minor embeddings compare for different annealing times.

chain size, the linear path minor embedding results always had samples with broken chains.

Figure 10 shows the energy results for an  $N = 32$  Ising model problem instance. The chain lengths for this

problem size are dramatically larger than all of the other tested minor embeddings (see Table II), and consequently much larger chain strengths were required to obtain reasonable results. At a chain strength of 1.4, nearly all of

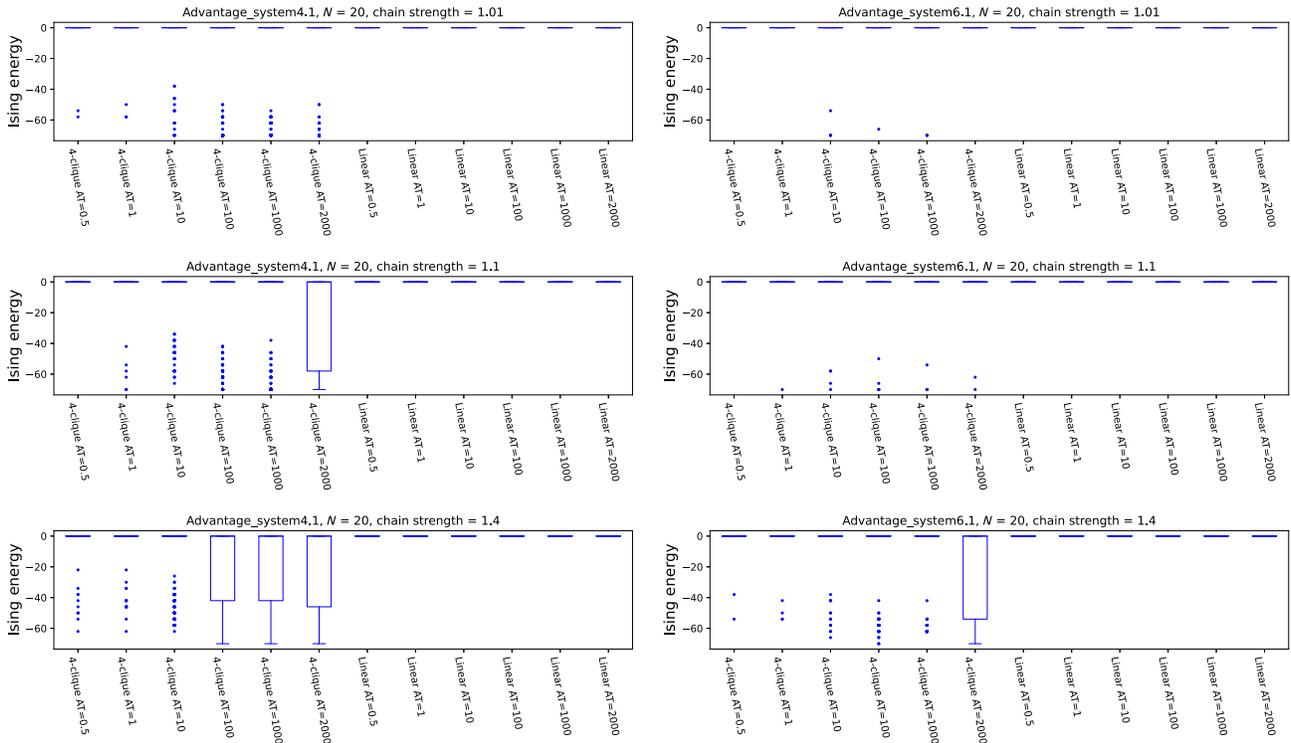


FIG. 9.  $N = 20$  variable random spin glass sample energies from Advantage\_system4.1 (left column) and Advantage\_system6.1 (right column). Chain strength of 1.01 (top row), 1.1 (middle row), and 1.4 (bottom row). Each plot is comprised of the distribution of energy results from the 4-clique minor embedding in the left-hand portion of the plot, and the corresponding equivalent linear path minor embedding energy spectrum plots in the right-hand portion of the plot. Annealing times of 0.5, 1, 10, 100, 1000, and 2000  $\mu\text{s}$  are used so as to evaluate how the two minor embeddings compare over different annealing times.

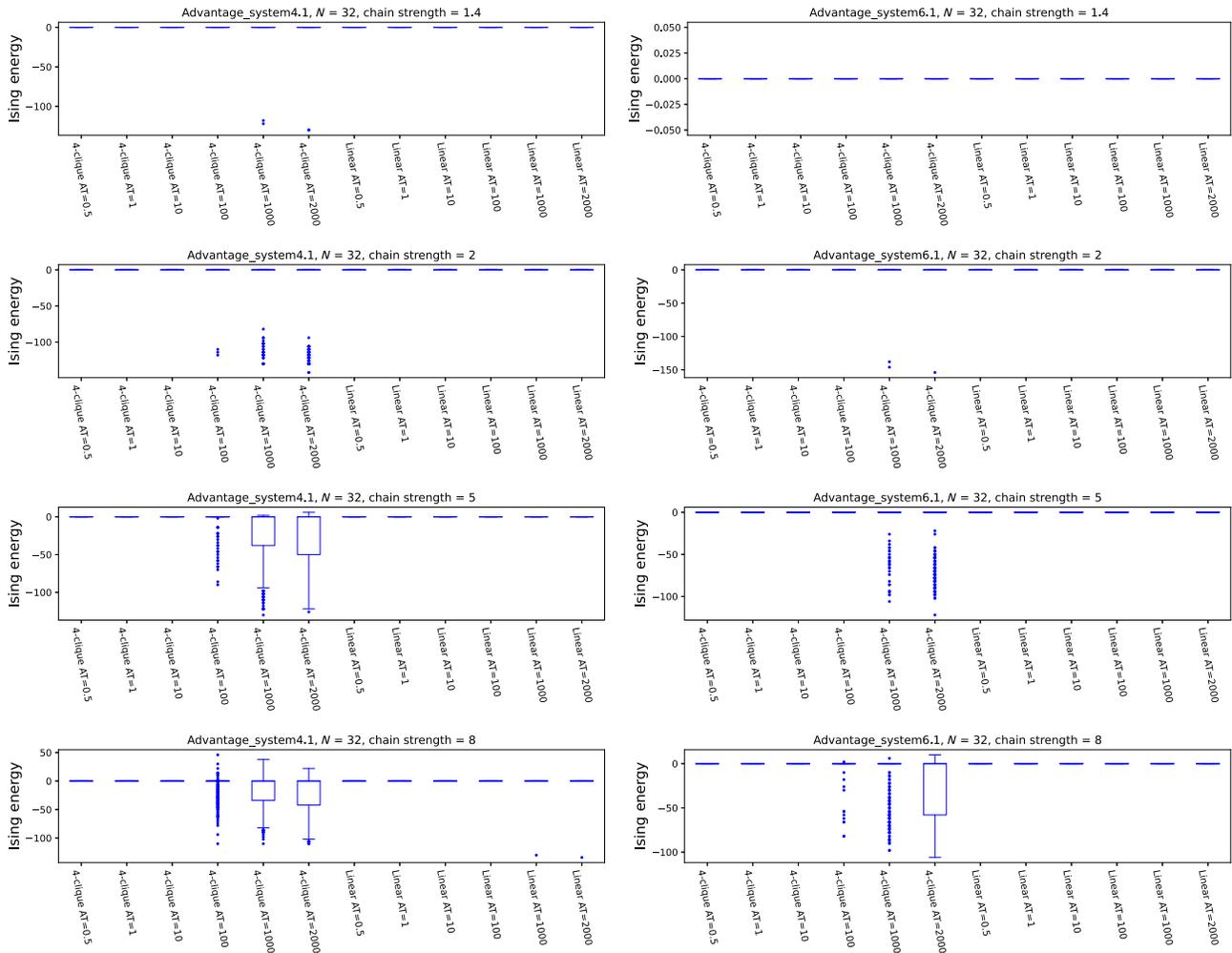


FIG. 10.  $N = 32$  variable random spin glass energy results from Advantage\_system4.1 (left column), Advantage\_system6.1 (right column). Chain strength of 1.4 (top row), 2 (second row), 5 (third row), and 8 (bottom row). Each plot is comprised of the spectrum of energy results, represented by separate boxplots, from the 4-clique minor embedding in the left-hand portion, and the corresponding equivalent linear path minor embedding energy spectrum plots in the right-hand portion. Annealing times of 0.5, 1, 10, 100, 1000, and 2000  $\mu\text{s}$  are used so as to evaluate how the two minor embeddings compare over different annealing times.

the samples had broken chains or an energy of 0; only the 4-clique minor embedding at anneal times of 1000 and 2000  $\mu\text{s}$  produced a few low-energy samples. Once again, even at these larger chain strengths, the 4-clique minor embedding still had much fewer chain breaks compared to the linear path minor embedding and therefore better samples, at least at sufficiently long annealing times (e.g., 1000 or 2000  $\mu\text{s}$ ). However, here we begin to see where too large of chain strengths can be detrimental for the 4-clique minor embedding. At a chain strength of 8, the 4-clique energy spectrum begins to clearly get worse than 0, whereas with a chain strength of 5 the energy results were actually better than with a chain strength of 8. This shows that the chain strength used in the 4-clique minor embedding needs to be carefully tuned such that results do not get worse because of the chain strength using too much of

the available programmable coupler weight. At comparatively small chain strengths the 4-clique minor embedding performs very well.

Figures 6–10 all show there is a clear trend for the 4-clique minor embedding across annealing times; the longer annealing times result in consistently lower-energy solutions. These figures also all show that Advantage\_system4.1 samples lower-energy solutions more consistently than Advantage\_system6.1.

#### IV. DISCUSSION AND CONCLUSION

In this paper we have introduced a new method for minor-embedding discrete combinatorial optimization problems onto D-Wave quantum annealers with hardware graphs that contain networks of 4-cliques. Importantly,

we do not expect this method to be used for the current  $P_{16}$  Pegasus devices. The reason for this is that the largest minor embeddings that can be formed from the contracted 4-clique networks of  $P_{16}$  graphs are not especially large in comparison to equivalent linear path minor embeddings that can be computed, and in particular linear path minor embeddings have considerably shorter chain lengths for small problem sizes and will thus perform considerably better and use less of the hardware graph than the proposed 4-clique network minor embedding. The reason for this is the longer chain lengths required to create a minor embedding on the contracted 4-clique graphs; linear path minor embeddings can be constructed to use much shorter chain lengths than the side-by-side comparisons done in Sec. III. For example, for problem instances with a small number of variables (such as of the order of 10 variables) and a standard coefficient precision range, the standard optimized linear path minor embeddings will perform much better than a 4-clique network minor embedding on a Pegasus graph due to decreased hardware usage and likely more coherent quantum annealing. The regime where the 4-clique network minor embedding could potentially be useful is for much larger quantum annealers (which can form a 4-clique network, such as devices with a Pegasus hardware graph) where minor embedding of significantly large problem sizes (e.g., hundreds or thousands of logical variables) will require exceedingly large ferromagnetic chain strengths and long chains using the equivalent linear path chain

minor embedding (which, if the chains are sufficiently long, will be prone to breaking). There, the 4-clique chains could provide lower levels of chain breaks due to the increased interchain integrity and require a smaller proportion of the programmable energy scale of the quantum annealer than alternative minor embedding methods. For example, if for a fully connected problem instance embedded on a large sparse hardware graph the linear path chain lengths are already significantly long such that significant chain breaks occur (and the maximum chain strengths are limited), then the 4-clique network minor embedding could be applied in order to perform improved quantum annealing on that hardware platform. On sparse hardware graphs, fully connected minor embeddings quickly use the available hardware graph as the number of variables increases and thus use long chains. Therefore 4-clique network minor embeddings could be used most effectively for fully connected minor embeddings on future large quantum annealing hardware graphs where long chains are inevitable even for linear path minor embeddings. Because of the limited size of current quantum annealing hardware, this type of large problem instance minor embedding comparison cannot be performed on current D-Wave quantum annealers, but this comparison should be performed on future quantum annealing hardware.

Additionally, even for small problem sizes, if the energy scale range of the problem coefficients is very important for a minor-embedded problem, then the 4-clique network minor embedding could be employed to make more of the

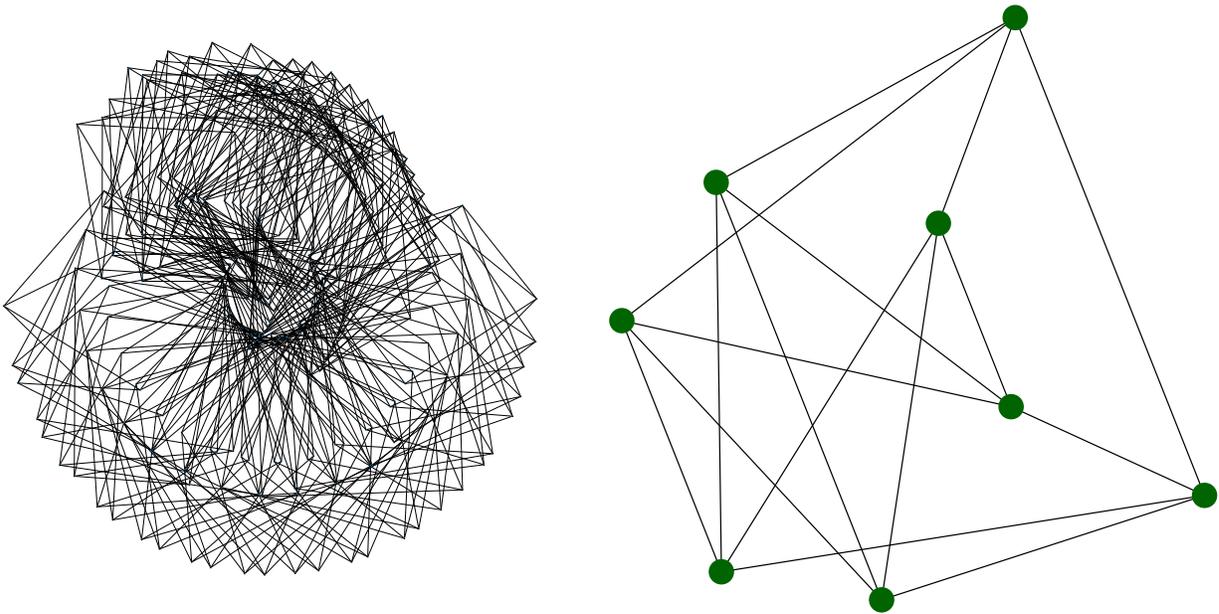


FIG. 11. Contracted 4-clique graph for a Zephyr  $Z_{16}$  graph (left), which is composed of 31 unconnected graphs. Each of those 31 unconnected subgraphs are isomorphic to each other; their structure is shown in the right-hand-side plot. These graphs are drawn using the Kamada-Kawai layout algorithm.

coefficient range available on the chip to be used for problem coefficients, instead of interchain couplers. This type of problem may occur for optimization problems where the high precision of the coefficients being programmed onto the hardware graph is important to represent the problem faithfully compared to what can be programmed using linear path minor embeddings, which can occur if the coefficient range of the original problem is large.

There are several future research questions that can be considered:

1. Make the weights on the chain nonuniform—perhaps make them proportional to their degree within the 4-clique chain.

2. Use flux bias offset and anneal offset calibration [47,48] to balance the chain statistics in the minor embeddings, both for 4-clique and the equivalent linear minor embeddings, with the goal of reducing bias in the quantum annealing samples.

3. Create more structured minor embeddings of the contracted 4-clique graphs, specifically with the goal of making more uniform chain lengths in the minor embeddings.

4. Investigate encoding variable states across even larger subgraphs of the hardware topology. By encoding variable states into larger pieces of the physical hardware, it becomes harder for noise to induce errors in the physical group of qubits.

5. Investigate how the 4-clique chains break. Is there a pattern with respect to the qubit degree within the 4-clique chain, or with respect to the position of the qubit in the chain?

6. Numerical simulations of exact quantum state evolution when using the standard linear minor embedding compared to the proposed 4-clique network minor embedding. Currently, such classical simulators are severely limited in the total number of qubits that can be simulated, making comparisons between even smaller toy examples computationally difficult. Future improved quantum annealing simulation software could be used to perform such numerical simulations, giving better insights into how these two minor-embedded methods work.

7. Use denser minor embeddings, such as the 4-clique network minor embedding, selectively and not for every single logical variable. For example, nodes in the logical graph which are highly connected could be represented by a large dense chain (such as a 4-clique chain), while less dense variables are represented by smaller chains.

## ACKNOWLEDGMENTS

This work was supported by the U.S. Department of Energy through the Los Alamos National Laboratory. The Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security

Administration of the U.S. Department of Energy (Contract No. 89233218CNA000001). The research presented in this paper was supported by the Laboratory Directed Research and Development program of the Los Alamos National Laboratory under project numbers 20220656ER and 20210114ER and by the NNSA’s Advanced Simulation and Computing Beyond Moore’s Law Program. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy

TABLE II. Summary statistics for random 4-clique all-to-all minor embedding chain lengths that were computed using several iterations of the Minorminer embedding heuristic. The all-to-all minor embeddings were computed for sizes  $3, \dots, 32$ . Specifically, the minimum, maximum, mean, and standard deviation of the chain lengths in the minor embedding are reported for the minor embeddings computed on the contracted 4-clique graphs of Advantage\_system4.1 and Advantage\_system6.1. Here chain length refers to the total number of physical qubits used in the minor embedding. All quantities are rounded to three decimal places.

N	Advantage_system4.1 4-clique minor embedding chain lengths (min, mean $\pm\sigma$ , max)	Advantage_system6.1 4-clique minor embedding chain lengths (min, mean $\pm\sigma$ , max)
3	(2, 2.667 $\pm$ 0.943, 4)	(2, 2.667 $\pm$ 0.943, 4)
4	(2, 3.0 $\pm$ 1.0, 4)	(2, 3.0 $\pm$ 1.0, 4)
5	(2, 3.2 $\pm$ 0.98, 4)	(2, 3.2 $\pm$ 0.98, 4)
6	(2, 4.0 $\pm$ 1.633, 6)	(2, 4.0 $\pm$ 1.633, 6)
7	(2, 4.857 $\pm$ 2.1, 8)	(2, 4.857 $\pm$ 2.1, 8)
8	(4, 7.0 $\pm$ 2.236, 10)	(4, 7.0 $\pm$ 2.646, 10)
9	(4, 8.222 $\pm$ 2.393, 12)	(4, 8.222 $\pm$ 2.393, 12)
10	(6, 10.4 $\pm$ 2.939, 14)	(6, 10.4 $\pm$ 2.939, 14)
11	(8, 13.818 $\pm$ 3.459, 18)	(6, 12.182 $\pm$ 3.242, 18)
12	(10, 15.667 $\pm$ 3.986, 22)	(8, 16.167 $\pm$ 6.189, 26)
13	(12, 18.769 $\pm$ 4.475, 26)	(12, 19.231 $\pm$ 4.933, 28)
14	(14, 23.429 $\pm$ 5.368, 30)	(12, 22.429 $\pm$ 5.716, 28)
15	(22, 27.467 $\pm$ 3.222, 32)	(16, 25.867 $\pm$ 4.646, 34)
16	(16, 31.75 $\pm$ 7.71, 42)	(18, 30.75 $\pm$ 7.137, 42)
17	(24, 34.824 $\pm$ 5.576, 42)	(16, 35.647 $\pm$ 10.295, 46)
18	(26, 37.778 $\pm$ 6.25, 46)	(24, 37.222 $\pm$ 7.634, 46)
19	(26, 43.263 $\pm$ 7.9, 52)	(32, 41.579 $\pm$ 7.802, 56)
20	(34, 47.7 $\pm$ 7.246, 58)	(28, 47.5 $\pm$ 11.897, 64)
21	(36, 50.476 $\pm$ 7.998, 60)	(32, 50.095 $\pm$ 7.47, 60)
22	(32, 54.455 $\pm$ 9.552, 66)	(32, 51.909 $\pm$ 11.305, 64)
23	(40, 60.261 $\pm$ 13.835, 82)	(38, 56.087 $\pm$ 7.235, 64)
24	(44, 62.417 $\pm$ 11.604, 84)	(34, 61.667 $\pm$ 13.972, 80)
25	(44, 66.48 $\pm$ 9.753, 78)	(44, 63.92 $\pm$ 10.438, 82)
26	(44, 70.462 $\pm$ 13.165, 92)	(52, 69.538 $\pm$ 9.548, 86)
27	(54, 73.185 $\pm$ 9.495, 86)	(56, 73.778 $\pm$ 8.35, 84)
28	(60, 81.214 $\pm$ 13.356, 106)	(56, 79.5 $\pm$ 9.571, 92)
29	(60, 85.31 $\pm$ 12.879, 104)	(54, 77.655 $\pm$ 12.606, 96)
30	(66, 94.4 $\pm$ 16.584, 122)	(62, 92.867 $\pm$ 14.603, 112)
31	(70, 96.839 $\pm$ 18.815, 124)	(68, 94.645 $\pm$ 14.063, 116)
32	(62, 99.625 $\pm$ 20.152, 134)	(70, 97.688 $\pm$ 20.689, 132)

National Nuclear Security Administration under Contract No. 89233218CNA000001. Thanks to Carleton Coffrin for helpful discussions. This work has been assigned the LANL technical report number LA-UR-23-20504.

## APPENDIX A: CONTRACTED 4-CLIQUE GRAPHS ON ZEPHYR

Figure 11 shows the contracted 4-clique graphs for a logical Zephyr  $Z_{16}$  graph with no hardware defects.

## APPENDIX B: CONTRACTED 4-CLIQUE PEGASUS MINOR EMBEDDING CHAIN LENGTHS

Table II details chain length statistics of the computed contracted 4-clique random minor embeddings.

- 
- [1] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, Quantum computation by adiabatic evolution, [arXiv:quant-ph/0001106](https://arxiv.org/abs/quant-ph/0001106) (2000).
- [2] S. Morita and H. Nishimori, Mathematical foundation of quantum annealing, *J. Math. Phys.* **49**, 125210 (2008).
- [3] A. Das and B. K. Chakrabarti, Colloquium: Quantum annealing and analog quantum computation, *Rev. Mod. Phys.* **80**, 1061 (2008).
- [4] P. Hauke, H. G. Katzgraber, W. Lechner, H. Nishimori, and W. D. Oliver, Perspectives of quantum annealing: Methods and implementations, *Rep. Progr. Phys.* **83**, 054401 (2020).
- [5] T. Lanting, *et al.*, Entanglement in a quantum annealing processor, *Phys. Rev. X* **4**, 021041 (2014).
- [6] G. E. Santoro and E. Tosatti, Optimization using quantum mechanics: Quantum annealing through adiabatic evolution, *J. Phys. A: Math. General* **39**, R393 (2006).
- [7] M. W. Johnson, M. H. Amin, S. Gildert, T. Lanting, F. Hamze, N. Dickson, R. Harris, A. J. Berkley, J. Johansson, and P. Bunyk, *et al.*, Quantum annealing with manufactured spins, *Nature* **473**, 194 (2011).
- [8] S. Boixo, T. Albash, F. M. Spedalieri, N. Chancellor, and D. A. Lidar, Experimental signature of programmable quantum annealing, *Nat. Commun.* **4**, 1 (2013).
- [9] D. Venturelli, S. Mandrà, S. Knysh, B. O’Gorman, R. Biswas, and V. Smelyanskiy, Quantum optimization of fully connected spin glasses, *Phys. Rev. X* **5**, 031040 (2015).
- [10] Z. Tabi, K. H. El-Safty, Z. Kallus, P. Haga, T. Kozsik, A. Glos, and Z. Zimboras, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2020), p. 56.
- [11] B. Tasseff, T. Albash, Z. Morrell, M. Vuffray, A. Y. Lokhov, S. Misra, and C. Coffrin, On the emerging potential of quantum annealing hardware for combinatorial optimization, [arXiv:2210.04291](https://arxiv.org/abs/2210.04291) (2022).
- [12] H. M. Bauza and D. A. Lidar, Scaling advantage in approximate optimization with quantum annealing, [arXiv:2401.07184](https://arxiv.org/abs/2401.07184) (2024).
- [13] T. Albash and D. A. Lidar, Demonstration of a scaling advantage for a quantum annealer over simulated annealing, *Phys. Rev. X* **8**, 031016 (2018).
- [14] E. Pelofske, G. Hahn, and H. N. Djidjev, Parallel quantum annealing, *Sci. Rep.* **12** (2022).
- [15] A. D. King, *et al.*, Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets, *Nat. Commun.* **12**, 1113 (2021).
- [16] A. D. King, C. Nisoli, E. D. Dahl, G. Poulin-Lamarre, and A. Lopez-Bezanilla, Qubit spin ice, *Science* **373**, 576 (2021).
- [17] A. Lopez-Bezanilla, J. Raymond, K. Boothby, J. Carrasquilla, C. Nisoli, and A. D. King, Kagome qubit ice, *Nat. Commun.* **14**, 1105 (2023).
- [18] C. Klymko, B. D. Sullivan, and T. S. Humble, Adiabatic quantum programming: Minor embedding with hard faults, *Quantum Inf. Process.* **13**, 709 (2014).
- [19] D. E. Bernal, K. E. Booth, R. Dridi, H. Alghassi, S. Tayur, and D. Venturelli, in *International Conference on Integration of Constraint Programming, Artificial Intelligence, and Operations Research* (Springer, 2020), p. 112.
- [20] V. Choi, Minor-embedding in adiabatic quantum computation: II. Minor-universal graph design, *Quantum Inf. Process.* **10**, 343 (2011).
- [21] V. Choi, Minor-embedding in adiabatic quantum computation: I. The parameter setting problem, *Quantum Inf. Process.* **7**, 193 (2008).
- [22] T. Boothby, A. D. King, and A. Roy, Fast clique minor generation in Chimera qubit connectivity graphs, *Quantum Inf. Process.* **15**, 495 (2016).
- [23] M. S. Konz, W. Lechner, H. G. Katzgraber, and M. Troyer, Embedding overhead scaling of optimization problems in quantum annealing, *PRX Quantum* **2**, 040322 (2021).
- [24] S. Zbinden, A. Bartschi, H. Djidjev, and S. Eidenbenz, in *International Conference on High Performance Computing* (Springer, 2020), p. 187.
- [25] A. Lucas, Hard combinatorial problems and minor embeddings on lattice graphs, *Quantum Inf. Process.* **18**, 1 (2019).
- [26] K. Boothby, P. Bunyk, J. Raymond, and A. Roy, Next-generation topology of D-Wave quantum processors, [arXiv:2003.00133](https://arxiv.org/abs/2003.00133) (2020).
- [27] N. Dattani, S. Szalay, and N. Chancellor, Pegasus: The second connectivity graph for large-scale quantum annealing hardware, [arXiv:1901.07636](https://arxiv.org/abs/1901.07636) (2019).
- [28] K. Boothby, *et al.*, Architectural considerations in the design of a third-generation superconducting quantum annealing processor, [arXiv:2108.02322](https://arxiv.org/abs/2108.02322) (2021).
- [29] E. Lobe, L. Schurmann, and T. Stollenwerk, Embedding of complete graphs in broken Chimera graphs, *Quantum Inf. Process.* **20**, 234 (2021).
- [30] `dwave_networkx.zephyr_graph`, [https://web.archive.org/web/20230608182151/https://docs.ocean.dwavesys.com/en/stable/docs\\_dnx/reference/generated/dwave\\_networkx.zephyr\\_graph.html](https://web.archive.org/web/20230608182151/https://docs.ocean.dwavesys.com/en/stable/docs_dnx/reference/generated/dwave_networkx.zephyr_graph.html).
- [31] K. L. Pudenz, T. Albash, and D. A. Lidar, Quantum annealing correction for random Ising problems, *Phys. Rev. A* **91**, 042302 (2015).
- [32] W. Vinci, T. Albash, G. Paz-Silva, I. Hen, and D. A. Lidar, Quantum annealing correction with minor embedding, *Phys. Rev. A* **92**, 042310 (2015).

- [33] W. Vinci, T. Albash, and D. A. Lidar, Nested quantum annealing correction, *npj Quantum Inf.* **2**, 1 (2016).
- [34] A. Mishra, T. Albash, and D. A. Lidar, Performance of two different quantum annealing correction codes, *Quantum Inf. Process.* **15**, 609 (2016).
- [35] K. L. Pudenz, T. Albash, and D. A. Lidar, Error-corrected quantum annealing with hundreds of qubits, *Nat. Commun.* **5**, 1 (2014).
- [36] S. Matsuura, H. Nishimori, W. Vinci, and D. A. Lidar, Nested quantum annealing correction at finite temperature: p-spin models, *Phys. Rev. A* **99**, 062307 (2019).
- [37] S. Matsuura, H. Nishimori, W. Vinci, T. Albash, and D. A. Lidar, Quantum-annealing correction at finite temperature: Ferromagnetic p-spin models, *Phys. Rev. A* **95**, 022308 (2017).
- [38] T. A. Caswell, *et al.*, matplotlib/matplotlib.
- [39] J. D. Hunter, Matplotlib: A 2D graphics environment, *Comput. Sci. Eng.* **9**, 90 (2007).
- [40] A. Hagberg, P. Swart, and D. S. Chult, *Exploring Network Structure, Dynamics, and Function Using NetworkX*, Tech. Rep. (Los Alamos National Lab. (LANL), Los Alamos, NM (United States), 2008).
- [41] D-Wave NetworkX, [https://web.archive.org/web/20230401000000\\*/https://github.com/dwavesystems/dwave-networkx](https://web.archive.org/web/20230401000000*/https://github.com/dwavesystems/dwave-networkx).
- [42] E. Pelofske, Dataset for 4-clique network minor embedding for quantum annealers (2023), <https://doi.org/10.5281/zenodo.7552776>.
- [43] Networkx connected\_components, [https://web.archive.org/web/20230907060839/https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.components.connected\\_components.html](https://web.archive.org/web/20230907060839/https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.components.connected_components.html).
- [44] J. Cai, W. G. Macready, and A. Roy, A practical heuristic for finding graph minors, *arXiv:1406.2741* (2014).
- [45] Minorminer, [https://web.archive.org/web/20230401000000\\*/https://github.com/dwavesystems/minorminer](https://web.archive.org/web/20230401000000*/https://github.com/dwavesystems/minorminer).
- [46] D. Sherrington and S. Kirkpatrick, Solvable model of a spin-glass, *Phys. Rev. Lett.* **35**, 1792 (1975).
- [47] K. Chern, K. Boothby, J. Raymond, P. Farré, and A. D. King, Tutorial: Calibration refinement in quantum annealing, *arXiv:2304.10352* (2023).
- [48] D-Wave Error-Correction Features, [https://web.archive.org/web/20240000000000\\*/https://docs.dwavesys.com/docs/latest/c\\_qpu\\_error\\_correction.html](https://web.archive.org/web/20240000000000*/https://docs.dwavesys.com/docs/latest/c_qpu_error_correction.html).