Check for updates

Tutorial: calibration refinement in quantum annealing

Kevin Chern, Kelly Boothby, Jack Raymond, Pau Farré and
Andrew D. King*

D-Wave, Burnaby, BC, Canada

Quantum annealing has emerged as a powerful platform for simulating and optimizing classical and quantum Ising models. Quantum annealers, like other quantum and/or analog computing devices, are susceptible to non-idealities including crosstalk, device variation, and environmental noise. Compensating for these effects through calibration refinement or "shimming" can significantly improve performance but often relies on *ad-hoc* methods that exploit symmetries in both the problem being solved and the quantum annealer itself. In this tutorial, we attempt to demystify these methods. We introduce methods for finding exploitable symmetries in Ising models and discuss how to use these symmetries to suppress unwanted bias. We work through several examples of increasing complexity and provide complete Python code. We include automated methods for two important tasks: finding copies of small subgraphs in the qubit connectivity graph and automatically finding symmetries of an Ising model via generalized graph automorphism. We conclude the tutorial by surveying additional methods, providing practical implementation tips, and discussing limitations and remedies of the calibration procedure. Code is available at: https://github.com/dwavesystems/shimming-tutorial.

1. Background

1.1. Introduction to quantum annealing

Quantum annealing (QA; Kadowaki and Nishimori, 1998; Johnson et al., 2011) is a computing approach that physically realizes a system of Ising spins in a transverse magnetic field. A common application of QA is to find low-energy spin states of the Ising problem Hamiltonian as follows:

$$\mathcal{H}_P = \sum_i h_i \sigma_i^z + \sum_{i<j} J_{ij} \sigma_i^z \sigma_j^z. \tag{1}$$

Here, $\{\sigma_i^z\}_{i=1}^N \in \{-1, 1\}^N$ is a set of Pauli z-operators, which can be thought of as a vector of classical ± 1 Ising spins; h_i denotes a longitudinal field (bias) on spin i, and J_{ij} (used interchangeably with $J_{i,j}$ depending on context) denotes a coupling (quadratic interaction) between spins i and j. Minimizing $\mathcal{H}_P$ is intractable, i.e., NP-hard (Barahona, 1982).

QA adds to $\mathcal{H}_P$ to an initial driving Hamiltonian as follows:

$$\mathcal{H}_D = -\sum_i \sigma_i^x. \tag{2}$$

The ground state of $\mathcal{H}_D$, which is a uniform quantum superposition of all classical states, is easy to prepare. QA guides a time-dependent Hamiltonian $\mathcal{H}(s)$ from $\mathcal{H}_D$ to $\mathcal{H}_P$ by linearly combining $\mathcal{H}_D$ and $\mathcal{H}_P$ as follows:

$$\mathcal{H}(s) = \Gamma(s)\mathcal{H}_D + \mathcal{J}(s)\mathcal{H}_P, \tag{3}$$

FIGURE 1
Annealing schedule for Hamiltonian (3) in a D-Wave™ Advantage™ processor. $\Gamma(s)$ and $\mathcal{J}(s)$ control the magnitude of quantum fluctuations and the Ising energy scale, respectively. These values vary from one processor to another.
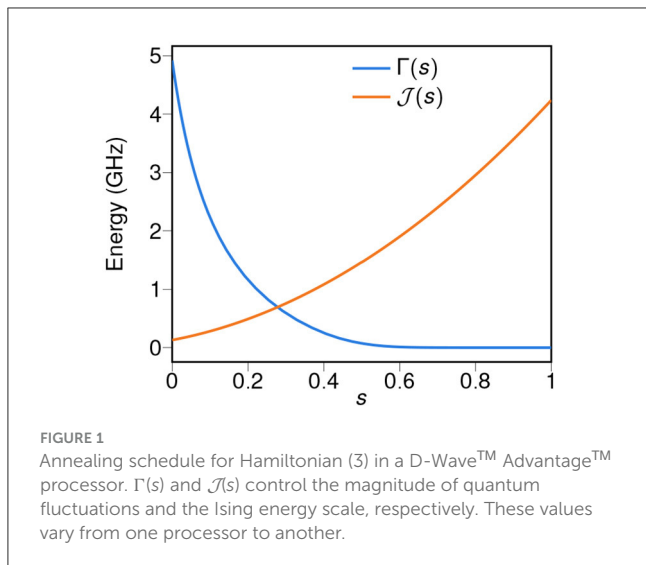
where s is a unitless annealing parameter ranging from 0 to 1. Unless stated, s is simply t/t_a: time normalized by annealing time. The functions $\Gamma(s)$ and $\mathcal{J}(s)$ define the *annealing schedule*: $\Gamma(s)$ decreases toward 0 as a function of s, and $\mathcal{J}(s)$ increases as a function of s; $\Gamma(0) \gg \mathcal{J}(0)$. Units are GHz, convertible to Joules by multiplication by $\hbar$ (reduced Planck constant). An example is shown in Figure 1.

1.2. Calibration imperfections and refinement

Quantum processing units (QPUs, in this case quantum annealers) are typically made available with a single one-size-fits-all calibration. Non-idealities in the calibration can arise from a number of sources. For example, small fluctuations in the magnetic environment can bias qubits in one direction or the other. Moreover, *crosstalk*, in which a Hamiltonian term, e.g., a programmed coupler J_{ij}, can cause an undesired perturbation in another Hamiltonian term corresponding to a physically nearby device, e.g., a bias field h_i.

In short, no calibration is perfect. Oftentimes, in-depth studies of a single system (Ising model) or ensemble of systems (e.g., a set of realizations of a spin-glass model) can be improved by suppressing crosstalk and other non-idealities. This is achieved by "shimming:" inferring statistical features of an ideal annealer and tuning the Hamiltonian to produce these features. An ideal annealer, in this study, is defined simply as one that respects symmetries in the Hamiltonian—each qubit behaves identically and each coupler behaves identically.

Variations on the methods described herein have been used in many studies (King et al., 2018, 2021a,b,c, 2022, 2023; Kairys et al., 2020; Nishimura et al., 2020). Often, when behavior of the system relies on precise maintenance of energy degeneracy between states, or energy splitting from the transverse field, the results are highly sensitive to these tunings. Particularly for the simulation of exotic magnetic phases, calibration refinement is an essential ingredient

of successful experiments. However, so far the discussion of these methods has mostly been relegated to Supplementary material. Here, our aim is to provide an accessible guide that will encourage the use of these powerful but simple methods.

Specific visual demonstrations of the benefit of these methods "in the wild" include:

- Frustrated 2D lattice, King et al. (2018), Extended Data Figure 7.
- Diluted ferromagnet, Nishimura et al. (2020), Figures 34–35.
- 1D quantum Ising chain, King et al. (2022), Supplementary Figures S3, S4.
- 3D quantum spin glasses, King et al. (2023), Supplementary Figures S8–S9.

The tutorial is organized as follows. In the remainder of this section, we introduce concepts that form the bases of the QPU calibration procedure. In Section 2, we illustrate the essence of our approach through a toy example. In Section 3, we extend the method and improve calibration efficiency by exploiting symmetries in a given model. In Section 4, we introduce a non-trivial system to demonstrate additional concepts useful for realistic applications. Collectively, these sections provide a comprehensive walkthrough of the calibration procedure. In Section 5, we survey additional methods for narrower use cases of the QPU. Finally, we provide practical tips and considerations in Section 6 and conclude the tutorial in Section 7.
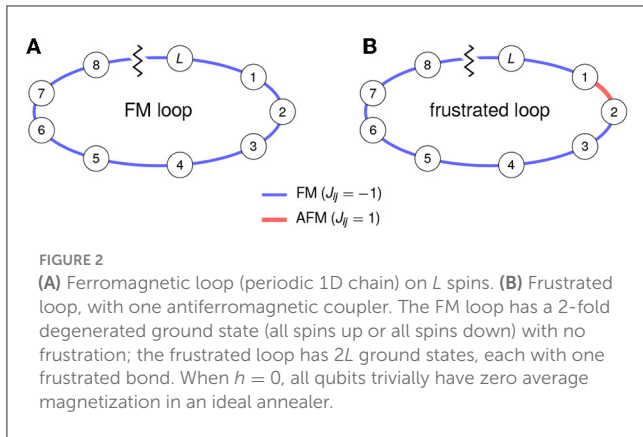
1.3. Inferring statistical features: qubit and coupler orbits

The approach described in this tutorial can be stated simply and generically. In theory, two observables of a QPU output are expected to be identical due to symmetries in the Ising model being studied. In experiment, they can differ systematically. We tune Hamiltonian terms to reduce these differences. In theory, symmetries in an Ising model admit identical expectation values of observables.[1] However, empirical averages over many realizations of these observables (from a QPU) may differ systematically.[2] We tune Hamiltonian terms to reduce the discrepancy between the expected value and observed averages. For example, given a Hamiltonian consisting of a single qubit with no bias, $\mathcal{H}_P = 0(s_1)$, we expect to observe a mean spin of 0 for s_1. However, this observed quantity may deviate from 0 systematically; we, thus, attempt to correct this deviation by perturbing the Hamiltonian.

In this study, we only consider one- and two-spin observables— spin magnetizations and frustration probabilities—in part because they can be fine-tuned easily using the available programmable terms in the QPU. A call to the QPU typically results in a number of classical samples, which we set to 100 for all examples. From these

---

1   An observable is a quantity that one can physically measure and observe. For example, the spin of a qubit.

2   Here, "differ systematically" refers to discrepancies between the expected value and the observed average as a result of biases in the physical system and not discrepancies due to finite samples.

FIGURE 2
(A) Ferromagnetic loop (periodic 1D chain) on L spins. **(B)** Frustrated loop, with one antiferromagnetic coupler. The FM loop has a 2-fold degenerated ground state (all spins up or all spins down) with no frustration; the frustrated loop has $2L$ ground states, each with one frustrated bond. When $h = 0$, all qubits trivially have zero average magnetization in an ideal annealer.

samples, we can compute a magnetization as follows:

$$m_i = \langle s_i \rangle \in [-1, 1] \tag{4}$$

For each spin s_i, a *frustration probability* is as follows:

$$f_{i,j} = \frac{1 + \text{sign}(J_{i,j})\langle s_i s_j \rangle}{2} \in [0, 1] \tag{5}$$

For each coupler $J_{i,j}$; $f_{i,j}$ is the observed probability of the coupler having a positive contribution to the energy in $\mathcal{H}_P$.

This raises the first question: how do we identify observables that should be identical in expectation? The answer is through symmetries of the Ising model under spin relabelling and gauge transformation (discussed below).[3] We understand and formalize these symmetries—and automate their detection—through *graph isomorphisms* (especially automorphisms) and generalizations (Godsil and Royle, 2001). We understand and formalize these symmetries—and automate their detection—through *graph isomorphisms* (especially automorphisms) and generalizations. We briefly introduce the concept of graph automorphisms and *orbits* below (see Godsil and Royle, 2001 for a more complete treatment). Notably, the symmetries we find and exploit here are a subset of all possible symmetries.

Given a graph $G = (V, E)$ with vertex and edge sets V, E, a graph automorphism is a mapping $\pi : V \mapsto V$ such that $(\pi(u), \pi(v)) \in E$ if and only if $(u, v) \in E$. Intuitively, a graph automorphism is an adjacency-preserving relabelling of vertices. Two vertices $u, v \in V$ are said to belong in the same *vertex orbit* if there exists an automorphism mapping u to v (or v to u). If $u, v \in V$ belong in the same vertex orbit, the edges incident to u or v also belong to the same *edge orbit*.

We now relate the definitions of graph automorphisms and orbits back to our goal of detecting and exploiting symmetries. These symmetries admit two types of equivalence relations on an Ising model $\mathcal{H}_P$: one on the qubits and the other on the couplers. We call the equivalence classes *qubit orbits* and *coupler orbits*, respectively. We use notation $\mathcal{O}(s_i)$ for a qubit orbit containing spin s_i, and $\mathcal{O}(s_i, s_j)$ for a coupler orbit containing coupler (s_i, s_j).

---

[3] A gauge transformation is also known as a *spin reversal transformation*, in which a subset of spins have their sign flipped.

We define them as having the following properties guaranteed by symmetry in an ideal annealer:

- All qubits in the same orbit have the same expected magnetization.
- All couplers in the same orbit have the same frustration probabilities.

Formally, a set $\mathcal{O}$ is said to be a qubit orbit if $s_i, s_j \in \mathcal{O}$, then $m_i = m_j$. Similarly, a set $\mathcal{O}$ is said to be a coupler orbit if $(s_i, s_j), (s_k, s_l) \in \mathcal{O}$, then $f_{i,j} = f_{k,l}$.

For example, consider the Hamiltonian $\mathcal{H}_P = hs_1 - hs_2 + hs_3$ for $h \neq 0$. The two *independent* spins s_1, s_3 can be trivially relabeled (permuted) by each other, thus the two qubits belong to the same qubit orbit; s_2 belongs in its own qubit orbit as it does not have the same magnetization as s_1, s_3. Now, let us consider the Hamiltonian $\mathcal{H}_{P,2} = \mathcal{H}_P + Js_1s_2 + Js_2s_3$ for $J \neq 0$. In this case, (s_1, s_2) and (s_2, s_3) exist in the same coupler orbit because s_1, s_3 can be swapped while preserving the couplers in $\mathcal{H}_{P,2}$; the permutation preserves adjacency structures. As a non-example, let us consider the Hamiltonian $\mathcal{H}_{P,3} = \mathcal{H}_P + Js_1s_2 - 2Js_2s_3$. In this case, (s_1, s_2) and (s_2, s_3) no longer exist in the same coupler orbit because swapping s_1, s_3 no longer preserves the couplers in $\mathcal{H}_{P,3}$.
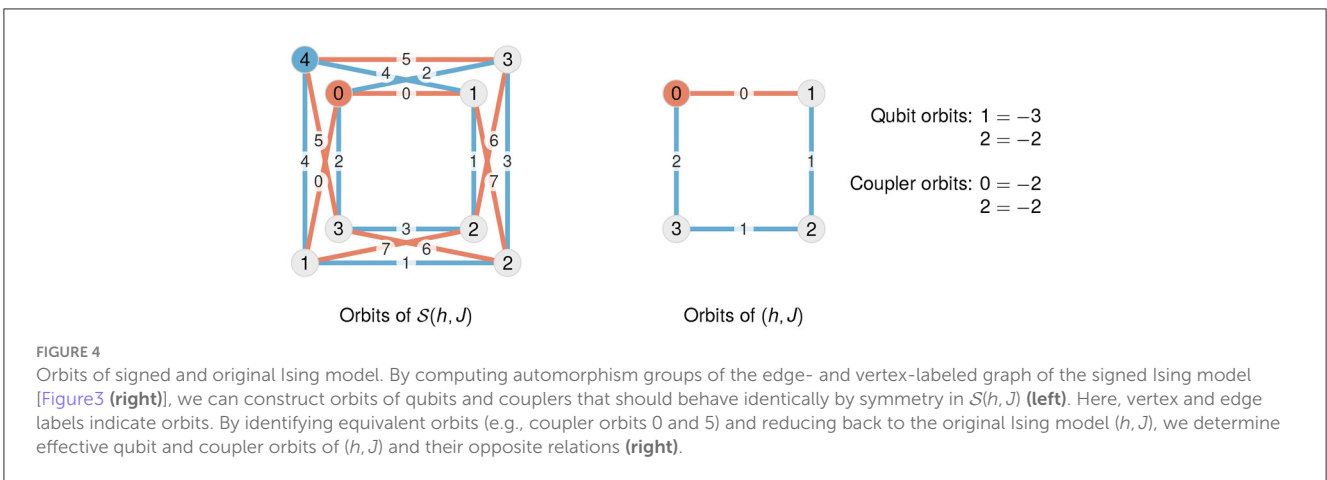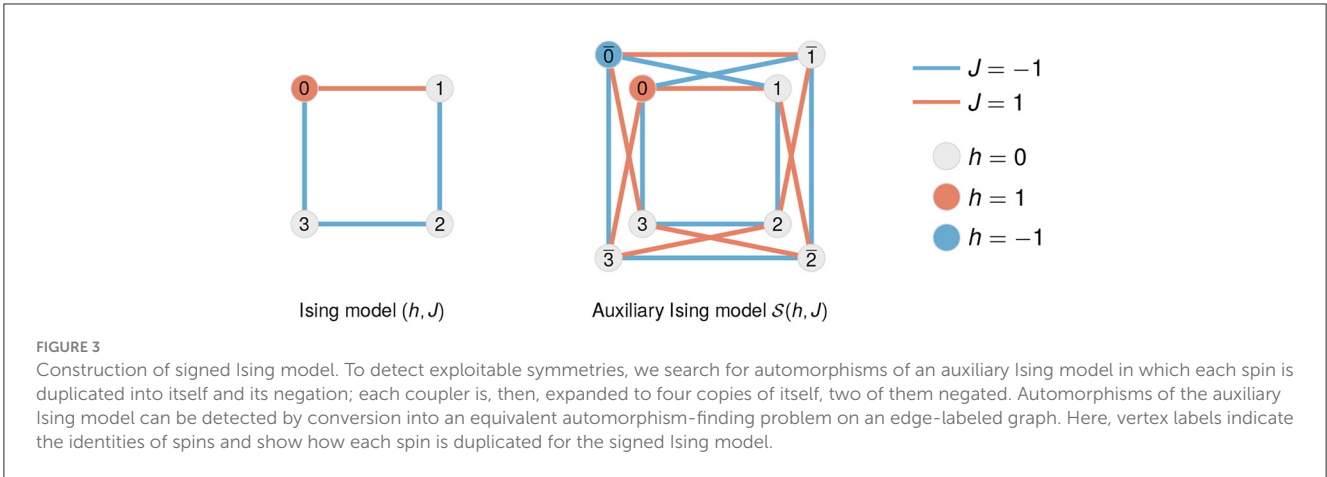
Due to spin-flip symmetries, or *spin reversal transformations (SRTs; described below)*, each qubit and coupler orbit can additionally have up to one non-empty orbit that is *opposite*.

- If qubit orbits $\mathcal{O}(s_i)$ and $\mathcal{O}(s_j)$ are opposite,

  - We write $\mathcal{O}(s_i) = -\mathcal{O}(s_j)$ and $-\mathcal{O}(s_i) = \mathcal{O}(s_j)$.
  - If $\mathcal{O}(s_i) = -\mathcal{O}(s_j)$, $h_i = -h_j$ and, in an ideal annealer, $m_i = -m_j$.

- If coupler orbits $\mathcal{O}(s_i, s_j)$ and $\mathcal{O}(s_k, s_\ell)$ are opposite,

  - We write $\mathcal{O}(s_i, s_j) = -\mathcal{O}(s_k, s_\ell)$ and $-\mathcal{O}(s_i, s_j) = \mathcal{O}(s_k, s_\ell)$.
  - $J_{i,j} = J_{k,\ell}$ and, in an ideal annealer, $f_{i,j} = f_{k,\ell}$. $J_{i,j} = -J_{k,\ell}$ and, in an ideal annealer, $f_{i,j} = f_{k,\ell}$.

An SRT, as its name suggests, flips the sign of a spin. For example, consider the Hamiltonian with a single qubit $\mathcal{H}_P = hs$. An SRT transforms on s yields an identical Hamiltonian $\mathcal{H}_P = -h\tilde{s}$ where $\tilde{s} = -s$. Similarly, for a Hamiltonian consisting of both biases and coupling terms such as $\mathcal{H}_P = h_1 s_1 + h_2 s_2 + J_{1,2}s_1s_2$, we can apply an SRT on one (or multiple) variable(s) to obtain $\mathcal{H}_P = -h_1\tilde{s}_1 + h_2 s_2 - J_{1,2}s_1s_2 = h_1 s_1 - h_2 s_2 - J_{1,2}s_1\tilde{s}_2 = -h_1\tilde{s}_1 - h_2\tilde{s}_2 + J_{1,2}\tilde{s}_1\tilde{s}_2$, where $\tilde{s}_1 = -s_1, \tilde{s}_2 = -s_2$.

In the earlier example $\mathcal{H}_P = hs_1 - hs_2 + hs_3$, qubit s_2 belongs to the orbit opposite of s_1, s_3's orbit. We will sometimes overload notation, conflating $\mathcal{O}(s_i)$ with $\mathcal{O}(i)$ and $\mathcal{O}(s_i, s_j)$ with $\mathcal{O}(i, j)$.

Qubit and coupler orbits are related to, but not identical to, automorphism orbits of an auxiliary graph. In particular, qubit and coupler orbits are not unique: putting each qubit and each coupler

FIGURE 3
Construction of signed Ising model. To detect exploitable symmetries, we search for automorphisms of an auxiliary Ising model in which each spin is duplicated into itself and its negation; each coupler is, then, expanded to four copies of itself, two of them negated. Automorphisms of the auxiliary Ising model can be detected by conversion into an equivalent automorphism-finding problem on an edge-labeled graph. Here, vertex labels indicate the identities of spins and show how each spin is duplicated for the signed Ising model.

FIGURE 4
Orbits of signed and original Ising model. By computing automorphism groups of the edge- and vertex-labeled graph of the signed Ising model [Figure3 **(right)**], we can construct orbits of qubits and couplers that should behave identically by symmetry in $\mathcal{S}(h, J)$ **(left)**. Here, vertex and edge labels indicate orbits. By identifying equivalent orbits (e.g., coupler orbits 0 and 5) and reducing back to the original Ising model (h, J), we determine effective qubit and coupler orbits of (h, J) and their opposite relations **(right)**.

in a separate orbit is sufficient to meet the definition but does not provide any useful information. We seek large orbits that satisfy the requirements.

Notably, in the commonly arising situation where $h_i = 0$ on all qubits, each qubit orbit is its own opposite, so all qubits have $m_i = 0$. The analogous situation does not exist for couplers because we do not consider symmetries between pairs of qubits with zero coupling between them. Two simple examples are shown in Figure 2: a frustrated loop and an unfrustrated loop. In each case, all qubits are expected to have magnetization and all couplers are expected to have the same probability of frustration, but this is less obvious in the frustrated case than in the ferromagnetic case. In each case, all qubits and couplers have, respectively, identical magnetization and frustration probabilities. The unfrustrated case is trivially true. The frustrated scenario is less obvious but can be verified by computing frustration probabilities for each edge.

Having defined qubit and coupler orbits, we now consider how to find them.

1.3.1. Automorphisms of the signed Ising model

We proposed a strategy for identifying exploitable symmetries for calibrating a QPU by finding qubit and coupler orbits. We

now introduce a method for identifying these orbits and begin by defining the *signed Ising model*.

Let (h, J) denote an Ising model with fields $h = \{h_i | v_i \in V\}$ and $J = \{J_{i,j} | e_{i,j} \in E\}$, with an underlying graph $G = (V, E)$ with vertex and edge sets V and E. We construct a signed Ising model $\mathcal{S}(h, J)$ as follows:

- For each spin $v_i \in V$, $\mathcal{S}(h, J)$ has two spins v_i and $\bar{v}_i$, with fields h_i and $-h_i$ respectively.
- For each coupler $(v_i, v_j) \in E$, $\mathcal{S}(h, J)$ has four couplers: two couplers (v_i, v_j) and $(\bar{v}_i, \bar{v}_j)$ with coupling $J_{i,j}$ and two couplers $(\bar{v}_i, v_j)$ and $(v_i, \bar{v}_j)$ with coupling $-J_{i,j}$.

Informally, we simply replace each spin with two: itself and its negation and replace each coupler with four couplers with appropriate parity-based sign flipping. Figure 3 shows an example of this construction applied to a four-spin Ising model.

Our aim is to find large qubit and coupler orbits for (h, J), and we will begin by finding the automorphism group of $\mathcal{S}(h, J)$, which can be considered as a vertex- and edge-labeled graph. Our aim is to find qubit and coupler orbits. Because automorphisms of the underlying graph of an Ising model are symmetries of the Ising model, we can generate qubit and coupler orbits by considering the graph symmetries alone. In other words, finding

graph automorphisms of the Ising model effectively give us qubit and coupler orbits. We can stop here and perform calibration based on these qubit and coupler orbits extracted from these symmetries (spin relabelling symmetries). However, we can similarly extract and exploit spin-flip symmetries by finding the automorphisms of the graph of $\mathcal{S}(h, J)$. These additional automorphisms give rise to qubit and coupler orbits opposite to the original. Intuitively, $\mathcal{S}(h, J)$ enumerates and concatenates all SRT configurations to the original Ising model. As a consequence, by finding its automorphisms, we are able to further identify symmetries due to SRTs. In other words, if a negated vertex is in the same orbit as a non-negated vertex, they exhibit symmetries through an SRT. In short, the automorphism group of $\mathcal{S}(h, J)$ naturally generates one equivalence relation defining qubit orbits and another equivalence relation defining coupler orbits (see Figure 4).[4] Our orbits of $\mathcal{S}(h, J)$ immediately give us orbits of (h, J) and constructed by simply discarding the qubits and couplers that do not exist in (h, J).

There is more usable information held in the orbits of $\mathcal{S}(h, J)$. First, we can combine coupler orbits of $\mathcal{S}(h, J)$ such that for each coupler $e_{i,j} \in E$, $(\bar{v}_i, v_j)$ and $(v_i, \bar{v}_j)$ are in the same orbit, and (v_i, v_j) and $(\bar{v}_i, \bar{v}_j)$ are in the same orbit. Second, we can, then, easily derive opposite orbits: $\mathcal{O}(v_i) = -\mathcal{O}(\bar{v}_i)$, and $\mathcal{O}(v_i, v_j) = -\mathcal{O}(\bar{v}_i, \bar{v}_j)$.

These orbits are already very useful, but we can combine some to make even larger orbits. As demonstrated in the example in Figure 3, in $\mathcal{S}(h, J)$, the couplers between pairs $(\bar{v}_i, v_j)$ and $(v_i, \bar{v}_j)$ are not necessarily automorphic. However, they are clearly equivalent under a flip of all spins. Thus, we combine the coupler orbits containing these two couplers. Likewise, the same applies to couplers between pairs $(\bar{v}_i, \bar{v}_j)$ and (v_i, v_j). This is all demonstrated in the accompanying code example0_1_orbits.py and shown in Figure 4.

We now consider how to exploit orbits to improve performance in quantum annealers, building up a set of tools in the following worked examples.

2. Worked example: ferromagnetic loop

Code reference: example1*.py.

For our first example of calibration refinement, we study the ferromagnetic (FM) loop (Figure 5) in which each coupling $J_{1,2} = J_{2,3} = \cdots = J_{N,1}$ is equal and each field h_i is zero. In this case, by rotation, it is obvious that all qubits are in the same orbit and all couplers are in the same orbit. Furthermore, the orbit containing all qubits is its own opposite. Thus, we will perform two refinements. First, we will balance each qubit at zero magnetization $m_i \approx 0$. Second, we will balance the couplings so that each coupler is frustrated with approximately equal probability.

Since the FM loop has no frustrated bonds in the ground state, the latter condition is only interesting if we sample excited states.

---

4 Since the automorphism-finding code nauty, McKay and Piperno (2014) only handles vertex-labeled graphs and not edge-labeled graphs, and we need to construct a vertex-labeled graph G'' from $\mathcal{S}(h, J)$, which gives us the appropriate automorphism group.

FIGURE 5
Ferromagnetic loop.
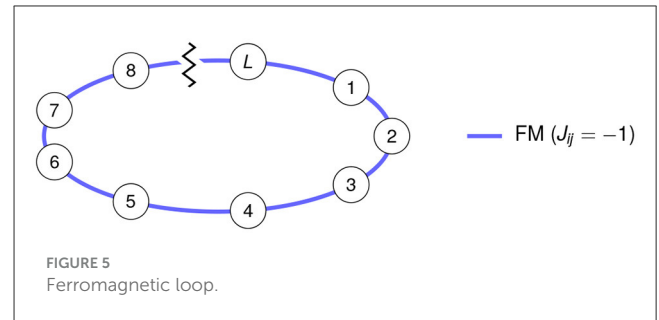
To ensure abundant excitations, we study a reasonably long loop with weak couplings: $L = 64$ and $J_{ij} = -0.2$.

2.1. Finding multiple embeddings of a small Ising model

Code reference: embed_loops.py.

The first task is to find a copy of the FM loop in the qubit connectivity graph A_{QPU} of the QPU being used. This is an *embedding*—a mapping of spins of an Ising model to qubits in a QPU. In an Advantage processor, a 64-qubit loop can be embedded many times on disjoint sets of qubits, so we can run many copies in parallel for a richer and larger set of measurements.

To find these embeddings, we use the Glasgow graph solver (McCreesh et al., 2020), which has been incorporated into the embedding finding module minorminer (D-Wave, 2023). To make the embedding search faster, we raster-scan across 2×2 blocks of unit cells in the QPU's Pegasus graph (Boothby et al., 2020) and then greedily construct a large set of non-intersecting embeddings. The file embed_loops.py provides a code example that finds multiple disjoint copies of a 64-qubit loop in A_{QPU}.
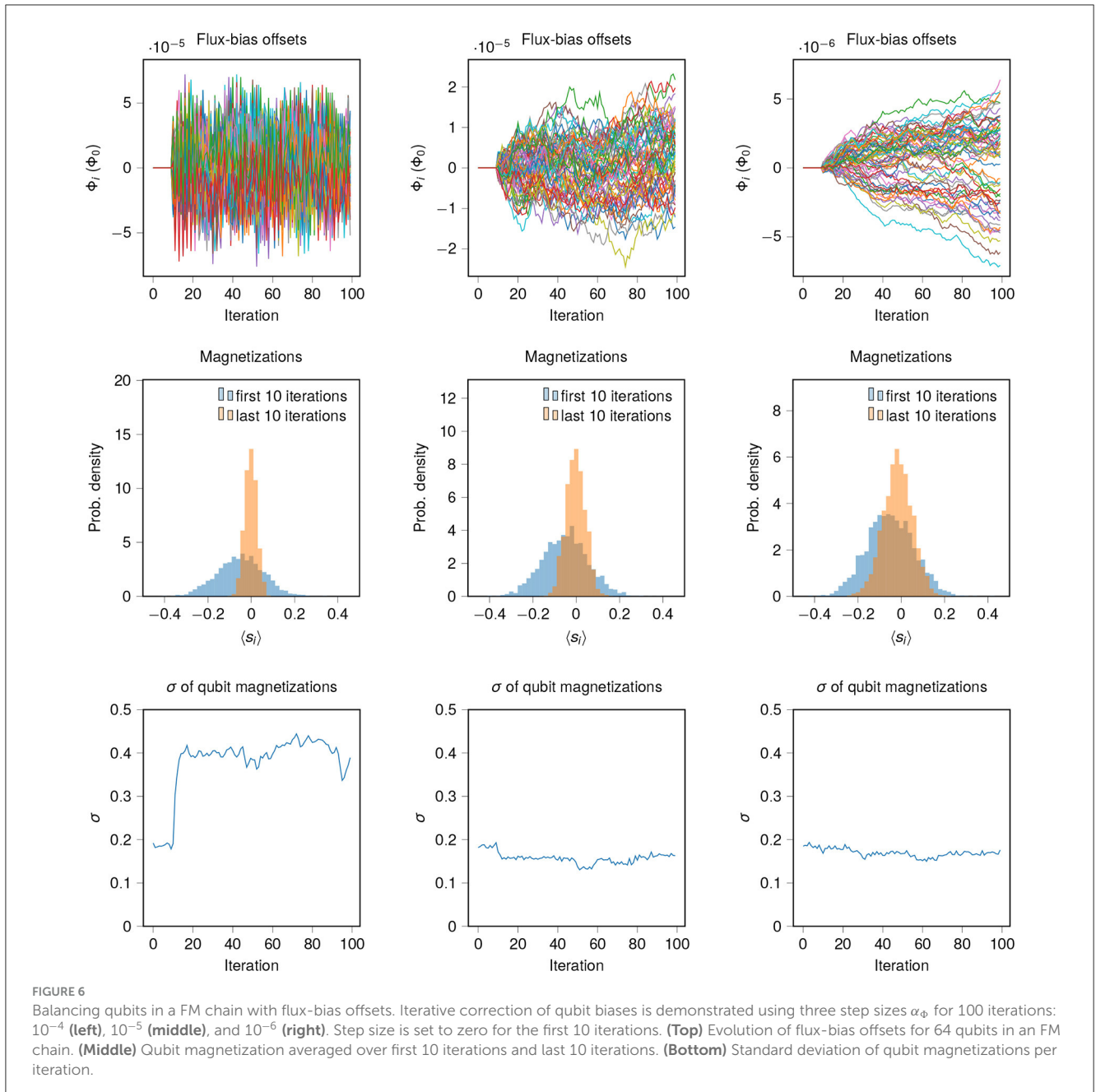
2.2. Balancing qubits at zero

Code reference: example1_1_fm_loop_balancing.py.

We will use simple parameters for the experiment, running $1 \, \mu s$ anneals forward anneals (where s increases linearly in time as t/t_a) and drawing 100 samples for each QPU call. We set auto_scale=False to ensure that the QPU will not automatically magnify the energy scale.

In D-Wave's annealing QPUs, each qubit s_i can be biased toward -1 or $+1$ in two ways: first, with a programmable longitudinal field h_i as in Equation 1; second, with a programmable flux-bias offset (FBO) Φ_i (Harris et al., 2009; D-Wave, 2022). In the quantum annealing Hamiltonian (3), the bias conferred by the h_i term is scaled by $\mathcal{J}(s)$, meaning that it changes as a function of s. The FBO Φ_i, in contrast, confers a constant bias that is independent of s. We prefer to mitigate biases using FBOs, in part, because they are programmed independently of h_i.

We employ an iterative gradient descent method for minimizing $|m_i|$ with a step size α_Φ. For a given iteration, we consider the observed magnetization $m_i = \langle s_i \rangle$. If $m_i < 0$, we adjust the FBO to push s_i toward $+1$; if $m_i > 0$, we adjust the FBO to push s_i toward -1. This is done by updating as follows:
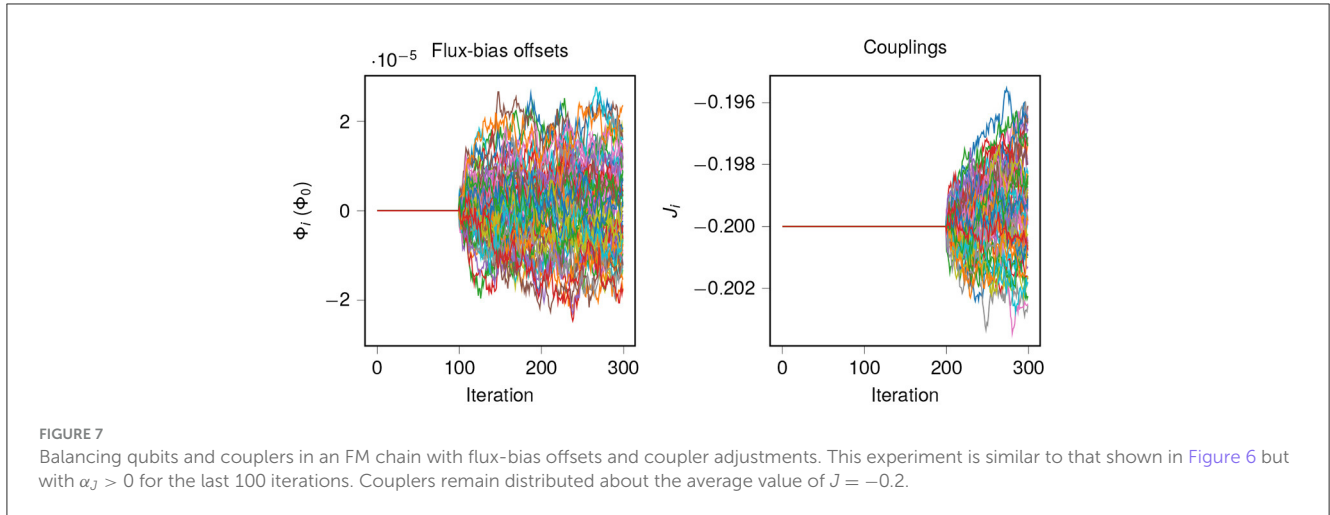
FIGURE 6
Balancing qubits in a FM chain with flux-bias offsets. Iterative correction of qubit biases is demonstrated using three step sizes α_Φ for 100 iterations: 10^{-4} **(left)**, 10^{-5} **(middle)**, and 10^{-6} **(right)**. Step size is set to zero for the first 10 iterations. **(Top)** Evolution of flux-bias offsets for 64 qubits in an FM chain. **(Middle)** Qubit magnetization averaged over first 10 iterations and last 10 iterations. **(Bottom)** Standard deviation of qubit magnetizations per iteration.

$$\Phi_i \leftarrow \Phi_i - \alpha_\Phi(m_i - \bar{m}) \qquad (6)$$

where $\bar{m}$ is the average observed magnetization across all qubits. In this case, we can simply replace $\bar{m}$ with 0 since $h_i = 0$ for all qubits.

The choice of a step size α_Φ has a strong influence on the convergence of the calibration procedure. In Figure 6, we show the resulting FBOs for a single copy of the 64-qubit chain, as well as magnetization statistics. We show experiments for three choices of α_Φ. One (flux 1×10^{-4}, in units of Φ_0) is too large and creates oscillations in Φ_i and m_i. One (1×10^{-6}) is too small and takes many iterations to converge. One (1×10^{-5}) is in between and performs well. The choice of step size is a common concern in gradient descent applications, and we will consider automatic tuning of α_Φ in a later section (Section 4.4). For best results, we should ensure:

- The calibration refinement appears to have converged to the vicinity of a fixed point.
- The parameters do not oscillate wildly.

When seeking evidence that qubit bias is improved by the FBOs, we should not just look at qubit statistics over a single QPU call, since fluctuations can be large. Rather, we should look at the average magnetization of a qubit over multiple calls, which indicates systematic bias. The middle row of Figure 6 shows the average magnetization of each qubit across the first and last 10 iterations. For each step size, the shim results in a significant improvement in variation of m_i from one qubit to another. However, the standard deviation among qubit magnetizations for *individual* iterations shows that the case $\alpha_\Phi = 1 \times 10^{-4}$ causes

FIGURE 7
Balancing qubits and couplers in an FM chain with flux-bias offsets and coupler adjustments. This experiment is similar to that shown in Figure 6 but with $\alpha_J > 0$ for the last 100 iterations. Couplers remain distributed about the average value of $J = -0.2$.

broad spreading of biases, so we need to be careful with our step sizes.

2.3. Balancing spin-spin correlations

Code reference:
example1_2_fm_loop_correlations.py.

Having balanced qubits at zero with linear terms with an FBO shim, we now address homogenizing the spin-spin correlations on adjacent qubits, which by symmetry should be equal for all coupled pairs. The couplings $J_{i,i+1}$ are all nominally -0.2; we will fine-tune the couplings in the vicinity of this value. This is similar to how we fine-tuned the FBOs but with the added constraint that we do not change the average coupling.

For a given iteration, we take the observed probability $f_{i,i+1}$ of the coupler being frustrated as follows:

$$f_{i,i+1} = (1 + \text{sign}(J_{i,i+1})\langle s_i s_{i_1}\rangle)/2. \tag{7}$$

Let $\bar{f}$ denote the average frustration across all couplers in all disjoint embeddings of the chain, in general, we will compute $\bar{f}$ across all couplers in the union of a coupler's orbit and its opposite orbit. We, then, adjust couplings based on the *residual frustration* $f_{i,i+1} - \bar{f}$:

$$J_{i,i+1} \leftarrow J_{i,i+1}(1 + \alpha_J(f_{i,i+1} - \bar{f})). \tag{8}$$

Figure 7 shows data for the same experiment as Figure 6 but with the "coupler shim" added, with $\alpha_J = 0.001$. To show the effect of the two shims, we run 100 iterations with $\alpha_\Phi = 0$ and $\alpha_J = 0$, then 100 iterations with $\alpha_\Phi = 1 \times 10^{-5}$ and $\alpha_J = 0$, then 100 with $\alpha_\Phi = 1 \times 10^{-5}$ and $\alpha_J = 0.001$. In this particular case, the coupler shim is small but some systematic signals can be observed. We will show more impactful cases later in the tutorial.

3. Worked example: frustrated loop

Code reference: example2*.py.

FIGURE 8
Frustrated loop.
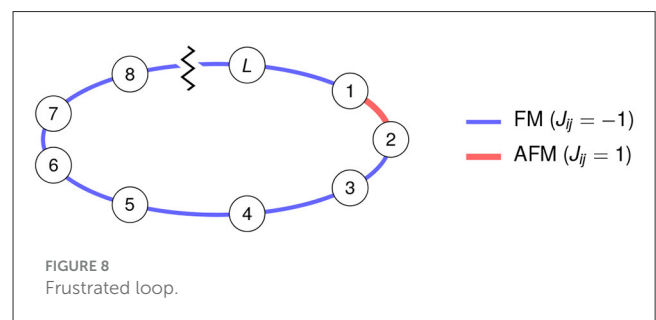
Taking the ferromagnetic loop considered in the previous example, the sign is flipped of a single coupler $J_{1,2}$ (Figure 8). It is again obvious that all spins should have zero average magnetization, since there is no symmetry-breaking field (i.e., $h_i = 0$ everywhere). Less obvious is the fact that we can have two coupler orbits: one containing all FM couplers and one containing the AFM coupler, and they are opposite. Consequently, every coupler should be frustrated with equal probability in an ideal annealer.
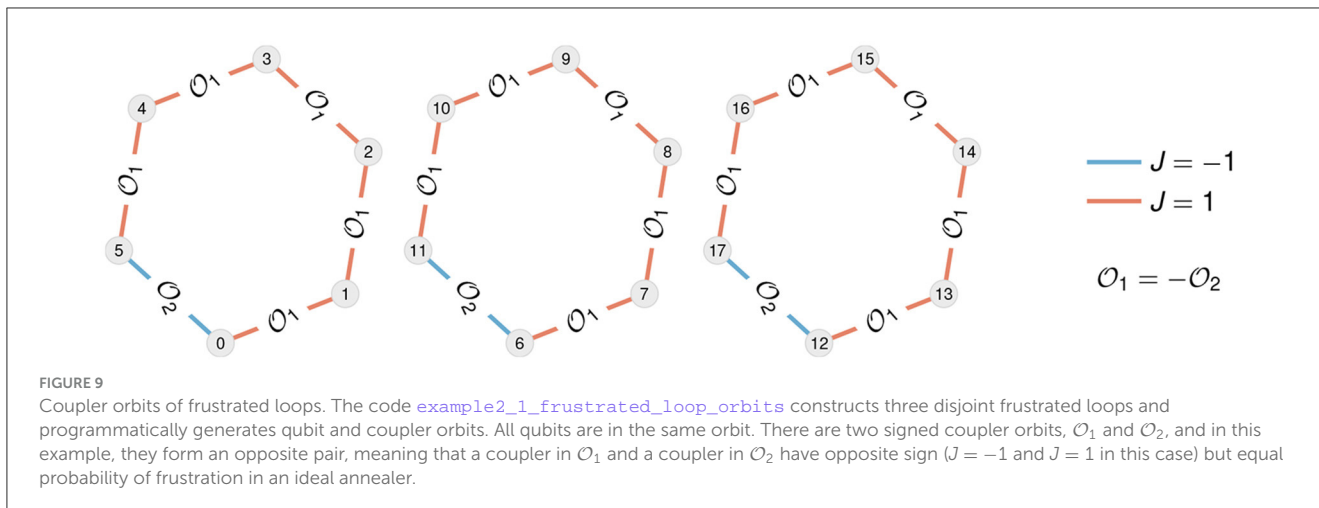
3.1. Finding orbits

Code reference:
example2_1_frustrated_loop_orbits.py.

We can derive this fact as follows. Flipping the sign of s_2, and the sign of both couplers incident to it, is a gauge transformation and, as such, will not change the probability of any coupler being frustrated in an ideal annealer. The result of this gauge transformation is again a frustrated loop with a single AFM coupler $J_{2,3}$; this is equivalent to the original loop by a cyclic shift of qubit labels. From this, we can infer that $J_{1,2}$ and $J_{2,3}$ should have the same frustration probability; repeating this argument tells us that all couplers should have the same frustration probability.

For more complicated examples, we would prefer to find such statistical identities programmatically as described in Section 1.3. We do this in the file

example2_1_frustrated_loop_orbits.py

FIGURE 9
Coupler orbits of frustrated loops. The code `example2_1_frustrated_loop_orbits` constructs three disjoint frustrated loops and programmatically generates qubit and coupler orbits. All qubits are in the same orbit. There are two signed coupler orbits, $\mathcal{O}_1$ and $\mathcal{O}_2$, and in this example, they form an opposite pair, meaning that a coupler in $\mathcal{O}_1$ and a coupler in $\mathcal{O}_2$ have opposite sign ($J = -1$ and $J = 1$ in this case) but equal probability of frustration in an ideal annealer.

by computing automorphisms of an auxiliary graph. The result is a mapping $\mathcal{O}$ of qubits and couplers to orbits. If spins s_i and s_j satisfy $\mathcal{O}(s_i) = \mathcal{O}(s_j)$, then in an ideal annealing experiment $m_i = m_j$. Similarly, if couplers $s_i s_j$ and $s_k s_\ell$ satisfy $\mathcal{O}(s_i s_j) = \mathcal{O}(s_k s_\ell)$, they have identical frustration probabilities $f_{i,j} = f_{k,\ell}$. The code also gives us a mapping of orbits to "opposite" orbits, such that if spins s_i and s_j are in opposing orbits, $m_i = -m_j$, and if couplers $s_i s_j$ and $s_k s_\ell$ are in opposing orbits, $f_{i,j} = f_{k,\ell}$ and $J_{i,j} = -J_{k,\ell}$.

Running the code on three disjoint copies of a frustrated six-qubit loop tells us that all AFM couplers are in one coupler orbit $\mathcal{O}_1 = \{s_i, s_j \mid \mathcal{O}(s_i s_j) = 1\}$, and all FM couplers are in its opposite, $\mathcal{O}_2 = -\mathcal{O}_1$ (see Figure 9).

We point out an obvious but useful fact: If we are using multiple embeddings of an Ising model, all copies of a given qubit are in the same orbit, and all copies of a given coupler are in the same orbit. Here, we use disjoint embeddings, but they need not be disjoint: the embeddings could overlap, and be annealed in separate calls to the QPU.

3.1.1. Shimming

We can now approach the frustrated loop similarly to the unfrustrated loop: all qubits should have average magnetization zero, and all couplers should be frustrated with the same probability. Again, tuning FBOs and individual couplings helps to reduce bias in the system. This example shows how to exploit orbits for our shim.

There is one detail worth pointing out. In Equations 6, 8, the terms $\bar{m}$ and $\bar{f}$ can be computed as averages over an orbit. If we are dealing with opposing qubit orbits $\mathcal{O}_q$ and $-\mathcal{O}_q$, we can simply use $\bar{m} = 0$, as we do in the first example. For opposing coupler orbits $\mathcal{O}_c$ and $-\mathcal{O}_c$, we can compute $\bar{f}$ across the union of the two orbits. In this case, that means that $\bar{f}$ is the average frustration probability across all couplers.

Figure 10 shows the results of shimming FBOs and couplings for 165 parallel embeddings of a 16-qubit frustrated loop, using nominal coupling strength $|J_i| = 0.9$. Here, both components of the shim show a marked improvement of statistical homogeneity. Taking moving means for 10 iterations at a time, we see that both σ_m (standard deviation of qubit magnetization) and σ_f (standard

deviation of coupler frustration probability) decrease as a result of turning on the FBO shim and the coupler shim, respectively.

3.1.2. Finding orbits of an arbitrary Ising model
Code reference: `example2_3_buckyball_orbits.py`.

Here we present an example of an Ising model that is read from a text file and run through our orbit-finding code. The user may want to edit this code to analyze other Ising models of interest.

Consider another antiferromagnetic Ising model ($J_{ij} = 1$) with a *Buckyball graph* as its underlying structure and no linear fields ($h_i = 0$). We apply the same methodology described in Section 1.3 to find its orbits. Figure 11 visualizes the Buckyball model with its orbits labeled by text, as well as its signed Ising counterpart with coupling values encoded by color.
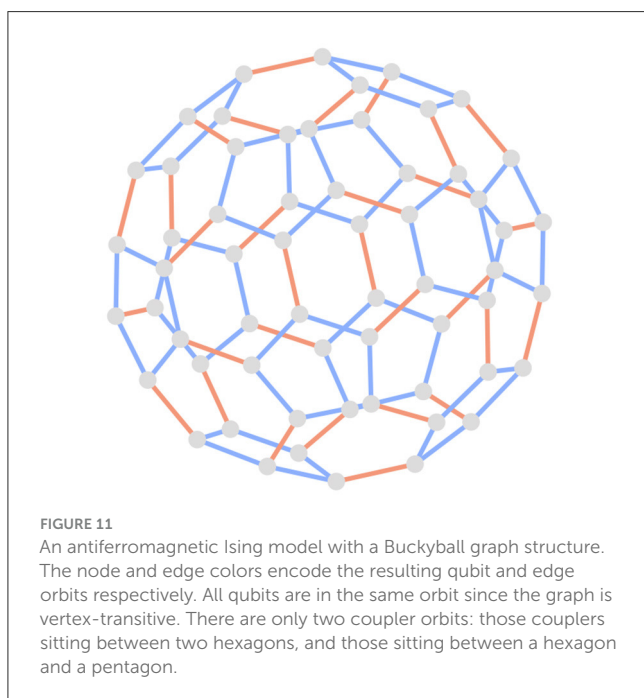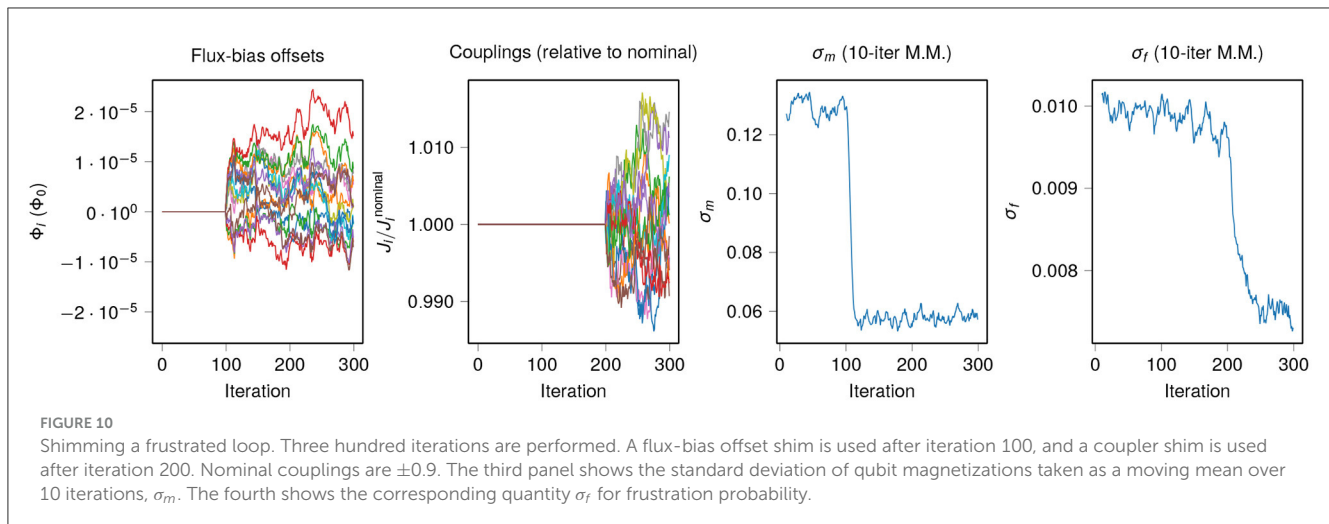
4. Worked example: triangular antiferromagnet

Code reference: `example3*.py`.

In the previous examples we demonstrated several key methods:

- Finding qubit and coupler orbits.
- Homogenizing magnetizations with FBOs.
- Homogenizing frustration by tuning couplers.

We can now apply these tools to a non-trivial system: the triangular antiferromagnet (TAFM; Figure 12). This is a classic example of a frustrated 2D spin system. Moreover, the addition of a transverse field to a TAFM leads to order-by-disorder at low temperature (Moessner and Sondhi, 2001; Isakov and Moessner, 2003). For this and other reasons, including qualitative similarity to real materials, the TAFM has been simulated extensively using quantum annealers (King et al., 2018, 2022). We will use it as an example to showcase several concepts in calibration refinement for quantum simulation:

- Truncating and renormalizing Hamiltonian terms.

FIGURE 10
Shimming a frustrated loop. Three hundred iterations are performed. A flux-bias offset shim is used after iteration 100, and a coupler shim is used after iteration 200. Nominal couplings are ±0.9. The third panel shows the standard deviation of qubit magnetizations taken as a moving mean over 10 iterations, σ_m. The fourth shows the corresponding quantity σ_f for frustration probability.

FIGURE 11
An antiferromagnetic Ising model with a Buckyball graph structure. The node and edge colors encode the resulting qubit and edge orbits respectively. All qubits are in the same orbit since the graph is vertex-transitive. There are only two coupler orbits: those couplers sitting between two hexagons, and those sitting between a hexagon and a pentagon.

- Simulating logical vs. embedded systems.
- Simulating an infinite system vs. faithfully simulating boundary conditions. When simulating an infinite system, we use the same geometry but suppress effects of any open boundaries, which otherwise cause statistics such as nearest-neighbor correlations to vary depending on distance from the boundary. To do this we determine our qubit and coupler orbits assuming an infinite lattice, instead of computing them from the finite lattice at hand.

4.1. Embedding as a square lattice

Code reference: example3_1_tafm_get_orbits.py.

In D-Wave's Advantage systems, we can minor-embed the TAFM using two-qubit FM chains. First, we will embed a 12×12 square lattice with cylindrical boundary conditions, then we ferromagnetically couple pairs of qubits with a strong coupling J_{FM}. The cylindrical boundaries are very helpful in providing rotational symmetries that we can exploit in our calibration refinement methods (as in the 1D chains already studied).

The provided code uses the Glasgow subgraph solver to find embeddings of the 12×12 square lattice, but note that this can take several hours. For larger square lattices, up to 32×32 or even larger depending on the location of inoperable qubits, one can inspect embeddings of smaller lattices and generalize the structure, since subgraph solvers are unlikely to be efficient at that size. We proceed with 10 disjoint 12×12 embeddings generated by the code.

In this example we will set AFM couplers to $J_{\text{AFM}} = 0.9$, and all FM couplers to $J_{\text{FM}} = -2 * J_{\text{AFM}}$. Since FM couplers are very rarely frustrated in this system, we will only shim the AFM couplers.

4.2. Annealing with and without shimming

Code reference: example3_2_tafm_forward_anneal.py.

As in the previous example, we will compare performance of three methods: no shim, FBO shim only, and FBO and coupler shims together. We perform 800 iterations, turning on the FBO shim after 100 iterations and the coupler shim after 300 iterations. Figure 13 shows data for this experiment, and we can see that as with the frustrated loop example, shimming improves statistical homogeneity of magnetizations and frustration. Note, however, that there is no appreciable impact on the average magnitude of the order parameter $\langle |\psi| \rangle$. This will change when we vary boundary conditions (see Figure 16).

4.3. Manipulating orbits to simulate an infinite system

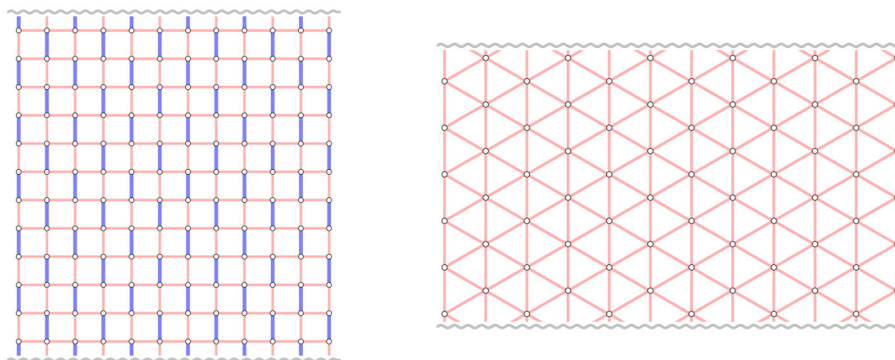**Code reference:** example3_2_tafm_forward_anneal.py.

FIGURE 12
A 12 × 12 square lattice with cylindrical boundary conditions (periodic top/bottom). Contracting two-qubit FM chains into single spins results in a triangular antiferromagnet.
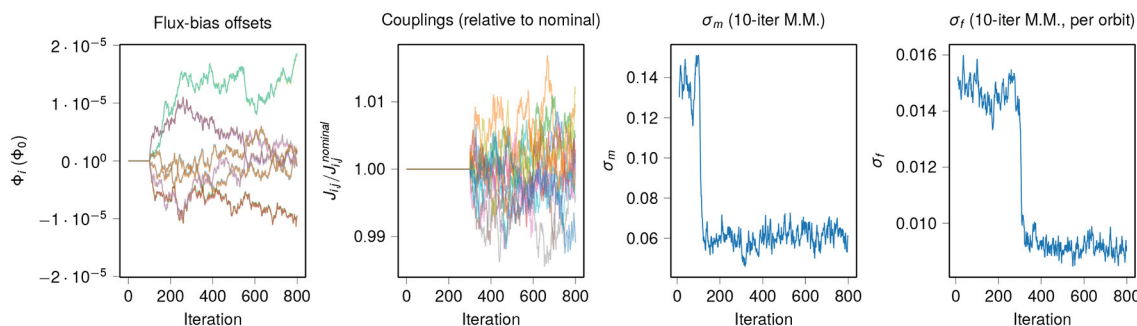

FIGURE 13
Shimming an embedded cylindrical triangular antiferromagnet. Eight hundred iterations are performed. A flux-bias offset shim is used after iteration 100, and a coupler shim is used after iteration 300. For clarity, we only show FBOs for 12 qubits, and couplings for 12 couplers in the same orbit. Standard deviation of frustration probabilities, σ_f, is computed for the couplers in each orbit, and the average over all orbits is taken.
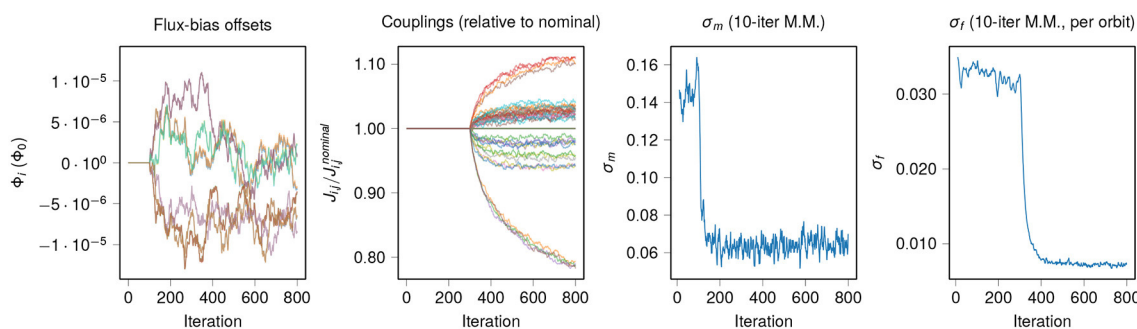

FIGURE 14
Shimming an isotropic, infinite triangular antiferromagnet. The experiment from Figure 13 is repeated, but with all AFM couplers placed in the same orbit. For clarity, we only show FBOs for 12 qubits, and every 5th coupling from the AFM orbit.

The shim shown in Figure 13 used coupler orbits for the square lattice with cylindrical boundaries, which are naturally different for couplers that are different distances from the boundary, or different orientations with respect to the boundary (and to FM chains). But what if we want to simulate, to the extent possible, an infinite TAFM? In that system, a coupler's probability of frustration is independent of its orientation and position, unlike in the square-lattice embedded system. We can simulate this case by putting all AFM couplers in one orbit, and all FM couplers in a second orbit, and proceeding as before. The coupler orbits no longer reflect the structure of the programmed Ising model, but rather the structure of the Ising model we wish to simulate.

Results for the "infinite triangular" shim are shown in Figure 14. This experiment is performed just like the previous one, but with

the parameter

```
shim['type']='triangular_infinite'
```

instead of

```
shim['type']='embedded_finite'.
```

The coupler shim deviates significantly from nominal values (note axis scale), and has not converged even after 500 iterations.

4.3.1. Truncating and renormalizing couplers

In this code example (and others) we use an important method in the coupler shim: truncation. Programmed couplings must be in the range $[-2, 1]$, so AFM couplers must remain < 1, which is $1.11 * J_{\text{AFM}}$. Therefore, when couplers go out of range, we truncate them to within the range. To avoid persistent shrinking of the couplings due to truncation, we renormalize to the correct average coupling value (0.9) before truncation—this prevents cumulative shrinkage over many iterations.

4.3.2. Better initial conditions

Looking at the data, we can see that the most reduced couplers are those on the boundary. This suggests that if we want to simulate the infinite TAFM, we should start with a thoughtful setting of couplers. In this case, setting the AFM couplers on the boundary to $J_{\text{AFM}}/2$ reduces the need to shim enormously. This makes sense, since doing so maximizes the ground-state degeneracy of the classical system, as previously noted (King et al., 2018).

This shim is shown in Figure 15. The experiment is performed just like the previous one, but with the parameter

```
param['halve_boundary_couplers']=True
```

instead of

```
param['halve_boundary_couplers']=False.
```

We can see that now, the coupler shim only deviates a few percent from nominal, at most.

4.3.3. Complex order parameter

Order in the TAFM can be characterized by a complex order parameter ψ, which we define now. Let $c: S \rightarrow \{0, 1, 2\}$ be a 3-coloring of the spins of the TAFM, mapping them onto three sublattices so that no two coupled spins are in the same sublattice (this coloring is unique, up to symmetries). Then for a spin state S we can define

$$\psi(S) = \frac{\sqrt{3}}{N} \sum_{\ell}^{N} \left(s_\ell e^{c_\ell 2\pi/3} \right), \tag{9}$$

where $c_i = c(s_i)$ and $i = \sqrt{-1}$. Due to symmetries among the sublattices arising from the cylindrical boundary condition, as well as up-down symmetry of spins since $h = 0$, we expect sixfold rotational symmetry (among other symmetries) in the distribution

of ψ in an ideal annealer. Thus ψ can serve as a good indicator of any biases in the system, as well as global ordering.

We can use ψ to compare the "embedded finite" shim and "triangular infinite" shim, as seen in Figure 16. Although we are simply forward-annealing the system, and therefore not sampling from the mid-anneal Hamiltonian, we expect the same characteristic ring histogram—without a peak near $\psi = 0$—that is seen in the quantum system (cf. King et al., 2018, Figure 3C). This is seen only after the "triangular infinite" shim. We mainly attribute this to the halving of the boundary couplings. In all cases, the shim improves the theoretically expected sixfold rotational symmetry of ψ.

4.4. Adaptive step sizes

Code reference: example3_2_tafm_forward_anneal.py.

It is often difficult or impractical to determine appropriate step sizes *a priori*. Here we demonstrate a simple method for adapting step sizes based on statistics of the shim. Note that due to noise in the QPU's surrounding environment, there is no well-defined asymptote or steady state for a shim. However, we act as though such a state exists: we expect high-frequency fluctuations in the environment to be small compared to low-frequency fluctuations and static cross-talk.

If the step size is sufficiently small and we are sufficiently close to the steady state, we can expect fluctuations of the Hamiltonian terms (FBOs, couplers, or fields) to behave like unbiased random walks. In an unbiased random walk with position $x(t)$ at time $t = 0, 1, \ldots$, the probability distribution of $x(t)$ approaches the normal distribution with mean 0 and variance t.

If the shim is far from the steady state and has a relatively small step size, the random walks will be biased in one direction, and thus the variance of fluctuations will grow superlinearly in t. Finally, if the step size is very large, then it will tend to overshoot the steady state, and oscillate. This leads to variance of fluctuations growing sublinearly in t. Thus we can periodically adjust the step size of a shim as follows, using a 20-iteration lookback and a tuning term $\varepsilon = 0.1$:
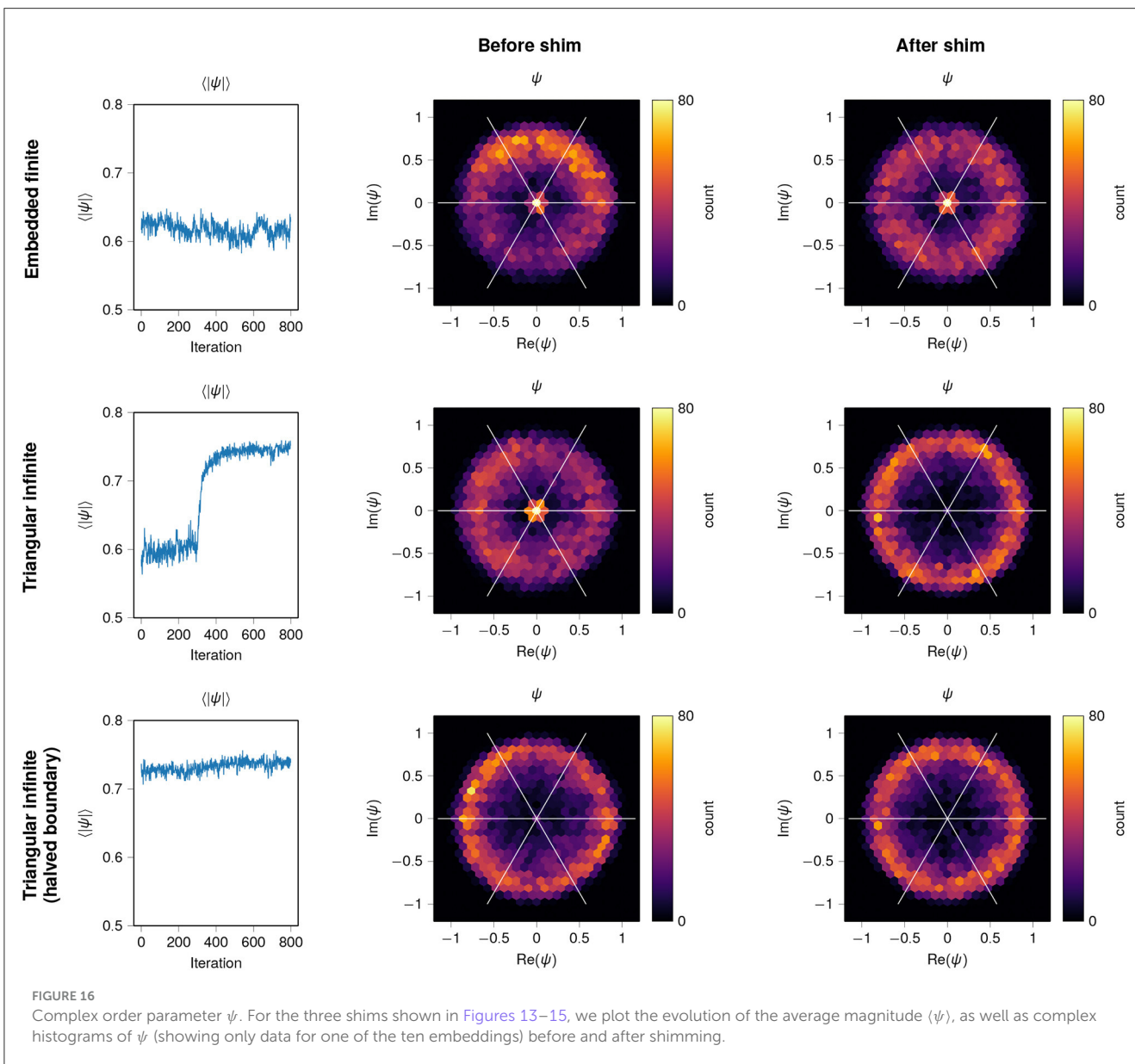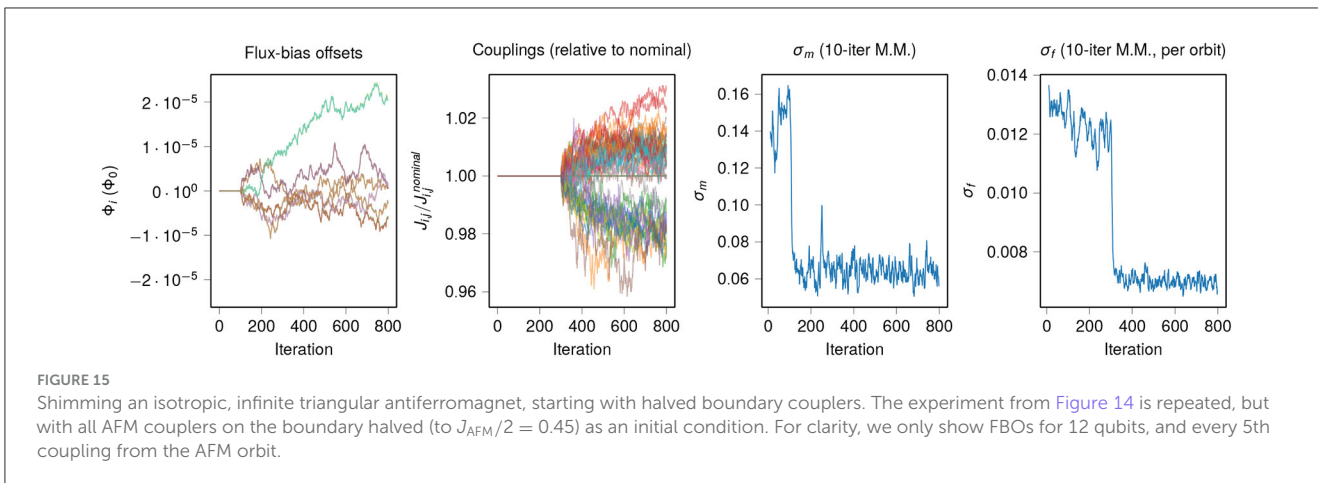
1. For $d \leq 20$, $x(t) - x(t - d)$ is the difference between the current shim value for a term (e.g. FBO) and the value d iterations previous. Let X_d be the set of all $x(t) - x(t - d)$ for all x being tuned.
2. Find a best-fit exponent b describing $\text{var}(X_d) \propto d^b$.
3. If $b > 1.1$, multiply the step size α by $1 + \varepsilon$.
4. If $b < 0.9$, divide the step size α by $1 + \varepsilon$.

In the example code example3_2_tafm_forward_anneal.py, this method is applied by setting

```
adaptive_step_size = True.
```

This check is done every iteration, but this is not necessary.

Adaptive step sizes are so far a largely unexplored research area, and various approaches could be taken. Using different step sizes for each orbit is certainly worth exploring; note in Figure 14 that different coupler orbits have hugely varying deviations from

FIGURE 15
Shimming an isotropic, infinite triangular antiferromagnet, starting with halved boundary couplers. The experiment from Figure 14 is repeated, but with all AFM couplers on the boundary halved (to $J_{AFM}/2 = 0.45$) as an initial condition. For clarity, we only show FBOs for 12 qubits, and every 5th coupling from the AFM orbit.

FIGURE 16
Complex order parameter ψ. For the three shims shown in Figures 13–15, we plot the evolution of the average magnitude $\langle |\psi| \rangle$, as well as complex histograms of ψ (showing only data for one of the ten embeddings) before and after shimming.

the mean. More general frameworks like "Adam" (Kingma and Ba, 2014) could also be useful in this context.

5. A survey of additional methods

We have provided detailed demonstrations and free-standing Python implementations for several worked examples. These cover the basics of calibration refinement. Here we discuss some additional methods that have been used successfully in recent works.

5.1. Shimming a system in a uniform magnetic field

Certain Ising models in a uniform magnetic field are of interest to physicists, and these have been simulated in quantum annealers both at equilibrium (Kairys et al., 2020) and out of equilibrium (King et al., 2021a). If we want to simulate an infinite system, we would ideally study a large system with no missing spins, and with fully periodic boundaries. However, this is often not possible, so we wish to make the magnetization m_i independent of the spin's position relative to the boundary (although it may depend on the spin's position in a unit cell of the lattice being simulated). In a typical experiment, we want to measure a system under an average field $\bar{h}$ for each value in an increasing set of equally spaced values $\{\bar{h}^{(1)}, \bar{h}^{(2)}, \ldots, \bar{h}^{(m)}\}$.

To deal with this, we can shim individual longitudinal field terms, h_i, such that all spins of a given type (i.e., in the same position of the unit cell) are in the same orbit. We can then shim all h_i terms for each simulated field magnitude $\bar{h}^{(k)}$ that we want to study, and denote each individual term $h_i^{(k)}$. After each iteration we renormalize the fields so the average value $\frac{1}{N} \sum_{i=1}^{N} h_i^{(k)}$ remains equal to $\bar{h}^{(k)}$ throughout the shim, perhaps with an adjustment arising from boundary spins.

To shim the case $\bar{h} = 0$, we use FBOs (as in the worked examples) instead of tuning h_i. To shim the case $\bar{h}^{(k)} = 0$, we use FBOs (as in the worked examples) to set a zero point instead of tuning $h_i^{(k)}$. In doing so, we can compensate for time-dependent flux drift, particularly when determining the location (in $\bar{h}$) of a phase transition.

We can additionally ensure that each h_i is a locally smooth function of $\bar{h}$ by adding a smoothing term. For example, if h_i has values h_i^- and h_i^+ for the next lower and higher values of $\bar{h}$ being simulated, we can make the adjustment.

We can additionally ensure that each sequence $\{h_i^{(1)}, \ldots, h_i^{(m)}\}$ is a smooth function of $\bar{h}$ by adding a smoothing term.[5] For example, we can add make an adjustment in two steps. First, set all

$$\tilde{h}_i^{(k)} \leftarrow (1 - \varepsilon)h_i^{(k)} + \epsilon \left(h_i^{(k-1)} + h_i^{(k+1)} \right)/2, \qquad (10)$$

for intermediate values of k and some smoothing constant $0 < \epsilon < 1$. Second, set all $h_i^{(k)} \leftarrow \tilde{h}_i^{(k)}$.

5.2. Shimming an Ising model with no symmetries

In King et al. (2021b), a qubit spin ice was implemented using a checkerboard Ising model. The system had open boundary conditions and missing spins due to inoperable qubits, so no geometric symmetries were available. However, due to the rich automorphism group of the qubit connectivity graph (ignoring unused qubits), it was possible to generate many distinct embeddings of the same system, using different mappings of qubits to spins. Therefore we could simulate a collection of distinct embeddings (in this case, 20) and shim in the same way we did in the worked examples. The only difference is that in the qubit spin ice example, the embeddings are not disjoint and therefore must be sampled from using separate calls to the QPU. However, once we have a set of samples from each embedding, we can analyze the data as though the embeddings are disjoint, whether or not this is actually the case. The benefit remains the same: by simulating with 20 distinct embeddings, we get qubit and coupler orbits of size at least 20.

5.3. Shimming a collection of random inputs

In King et al. (2023), shimming was used to study *spin-glass ensembles*—collections of random problems with certain parameters. As we have seen, we can spend hundreds of iterations shimming a single problem, and this becomes impractical when studying ensembles of thousands of instances.

The approach used was to exploit a common symmetry: all problems in the ensembles had $h = 0$. Shimming the couplers was abandoned as being impractical for such a large set of inputs. Shimming FBOs, however, is straightforward. By cycling through 300 spin-glass realizations using the same set of qubits and couplers, simulating each realization several times, it is possible to combine the work and arrive at a good set of FBOs that mitigates the majority of systematic offsets.

5.4. Shimming anneal offsets for fast anneals

As described in the Supplementary material to King et al. (2022), D-Wave quantum annealing processors have recently demonstrated the capacity to anneal much faster than currently generally available, at an anneal time of 10 nanoseconds or less (King et al., 2022, 2023). This speed exceeds the ability of the control electronics to synchronize the annealing lines (eight in the Advantage processor, four in D-Wave 2000Q™) satisfactorily. Therefore, frustration statistics can be used to infer which lines are out of sync with the others, and in which direction. Anneal offsets, which allow individual qubits to be annealed slightly ahead

---

5 A smooth function is desirable because a shim is a perturbation of the Hamiltonian intended to compensate for small non-idealities in the quantum annealer.

6. Additional tips

6.1. Limitations

The shimming techniques we have discussed in this tutorial are efficient to perform and are simple to justify theoretically. However, we note a couple of its limitations here. First, there exists a potential computational bottleneck in the full shimming paradigm. The method partly relies on identifying qubit and coupler orbits. Identifying these orbits can be reduced to the problem of identifying the automorphism group of a graph, which can usually be done quickly in practice even though no general polynomial-time algorithm is known. Second, shims inherently exhibit time-dependent fluctuations due to noise in the environment. These fluctuations tend to be smaller than the shim terms themselves, so do not significantly diminish the potential benefit of calibration refinement.

6.2. Making calibration refinement more efficient

As we have seen, shimming can take many iterations to converge. Naively repeating the process across many combinations of parameters (e.g., annealing time, energy scale, etc.) can be extremely time consuming. However, there are ways to improve the efficiency of the process. Here we outline some important things to bear in mind.

6.2.1. Adjustments are often continuous functions of other parameters

If we determine a set of adjustments for a given experiment, then slightly vary some parameters of the experiment, we can generally expect that the adjustments will not change much. For example, FBOs and coupling adjustments are expected to vary smoothly as functions of annealing time, energy scale, and various perturbations to the system (for example the ratio between FM and AFM couplers in an embedded triangular antiferromagnet). This assumption is natural outside the vicinity of a phase transition, and near a phase transition we adhere to the principle that we should not make discontinuous compensations to a simulator which is itself under smooth parametric modulation. An important example of a smoothly tuned parameter is the annealing parameter s, in cases where we simulate a system at $0 \ll s \ll 1$ (King et al., 2018, 2021a, 2022, etc.).

As an example of how this can help speed up a shim, if we double the annealing time, FBOs and coupling adjustments will remain relatively stable. Thus, rather than starting our shim anew from the nominal Hamiltonian, we can start from an adjusted Hamiltonian that was determined using similar parameters. One could go further than this, and extrapolate or interpolate based on multiple values.

6.2.2. Predictable adjustments should be programmed into the initial Hamiltonian

As shown in Figures 14, 15, starting with halved boundary couplings can immediately bring the couplings close to their converged values. If we are aware of such adjustments, using them as initial conditions can make shims converge far faster.

6.3. Damping shim terms

It is sometimes useful to gently encourage a shim to remain close to the nominal values, for example to prevent drifting Hamiltonian terms. This issue can be particularly important near a phase transition, where statistical fluctuations can be very large. Drift can be suppressed by adding a damping term to the shim. For example, we can set a damping constant $0 \leq \rho \leq 1$, and after every iteration we can move each coupler J_{ij} toward its nominal value $\hat{J}_{ij}$:

$$J_{ij} \rightarrow J_{ij} - \rho(J_{ij} - \hat{J}_{ij}). \qquad (11)$$

Doing this can discourage random fluctuations, but can also lead to under compensation of biases. It is only recommend to use damping when the shim is otherwise badly behaved. In practice, a suitable value for ρ is determined through trial and error, i.e., by assessing whether shims have converged (see Section 2 for discussion on shim convergence).

7. Conclusions

In this document we have presented several basic examples that introduce the value of calibration refinement or "shimming" in quantum annealing processors. These methods should be applied to any detailed study of quantum systems in a quantum annealer, and will generally provide a significant improvement to the results. Depending on the sensitivity of the system under study, these methods can mean the difference between an unsuccessful experiment and an extremely accurate simulation.

We have provided fully coded examples in Python, which should be easy to generalize and adapt. As part of these examples, we include methods for embedding many copies of a small Ising model in a large quantum annealing processor. This is a valuable and straightforward practice that can enormously improve both the quantity and the quality of results drawn from a single QPU programming.

Another important perspective, which has been introduced here for the first time, is the notion of constructing an auxiliary Ising model and using automorphisms of it to infer qubit and coupler orbits automatically. We encourage users to experiment with this method and report on any challenges or benefits found.

The examples in this document are written for use in an Advantage processor, but are not specific to that model, or even to D-Wave quantum annealers in general. These results may prove useful in diverse analog Ising machines, both quantum and classical.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

AK and JR devised the shimming methods. All authors contributed to the writing and revision of the paper and code examples.

Funding

Acknowledgments

Conflict of interest

KC, KB, JR, PF, and AK are employees of D-Wave Systems Inc. The reviewer AL declared a past co-authorship with the author AK to the handling editor.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fcomp.2023.1238988/full#supplementary-material

References

Barahona, F. (1982). On the computational complexity of Ising spin glass models. *J Phys A*. 15, 3241–3253.

Boothby, K., Bunyk, P., Raymond, J., and Roy, A. (2020). *Next-Generation Topology of D-Wave Quantum Processors*. Available online at: https://arxiv.org/abs/2003.00133

D-Wave (2022). *D-Wave System Documentation: "Flux-Bias Offsets"*. Available online at: https://docs.dwavesys.com/docs/latest/c_qpu_error_correction.html#qpu-error-fix-fbo (accessed January 3, 2023).

D-Wave (2023). *Minorminer*. Available online at: https://github.com/dwavesystems/minorminer (accessed January 23, 2023).

Godsil, C., and Royle, G. (2001). *Algebraic Graph Theory, Vol 207 of Graduate Texts in Mathematics*. New York, NY: Springer.

Harris, R., Lanting, T., Berkley, A. J., Johansson, J., Johnson, M. W., Bunyk, P., et al. (2009). Compound Josephson-junction coupler for flux qubits with minimal crosstalk. *Phys. Rev. B* 80, e052506. doi: 10.1103/PhysRevB.80.052506

Isakov, S. V., and Moessner, R. (2003). Interplay of quantum and thermal fluctuations in a frustrated magnet. *Phys. Rev. B* 68, 104409. doi: 10.1103/PhysRevB.68.104409

Johnson, M. W., Amin, M. H., Gildert, S., Lanting, T., Hamze, F., Dickson, N., et al. (2011). Quantum annealing with manufactured spins. *Nature* 473, 194–198. doi: 10.1038/nature10012

Kadowaki, T., and Nishimori, H. (1998). Quantum annealing in the transverse Ising model. *Phys. Rev. E* 58, 5355.

Kairys, P., King, A. D., Ozfidan, I., Boothby, K., Raymond, J., Banerjee, A., et al. (2020). Simulating the Shastry-Sutherland ising model using quantum annealing. *PRX Quantum* 1, e020320. doi: 10.1103/PRXQuantum.1.020320

King, A. D., Batista, C. D., Raymond, J., Lanting, T., Ozfidan, I., Poulin-Lamarre, G., et al. (2021a). Quantum annealing simulation of out-of-equilibrium magnetization in a spin-chain compound. *PRX Quantum* 2, e030317. doi: 10.1103/PRXQuantum.2.030317

King, A. D., Carrasquilla, J., Raymond, J., Ozfidan, I., Andriyash, E., Berkley, A., et al. (2018). Observation of topological phenomena in a programmable lattice of 1,800 qubits. *Nature* 560, 456–460. doi: 10.1038/s41586-018-0410-x

King, A. D., Nisoli, C., Dahl, E. D., Poulin-Lamarre, G., and Lopez-Bezanilla, A. (2021b). Qubit spin ice. *Science* 373, 576–580. doi: 10.1126/science.abe2824

King, A. D., Raymond, J., Lanting, T., Harris, R., Zucca, A., Altomare, F., et al. (2023). Quantum critical dynamics in a 5,000-qubit programmable spin glass. *Nature* 617, 61–66. doi: 10.1038/s41586-023-05867-2

King, A. D., Raymond, J., Lanting, T., Isakov, S. V., Mohseni, M., Poulin-Lamarre, G., et al. (2021c). Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets. *Nat. Commun.* 12, 1113. doi: 10.1038/s41467-021-20901-5

King, A. D., Suzuki, S., Raymond, J., Zucca, A., Lanting, T., Altomare, F., et al. (2022). Coherent quantum annealing in a programmable 2,000 qubit Ising chain. *Nat. Phys.* 18, 1324–1328. doi: 10.1038/s41567-022-01741-6

Kingma, D. P., and Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. Available online at: https://arxiv.org/abs/1412.6980

McCreesh, C., Prosser, P., and Trimble, J. (2020). "The Glasgow subgraph solver: using constraint programming to tackle hard subgraph isomorphism problem variants." in *Graph Transformation*, eds F. Gadducci and T. Kehrer (Cham: Springer International Publishing), 316–324.

McKay, B. D., and Piperno, A. (2014). Practical graph isomorphism, II. *J. Symbol. Comput.* 60, 94–112. doi: 10.1016/j.jsc.2013.09.003

Moessner, R., and Sondhi, S. L. (2001). Ising models of quantum frustration. *Phys. Rev. B* 63, 1–19. doi: 10.1103/PhysRevB.63.224401

Nishimura, K., Nishimori, H., and Katzgraber, H. G. (2020). Griffiths-McCoy singularity on the diluted Chimera graph: Monte Carlo simulations and experiments on quantum hardware. *Phys. Rev. A* 102, e042403. doi: 10.1103/PhysRevA.102.042403