



Technical Description of the D-Wave Quantum Processing Unit

USER MANUAL

2021-05-05

Overview

This document describes the D-Wave quantum processing unit (QPU) in detail. It defines terms, describes the QPU architecture and the quantum annealing process, and explains the effects of integrated control errors (ICE) that can affect results. Use this document in concert with the QPU properties of your particular system.

CONTACT

Corporate Headquarters
3033 Beta Ave
Burnaby, BC V5G 4M9
Canada
Tel. 604-630-1428

US Office
2650 E Bayshore Rd
Palo Alto, CA 94303

Email: info@dwavesys.com

www.dwavesys.com

Notice and Disclaimer

D-Wave Systems Inc. (D-Wave), its subsidiaries and affiliates, makes commercially reasonable efforts to ensure that the information in this document is accurate and up to date, but errors may occur. NONE OF D-WAVE SYSTEMS INC., its subsidiaries and affiliates, OR ANY OF ITS RESPECTIVE DIRECTORS, EMPLOYEES, AGENTS, OR OTHER REPRESENTATIVES WILL BE LIABLE FOR DAMAGES, CLAIMS, EXPENSES OR OTHER COSTS (INCLUDING WITHOUT LIMITATION LEGAL FEES) ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS DOCUMENT OR ANY INFORMATION CONTAINED OR REFERRED TO IN IT. THIS IS A COMPREHENSIVE LIMITATION OF LIABILITY THAT APPLIES TO ALL DAMAGES OF ANY KIND, INCLUDING (WITHOUT LIMITATION) COMPENSATORY, DIRECT, INDIRECT, EXEMPLARY, PUNITIVE AND CONSEQUENTIAL DAMAGES, LOSS OF PROGRAMS OR DATA, INCOME OR PROFIT, LOSS OR DAMAGE TO PROPERTY, AND CLAIMS OF THIRD PARTIES.

D-Wave reserves the right to alter this document and other referenced documents without notice from time to time and at its sole discretion. D-Wave reserves its intellectual property rights in and to this document and its proprietary technology, including copyright, trademark rights, industrial design rights, and patent rights. D-Wave trademarks used herein include D-Wave®, Leap™ quantum cloud service, Ocean™, Advantage™ quantum system, D-Wave 2000Q™, D-Wave 2X™, and the D-Wave logos (the D-Wave Marks). Other marks used in this document are the property of their respective owners. D-Wave does not grant any license, assignment, or other grant of interest in or to the copyright of this document, the D-Wave Marks, any other marks used in this document, or any other intellectual property rights used or referred to herein, except as D-Wave may expressly provide in a written agreement. This document may refer to other documents, including documents subject to the rights of third parties. Nothing in this document constitutes a grant by D-Wave of any license, assignment, or any other interest in the copyright or other intellectual property rights of such other documents. Any use of such other documents is subject to the rights of D-Wave and/or any applicable third parties in those documents.

Contents

1	About this Document	1
1.1	Intended Audience	1
1.2	Scope	1
1.3	Technical Terms	2
2	Background	3
2.1	Discrete Optimization	3
2.1.1	Ising Model	3
2.1.2	Formulation of the Ising Problem as a QUBO	4
2.2	Quantum Annealing with Ising Spins in a Transverse Field	4
2.3	Coupled rf-SQUID Qubits	5
2.4	Annealing Energy Functions	7
2.4.1	Energy Scales	7
2.4.2	Freezeout Points	8
2.5	Annealing Controls	12
2.5.1	Anneal Offsets	12
2.5.2	Variations on the Global Anneal Schedule	16
2.6	D-Wave QPU Operation	22
2.6.1	Programming Cycle	22
2.6.2	Anneal-Read Cycle	22
2.7	D-Wave QPU Architecture	23
2.7.1	Chimera	23
2.7.2	Pegasus	25
2.7.3	Virtual Graphs	26
3	ICE: Dynamic Ranges in h and J Values	33
3.1	Overview of ICE	33
3.2	Sources of ICE	35
3.2.1	Background Susceptibility (ICE 1)	35
3.2.2	Flux Noise of the Qubits (ICE 2)	37
3.2.3	DAC Quantization (ICE 3)	39
3.2.4	I/O System Effects (ICE 4)	39
3.2.5	Distribution of h Scale Across Qubits (ICE 5)	40
3.3	Measuring ICE	40
3.3.1	Measuring $1/f$ Noise	40
3.3.2	Using Two-Spin Systems to Measure ICE	41
3.4	Example of ICE Effects on Solution Quality	43
4	Other Factors Affecting Performance	45
4.1	Temperature	45
4.2	High-Energy Photon Flux	47
4.3	Readout Fidelity	47
4.4	Programming Errors	47
4.5	Spin-Bath Polarization Effect	48
5	Usage Guidelines	49
5.1	Decomposing Large Problems to Fit the QPU	49

5.2	Mapping General Graphs to the QPU Graph	49
5.3	Circumventing Precision Limits on h_i and $J_{i,j}$	50
5.4	Avoiding I/O Errors	50
5.5	Using Spin-Reversal Transforms	50
5.6	Adjusting the Problem Scale	51
6	Bibliography	53
A	Flux Noise and Quantum Annealing	55
A.1	Effects of Flux Noise	55
A.2	Drift Correction	56
	Bibliography	57

ABOUT THIS DOCUMENT

1.1 Intended Audience

This document is for users of the D-Wave™ quantum computer system who want to better understand and leverage the physical implementation of the quantum processing unit (QPU) architecture. It assumes that readers have a background in quantum annealing and are familiar with Ising problem formulations.

1.2 Scope

This document covers the following topics:

- Background on discrete optimization, quantum annealing, D-Wave QPU operation, and the QPU architecture.
- Integrated control errors (ICE): Dynamic ranges of h and J values and how they may affect results.
- Other factors that affect performance, including temperature, photon flux, readout fidelity, and programming problems.
- Some approaches for maximizing the performance of the QPU.
- Description of the effects of flux noise on the quantum annealing process; includes the procedure that D-Wave uses to correct for drift.

The values discussed in this document are representative properties for a D-Wave QPU. They are not product specifications.

This document does not provide programming instructions. For instructions on programming the system using D-Wave's open-source Ocean tools, see the [Ocean documentation](#).

1.3 Technical Terms

The table below defines some of the technical terms that are used throughout this document.

Table 1.1: Technical terms.

Term	Context	Definition
q_i	QPU	Qubit i for $i \in \{0, \dots, N-1\}$
N	QPU	Number of qubits in a QPU
s_i	Ising problems	Spin state at graph vertex i for $i \in \{1, \dots, N\}$; $s_i \in \{+1, -1\}$
\mathbf{s}	Ising problems	Vector of spin states (s_1, \dots, s_N)
$E(\mathbf{s})$	Ising problems	Energy at spin configuration \mathbf{s}
h_i	Ising problems	Linear coefficient (bias) on qubit i
$J_{i,j}$	Ising problems	Coupling between spins s_i and s_j
$J_{i,j} < 0$	Ising problems	Ferromagnetic coupling between spins s_i and s_j
$J_{i,j} > 0$	Ising problems	Antiferromagnetic coupling between spins s_i and s_j
$J_{i,j} = 0$	Ising problems	No coupling between spins s_i and s_j
x_i	QUBO problems	Binary state at graph vertex i for $i \in \{1, \dots, N\}$; $x_i \in \{0, 1\}$
\mathbf{x}	QUBO problems	Vector of binary states (x_1, \dots, x_N)
\mathbf{Q}	QUBO problems	Matrix of interactions between variables
$Q_{i,j}$	QUBO problems	Coupling between variables x_i and x_j
t	Anneal schedule	Current time during anneal
t_f	Anneal schedule	Total time for the anneal
s	Anneal schedule	Anneal fraction; abstract parameter ranging from 0 to 1. A linear anneal sets $s = t/t_f$.
$A(s)$	Anneal schedule	Tunneling energy at anneal fraction s
$B(s)$	Anneal schedule	Problem Hamiltonian energy at anneal fraction s

This chapter provides general background information on the discrete optimization of objective functions and introduces the D-Wave QPU operation and architecture to provide context for the remainder of the document.

2.1 Discrete Optimization

Discrete optimization, also known as combinatorial optimization, is the optimization of objective functions defined over search spaces consisting of a finite set of objects. This section focuses on two classes of optimization objectives:

- Ising problems defined over strings of -1/+1 symbols
- Quadratic unconstrained binary optimization (QUBO) problems defined over strings of 0/1 symbols

Note: This limited focus is not constraining: any problem in the NP complexity class may be translated into this form.

2.1.1 Ising Model

The D-Wave QPU can be viewed as a heuristic that minimizes Ising objective functions using a physically realized version of quantum annealing. The Ising objective function of N variables $\mathbf{s} = [s_1, \dots, s_N]$ where $s_i \in \{+1, -1\}$ is given by

$$E_{\text{ising}}(\mathbf{s}) = \sum_{i=1}^N h_i s_i + \sum_{i=1}^N \sum_{j=i+1}^N J_{i,j} s_i s_j. \quad (2.1)$$

Each variable s_i corresponds to a physical Ising spin that can be in a +1 or -1 state with a local applied field on each spin that causes it to prefer either the +1 or -1 state. The sign and magnitude of this preference—that is, the value of the local field—is denoted h_i . There may also be couplings between spins i and j such that the system prefers the pair of spins to be in either of the two sets defined by $s_i = s_j$ (ferromagnetic coupling) or $s_i = -s_j$ (antiferromagnetic coupling). The sign and magnitude of this preference is denoted $J_{i,j}$.

For example, in a single-spin system where h_1 has a large positive value, then the lowest energy occurs when s_1 is -1. Similarly, for two spins, s_1 and s_2 , with $h_1 = h_2 = 0$ and

$J_{1,2} = -1$, the system has two degenerate lowest energy states: one with $s_1 = -1$ and $s_2 = -1$, and the other with $s_1 = +1$ and $s_2 = +1$.

The D-Wave QPU is based on a physical lattice of qubits and couplers referred to as the *Chimera* or *Pegasus* architecture; see the [D-Wave QPU Architecture](#) section. The number of spins depends on the number of qubits in the QPU. The Advantage QPU, for example, contains over 5000 spins s_i . Each is coupled to a maximum of 15 other spins. The domains over which h and J can be chosen for a given system are available by querying the solver properties.

2.1.2 Formulation of the Ising Problem as a QUBO

In computer science, it is often more convenient to model with 0/1-valued variables than the $-1/1$ -valued Ising variables. Consider, for example, a maximum independent set (MIS) problem where the goal is to select the largest subset of vertices (the independent set) of a graph such that no pair of selected vertices is connected by an edge. For such a problem, it is more intuitive and succinct to define penalties to represent constraints using 0/1-valued variables.

QUBOs are typically expressed using matrix notation. For more information on expressing QUBOs, and for guidelines on converting between Ising and QUBO formulations, see [Getting Started with the D-Wave System](#).

Ising and QUBO problems are NP-hard [Bar1982]. NP-hardness holds even when the problem is restricted to the Chimera architecture and to the QPU-specific range of h_i and $J_{i,j}$ values. There are a number of tractable subclasses [Ber1996] of Ising/QUBO problems; for details, see [Kol2004], [Sch2009], or [Cou2009]. For information on the mapping of NP problems to the Chimera architecture, see the [D-Wave QPU Architecture](#) section.

2.2 Quantum Annealing with Ising Spins in a Transverse Field

The superconducting QPU at the heart of the D-Wave system, which operates at a temperature of approximately 12 mK, is a controllable, physical realization of the quantum Ising spin system in a transverse field. Each qubit and coupler on the QPU has several controls that are manipulated by individual on-QPU digital-to-analog converters (DACs); see [Bun2014] and [Joh2010]. Along with the DACs, a small number of analog control lines provide the time-dependent control required by the quantum Hamiltonian:

$$\mathcal{H}_{\text{ising}} = -\frac{A(s)}{2} \left(\sum_i \hat{\sigma}_x^{(i)} \right) + \frac{B(s)}{2} \left(\sum_i h_i \hat{\sigma}_z^{(i)} + \sum_{i>j} J_{i,j} \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \right) \quad (2.2)$$

where $\hat{\sigma}_{x,z}^{(i)}$ are Pauli matrices operating on a qubit q_i (the quantum one-dimensional Ising spin), and nonzero values of h_i and $J_{i,j}$ are limited to those available in the QPU graph; see the [D-Wave QPU Architecture](#) section.

The D-Wave QPU uses a method called *quantum annealing* (QA) to return low-energy spin states of E_{ising} for given inputs \mathbf{h} and \mathbf{J} . The quantum annealing process starts at time $t = 0$

with $A(0) \gg B(0)$, which leads to a trivial and easily initialized quantum ground state of the system where each spin, s_i , is in a delocalized combination of its classical states $s_i = \pm 1$. The system is then slowly annealed by decreasing A and increasing B until time t_f , which users specify via the *annealing_time* parameter. Figure 2.1 shows how A and B change over time. (For simplicity, we introduce a normalized anneal fraction, s , parameter ranging from 0 to 1.)

At the end of the computation, when $s = 1$ and $A(1) \ll B(1)$, the qubits have dephased to classical systems and the $\hat{\sigma}_z^{(i)}$ can be replaced by classical spin variables $s_i = \pm 1$. At this point, the system is described by the classical Ising spin system

$$E_{\text{ising}}(\mathbf{s}) = \sum_i h_i s_i + \sum_{i>j} J_{i,j} s_i s_j \quad (2.3)$$

such that the classical spin states represent a low-energy solution.

Note: If you are using the Leap™ quantum cloud service from D-Wave, you can find the $A(s)$ and $B(s)$ values for the QPUs in the cloud here: <https://support.dwavesys.com/hc/en-us/articles/360005267253-QPU-Specific-Anneal-Schedules>. If you have an installed system, contact D-Wave to obtain the values for your system.

2.3 Coupled rf-SQUID Qubits

The D-Wave QPU is built with a network of tunably coupled rf superconducting quantum-interference device (rf-SQUID) qubits; see [Har2010_2]. The physical Hamiltonian of this set of coupled rf-SQUIDs in the qubit approximation is

$$H = -\frac{1}{2} \sum_i \left[\Delta_q(\Phi_{\text{CCJJ}}(s)) \hat{\sigma}_x^{(i)} - 2h_i |I_p(\Phi_{\text{CCJJ}}(s))| \Phi_i^x(s) \hat{\sigma}_z^{(i)} \right] + \sum_{i>j} J_{i,j} M_{\text{AFM}} I_p(\Phi_{\text{CCJJ}}(s))^2 \hat{\sigma}_z^{(i)} \hat{\sigma}_z^{(j)} \quad (2.4)$$

where Δ_q is the energy difference between the two eigenstates of the rf-SQUID qubit with no external applied flux (the degeneracy point) where the eigenstates are $(|0\rangle \pm |1\rangle)/\sqrt{2}$. This energy difference captures the contribution of coherent tunneling between the two wells. I_p represents the magnitude of the current flowing in the body of the rf-SQUID loop; see Figure 2.2. M_{AFM} is the maximum mutual inductance generated by the couplers between the qubits (typically 2 pH), $\Phi_i^x(s)$ is an external flux applied to the qubits, and $\Phi_{\text{CCJJ}}(s)$ is an external flux applied to all qubit compound Josephson-junction structures to change the potential energy shape of the rf-SQUID qubit.

To map this system to the Ising spin in the transverse field Hamiltonian discussed in the *Quantum Annealing with Ising Spins in a Transverse Field* section, set $\Phi_i^x(s) = M_{\text{AFM}} |I_p(s)|$. Thus, as $\Phi_{\text{CCJJ}}(s)$ changes during the anneal, $\Phi_i^x(s)$ changes as required to keep the relative energy ratio between the h and J terms constant. In particular, the physical flux applied to the qubit to implement a fixed h value increases as the anneal progresses. Then, the mapping to the Ising Hamiltonian becomes:

$$\begin{aligned} A(s) &= \Delta_q(\Phi_{\text{CCJJ}}(s)) \\ B(s) &= 2M_{\text{AFM}} |I_p(\Phi_{\text{CCJJ}}(s))|^2 \end{aligned} \quad (2.5)$$

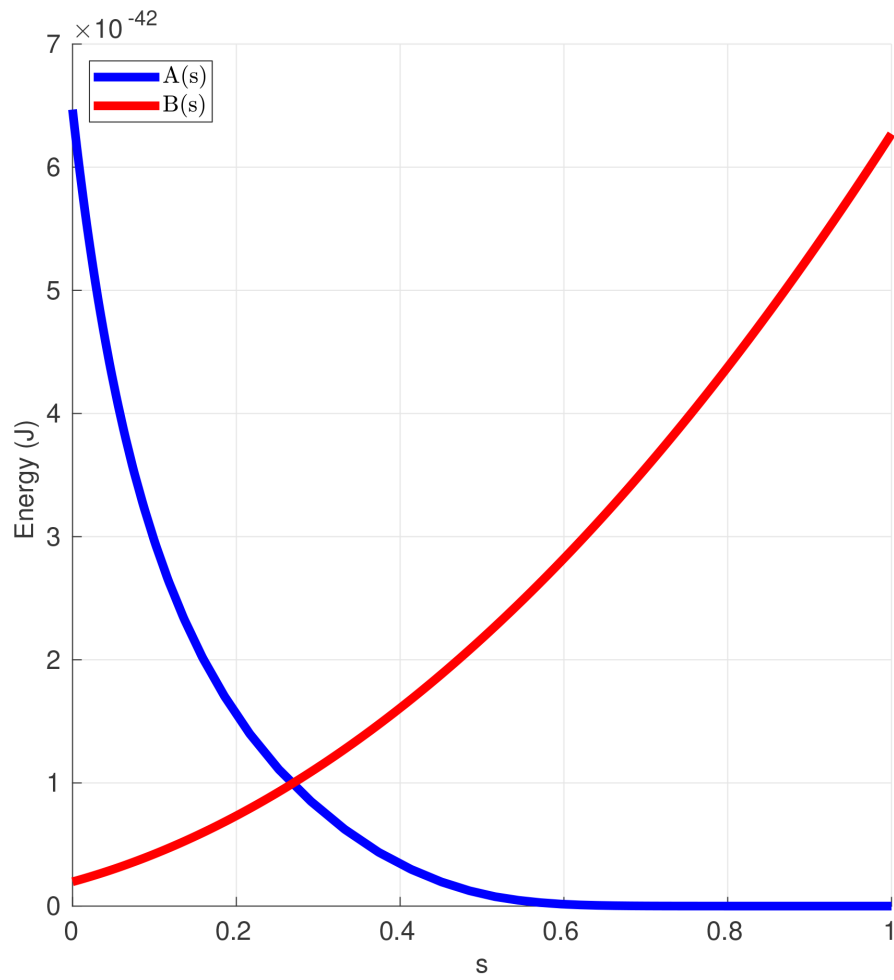


Figure 2.1: Annealing functions $A(s)$, $B(s)$. Annealing begins at $s = 0$ with $A(s) \gg B(s)$ and ends at $s = 1$ with $A(s) \ll B(s)$. Data shown are representative of Advantage systems.

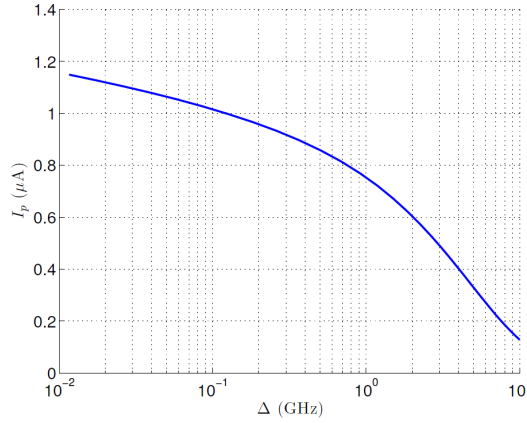


Figure 2.2: Typical I_p vs Δ_q . This relationship is fixed by the physical parameters of the rf-SQUID qubit. Data shown are representative of D-Wave 2X systems.

The relationship between $\Delta_q(\Phi_{\text{CCJJ}})$ and $I_p(\Phi_{\text{CCJJ}})$ is fixed by the physical parameters of the rf-SQUID qubit. Changing the applied Φ_{CCJJ} moves the rf-SQUID qubit along the curve shown in Figure 2.2.

For simplicity, we introduce a normalized annealing bias,

$$c(s) = \frac{\Phi_{\text{CCJJ}}(s) - \Phi_{\text{CCJJ}}^{\text{initial}}}{\Phi_{\text{CCJJ}}^{\text{final}} - \Phi_{\text{CCJJ}}^{\text{initial}}}, \quad (2.6)$$

where $\Phi_{\text{CCJJ}}^{\text{initial}}$ and $\Phi_{\text{CCJJ}}^{\text{final}}$ are the values of Φ_{CCJJ} at $s = 0$ and $s = 1$, respectively ($c(0) = 0$ and $c(1) = 1$).

The signal $c(s)$ is provided by an external room temperature current source. The time-dependence of this bias signal is chosen to produce a linear growth in time of the persistent current flowing in the rf-SQUID flux qubits, $I_p(s)$. Because $B(s) = 2M_{\text{AFM}}I_p(s)^2$, the problem energy scale grows quadratically in time (as seen in Figure 2.1).

2.4 Annealing Energy Functions

This section describes quantum annealing energy functions and freezeout points on the D-Wave QPU and explains how tunneling energy changes as rf-SQUID qubits are coupled.

2.4.1 Energy Scales

Two energy scales are relevant during quantum annealing, $A(s)$ and $B(s)$:

- $A(s)$ represents the transverse, or tunneling, energy. It equals Δ_q , as defined in the *Coupled rf-SQUID Qubits* section.
- $B(s)$ is the energy applied to the problem Hamiltonian. It equals $2M_{\text{AFM}}I_p(s)^2$, where M_{AFM} represents the maximum available mutual inductance achievable between pairs of flux qubit bodies.

Energy scales $A(s)$ and $B(s)$ change during the quantum annealing process. In particular, a single, global, time-dependent bias controls the trajectory of A and B . At any intermediate value of s , the ratio $A(s)/B(s)$ is fixed. We can choose the trajectory of one of A or B with time. The standard annealing schedule, $s = t/t_f$, results in $I_p(s)$ growing linearly with time, producing a quadratic growth in $B(s)$. Typical values of $A(s)$ and $B(s)$ are shown in [Figure 2.1](#).

2.4.2 Freezeout Points

$A(s)$ sets the time scale for qubit dynamics. As annealing progresses, $A(s)$, and therefore this time scale, decreases. When the dynamics of the complex Ising spin system become slow compared to t_f , the network is *frozen*—that is, the spin state does not change appreciably as the Ising spin Hamiltonian evolves. While in general each Ising spin problem has different dynamics, it is instructive to analyze a simple system consisting of clusters of uniformly coupled qubits. These clusters of coupled qubits are called *logical qubits*.¹

Networks of logical qubits *freeze out* at different points in the annealing process, depending on several factors, including:

- Number of qubits in the network
- Coupling strengths between the qubits
- Overall time scale of the anneal, t_f

In general, freezeout points move earlier in s for larger logical qubit sizes, for more strongly coupled logical qubits, and for smaller annealing time t_f . [Figure 2.3](#) and [Figure 2.4](#) show representative freezeout points for several qubit network sizes. A network of logical qubits is created by coupling multiple qubits to a single central qubit using $J = +1$; see the [Using Two-Spin Systems to Measure ICE](#) section for more details.

Measuring² I_p at the freezeout point of various-sized logical qubits results in [Figure 2.5](#). The figure shows that less annealing time and larger clusters move the freezeout point earlier in the anneal, where I_p is lower.

¹ For a logical qubit made of 3 qubits, for example, the relevant multiqubit states might be $|\uparrow\uparrow\uparrow\rangle$ and $|\downarrow\downarrow\downarrow\rangle$.

² While absolute calibration is difficult through the SAPI interfaces, if temperature is used as an absolute calibration factor, the population statistics of a simple logical qubit transition can approximately determine the relative value of I_p at freezeout.

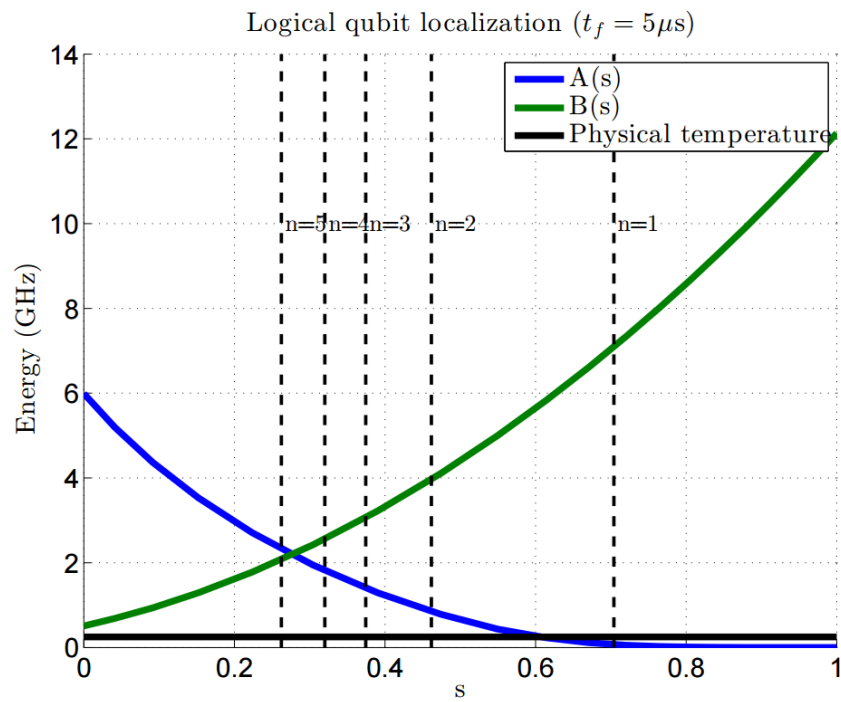


Figure 2.3: Representative annealing schedules with freezeout points for several logical qubit sizes and $t_f = 5\mu s$. The dashed lines show localization points for a $N = 1$ through $N = 5$ logical qubit clusters from right to left, respectively. Data shown are representative of D-Wave 2X systems.

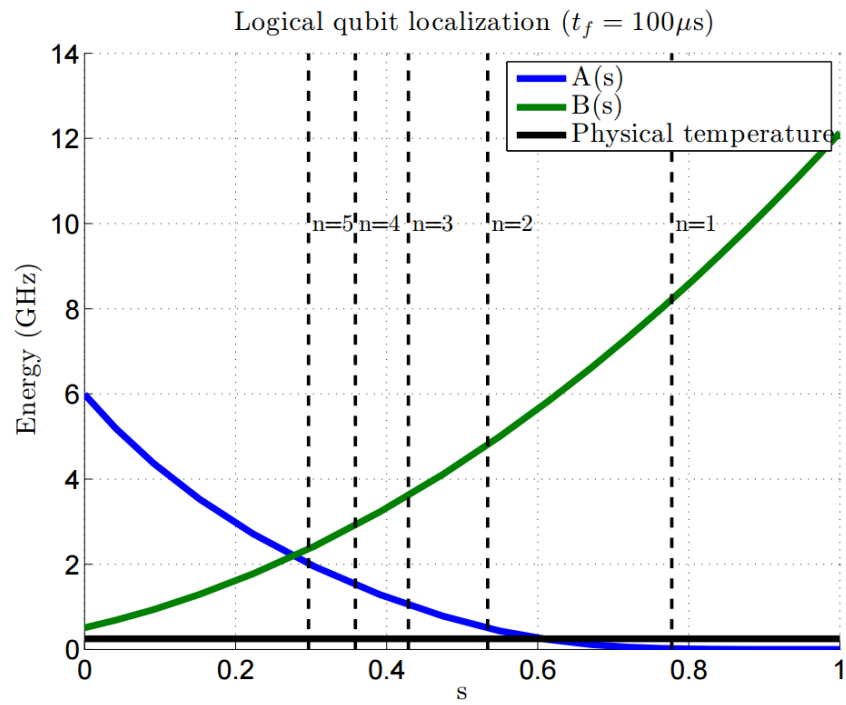


Figure 2.4: Representative annealing schedules with freezeout points for several logical qubit sizes and $t_f = 100\mu s$. The dashed lines show localization points for a $N = 1$ through $N = 5$ logical qubit clusters from right to left, respectively. Data shown are representative of D-Wave 2X systems.

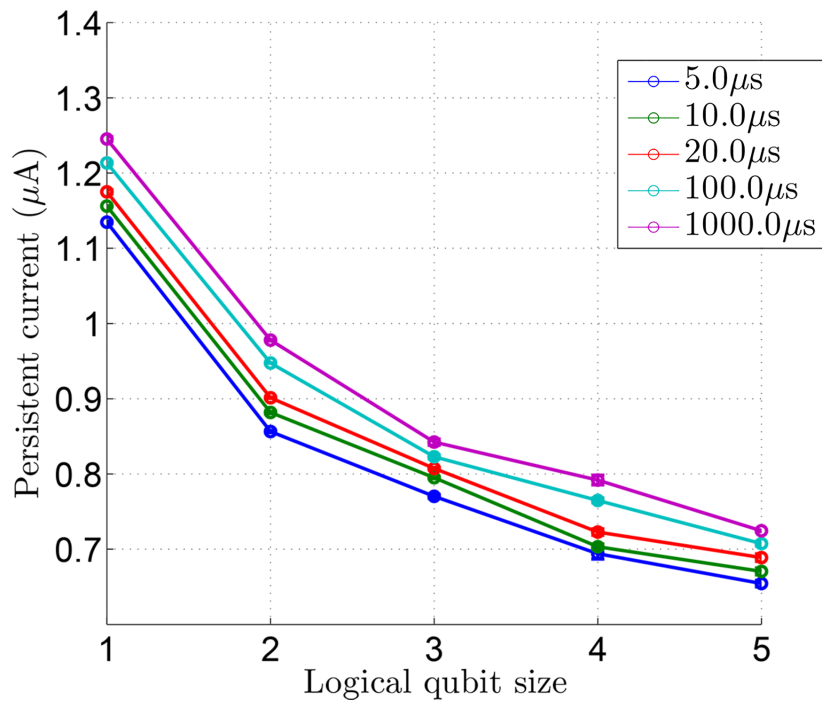


Figure 2.5: I_p (persistent current) at the freezeout point as a function of logical qubit size and annealing time. The change in I_p at freezeout, as the annealing time varies from 5 to 1000 μs and the logical spin cluster size varies from 1 to 5. Shorter annealing time and larger clusters move the freezeout point earlier in the anneal, where I_p is lower. Data shown are representative of D-Wave 2X systems.

The signal corresponding to fixed h values scales as I_p . Larger logical qubits freeze out earlier in the annealing process: at lower I_p . Thus, if an erroneous fixed flux offset exists in the physical body (a static addition to the term Φ_i^x), then the corresponding δh at freezeout grows with the logical qubit size because of the lower persistent current at the freezeout point. This error should roughly double going from a logical qubit of size 1 to a logical qubit of size 5. The standard deviation of δh versus logical qubit size plot in [Figure 3.1](#) indicates that this is approximately true.

2.5 Annealing Controls

This section describes the features that allow you to control the annealing process³: per-qubit anneal offsets and global anneal schedule changes. The latter supports mid-anneal quench and pause as well as reverse annealing.

2.5.1 Anneal Offsets

The standard annealing trajectory lowers $A(s)$ and raises $B(s)$ identically for all qubits in the QPU. This single annealing path, however, may not be ideal for some applications of quantum annealing. This section describes *anneal offsets*, which allow you to adjust the standard annealing path per qubit.

Note: Anneal offsets are not supported on D-Wave 2X and earlier systems. Before using this feature, query the solver properties using SAPI calls to determine whether it is supported and, if so, to obtain the available tuning ranges per qubit.

As discussed above, the annealing process is controlled via a global, time-dependent bias signal $c(s)$ that simultaneously modifies both $A(s)$ and $B(s)$. [Figure 2.1](#) shows typical $A(s)$ and $B(s)$ across the annealing algorithm. [Figure 2.6](#) plots the annealing bias $c(s)$ versus s . Because of the shape of the rf-SQUID flux qubit energy potential, $c(s)$ is not linear in s but is chosen to ensure that $I_p(s)$ grows linearly with s .

On-QPU DACs allow adjustments of static annealing offsets δc_i per qubit, thereby advancing or delaying the annealing signal locally for each. [Figure 2.7](#) shows an example of the annealing control bias with $\delta c_i = 0.05$ and $\delta c_i = -0.05$. Note that the anneal offset is a *vertical* shift up or down in annealing control bias, not a shift in s .

Advancing or delaying the annealing bias by setting $\delta c_i \neq 0$ changes the transverse field $A_i(s)$ with respect to the original global $A(s)$. This allows you to increase or decrease $A_i(s)$. Note that $\delta c_i > 0$ advances the annealing process ($A_i(s) < A(s)$) and $\delta c_i < 0$ delays the annealing process ($A_i(s) > A(s)$). [Figure 2.8](#) shows typical $A_i(s)$ versus s for two values of δc_i . Both $A(s)$ and $B(s)$ simultaneously change with control bias c . Thus, a consequence of advancing or delaying the annealing process with anneal offset δc_i is that $B(s) \rightarrow B_i(s)$. [Figure 2.8](#) also shows typical $B_i(s)$ versus s for the same set of δc_i .

³ Another feature that enables control of the annealing process, the time-dependent gain applied to linear coefficients (biases), h_i , described in the [Solver Properties and Parameters Reference](#) guide, is currently used only experimentally and not described here.

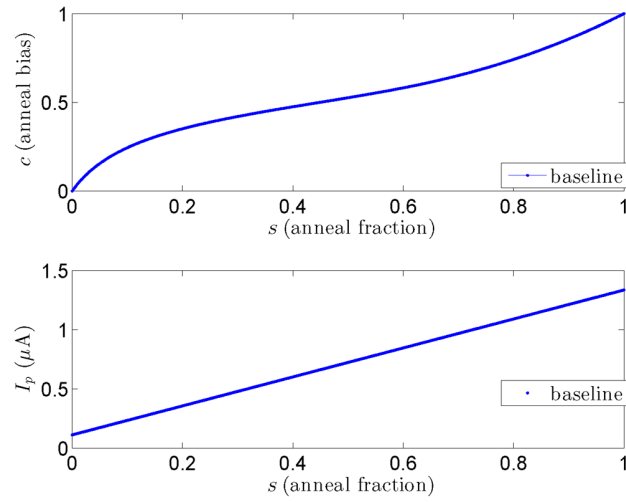


Figure 2.6: Annealing control bias c versus anneal fraction s . At $c = 0, s = 0, A(s) \gg B(s)$, and at $c = 1, s = 1, A(s) \ll B(s)$. Data shown are representative of D-Wave 2000Q systems.

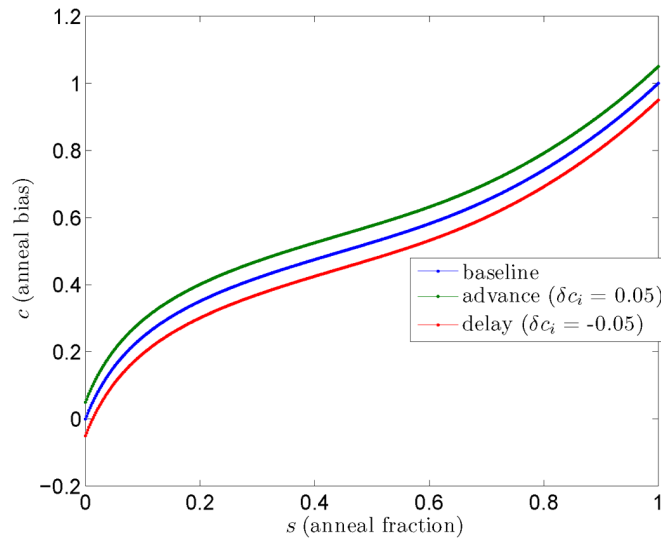


Figure 2.7: Annealing control bias c versus anneal fraction s for several anneal offset values. Data shown are representative of D-Wave 2000Q systems.

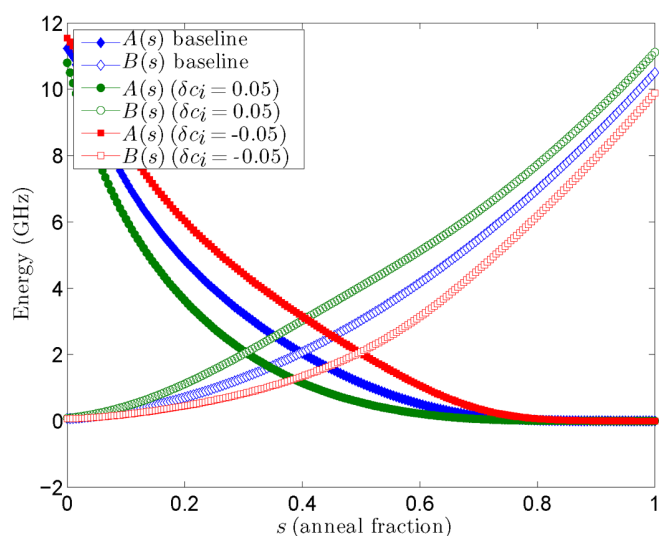


Figure 2.8: $A(s)$ and $B(s)$ versus anneal fraction s . At $c(s = 0) = 0$, $A(s) \gg B(s)$, and at $c(s = 1) = 1$, $A(s) \ll B(s)$. The baseline curve ($\delta c_i = 0$) is shown along with $A_i(s), B_i(s)$ for anneal offset $\delta c_i = 0.05$ (advances the annealing process for the qubit) and $A_i(s), B_i(s)$ for anneal offset $\delta c_i = -0.05$ (delays the annealing process for the qubit). Data shown are representative of D-Wave 2000Q systems.

The change of $B(s) \rightarrow B_i(s)$ has consequences for the target Ising spin Hamiltonian parameters h_i and $J_{i,j}$. The anneal offset for the i th qubit deflects $h_i \rightarrow h_i(\delta c_i, s)$, and the anneal offsets for the i th and j th qubit deflect $J_{i,j} \rightarrow J_{i,j}(\delta c_i, \delta c_j, s)$. Figure 2.9 shows plots of $h_i(\delta c_i, s)$ and $J_{i,j}(\delta c_i, 0, s)$ for several values of δc_i . Figure 2.10 shows plots of $J_{i,j}(\delta c_i, \delta c_j, s)$ for several values of $\delta c_i, \delta c_j$.

The changes shown in Figure 2.9 and Figure 2.10 are s -dependent. The largest changes are earlier in the annealing process. You can choose a particular value of s^* at which to exactly compensate these changes in target parameters by rescaling the requested target parameters. However, for values of s before or after s^* , a residual change in target parameter remains.

Anneal offsets may improve results for problems in which the qubits have irregular dynamics for some easily determined reason. For example, if a qubit's final value does not affect the energy of the classical state, you can advance it (with a positive offset) to reduce quantum bias in the system; see [Kin2016]. Anneal offsets can also be useful in embedded problems with varying chain length: longer chains may freeze out earlier than shorter ones—at an intermediate point in the anneal, some variables act as fixed constants while others remain undecided. If, however, you advance the anneal of the qubits in the shorter chains, they freeze out earlier than they otherwise would. The correct offset will synchronize the annealing trajectory of shorter chains with that of the longer ones. As a general rule, if a qubit is expected to be subject to a strong effective field relative to others, delay its anneal with a negative offset.

Determining the optimum offsets for different problem types is an area of research at D-Wave. Expect that the appropriate offsets for two different qubits in the same problem to be within 0.2 normalized offset units of each other.

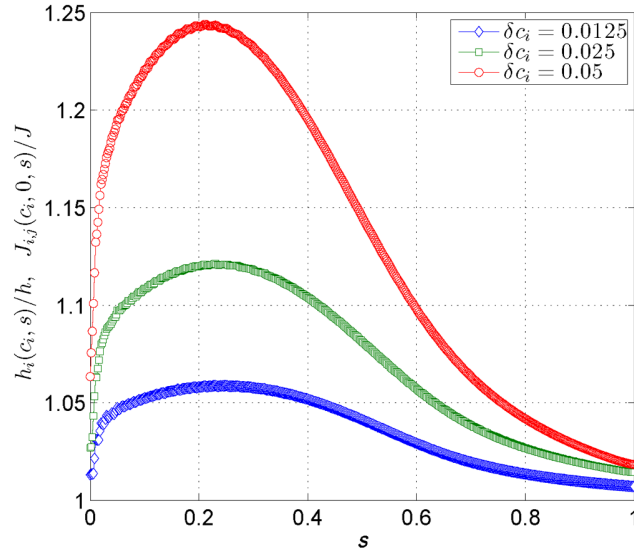


Figure 2.9: $h_i(\delta c_i, s)/h$ and $J_{ij}(\delta c_i, 0, s)/J$ versus s for several values of δc_i . The deviation of these quantities is s -dependent. You can choose a particular value of s at which to exactly compensate the change in target parameter by adjusting $h_i(\delta c_i, s) \rightarrow h'_i$, $J_{ij}(\delta c_i, 0, s) \rightarrow J'_{ij}$. For values of s before or after this point, a residual change in target parameter remains. Data shown are representative of D-Wave 2000Q systems.

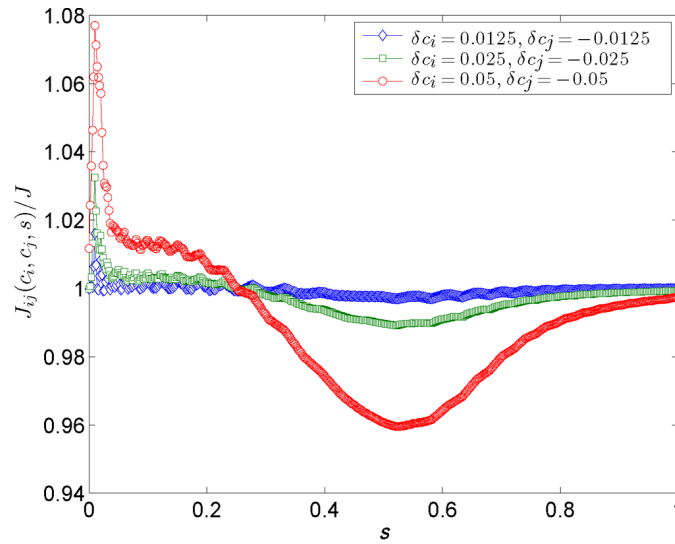


Figure 2.10: $J_{ij}(\delta c_i, \delta c_j, s)/J$ versus s for several values of δc_i and δc_j . The deviation of these quantities away from 1 is s -dependent. You can choose a particular value of s at which to exactly compensate the change in target parameter by adjusting $J_{ij}(\delta c_i, \delta c_j, s) \rightarrow J'_{ij}$. For values of s before or after this point, a residual change in target parameter remains. Data shown are representative of D-Wave 2000Q systems.

2.5.2 Variations on the Global Anneal Schedule

You can make changes to the global anneal schedule by submitting a set of points that that define the piece-wise linear (PWL) waveform of the annealing pattern you want. You can change the standard (forward) schedule by introducing a pause or a quench, or you can initialize the qubits into a specific classical state and anneal in reverse from there. This section describes these features.

Note: Variations on the global anneal schedule are not supported on D-Wave 2X and earlier systems. For D-Wave 2000Q and Advantage systems, the maximum number of points in a schedule is system-dependent.

Note: The maximum number of points permitted in an anneal schedule varies by system and by release. Check the `max_anneal_schedule_points` solver property to obtain this value. For reverse annealing, the maximum number of points allowed is one *more* than the number given by this property.

Pause and Quench

Advantage and D-Wave 2000Q system provide more control over the global annealing trajectories than was possible in previous products. As before, you can scale the quadratic growth in persistent current—that is, quadratic growth in $B(t)$ —using the `annealing_time` parameter. New with the D-Wave 2000Q and later systems, however, is the `anneal_schedule` parameter, which allows for a *pause* or *quench* partway through the annealing process.⁴ A pause dwells for some time at a particular anneal fraction; a quench abruptly terminates the anneal within a few hundred nanoseconds of the point specified. Unlike the anneal offsets feature—which allows you to control the annealing path of individual qubits separately—anneal schedule changes apply to all qubits in the working graph.

Changes to the schedule are controlled by a PWL waveform comprising n pairs of points. The first element is time t in microseconds; the second, the anneal fraction, s , as a value between 0 and 1. This input causes the system to produce linear changes in s between s_i and s_{i+1} . The maximum slope for each segment of the PWL waveform is the upper limit given in the Annealing Slope Range parameter.⁵

The following rules apply to the set of anneal schedule points provided:

- Time t must increase for all points in the schedule.
- For forward annealing, the first point must be $(0, 0)$.
- For reverse annealing, the anneal fraction s must start and end at $s = 1$.
- In the final point, anneal fraction s must equal 1 and time t must not exceed the maximum value in the `annealing_time_range` property.
- The number of points must be ≥ 2 .
- The upper bound is system-dependent; check the `max_anneal_schedule_points`

⁴ The `annealing_time` and `anneal_schedule` parameters are mutually exclusive.

⁵ This and other properties of your system are given in the *QPU Properties* document.

property. For reverse annealing, the maximum number of points allowed is one *more* than the number given by this property.

- Additional rules that govern maximum slope vary by product; check the QPU properties document for your system.

The table below gives three valid examples of anneal schedule points, producing the varying patterns of $B(t)$ that appear in Figure 2.11.

Table 2.1: Anneal Schedule Tuples: Examples

Points	Result
$(0.0, 0.0)(20.0, 1.0)$	Standard trajectory of 20- μs anneal. Here, $B(t)$ grows quadratically with time.
$(0.0, 0.0)(10.0, 0.5)(110.0, 0.5)(120.0, 1.0)$	Mid-anneal pause at $s = 0.5$. The quadratic growth of $B(t)$ is interrupted by a 100- μs pause halfway through.
$(0.0, 0.0)(10.0, 0.5)(12.0, 1.0)$	Mid-anneal quench at $s = 0.5$. The quadratic growth of $B(t)$ is interrupted by a rapid 2- μs quench halfway through.

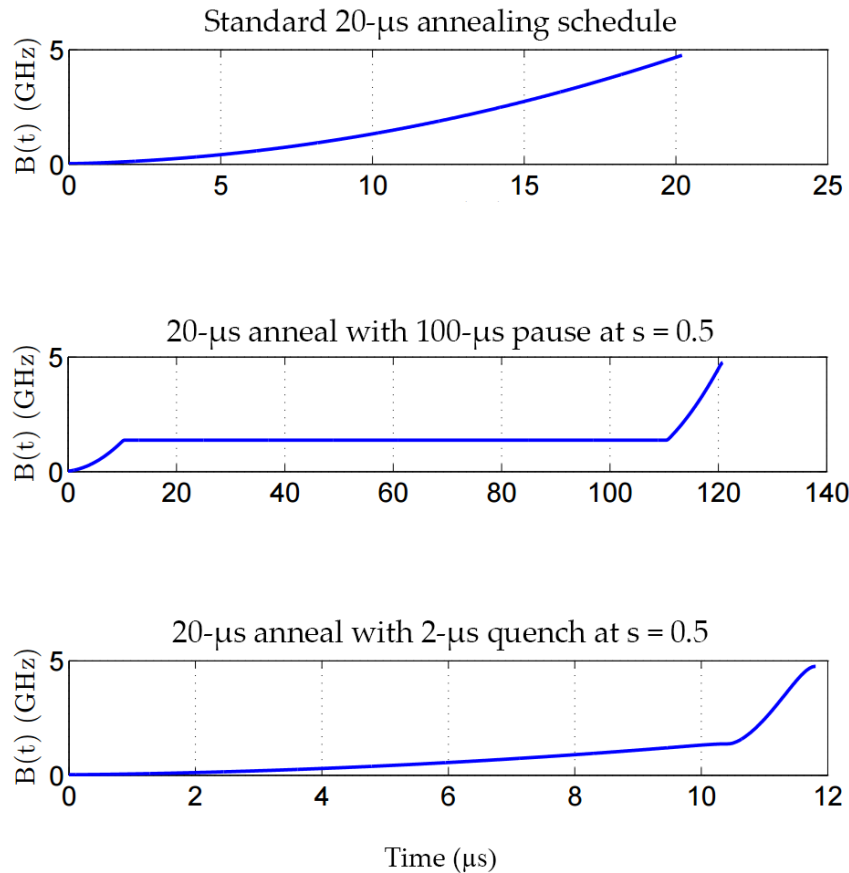


Figure 2.11: Annealing schedule variations. **Top:** Standard quadratic growth of B for $t_f = 20 \mu s$. **Middle:** Mid-anneal pause of 100 μs at $s = 0.5$ for a background $t_f = 20 \mu s$. **Bottom:** Mid-anneal quench of 2 μs at $s = 0.5$ for a background $t_f = 20 \mu s$.

This degree of control over the global annealing schedule allows you to study the quantum annealing algorithm in more detail. For example, a pause can be a useful diagnostic tool for instances with a small perturbative anticrossing. Figure 2.12 shows typical measurements of the 16-qubit instance reported in [Dickson2013] with a pause inserted. While pauses early or late in the anneal have no effect, a pause near the expected perturbative anticrossing produces a large increase in the ground-state success rate.

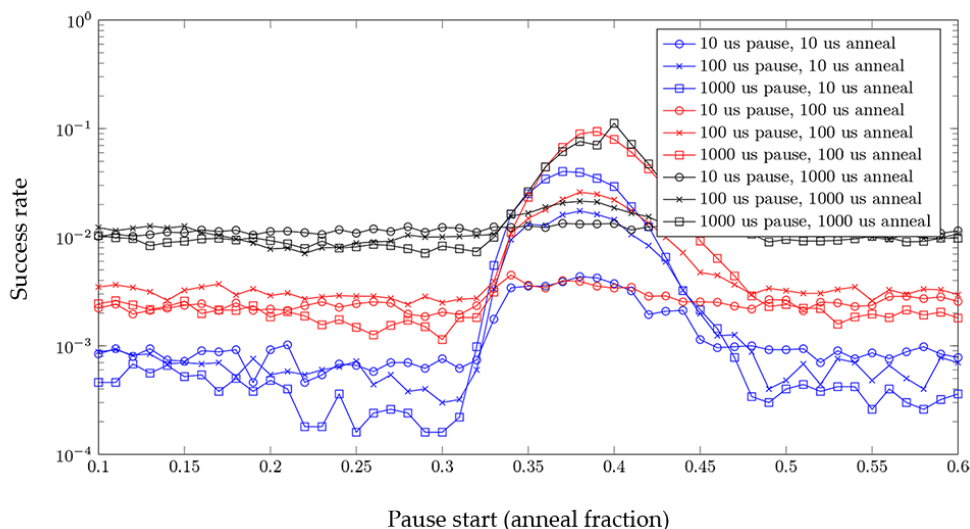


Figure 2.12: Typical measurements from a 16-qubit reference instance showing pauses of different lengths inserted at different points in the annealing schedule. Pauses inserted near the expected anticrossing increase the likelihood of obtaining results from the ground state. Data shown are representative of D-Wave 2000Q systems.

Another example is a quench inserted at $s < 1$. If the quench is fast compared to problem dynamics, then the distribution of states returned by the quench can differ significantly from that returned by the standard annealing schedule. Figure 2.13 shows typical measurements of the same 16-qubit instance with a quench added. The probability of obtaining ground state samples depends on when in the anneal the quench occurs, with later quenches more likely to obtain samples from the ground state.

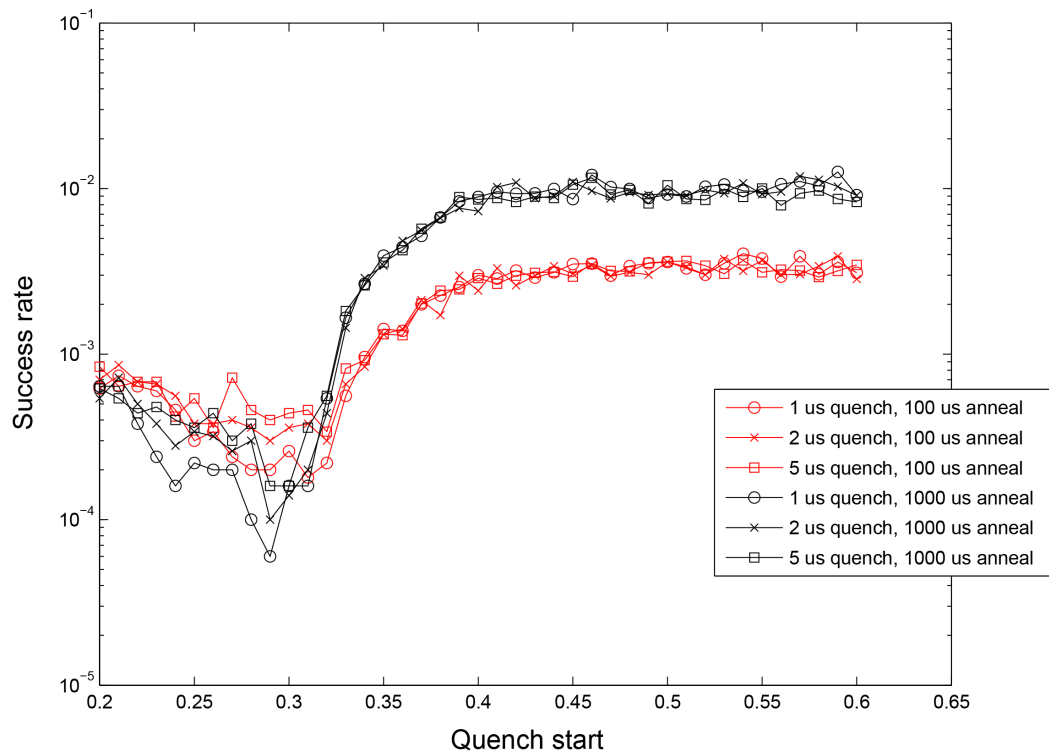


Figure 2.13: Typical measurements from a 16-qubit reference instance showing quenches of different speeds compared to problem dynamics at different points in the anneal schedule. Later quenches increase the likelihood of obtaining results from the ground state. Data shown are representative of D-Wave 2000Q systems.

Reverse Annealing

As described above, the annealing functions $A(s)$ and $B(s)$ are defined such that $A(s) \gg B(s)$ at $s = 0$ and $A(s) \ll B(s)$ at $s = 1$, where s is the normalized annealing fraction. In the standard quantum annealing protocol, s increases linearly with time, with $s(0) = 0$ and $s(t_f) = 1$, where t_f is the total annealing time. The network of qubits starts in a global superposition over all possible classical states and, as $s \rightarrow 1$, the system localizes into a single classical state; see Figure 2.1.

Reverse annealing allows you to initialize the qubits into a specific classical state, begin the evolution at $s = 1$, anneal along a path toward $s = 0$, and then return back up to $s = 1$. Figure 2.14 shows a typical reverse annealing process where the system reverses to $s = 0.5$, pauses for $25 \mu s$ at $s = 0.5$, and ends at $s = 1$.

Note: Spin-reversal transforms are incompatible with reverse annealing.

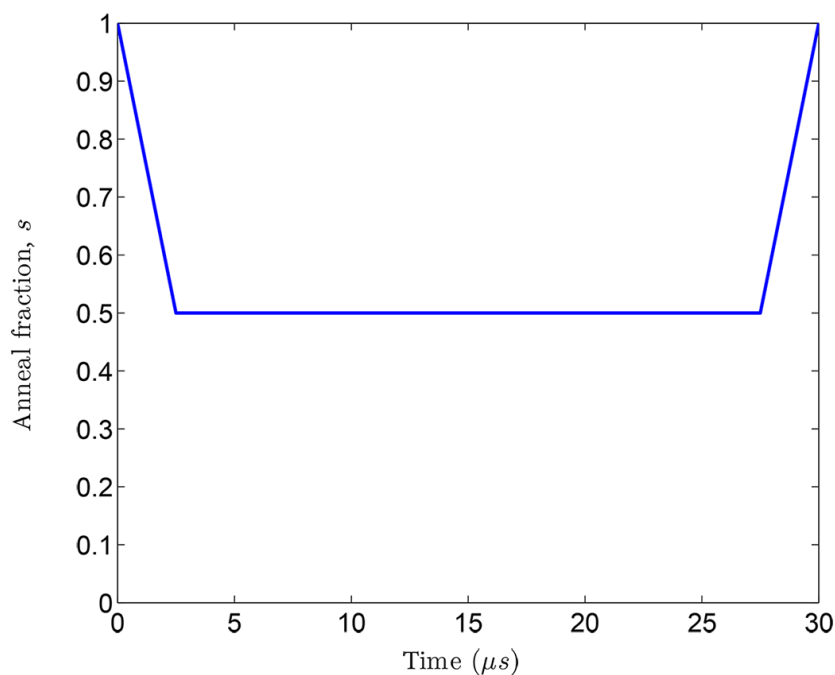


Figure 2.14: Typical reverse annealing protocol with a $5 \mu s$ equivalent ramp rate from $s = 1$ to $s = 0.5$, followed by a pause at $s = 0.5$ for $25 \mu s$, and then a ramp back to $s = 1$.

The reverse annealing feature gives you additional control over and insight into the quantum annealing process in the D-Wave QPU.⁶ Examples of how you might use reverse annealing include:

- Quantum Boltzmann sampling—Prepare a classical state and then draw a set of samples from a probability distribution that may be changing with s .
- Hybrid algorithms—Prepare a classical state provided by a classical heuristic and

⁶ For more information, see *Reverse Annealing for Local Refinement of Solutions*, D-Wave White Paper Series, no. 14-1018A-A, 2017.

then turn on a finite $A(s)$ and allow the system to evolve.

- Tunneling rate measurements—Prepare a particular classical state and measure the rate at which the system tunnels from this state to another for a range of s .
- Relaxation rate measurements—Prepare a classical state that is an excited state of the problem Hamiltonian and measure the rate at which the system relaxes to lower energy states for a range of s .

The reverse annealing interface uses three parameters: *anneal_schedule* defines the waveform $s(t)$, and *initial_state* and *reinitialize_state* control the system state at the start of an anneal.

As for pause and quench, the reverse annealing schedule is controlled by a PWL waveform comprising n pairs of points. The first element of the pair is time t in microseconds; the second, the anneal fraction, s , as a value between 0 and 1. Just as in forward annealing, time t must increase for all points in the schedule. For a reverse anneal, however, the anneal fraction *must* start and end at $s = 1$.

The following table shows the tuples that you submit to get the pattern in [Figure 2.14](#).

Table 2.2: Reverse Annealing Schedule Tuples: Example

Points	Result
(0.0, 0.1)(2.5, 0.5)(27.5, 0.5)(30.0, 1.0)	Reverse anneal, therefore begins at $s = 1$. A 30.0 μs anneal with a mid-anneal pause at $s = 0.5$ that lasts for 25 μs .

When supplying a reverse annealing waveform through *anneal_schedule*, you must also supply the initial state to which the system is set. When multiple reads are requested in a single call to SAPI, you have two options for the starting state of the system. These are controlled by the *reinitialize_state* Boolean parameter:

- *reinitialize_state*=true (default)—Reinitialize the initial state for every anneal-readout cycle. Each anneal begins from the state given in the *initial_state* parameter. Initialization time is required before every anneal-readout cycle. The amount of time required to reinitialize varies by system.
- *reinitialize_state*=false—Initialize only at the beginning, before the first anneal cycle. Each anneal (after the first) is initialized from the final state of the qubits after the preceding cycle. Initialization time is required only once.

The *reinitialize_state* parameter affects timing. See [Solver Computation Time](#) for more information.

2.6 D-Wave QPU Operation

This section describes the programming cycle and the anneal-read cycle of the D-Wave QPU. For information about system timing, see [Solver Computation Time](#).

2.6.1 Programming Cycle

When an Ising problem is provided as a set of h and J values,⁷ the D-Wave system conveys those values to the DACs located on the QPU. Room-temperature electronics generate the raw signals that are sent via wires into the refrigerator to program the DACs. The DACs then apply static magnetic-control signals locally to the qubits and couplers. This is the *programming cycle* of the QPU.⁸ After the programming cycle, the QPU is allowed to cool for a postprogramming thermalization time of, typically, 1 ms; see the [Temperature](#) section for more details about this cooling time.

The total time spent programming the QPU, including the postprogramming thermalization time, is reported back as `qpu_programming_time`.

2.6.2 Anneal-Read Cycle

After the programming cycle, the system switches to the annealing phase during which the QPU is repeatedly annealed and read out. Annealing is performed using the analog lines over a time specified by the user as `annealing_time` and reported by the QPU as `qpu_anneal_time_per_sample`. Afterward, the digital readout system of the QPU reads and returns the spin states of the qubits. The system is then allowed to cool for a time returned by the QPU as `qpu_delay_time_per_sample`—an interval comprising a constant value plus any additional time optionally specified by the user via the `readout_thermalization` parameter.

The anneal-read cycle is also referred to as a *sample*. The cycle repeats for some number of samples specified by the user in the `num_reads` parameter, and returns one solution per sample. The total time to complete the requested number of samples is returned by the QPU as `qpu_sampling_time`.

For more information on timing, see [Solver Computation Time](#).

⁷ Several other instructions to the system are provided by the user: for example, an annealing time over which the quantum annealing process is to occur. See [Solver Properties and Parameters Reference](#) for details.

⁸ In some descriptions, the programming cycle is subdivided into a reset step that erases previous data stored in the DACs, followed by a programming step.

2.7 D-Wave QPU Architecture

The D-Wave QPU is a lattice of interconnected qubits. While some qubits connect to others via couplers, the D-Wave QPU is not fully connected. Instead, the qubits of D-Wave 2000Q and earlier generations of QPUs interconnect in a topology known as *Chimera* while Advantage QPUs incorporate the *Pegasus* topology.

In a D-Wave QPU, the set of qubits and couplers that are available for computation is known as the *working graph*. The *yield* of a working graph is typically less than the total number of qubits and couplers that are fabricated and physically present in the QPU.

A given *logical problem* defined on a general graph can be mapped to a *physical problem* defined on the working graph using *chains*. A chain is a collection of qubits bound together to represent a single logical node. The association between the logical problem and the physical problem is carried out by *minor embedding*. For more details on minor embedding, see [D-Wave Problem-Solving Handbook](#).

The qubits, denoted q_i , implement the Ising spins. Their physical connectivity determines which couplings, $J_{i,j}$, can be set to nonzero values. The allowed connectivity is described with a Chimera or Pegasus graph; see [Figure 2.15](#) and [Figure 2.18](#).

The following subsections provide a brief overview of the topologies. For further details, see the *Getting Started with the D-Wave System* guide.

2.7.1 Chimera

The Chimera architecture comprises sets of connected *unit cells*, each with four horizontal qubits connected to four vertical qubits via couplers. Unit cells are tiled vertically and horizontally with adjacent qubits connected, creating a lattice of sparsely connected qubits. The notation CN refers to a Chimera graph consisting of an $N \times N$ grid of unit cells. The D-Wave 2000Q QPU supports a C16 Chimera graph: its more than 2000 qubits are logically mapped into a 16×16 matrix of unit cells of 8 qubits.

An $M \times N \times L$ Chimera graph consists of an $M \times N$ two-dimensional lattice of blocks, with each block consisting of $2L$ variables, for a total of $2MNL$ variables.

As shown in [Figure 2.16](#), Chimera qubits are characterized as having:

- nominal length 4—each qubit is connected to 4 orthogonal qubits through internal couplers
- degree 6—each qubit is coupled to 6 different qubits

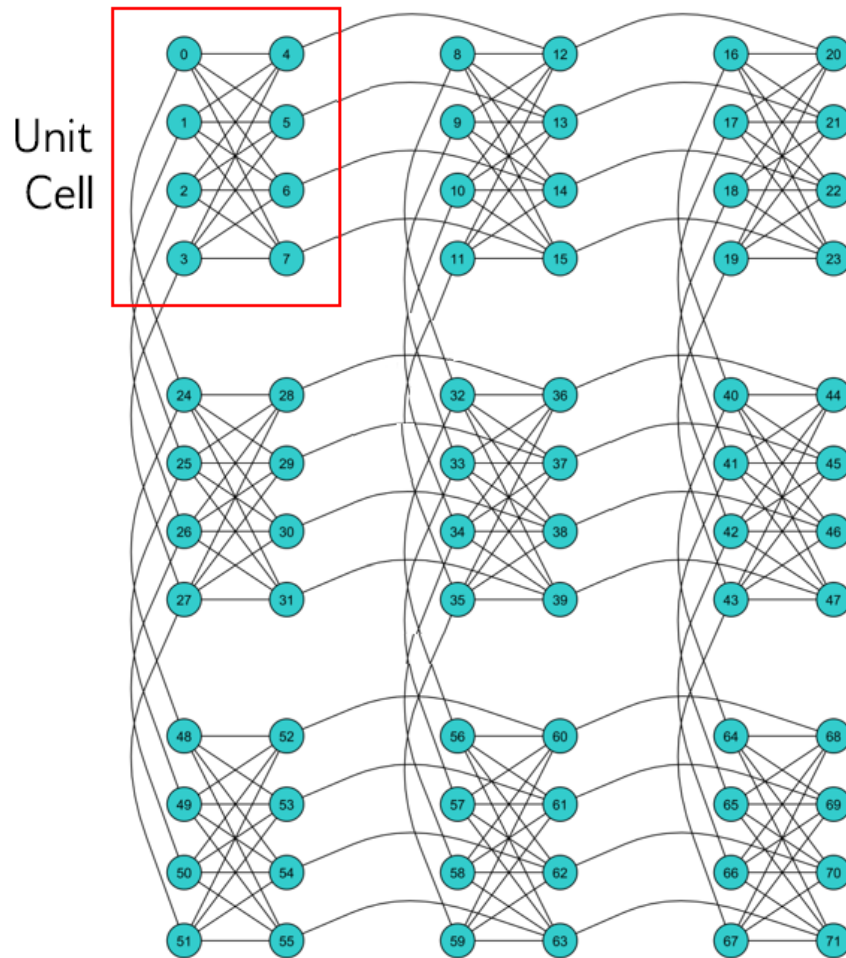


Figure 2.15: A $3 \times 3 \times 4$ Chimera graph. Nodes in an $M \times N \times L$ Chimera graph represent each of the $2MNL$ qubits, q_i . Edges (connections between nodes) in the graph, $J_{i,j}$, indicate couplings that may be nonzero. As an example, $J_{3,4}$ may be nonzero because an edge connects qubits 3 and 4, but $J_{2,3}$ must always be zero because no edge connects qubits 2 and 3. The basic repeating block of Chimera (a block of $2L$ variables with complete bipartite connectivity) may be tiled into an $M \times N$ lattice. The left-side variables within each block connect vertically; the right-side variables, horizontally.

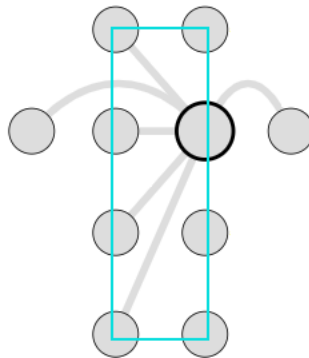


Figure 2.16: Each qubit has 6 couplers, 4 within and 2 between unit cells.

2.7.2 Pegasus

In the Pegasus topology, qubits are “oriented” vertically or horizontally, as in Chimera, but similarly aligned qubits are also shifted, as illustrated in [Figure 2.17](#).

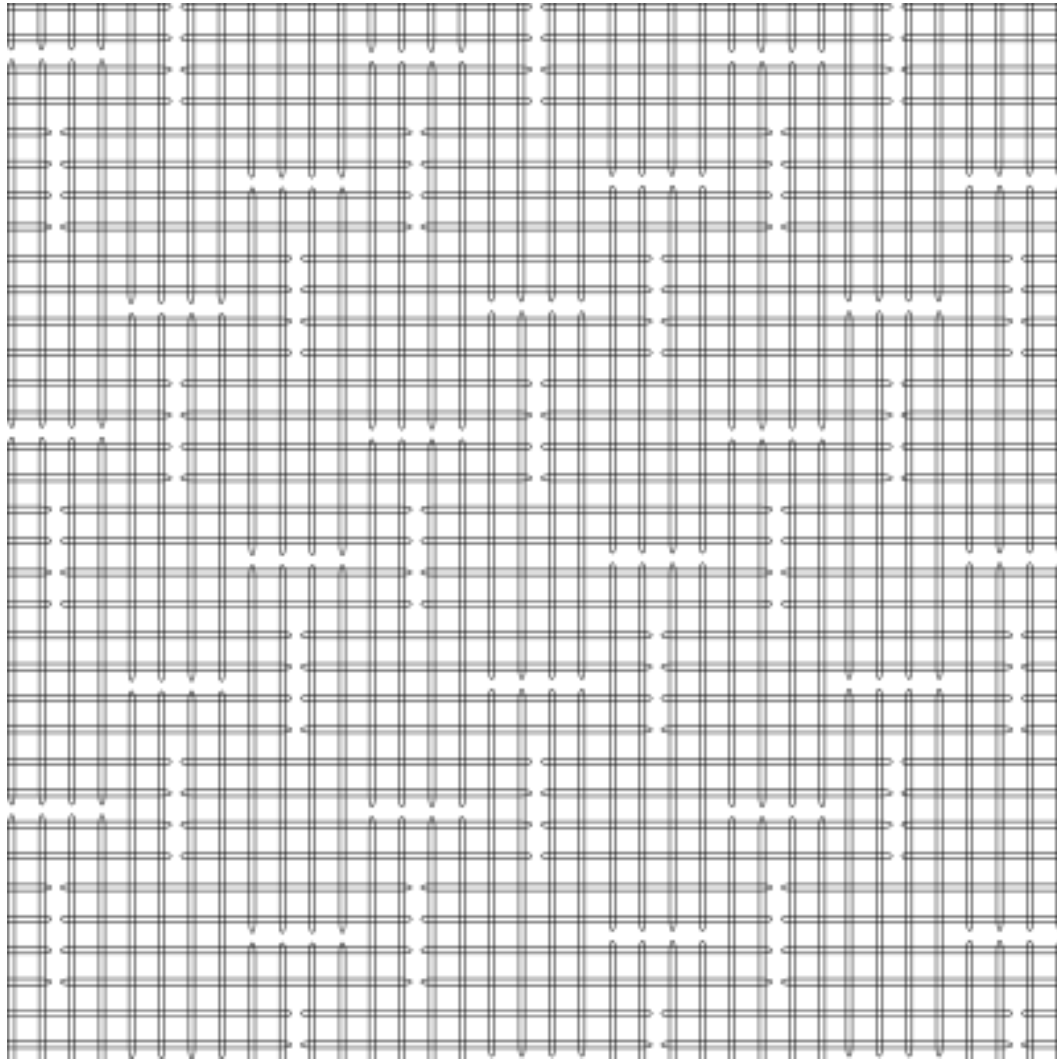


Figure 2.17: A cropped view of the Pegasus topology with qubits represented as horizontal and vertical loops. This graphic shows approximately three rows of 12 vertical qubits and three columns of 12 horizontal qubits for a total of 72 qubits, 36 vertical and 36 horizontal.

For QPUs with the Pegasus topology it is conceptually useful to categorize couplers as internal, external, and odd. [Figure 2.18](#) shows the coupling of qubits in this topology.

Pegasus features qubits of degree 15 and native K_4 and $K_{6,6}$ subgraphs. Pegasus qubits are considered to have a nominal length of 12 (each qubit is connected to 12 orthogonal qubits through internal couplers) and degree of 15 (each qubit is coupled to 15 different qubits).

As we use the notation C_n to refer to a Chimera graph with size parameter N , we refer to instances of Pegasus topologies by P_n ; for example, P_3 is a graph with 144 nodes.

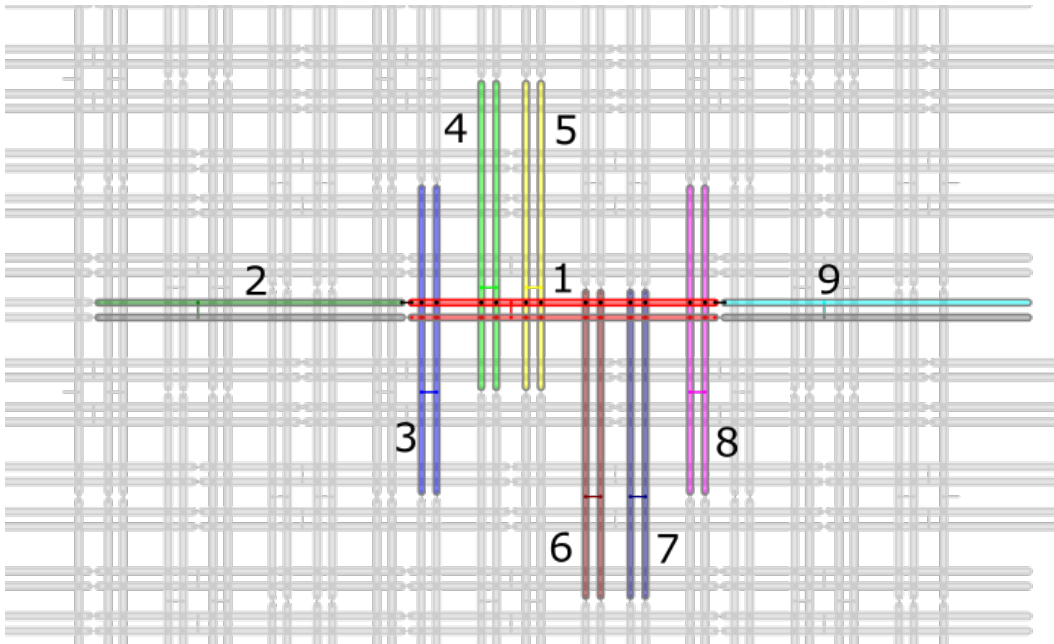


Figure 2.18: Coupled qubits (represented as horizontal and vertical loops): the horizontal qubit in the center, shown in red and numbered 1, with its odd coupler and paired qubit also in red, is internally coupled to vertical qubits, in pairs 3 through 8, each pair and its odd coupler shown in a different color, and externally coupled to horizontal qubits 2 and 9, each shown in a different color.

2.7.3 Virtual Graphs

The D-Wave *virtual graph* feature provides tools and interactive examples that simplify the process of minor-embedding by enabling you to more easily create, optimize, use, and reuse an embedding for a given working graph. When you submit an embedding and specify a chain strength using these tools, they automatically calibrate the qubits in a chain to compensate for the effects of biases that may be introduced as a result of strong couplings. For more information on virtual graphs, see *Virtual Graphs for High-Performance Embedded Topologies*, D-Wave White Paper Series, no. 14-1020A, 2017. This and other white papers are available from <https://www.dwavesys.com/resources/publications>.

This section discusses the underlying controls that support virtual graphs, and that are also available for direct use: extended J range and flux-bias offsets. These controls allow chains to behave more like physical qubits on the working graph, thereby improving the performance of embedded sampling and optimization problems.

Note: Despite the similarity in name, the virtual graphs feature is unrelated to D-Wave's virtual full-yield chip (VFYC) solver.

Extended J Range

As explained above, the Ising minimization problems that the D-Wave system solves may require that the model representing a problem be minor-embedded on the working graph, a process that involves creating qubit chains to represent logical variables. In an embedding, intra-chain qubit couplings must be strong compared to the input couplings between the chains.

Most discussions of chain strength involve the ratio of two absolute values:

- *Chain coupling strength*—Magnitude of couplings between qubits in a chain
- *Problem scale*—Maximum magnitude of couplings among qubits (physical or logical) in a problem

that is,

$$\frac{\text{chain_coupling_strength}}{\text{problem_scale}}. \quad (2.7)$$

For example, if all of our chains have J values of -1 , and our rescaled logical problem has J values of $-\frac{1}{4}$ to $+\frac{1}{4}$, we say that the chain strength is 4. Likewise, if the chains have J values of -2 , and our rescaled logical problem has J values of $-\frac{1}{2}$ to $+\frac{1}{2}$, again we say that the chain strength is 4.

Because the range of coupling strengths available is finite, chaining is typically accomplished by setting the coupling strength to the largest allowed negative value and scaling down the input couplings relative to that. Yet a reduced energy scale for the input couplings may make it harder for the QPU to find global optima for a problem.

To address this issue, some solvers support stronger chain couplings through an *extended J range*. Because embedded problems typically have chain couplings that are at least twice as strong as the other couplings, and standard chain couplings are all negative, this feature effectively doubles the energy scale available for embedded problems; see [Figure 2.19](#).

Using the available larger negative values of J increases the dynamic J range. On embedded problems, which use strong chains of qubits to build the underlying graph, this increased range means that the problem couplings are less affected by ICE and other effects. However, strong negative couplings can bias a chain and therefore flux-bias offsets must be applied to recalibrate it to compensate for this effect; see the next section for more information.

Flux-Bias Offsets

In an optimal QPU calibration, annealing an unbiased chain produces spin-state statistics that are equally split between spin-up and spin-down. When plotted against the h values, this even distribution results in a sigmoid curve that passes through the point of origin (0,0); see [Figure 2.20](#). However, qubits in a chain with strong negative J values experience a J -induced bias—an offset magnetic field that is potentially s -dependent.⁹ This field shifts the sigmoid curve of plotted h values from its ideal path. To compensate, chains using strong negative J couplings must be recalibrated to remove the bias from the chain and reduce the incidence of analog errors that may be associated with minor-embedding.

Recalibration involves applying per-qubit¹⁰ *flux-bias offsets* to nudge the plotted h sigmoid

⁹ This offset occurs because of the higher susceptibility of the tunable coupler [[Har2009](#)].

¹⁰ The same flux-bias offset values are used for all qubits in a chain.

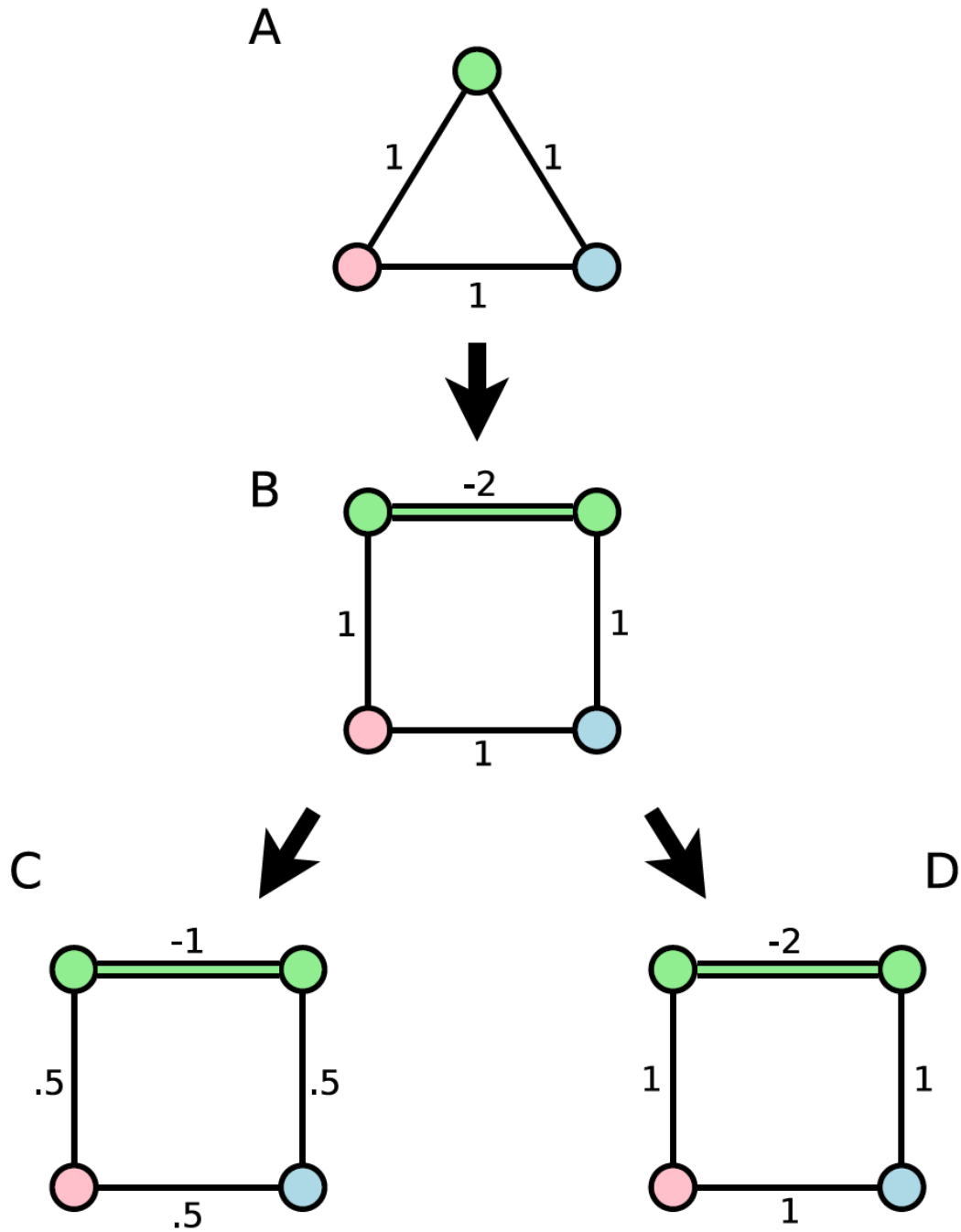


Figure 2.19: Embedding an input that does not fit directly on a D-Wave QPU. The original problem (A) has a green vertex that is replaced by a chain of two vertices connected with a ferromagnetic coupling (B). With the standard coupling range, the embedded problem must be scaled down for the QPU, which can lead to decreased performance (C). However, with an extended coupling range, no rescaling is necessary (D).

to its ideal position. The optimal offset value for a chain depends on the qubits and couplers involved and on the chain coupling strength.

Note: The applied flux bias is constant in time (and s), but it appears in the Hamiltonian shown in Eqn. 2.7 as a term $I_p \phi_{\text{fluxbias}} \sigma_z$ —the applied energy grows as $\sqrt{A(s)}$. An applied flux bias is different from an applied h ; do not use one to correct an error in the other.

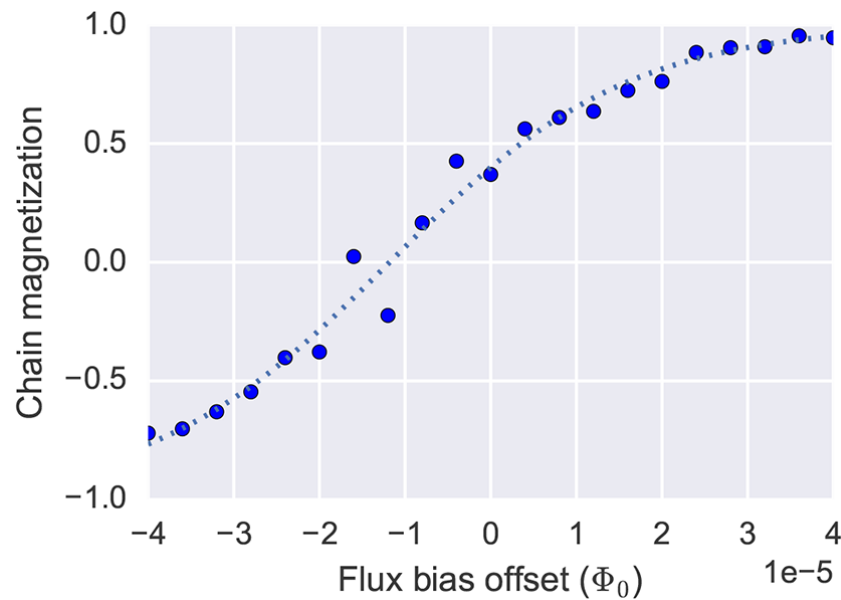


Figure 2.20: Magnetization (mean spin value) of a single chain as a function of applied per-qubit flux-bias offset. A sigmoid function (dashed line) is fit to the data points, and the optimal per-qubit flux-bias offset is given by the x-intercept of this function.

As stated above, D-Wave provides tools that automatically calculate and apply the appropriate flux-bias offsets when you submit an embedding that has strongly coupled chains. At a high level, however, the process of determining the value for the flux-bias offsets involves:

1. Define the strong chain that you intend to use in your embedding. Define it independently from any problem.
2. Sweeping the per-qubit flux-bias offset from the minimum allowed value to the maximum allowed value, anneal the chain many times (e.g., 1000) and calculate the average spin state of the chain (the magnetization).
3. Plot the results, with the per-qubit flux-bias offset on the x-axis and the chain magnetization on the y-axis. Fit a sigmoid model (in this case, a tanh function) to the data.
4. Measure the x-intercept of the sigmoid model (i.e., the per-qubit flux-bias offset value that results in a perfectly balanced magnetization of 0).
5. Add flux-bias offsets across the qubits in the chain to achieve an equal probability of reading spin up or spin down.

This operation is known as *balancing the chain*.¹¹

Usage Guidelines for Virtual Graph Features

Whether virtual graph features are available may vary by solver; check for the *extended_j_range* property to see if it is present and what the range is. (The *j_range* property is unchanged.) When using an extended *J* range, be aware that there are additional limits on the total coupling per qubit: the sum of the *J* values for each qubit must be in the range specified by the *per_qubit_coupling_range* solver property.

Flux-biases are set through the *flux_biases* parameter, which takes a list of doubles (floating-point numbers) of length equal to the number of working qubits. Flux-bias units are Φ_0 ; typical values needed are $< 10^{-4} \Phi_0$. Maximum allowed values are typically $> 10^{-2} \Phi_0$. The minimum step size is smaller than the typical levels of intrinsic $1/f$ noise; see the [Flux Noise and Quantum Annealing](#) appendix.

Note: By default, the D-Wave system automatically compensates for flux drift as described in the [Flux Noise and Quantum Annealing](#) appendix. If you choose to disable this behavior, we recommend that you apply flux-bias offsets manually through the *flux_biases* parameter.

Be aware of the following points when using virtual graph features:

- Neither spin-reversal transformation (SRT) nor autoscaling is possible with extended *J* ranges or flux-bias offsets. If you specify one or more SRTs with a problem submission, the problem is scaled down to use *J* values that fall within the *j_range* property instead.
- When you use these features, autoscaling is automatically disabled by default, unlike in the usual system behavior.
- The *J* value of every coupler must fall within the *extended_j_range* property.
- The sum of all the *J* values of the couplers connected to a qubit must fall within the *per_qubit_coupling_range* property. For example, if this property is $[-9.0 \ 6.0]$, the following *J* values for a six-coupler qubit are permissible:

$$1, 1, 1, 1, 1, 1,$$

where the sum is 6, and also

$$1, 1, 1, -2, -2, -2,$$

where the sum is -3 . However, the following values, when summed, exceed the range and therefore are impermissible:

$$-2, -2, -2, -2, -2, -2.$$

¹¹ While you can also balance a chain by adjusting the *h* values on each of the chained qubits, this is less effective as a means to improve performance.

- While the extended J range in principle allows you to create almost arbitrarily long chains without breakage, the maximum chain length where embedded problems work well is expected to be in the range of 5 to 7 qubits.
- When embedding logical qubits using the extended J range, limit the degree, D , of each node in the logical qubit tree to

$$D = \text{floor} \left[\frac{\min(\text{per_qubit_coupling_range}) - \text{num_couplers_per_qubit} \times \min(j_range)}{\min(\text{extended_j_range}) - \min(j_range)} \right] \quad (2.8)$$

where $\text{num_couplers_per_qubit} = 6$ for the D-Wave 2000Q QPU; see [Figure 2.16](#).

ICE: DYNAMIC RANGES IN H AND J VALUES

The dynamic range of h and J values may be limited by *integrated control errors* (ICE). The term *ICE* refers collectively to these sources of infidelity in problem representation. This chapter provides an overview of ICE, describes the main sources that contribute to it, and provides guidance on measuring its effects.

Note: Data given in this chapter are representative of D-Wave systems.

3.1 Overview of ICE

Although h and J may be specified as double-precision floats, some loss of fidelity occurs in implementing these values in the D-Wave QPU. This fidelity loss may affect performance for some types of problems.

Specifically, instead of finding low-energy states to an optimization problem defined by h and J as in Eqn. 2.1, the QPU solves a slightly altered problem that can be modeled as

$$E_{ising}^{\delta}(\mathbf{s}) = \sum_{i=1}^N (h_i + \delta h_i) s_i + \sum_{i=1}^N \sum_{j=i+1}^N (J_{i,j} + \delta J_{i,j}) s_i s_j, \quad (3.1)$$

where δh_i and $\delta J_{i,j}$ characterize the errors in parameters h_i and $J_{i,j}$, respectively.¹

The error δh_i depends on h_i and on the values of all incident couplers $J_{i,j}$ and neighbors h_j , as well as *their* incident couplers $J_{j,k}$ and next neighbors h_k . That is, if spin i is connected to spin j , and spin j is connected to spin k in the topological graph, then δh_i may depend, to varying extents, on $h_i, h_j, h_k, J_{i,j}$, and $J_{j,k}$. This dependency holds when the relevant qubits and couplings are present in the graph, irrespective of whether they are set to nonzero values. Similarly, $\delta J_{i,j}$ depends on spins and couplings in the local neighborhood of $J_{i,j}$. For example, if a given problem is specified by $(h_1 = 1, h_2 = 1, J_{1,2} = -1)$, the QPU might actually solve the problem $(h_1 = 1.01, h_2 = 0.99, J_{1,2} = -1.01)$. Changing just a single parameter in the problem could change all three error terms, altering the problem in different ways.

The *probability distribution* of δh and δJ is an ensemble across a set of problem settings h and J and across all i and i, j pairs. The δh and δJ values are Gaussian-distributed with mean μ and standard deviation σ that vary with anneal fraction s during the anneal.² The

¹ Because δh and δJ are summed over N , fidelity limitations tend to have a greater effect on performance for full-QPU sized problems, for a given dynamic range and distribution of h and J .

² Assumed for simplicity; distributions seen in actual results are close to this.

QPU control system is calibrated so that there is typically a value of s for which μ is zero. This point is chosen to be somewhere between the single qubit freezeout point described in the [Freezeout Points](#) section (later in the anneal), and at the quantum critical point of a one-dimensional Ising chain (earlier in the anneal). Distributions with nonzero μ for some values of s are considered to be due to *systematic errors* discussed later in this chapter. Thus, the expected deviations of h and J during operation are the sum of a systematic contribution μ and a random component with standard deviation σ .

[Figure 3.1](#) and [Figure 3.2](#) show example measurements of δh and δJ distributions at different fractional times s . See the [Using Two-Spin Systems to Measure ICE](#) section for a description of the measurement method.

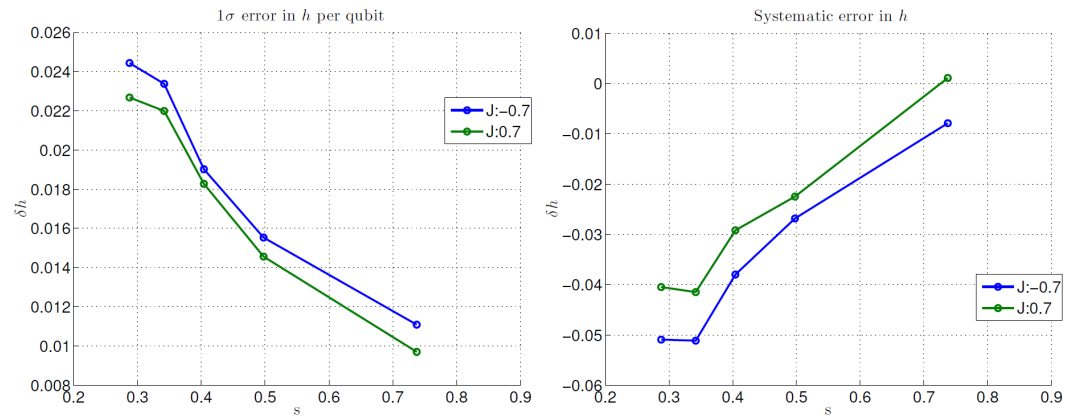


Figure 3.1: δh distributions estimated from best-fit parameters to a thermal model. The left plot shows the standard deviation of the distribution and the right plot shows the mean. Data were taken for logical qubits between size 1 (larger s) and size 5 (smaller s) to get an estimate of δh over a range of s . Data shown are representative of D-Wave systems.

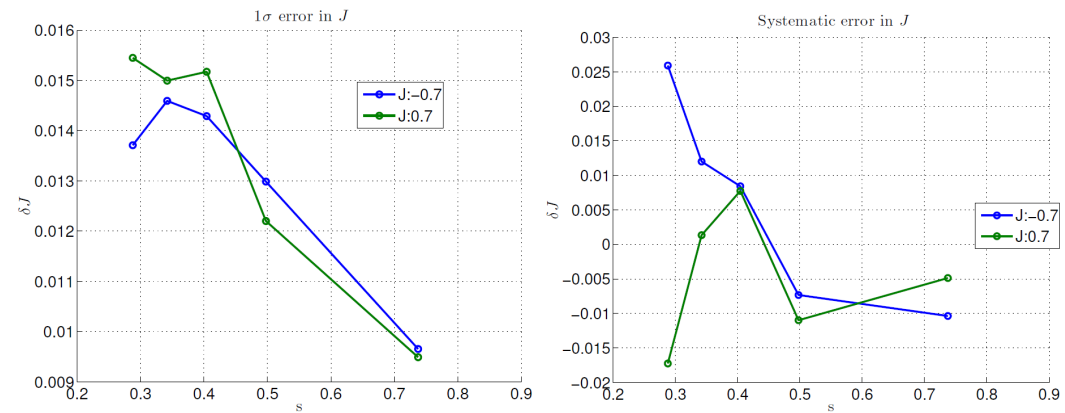


Figure 3.2: δJ distributions estimated from best-fit parameters to a thermal model. The left plot shows the standard deviation of the distribution and the right plot shows the mean. Data were taken for logical qubits between size 1 (larger s) and size 5 (smaller s) to get an estimate of δJ over a range of s . Data shown are representative of D-Wave systems.

3.2 Sources of ICE

The Ising spins on the D-Wave QPU are intrinsically analog, controlled through spatially local magnetic fields. The controls are a combination of the output of on-QPU DACs and other analog signals shared among groups of spins. A calibration procedure, conducted when the QPU first comes online, determines the mapping between Ising problem specifications h and J , and the control values used during annealing.

QPU calibration is a significant part of the time required to install a D-Wave system at a site. It involves a series of measurements of the QPU and the refrigerator to obtain data used to build models that achieve the desired Ising spin Hamiltonian.

These models identify several factors that contribute to distortions of h and J due to ICE. Understanding these factors, and how to compensate for them, can guide your choices in h and J when you specify a problem. Listed below are the dominant sources of ICE. The subsections that follow give additional details.

- **ICE 1**—The on-QPU rf-SQUID qubits behave weakly as couplers, which leads to effective next-nearest-neighbor (NNN) J interactions and a leakage of applied h biases from a qubit to its neighbors. See the [Background Susceptibility \(ICE 1\)](#) section.
- **ICE 2**—The qubits experience low-frequency $1/f$ -like flux noise. This noise contributes an error that varies in time (that is, between runs on the QPU), and also varies with s . See the [Flux Noise of the Qubits \(ICE 2\)](#) section.
- **ICE 3**—The on-QPU DACs that provide the specified h and J values have a finite quantization step size. See the [DAC Quantization \(ICE 3\)](#) section.
- **ICE 4**—The ratio of h/J may differ slightly for different annealing parameters such as t_f . See the [I/O System Effects \(ICE 4\)](#) section.
- **ICE 5**—Qubits cannot be made perfectly identical. As a result, individual spins may have slightly different magnitudes (persistent currents) and tunneling energies. These differences also vary with anneal fraction s . See the [Distribution of \$h\$ Scale Across Qubits \(ICE 5\)](#) section.

3.2.1 Background Susceptibility (ICE 1)

During the annealing process, every Ising spin has a coupler-like effect on its neighbors (that is, the spins to which it is connected by couplings) that is not captured by the problem Hamiltonian. This effect takes two main forms:

- Spin i induces *next-nearest neighbor* (NNN) couplings between pairs of its neighboring spins.
- The applied h bias *leaks* from spin i to its neighboring spins.³

The strength of this *background susceptibility* effect is characterized by a parameter χ .⁴

For example, consider the three-spin problem shown in [Figure 3.3](#). This system is described

³ A more detailed description of how this arises from rf-SQUID qubits is in [Har2010].

⁴ Use of the χ symbol is consistent with the usage in physics of χ to characterize the susceptibility of a magnetic system. We use a normalized $\chi = M_{\text{AFM}}\chi_q$, where M_{AFM} is the maximum available antiferromagnetic mutual inductance, and $\chi_q = \frac{dI_p}{d\Phi_q}$ is the physical qubit susceptibility.

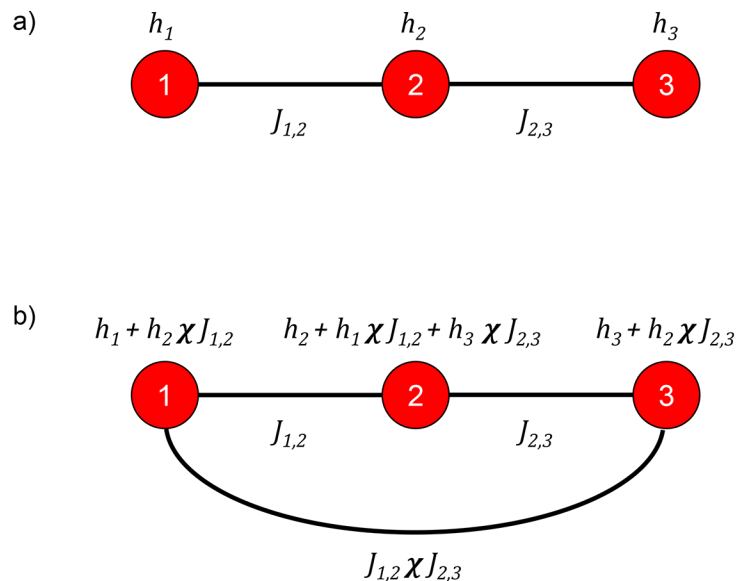


Figure 3.3: a) Three-spin system to look at the effect of χ . Three spins, red circles, labeled 1, 2, 3 with local h -biases of h_1, h_2, h_3 , respectively, and couplings between neighboring spins denoted by lines of $J_{1,2}$ between spin 1 and spin 2, and $J_{2,3}$ between spin 2 and spin 3. b) The model for the system showing induced couplings and h -biases due to qubit susceptibility, χ .

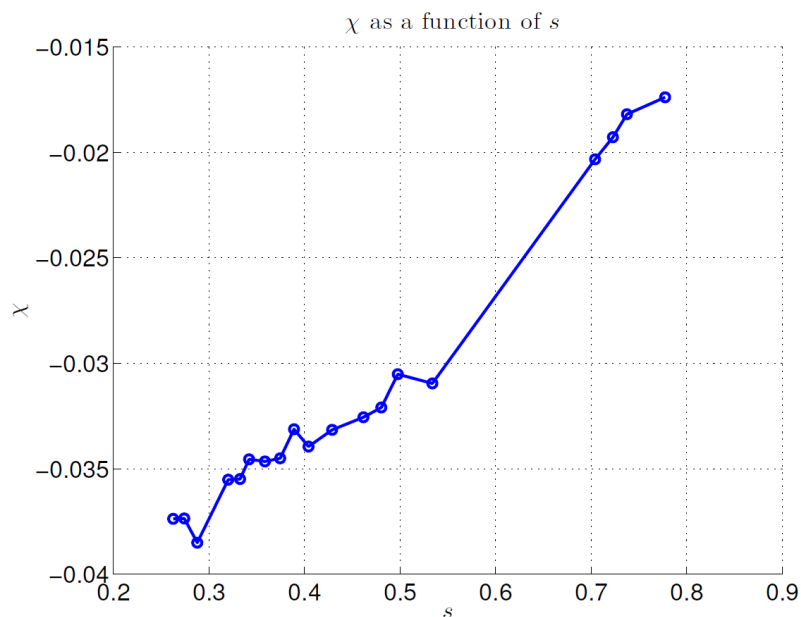


Figure 3.4: χ variation during the anneal. Data shown are representative of D-Wave 2X systems.

by the Ising energy function

$$E_3(\mathbf{s}) = +h_1s_1 + h_2s_2 + h_3s_3 + J_{1,2}s_1s_2 + J_{2,3}s_2s_3. \quad (3.2)$$

The energy function solved by the QPU, however, has some extra terms:

$$E_\chi^\delta(\mathbf{s}) = \underbrace{+h_2\chi J_{1,2}s_1 + h_1\chi J_{1,2}s_2 + h_3\chi J_{2,3}s_2 + h_2\chi J_{2,3}s_3}_{h \text{ leakage}} + \underbrace{J_{1,2}\chi J_{2,3}s_1s_3}_{\text{NNN coupling}}. \quad (3.3)$$

The NNN coupling occurs because spin 1 induces an interaction between spins 1 and 3 with magnitude $J_{1,2}\chi J_{2,3}$. Similarly, h_2 leaks onto spin 1, with magnitude $h_2\chi J_{1,2}$, and so on for the other terms.

Figure 3.4 shows how χ typically varies with s , from around -0.04 early in the anneal to near -0.015 late in the anneal, at which point single-spin dynamics freeze out. More exact values may be found in the properties for individual QPUs, available from D-Wave.

3.2.2 Flux Noise of the Qubits (ICE 2)

As another component of ICE, each h_i is subject to an independent (but time-dependent) error term that comes from the $1/f$ flux noise of the qubits.⁵ There are fluctuations in the flux noise that have lower frequency than the typical inverse annealing time, so problems solved in quick succession have correlated contributions from flux noise. By default, flux drift is automatically corrected every hour by the D-Wave system so that it is bounded and approximately Gaussian when averaged across all times; see the [Drift Correction](#) section for the procedure. You can disable this automatic correction by setting the `flux_drift_compensation` solver parameter to `false`. If you do so, apply flux-bias offsets manually; see the [Flux-Bias Offsets](#) section.

Because the physical source of noise is flux fluctuations on the qubits, the *effective* level of noise is larger earlier in the anneal when the persistent current in qubits is smaller, so that h is relatively smaller in physical flux units; see the [Coupled rf-SQUID Qubits](#) section and the [Flux Noise and Quantum Annealing](#) appendix for details.

The Fourier spectrum of the fluctuations of δh^2 varies approximately as A^2/f^α , where f is frequency from 1 mHz to the highest-resolvable frequency, A is the amplitude of the fluctuations at $f = 1$ Hz, and α is the spectral tilt—or frequency dependence—of the fluctuations. Due to the flux drift compensation, the spectrum becomes flat and close to zero below 1 mHz. Integrating the noise from 1 mHz to 1 MHz (the relevant band of interest observable directly), the Gaussian contribution to δh is approximately 0.009 early in the anneal and less later in the anneal. Figure 3.5 shows typical data for a 64-qubit system.

⁵ Couplers also have $1/f$ flux noise, but this effect is insignificant compared to δh and δJ .

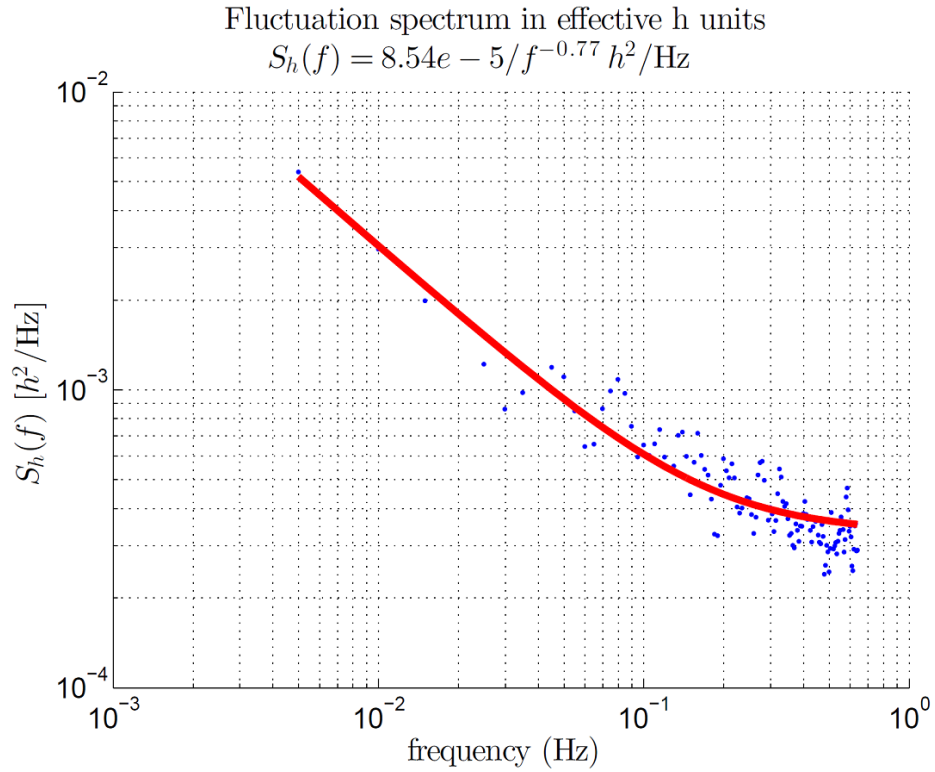


Figure 3.5: Power spectral density of h fluctuations for a 64-qubit cluster. Data are taken by strongly coupling 64 qubits together and measuring the magnetization fluctuation of the system over time. The power spectral density of this signal, in effective h units, is plotted as a function of frequency. The solid line shows a best fit to the model $A^2/f^\alpha + wn$, where wn is the statistical noise floor of the measurement technique and is not a measurement of any intrinsic broadband fluctuation of the qubit environment. See the [Flux Noise and Quantum Annealing](#) appendix for more details. Data shown are representative of D-Wave 2X systems.

3.2.3 DAC Quantization (ICE 3)

The on-QPU DACs that provide the user-specified h and J values have a finite quantization step size. That step size depends on the value of the h or J applied because the response to the DAC output is nonlinear.

This random error contribution is described by a uniform distribution centered at 0 and having errors $\pm b$. Typical errors for h and J are shown in Figure 3.6. Note that these errors are smaller than other contributors to ICE, and that they are more pronounced for negative values of h and J .

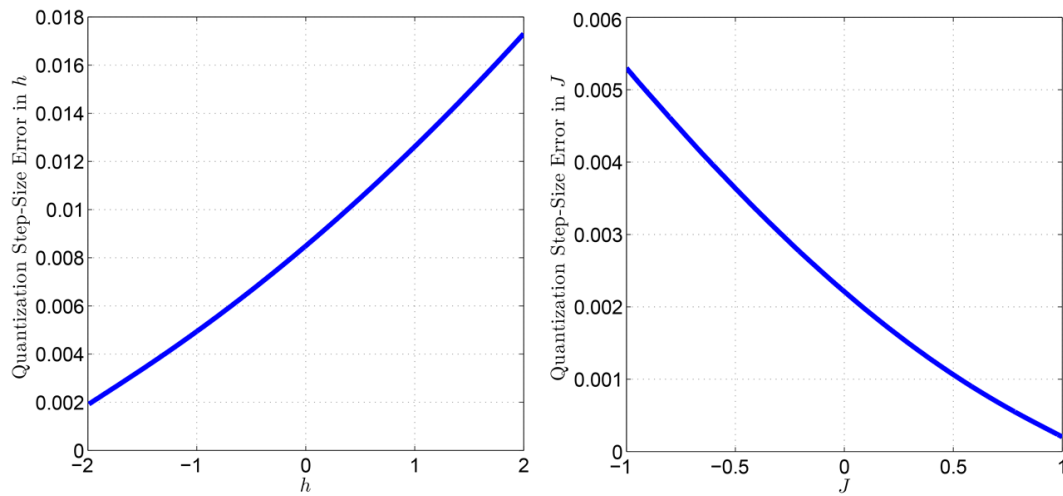


Figure 3.6: Typical quantization step for the DAC controlling the h parameter (left) and J parameter (right). For example, $h = 0.000$ may be off by as much as ± 0.0038 when realized on the QPU. Over all qubits, you might find a worst-case qubit biased with $h = 0.0038$, and another worst-case qubit biased with $h = -0.0038$. Data shown are representative of D-Wave 2X systems.

3.2.4 I/O System Effects (ICE 4)

Several time-dependent analog signals are applied to the QPU during the annealing process. Because the I/O system that delivers these signals has finite bandwidth, the waveforms must be tuned for each anneal to minimize any potential distortion of the signals throughout the annealing process. As a result, the ratio of h/J may vary slightly with t_f and with scaled anneal fraction s .

3.2.5 Distribution of h Scale Across Qubits (ICE 5)

Assuming a fixed temperature of the system, and $J = 0$, the expected graph of the relationship between user-specified h and QPU-realized h has slope = 1, reflecting a constant mean offset μ in the distribution of δh . The measured slopes, however, are different for each spin, and are Gaussian-distributed.

This divergence results from small variations in the physical size of each qubit and from imperfections that arise when attempting to homogenize the macroscopic parameters—physical inductance L , capacitance C , and Josephson junction critical current I_c —across all physical qubits in the QPU.

Assuming a fixed temperature for the system, the ideal relationship, or slope, between qubit population and applied h is identical across an ensemble of devices. The distribution of measured slopes, however, is Gaussian-distributed with a standard deviation of approximately 1%, which contributes to δh and δJ . This distribution results from differences in the magnitude of each spin s_i ; see the [Using Two-Spin Systems to Measure ICE](#) section.

3.3 Measuring ICE

This section describes how to measure the effects of $1/f$ noise as well as how to harness effective two-spin systems to measure the effects of ICE.

3.3.1 Measuring $1/f$ Noise

A straightforward way to look at the $1/f$ noise in the h parameter is to set all h and J values that are equal to 0 on the full QPU and observe how resulting qubit distributions drift over time in repeated tests. This approach identifies the effective $1/f$ noise error on h when referenced to the single qubit freezeout point s_q^* . To probe the $1/f$ noise at earlier anneal times (and to amplify the error signal), create clusters of strongly coupled qubits and measure the time-dependent behavior of the net magnetization of system as described in the next section.

3.3.2 Using Two-Spin Systems to Measure ICE

We can use the results of problems defined on pairs of Ising spins scattered over the topological graph to help to characterize ICE effects.

Simple Two-Spin Systems

The simplest version of such problems has two Ising spins with biases h_1 and h_2 and a coupling between them of weight J . The energy of this system is

$$E_{2spin}(\mathbf{s}) = +h_1s_1 + h_2s_2 + Js_1s_2. \quad (3.4)$$

To measure ICE using simple two-spin systems, we first find all independent edge sets of the available graph; that is, the sets of two-spin systems that can be manipulated independently. Simultaneously, for each set, we ask the QPU to solve the independent two-spin problems at a variety of h_1 and h_2 settings near the phase boundaries indicated in [Figure 3.7](#). We then fit the resulting data to a thermal model to estimate the deviations, δh and δJ , from Eqn. 3.4 using Eqn. 3.5.

The two-spin phase diagram in [Figure 3.7](#) characterizes the lowest energy state of the system as a function of h_1 and h_2 for given J , here set to $J = -0.5$. The dashed line delineates regions of the h_1 and h_2 space where resulting Ising spins are indicated by the up ($s = 1$) and down ($s = -1$) arrows. Applying a larger magnitude (more negative) J value increases ferromagnetic interactions, growing the regions of $\downarrow\downarrow$ and $\uparrow\uparrow$ while shrinking the regions of $\downarrow\uparrow$ and $\uparrow\downarrow$.

The ideal diagram may be compared to one obtained assuming the existence of background susceptibility errors. For this problem, the χ correction terms are

$$E_{2spin}^\delta(\mathbf{s}) = +h_2\chi Js_1 + h_1\chi Js_2. \quad (3.5)$$

The h -leakage effects are shown in [Figure 3.7](#): the blue lines denote the boundaries of the different spin ordering regions for the case where $\chi = -0.05$, versus the ideal case shown by the dashed lines. (The two-spin system does not have NNN effects.)

Given the form of the h -leakage terms, applying small adjustments to the h biases on the original problem can compensate somewhat for this error. However, χ varies during the anneal (see [Figure 3.4](#)), and this correction corresponds to one specific point during the anneal. The single qubit freezeout point for typical anneal times occurs when $s \approx 0.8$ and $A(s)$ is around 100 MHz (see [Figure 2.1](#)). This is the last point in the anneal where any meaningful spin-flip dynamics occur; at that point χ is approximately -0.015, so that value can, in principle, compensate for h -leakage. A better approach, however, is to choose χ to correspond to a point earlier in the anneal—at or before the crossing point of $A(s)$ and $B(s)$. This is the localization point of a one-dimensional chain of spins.

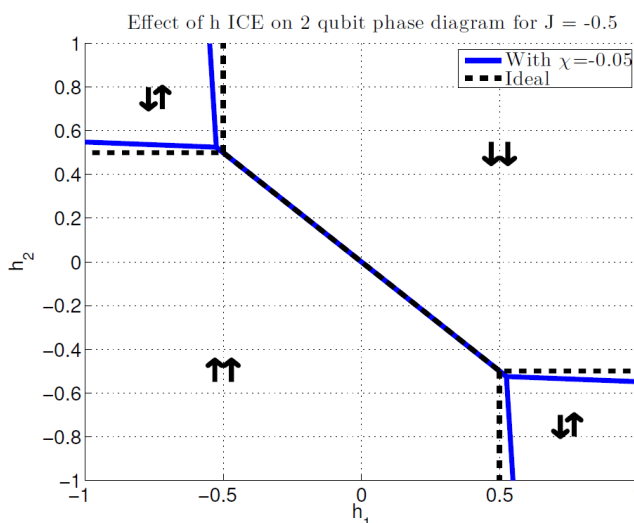


Figure 3.7: Two-qubit (two Ising spin) phase diagram for $J = -0.5$. The dashed lines show the locations of the expected phase boundaries between the four possible states of the system under the assumption of ideal Ising spin behavior. The solid lines show the locations of the phase boundaries for $\chi = -0.05$. To estimate the Ising spin parameters without systematic bias, χ must be included in any model of the physical system. Data shown are representative of D-Wave 2X systems.

Effective Two-Spin Systems For Larger Problems

Similar measurements may be repeated for larger problems made up of *logical spins* formed by strongly coupled groups of spins with strong intracluster coupling weights J_{cluster} . Pairs of clusters are connected by the intercluster coupling weight J , and the h_1 and h_2 weights are assigned to all spins in each cluster. These clusters freeze out at different anneal times depending on the number of spins and on coupling strengths J and J_{cluster} , as shown in [Figure 2.3](#) and [Figure 2.4](#). These *effective two-spin instances* make it possible to probe ICE effects on h and J at different times during the anneal.

For each logical qubit size up to 6 (corresponding to 6+6 spins) held together with $J_{\text{cluster}} = 1$, we ask the QPU to solve independent problems on the full-sized graph for varying J , h_1 and h_2 . We then fit the measured results to a fixed-temperature model of the phase diagram for the ideal problem—appropriately adjusted to the larger qubit counts, including the proper χ contributions from the logical qubit components.

The dominant effects taken into account during the fitting to the phase diagram model are:

- The J realized may vary from the J requested. ICE effects on couplings characterized by δJ are shown in [Figure 3.2](#).
- With large-enough samples of logical two-spin problems, random δh errors cancel out; what remains is an offset field for each spin. This offset, h_{offset} , characterizes the intrinsic flux offset of the logical qubit that is independent of h .
- The difference between specified h and realized h may vary as described in the [Distribution of \$h\$ Scale Across Qubits \(ICE 5\)](#) section. [Figure 3.8](#) shows the effect of this type of error on the phase diagram.
- Additional warping of the phase diagram occurs as χ changes from -0.04 to -0.015 through anneal time s .

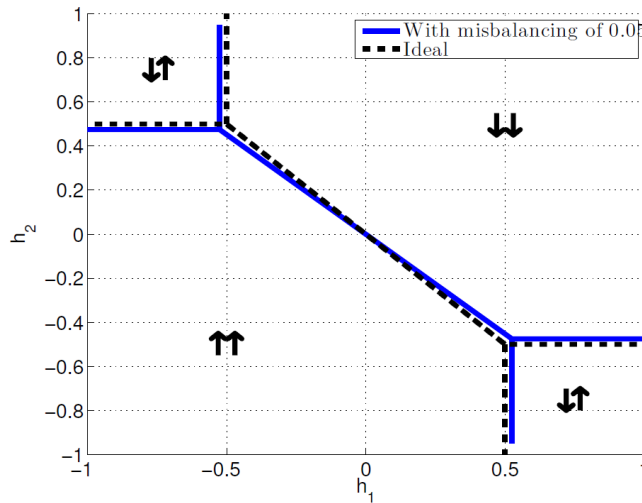


Figure 3.8: Two-qubit (two Ising-spin) phase diagram for two spins of different magnitude (two qubits with different I_p) for a sample χ value of $J = -0.5$. The dashed lines show the locations of the expected phase boundaries between the four possible states of the system under the assumption of identical spin magnitudes. The solid lines show the locations of the phase boundaries for spin magnitudes that differ by 0.05. Data shown are representative of D-Wave 2X systems.

3.4 Example of ICE Effects on Solution Quality

As discussed in the *Sources of ICE* section, the distributions δh and δJ depend on annealing time t_f and vary with anneal fraction s during the anneal. Because δh and δJ may vary with s (as well as with any errors in the ratio h/J), ICE can drive a system across a phase boundary—whether a quantum phase transition or a later classical phase transition.

Consider a simple two-spin problem with biases $h_1 = h_2 = 0.01$ and $J = -0.5$. We expect to see spins $\downarrow\downarrow$ (that is, $s_1, s_2 = -1$) in the problem solution. If δh is such that the QPU sees $h_1, h_2 = -0.01$ early in the anneal, the system localizes in the $\uparrow\uparrow$ state. Perhaps later, the effective ICE signal becomes smaller and the system crosses a phase boundary to prefer $\downarrow\downarrow$. Depending on when this happens in relation to freezeout time, the system may or may not be able to respond:

- If δh changes early enough in the anneal, the system can respond and provide the correct answer.
- If δh occurs too late, the early ICE is effectively locked in and both spins remain up.

This example shows that there are two components to ICE: the final error on the classical Ising spin system defined by Eqn. 2.1, and the rest of the error contributing to a variation in the anneal path on the way to the final classical Hamiltonian. Depending on problem details, either of these effects may dominate—exchanging roles earlier, later, or at multiple times in the anneal.

OTHER FACTORS AFFECTING PERFORMANCE

As well as the ICE effects discussed at length in the *ICE: Dynamic Ranges in h and J Values* chapter, a number of other factors may affect system performance, including temperature, high-energy photon flux, readout fidelity, programming errors, and spin-bath polarization. This chapter looks at each of these in turn. For guidance on how to work around some of these factors to get improved performance, see the *Usage Guidelines* chapter.

4.1 Temperature

In a formal sense, temperature is defined for a system in thermodynamic equilibrium. In a practical sense, however, if the equilibrium time of a given subsystem is very short compared to the relaxation time between subsystems, then that subsystem may not be in thermal equilibrium with other subsystems capable of exchanging energy.

For example, the superconducting QPU and the block of metal attached to both it and a separate low-temperature thermometer have distinct temperatures. The thermometer measures the temperature of the dilution refrigerator's mixing chamber; however, this may be a couple of millikelvin colder than the QPU. Furthermore, there is a distinction between the temperature of the quasiparticles in a superconducting wiring layer of the QPU and its phonons, and between the phonons of the wiring layer and those of the silicon substrate.

For timescales over which the annealing algorithm operates, the qubits are considered to be in equilibrium with a thermal bath at an effective temperature that can be measured, for example, by looking at the equilibrium distribution of single, uncoupled qubits when held in a fixed longitudinal and transverse field. (See [Figure 2.1](#) and [\[Joh2011\]](#), Supplemental Information, section II.D page 8.)

Ensuring that a QPU operates at millikelvin-scale temperatures requires minimizing the amount of energy deposited on the QPU, as well as ensuring that energy is efficiently removed from it. Nevertheless, some heat is dissipated on the QPU during normal operation, particularly during the programming cycle. This can, depending on the frequency of programming, increase the effective temperature of the qubits and therefore affect solution quality.

Temperature effects vary from QPU to QPU, and depend on a number of manufacturing-related details. In practice, the effects can be quantified by performing the aforementioned equilibrium distribution tests when no programming is being performed and again when programming is done at maximum frequency, and then comparing the values.¹

¹ This corresponds to a case where many problems, each requesting only a small number of reads, are queued to run on the QPU.

Note: Contact D-Wave [Support](mailto:dwsupport@dwavesys.com) at dwsupport@dwavesys.com to obtain the detailed properties of your system.

As a simple illustration of the effect of thermal equilibration, consider a single-qubit problem. As shown in [Figure 2.3](#), we find that single qubits freeze out at an s value of roughly 0.7. At that point, $B(s)$ —for the particular QPU shown—is approximately 7 GHz. Before this freezeout point, we assume that the single qubits are in thermal equilibrium at some temperature T after which no further dynamics are present. At $s = 0.7$, an applied h of 1.0 to the single qubit results in an energy difference between the up and down states of 7 GHz. The expected distribution for the spin up population is: $P_{\uparrow}/1 - P_{\uparrow} = e^{-\delta E/k_b T}$.

Because $\delta E = B(s)h$, we can sweep h over a small range and fit the result to a hyperbolic tangent and extract the resulting T of the QPU. [Figure 4.1](#) shows the results of a measurement where we sweep the h values on all the qubits on a QPU, shift the curves to have the same centers, and then fit the averaged data to a hyperbolic tangent to extract the effective temperature of the qubits.

As a consequence, even when the temperature of the mixing chamber is constant, the effective temperature of the qubits may vary by a few millikelvin, depending on usage.

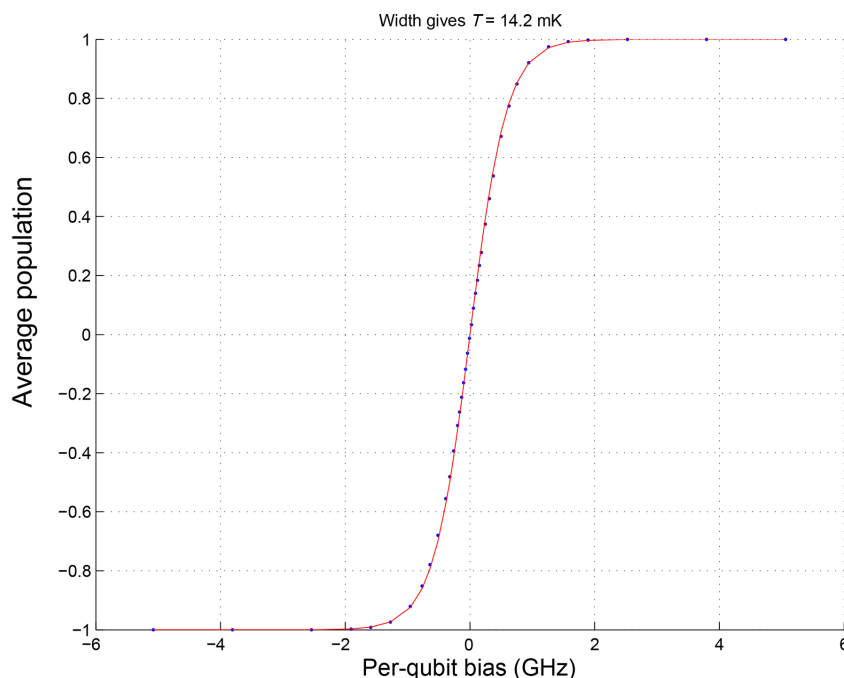


Figure 4.1: Single qubit temperature extraction: Sampling the trivial 1-qubit problem, $H = -h_i s_i$ on the QPU for qubit i (averaged over all qubits after centering each data set) for several values of h_i on the x-axis, results in the blue points shown in this plot. Fitting the data to a Boltzmann distribution gives the red curve and allows us to extract an effective temperature of 14.2 mK for this sample data. Conversion from h_i to energy bias is done by looking at the anneal schedule $B(s)$ at the single qubit freezeout point in [Figure 2.3](#).

4.2 High-Energy Photon Flux

The presence of relatively high-energy photons also plays a role. *High-energy*, in this context, means that photon energy and population exceed what would be expected at equilibrium at the measured effective qubit temperature.

These photons enter the QPU from higher-temperature stages through cryogenic filtering. When the annealing algorithm runs, they can cause transitions to much higher-energy problem states than would be expected given the equilibrium picture discussed in the previous section. Given a constant flux of high-energy photons, the probability of such a transition should increase the longer the annealing algorithm takes. This phenomenon may manifest as an appearance of solutions with energies much higher than the expected thermal distribution. This effect grows with longer anneal times.

Note: The anneal schedule feature discussed in the [Variations on the Global Anneal Schedule](#) section allows you to insert pauses at intermediate values of s . Be aware that pausing at intermediate points in the quantum annealing process may increase the probability of transitions to higher energy states. As with the standard annealing schedule, the phenomenon may manifest as an appearance of solutions with energies much higher than the expected thermal distribution of energies and the effect grows with pause time.

4.3 Readout Fidelity

Readout fidelity is not typically a significant factor in the D-Wave quantum computer. Averaged over an ensemble of randomized bit strings, the typical readout fidelity of the system is greater than 99%. That is, one out of every 100 reads of a given problem may report a solution that is different from that found by the QPU by one or more bit flips.

For guidelines on how to use multiple reads to identify readout errors and make the best use of QPU time, see the [Avoiding I/O Errors](#) section in the [Usage Guidelines](#) chapter.

4.4 Programming Errors

A problem that uses all available qubits and couplers has a greater than 90% chance of being programmed without error. Occasionally, however, programming (or reset) issues occur during the programming cycle of the QPU, moving the problem implementation on the QPU far enough away from the intended problem that the solutions computed by the QPU do not overlap the low-energy subspace of the intended problem. If low-energy answers for a single problem are critical, submit the problem at least twice.

Note: Spin-reversal transforms reprogram the system and thereby offer another way to reduce the impact of these errors; for more information on this technique, see the [Using Spin-Reversal Transforms](#) section in the [Usage Guidelines](#) chapter.

4.5 Spin-Bath Polarization Effect

One of the main sources of environmental noise affecting the qubit is magnetic fluctuations from an ensemble of spins local to the qubit wiring. This produces low-frequency flux noise that can cause the misspecification errors referred to in this document as “ICE 1.” In addition, the persistent current flowing in the qubit body during the quantum annealing algorithm produces a magnetic field that can partially align or *polarize* this ensemble of spins.

This partially polarized environment may bias the body of the qubit. This phenomenon may manifest, for example, in measurements of single qubit transition widths (the width of qubit population versus h bias). For very long anneal times, the polarized environment may add to the external h bias, producing a transition width that is narrower than the width expected from a thermal distribution. The partially polarized environment can also produce sample-to-sample correlations, biasing the QPU towards previously achieved spin configurations.

To reduce these sample-to-sample correlations, enable the *reduce_intersample_correlation* solver parameter. This setting adds optimal delay times before each anneal, giving the spin bath time to depolarize and thereby lose the effects from the previous read. It adds a delay that varies (approximately) between 200 microseconds and 10 milliseconds, increasing linearly with increasing length of the schedule:

$$delay = 500 + \frac{T(10000 - 500)}{2000} \quad (4.1)$$

where T is the total time of the anneal schedule.

Important: Enabling this parameter drastically increases problem run times. To avoid exceeding the maximum problem run time configured for your system, limit the number of reads when using this feature. For more information, see [Solver Computation Time](#).

The anneal schedule feature discussed in the [Background](#) chapter allows you to insert pauses at intermediate values of s . Be aware that pausing at intermediate points in the quantum annealing process may increase the degree of polarization of the spin environment and thus the size of the bias back on the body of the qubit.

USAGE GUIDELINES

This chapter introduces some approaches for maximizing the performance of the D-Wave QPU. For more details on some of the techniques mentioned here, see *D-Wave Problem-Solving Handbook*.

Note: For instructions on programming the system using D-Wave’s open-source Ocean tools, see the [Ocean documentation](#).

5.1 Decomposing Large Problems to Fit the QPU

To solve a problem with more variables than the available number of qubits, break the problem into subproblems, solve the subproblems, and then reconstruct an answer to the original problem from the subproblem solutions. Divide-and-conquer and dynamic programming algorithms have a rich history in computer science for problems of this type and can be adapted to map problems onto the D-Wave QPU.

Several approaches to problem decomposition, including cut-set conditioning, branch and bound, and large-neighborhood local search, are discussed in *D-Wave Problem-Solving Handbook*.

5.2 Mapping General Graphs to the QPU Graph

The D-Wave QPU minimizes the energy of an Ising spin configuration whose pairwise interactions lie on the edges of either a Chimera graph or a Pegasus graph, for the D-Wave 2000Q or Advantage respectively. To solve an Ising spin problem with arbitrary pairwise interaction structure, the corresponding graph must be *minor embedded* into a QPU’s graph; see *D-Wave Problem-Solving Handbook* for details.

5.3 Circumventing Precision Limits on h_i and $J_{i,j}$

Ising problems with high-precision parameters (h ; J) present special challenges for quantum annealers. As described in the [ICE: Dynamic Ranges in \$h\$ and \$J\$ Values](#) chapter, the problem Hamiltonian delivered to the QPU is slightly different from the one specified by the user. The quality of solutions returned by the QPU depends on the magnitude of this difference at low-energy regions of the solution space. This difference is most noticeable when the application requires optimal solutions.

For problems scaled to lie within the range available on the QPU, you can limit the parameter ranges through embedding or by simplifying the problem; see [D-Wave Problem-Solving Handbook](#) for details.

5.4 Avoiding I/O Errors

With a readout fidelity of 99%, we recommend at least two read-anneal cycles per problem to expose possible outliers. The most cost-effective use of QPU time (that is, to best amortize the fixed cost of programming the QPU against total cost), is found by taking enough reads to at least equal the programming time.

To guard against occasional programming errors, we recommend splitting a given problem into two separate QPU requests to trigger a reprogramming cycle and reduce the probability of error to less than 1%.

Contact D-Wave for specific QPU properties, which include a readout fidelity value.

5.5 Using Spin-Reversal Transforms

Applying a *spin-reversal transform* can improve results by reducing the impact of analog errors, described in the [ICE: Dynamic Ranges in \$h\$ and \$J\$ Values](#) chapter, that may exist on the QPU. A spin-reversal transform does not alter the Ising problem, in the sense that there is a direct one-to-one mapping between all solutions to the original problem to the altered problem that preserves their energies—solutions of the original problem and of the transformed problem have identical energies. Rather, the transform reverses the meanings of a collection of individual spins s_p as follows:

$$\begin{aligned} s_p &\longrightarrow s'_p = -s_p \\ h_p &\longrightarrow h'_p = -h_p \\ J_{i,j} &\longrightarrow J'_{i,j} = -J'_{i,j} \end{aligned} \tag{5.1}$$

for either $i = p$ or $j = p$.

This mapping ensures that the classical Ising spin energy given Eqn. 2.1 in the [Ising Model](#) section is unchanged; the transform simply amounts to reinterpreting spin up as spin down, and visa-versa, for a particular spin. Any number of spins may be transformed in this way, but the Ising problem remains the same and therefore we expect the same answers. A spin-reversal transform is defined according to a subset of spins for which this transform is applied.

As discussed in the *ICE: Dynamic Ranges in h and J Values* chapter, some sources of ICE such as ICE 3 and ICE 5 depend on the magnitude of the (h, J) problem parameters. Random spin transformations can be used to avoid systematic errors arising from multiple requests of the same problem, by “breaking up” these dependencies.

From a single problem defined by a set of parameters $\{h_i, J_{i,j}\}$, a sequence of spin-reversal transforms $T', T'' \dots$, each applied to a random subset of spins, defines a sequence of equivalent problems $(h', J'), (h'', J'') \dots$. All have the same energy spectrum, but their implementation on the QPU may invoke different ICE effects (some better and some worse than the original).

When studying the behavior of the QPU in solving a problem, include in the analysis a number of spin-reversal transforms of the problem. The transform samples randomly over sets of possible systematic errors. The correlations between these error sets, at least as they relate to certain error sources, can be reduced with the following considerations:

- Changing too few spins leaves most errors unchanged, and therefore has little effect.
- Changing too many spins means that most couplers connect spins that are both transformed, thus $J_{i,j}$ does not change sign. As a result, some systematic errors associated with the couplers are unaffected.

To sample over a wide range of systematic errors, consider the number of couplers and spins involved in a particular transform.

Note: Spin-reversal transforms do not necessarily remove errors due to ICE; rather, they randomize the effects of certain errors that depend on (h, J) . This creates a spectrum of error magnitudes from which the best results can be selected. Some types of systematic error—ICE 1, for example—are not affected by spin-reversal transforms.

Specify the number of spin-reversal transforms using the `num_spin_reversal_transforms` parameter when submitting a problem.

5.6 Adjusting the Problem Scale

In general, use the full range of h and J values available for the QPU when submitting a problem. To avoid having to explicitly deal with these ranges, encode the h and J settings as arbitrary floating-point numbers and set the `auto_scale` parameter to true (1, the default). This setting automatically scales the problem to make maximum use of the available ranges.

That said, using $J < -0.8$ may reduce the benefit of quantum annealing for some problem types. While no definitive model captures this effect, it might be useful to scale the overall problem energy down with a prefactor of 0.8. If you do so, disable the `auto_scale` parameter when submitting the problem.

BIBLIOGRAPHY

FLUX NOISE AND QUANTUM ANNEALING

This section characterizes the effects of flux noise on the quantum annealing process and describes the procedure that D-Wave uses to correct for drift.

A.1 Effects of Flux Noise

Let there be a flux qubit biased at degeneracy $h = 0$ with tunneling energy Δ_q . Let the qubit be subject to flux noise with noise spectral density $S_\Phi(f)$. If this qubit is subjected to experiments over a time interval t_{exp} , then it is subject to a random flux bias whose flux-flux correlator can be expressed as

$$\langle \Phi_n(t) \Phi_n(0) \rangle = \int_{f_{\min}}^{f_{\max}} df S_\Phi(f) \frac{\sin^2 2\pi f t}{(\pi f)^2} \quad (1.1)$$

where $f_{\min} = 1/t_{\text{exp}}$ and $f_{\max} = \Delta_q/h$.¹

The ambiguity with quantum annealing is in defining the appropriate choice of Δ_q , since this quantity is swept during an experiment. The value of Δ_q should be the point in the anneal at which a given qubit localizes. For a single isolated qubit, the appropriate value of Δ_q is on the order of the inverse decoherence time $1/T_2^*$, where T_2^* is a function of Δ_q . Thus, Δ_q becomes that of the coherent-incoherent crossover. For a system of coupled qubits, localization can occur at much larger values of Δ_q . In this case, the coupled qubit system can undergo a phase transition earlier in the anneal where the qubits are coherent. D-Wave QPUs realize these phase transitions at $\Delta_q/h \sim 2$ GHz.

For example, the calculation of the fractional error in the dimensionless 1-local bias h_i proceeds as follows. Typical qubits experience low-frequency flux noise characterized by a noise spectral density of the form

$$S_\Phi(f) = \frac{A}{f^\alpha} \quad (1.2)$$

with amplitude such that $\sqrt{S_\Phi(1\text{Hz})} \sim 2\mu\Phi_0/\sqrt{\text{Hz}}$ and exponent $0.75 \leq \alpha < 1$. Given that the uncertainty in α is large and that we have no experimental evidence that the form given above is valid up to frequencies of order $\Delta_q/h \sim 2$ GHz, we have used $\alpha = 1$ in our calculations. Integrating this equation from $f_{\min} = 1$ mHz to $f_{\max} = 2$ GHz yields an integrated flux noise

$$\delta\Phi_n = \sqrt{\int_{f_{\min}}^{f_{\max}} df S_\Phi(f)} \approx 10 \mu\Phi_0. \quad (1.3)$$

¹ The derivation of this expression follows the same logic as that for the phase-phase correlator given in Eq. 10c of Phys. Rev. B 67, 094510 (2003), albeit with an appropriate low-frequency cutoff.

A qubit with $\Delta_q/h = 2$ GHz also possesses a persistent current $|I_p| \approx 0.8 \mu\text{A}$. The maximum achievable antiferromagnetic coupling between a pair of qubits is $M_{\text{AFM}} \approx 2$ pH. Thus, the scale of h is set by

$$\Phi_h \equiv M_{\text{AFM}}|I_p| \approx 0.8 \text{ m}\Phi_0. \quad (1.4)$$

The relative error in the dimensionless parameter h_i is then

$$\delta h_i \equiv \frac{\delta \Phi_n}{\Phi_h} \approx 0.01. \quad (1.5)$$

A.2 Drift Correction

By default, the D-Wave system uses the following procedure to measure and correct for the longest drifts once an hour. You can disable the application of any correction by setting the `flux_drift_compensation` parameter to `false`. If you do so, we recommend that you apply flux-bias offsets manually; see the [Flux-Bias Offsets](#) section.

1. The number of reads for a given measurement, N_{reads} , is set to 2000.
2. A measurement of the zero-problem, with all $h_i = J_{i,j} = 0$ is performed, and the average spin computed for the i -th qubit according to $\langle s_i \rangle = \sum_j s_i^{(j)} / N_{\text{reads}}$, where $s_i^{(j)} \in \{+1, -1\}$ and the sum is performed over the N_{reads} independent anneal-read cycles.
3. The flux offset drift of the i -th qubit is estimated as $\delta \Phi_i = w_i \langle s_i \rangle$, where w_i is the thermal transition width of qubit i ; defined below.
4. The measured $\delta \Phi_i$ are corrected with an opposing on-QPU qubit flux-bias shift. The magnitude of the shift applied on any given iteration is capped to minimize problems due to (infrequent) large $\delta \Phi_i$ measurement errors.
5. N_{reads} is doubled, up to a maximum of 20,000.
6. The procedure repeats from step 2 at least 6 times. It repeats beyond 6 if the magnitude of any of the $\delta \Phi_i$ after the last iteration is significantly larger than the expected variation due to $1/f$ flux noise.

The thermal width, w_i , of qubit i is determined during QPU calibration by measuring the isolated qubit ($J_{i,j} = 0$ everywhere) average spin $\langle s_i(\Phi_i^{(x)}) \rangle$ as a function of applied flux bias $\Phi_i^{(x)}$ for each qubit, and fitting to the expression $\tanh [(\Phi_i^{(x)} - \Phi_i^{(0)})/w_i]$, where $\Phi_i^{(0)}$ and w_i are fit parameters.

For a typical w_i of order $100 \mu\Phi_0$, statistical error is measured at $100 \mu\Phi_0 / \sqrt{20000} \simeq 1 \mu\Phi_0$. This is much smaller than the root mean square (RMS) flux noise, which is on the order of $10 \mu\Phi_0$ for the relevant time scales.

BIBLIOGRAPHY

- [Bar1982] Barahona. “On the Computational Complexity of Ising Spin Glass Models.” *J. Phys. A* 15 (1982), pp. 3241–3253.
- [Ber1996] D. Bertsimas, C.-P. Teo, and R. Vohra. “On dependent randomized rounding algorithms.” *Oper. Res. Lett.* 24.3 (May 1996), pp. 105–114. <http://www.mit.edu/~dbertsim/papers/ApproximationAlgorithms/On%20dependent%20randomized%20rounding%20algorithms.pdf>
- [Boo2016] K. Boothby et al. “Fast clique minor generation in Chimera qubit connectivity graphs.” *Quantum Information Processing* 15.1 (2016), pp. 495–508. <https://arxiv.org/abs/1507.04774>
- [Boy2007] S. Boyd and A. Mutapcic. “Subgradient methods.” (2007). http://www.stanford.edu/class/ee364b/lectures/subgrad_method_notes.pdf
- [Bun2014] P. I. Bunyk et al. “Architectural considerations in the design of a superconducting quantum annealing processor.” *IEEE Transactions on Applied Superconductivity* 24.4 (Aug. 2014), pp. 1–10. <https://arxiv.org/abs/1401.5504>
- [Cou2009] J. Coughlan. *A tutorial introduction to belief propagation*. Tech. rep. (2009). <http://www.ski.org/tutorial-introduction-belief-propagation>.
- [Dec1987] R. Dechter and J. Pearl. “The cycle-cutset method for improving search performance in AI applications.” *Proceedings of the Third IEEE on Artificial Intelligence Applications* (1987), pp. 224–230.
- [Dickson2013] N. G. Dickson et al. “Thermally assisted quantum annealing of a 16-qubit problem.” *Nature communications*. 4:1903 (May 21, 2013).
- [DW2016_1] D-Wave Systems Inc. *Postprocessing Methods on D-Wave Systems*, 09-1109A (2016).
- [DW2016_2] D-Wave Systems Inc. *Solver Computation Time*, 09-1107A (2016).
- [Har2009] R. Harris et al. “Compound Josephson-junction coupler for flux qubits with minimal crosstalk.” *Phys. Rev. B* 80, (20 Aug. 2009). <https://journals.aps.org/prb/abstract/10.1103/PhysRevB.80.052506>.
- [Har2010] R. Harris et al. “Experimental demonstration of a robust and scalable flux qubit.” *Phys. Rev. B* 81 (13 Apr. 2010). <https://arxiv.org/abs/0909.4321>
- [Har2010_2] R. Harris et al. “Experimental investigation of an eight qubit unit cell in a superconducting optimization processor.” *Phys. Rev. B* 82, 024511 (2010)

arXiv:1004.1628

- [Joh2007] J. K. Johnson, D. M. Malioutov, and A. S. Willsky. "Lagrangian relaxation for MAP estimation in graphical models." *Proceedings of The 45th Allerton Conference on Communication, Control and Computing* (Sept. 2007).
- [Joh2010] M. W. Johnson et al. "A scalable control system for a superconducting adiabatic quantum optimization processor." *Superconductor Science and Technology* 23.6 (2010). <http://stacks.iop.org/0953-2048/23/i=6/a=065004>
- [Joh2011] M. W. Johnson et al. "Quantum annealing with manufactured spins." *Nature* 473 (May 12, 2011), pp. 194–198.
- [Kin2014] A. D. King and C. C. McGeoch. "Algorithm engineering for a quantum annealing platform." *arXiv preprint arXiv:1410.2628* (2014). <http://arxiv.org/abs/1410.2628>
- [Kin2016] A. D. King et al. "Degeneracy, degree, and heavy tails in quantum annealing." *Physical Review A* 93.5 (2016): 052320. <https://arxiv.org/abs/1512.07325>
- [Koc2004] G. Kochenberger et al. "A unified modeling and solution framework for combinatorial optimization problems." *OR Spectrum* 26 (2004), pp. 237–250. [http://leeds-faculty.colorado.edu/glover/fred%20pubs/333%20-%20xQx%20-%20Unified%20modeling%20and%20solution%20framework%20\(short\).doc](http://leeds-faculty.colorado.edu/glover/fred%20pubs/333%20-%20xQx%20-%20Unified%20modeling%20and%20solution%20framework%20(short).doc).
- [Kol2004] V. Kolmogorov and R. Zabih. "What energy functions can be minimized via graph cuts?" *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (2004), pp. 65–81. <http://www.cs.cornell.edu/rdz/Papers/KZ-ECCV02-graphcuts.pdf>
- [Liu2005] W. Liu et al. "A hybrid multi-exchange local search for unconstrained binary quadratic program, University of Mississippi, Hearin Center for Enterprise Science, HCES-09-05, 2005.
- [Mar2007] R. Marinescu and R. Dechter. "Best-first AND/OR search for 0-1 integer linear programming." *Proceedings of the 4th International Conference on Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR)* (2007). <http://www.ics.uci.edu/~radum/papers/cpaior07-aobf.pdf>.
- [Pud2014] K. L. Pudenz et al. "Error-corrected quantum annealing with hundreds of qubits." *Nature communications* 5 (2014). <http://www.nature.com/ncomms/2014/140206/ncomms4243/full/ncomms4243.html>
- [Pud2015] K. L. Pudenz et al. "Quantum annealing correction for random Ising problems." *Physical Review A* 91.4 (2015): 042302. <http://journals.aps.org/prabstract/10.1103/PhysRevA.91.042302>
- [Ros2016] G. Rosenberg et al. "Building an iterative heuristic solver for a quantum annealer," *Computational Optimization and Applications*, Springer (2016), pp. 1-25. <http://arxiv.org/abs/1507.07605>
- [Roy2016] A. Roy et al. "Mapping constrained optimization problems to quantum annealing with application to fault diagnosis." *Frontiers in ICT* 3 (2016): p. 14. <http://arxiv.org/abs/1603.03111>
- [Sch2009] N. N. Schraudolph and D. Kamenetsky. "Efficient exact inference in planar Ising models." *Advances in Neural Information Processing Systems* 21 (2009). <http://users.rsise.anu.edu.au/~dkamen/nips08.pdf>.