D-Wave
The Quantum Computing Company™

Boosting integer factoring performance via quantum annealing offsets

TECHNICAL REPORT

Evgeny Andriyash, Zhengbing Bian, Fabian Chudak, Marshall Drew-Brook, Andrew D. King, William G. Macready, Aidan Roy

2016-12-09

Overview

D-Wave quantum computing systems now allow a user to advance or delay the annealing path of individual qubits through the *anneal offsets* feature. Here we demonstrate the potential of this feature by using it in an integer factoring circuit. Offsets allow the user to homogenize dynamics of various computational elements in the circuit. This gives a remarkable improvement over baseline performance, in some cases making the computation more than 1000 times faster.

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

14-1002A-B
D-Wave Technical Report Series

Notice and Disclaimer

Summary

Mapping a boolean circuit or optimization problem onto a D-Wave system typically requires the use of *chains*; that is, groups of qubits representing the same logical variable. Chains of different length produce different quantum tunneling dynamics, as longer chains have a lower effective tunneling energy and may *freeze out* in a fixed state earlier in the anneal. To balance these disparate tunneling dynamics, we can boost the relative tunneling energy of longer chains by delaying the anneal of their qubits. 2000-qubit D-Wave systems allow the user to advance or delay the annealing path of individual qubits through the *anneal offsets* feature.

Here we review a demonstration of this capability, as applied to the problem of integer factoring: given an input number p, find integers a and b such that $a \times b = p$. The 1994 formulation of Shor's algorithm, along with the ubiquity of factoring in cryptographic systems, makes this a problem of central interest in the context of quantum computing.

One way to factor an integer using quantum annealing is to run a multiplication circuit backwards. We begin by setting up a minimization problem representing a multiplication circuit. This circuit has inputs a and b, and output p, and can achieve an optimal state when $a \times b = p$. If we specify a and b but not p, then the solution to the minimization problem tells us the value of p. If instead we specify p but not a or b, the solution gives us a and b such that $a \times b = p$.

Anneal offsets give a remarkable improvement in performance for these problems, in some cases making the computation more than 1000 times faster. Figure 1 shows performance of a D-Wave system on a testbed of factoring problems, run both with and without anneal offsets.

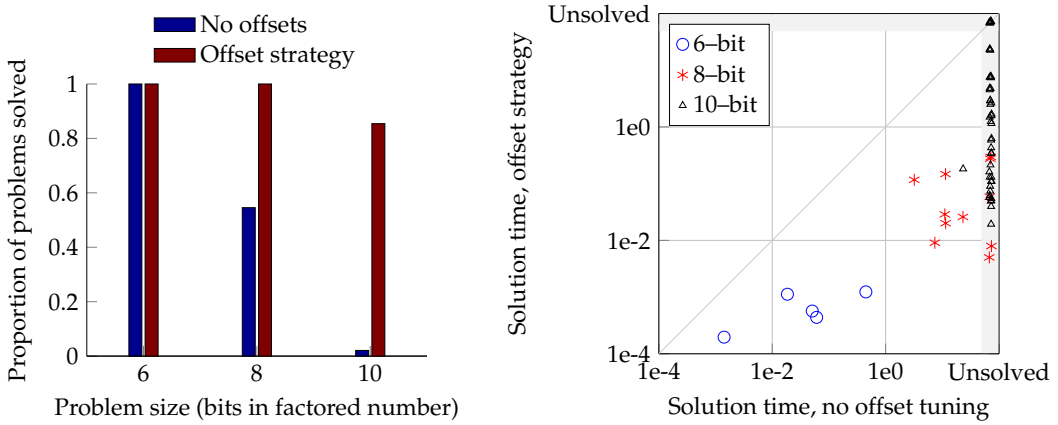


Figure 1: Performance of a 2000-qubit D-Wave system on integer factoring problems improves dramatically when anneal offsets are used. (a) Percentage of instances solved using 250,000 samples per instance, when factoring 6-, 8-, and 10-bit semiprimes. (b) Time required to find a solution for each instance.

Contents

1 Anneal offsets

Ising model quantum annealing can be modeled by the Hamiltonian

$$\boldsymbol{H}(s) = A(s)\boldsymbol{H}_0 + B(s)\boldsymbol{H}_P,$$

which evolves over time s from an initial Hamiltonian $\boldsymbol{H}_0 = \sum_i \boldsymbol{\sigma}_x^{(i)}$ at $s = 0$ to a final Hamiltonian $\boldsymbol{H}_P = \sum_i h_i \boldsymbol{\sigma}_z^{(i)} + \sum_{i<j} J_{i,j}\, \boldsymbol{\sigma}_z^{(i)} \otimes \boldsymbol{\sigma}_z^{(j)}$ at $s = 1$. If the system evolves slowly enough, it remains in its ground state throughout, and the ground state of $\boldsymbol{H}_P$ encodes the solution to the classical energy minimization problem of the Ising model $[h, J]$. A typical schedule for the tunneling energy $A(s)$ and problem energy $B(s)$ is shown in Figure 2(a). In the 2000-qubit D-Wave system, the default $A(s)$ and $B(s)$ is identical for every qubit. However, this schedule may be modified on a per-qubit basis, allowing for alternative schedules that better optimize performance.

By default, the D-Wave schedule is controlled by a global, time-dependent signal $c(s)$ that simultaneously determines $A(s)$ and $B(s)$. The anneal offsets feature perturbs this signal at each qubit i by an offset δc_i, so that the scheduling signal at time s on qubit i is $c_i(s) = c(s) + \delta c_i$. This offset has the effect of giving each qubit its own schedule $[A_i(s), B_i(s)]$, which may be advanced or delayed from the global schedule $[A(s), B(s)]$, resulting in a modified Hamiltonian

$$\boldsymbol{H}(s) = \sum_i A_i(s)\boldsymbol{\sigma}_x^{(i)} + \sum_i B_i(s) h_i \boldsymbol{\sigma}_z^{(i)} + \sum_{i<j} \sqrt{B_i(s)B_j(s)}\, J_{i,j}\, \boldsymbol{\sigma}_z^{(i)} \otimes \boldsymbol{\sigma}_z^{(j)}. \tag{1}$$

Figure 2(b) shows the effect of these offsets: if $\delta c_i > 0$, the schedule is advanced, so $A_i(s) < A(s)$ and $B_i(s) > B(s)$.

While modifying the annealing schedule on a per qubit basis allows a great deal of flexibility for optimization, the perturbation from $B(s)$ to $B_i(s)$ also introduces a new source of error: the individual h and J terms of the Ising model in (1) are no longer amplified uniformly during the anneal. This means that the benefits of individual anneal schedules must be balanced against larger problem misspecification.

1.1 Chain dynamics

One particularly valuable application of the anneal offset feature is homogenizing the dynamics of chains. A *chain* is a collection of qubits intended to act as a single logical spin; to impose this behavior, we apply a strong ferromagnetic coupling ($J = -1$) between those qubits. The *length* of the chain is the number of qubits in it.

If we impose a chain coupling $J_{ij} = -1$ between qubits i and j, and express the problem Hamiltonian at a smaller energy scale, then the Ising model energy is minimized when i and j take the same spin; therefore we may treat i and j as a single logical qubit, call it q_{ij}. However, at any point in the anneal, the effective tunneling energy of q_{ij} is smaller than that of a single qubit. Intuitively, two qubits are less likely to tunnel simultaneously than just one in isolation. More precisely, at time s, if the tunneling energy and problem energy for a single qubit are $A(s)$ and $B(s)$, then the effective tunneling energy of q_{ij} is

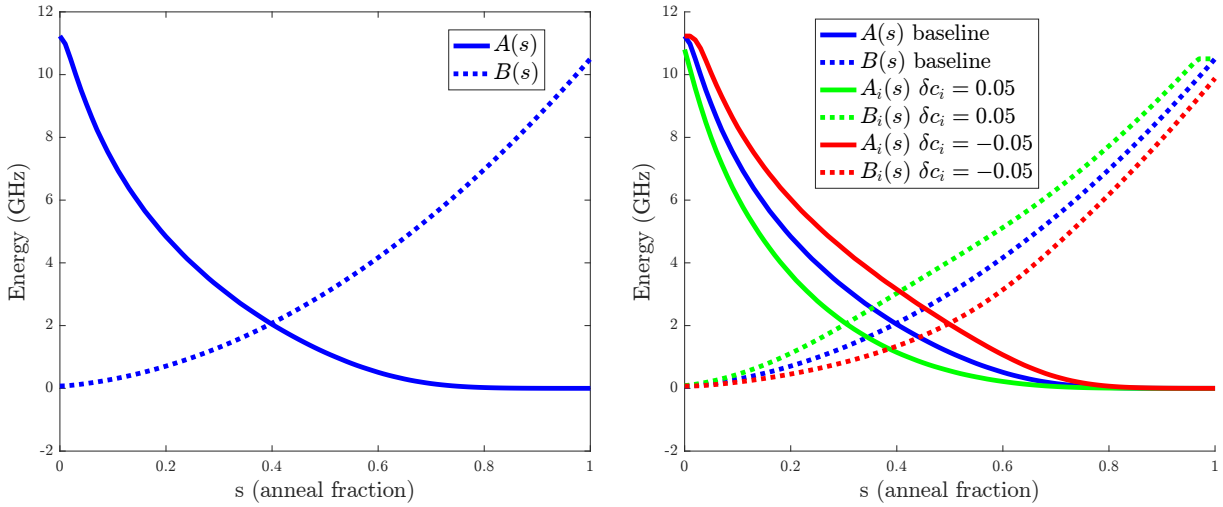


Figure 2: Annealing schedules. (a) Typical anneal parameters $A(s)$, $B(s)$. Annealing begins at $s = 0$ with $A(s) \gg B(s)$ and ends at $s = 1$ with $A(s) \ll B(s)$. (b) Anneal parameters with offsets. The baseline curve ($\delta c_i = 0$) is shown along with $A_i(s)$, $B_i(s)$ for a qubit that is advanced ($\delta c_i = 0.05$) or delayed ($\delta c_i = -0.05$).

proportional to

$$\sqrt{\frac{B(s)^2}{4} + A(s)^2} - \frac{B(s)}{2}. \tag{2}$$

(See [1] for details.) For $k > 2$ qubits acting as a single logical variable, the effective tunneling energy is reduced even further.

As a result of reduced tunneling energy, chains suffer from early *freeze-out*: longer chains become fixed to a particular spin value earlier in the anneal than shorter chains. This effect usually has a negative impact on performance. However, anneal offsets can mitigate this effect: by delaying the anneal for longer chains, their tunneling energy is increased and brought in line with others.

Using perturbation theory and the particular schedule of the D-Wave system, we can compute the anneal offsets required to synchronize the effective tunneling energy across all chain lengths at a fixed time s during an anneal (see Appendix A). The delay on an isolated chain of k qubits is reasonably well approximated by a function of the form $f(k) = \beta^{\frac{1-k}{k}} - 1$, where $0 < \beta < 1$ depends on the time s at which the tunneling energies are synchronized (see Figure 3).

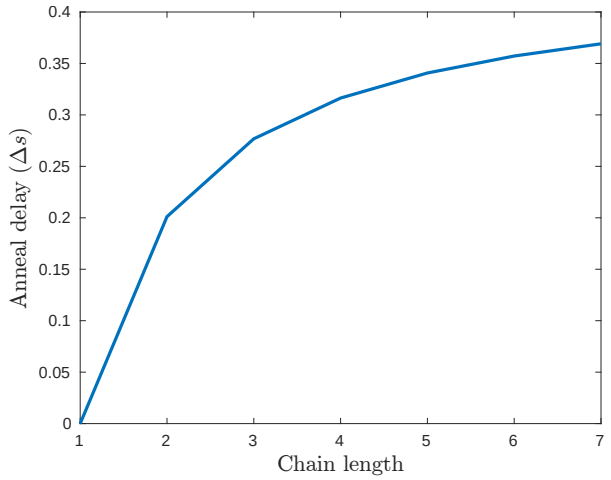


Figure 3: Anneal offset delay $f(k) = \beta^{\frac{1-k}{k}} - 1$ as a function of chain length k. Shown is $\beta = \log 2$.

2 Factoring

As an application to measure the performance benefits of anneal offsets, we consider the problem of factoring integers. This is a problem whose presumed hardness is the basis for many cryptographic protocols. While integers can be factored in polynomial time on a universal quantum computer using Shor's algorithm [2], no efficient classical algorithm is known. Our approach (introduced in [3], see also [4, 5]) is completely different from Shor's algorithm and offers no guarantee of a solution in polynomial time. It does demonstrate a way that factoring can be performed on D-Wave systems and is used to test the benefit of anneal offsets.

We wish to factor an integer p as a product of a pair of n-bit integers a and b. To solve this problem, we represent a, b, and p in binary and construct a Boolean circuit that performs n-bit by n-bit multiplication. We then encode the circuit as an optimization objective, so that input/output bit strings that satisfy the circuit have a known minimum energy (and all other bit strings have higher energy). Finally, by fixing the output of the circuit to the binary representation of p and minimizing the resultant optimization objective, we can obtain the inputs giving the desired output, effectively running the circuit in reverse.

More generally, for any decision problem, we imagine encoding the Boolean circuit that verifies the answer as either true or false, clamping the output of the circuit to true, and minimizing. For NP problems this circuit is polynomially sized. (However, factoring a $2n$-bit integer requires $\theta(n^2)$ qubits: integers of cryptographic interest are beyond the scale of 2000-qubit D-Wave systems.) Clamping may be effected by appropriate local fields acting on the output bits, or by eliminating the output bits entirely and adding their contributions to their neighbors.

				a_3	a_2	a_1	a_0
				b_3	b_2	b_1	b_0
				a_3b_0	a_2b_0	a_1b_0	a_0b_0
			a_3b_1	a_2b_1	a_1b_1	a_0b_1	
		a_3b_2	a_2b_2	a_1b_2	a_0b_2		
	a_3b_3	a_2b_3	a_1b_3	a_0b_3			
p_7	p_6	p_5	p_4	p_3	p_2	p_1	p_0

Figure 4: Multiplication of two 4-bit integers (a_3, a_2, a_1, a_0) and (b_3, b_2, b_1, b_0) to form the 8-bit product $(p_7, p_6, p_5, p_4, p_3, p_2, p_1, p_0)$.

2.1 Multiplication circuits as constraint satisfaction

Consider the multiplication of two 4-bit integers. Schematically, the product is formed from the multiplication table given in Figure 4. Any output bit p_i is formed as the sum of appropriate products of the bits of a and b. For example, the first output bit is given by

$$p_0 = a_0 b_0.$$

This imposes a constraint between inputs a_0, b_0 and the output p_0. This constraint, which we denote $C_\wedge(a_0, b_0, p_0)$, is the AND gate of digital logic.

To represent a sum of bits, we use digital adders. A half adder, denoted $C_{2A}(t, t', p, c)$, is a constraint with inputs t, t' and outputs s, c (sum and carry), enforcing the constraint $t + t' = s + 2c$. The full adder constraint, denoted $C_{3A}(t, t', t'', p, c)$ enforces the constraint $t + t' + t'' = p + 2c$ (see Figure 5 for truth tables). Arranging half and full adders in a sequence, we can enforce the summation constraints on the output bits p imposed by the columns of the multiplication table in Figure 4.

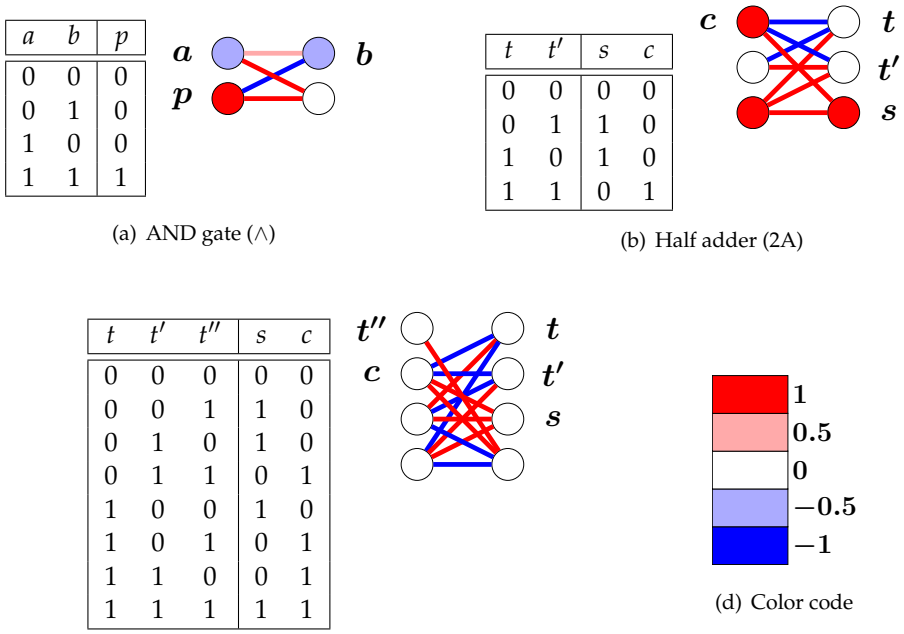
A network of constraints representing the full multiplication circuit is shown in Figure 6, with optimization variables represented as edges. In addition to the variables representing inputs a and b and outputs p, there are a number of intermediate variables. The s^i_j and c^i_j variables represent the sum and carry bits from adder constraints respectively, while the $t_{i,j}$ variables represent the products $a_i b_j$.

2.2 Constraints as optimization objectives

Having reduced integer factoring to a constraint satisfaction problem, we next reduce constraint satisfaction to Ising model optimization.

Given a constraint $C(\boldsymbol{x})$ defined over a set of Boolean variables $\boldsymbol{x} \in \{0, 1\}^n$, we identify $\boldsymbol{x}$ with spin variables $\boldsymbol{s} = 2\boldsymbol{x} - \boldsymbol{1} \in \{-1, 1\}^n$, and represent $C(\boldsymbol{x})$ by an Ising model whose ground states coincide with the feasible configurations of the constraint. Typically, this requires the use of ancillary variables. We write a spin configuration as $\boldsymbol{z} = (\boldsymbol{s}, \boldsymbol{a})$, where $\boldsymbol{s}$ and $\boldsymbol{a}$ are the constraint and ancillary variables respectively, and denote energy of an Ising model $[h, J]$ at spin configuration $\boldsymbol{z}$ as

$$\mathbb{E}_{[h,J]}(\boldsymbol{z}) = h^T \boldsymbol{z} + \boldsymbol{z}^T J \boldsymbol{z}.$$



a	b	p
0	0	0
0	1	0
1	0	0
1	1	1

(a) AND gate ($\wedge$)

t	t'	s	c
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

(b) Half adder (2A)

t	t'	t''	s	c
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

(c) Full adder (3A)

(d) Color code

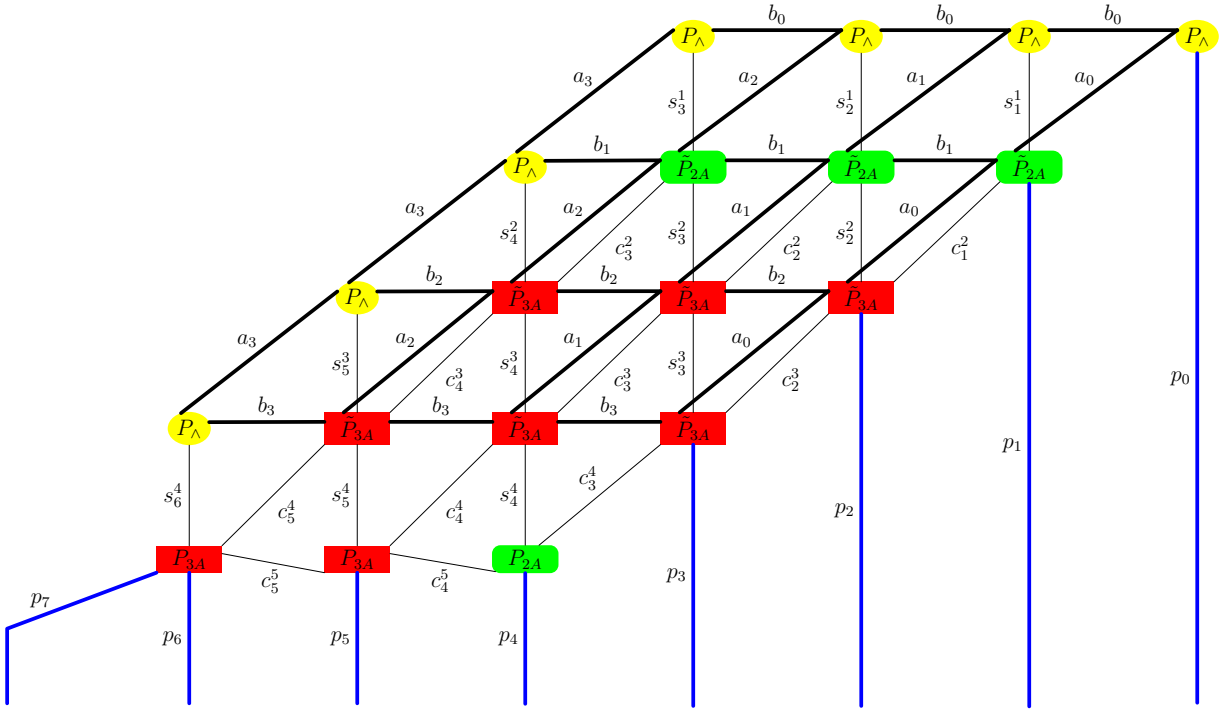
Figure 5: Truth tables and Ising penalty models for the (a) AND gate, (b) half adder, and (c) full adder. Colors shown in (d) identify the different h and J values for each Ising model.

To separate feasible and infeasible solutions in $C(\boldsymbol{x})$, we require that for some global minima e_0 and positive energy gap g:

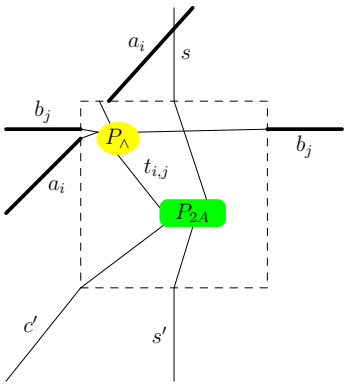
$$\min_{\boldsymbol{a}} \mathbb{E}_{[h,J]}(\boldsymbol{s},\boldsymbol{a}) = \begin{cases} e_0 & \text{if } C(\boldsymbol{x}) \text{ is satisfied (where } \boldsymbol{x} = \frac{\boldsymbol{s}+\mathbf{1}}{2}\text{);} \\ \geq e_0 + g & \text{if } C(\boldsymbol{x}) \text{ is not satisfied.} \end{cases}$$

The multiplication constraints $C_\wedge$, C_{2A}, and C_{3A} can be realized as Ising penalty models $P_\wedge$, P_{2A}, P_{3A}, as shown in Figure 5. These models, found using the techniques in [6], were constructed to match the graph structure of the D-Wave system. The available Ising model interactions form a *Chimera* graph, consisting of a 2-dimensional grid of interconnected unit cells, where each unit cell is a $K_{4,4}$ complete bipartite graph.

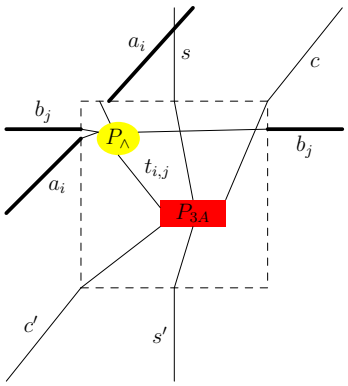
After penalty models are defined for all constraints in the circuit, the models are placed onto disjoint subgraphs of the Chimera graph. Variables may occur in multiple constraints, but we connect the different instantiations of a variable together using chains. The constraints of the multiplication circuit in Figure 6 have a natural grid-like layout; this fits well with the Chimera grid structure. Figure 7 shows a 3-bit multiplication circuit completely embedded onto a Chimera graph.



(a) 4-bit by 4-bit multiplication circuit



(b) Box $\tilde{P}_{2A}$: half adder detail



(c) Box $\tilde{P}_{3A}$: full adder detail

Figure 6: The constraint network (a) represents the multiplication of two 4-bit numbers. Boxes labeled $\tilde{P}_{2A}$ and $\tilde{P}_{3A}$ in (b) and (c) respectively represent pairs of constraints (an AND gate with a half adder or full adder).

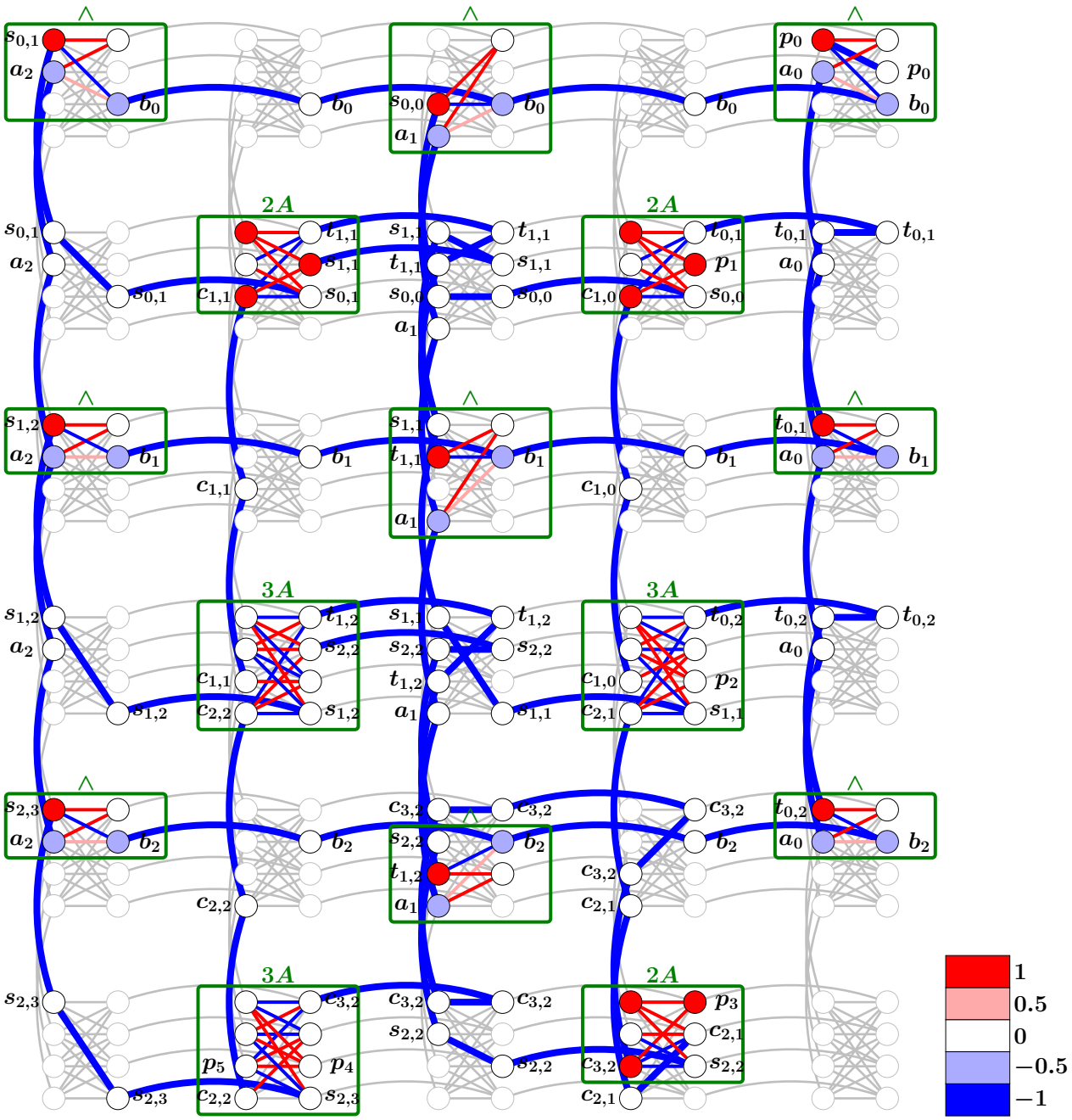


Figure 7: A 3-bit multiplication circuit mapped onto D-Wave's Chimera graph. Constraints circled in green labeled $\wedge$, $2A$, and $3A$ represent AND, half adder, and full adder respectively. Chains of logical variables are indicated with thick blue lines. In a working D-Wave system, this embedding is modified to avoid missing qubits. Input variables (labeled a and b) have chains that span the length or width of the graph; the slower tunneling dynamics of these chains are mitigated using anneal offsets.

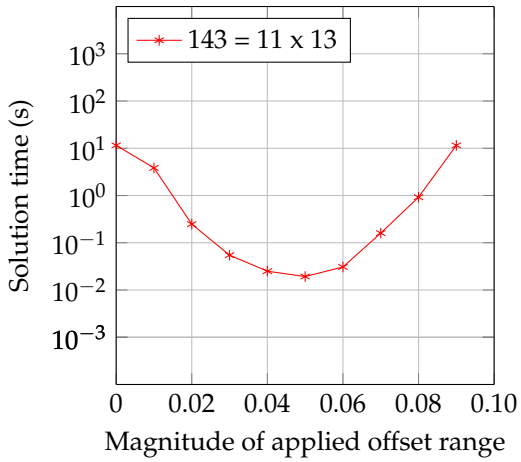


Figure 8: Performance of the 2000-qubit D-Wave system using different magnitudes of anneal offsets for the factoring problem $143 = 11 \times 13$. For each offset magnitude, the D-Wave system was sampled $250,000$ times using $20\,\mu s$ per anneal. Given a fraction p of returned samples in the ground state, solution time (shown in seconds) is defined as $(20\,\mu s)\frac{\log(0.99)}{\log(1-p)}$; that is, the time required to generate a ground state with at least 99% confidence.

3 Results

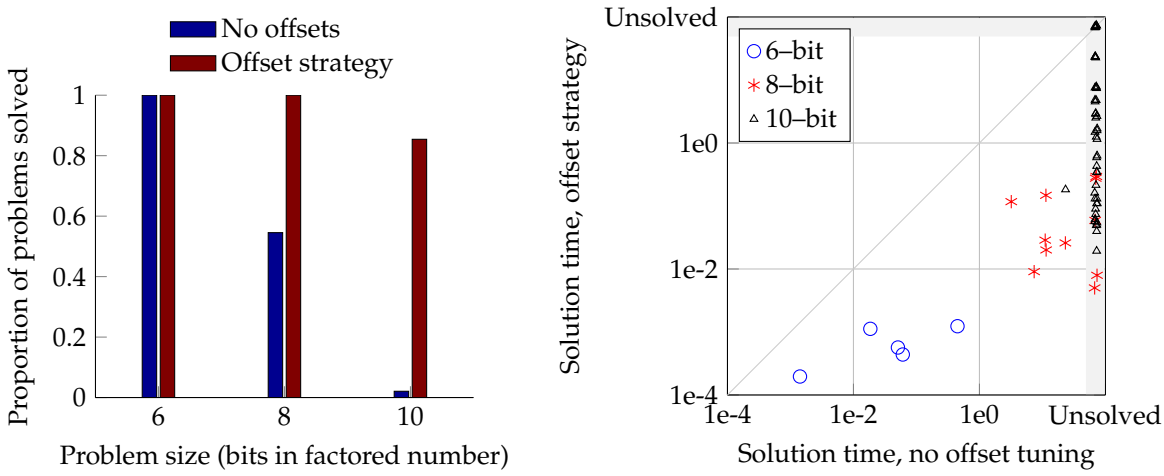
In this section, we apply the anneal offsets feature to the embedded integer factoring problems of the previous section, in an attempt to mitigate the issues of chain dynamics.

We consider factoring $2n$-bit integers using an n-bit by n-bit multiplication circuit, for each of $n \in \{3,4,5\}$. For each n, we consider all *semiprime* factoring problems: that is, we attempt to factor every integer of the form $p = a \times b$, where a and b are primes of at most n bits, and p has more than n bits. Semiprimes represent the most difficult factoring problems because they have the fewest factors.

Rather than attempting to find the optimal offset for each chain in the embedded problem, we choose a single offset function determined by the chain length.

For chains of length k, we use a delay of $f(k) = \alpha\big((\log 2)^{\frac{1-k}{k}} - 1\big)$, where the scaling parameter $\alpha \geq 0$ is determined experimentally. Choosing $\alpha = 0$ results in no offsets, which is the default behavior of the system. Varying the magnitude α allows us to balance an increasing effect from anneal offsets against the increasing error as a result of the perturbations to the Ising model. Figure 8 shows an example of how the performance of the D-Wave system varies with α for one particular factoring problem.

Sweeping over the available offset magnitudes, we can choose the best α for each factoring instance. Figure 9 compares the performance of the D-Wave system with the best anneal offset magnitude to the performance with no anneal offset. Most problems at the largest scale considered (10-bit semiprimes) were solved in 250,000 samples only when anneal offsets were used. For instances that were solved both with and without anneal offsets, using anneal offsets typically reduced the time required to find a solution by 2 to 3 orders of magnitude.



(a) (b)

Figure 9: Performance of the 2000-qubit D-Wave system on factoring all 6-, 8-, and 10-bit semiprimes, with and without anneal offsets. (a) Percentage of instances solved using 250,000 samples per instance. (b) Time required to find a solution for each instance. Points below the diagonal show a performance benefit from employing the anneal offset strategy.

In general, different problems will have different optimal offset magnitudes, even when those instances use the same chains. However, the pattern in Figure 8 is fairly typical. Choosing too large an offset magnitude can overcompensate for the effective tunneling energies as well as increase the misspecification of the Ising model. Figure 10 shows the median time to solution across all problem instances, using various anneal offset magnitudes.

4 Discussion

Anneal offsets are a newly available feature in D-Wave 2000-qubit quantum annealing systems. We have demonstrated a class of problems for which anneal offsets greatly improve performance, both in the number of problems solved and the frequency with which optimal solutions appear. Anneal offsets are not always necessary, and are likely not always effective, but promise an avenue for mitigating computational bottlenecks in inputs with certain structural characteristics.

Here we have determined good offset values by looking at chains as isolated systems, whose dynamics can be synchronized effectively using a theoretical model. This strategy may work particularly well for factoring because of the regular structure of the embeddings used. These problems have a wide range of chain lengths (long chains for input variables and single-qubit chains for ancillary variables), but a relatively small number of distinct chain lengths (all input variables have the same chain length, as do all ancillary variables). For problems with a wider variety of chain lengths, homogenization via anneal offsets may be less effective.

The regular structure of the constraint satisfaction problem may also play a role in the

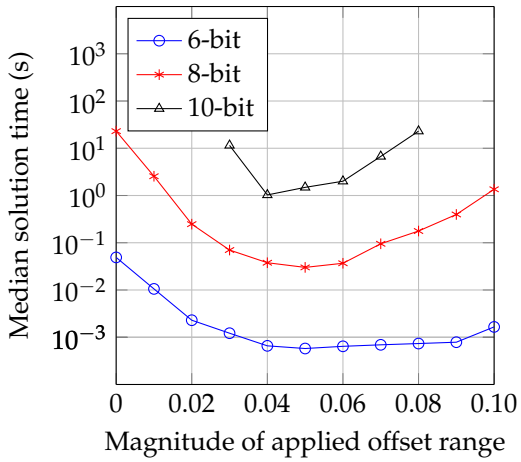


Figure 10: Median time to solution across all problems, for various anneal offset magnitudes.

dramatic performance improvements that we have observed. Every input variable in an $n \times n$-bit multiplication circuit is contained in exactly n constraints, and those constraints form a grid-like pattern. As a result, the optimal choice of tunneling energies might be uniform across all input variables. In contrast, Farhi et al. [7] suggest that nonuniform tunneling amplitudes may be preferable when variables are incident with disparate numbers of constraints. In general, determining an optimal set of anneal offsets for a given optimization problem is difficult and will be the subject of continued research.

References

[1] T. Lanting, R. Harris, J. Johansson, M. H. S. Amin, A. J. Berkley, *et al.*, "Cotunneling in pairs of coupled flux qubits," *Phys. Rev. B*, vol. 82, p. 060 512, 6 2010.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, 1997.

[3] C. J. Burges, "Factoring as optimization," *Microsoft Research Reports*, vol. MSR-TR-2002-83, 2002.

[4] G. Schaller and R. Schützhold, "The role of symmetries in adiabatic quantum algorithms," *Quantum Information & Computation*, vol. 10, pp. 109–140, 1 2010.

[5] R. Dridi and H. Alghassi, "Prime factorization using quantum annealing and computational algebraic geometry," *ArXiv:1604.05796*, 2016.

[6] Z. Bian, F. Chudak, R. Israel, B. Lackey, W. G. Macready, and A. Roy, "Discrete optimization using quantum annealing on sparse Ising models," *Frontiers in Physics*, vol. 2, no. 56, 2014, ISSN: 2296-424X.

[7] E. Farhi, J. Goldstone, S. Gutmann, and M. Sipser, "Quantum computation by adiabatic evolution," *ArXiv:quant-ph/0001106*, 2000.

A Derivation of the anneal delay function for chains

Using perturbation theory, we can write the expression for the tunneling energy of a chain of length k as

$$A^{(k)} = A \sum_{p \in \mathrm{P}_{+-}} \left(\frac{A}{B}\right)^{|p|-1} \frac{1}{\prod_{x_{i\neq 0} \in p} (\boldsymbol{H}_P(x_i) - \boldsymbol{H}_P(x_0))},$$

where P_{+-} is a set of paths in the state space connecting classical ground states of the chain (all $+$ and all $-$ states), x_i are states along the path, and x_0 is one of the ground states. When the transverse field A becomes small compared to the energy scale B, the leading contribution has the form

$$A^{(k)} = 2A \left(\frac{A}{B}\right)^{k-1} \frac{1}{\prod_{i\neq 0 \in p_{+-}} (\boldsymbol{H}_P(x_i) - \boldsymbol{H}_P(x_0))} + O\left(\frac{A^{k+1}}{B^{k+1}}\right),$$

where p_{+-} denotes the path connecting two ground states by consecutively flipping spins from one end of the chain to the other. Because the energy penalty of states along this path are all equal to 2 (we have chosen $J = -1$ as chain coupling), the above expression simplifies to

$$A^{(k)} = 2A \left(\frac{A}{2B}\right)^{k-1} + O\left(\frac{A^{k+1}}{B^{k+1}}\right).$$

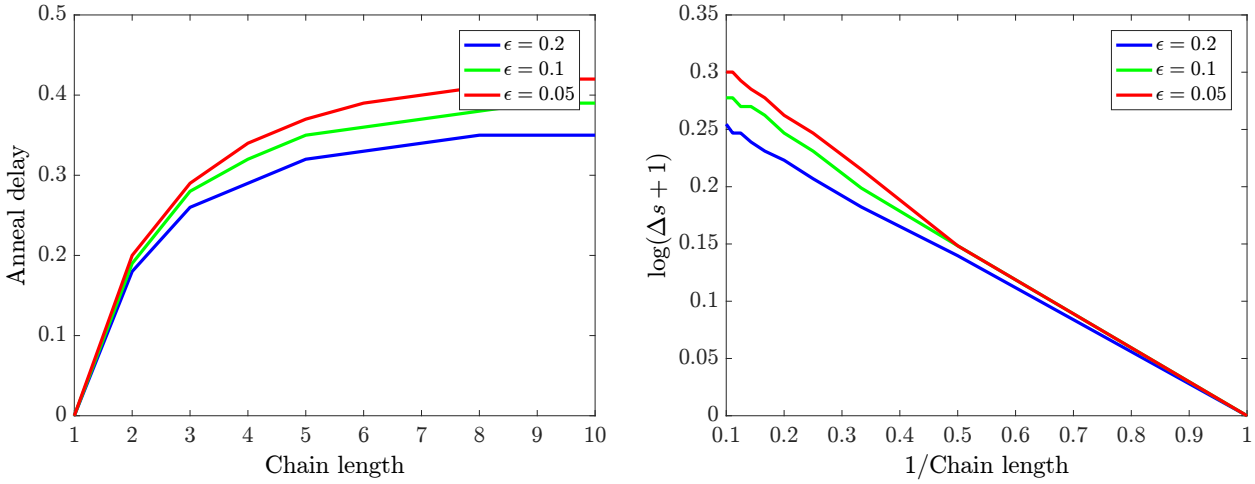
Tunneling energy varies during the anneal and we can heuristically determine the freeze-out time $s^{(k)}$ by equating this energy to some small value:

$$s^{(k)} : \; A^{(k)}(s^{(k)}) \approx 2A(s^{(k)}) \left(\frac{A(s^{(k)})}{2B(s^{(k)})}\right)^{k-1} = \epsilon.$$

Using a particular annealing schedule of one of the D-Wave systems, we get offsets between chain freeze-out times $s_\epsilon^{(k)} - s_\epsilon^{(1)}$ depicted on Figure 11(a). In all cases, the dependency of anneal offset on chain length can be approximately parametrized as

$$\Delta s^{(k)} = s^{(1)} - s^{(k)} = \beta^{\frac{1-k}{k}} - 1,$$

which is supported by the dependency of $\log\left(\Delta s^{(k)} + 1\right)$ on $\frac{1}{k}$ shown in Figure 11(b). For the given choices of ϵ, parameter β is in the range $\beta \sim 0.69 - 0.76$. We choose the value $\beta = \log 2$ in the main text.



(a) (b)

Figure 11: (a) Anneal offset delay $\Delta s^{(k)}$ as a function of chain length k. (b) Nearly linear dependency of $\log\left(\Delta s^{(k)} + 1\right)$ on $\frac{1}{k}$.