The Quantum Computing Company™

Advantage Processor Overview

TECHNICAL REPORT

Catherine McGeoch and Pau Farré

2022-01-17

Overview

This report presents a high-level overview of features and performance of D-Wave's Advantage quantum processing units (QPUs). These QPUs can hold application inputs that are three times larger, on average, than those that fit on previous-generation D-Wave 2000Q QPUs. Beyond the capacity to read bigger problems, Advantage QPUs deliver significantly better performance than D-Wave 2000Q processors on application-relevant inputs. Furthermore, the "performance update" series of Advantage QPUs released in 2021 outperforms the original Advantage series launched in 2020. We attribute this improved performance to innovations in chip design as well as new fabrication materials and processes.

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

14-1058A-A
D-Wave Technical Report Series

Notice and Disclaimer

Summary

This report presents a high-level overview of D-Wave's Advantage™ quantum computing systems. Technological advances in the design of the quantum processing unit (QPU) at their core make Advantage systems by far the largest and most powerful quantum computers in existence today:

- The Advantage QPU contains at least 5,000 qubits, about 2.5 times more than its predecessor, the D-Wave 2000Q QPU. The number of couplers per qubit has increased from 6 to 15, totaling at least 35,000 couplers, about a six-fold increase over earlier-generation systems.
- More qubits and couplers means that larger application problems can be solved directly on the new systems. Advantage QPUs can hold inputs that are on average three times larger than similarly structured inputs that fit on D-Wave 2000Q QPUs.
- Furthermore, application problems can be mapped more compactly onto the Advantage QPUs, where compactness is measured by *chain length*. Chains on Advantage QPUs are typically less than half as long as chains on D-Wave 2000Q QPUs.

More compact embeddings in Advantage QPUs are associated with better-quality solutions, and faster runtimes to achieve same-quality solutions, compared to 2000Q QPUs. Additional performance improvements are observed due to innovations in fabrication materials and processes.

To illustrate these points, this report presents results from case studies comparing performance of Advantage and 2000Q systems. Here are some highlights:

- On clique problems, in cases where both QPUs found (putative) optimal solutions, Advantage was up to 54 times faster in terms of pure anneal time (about 14 times faster in wall-clock time). In cases with no ties in solution quality, Advantage solutions were better 87% of the time.
- On inputs for the NAE3SAT problem, Advantage again outperformed the 2000Q. In one test, on cases with no ties in solution quality, the Advantage solver found better solutions 94% of the time.
- On inputs for 3D lattice problems, the Advantage solver found optimal solutions about 10 times faster than the 2000Q solver on largest inputs, considering pure anneal times.
- A small comparison study demonstrated additional performance gains from the Advantage performance update series, first released in October 2021, over the original Advantage QPU launched in September 2020.

Contents

1 Introduction

This report presents a high-level overview of the Advantage generation of quantum processors, summarizing results from previous technical reports published with the launch of the first Advantage system in September 2020 and the release of the Advantage "performance update" series in October 2021 [1, 2]. The newer Advantage systems incorporate several innovations in fabrication materials and processes, and exhibit even better performance than their predecessors from the previous year.

The quantum processing units (QPUs) at the core of Advantage systems represent a new pinnacle of design innovation and technological advances brought about by D-Wave engineers and scientists. Advantage processors are by far the largest and most performant quantum computers in existence today. The key results in this report are summarized as follows.

Advantage QPUs can represent bigger problems. Section 2 describes the main design features of the Advantage QPU in comparison to its predecessor, the D-Wave 2000Q QPU. Advantage QPUs are based on the *Pegasus* graph topology, which replaces the *Chimera* topology found in all previous D-Wave processors, including the 2000Q.

Advantage QPUs contain at least 5,000 qubits, about 2.5 times more than found in 2000Q QPUs. Furthermore, the number of couplers per qubit increased from 6 in Chimera to 15 in Pegasus: thus, Advantage QPUs contain at least 35,000 couplers in total, about a six-fold increase over that in 2000Q systems.

This increase in qubit count and connectivity means that larger application problems can be solved directly on the quantum chip. Advantage QPUs can hold inputs that are on average 3 times larger than similarly-structured inputs held by 2000Q QPUs. Furthermore, application problems can be embedded more compactly onto Advantage QPUs (more compact embeddings are associated with better-quality solutions). Compactness is measured by *chain length*: we observe that chains on Advantage QPUs are typically half as long as chains on 2000Q QPUs.

Advantage solvers outperform 2000Q solvers. We use the term solver to refer to a specific QPU, its control system, and (in some contexts) software utilities that may be used. Section 3 presents results from three case studies using specific solvers from the 2000Q and the Advantage product lines. Performance is tested on a variety of input classes and solutions, all leading to the same conclusion: Advantage solvers return significantly better-quality solutions in equivalent time frames and are faster at returning same-quality solutions.

Here are some highlights:

- For clique problems, on inputs where both solvers found (putative) optimal solutions, the Advantage solver was up to 54 times faster in pure anneal time (about 14 times faster in wall clock time). In cases with no ties in solution quality, the Advantage solver found better-quality solutions 87 percent of the time.
- In a study using inputs for two variations of the Not-all-equal 3-Satisfiability (NAE3SAT) problem, the Advantage solver again showed superior performance. In cases where

there were no ties, the Advantage solver found better-quality solutions 95% of the time for one variation and 75% of the time for the other.

- On inputs for 3D lattice problems, the Advantage solver found optimal solutions up to 5 times faster than the 2000Q solver in the median case, and 100 times faster on the 90th-percentile case (considering pure anneal times).
- Comparison of an original Advantage solver from 2020 and an Advantage performance update solver from 2021 shows that the latter can embed inputs that are between 4 and 49 percent larger. In all tests using all three of the above-mentioned input classes, the newer Advantage solver consistently finds better-quality solutions than the original Advantage solver.

The results are unequivocal: beyond the capability to hold larger and more complex inputs than 2000Q solvers, Advantage solvers show better performance on a variety of inputs and metrics, returning better-quality solutions when given the same time limits, and finding same-quality solutions faster.

We attribute these performance gains primarily to new chip design, most importantly the switch from Chimera to the Pegasus connection topology. Additional performance enhancements may be observed due to innovations in chip fabrication, materials, and processes that were incorporated into the Advantage performance update series.

Section 4 summarizes our conclusions and considers future prospects for annealing-based quantum computers manufactured by D-Wave.

Learn more about D-Wave products and services. The Advantage and D-Wave 2000Q quantum computers, the Leap quantum cloud service, and the Ocean software developers kit (an open-source repository of tools, tutorials, and documentation), are available to the public in North America, Europe, Japan, Australia, India, and Singapore, for limited small-scale use at no cost. Larger blocks of QPU and system time are available for purchase from D-Wave or third-party providers.

The D-Wave hybrid solver service (HSS) provides users with solutions to inputs for combinatorial optimization problems that are too large to fit onto current-generation QPUs, and is available on a subscription basis. As of Fall 2021, HSS contains three hybrid solvers: the BQM solver for unconstrained binary quadratic models; the DQM solver for unconstrained quadratic models defined on discrete variables; and the CQM solver for *constrained* quadratic models defined on binary and integer variables. All hybrid solvers incorporate queries sent to a back-end Advantage QPU; responses are used to guide solution search on the full-sized problem. Two D-Wave technical reports [3, 4] describe HSS features and performance.

Visit *dwavesys.com* to learn more about the Advantage QPU, Leap, the Ocean software stack, and hybrid computing.

Throughout this report, the discussion of QPU features and performance refers generally to all QPUs within their respective design series, either D-Wave 2000Q or Advantage. Of course, empirical results must be based on tests of specific solvers from each product line. We have not performed all tests on all QPUs, and given small chip-to-chip variations that are known to exist, we cannot guarantee that our tests would produce identical results on

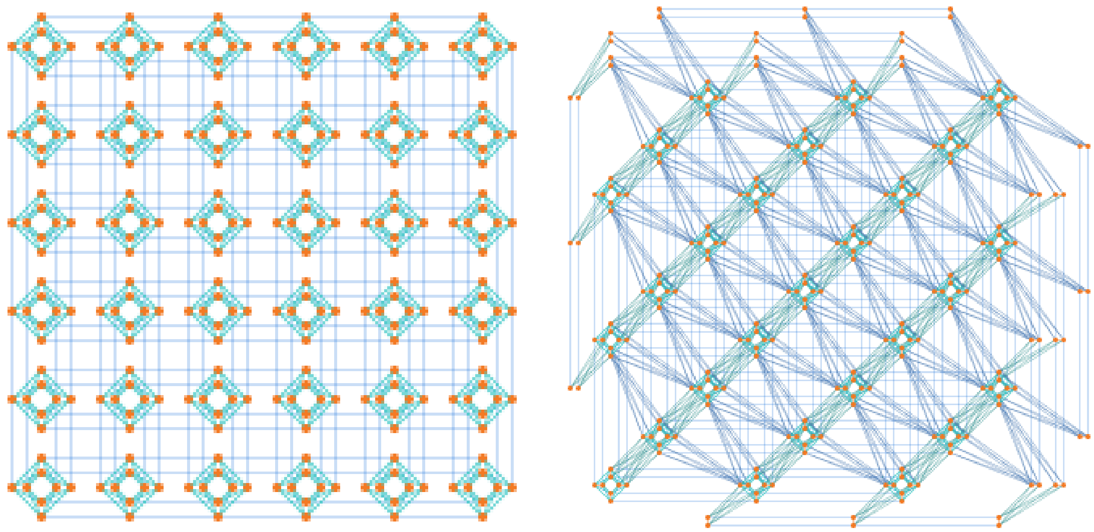


Figure 1: A C6 Chimera graph (left) with 36 unit cells containing 288 qubits. A P4 Pegasus graph (right) with 27 unit cells and several partial cells, containing 264 qubits. The comparatively rich connectivity structure of the P4 is clearly seen.

similar quantum solvers. However, we believe that the empirical results shown here are reasonably representative of typical performance within each technology family.

2 New Advantage Features

In a quantum annealing system, the hardware graph topology describes the pattern of physical connections between qubits and their couplers. The most important and obvious difference between 2000Q and Advantage QPUs is the upgrade from Chimera to the Pegasus topology, as shown in Figure 1.

The figure compares a C6 Chimera graph — a 6-by-6 grid of unit cells — with a P4 Pegasus graph, which contains 27 unit cells on a diagonal grid, plus partial cells around the perimeter. (Note that unit cells on the Pegasus graph contain four extra couplers.) Both graphs contain about the same number of qubits: 288 in the C6 and 264 in the P4. However, Chimera has just 6 couplers per qubit while Pegasus has 15 couplers per qubit, creating a visibly more complex connection structure.

In addition to greater size and connectivity, the Pegasus graph has been modified in other ways to improve performance on general problems. For example, Pegasus contains triangles, which means that more nonbipartite graph structures can be represented directly on the physical hardware. See [5] for more.

Table 1 presents a comparison of the two QPU designs in terms of typical component counts. The hardware graphs inside the 2000Q and Advantage QPUs are much larger than shown in Figure 1, corresponding to a C16 and a P16. The exact number of active qubits and couplers can vary across individual QPUs, because a small percentage of components may fail to meet technical specifications: the table shows the minimum number of active

	2000Q	Advantage
Graph topology	Chimera	Pegasus
Graph size	C16	P16
Number of qubits	> 2000	> 5000
Number of couplers	> 6000	$> 35,000$
Couplers per qubit	6	15

Table 1: Typical characteristics of Chimera- and Advantage-generation QPUs.

qubit and couplers in any QPU made available to the public.

More compact embeddings. An application input $\mathcal{I}$ for some problem $\mathcal{P}$ typically goes through two transformation steps before being sent to a D-Wave QPU. The first step is to reformulate $\mathcal{I}$ as an input for the quadratic unconstrained binary optimization problem (QUBO) (or for the equivalent Ising Model problem (IM)). This step uses standard cookbook techniques from NP-completeness theory and is not further discussed here; see [6] for more information.

The resulting QUBO input is represented by a logical graph $G = (V, E)$ containing n nodes and m edges; the input is specified by the pair (h, J) consisting of n node weights $h = \{h_i\}$ and m edge weights $J = \{J_{ij}\}$. The input is sent to the QPU, and the QPU returns a sample of R output solutions, each associated with a quality score defined by an *objective function* (also known as an *energy function*) calculated on (h, J).

In order to be solved directly on a given QPU, the graph G must be mapped onto its physical hardware graph, either Chimera or Pegasus. This mapping is normally performed by software utilities available in the Ocean SDK, using a technique called *minor embedding* (or informally, *embedding*). Figure 2 shows an example graph G before and after minor-embedding onto a P4 Pegasus graph. Each colored node in G corresponds to a *chain* of one or more qubits in the P4, connected by chain edges of the same color.[1] For example, the blue circles highlight a pink node in G that is mapped to a 2-qubit chain in the P4. The logical edges of G are shown in black on the P4; qubits and couplers that are not used in the embedding are shown in light gray.

The next section explains the relationship between graph connectivity and solution quality in quantum annealing processors.

Bigger inputs, shorter chains. The greater connectivity of Pegasus compared to Chimera has two important consequences:

- Because embeddings on Advantage QPUs use fewer qubits per node, the same number of qubits can hold larger logical graphs. This boosts the input capacity of Advantage QPUs beyond that obtained by simply increasing the qubit count.
- Advantage embeddings typically require shorter chains than 2000Q embeddings.

The *degree* of a graph is the maximum number of edges per node. The logical graph of Figure 2 has degree $d = 3$, which makes it fairly sparse and relatively easy to embed onto

[1]Despite the name, chained nodes are not always connected in a strict sequence; treelike structures may also occur.

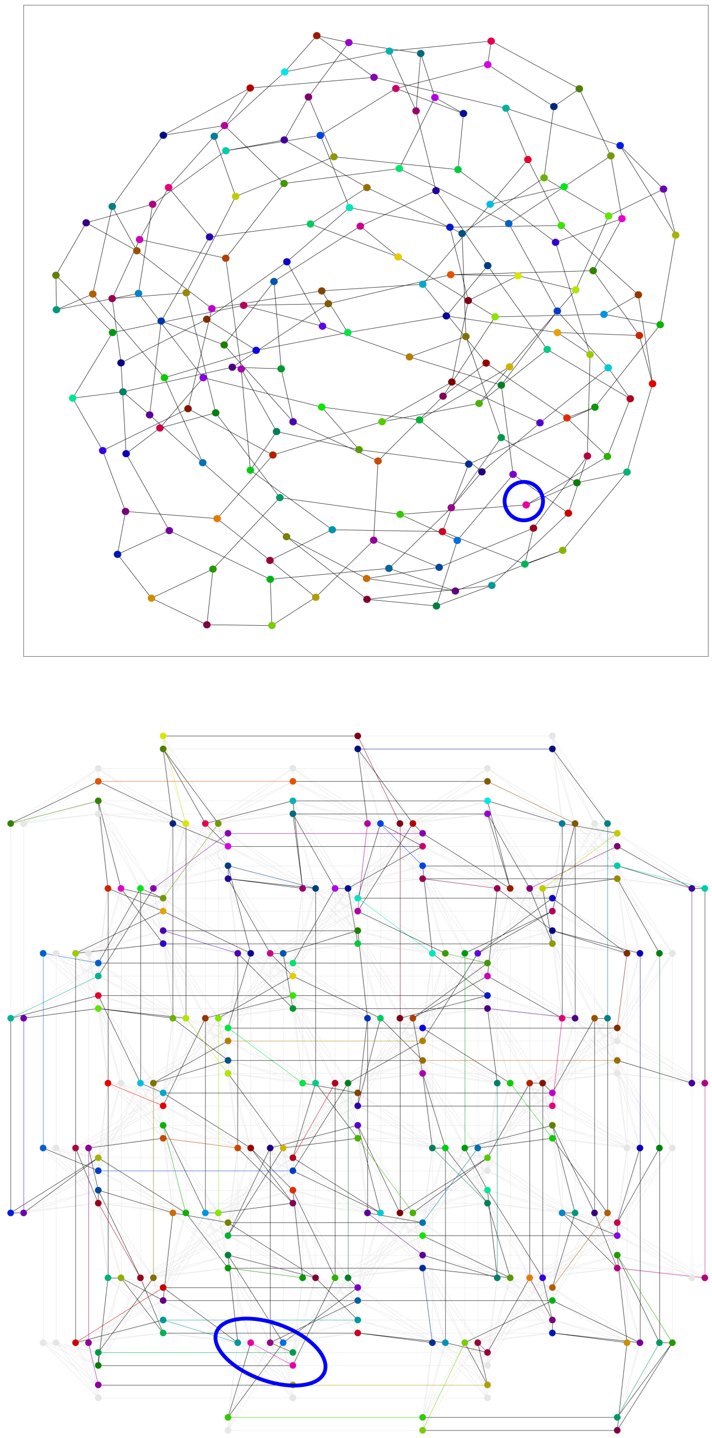


Figure 2: Minor embedding of a general QUBO graph (top) onto a P4 Pegasus graph (bottom). In the physical embedding, chain edges are colored to match their nodes, and logical edges are black. The blue circle and oval highlight a pink node that is mapped to a chain of two qubits during minor-embedding.

the Pegasus graph. In contrast, a fully-connected graph on n nodes, known as a *clique*, has degree $d = n - 1$. Dense graphs like cliques are among the hardest to embed on sparse hardware topologies and tend to produce the longest chains of any logical graph type.

The graph in Figure 3 compares clique embeddings obtained using an Advantage (orange) and a 2000Q (blue) hardware graph.[2] Each step-curve shows the largest clique size n that can be embedded using a given chain length L.

The vertical dotted line marks the largest embedded cliques on the 2000Q and Advantage solvers for maximum chain length 17, which are respectively $n = 64$ and $n = 119$. The horizontal dotted line compares chain lengths for $n = 64$ on both QPUs; the 2000Q embedding has $L = 17$ while the Advantage embedding has $L = 7$. That is, compared to the 2000Q QPU, the Advantage QPU can embed cliques that are almost 2 times larger, using chains that are less than half as long (when cliques are matched by size).

Below the graph, the middle table shows maximum embeddable n for other graph types, ordered by degree (with highest-degree cliques on the bottom row). A native input is defined directly on the hardware topology and does not have chains; the other graph types are described in Section 3. On average, Advantage QPUs can hold graphs that are 3 times larger than those held by 2000Q QPUs.

The bottom table shows chain lengths L for these graph types, using values of n that are small enough to fit on the 2000Q. The 3D lattice embeddings use a custom embedder; chain length is identical over all embedded nodes and is constant in n. Random NAE3SAT inputs were embedded using the *minorminer* heuristic, which finds embeddings having chains of uneven length: the table shows mean graph degree and the mean of *maximum* chain lengths per input, observed across several random inputs. Clique embeddings obtained by the polynomial-time *find_clique* utility have chains of uniform length. The mean of ratios in this table is 0.42: embeddings on Pegasus graphs typically have chains that are less than half as long as those on Chimera (see also [5]).

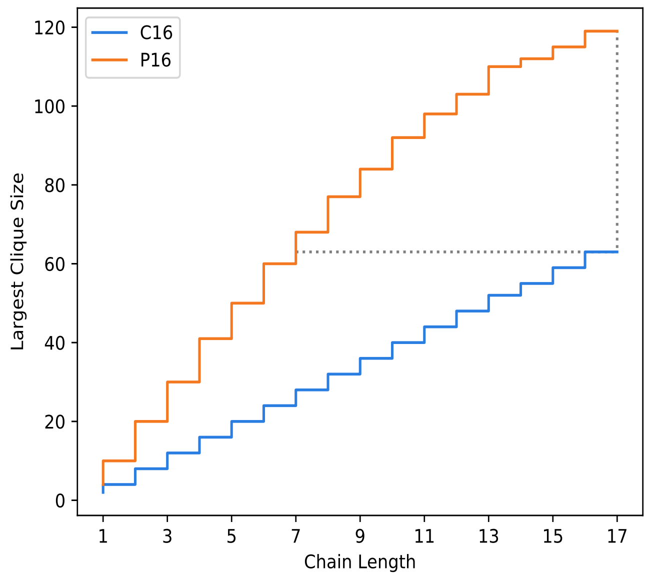
3 Performance Comparisons

This section describes three small case studies using embedded inputs to compare performance of an Advantage solver and a 2000Q solver, under a variety of performance metrics. We look at three problem categories: random cliques, inputs for the Not-All-Equal 3-Satisfiability (NAE3SAT) problem, and inputs defined on 3D Lattices.

We start with a brief explanation of how chain length affects solution quality and how Ocean features may be applied to boost quantum performance. More detailed discussion may be found in previous tech reports [1, 2].

Chain length and solution quality. Chained qubits in an embedding are connected by couplers assigned a weight J_{chain}, known as the *chain strength*. Suppose the logical weights (h, J) all lie within some numerical range $[-x, x]$, and that chain strength (a negative value)

[2] All empirical results in this section refer to specific solvers from each product line; their official online names are `DW_2000Q_6` and `Advantage_system1.1`. The latter (Advantage) solver has been decommissioned and is no longer available to the public.



Maximum Graph Sizes				
Logical Graph	Degree	2000Q C16 max n	Advantage P16 max n	Ratio P16/C16
Native	Chi=6, Peg=15	2030	5436	2.7
3D lattice (w/defects)	6	512	2354	4.6
NAE3SAT $\rho = 2.1$	13.2	100	286	2.9
NAE3SAT $\rho = 3$	18	90	242	2.7
Clique	Chi=63,Peg=118	64	119	1.9

Chain Lengths at Matched Graph Sizes				
Logical Graph	Graph Size	2000Q Chain Length	Advantage Chain Length	Ratio P16/C16
3D lattice (defects)	(all n)	4	2	0.50
NAE3SAT $r = 2.1$	$n = 70$	12	5	0.42
NAE3SAT $\rho = 3$	$n = 70$	33	11	0.33
Clique	$n = 60$	16	7	0.44

Figure 3: The graph at the top shows the largest input size for each chain length, when cliques are embedded onto the Advantage (orange) and 2000Q (blue) hardware graphs. The vertical dotted line shows differences in clique sizes for maximum chain length 17; the horizontal line shows differences in chain length for the largest clique embedded on the 2000Q. The middle table shows the maximum embeddable graph sizes n for a variety of input types; the mean of ratios in the table is 3.0. The bottom table shows typical chain lengths on graphs that are matched by size; the mean ratio of chain lengths is 0.42.

is set to $-J = cx$ for some positive coefficient c. The optimal choice of c lies in a sweet spot between two hazards.

Setting c too low might introduce spurious "optimal" solutions to the objective function, causing the physical problem to diverge from the logical problem and raising the chances of encountering output solutions containing *broken chains*: a broken chain contains qubit values that disagree, creating ambiguity as to which value should be assigned to the logical note that they represent. A *broken solution* contains at least one broken chain. One normally sets $c > 1$ to avoid the certainty of encountering broken chains, and higher values might be necessary.

On the other hand, all physical inputs are scaled to a fixed energy range $[-1, 1]$, according to their largest-magnitude weight, which is J_{chain} when $c > 1$. In this scenario, logical problem weights in $[-x, x]$ are scaled to $[-1/c, 1/c]$, which compresses their energy range: set c too high for the precision limits of the quantum control system, and problem weights cannot be distinguished from one another.[3]

All else being equal, shorter chains are associated with better-quality solutions, because shorter chains can tolerate smaller values of c without breaking; smaller values of c increase the problem scale and reduce QPU vulnerability to precision errors. However this argument does not hold when switching from Chimera to the Pegasus topology: although chains are shorter, the higher connectivity in Pegasus creates more "torque" per qubit which might *increase* the proportion of broken chains observed at equivalent chain strengths. For any given input type, the net effect of shorter chains and greater torque is challenging to model and impossible to predict exactly.

The situation is further complicated by the availability of fast (classical) postprocessing utilities that repair broken chains. For example, the *majority vote* (MV) utility, which is invoked by default in some Ocean solvers, assigns a value to each logical node based on a "vote" of the qubits in its chain (broken or not). The MV utility converts any broken solution into an intact solution, in just a few microseconds. The optimal choice of chain strength for any input class also depends on whether MV is in use.

The next three sections describe case studies using embedded inputs to compare performance of Advantage and 2000Q solvers under a variety of performance metrics. We look at three problem categories: random cliques as described in Section 2, inputs for the Not-all-equal 3-Satisfiability (NAE3SAT) problem, and inputs defined on 3D Lattices. The fourth section compares performance of a newer Advantage performance update processor released in September 2021 to an original Advantage processor launched in October 2020, on these three problems.

Throughout, we take a user-centric approach by incorporating tools and utilities available in the Ocean SDK.[4] Specifically, in all tests we apply Majority Voting to repair chains, and in some tests we tune chain strengths by solver and input type rather than relying on system defaults.

---

[3]This discussion oversimplifies the scenario that arises in practice. D-Wave solvers have an `extended_j_range` option that supports physical energy scales in range $[-4, +1]$ (Advantage) or $[-2, 1]$ (2000Q). Thus, setting $-J_{chain} = 4x$ would not affect the logical problem scale on an Advantage solver, but would shrink the logical problem scale by one-half on a 2000Q solver.

[4]These utilities are included in the open-source Ocean SDK [7, 8], which is part of a broad effort by D-Wave developers to make quantum-based problem-solving accessible. Growing experience with Ocean tools suggests combining the best features of quantum and classical computation can often produce better results than either approach used alone. Ocean SDK provides other chain-repair utilities not discussed here; see [9] for details.

Tuned Chain Strength						
n : 10	20		30	40	50	60
2000Q C16	4	6	7.3	8.4	12	10
Advantage P16	4	6	6	8	8	10

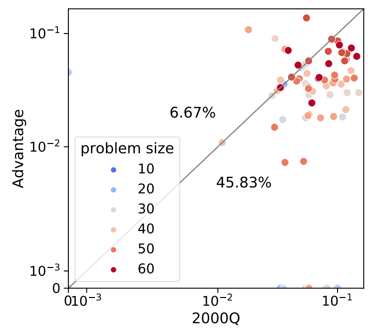


Figure 4: The table shows the optimal chain strength $-J_{chain}$ for clique inputs, for each QPU and problem size, to minimize the median scaled error M. The graph shows an input-by-input comparison of M for the Advantage (y-axis) and the 2000Q (x-axis) solvers using tuned chain strengths. Points are color-coded by problem size n. Percentages do not sum to 100 because many outcomes were ties, especially on small problems where both solvers achieved $M = 0$.

3.1 Random Cliques

In this section we compare performance of Advantage and 2000Q solvers using the clique inputs described previously. We generate 10 random cliques at each problem size $n = [10, \ldots, 60]$. Anneal time is set to 200µs, $R = 1000$ output samples are read, and the `extended_j_range` parameter with $[-2, 1]$ is turned on. The MV postprocessing tool was used throughout.

Since it is computationally infeasible to compute certified optimal solutions to these problems at largest n, we instead employ a reliable classical heuristic running for a long time to find *ersatz* optimal solutions. Let E_{ref} denote the *reference energy* found by this method.

A small exploratory study was performed to identify optimal values for chain strength J_{chain}, for each solver and each input size n. The study considered integer values of J_{chain} as well as a functional form $J_{chain} = c\sqrt{n}$ with c found by a regression analysis.

Here, J_{chain} was selected to minimize the *median relative error* M, computed as follows. For each sample of $R = 1000$ solutions, the median relative error is the scaled difference between the reference energy and the median energy in the sample, that is:

$$M = \frac{\Delta(E_{ref}, E_{median})}{|E_{ref}|},$$

Where $\Delta(E_{ref}, E_{median})$ is the absolute distance between the two energies, accounting for possible sign differences. A result of $M = 0$ indicates that the sample median — and at least half the solutions returned — have energies equivalent to the reference energy E_{ref}.

Although practitioners are arguably more interested in the relative error of the sample *minimum*, denoted L, we use M because both solvers frequently find optimal solutions on

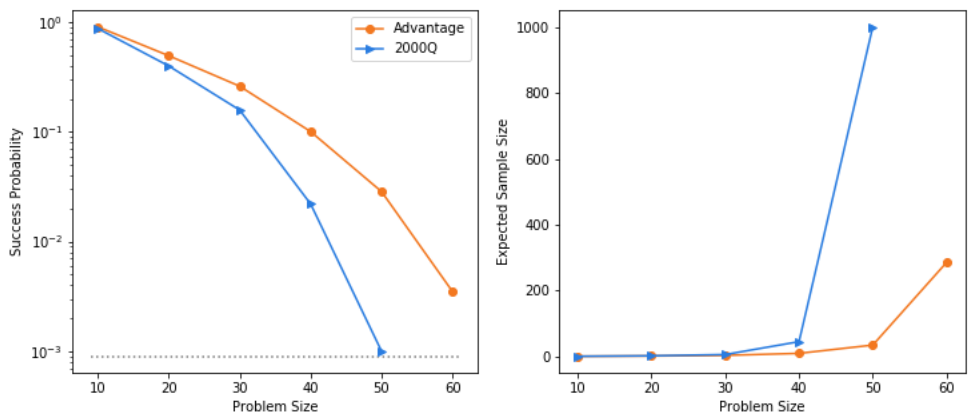


Figure 5: The left panel shows median success probabilities for Advantage and 2000Q solvers on random cliques. In these tests using 1000 reads, a success probability below 0.001 (i.e. below the gray dotted line), observed for 2000Q at $n = 60$, is recorded as zero and not plotted. The right panel shows $R = 1/\pi$, corresponding to the expected sample size needed to observe a reference solution in the output sample.

these problems, with the result that $L = 0$ in nearly all cases: the minimum relative error could not be used to optimize chain strength nor to distinguish performance of the two quantum solvers.

The results are shown in Figure 4. The table at the top shows the best chain strength J_{chain} found in the exploratory study. Note that even though chain *lengths* are less than half as long on Advantage compared to 2000Q QPUs, the best chain *strengths* are quite similar, reflecting the trade-off between chain length and torque discussed in the previous section. The one case of non-monotonicity in the table appears to be due to experimental noise. We remark that both QPUs can tolerate some imprecision in selection of optimal J_{chain}: values within, say, ± 2 of those shown in the table have a negligible effect on performance as measured by M.

The graph shows an input-by-input comparison of M for the Advantage (y-axis) and 2000Q (x-axis) solvers. Points below the diagonal indicate cases where Advantage found better solutions than the 2000Q and points above the line indicate cases where the 2000Q outperformed Advantage. The percentages shown do not sum to 100 because the outcomes were tied in 47.5 percent of cases, especially on small inputs for which both solvers achieved $M = 0$ (not shown in the graph). On the remaining (untied) cases shown here, the Advantage solver wins in 45.83 of 52.5 cases, that is, 87% of the time.

Figure 5 compares performance of the two solvers under the *success probability* metric denoted π, which is the estimated proportion of sampled solutions matching the reference energy, based on 1000 reads for each of 10 random inputs generated at each n.

The left panel shows median values of π for each solver at for each n, using tuned chain strengths and MV postprocessing. At $n = 60$ the 2000Q solver found no reference solutions in 1000 samples: this indicates $\pi < 0.001$ (below the grey dotted line), which is not plotted. The largest problem size for which the 2000Q solver achieves $\pi > 0.001$ is $n = 50$.

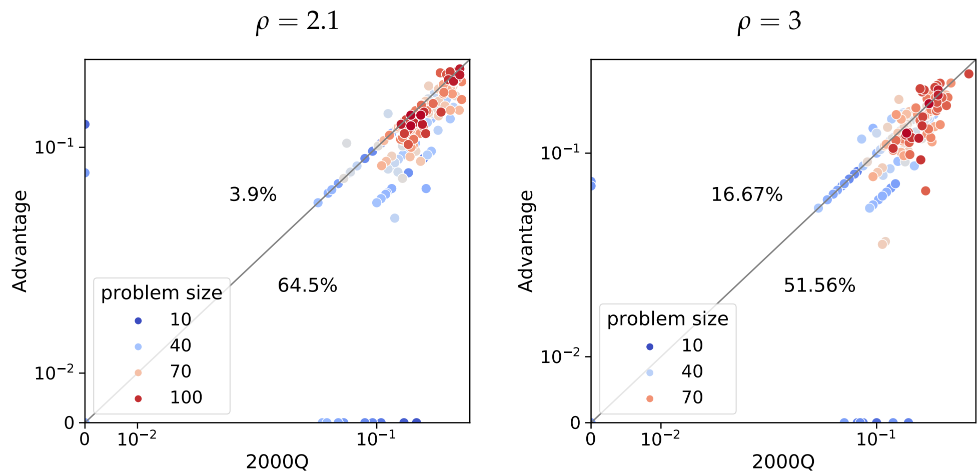


Figure 6: The left and right panels show results for $\rho = 2.1$ and $\rho = 3$, respectively. Maximum input size is $n = 100$ in the left panel and $n = 90$ in the right panel. The graphs show input-by-input comparisons of median relative error M for the Advantage (y-axis) and 2000Q (x-axis) solvers. Points below the diagonal line indicate cases where Advantage returned superior solutions. The percentages do not sum to 100 because many ties were observed, especially at small problem sizes where both found optimal solutions.

The right panel shows $R = 1/\pi$, the expected number of solution samples needed to observe a reference solution. At $n = 50$ the Advantage solver achieves a 54-fold speedup in computation time as measured by the expected number of samples needed to observe an optimal solution. Note that wall clock times at small R tend to be dominated by chip I/O overhead (programming and readout time): in this context, a 54-fold speedup in R maps to over a 14-fold speedup in wall clock time, from about 0.42 seconds to 0.03 seconds.

3.2 NAE3SAT Problems

Next we compare solver performance on inputs for the Not-All-Equal 3-Satisfiability (NAE3SAT) problem. A random NAE3SAT input is a boolean expression with n variables $v = \{v_1, \dots v_n\}$ organized in m clauses: each clause contains three randomly selected variables or their negations. A clause is *satisfied* (i.e., has the value True) if the values of the three variables are not all equal. Clauses are joined by disjunctions (logical ”and”), so that the whole formula is satisfied only when every clause is True.

The combinatorial hardness of random formulas constructed this way depends on the clause-to-variable ratio $\rho = m/n$. Our tests use random NAE3SAT inputs generated at $\rho = 2.1$ (the so-called *critical point*, interesting for testing performance at finding optimal solutions) and at $\rho = 3$ (above the critical point, interesting for testing performance at finding approximate solutions). The boolean expressions thus generated are translated into QUBO logical inputs using standard techniques. Empirical tests show that the logical graphs resulting from problems generated at $\rho = 2.1$ and $\rho = 3$ have average node degrees $d = 13.2$ and $d = 18$, respectively.

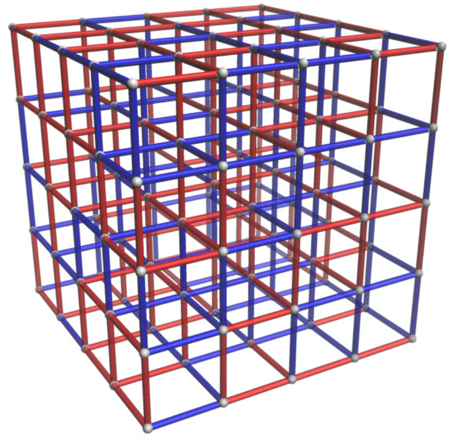


Figure 7: A $5 \times 5 \times 5$ 3D spin glass, from [12]. The edge weights are assigned $J_{ij} = +1$ (red) or $J_{ij} = -1$ (blue) uniformly at random.

These inputs were embedded using the heuristic `find_embedding` tool available in Ocean. This embedder differs from the polynomial-time `find_clique_embedding` tool used for clique inputs in several ways; for example, embeddings may contain chains of varying length. See [10, 11] for more about embedders in Ocean.

In these tests we generated five random inputs at each problem size $n \in [10, 12, \ldots, n_\rho]$ where $n_{2.1} = 100$ and $n_3 = 90$. Embeddings on the Advantage QPU tend to have mean and maximum chain lengths about half as long as embeddings on the 2000Q QPU.

Putative optimal solution energies were obtained using reliable heuristics applied to the physical problems. For each input we took 1000 samples from each QPU. Anneal time was set to 200µs; chain strength was set by the scale multiplier $d = -3$ in all tests (based on a small exploratory study); and the MV postprocessing option was turned on. Figure 6 shows the results for input sets $\rho = 2.1$ (left panel) and $\rho = 3$ (right panel). Solution quality is measured by the median relative error metric M described in the previous section. Points below the diagonal line correspond to cases where the Advantage solver returned better solutions than the 2000Q solver. The percentages in each panels do not sum to 100 because the two solvers tied in many cases.

As with cliques, in cases where the solvers are not tied, Advantage finds better solutions than the 2000Q on a higher proportion of inputs. For $\rho = 2.1$, Advantage found better solutions in 64.5 out of 68.4 = 64.5 + 3.9 cases; that is, 94% of the time. For $\rho = 3$, the Advantage solver returned better solutions in 75 percent of cases without ties.

3.3 3D Lattices

We finish the section with a summary of a performance study described by King and Bernoudy [12]. That work compares performance of the Advantage and D-Wave 2000Q LN (lower noise) solvers studied in previous sections, at the task of sampling ground states in 3D lattices.

A 3D lattice problem consists of a logical grid of size $L \times L \times L$, as shown in Figure 7, with

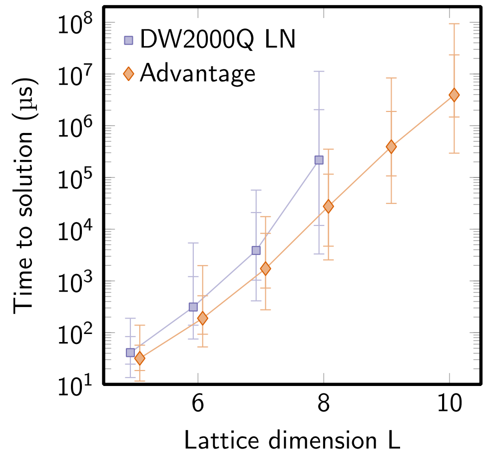


Figure 8: Time to solution (TTS), from [12]. The graph shows TTS as a function of input size for each solver; the error bars indicate percentiles [10,25,70,90] over 100 instances and the lines join median points. Beyond the capability of solving larger problems, the Advantage solver outperforms the 2000Q solver both in terms of absolute computation time and of better scaling on 3D spin glasses.

$h_i = 0$, and with edge weights $J_{ij} \in \{\pm 1\}$ (red and blue in the figure) assigned uniformly at random to lattice edges.

One test described in [12] generates 100 random instances for each $L \in [5, 6, \ldots, 10]$ and embeds them onto both QPUs using a custom embedder that exploits the regular structures of lattice-like graphs. Note that lattices of size $L > 8$ cannot fit on the 2000Q QPU. Embeddings in the Pegasus graph have chain length 2, whereas embeddings in the Chimera graph have chain length 4.

These tests allow the logical lattices to contain *defects* (missing nodes), since the hardware graphs in both QPUs contain a small number of inoperable qubits. If some node in the lattice cannot be represented on either one of the two hardware graphs, it is removed from the logical problem. Thus, both QPUs are given identical lattice structures to solve. Chain strength was set to $J_{chain} = -2$, near the optimal value for both solvers; the `extended_j_range` option and the majority vote postprocessor were turned on.

A pilot study was used to find putative ground states (reference energies) using reliable heuristics running on the logical problems. Then, for each input, a batch of 500 solutions was read for each anneal time $t_a \in [2, 4, 8, \ldots, 256]$µs, iterating for at least 100 batches or until the reference energy was found at least 50 times. The ground state probability $p_{GS}(t_a)$ was estimated for each anneal time t_a as the proportion of reference solutions found in the full sample set.

The performance metric is time to solution (TTS), which based on the total number of reads needed to ensure that the ground state is observed with a confidence of at least 99%. For each input, TTS is calculated as the minimum value of TTS(t_a) over all t_a in the test:

$$\mathrm{TTS}(t_a) \quad = \quad t_a \frac{\log(.01)}{\log(1 - p_{GS}(t_a))}.$$

The results are shown in Figure 8, which shows time to solution over all inputs at each problem size L. The error bars at each L indicate percentiles $10, 25, 75, 90$ over 100 instances, and the lines join median points.

At $L = 8$, the largest problem that fits on the 2000Q QPU, median TTS for the Advantage solver is about 10 times faster than that for the 2000Q solver. The better scaling (lower slope) of the Advantage processor times with respect to input size is also notable. These results show that beyond the capability of solving larger problems, the Advantage QPU outperforms the 2000Q QPU both in terms of absolute computation times and in better scaling on 3D lattice problems.

3.4 Advantage Performance Update

We conclude this section with a small comparison of the Advantage 1.1 QPU from the original Advantage series launched in September 2020 with a QPU from the Advantage performance update series, first released in October 2021. At the time of this writing, two performance update systems are available to the public, one at D-Wave headquarters in Canada and one at the Forschungszentrum Jülich in the JARA Institute for Quantum Information in Germany.

We have not performed all tests in this report on all Advantage QPUs in existence; given small QPU-to-QPU variations in any product line, we cannot say with certainty that identical outcomes would be observed in all cases. However, on tests that we have carried out, Advantage performance update solvers consistently outperform both the 2000Q and Advantage 1.1 solvers. Performance differences among Advantage QPUs (with Pegasus hardware graphs) tend to be smaller than differences between Advantage and 2000Q QPUs.

Performance gains from this latest product series may be attributed to improvements in fabrication materials and processes, which impact solution quality in two important ways: *higher yield* and *lower noise*.

Higher yield, larger inputs. Advantage performance update QPUs exhibit *higher yield*: that is, a higher proportion of qubits and couplers in the hardware graph are fully functional and active after calibration. Higher yield means that the newer Advantage QPUs can embed larger logical inputs than can the original Advantage QPUs. We have observed increases in maximum embedded problem size of between 4 percent and 50 percent, depending on problem type.

Yield differences among Pegasus-structured graphs do not always translate to differences in chain lengths. For example, chain lengths in custom embeddings of regular graph structures such as lattices and cliques do not vary with yield. We observe that, depending on input type, the heuristic embedder can find slightly better results on higher-yielded QPUs, reducing maximum chain length by perhaps one or two links. We do not observe any significant differences in optimal chain strengths between the two QPU product lines.

Lower noise, better-quality solutions. Technological improvements to design, materials, and fabrication processes contribute to produce QPUs in the Advantage performance update series that are more resilient against environmental noise than earlier Advantage systems.

Figure 9 illustrates how this technology upgrade translates to better-quality solutions. The

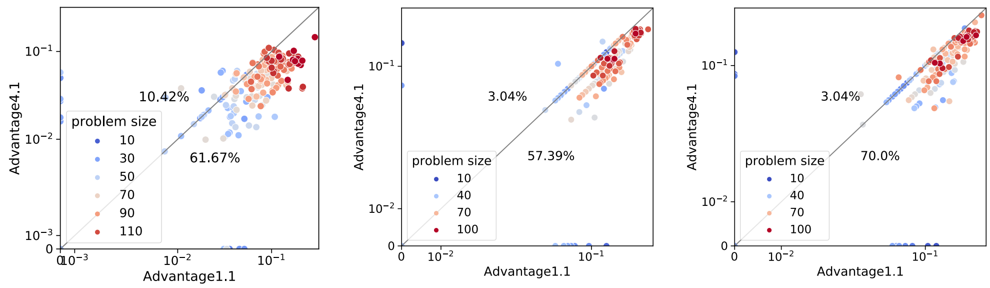


Figure 9: Input-by-input comparisons of solution quality returned by Advantage 4.1 and Advantage 1.1 solvers. Points below the diagonal correspond to cases where Advantage 4.1 returned better-quality solutions than Advantage 1.1. Left panel, clique inputs; center panel, NAE3SAT instances $\rho = 2.1$; right panel, NAE3SAT instances $\rho = 3$.

three panels show input-by-input comparisons of solutions returned by the Advantage 4.1 solver (y-axis) and the Advantage 1.1 solver (x-axis).[5]

These tests replicate the studies in Section 3.1 on clique inputs (left panel) and in Section 3.2 on NAE3SAT instances for $\rho = 2.1$ (center panel) and $\rho = 3.0$ (right panel). Points below the diagonal correspond to inputs for which solution quality from Advantage 4.1 was strictly better than that from Advantage 1.1. In all three panels, considering only cases where the two solvers did not tie, Advantage 4.1 returned better-quality solutions 86% (left panel), 95% (center panel), and 96% (right panel) of the time.

4 Conclusions

This report describes features of the D-Wave Advantage quantum processor series in comparison to its predecessor, the D-Wave 2000Q series, focusing on the quantum processing units (QPUs) at the core of each system.

Because of higher qubit counts and greater connectivity, Advantage QPUs can hold inputs that are three times larger, on average, than those that fit on 2000Q QPUs, while embedding chain lengths are generally half as long (on inputs that are matched by size).

Three case studies in Section 3 compares performance of an Advantage solver from the original product launch in 2020 to a 2000Q LN system, using three categories of embedded inputs: cliques, NAE3SAT inputs, and 3D lattices. Results from this section demonstrate that, beyond the ability to hold larger inputs, Advantage solvers can find better-quality solutions than 2000Q solvers when both are given equivalent time limits; depending on the context, better-quality solutions translate to faster times to find equivalent-quality solutions, and to faster times to solve problems to optimality.

We attribute this performance improvement to the new Pegasus graph topology which supports minor-embeddings with shorter chain lengths. Furthermore, Section 3.4 demonstrates that the Advantage performance upgrade solvers released in 2021 achieve better

[5] Advantage 4.1 is a performance update solver released in 2021. Advantage 1.1 is an original Advantage system launched in September 2020; this QPU was decommissioned in late 2021 and is no longer publicly available.

overall performance than the original Advantage 1.1 solver launched in 2020. The tests described in this report focus on performance as would be experienced by practitioners with real-world applications in hand (which typically require embedding), who make use of utilities in the Ocean SDK, which can boost quantum performance.

The preliminary results presented here are only the beginning: we look forward to more demonstrations of superior performance by Advantage-generation quantum processors in comparison to both quantum and classical alternatives as well as by future generations of D-Wave annealing-based quantum computers.

References

[1] McGeoch, C. and Farré, P., *The Advantage System: An Overview*, dwavesys.com/resources (2020).

[2] McGeoch, C. and Farré, P., *The Advantage System: Performance Update*, dwavesys.com/resources (2021).

[3] C. McGeoch et al., *Hybrid Solver Service + Advantage: Technology Update*, dwavesys.com/resources (2020).

[4] C. McGeoch and P. Farré, *Hybrid Solver for Constrained Quadratic Models*, dwavesys.com/resources (2021).

[5] K. Boothby et al., *Next-Generation Topology of D-Wave Quantum Processors*, dwavesys.com/resources (2019).

[6] A. Lucas, "Ising formulations of many NP problems," Frontiers in Physics **2** (2014).

[7] "D-Wave's Ocean Software," http://ocean.dwavesys.com (2020).

[8] "D-Wave Ocean Software Documentation," http://docs.ocean.dwavesys.com/en/stable/index.html (2020).

[9] "D-Wave Ocean Software Documentation, keyword: post processing," http://docs.ocean.dwavesys.com/en/stable/index.html (2020).

[12] A. King and W. Bernoudy, "Performance benefits of increased qubit connectivity in quantum annealing 3-dimensional spin glasses," arXiv:200912479[quant-ph] (2020).

[10] "D-Wave Ocean Software Documentation, keyword: minor embedding," http://docs.ocean.dwavesys.com/en/stable/index.html (2020).

[11] "D-Wave Ocean Software Documentation, keyword: find clique embedding," http://docs.ocean.dwavesys.com/en/stable/index.html (2020).