Analysis of a Programmable Quantum Annealer as a Random Number Generator

Elijah Pelofske*[1]

[1]Los Alamos National Laboratory, CCS-3 Information Sciences

Abstract

Quantum devices offer a highly useful function - that is generating random numbers in a non-deterministic way since the measurement of a quantum state is not deterministic. This means that quantum devices can be constructed that generate qubits in a uniform superposition and then measure the state of those qubits. If the preparation of the qubits in a uniform superposition is unbiased, then quantum computers can be used to create high entropy, secure random numbers. Typically, preparing and measuring such quantum systems requires more time compared to classical pseudo random number generators (PRNGs) which are inherently deterministic algorithms. Therefore, the typical use of quantum random number generators (QRNGs) is to provide high entropy secure seeds for PRNGs. Quantum annealing (QA) is a type of analog quantum computation that is a relaxed form of adiabatic quantum computation and uses quantum fluctuations in order to search for ground state solutions of a programmable Ising model. Here we present extensive experimental random number results from a D-Wave 2000Q quantum annealer, totaling over 20 billion bits of QA measurements, which is significantly larger than previous D-Wave QA random number generator studies. Current quantum annealers are susceptible to noise from environmental sources and calibration errors, and are not in general unbiased samplers. Therefore, it is of interest to quantify whether noisy quantum annealers can effectively function as an unbiased QRNG. The amount of data that was collected from the quantum annealer allows a comprehensive analysis of the random bits to be performed using the NIST SP 800-22 Rev 1a testsuite, as well as min-entropy estimates from NIST SP 800-90B. The randomness tests show that the generated random bits from the D-Wave 2000Q are biased, and not unpredictable random bit sequences. With no server-side sampling post-processing, the 1 microsecond annealing time measurements had a min-entropy of 0.824.

1 Introduction

Random number generation (RNG) is a very important capability in information computing. In particular *unbiased* random number generation is extremely important in many computing applications. Pseudo-Random Number Generators (PRNGs) are deterministic and very fast software level algorithms that can reliably generate random numbers. True Random Number Generators (TRNGs) are based on a physical property of a system that makes the random number generation inherently non-deterministic. Quantum systems have this property of non-determinism where it is not possible to know deterministically what the measured state of a quantum system will be before it has been measured.

Testing for randomness, in particular secure and unbiased randomness, is not directly possible. Instead, tests for patterns and biases that are clearly *not* random can be tested for [1–5]. If a proposed RNG is tested against enough of these tests which can detect non-random data, then you can be reasonably confident in the ability of the RNG to generate uniformly random numbers.

One of the types of programmable quantum computers that have become available to test, typically as cloud computing resources, are D-Wave quantum annealers. Quantum annealing is a specialized type of quantum computation that aims to sample the optimal solution(s) of a combinatorial optimization problem, ideally using adiabatic evolution [6–10]. Quantum annealing hardware is typically implemented using the transverse driving Hamiltonian where the system is initialized in the groundstate of the Transverse-field Hamiltonian [7–12]. D-Wave quantum annealers are physically implemented using programmable superconducting flux qubits [13–19]. Quantum annealers, and more generally quantum computers, are potentially interesting as secure entropy sources for generating random numbers because of the inherent stochasticity of measuring quantum states - there is not a deterministic mechanism to compute what the measured state will be of an arbitrary quantum state. For this reason, quantum

*Email: epelofske@lanl.gov

computers, and more generally physical sources of measurements of quantum information, are True Random Number Generators (TRNGs) (or QRNGs) [4, 20, 21]. Importantly, there exist current technologies which are secure, high bit-rate, QRNGs [22, 23].

The primary reason that modern quantum annealers are not perfect random number generators is because there are a large number of sources of error and bias in the computation, for example the spin bath polarization effect [24, 25] can cause sequential anneal-readout cycles to have self correlations (in time), and programmed coefficients (even if they are 0) have slightly different effective weights on the hardware [26]. Furthermore, it has been shown that modern D-Wave quantum annealers have a measurable performance change over time [27, 28]. There have also been cross-qubit correlations observed on a D-Wave 2000Q chip [29, 30]. There have been studies which aim to reduce biases and noise present in minor-embedded QA computations, which in the case of reducing biases in the constraint of the graph partitioning problem results in effectively attempting to create an unbiased Quantum Annealing random number sampler, see ref. [31]. Interestingly, quantum annealing (even in an ideal computation, with no noise), does not in general sample degenerate groundstates (i.e. optimal solutions of the combinatorial optimization problem) uniformly due to the transverse field driving Hamiltonian [32–36], meaning that using the QA sampling of the groundstates of non-trivial Hamiltonians would not be a good source of unbiased random numbers. Instead, the much simpler case of an all 0 coefficient Ising model is the most direct way to program these devices to produce random numbers (see more details in Section 2.1). D-Wave quantum annealers have been evaluated, on somewhat small problem sizes, for the possibility of utilizing them as TRNGs [1] in previous studies [37, 38].

Quantum random number generators in general are a topic of much interest, for example there have been several studies which examined using gate model quantum computers as random number generators [39–44], boson sampling [45], using quantum walks to generate random numbers [46, 47], and device-independent secure random number generation [48]. The idea of using random dense quantum volume circuits (see refs. [49–51] for details on quantum volume circuits) as random number generators in the gate model setting have also been proposed [52].

This paper presents the most comprehensive review of using a quantum annealer as a random number generator performed to date, totalling over 20 billion bits of qubit measurements, and testing 8 different QA device settings for how they impact the measured bits. In particular, this very large dataset allows all of the NIST SP800-22 randomness tests, and all of the NIST SP800-90B min-entropy (non-IID) tests, to be executed on the data (some of the tests have a minimum bit length requirement). This has not been able to be done before for quantum annealing random bits [37, 38] or more generally for cloud accessible quantum computers.

All data from this study are publicly available as a Zenodo dataset [53].

2 Methods

Section 2.1 details the Quantum Annealing implementation details, and Section 2.2 details the randomness testsuites that are used.

2.1 Quantum Annealing

The computation performed by D-Wave quantum annealers is described by eq. (1), and eq. (2) describes the discrete optimization Ising model that a user can program to be sampled by the quantum annealer (the quadratic coefficients are subject to the constraint of the native connectivity of the quantum annealing hardware). The functions $A(s)$ and $B(s)$ define the Transverse field driving Hamiltonian strength and the programmed Ising model Hamiltonian, respectively, parameterized by the variable s. In standard quantum annealing, which is the setting used in this study, s defines a linear schedule as a function of anneal time, and the strengths of $A(s)$ and $B(s)$ at each s step are system defined quantities. At the beginning of the anneal the $A(s)$ term dominates, and then over the course of the anneal $A(s)$ is reduced in strength and $B(s)$ is increased in strength. Eq. (2) is a slight re-formulation of the Ising model defined in the second summation term of eq. (1). The goal of the quantum annealer is to find a minimum variable assignment vector z given the objective function eq. (2). The variable states can be either $\{0,1\}^n$, in which case the combinatorial optimization problem is a Quadratic Unconstrained Binary Optimization (QUBO) problem, or the variables can be spins $\{+1,-1\}^n$, in which case the combinatorial optimization problem is an Ising model.

[1] Sometimes the acronym that is used for quantum devices generating random numbers is Quantum Random Number Generator (QRNG)

$$H = -\frac{A(s)}{2}\Big(\sum_i \hat{\sigma}_x^{(i)}\Big) + \frac{B(s)}{2}\Big(\sum_i h_i \hat{\sigma_z}^{(i)} + \sum_{i>j} J_{i,j}\hat{\sigma_z}^{(i)}\hat{\sigma_z}^{(j)}\Big) \tag{1}$$

$$f(z_1, \ldots, z_n) = \sum_{i=1}^{n} h_i z_i + \sum_{i<j} J_{ij} z_i z_j, \tag{2}$$

The D-Wave quantum annealer that is used to generate random bits is `DW_2000Q_LANL`, the Chimera hardware graph for this device is shown in Figure 1. The simplest way to generate random bits using a quantum annealer is simply to set the user programmed coefficients for all linear terms (e.g. hardware qubits) and quadratic terms (e.g. hardware couplers) to 0, meaning that only the transverse field Hamiltonian is present in the computation (ideally), which means that the qubits are in a uniform superposition, while the computation is coherent, during the anneal. There are certainly more complicated ways that could be utilized with the goal of extracting good random bits, such as random circuit sampling on gate model devices [52] or by tuning devices biases to improve sampling of balanced partitions for the graph partitioning problem [31], however in general RNG's need to be as fast as possible and therefore minimizing the complexity of the computation is likely a good motivation to aim for. Explicitly, the Ising model (variable states are $\in \{+1, -1\}$) that we will sample is given in eq. (3).

$$f(z_1, \ldots, z_n) = \sum_{i=1}^{n} 0 z_i + \sum_{i<j} 0 z_i z_j, \tag{3}$$

In many QRNG systems readout time is much longer than comparable PRNG's, and for cloud based quantum annealers this is also an important aspect to consider. In the experiments we perform, the total annealing time will be varied from 1 microsecond up to 2000 microseconds (1 microsecond is the shortest annealing time available on the D-Wave 2000Q devices). The potentially relevant thing that can be investigated is whether the 1 microsecond annealing times can give high quality random bits, because this utilizes a relatively small amount of compute time (certainly compared to longer annealing times).

The main D-Wave parameters that will be varied are the *annealing time*, which will use 1, 10, 100, and 2000 microseconds. These annealing times span the range of allowed annealing times on the `DW_2000Q_LANL` chip; 1 microsecond is the smallest annealing time and 2000 is the longest available annealing time. The other parameter that will be tested is turning on server side classical post processing which aims to improve *sampling* (although in this specific case, the sampling is being done on an all zero coefficient Ising model). In order to turn on this server side post processing, the user facing post process option was set to `sampling` [54]. The `sampling` server side post-processing option performs local bit changes on the measurements (before sending the results to the user) with the goal of obtaining a post-processes set of samples that corresponds to a Boltzmann distribution with inverse temperature β, where β is β is set to a value near to the inverse temperature corresponding to the raw samples (see ref. [54] for more details). We test turning this server-side post processing option on and off. The reasoning is that we would ideally want to the quantum annealer to be able to produce unbiased random bits without this post processing, however it may be the case that the classical post processing helps reduce bias in the samples at with a small computational overhead, in which case it would be interesting to quantify this. Therefore, in total there will be 8 datasets, each using a different quantum annealer parameter choice. Each of the datasets will be strictly sequential in time - e.g. the order of the bits will not be changed by some other entropy source. This time series representation of the data is especially important since it has been shown that there are long term trends that can be observed in current D-Wave quantum annealing processors [27]. Additionally, the exact ordering of the bits within each anneal (e.g. whole-chip readout cycle) is strictly based on the logical qubit indexing within the hardware, which is fixed for all samples, but is arbitrarily set. Each anneal-readout cycle is concatenated with the next anneal-readout cycle that was executed in time - no other source of entropy is present in the data.

With the goal of mitigating the spin bath polarization effect [24] self-sample correlations, all of the data is constructed by sequentially calling the D-Wave backend for a single anneal-readout cycle (i.e. instead of measuring many anneals in a single job). Each job is sampled as an Ising model, meaning that spins are the measured states (although whether the model was specified as QUBO or Ising would in principle have no impact on the results). Additionally, although the bits are all time ordered, because of network interruptions or device power losses there are gaps in the time of the sequential random bit sampling.

The parameters used for the 8 datasets is as follows:

1. *Test 1* uses server side classical post-processing, and an annealing time of 1 microseconds.

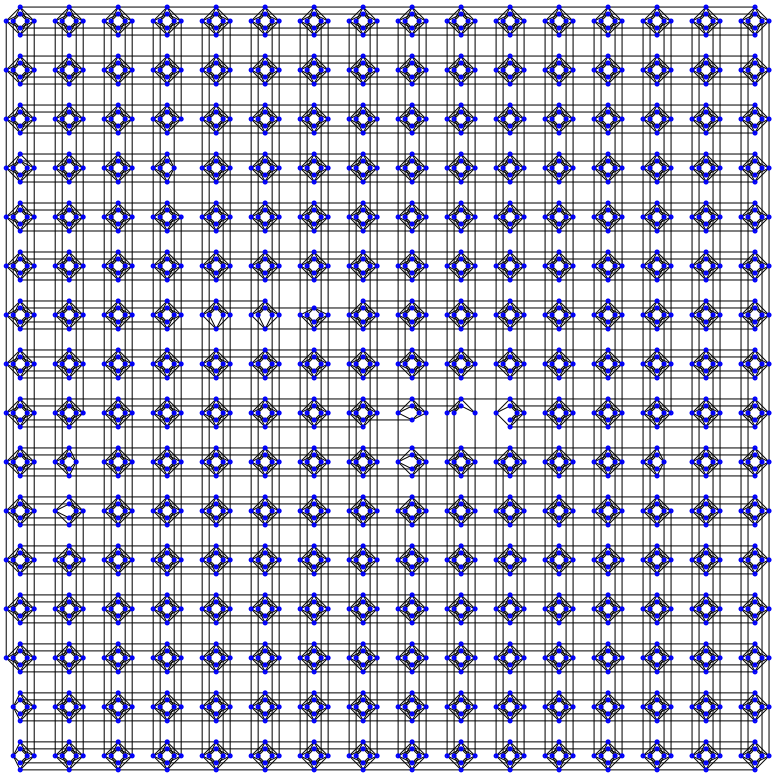

Figure 1: LANL D-Wave 2000Q hardware connectivity graph (the name of this type of connectivity graph is Chimera, which is in general a sparse but scalable hardware implementation of quantum annealing). This device has 2032 active qubits (due to hardware defects the full Chimera lattice of 2048 qubits is not active). The chip id of this device is `DW_2000Q_LANL`.

2. *Test 2* uses server side classical post-processing, and an annealing time of 2000 microseconds.
3. *Test 3* uses server side classical post-processing, and an annealing time of 10 microseconds.
4. *Test 4* uses server side classical post-processing, and an annealing time of 100 microseconds.
5. *Test 5* uses default sampling with no server side classical post-processing, and an annealing time of 1 microseconds.
6. *Test 6* uses default sampling with no server side classical post-processing, and an annealing time of 2000 microseconds.
7. *Test 7* uses default sampling with no server side classical post-processing, and an annealing time of 10 microseconds.
8. *Test 8* uses default sampling with no server side classical post-processing, and an annealing time of 100 microseconds.

For all 8 datasets, the `programming_thermalization` and `readout_thermalization` are set to 0 microseconds so as to remove any thermalization effects (beyond thermalization that occurs after the qubits lose coherence; the qubit coherence times are estimated to be on the order of 10's of nanoseconds [55]). All other parameters are set to default. The motivation for evaluating these different parameter choices is the following.

1. Although in general increasing annealing times on D-Wave quantum annealers results in better sampling success rate of combinatorial optimization problems, these long annealing times are much longer than the qubit coherence times of current D-Wave quantum annealers [55, 56]. Therefore, the longer annealing times are using thermalization to marginally improve the sampling success probability [57]. However, in this case there is not a combinatorial optimization problem being sampled - therefore in principle it may be the case that longer anneal times accumulate more errors in the computation, in particular more biases in the random sampling computation we aim to perform. Therefore, it may be advantageous to sample using the shortest annealing times available on the hardware – whether this is true or not is the aim of the varying annealing times. Furthermore, sampling rate for random numbers is extremely important – faster random bit sampling

is more useful than slower sampling. Therefore, if the smaller annealing times produce high entropy random bits this would be better than using longer annealing times.

2. The server side classical post-processing is not ideal since we wish to evaluate whether the bare quantum annealing hardware can produce good random bits. However, this post processing is intended to improve sampling of combinatorial optimization problems, and in this case there is nothing to optimize with respect to energy. Nevertheless, it is an interesting question to consider whether there is a clear difference between the QA sampling with and without the server side post processing for sampling optimization.

2.2 Testing for randomness

The randomness test that will be applied to the data are all of the tests from SP800-22 Rev 1a by National Institute of Standards and Technology [58], titled *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. This testsuite contains 15 randomness tests, two of which contain several sub-tests. In total each of the 8 datasets will be tested against 38 randomness tests, each giving a p-value output. For the purposes of maintaining consistency, and using the original NIST SP 800-22 test definitions [58], a computed P-value which is ≥ 0.01 would accept the sequence as being random, and otherwise we would consider it to be non-random. This p-value threshold criteria is applied to all of the randomness tests. In the tests where there are multiple computed P-values, such as `cumulative sums` where there is a forward and backward mode of operation, all P-values must be greater than or equal to 0.01 to be considered random. The `serial` test also outputs two P-values.

The implementation used for this analysis is the Python 3 package nistrng [2]; this package was chosen primarily for its compatibility with NumPy [59] arrays, which was necessary for the size of the datasets being tested. Other implementations that are, for example, based on casting the bits to integers does not scale well to these large dataset sizes.

The randomness test implementation details and references are not enumerated here - all details can be found in ref. [58] along with the linked open source code implementations.

In the context of verifying entropy sources, a useful measure is the *min-entropy*, which is a conservative measure of entropy sources. The *min-entropy* metric gives a clear way of determining how unpredictable a set of random variable samples is, and therefore is another way of quantifying bitstring randomness. Min-entropy is maximized for a uniform distribution, as with standard Shannon entropy [60], and minimized closer to 0 for biased distributions. In this case, we apply the NIST SP 800-90B testsuite (titled *Recommendation for the Entropy Sources Used for Random Bit Generation*) [61] [3] in order to compute *min-entropy* estimates on the quantum annealing bitstrings. In particular, the *non IID* testsuite is executed in order to obtain the h_{Original} `min-entropy` estimates from the 10 tests in the testsuite. The NIST SP 800-90B non-IID track is intended to be applied to noise sources that do not generate Independent and Identically Distributed (IID) samples. The testsuite was executed with the help of Charliecloud containerization [62].

Parameter combination	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Bits	2563745952	2573540192	2540168656	2553250672	2610469760	2580696896	2574846768	2580371776

Table 1: Each dataset with different D-Wave settings contains at least 2.5 billion time ordered bits.

3 Results

Table 2 shows the complete randomness testsuite data for the 8 QA implementation variations. Table 1 shows the total size of each of the 8 datasets. The threshold for failing each randomness test varies depending on the test, but a p-value less than 0.01 definitely shows that the dataset fails that randomness test. The result is that there is no QA device setting that generates random bit strings that pass all of the randomness tests.

Notably, the server side classical post processing did improve the random bitstring - in the sense that more of the tests passed when that post processing was applied. Also very notable is that the raw non post-processed QA data failed the monobit test, arguably the most fundamental randomness test that can be applied. This shows

[2] `https://github.com/InsaneMonster/NistRng`

[3] `https://github.com/usnistgov/SP800-90B_EntropyAssessment`

Test name	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Monobit	0.00298	0.2144	0.03651	0.26875	0.0	0.0	0.0	0.0
frequency within block	0.26277	0.81603	0.31771	0.44942	0.0	0	0.0	0.0
Runs	0.2031	0.81369	0.07247	0.06147	0.0	0.0	0.0	0.0
Longest runs in a block	0.42578	0.05784	0.42767	0.37912	0.0	0.00002	0.0	0.00002
binary matrix rank	0.41732	0.13034	0.87238	0.23248	0.8351	0.6680	0.27403	0.42267
Spectral (dft)	0.5188	0.97025	0.11017	0.21241	0.0	0.0	0	0.70141
non overlapping template matching	1.0	1.0	1.0	1.0	1.0	1.0	0.99999	1.0
overlapping template matching	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
maurers universal	0.87378	0.41008	0.32131	0.90424	0.0	0.0	0.0	0.0
linear complexity	0.02116	0.90444	0.66187	0.40974	0.07821	0.72340	0.8196	0.96489
Serial	(0.15763, 0.97128)	(0.48585, 0.36874)	(0.08526, 0.4088)	(0.08526, 0.40881)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
Approximate entropy	0.15762	0.48584	0.04653	0.08525	0.0	0	0.0	0.0
cumulative sums	(0.00463, 0.00502)	(0.31615, 0.23325)	(0.05785, 0.06310)	(0.33323, 0.46560)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
random excursion $x = -4$	0.81758	0.77005	0.89501	0.01337	0.68502	0.86836	0.02697	0.69493
random excursion $x = -3$	0.42183	0.50305	0.95679	0.28226	0.44370	0.76127	0.04336	0.83775
random excursion $x = -2$	0.95911	0.61655	0.92114	0.0906	0.39231	0.50231	0.000001	0.71308
random excursion $x = -1$	0.92394	0.06198	0.44488	0.21993	0.05059	0.39755	0.01036	0.68146
random excursion $x = 1$	0.64727	0.07625	0.19236	0.16096	0.35359	0.25673	0.96257	0.4132
random excursion $x = 2$	0.24511	0.58911	0.11843	0.45651	0.33907	0.01470	0.99698	0.88766
random excursion $x = 3$	0.17254	0.88776	0.13784	0.24313	0.37950	0.00178	0.99922	0.11974
random excursion $x = 4$	0.7213	0.27047	0.5715	0.41976	0.59339	0.00011	0.99978	0.79418
random excursion variant $x = -9$	0.33558	0.90717	0.33129	0.05356	0.40513	0.53635	0.17007	0.13698
random excursion variant $x = -8$	0.52838	0.91869	0.25061	0.1335	0.38889	0.51036	0.20124	0.05437
random excursion variant $x = -7$	0.72358	0.79963	0.31722	0.29155	0.45903	0.47950	0.16981	0.06413
random excursion variant $x = -6$	0.57244	0.93139	0.45361	0.21843	0.60483	0.44207	0.05501	0.19806
random excursion variant $x = -5$	0.4003	0.92934	0.43297	0.2066	0.97465	0.39542	0.00952	0.33844
random excursion variant $x = -4$	0.37782	0.84358	0.32641	0.25193	0.49353	0.33523	0.01616	0.31856
random excursion variant $x = -3$	0.65639	0.45278	0.34576	0.05905	0.52243	0.25421	0.2059	0.42467
random excursion variant $x = -2$	0.97501	0.23939	0.485	0.01502	0.69999	0.21295	0.22067	0.68673
random excursion variant $x = -1$	0.94952	0.89322	0.48452	0.0321	0.50450	0.11666	0.1573	0.43766
random excursion variant $x = 1$	0.59362	0.56961	1.0	0.03362	0.07005	0.07756	0.4795	0.81588
random excursion variant $x = 2$	0.97917	0.66431	0.9082	0.04791	0.07815	0.07004	0.68309	0.75377
random excursion variant $x = 3$	0.41162	0.6034	0.88949	0.1526	0.23251	0.21949	0.75183	0.29773
random excursion variant $x = 4$	0.20966	0.44556	0.48631	0.57272	0.54017	0.94091	0.78927	0.19678
random excursion variant $x = 5$	0.12418	0.22361	0.51268	0.84155	0.56727	0.89598	0.81366	0.48484
random excursion variant $x = 6$	0.03772	0.13563	0.74301	0.65679	0.47237	0.51541	0.83117	0.52748
random excursion variant $x = 7$	0.02872	0.14485	0.93867	0.61234	0.52569	0.30139	0.84452	0.30148
random excursion variant $x = 8$	0.04955	0.28153	0.77011	0.64054	0.47527	0.41783	0.85513	0.3779
random excursion variant $x = 9$	0.06867	0.38781	0.79821	0.70145	0.39221	0.3919	0.86383	0.54692

Table 2: Randomness test computed p-values for all of the tests within the NIST SP800-22 Rev 1a testsuite, for all 8 parameter variations of the quantum annealing experiments, executed on `DW_2000Q_LANL`.

clearly that there was too much bias in the computation on the D-Wave 2000Q device to produce high quality random bit sequences.

Figures 2 and 3 show bit map plots of a small subset of the QA measurements, for tests 5 and 6 respectively. Notice that in Figure 2 there are clearly time correlated trends (seen as vertical lines). Figure 3 contains some periodic visual trends across groups of many qubits that are time correlated, but are less pronounced than Figure 2.

Lastly, Table 3 enumerates the `min-entropy` estimates of the entirety of the QA bitstrings for all 8 device settings, which shows that the post-processed datasets have a higher (better) `min-entropy` compared to the raw measurements.

Test name	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8
Most Common Value	0.999842	0.999891	0.999866	0.999895	0.923050	0.963594	0.926796	0.989233
Collision Test	0.988907	0.988040	0.986848	0.983599	0.861706	0.954721	0.897248	1.000000
Markov Test	0.999951	0.999971	0.999949	0.999915	0.914111	0.963345	0.923366	0.990740
Compression Test	0.980403	0.965174	0.974589	0.978102	0.824339	0.847210	0.828144	0.888519
T-Tuple Test	0.949592	0.948716	0.951325	0.949373	0.893867	0.886097	0.914859	0.919266
LRS Test	0.994078	0.999435	0.981888	0.991583	0.993062	0.997555	0.978171	0.997016
Multi Most Common in Window Test	0.999935	0.999913	0.999952	0.999927	0.923147	0.967941	0.926869	0.991193
Lag Prediction Test	0.999943	0.999927	0.999891	0.999920	0.970287	0.978889	0.979499	0.983893
Multi Markov Model with Counting Test	0.999889	0.999882	0.999883	0.999938	0.921688	0.955856	0.926797	0.972710
LZ78Y Test	0.999855	0.999920	0.999896	0.999889	0.923050	0.963594	0.926796	0.989233
Overall `min-entropy`	0.949592	0.948716	0.951325	0.949373	0.824339	0.847210	0.828144	0.888519

Table 3: Min-entropy estimates (h_{Original}) for all quantum annealing experiment binary datasets, executed on `DW_2000Q_LANL`. For each of the 8 datasets, the overall min-entropy is the minimum h_{Original} computed across the suite of tests. The min-entropy, like standard information entropy, is maximized for a uniform distribution, and in this case (for bitstrings) the maximum entropy is 1 and the closer the entropy is to 0 corresponds to more biased samples.

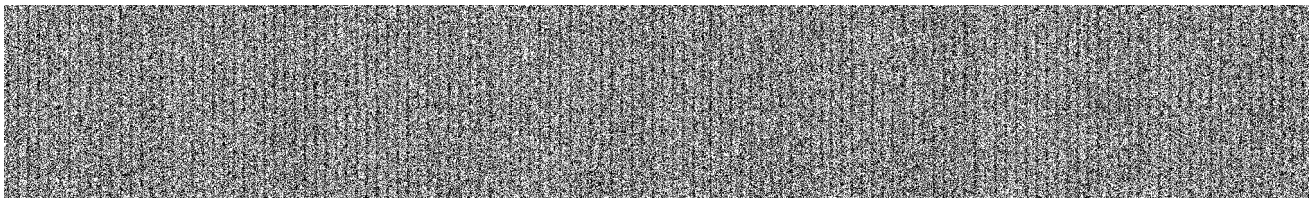

Figure 2: Bit plot of a subset of the D-Wave QRNG measurements visually showing +1 and −1 qubit states, for the 1 microsecond annealing time and no server side post processing runs (Test 5). There are 2032 row indices, corresponding to the 2032 qubit indices, and 300 column indices corresponding to 300 time ordered anneal-readout cycles. Noticeably, there are clearly correlations in the time ordered bit sequences.

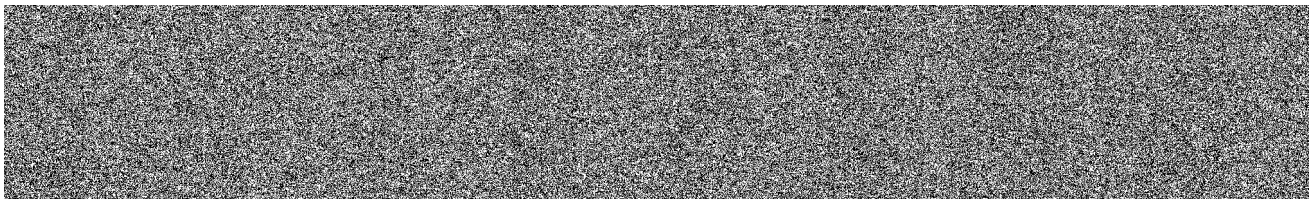

Figure 3: Bit plot of a subset of the D-Wave QRNG measurements visually showing +1 and −1 qubit states, for the 2000 microsecond annealing time and no server side post processing runs (Test 6). There are 2032 row indices, corresponding to the 2032 qubit indices, and 300 column indices corresponding to 300 time ordered anneal-readout cycles.

4 Discussion and Conclusion

Even if QRNGs based on near term devices are fundamentally non-deterministic, noise present in the computation can still produce biased random bitstrings. This is what is observed in this D-Wave quantum annealer data. This is not unexpected, especially given the observed trends over time on multiple D-Wave quantum annealers [27]. However, it is important to note that this type of biased random sampling is very likely to occur with other Noisy Intermediate Scale Quantum (NISQ) computers [63]. It is necessary that extensive tests, such as the ones presented in this paper, must be executed in order for such quantum devices to pass the threshold of being unbiased random bit samplers. Importantly, even large testsuites can not absolutely determine that the generator is indeed random - there are only tests which can show that a bit sequence is not random, or in other words the null hypothesis can never be proven to be true, it can only be observed to fail. Indeed, it has been shown that the NIST SP 800-22 testsuite is not sufficiently rigorous for verifying randomness [2, 64, 65]. However, it does serve as a reasonable minimum threshold test, which in this case the D-Wave 2000Q device did not pass.

Interestingly, within each QA dataset there are sometimes only a few tests which failed, but on the whole many of the tests were passed with p-values much greater than 0.01. In terms of the `min-entropy` measure, all 4 device settings which used server-side sampling post processing had a higher (e.g. better) min-entropy compared to the raw results. Across the battery of tests on the QA bitstrings with no post-processing, the 1 microsecond annealing time bitstrings had a min-entropy of 0.824339, the 2000 microsecond annealing time bitstrings had a min-entropy of 0.847210, the 10 microsecond annealing time bitstrings had a min-entropy of 0.828144, and the 100 microsecond annealing time bitstrings had a min-entropy of 0.888519.

Evaluating this source of random numbers using more comprehensive testsuites, such as *dieharder* [66] would be good - the limitation is that those tools require a significant amount of data to be analyzed (more than used in this analysis), which is currently not feasible to obtain using cloud based quantum computer access. In general, we expect that longer coherence times [55, 56] and lower error rates of manufactured quantum annealers would correspond to being able to produce higher quality random bit strings.

A potentially interesting analysis on this existing data from `DW_2000Q_LANL` would be to determine if there are strong cross-qubit correlations on the chip. If such correlations exist, then this could indicate cross-talk errors from the control system.

5 Acknowledgments

This work was supported by the U.S. Department of Energy through the Los Alamos National Laboratory. Los Alamos National Laboratory is operated by Triad National Security, LLC, for the National Nuclear Security Administration of U.S. Department of Energy (Contract No. 89233218CNA000001). Research presented in this article was supported by the NNSA's Advanced Simulation and Computing Beyond Moore's Law Program at Los Alamos National Laboratory. This research used resources provided by the Darwin testbed, including usage of Charliecloud containerization [62], at Los Alamos National Laboratory (LANL) which is funded by the Computational Systems and Software Environments subprogram of LANL's Advanced Simulation and Computing program (NNSA/DOE). The research presented in this article was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20220656ER and 20190065DR. This research used resources provided by the Los Alamos National Laboratory Institutional Computing Program, which is supported by the U.S. Department of Energy National Nuclear Security Administration under Contract No. 89233218CNA000001. This work has been assigned the LANL report number LA-UR-23-23112.

References

[1] Andrew Rukhin et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. Tech. rep. NIST-SP-800-02. Booz-Allen And Hamilton Inc, 2001. URL: `https://apps.dtic.mil/sti/tr/pdf/ADA393366.pdf`.

[2] Darren Hurley-Smith, Constantinos Patsakis, and Julio Hernandez-Castro. "On the Unbearable Lightness of FIPS 140–2 Randomness Tests". In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 3946–3958. DOI: `10.1109/TIFS.2020.2988505`.

[3] Dongyu Chen et al. "Error Analysis of NIST SP 800-22 Test Suite". In: *IEEE Transactions on Information Forensics and Security* 18 (2023), pp. 3745–3759. DOI: `10.1109/TIFS.2023.3287391`.

[4] Nhan Duy Truong et al. "Machine Learning Cryptanalysis of a Quantum Random Number Generator". In: *IEEE Transactions on Information Forensics and Security* 14.2 (2019), pp. 403–414. DOI: 10.1109/TIFS.2018.2850770.

[5] Luca Crocetti et al. "Review of Methodologies and Metrics for Assessing the Quality of Random Number Generators". In: *Electronics* 12.3 (2023). ISSN: 2079-9292. DOI: 10.3390/electronics12030723. URL: https://www.mdpi.com/2079-9292/12/3/723.

[6] A.B. Finnila et al. "Quantum annealing: A new method for minimizing multidimensional functions". In: *Chemical Physics Letters* 219.5–6 (Mar. 1994), 343–348. ISSN: 0009-2614. DOI: 10.1016/0009-2614(94)00117-0. URL: http://dx.doi.org/10.1016/0009-2614(94)00117-0.

[7] Giuseppe E Santoro and Erio Tosatti. "Optimization using quantum mechanics: quantum annealing through adiabatic evolution". In: *Journal of Physics A: Mathematical and General* 39.36 (2006), R393. DOI: 10.1088/0305-4470/39/36/R01.

[8] Tadashi Kadowaki and Hidetoshi Nishimori. "Quantum annealing in the transverse Ising model". In: *Physical Review E* 58.5 (1998), pp. 5355–5363. DOI: 10.1103/physreve.58.5355. URL: https://doi.org/10.1103%2Fphysreve.58.5355.

[9] Satoshi Morita and Hidetoshi Nishimori. "Mathematical foundation of quantum annealing". In: *Journal of Mathematical Physics* 49.12 (2008), p. 125210. DOI: 10.1063/1.2995837.

[10] Arnab Das and Bikas K Chakrabarti. "Colloquium: Quantum annealing and analog quantum computation". In: *Reviews of Modern Physics* 80.3 (2008), p. 1061. DOI: 10.1103/revmodphys.80.1061.

[11] Philipp Hauke et al. "Perspectives of quantum annealing: Methods and implementations". In: *Reports on Progress in Physics* 83.5 (2020), p. 054401. DOI: 10.1088/1361-6633/ab85b8.

[12] Giuseppe E Santoro et al. "Theory of quantum annealing of an Ising spin glass". In: *Science* 295.5564 (2002), pp. 2427–2430. DOI: 10.1126/science.1068774.

[13] Mark W Johnson et al. "Quantum annealing with manufactured spins". In: *Nature* 473.7346 (2011), pp. 194–198.

[14] Sergio Boixo et al. "Experimental signature of programmable quantum annealing". In: *Nature communications* 4.1 (2013), pp. 1–8. DOI: 10.1038/ncomms3067.

[15] T. Lanting et al. "Entanglement in a Quantum Annealing Processor". In: *Phys. Rev. X* 4 (2 2014), p. 021041. DOI: 10.1103/PhysRevX.4.021041. URL: https://link.aps.org/doi/10.1103/PhysRevX.4.021041.

[16] Davide Venturelli et al. "Quantum Optimization of Fully Connected Spin Glasses". In: *Physical Review X* 5.3 (2015). DOI: 10.1103/physrevx.5.031040. URL: https://doi.org/10.1103%2Fphysrevx.5.031040.

[17] R Harris et al. "Phase transitions in a programmable quantum spin glass simulator". In: *Science* 361.6398 (2018), pp. 162–165.

[18] Sergio Boixo et al. "Computational multiqubit tunnelling in programmable quantum annealers". In: *Nature communications* 7.1 (2016), p. 10327. DOI: 10.1038/ncomms10327.

[19] Andrew D King et al. "Scaling advantage over path-integral Monte Carlo in quantum simulation of geometrically frustrated magnets". In: *Nature communications* 12.1 (2021), pp. 1–6. DOI: 10.1038/s41467-021-20901-5.

[20] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin. "Quantum random number generators". In: *Rev. Mod. Phys.* 89 (1 2017), p. 015004. DOI: 10.1103/RevModPhys.89.015004. URL: https://link.aps.org/doi/10.1103/RevModPhys.89.015004.

[21] Xiongfeng Ma et al. "Quantum random number generation". In: *npj Quantum Information* 2.1 (2016), pp. 1–9. DOI: 10.1038/npjqi.2016.21.

[22] Bing Bai et al. "18.8 Gbps real-time quantum random number generator with a photonic integrated chip". In: *Applied Physics Letters* 118.26 (June 2021). ISSN: 1077-3118. DOI: 10.1063/5.0056027. URL: http://dx.doi.org/10.1063/5.0056027.

[23] Xiaomin Guo et al. "Parallel real-time quantum random number generator". In: *Opt. Lett.* 44.22 (2019), pp. 5566–5569. DOI: 10.1364/OL.44.005566. URL: https://opg.optica.org/ol/abstract.cfm?URI=ol-44-22-5566.

[24] T. Lanting et al. *Probing Environmental Spin Polarization with Superconducting Flux Qubits*. 2020. arXiv: 2003.14244 [quant-ph].

[25] Paul Kairys et al. "Simulating the Shastry-Sutherland Ising Model Using Quantum Annealing". In: *PRX Quantum* 1 (2 2020), p. 020320. DOI: 10.1103/PRXQuantum.1.020320. URL: https://link.aps.org/doi/10.1103/PRXQuantum.1.020320.

[26] Andrew D. King, Trevor Lanting, and Richard Harris. *Performance of a quantum annealer on range-limited constraint satisfaction problems*. 2015. DOI: 10.48550/ARXIV.1502.02098. URL: https://arxiv.org/abs/1502.02098.

[27] Elijah Pelofske, Georg Hahn, and Hristo N Djidjev. "Noise dynamics of quantum annealers: estimating the effective noise using idle qubits". In: *Quantum Science and Technology* 8.3 (2023), p. 035005. DOI: 10.1088/2058-9565/accbe6. URL: https://doi.org/10.1088%2F2058-9565%2Faccbe6.

[28] Takayuki Suzuki and Hiromichi Nakazato. *A proposal of noise suppression for quantum annealing*. 2020. DOI: 10.48550/ARXIV.2006.13440. URL: https://arxiv.org/abs/2006.13440.

[29] Jessica Park, Susan Stepney, and Irene D'Amico. "Spatial Correlations in the Qubit Properties of D-Wave 2000Q Measured and Simulated Qubit Networks". In: *Unconventional Computation and Natural Computation*. Cham: Springer Nature Switzerland, 2023, pp. 140–154. DOI: 10.1007/978-3-031-34034-5_10. arXiv: 2305.07385 [quant-ph].

[30] Thomas Krauss et al. "Statistical bias in D-wave qubits". In: *Journal of Physics: Conference Series*. Vol. 1936. 1. IOP Publishing. 2021, p. 012010.

[31] Elijah Pelofske, Georg Hahn, and Hristo N. Djidjev. "Reducing Quantum Annealing Biases for Solving the Graph Partitioning Problem". In: *Proceedings of the 18th ACM International Conference on Computing Frontiers*. CF '21. Virtual Event, Italy: Association for Computing Machinery, 2021, 133–139. ISBN: 9781450384049. DOI: 10.1145/3457388.3458672. URL: https://doi.org/10.1145/3457388.3458672.

[32] Yoshiki Matsuda, Hidetoshi Nishimori, and Helmut G Katzgraber. "Quantum annealing for problems with ground-state degeneracy". In: *Journal of Physics: Conference Series*. Vol. 143. 1. IOP Publishing. 2009, p. 012003. DOI: 10.1088/1742-6596/143/1/012003.

[33] Mario S. Könz et al. "Uncertain fate of fair sampling in quantum annealing". In: *Phys. Rev. A* 100 (3 2019), p. 030303. DOI: 10.1103/PhysRevA.100.030303. URL: https://link.aps.org/doi/10.1103/PhysRevA.100.030303.

[34] Salvatore Mandrà, Zheng Zhu, and Helmut G. Katzgraber. "Exponentially Biased Ground-State Sampling of Quantum Annealing Machines with Transverse-Field Driving Hamiltonians". In: *Phys. Rev. Lett.* 118 (7 2017), p. 070502. DOI: 10.1103/PhysRevLett.118.070502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.118.070502.

[35] Zheng Zhu, Andrew J. Ochoa, and Helmut G. Katzgraber. "Fair sampling of ground-state configurations of binary optimization problems". In: *Phys. Rev. E* 99 (6 2019), p. 063314. DOI: 10.1103/PhysRevE.99.063314. URL: https://link.aps.org/doi/10.1103/PhysRevE.99.063314.

[36] Brian Hu Zhang et al. "Advantages of unfair quantum ground-state sampling". In: *Scientific reports* 7.1 (2017), pp. 1–12. DOI: 10.1038/s41598-017-01096-6.

[37] Harshil Bhatia et al. "Generation of Truly Random Numbers on a Quantum Annealer". In: *IEEE Access* 10 (2022), pp. 112832–112844. DOI: 10.1109/ACCESS.2022.3215500.

[38] Rick Picard Sarah Michalak. *Leveraging LANL's D-WAVE 2X for Random Number Generation*. https://www.lanl.gov/projects/national-security-education-center/information-science-technology/dwave/assets/michalak_dwave2017.pdf. 2017.

[39] Kentaro Tamura and Yutaka Shikano. "Quantum Random Numbers Generated by a Cloud Superconducting Quantum Computer". In: *International Symposium on Mathematics, Quantum Theory, and Cryptography*. Springer Singapore, 2020, pp. 17–37. DOI: 10.1007/978-981-15-5191-8_6. URL: https://doi.org/10.1007\%2F978-981-15-5191-8_6.

[40] Yuanhao Li et al. "Quantum random number generator using a cloud superconducting quantum computer based on source-independent protocol". In: *Scientific Reports* 11.1 (2021), pp. 1–11.

[41] Janusz E Jacak et al. "Quantum random number generators with entanglement for public randomness testing". In: *Scientific Reports* 10.1 (2020), pp. 1–9. DOI: s41598-019-56706-2.

[42] Kentaro Tamura and Yutaka Shikano. *Quantum Random Number Generation with the Superconducting Quantum Computer IBM 20Q Tokyo*. Cryptology ePrint Archive, Paper 2020/078. https://eprint.iacr.org/2020/078. 2020. URL: https://eprint.iacr.org/2020/078.

[43] Randy Kuang et al. "Pseudo Quantum Random Number Generator with Quantum Permutation Pad". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 359–364. DOI: 10.1109/QCE52317.2021.00053.

[44] Aviraj Sinha et al. *A Programmable True Random Number Generator Using Commercial Quantum Computers*. 2023. arXiv: 2304.03830 [quant-ph].

[45] Jinjing Shi et al. *An Unbiased Quantum Random Number Generator Based on Boson Sampling*. 2022. arXiv: 2206.02292 [quant-ph].

[46] Anupam Sarkar and C. M. Chandrashekar. "Multi-bit quantum random number generation from a single qubit quantum walk". In: *Scientific Reports* 9.1 (2019). DOI: 10.1038/s41598-019-48844-4. URL: https://doi.org/10.1038%2Fs41598-019-48844-4.

[47] Yu-Guang Yang and Qian-Qian Zhao. "Novel pseudo-random number generator based on quantum random walks". In: *Scientific reports* 6.1 (2016), pp. 1–11. DOI: 10.1038/srep20362.

[48] Yang Liu et al. "Device-independent quantum random-number generation". In: *Nature* 562.7728 (2018), pp. 548–551. DOI: 10.1038/s41586-018-0559-3. URL: https://doi.org/10.1038%2Fs41586-018-0559-3.

[49] Andrew W. Cross et al. "Validating quantum computers using randomized model circuits". In: *Phys. Rev. A* 100 (3 2019), p. 032328. DOI: 10.1103/PhysRevA.100.032328. URL: https://link.aps.org/doi/10.1103/PhysRevA.100.032328.

[50] Charles H. Baldwin et al. "Re-examining the quantum volume test: Ideal distributions, compiler optimizations, confidence intervals, and scalable resource estimations". In: *Quantum* 6 (May 2022), p. 707. ISSN: 2521-327X. DOI: 10.22331/q-2022-05-09-707. URL: http://dx.doi.org/10.22331/q-2022-05-09-707.

[51] Elijah Pelofske, Andreas Bärtschi, and Stephan Eidenbenz. "Quantum Volume in Practice: What Users Can Expect From NISQ Devices". In: *IEEE Transactions on Quantum Engineering* 3 (2022), 1–19. ISSN: 2689-1808. DOI: 10.1109/tqe.2022.3184764. URL: http://dx.doi.org/10.1109/TQE.2022.3184764.

[52] Gabriele Cenedese et al. "Generation of Pseudo-Random Quantum States on Actual Quantum Processors". In: *Entropy* 25.4 (2023). ISSN: 1099-4300. DOI: 10.3390/e25040607. URL: https://www.mdpi.com/1099-4300/25/4/607.

[53] Elijah Pelofske. *Dataset for Analysis of a Programmable Quantum Annealer as a Random Number Generator*. Jan. 2024. DOI: 10.5281/zenodo.10583977. URL: https://doi.org/10.5281/zenodo.10583977.

[54] *D-Wave Post Processing*. https://web.archive.org/web/20231122203833/https://docs.dwavesys.com/docs/latest/c_qpu_pp.html.

[55] Andrew D. King et al. "Coherent quantum annealing in a programmable 2,000 qubit Ising chain". In: *Nature Physics* 18.11 (2022), pp. 1324–1328. DOI: 10.1038/s41567-022-01741-6. URL: https://doi.org/10.1038%2Fs41567-022-01741-6.

[56] Andrew D. King et al. "Quantum critical dynamics in a 5,000-qubit programmable spin glass". In: *Nature* 617.7959 (2023), pp. 61–66. DOI: 10.1038/s41586-023-05867-2. URL: https://doi.org/10.1038%2Fs41586-023-05867-2.

[57] Neil G Dickson et al. "Thermally assisted quantum annealing of a 16-qubit problem". In: *Nature communications* 4.1 (2013), p. 1903. DOI: 10.1038/ncomms2920.

[58] Lawrence Bassham et al. *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*. en. 2010. URL: https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762.

[59] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (Sept. 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2.

[60] C. E. Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: 10.1002/j.1538-7305.1948.tb01338.x.

[61] Meltem Sönmez Turan et al. "Recommendation for the entropy sources used for random bit generation". In: *NIST Special Publication* 800.90B (2018), p. 102. DOI: 10.6028/NIST.SP.800-90B. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90B.pdf.

[62] Reid Priedhorsky and Tim Randles. "Charliecloud: Unprivileged Containers for User-Defined Software Stacks in HPC". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '17. Denver, Colorado: Association for Computing Machinery, 2017. ISBN: 9781450351140. DOI: 10.1145/3126908.3126925. URL: https://doi.org/10.1145/3126908.3126925.

[63] John Preskill. "Quantum Computing in the NISQ era and beyond". In: *Quantum* 2 (2018), p. 79. DOI: 10.22331/q-2018-08-06-79. URL: https://doi.org/10.22331%2Fq-2018-08-06-79.

[64] Carmina Georgescu et al. "A view on NIST randomness tests (In)Dependence". In: *2017 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. 2017, pp. 1–4. DOI: 10.1109/ECAI.2017.8166460.

[65] Kinga Marton and Alin Suciu. "On the interpretation of results from the NIST statistical test suite". In: *Science and Technology* 18.1 (2015), pp. 18–32.

[66] Robert G Brown, Dirk Eddelbuettel, and David Bauer. "Dieharder". In: *Duke University Physics Department Durham, NC* (2018), pp. 27708–0305.