The Quantum Computing Company™

A Head-to-Head Comparison of D-Wave and Rigetti QPUs

WHITEPAPER

Summary

Gate-model quantum computers are theoretically capable of exceptional performance in certain applications, although it is unclear how useful they will be in general. The Quantum Approximate Optimization Algorithm (QAOA) of Farhi et al. [1] has been proposed as a possible path towards making gate-model quantum computers effective at solving problems in combinatorial optimization [2, 3].

Recently, Rigetti Computing published results of QAOA run on their 19-qubit gate-model quantum computer [4]. The inputs they considered can also be solved on D-Wave quantum annealing systems, providing an opportunity to compare the two quantum processing units (QPUs) directly. Reproducing their tests, we found the probabilities of returning an optimal solution to be 99.6% for the D-Wave 2000Q and 0.001% for the Rigetti 19Q. In addition, the D-Wave 2000Q was able to solve 102 copies of the problem in parallel. The advantages in quality and size of the D-Wave 2000Q, taken together, provide an improvement of 10 million times in terms of ground-state throughput per sample.

Also notable in Ref. [4] are results for QAOA running on a classical simulation of a noiseless gate-model quantum computer. Even when running on a small, easy input using ideal hardware, QAOA success probabilities appear to be four orders of magnitude lower than D-Wave QPU success probabilities. This indicates that the single step of QAOA used in Ref. [4] is insufficient. Running more steps of QAOA as necessary will require significantly higher coherence and lower error rates.

The results in Ref. [4] do not provide evidence that QAOA will be practical for solving these problems on near-term gate-model devices.

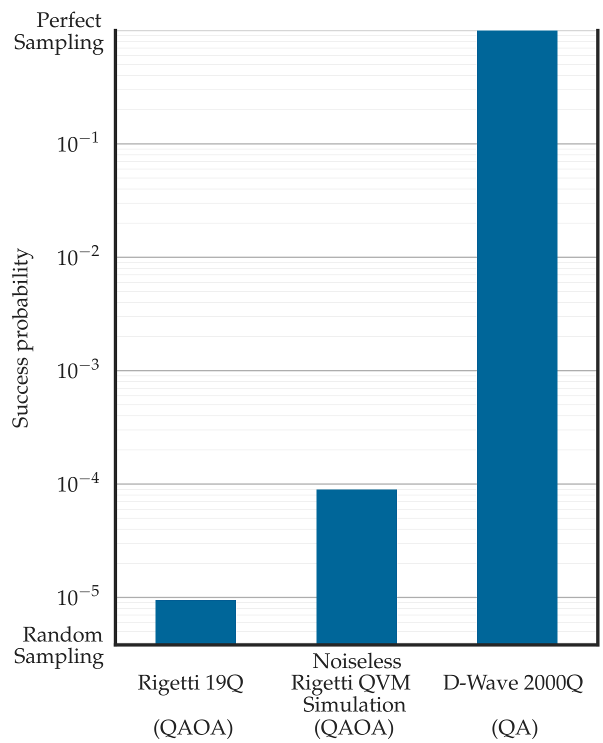

Figure 1: Success probabilities for Rigetti 19Q, Rigetti Quantum Virtual Machine (QVM) simulator, and D-Wave 2000Q on the 19-qubit input from Otterbach et al. [4]. The Rigetti 19Q is a gate-model quantum computer and the Rigetti QVM is a noiseless classical simulator, in this case simulating the 19Q. The Rigetti QPU and simulator both run the quantum approximate optimization algorithm (QAOA) [1]. The D-Wave 2000Q runs the quantum annealing algorithm (QA) [5]. The success probabilities from QAOA are at least 4 orders of magnitude lower than those from the D-Wave quantum annealing system.

1 Introduction

Until recently, there has not been the opportunity to compare the D-Wave quantum processing unit (QPU) with any quantum competitor. This is in part because the competition is far less mature, and in part because all notable competitors are gate-model quantum computers that specialize in different classes of problems than D-Wave's quantum-annealing QPUs. However, Rigetti Computing recently published a study [4] demonstrating the use of the Quantum Approximate Optimization Algorithm (QAOA) of Farhi et al. [1], which has been proposed as a means to make gate-model quantum computers useful for applications in combinatorial optimization. D-Wave QPUs are flexible platforms for heuristically solving optimization problems. This provides an opportunity to compare the two directly.

2 Inputs from the Rigetti study

The problems studied by Otterbach et al. [4], i.e., the Rigetti inputs, are Ising models with all antiferromagnetic (AFM) couplings. This restricted subclass of Ising model inputs is computationally difficult (NP-complete) in general. However, when the underlying graph is bipartite, as is the case for the Rigetti inputs, the problems are trivial: Ising models with all AFM couplings cannot have any frustration on bipartite graphs because there are no odd cycles. Seen another way, if the underlying graph is bipartite, a spin-reversal transformation can be applied to make all couplings ferromagnetic (FM); a ferromagnet is trivial to solve in linear time using a greedy method.

The authors describe their inputs as examples of unsupervised machine learning, specifically, clustering. However, this problem is equivalent to MAX-CUT, which is a type of discrete optimization problem.

The specific coupling values chosen for the Rigetti inputs were randomly generated, though the exact generation method is not clear from the paper. Coupling values appear to be between 0 and 1, and Figure 3 of Ref. [4] shows one of the inputs (we show this as Figure 2). The input from this figure is the one we use for testing on the D-Wave 2000Q system.

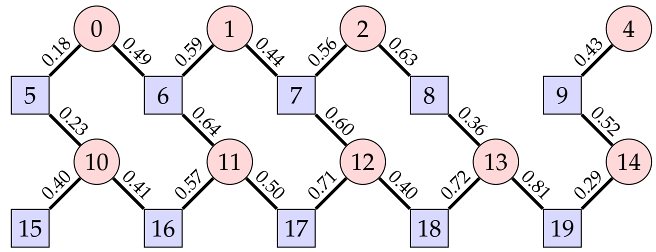

Figure 2: Working graph of the Rigetti 19Q, along with an input. The partition of the qubits into two disjoint independent sets is indicated by red circles and blue squares. Qubit 3 is not usable and is therefore omitted from the figure. Edge labels indicate the coupling values of one of the Rigetti inputs provided, which is the input we used in our experiments.

3 Rigetti 19Q

The inputs from the Rigetti study have 19 binary variables and a solution space of 2^{19} states, two of which are ground states. Random sampling from the solution space therefore has a success probability of $2/2^{19} = 2^{-18}$. In the remainder of this section, we show how we infer the success probability of the Rigetti 19Q as slightly less than 1/100,000.

Performance numbers for the Rigetti 19Q are extracted from the text and plots of the Rigetti Computing paper by Otterbach et al. [4], posted to arXiv on December 15, 2017. We specifically cite version 1 of this arXiv posting so that we can refer to figures in that paper by number.

Because the authors do not explicitly state the success probability (ground-state probability) for their processor, we must infer it; we rely primarily on Figure 5 of Ref. [4].

Our calculation of the ground-state probability is as follows:

- The probability of reaching a ground state after 55 steps of 2500 samples each is roughly 73% (see Figure 5 of Ref. [4]).
- We make the simplifying assumption that the samples are independent and identically distributed. While this should not be true in general, particularly if the parameters of QAOA make a difference, in both Figure 5 and Figure S1 of Ref. [4], the success probability does not increase perceptibly over the course of the 55 steps.

- Where N is the total number of independent Bernoulli trials and p_T is the probability of having at least 1 success (i.e., ground state), the success probability p for a single trial is as follows:

$$(1-p)^N = 1 - p_T \qquad (1)$$
$$p = 1 - \exp \frac{\log(1-p_T)}{N}. \qquad (2)$$

- Using values of $N = 55 \cdot 2500$ and $p_T = 0.73$ in Equation 2, we get a success probability of $p = 9.52 \times 10^{-6}$, or slightly less than 1/100,000.

- To check our method, we do the same for the random sampler, for which p_T is roughly 0.4. We get a success probability of $p = 3.72 \times 10^{-6}$, giving our method an error of 2% relative to the true value of $2^{-18} = 3.81 \times 10^{-6}$.

We note that the Rigetti QPU success probabilities are no more than three times better than random sampling, and that tuning QAOA parameters using Bayesian optimization does not appear to have any effect on performance. This can be explained by a mismatch between the design purpose of QAOA and the metric used to evaluate its performance: this issue is addressed in the next section.

4 Simulated QAOA

In addition to the empirical study of QAOA, Otterbach et al. [4] simulated an error-free quantum computer running QAOA as it was implemented on the Rigetti QPU (see 'Rigetti QVM' in Figure 5 of Ref. [4]). While QVM results were significantly better than those for the Rigetti QPU, they were unimpressive considering the trivial nature of the inputs.

Although success probabilities for the Rigetti QVM are slightly more challenging to estimate from Figure 5 of Ref. [4], they appear to be consistent with independent Bernoulli trials where each batch of 2500 samples contains a ground state with 20% probability. (While one would not expect the assumption of independent Bernoulli trials to hold, the empirical data from the Rigetti QVM simulator does not appear to reject this null hypothesis.) This gives a single-sample success probability of 9×10^{-5}.

QAOA is an approximation algorithm that (under ideal closed-system conditions that exist on the QVM) guarantees to find a cut that is within a fixed ratio $\rho \leq 1$ of optimal. The value of ρ depends on a parameter $\mathbf{p}$ corresponding to the depth of the circuit used to implement QAOA. For example, Farhi et al. [1] show that on bipartite graphs with $\mathbf{p} = 2$, the algorithm always finds cuts at least $\rho = 0.7559$ as good as optimal; this ratio increases with $\mathbf{p}$.

Note that having a bound on ρ for low-valued $\mathbf{p}$ says nothing about the probability of finding optimal solutions, and there is no reason to expect QAOA to perform well under the success-probability metric used in Ref. [4]. Given that the authors were only able to implement a single step of QAOA ($\mathbf{p} = 1$) on their QPU, the performance of QAOA on QVM under this metric suggests that low expectations are appropriate.

Recent empirical studies of QAOA in error-free simulation [6, 7] have shown that performance improves as the number of steps $\mathbf{p}$ increases. A deeper study [8] shows that $1 - r$, the relative distance to optimal, actually decays exponentially as a function of $\mathbf{p}$. Zhou et al. [8] also provide an efficient heuristic for finding approximately optimal parameters for QAOA, removing a significant barrier to its practical use. A similar study measuring the effects of errors and decoherence on the performance of QAOA is a crucial next step in evaluating the algorithm's potential.

5 D-Wave 2000Q

The working graph of the Rigetti 19Q can be embedded into a D-Wave 2000Q graph using 2.5 Chimera unit tiles.[1] Two copies of the Rigetti graph can be placed in 5 Chimera tiles (a 3×2 Chimera graph with one corner tile removed). This 5-tile double embedding can then be placed many times on a large Chimera graph. These experiments were performed on a D-Wave 2000Q QPU with all 2048 qubits available; 102 independent copies of the Rigetti 19Q graph can be placed on this graph (see Figure 3).

While the Rigetti inputs have no frustration, D-Wave

[1] A Chimera unit tile is an 8-qubit tile—4 horizontal qubits and 4 vertical qubits—that is repeated in a square grid to make the Chimera topologies on which all commercially available D-Wave QPUs to date have been built.

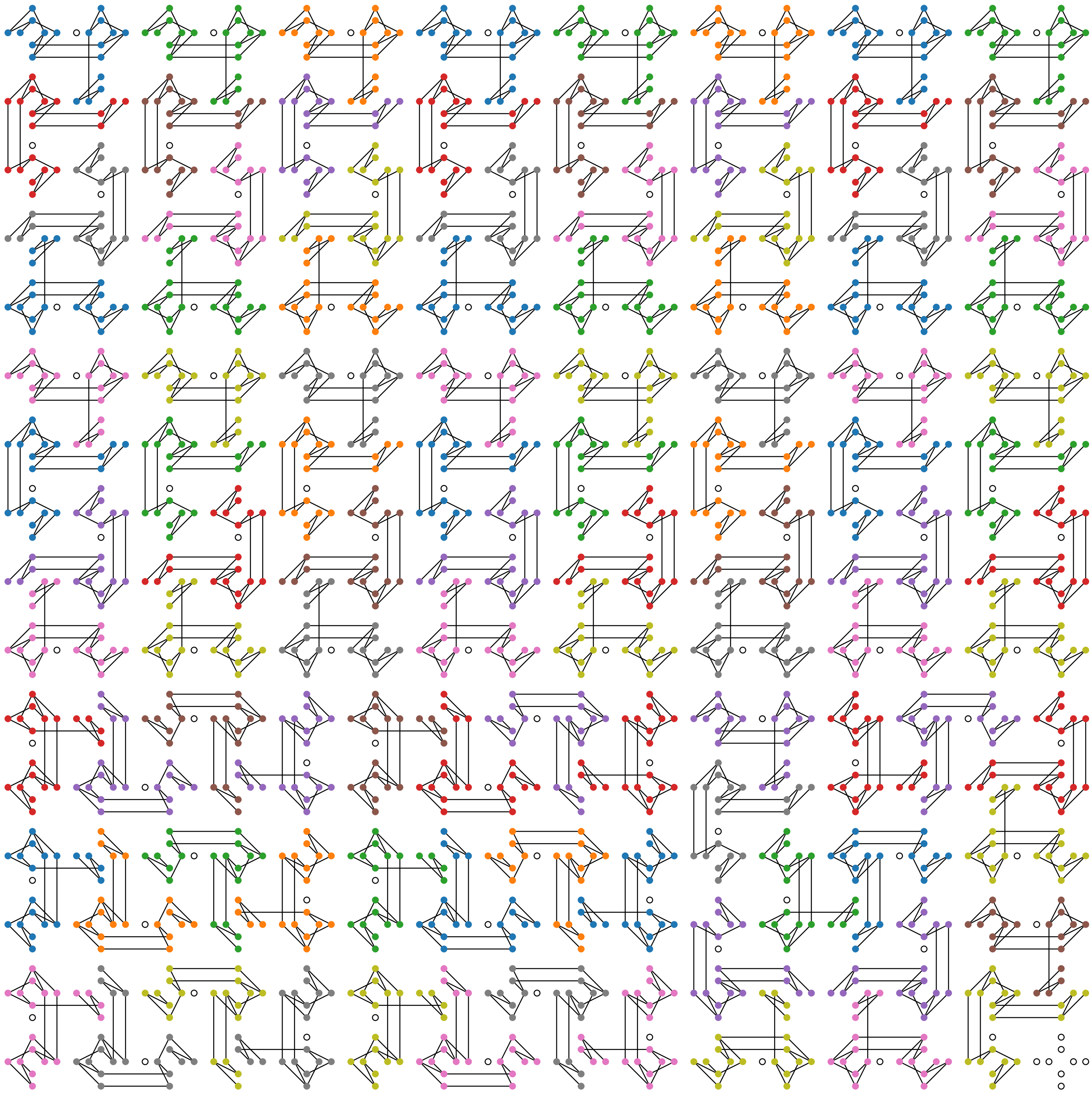

Figure 3: 102 independent copies of the Rigetti 19Q working graph embedded on a D-Wave 2000Q working graph.

QPUs may still return excited states due to thermalization, analog noise, etc. However, results like this are rare because the inputs are so small.

A test using the Rigetti input shown in Figure 2, embedded as described above on the D-Wave 2000Q system, returned the following results:[2]

- Submitting the problem as-is with autoscale off (i.e., not using the entire dynamic range of the QPU), we saw success probabilities of 98%.
- Submitting the problem as-is with autoscale on, increasing energy scales by 23%, we saw success probabilities of 99%.
- Applying a spin-reversal transformation (SRT) to the problem to turn all J values negative (FM), then exploiting the extended negative J range to double the energy scale, we further increased the success probability to 99.6%.
- Scaling the problem down to fit within 1% of the D-Wave 2000Q QPU's FM range, effectively simulating a 100-fold increase in noise and analog errors, we still saw success probabilities roughly 7 times higher than the Rigetti 19Q.

The above results held true whether we were solving one copy of the input in isolation, or whether we were solving 102 copies in parallel.

6 Wall-clock comparison

Wall-clock time is not a metric of great interest in such a lopsided comparison. However, we can determine the order of magnitude advantage that we see in the D-Wave 2000Q over the Rigetti 19Q. In the Rigetti study, the wall-clock time for 55 steps of 2500 samples was reported as approximately 10 minutes; this is the time required to return on the order of 1 ground state. The wall-clock time to receive a single sample from a D-Wave 2000Q is roughly 60 ms. Thus D-Wave can obtain a ground state in 60 ms as opposed to 600 s; this is 10,000 times faster. Note also that in this time, the D-Wave QPU obtains on the order of 100 ground states; however, because these are obtained in parallel, this factor of 100 does not contribute to the latency advantage.

It is worth noting that the D-Wave 2000Q wall-clock time could be improved significantly were it possible to program, and read-out, only 1% of the D-Wave QPU at a time. As things stand, even if only 1% of the D-Wave QPU is needed to solve a problem, the current programming API requires that all the biases and couplings in the QPU be programmed, whether they are needed to solve the problem or not. This adds an overhead that is not necessary when solving such a trivial problem. However, even with that overhead, the D-Wave QPU still beats the Rigetti chip with regard to wall-clock time by a significant factor.

7 Conclusion

In what is the most direct comparison between a D-Wave QPU and a competing QPU to date, the maturity and quality of D-Wave QPUs become clear. While the Rigetti QPU is only slightly better than random sampling at finding ground states, the D-Wave QPU returns almost nothing but ground states. In terms of ground-state probabilities, the D-Wave QPU is 100,000 times better than the Rigetti QPU. Even when we simulate a 100-fold increase in noise and misspecifications on the D-Wave 2000Q, it still significantly outperforms the Rigetti 19Q. Furthermore, because the D-Wave QPU is so much larger, it can solve 102 copies of the problem in parallel.

While the Quantum Approximate Optimization Algorithm (QAOA) has been proposed as a way to make gate-model quantum computers applicable to optimization problems [2, 9], the results of Ref. [4] cannot be used to argue that this algorithm will be useful in practice. Recent studies of QAOA in error-free simulation show promising results, with performance improving with the number of steps **p** of QAOA [6–8]. However, we are unaware of similarly promising results that account for errors and decoherence.

In conclusion, while QAOA is of great theoretical interest, quantum annealing is currently many orders of magnitude more performant in practice, and there is little evidence suggesting that this will change in the era of noisy intermediate-scale quantum computers.

[2] We report success probabilities for an individual embedding of the 19-qubit input; we run 102 copies of this input in parallel.

References

[1] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," 2014. arXiv: 1411.4028.

[2] E. Zahedinejad and A. Zaribafiyan, "Combinatorial optimization on gate model quantum computers: a survey," 2017. arXiv: 1708.05294.

[3] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018, ISSN: 2521-327X.

[4] J. S. Otterbach, R. Manenti, N. Alidoust, A. Bestwick, M. Block, *et al.*, "Unsupervised machine learning on a hybrid quantum computer," 2017. arXiv: 1712.05771v1.

[5] T. Kadowaki and H. Nishimori, "Quantum annealing in the transverse Ising model," *Physical Review E*, vol. 58, no. 5, p. 5355, 1998.

[6] G. G. Guerreschi and M. Smelyanskiy, "Practical optimization for hybrid quantum-classical algorithms," 2017. arXiv: 1701.01450.

[7] G. E. Crooks, "Performance of the quantum approximate optimization algorithm on the maximum cut problem," 2018. arXiv: 1811.08419.

[8] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, "Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices," 2018. arXiv: 1812.01041.

[9] E. Farhi and A. W. Harrow, "Quantum supremacy through the quantum approximate optimization algorithm," 2016. arXiv: 1602.07674.