LA-UR-23-22023

Approved for public release; distribution is unlimited.

Title:	Short-Depth QAOA circuits and Quantum Annealing on Higher-Order Ising Models
Author(s):	Pelofske, Elijah Autumn Rain Baertschi, Andreas Eidenbenz, Stephan Johannes
Intended for:	npj Quantum Information
Issued:	2023-06-13 (rev.2)

Short-Depth QAOA circuits and Quantum Annealing on Higher-Order Ising Models

Elijah Pelofske*[1], Andreas Bärtschi†[1], and Stephan Eidenbenz[1]

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

Abstract

The Quantum Alternating Operator Ansatz (QAOA) and Quantum Annealing (QA) are quantum algorithms that are both based on the adiabatic theorem and both have the goal of sampling the optimal solution(s) of combinatorial optimization problems. Quantum annealing has been physically instantiated on D-Wave devices using superconducting flux qubits, and QAOA can be programmed on digital gate-model quantum computers such as the programmable superconducting transmon qubits devices of the IBMQ series, for instance ibm_washington. QAOA and QA address the same types of problems, but it is unclear how they will scale to large problem sizes and to larger and higher-fidelity quantum computers. In this article, we present a direct comparison between QAOA, one and two rounds, run on all 127 qubits of ibm_washington and QA run on D-Wave Advantage_system4.1 and Advantage_system6.1. The problems which allow for this comparison are random Ising model problems whose connectivity matches the heavy hexagonal lattice topology of ibm_washington and the Pegasus graph connectivity of the two D-Wave devices. We create two classes of problem instances for this comparison: one with higher order terms (ZZZ variable interactions), linear terms, and quadratic terms, and a separate problem type with only linear and quadratic terms. Our QAOA circuits are novel and extremely short depth, with a CNOT depth of 6 per round, which allows whole chip usage of ibm_washington's heavy hexagonal lattice and can be applied to future heavy-hex chips. We also test the effectiveness of the error suppression technique digital dynamical decoupling on the QAOA circuits. The QAOA circuits compiled to ibm_washington are composed of several thousand circuit instructions, approximately $3,000$ depending on the details of the circuit, making these some the largest quantum circuits ever executed on a digital quantum processor. QAOA and QA are compared against the classical heuristic algorithm of simulated annealing and all problem instances are exactly solved using CPLEX in order to evaluate which samplers, if any, correctly found the ground state solution(s) of the problem instances. We find that (i) QA outperforms QAOA on all problem instances, (ii) QAOA samples the problems better than random sampling, and (iii) QAOA angle computation exhibits clear parameter concentration across the ensemble of Ising models.

1 Introduction

Quantum Alternating Operator Ansatz (QAOA) is a hybrid quantum-classical algorithm for sampling combinatorial optimization problems [1–3], the parameterized quantum component of which can be instantiated with a programmable gate-based universal quantum computer. The Quantum Approximate Optimization Algorithm [4] was the predecessor algorithm, which was then generalized to the Quantum Alternating Operator Ansatz algorithm [1] under the same name of QAOA. The classical component of QAOA involves learning the best parameters of the circuit to obtain low energy solutions of the combinatorial optimization problem.

Quantum annealing (QA) in the transverse field Ising model is type of quantum computation that utilizes quantum fluctuations in order to search for ground state solutions of a combinatorial optimization problem, encoded as a Hamiltonian [5–12]. D-Wave quantum annealers are programmable hardware implementations of quantum annealing that use superconducting flux qubits [13–16].

Both QAOA and Quantum Annealing are based on the adiabatic theorem, and in particular Adiabatic Quantum Computation. Both algorithms are implementing a type of adiabatic evolution, both algorithms aim to sample optimal solutions of combinatorial optimization problems, and both algorithms are being studied for demonstrating advantage over classical heuristics as quantum heuristic samplers for optimization problems [17, 18]. The exact characteristics of how both QA and QAOA will scale to large system sizes, especially on noisy hardware, is currently not fully understood [19–21]. There is evidence that QAOA may be more difficult for classical computers to simulate

*Email: epelofske@lanl.gov
†Email: baertschi@lanl.gov

than quantum annealing, which could make it a viable candidate for quantum advantage [22]. Therefore it is of interest to investigate differences between QAOA and QA and determine how these algorithms scale - both for large problem sizes and for larger QAOA rounds, or annealing times in the case of QA [23–28]. There have been a limited number of studies that directly compare Quantum Annealing and the QAOA algorithm [29–33] (or even more generally, gate model quantum computing and QA [34]), which in part motivates the direct comparison of the two protocols presented in this research. With respect to large QAOA implementations, there have been experiments that used up to 40 qubits [35], 27 qubits [36], and 23 qubits [37]. There have also been QAOA experiments which used circuit depths up to 148 [38] and 159 [39].

Because both QA and QAOA address combinatorial optimization problems, it is of considerable interest to experimentally and theoretically evaluate them. QAOA has been evaluated for a number of problems, including portfolio optimization [38], maximum cut [37, 40–46], maximum k-vertex cover [2], maximum independent set [47], the knapsack problem [48], and the sherrington-kirkpatrick model [37, 41, 49]. QA has been experimentally applied to a wide variety of problems, including semiprime factorization [50–54], graph coloring [55, 56], clustering [57], the Sherrington-Kirkpatrick model [58], and portfolio optimization [59–61]. See refs. [8, 9] for quantum annealing reviews. Both QAOA and QA have been used to simulate properties of magnetic systems [21, 62–66].

This article is a significant extension of ref. [33]. The contributions of this article are as follows:

1. We define a class of higher-order Ising model problems that have an interaction topology that fits well onto the native IBMQ heavy hex topology as well as onto D-Wave's Pegasus topology. The concept of adding optimization terms that are higher than quadratic is novel in NISQ experiments. We randomly generate and sample 10 different problems, each for cubic and quadratic maximum degree

2. We design optimized QAOA circuits for these Ising problems that are short depth, thus allowing the use of the entire chip of `ibm_washington` with 127 qubits for QAOA; we limit our experiments to two QAOA rounds. These experiments are the largest QAOA runs performed on a quantum computer to date – to the best of our knowledge

3. In order to solve the Ising models with cubic terms on D-Wave Quantum Annealing hardware, we order-reduce the Ising model instances to quadratic problems carefully ensuring that a scalable mapping to the Pegasus lattice remains viable and we then instantiate these problems on two different D-Wave machines

4. We put significant optimization efforts into both the QAOA and QA experiments. In particular, for QAOA on IBMQ, we perform angle optimization and test digital dynamical decoupling (DDD). For running QA on D-Wave devices, we vary the forward anneal schedule with symmetric pauses

5. In order to better assess the quality of the obtained NISQ results, we calculate theoretical results for the optimum solutions of our Ising problems using CPLEX, and the classical heuristic simulated annealing (giving a solution spectrum); we also calculate – pushing classical HPC capabilities somewhat – the theoretical means of the solution quality achieved by (noise-free) QAOA, as well expectation values for random solutions

The experiments on the NISQ devices of solving the same problem set are the key method for us to answer what the state-of-the-art is in QA and QAOA performance in a fair comparison. Our main insights are the following:

1. Quantum Annealing on D-Wave clearly outperforms QAOA on the `ibm_washington`. In particular, QA finds significantly better minimum energy solutions as well as better average solutions found on all Ising problems studied

2. QA finds optimum or very nearly optimum solutions for all problems and longer annealing times work the best

3. QAOA runs on IBMQ find solutions that are significantly worse than those found by QA, but they are better than random sampling. QAOA solutions from `ibm_washington` are also on average significantly worse than the theoretical noise-free QAOA expectation values

4. The optimal QAOA angles show pronounced parameter concentration - this parameter concentration is the same even between the instances with and without higher order terms. The results are consistent for the classically computed QAOA results as well as for the actual IBMQ runs. The finding of parameter concentration on this ensemble of random Ising problems is consistent with previous QAOA simulations and empirical results.

5. Digital dynamical decoupling improves performance for QAOA only for two rounds and cubic Ising problems, whereas it actually leads to poorer performance in all other cases

In Section 2 the QAOA and QA hardware implementations are detailed. Section 3 details the experimental results and how the two algorithms compare. Section 4 concludes with what the results show. The figures in this article are generated using a combination of plotly [67], matplotlib [68, 69], networkx [70], and Qiskit [71] in Python 3. Code, data, and additional figures are available in a public Github repository [1].

2 Methods

The problem instances are defined in Section 2.1. In Section 2.2 the QAOA circuit algorithm and hardware parameters are defined. In Section 2.3 the quantum annealing implementation is defined. Section 2.4 defines the classical solvers that are used; CPLEX and simulated annealing (SA).

Device name	Topology/chip name	Available qubits	Available couplers/ CNOTs	Computation type
`Advantage_system4.1`	Pegasus P_{16}	5627	40279	QA
`Advantage_system6.1`	Pegasus P_{16}	5616	40135	QA
`ibm_washington`	Eagle r1 heavy-hexagonal	127	142	Universal gate-model

Table 1: NISQ processor summary. The hardware yield (e.g., the number of available qubits, couplers, or two qubit gates) of these devices can be less than the logical lattice because of hardware defects, and can also change over time if device calibration changes.

2.1 Ising Model Problem Instances

Table 1 shows that the D-Wave quantum annealers have more available qubits than `ibm_washington`. The additional qubits available on the quantum annealers will allow us to embed multiple problem instances in parallel onto the chips. The current IBMQ devices have a hardware graph referred to as a heavy-hexagonal lattice [72]. The current D-Wave quantum annealers have three different families of hardware graphs - *Chimera* [73–75], *Pegasus* [73, 76], and *Zephyr* [2]. For this direct comparison we target the Pegasus hardware graph devices. The two current D-Wave quantum annealers with Pegasus hardware graphs have chip id names `Advantage_system6.1` and `Advantage_system4.1`. Among the current quantum annealing hardware that is available, the Zephyr Z_4 and Chimera C_{16} hardware graph devices are either not large enough or dense enough in order to instantiate the problem embeddings, whereas the P_{16} Pegasus chip devices are large enough and dense enough for the problem instances to be embedded directly. Therefore, for a direct QAOA and QA comparison we create Ising problems with connectivities that map directly onto both the logical heavy-hexagonal lattice of the IBMQ device for the QAOA circuits as well as onto the P_{16} Pegasus topology of the quantum annealers. For a fair comparison, we need to define problems that can be instantiated on *all* three of the devices in Table 1. In particular, we want these implementations to not be unfairly costly in terms of implementation overhead. For example, we do not want to introduce unnecessary qubit swapping in the QAOA circuit because that would introduce larger circuit depths, which would introduce more decoherence in the computation. We also want to avoid minor embedding the problems onto the quantum annealers because there are an additional set of problems introduced with minor embedding, including chain break resolution algorithms [77] and the large ferromagnetic chain strength dominating the programmed energy scale on the chip [78].

As an additional dimension to push boundaries of the state-of-the-art in quantum optimization, we introduce *higher-order terms*, specifically cubic ZZZ interactions [79, 80], which could be viewed as 3-body interacting terms, or hypergraphs. The introduction of higher order terms offers a way to increase the complexity of the problems, because more terms need to be addressed by the solver, while keeping the total number of variables the same. The introduction of higher order terms into the Ising models we wish to sample for the direct QAOA and QA

[1] `https://github.com/lanl/QAOA_vs_QA`

[2] `https://docs.ocean.dwavesys.com/en/stable/docs_dnx/reference/generated/dwave_networkx.zephyr_graph.html`

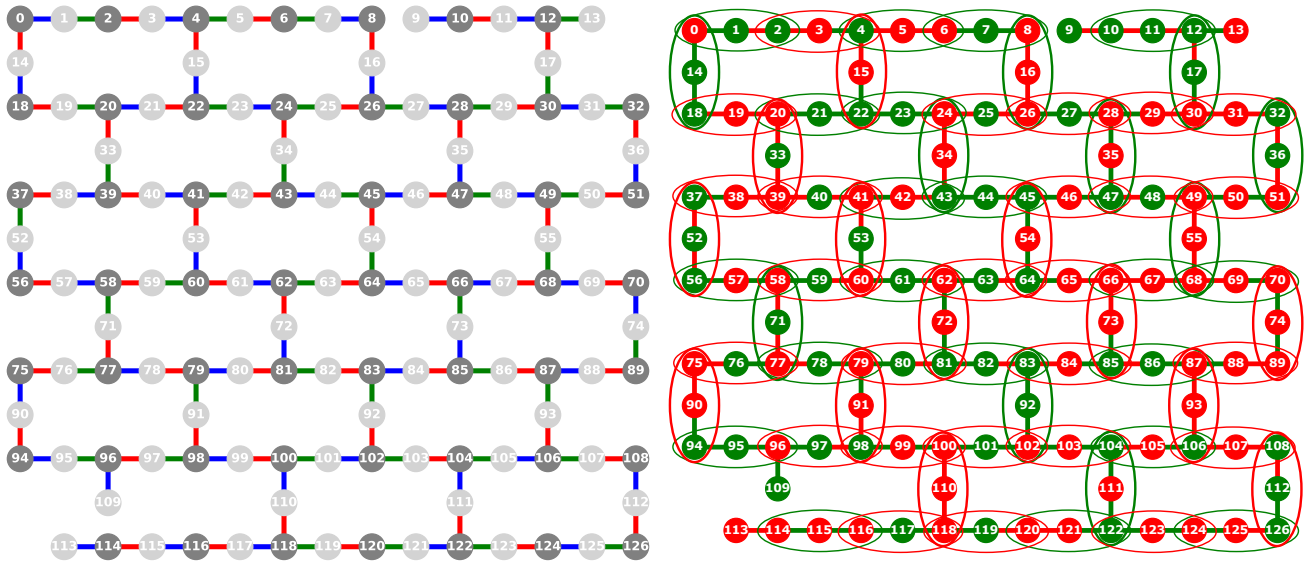

Figure 1: **Left:** `ibm_washington` graph connectivity, where qubits are connected by CNOT (also referred to as `cx`) gates. There are two missing graph edges from the `ibm_washington` lattice, with respect to the logical lattice, between qubits 8-9 and 109-114. The total number of qubits (nodes) is 127. The edges of the graph are three colored (red, blue, and green) such that no node shares two or more edges with the same color. The node colorings of light and dark gray show that the heavy hexagonal lattice is bipartite (meaning it can be partitioned into two disjoint sets). The three edge coloring is consistent with the QAOA circuit construction in Figure 2. **Right:** Example of a single random *cubic* problem instance (see Eq. (1)) on the `ibm_washington` graph. The linear and quadratic terms are shown using two distinct colors (red and green). The nodes and edges colored red denote a weight of -1 and the nodes and edges colored green denote a weight of $+1$. The cubic terms are represented by ovals around the three qubits which define the cubic variable interactions. Like the linear and quadratic terms, the color of the oval representing the cubic terms represents the sign of the weight on the terms, where green is $+1$ and red is -1. The *quadratic* problem instances (Eq. (2)) would be defined only by the randomly weighted nodes and edges, with no cubic terms. All *cubic* problem instances contain exactly 127 linear terms, 142 quadratic terms, and 69 cubic terms. All *quadratic* problem instances contain exactly 127 linear terms, 142 quadratic terms.

comparison requires both QAOA and QA to handle these higher order variable interactions, which is an additional test on the capability of both algorithms. Importantly, QAOA can naturally handle higher-order terms, which is a notable feature of the algorithm that has only been explored in a few previous studies [81–85]. Because D-Wave quantum annealing hardware only natively supports Ising models with linear and quadratic terms, implementing higher-order terms requires introducing auxiliary variables in order to perform order reduction to get a problem structure that is comprised of only linear and quadratic terms, that match the hardware graph, but whose optimal variable assignments (not including the auxiliary variables) are exactly the same as the optimal variable assignments of the original high order polynomial [8, 50, 86–89]. Note that the term HUBO (Higher-order Unconstrained Binary Optimization) problem [83, 88–90] is also used when referring to these types of combinatorial optimization problems which contain higher order terms, however HUBO is specifically for problems where the variable type is binary, and the Ising models we defined for this research have variables which are spins (e.g. $+1$, -1).

Taking each of these characteristics into account, we create a class of random problems that respect the native device connectivities in Table 1. The problem instances we will be considering are Ising problems defined on the hardware connectivity graph of the heavy hexagonal lattice of the device, which for these experiments will be `ibm_washington`. For a vector $z = (z_0, \ldots, z_{n-1}) \in \{+1, -1\}^n$ we define

$$C_1(z) = \sum_{v \in V} d_v \cdot z_v + \sum_{(i,j) \in E} d_{i,j} \cdot z_i \cdot z_j + \sum_{l \in W} d_{l,n_1(l),n_2(l)} \cdot z_l \cdot z_{n_1(l)} \cdot z_{n_2(l)} \tag{1}$$

Eq. (1) defines the class of random Ising models with cubic terms as follows. Any heavy hexagonal lattice is a bipartite graph with vertices $V = \{0, \ldots, n-1\}$ partitioned as $V = V_2 \cup V_3$, where V_3 consists of vertices with a maximum degree of 3, and V_2 consists of vertices with a maximum degree of 2. $E \subset V_2 \times V_3$ is the edge set representing available two qubit gates (in this case CNOTs where we choose targets $i \in V_2$ and controls $j \in V_3$).

W is the set of vertices in V_2 that all have degree exactly equal to 2. n_1 is a function that gives the qubit (variable) index of the first of the two neighbors of a degree-2 node and n_2 provides the qubit (variable) index of the second of the two neighbors of any degree-2 node. Thus d_v, $d_{i,j}$, and $d_{l,n_1(l),n_2(l)}$ are all coefficients representing the random selection of the linear, quadratic, and cubic coefficients, respectively. These coefficients could be drawn from any distribution - in this paper we draw the coefficients from $\{+1, -1\}$ with probability 0.5. Combined, any vector of variable states z can be evaluated given this objective function formulation of Eq. (1).

$$C_2(z) = \sum_{v \in V} d_v \cdot z_v + \sum_{(i,j) \in E} d_{i,j} \cdot z_i \cdot z_j \tag{2}$$

Eq. (2) defines the class of Ising problem that are the same as Eq. (1), except without any cubic terms; in particular this class of problems are easier since the problem is comprised of only linear and quadratic terms. Both of the Ising models defined by Eq. (2) and Eq. (1) are intended to be *minimization* combinatorial optimization problems, meaning that the goal is to find the variable assignment vector z such that the objective function evaluation is minimized. In particular, the aim is to find the global optimal solution of these Ising models.

The heavy hexagonal topology of `ibm_washington`, along with an overlay showing one of the random problem instances defined on `ibm_washington`, is shown in Figure 1. The edge set E shows a 3-edge-coloring due to Kőnig's line coloring theorem, which we will make use of in the QAOA Section 2.3.

Each term coefficient is sampled from $\{+1, -1\}$ in order to make the problem definitions very clear, but also make the problems still reasonably difficult to sample. The other advantage with choosing only two coefficients is that it mitigate the potential problem of limited precision for the programming control on all of the NISQ devices. Because all of these Ising models are random spin glasses, although fitting a very specific connectivity structure, it is expected that there will be a small number of degenerate ground states. However, we do not specifically compute all of the degenerate ground states, or even how many degenerate ground states exist. We leave further inquiries on the properties of the sampling of degenerate ground states for these specific Ising models, such as how fairly the ground states are sampled [30, 91, 92], to future research.

For the remainder of the article, the problem instances defined in Eq. (1) will be referred to as the *cubic* class of problems, meaning that the highest degree variable terms present in the problem are ZZZ terms. The problem instances defined in Eq. (2) will be referred to as the *quadratic* class of problems, meaning that the highest degree variable terms present in the model are two variable terms. For all experiments, we randomly generate 10 *cubic* Ising models and 10 *quadratic* Ising models, in order to provide a small ensemble of different problems to compare against each other, and importantly discern if there are significant differences that occur between different random instances of the same problem class.

2.2 QAOA

2.2.1 QAOA General Overview

For a combinatorial optimization problem over inputs $z \in \{+1, -1\}^n$, let $C(z)$: $\{+1, -1\}^n \to \mathbb{R}$ be the objective function from Equations (1), (2). For a minimization problem, the goal is to find a variable assignment vector z for which $C(z)$ is minimized. The QAOA algorithm consists of the following components:

- an initial state $|\psi\rangle$,
- a `phase separating` Cost Hamiltonian H_C,
  which is derived from $C(z)$ by replacing all spin variables z_i by Pauli-Z operators σ_i^z
- a `mixing` Hamiltonian H_M; in our case, we use the standard transverse field mixer, which is the sum of the Pauli-X operators σ_i^x
- an integer $p \geq 1$, the number of rounds to run the algorithm,
- two real vectors $\vec{\gamma} = (\gamma_1, ..., \gamma_p)$ and $\vec{\beta} = (\beta_1, ..., \beta_p)$, each with length p.

The QAOA algorithm consists of first preparing the initial state $|\psi\rangle$, and then applying p rounds of the alternating simulation of the phase separating Hamiltonian and the mixing Hamiltonian:

$$|\vec{\gamma}, \vec{\beta}\rangle = \underbrace{e^{-i\beta_p H_M} e^{-i\gamma_p H_C}}_{\text{round } p} \cdots \underbrace{e^{-i\beta_1 H_M} e^{-i\gamma_1 H_C}}_{\text{round 1}} |\psi\rangle \tag{3}$$

Within reach round, H_C is applied first, which separates the basis states of the state vector by phases $e^{-i\gamma f(x)}$. H_M then provides parameterized interference between solutions of different cost values. After p rounds, the state $|\vec{\gamma},\vec{\beta}\rangle$ is measured in the computational basis and returns a sample solution y of cost value $f(y)$ with probability $|\langle y|\vec{\gamma},\vec{\beta}\rangle|^2$.

The goal when using QAOA is to prepare the state $|\vec{\gamma},\vec{\beta}\rangle$ from which we can sample a variable assignment vector y with high cost value $f(y)$. Therefore, in order to use QAOA, the task is to find angles $\vec{\gamma}$ and $\vec{\beta}$ such that the expectation value $\langle\vec{\gamma},\vec{\beta}|H_C|\vec{\gamma},\vec{\beta}\rangle$ is large ($-H_C$ for minimization problems). In the limit $p \to \infty$, QAOA is effectively a Trotterization of of the Quantum Adiabatic Algorithm, and in general as we increase p we expect to see a corresponding increase in the probability of sampling the optimal solution [26]. The challenge is the classical outer loop component of finding good angles (not necessarily optimal) $\vec{\gamma}$ and $\vec{\beta}$ for all rounds p, which has a high computational cost as p increases.

There are a growing number of QAOA variants including GM-QAOA [93], ST-QAOA [94], CD-QAOA [95], Th-QAOA [96], RQAOA [97], warm start QAOA [98–101], FUNC-QAOA [102], FQAOA [103], FKL-QAOA [104], HLZ-QAOA [104], DC-QAOA [105], and multi-angle QAOA [106, 107], but here we do not use any of these QAOA variants instead we use the standard transverse field mixer implementation, which makes the comparison consistent with the quantum annealing devices.

Variational quantum algorithms, including QAOA, have been a subject of interest in quantum algorithms research in large part because of the problem domains that variational algorithms can address (such as combinatorial optimization) [108]. The primary challenge in variational quantum algorithms is the classical component of parameter selection which has not been solved and is even more difficult when noise is present in the computation [109]. Typically, for small scale experiments the optimal angles for QAOA are computed exactly for small problem instances [30, 110]. Our angle finding approach consists of a reasonably high-resolution gridsearch without knowing good angles a-priori. We note that a fine gridsearch scales exponentially with the number of QAOA rounds p, and is therefore not practical for higher round QAOAs [2, 4]. Classically computing good angles is computationally intensive, especially with the introduction of cubic terms. In our specific case of the class of sparse Ising models defined in Section 2.1, it is possible due to the relative locality of the problem instances to classically simulate the mean energy expectation values computed by QAOA (at least for $p = 1$ and $p = 2$), albeit at significant computational cost; Section 2.2.5 details how this can be accomplished.

2.2.2 Implementing Ising model QAOA Circuits on IBMQ Heavy-Hex Topology

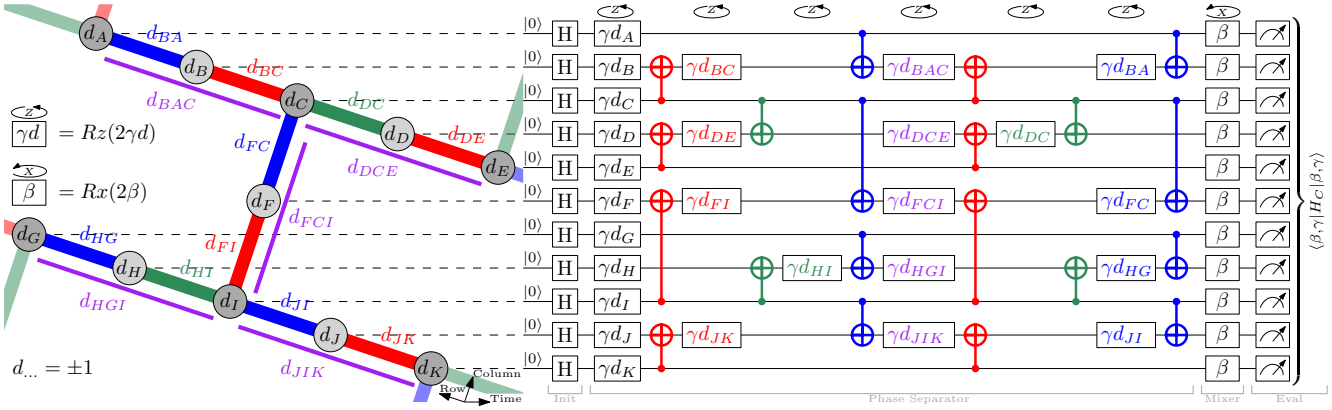

Figure 2: A 1-round QAOA circuit: **(left)** The problem instance is a hardware-native bipartite graph with an arbitrary 3-edge-coloring given by Kőnig's line coloring theorem. Figure 1 (left) shows a 3-edge-coloring and bipartite shading consistent with this figure. The purple lines denote the cubic terms. **(right)** Any quadratic term (colored edge) gives rise to a combination of two CNOTs and a Rz-rotation in the phase separator, giving a CNOT depth of 6 due to the degree-3 nodes. When targeting the degree-2 nodes with the CNOT gates, these constructions can be nested, leading to no overhead when implementing the three-qubit terms: these always have a degree-2 node in the middle (see Eq. (1)). For the problem instances without cubic terms, the QAOA circuit construction can simply disregard the purple single qubit rotations.

Figure 2 describes the short depth QAOA circuit construction for sampling a higher order Ising problem instance, described in Section 2.1. This algorithm can be applied to any heavy hexagonal lattice topology, which allows for

executing the QAOA circuits on the 127 variable instances on the IBMQ `ibm_washington` backend (or more generally, any heavy-hex hardware graph quantum computer. This particular QAOA circuit could be viewed as a type of Hardware Efficient Ansatz (HEA) [111–113], but is applied to QAOA, not general Variational Quantum Algorithms (VQA's), and is restricted to a very particular form of random Ising model (defined in Section 2.1) that itself is hardware specific.

For considering the angle parameter searchspace, note that each *quadratic* and each *cubic* problem instance has the property that all its samples will have *the same parity* as each term contributes either $+1$ or -1. Therefore, for any objective value C (i.e. either $C_1(x)$ or $C_2(x)$), we have $C = 2k + \texttt{parity}$ and thus

$$\begin{aligned} e^{-i(\gamma+\pi)C} &= e^{-i\gamma C} \cdot e^{-i\pi(2k+\texttt{parity})} \\ &= e^{-i\gamma C} \cdot \underbrace{e^{-i2\pi k}}_{1} \cdot \underbrace{e^{-i\pi\cdot\texttt{parity}}}_{\texttt{global phase}} \\ &\cong e^{-i\gamma C}. \end{aligned}$$

Therefore, increasing the angle γ by π only adds a global phase, and hence γ has a periodicity of π which is why the angle range of $[0, \pi)$ was chosen to perform the grid-search within (although for example $[\frac{-\pi}{2}, \frac{\pi}{2})$ would also work). The same reasoning applies to β. Furthermore, there is an additional symmetry that can be used: Applying the two angle sequences of

1. $\beta = (\beta_0, \beta_1, \ldots, \beta_p)$, and $\gamma = (\gamma_0, \gamma_1, \ldots, \gamma_p)$
2. $-\beta = (-\beta_0, -\beta_1, \ldots, -\beta_p)$, and $-\gamma = (-\gamma_0, -\gamma_1, \ldots, -\gamma_p)$

give the same *expectation values* because they are mirror symmetric. This means that the search space can be cut in half; in this case we chose to divide the last β_p angle in half in order to take advantage of this mirror symmetry and reduce the angle search space. Therefore, for both quadratic and cubic problem instances, the QAOA angle ranges used are $\gamma_1, \ldots, \gamma_p \in [0, \pi)$ and $\beta_1, \ldots, \beta_{p-1} \in [0, \pi), \beta_p \in [0, \frac{\pi}{2})$ where p is the number of QAOA rounds. The halving of the angle search space for β applies when $p = 1$.

For optimizing the angles using the naive grid search for $p = 1$, β_0 is varied over 60 linearly spaced angles $\in [0, \frac{\pi}{2}]$ and γ_0 is varied over 120 linearly spaced angles $\in [0, \pi]$. For a comparable resolution gridsearch for $p = 2$, β_1 is varied over 5 linearly spaced angles $\in [0, \frac{\pi}{2}]$ and γ_0, γ_1, and β_0 are varied over 11 linearly spaced angles $\in [0, \pi]$. Therefore, for $p = 2$ the angle gridsearch uses 6655 separate circuit executions (for each of the 10 problem instances), and for $p = 1$ the angle gridsearch uses 7200 separate circuit executions. Each circuit execution measures $10,000$ samples in order to obtain a highly robust distribution for each angle combination.

2.2.3 Dynamical Decoupling for QAOA Circuits

With the goal of mitigating decoherence on idle qubits, digital dynamical decoupling (DDD) is tested on all QAOA circuits, for both $p = 1$ and $p = 2$. Dynamical Decoupling is an open loop quantum control technique error suppression technique for mitigating decoherence on idle qubits [114–119]. Dynamical decoupling can be implemented with pulse level quantum control, and digital dynamical decoupling can be implemented with circuit level gate instructions comprised of sequences of identity gates [118]. Digital dynamical decoupling is an approximation of pulse level dynamical decoupling. Dynamical decoupling has been experimentally demonstrated to be useful in certain computations for superconducting qubit quantum processors including IBMQ devices [120–125]. Dynamical decoupling in particular could applicable for high round QAOA circuits because the circuits can be relatively sparse and therefore have idle qubits [120]. Dynamical decoupling is not always effective at consistently reducing errors during computation (for example because of other control errors present on the device [117, 120, 125]), and therefore the raw QAOA circuits are compared against the QAOA circuits with DDD in Section 3. In order to apply the DDD sequences to the OpenQASM [126] QAOA circuits, the `PadDynamicalDecoupling` [3] method from Qiskit [71] is used, with the `pulse_alignment` parameter specified based on the `ibm_washington` backend properties. The circuit scheduling algorithm that is used for inserting the digital dynamical decoupling sequences is ALAP, which schedules the stop time of instructions as late as possible [4]. There are other scheduling algorithms that could be applied which may perform better. There are different DDD gate sequences that can be applied, including Pauli `Y, Y` or Pauli `X, X` gate sequences. Because the X Pauli gate is already a native gate of `ibm_washington`, the two Pauli X gate `X, X` DDD sequence is used for simplicity. The X-X sequence is expected to typically mitigate time correlated dephasing noise [5].

[3] `https://qiskit.org/documentation/locale/bn_BN/stubs/qiskit.transpiler.passes.PadDynamicalDecoupling.html`
[4] `https://qiskit.org/documentation/apidoc/transpiler_passes.html`
[5] `https://mitiq.readthedocs.io/en/latest/guide/ddd-5-theory.html`

Because of the magnitude error rates on current NISQ hardware, utilizing as many error mitigation strategies as possible would be ideal for these experiments. Unfortunately, other error mitigation strategies are either not scalable to large system sizes, such as measurement error mitigation requiring an exponential number of circuit executions [127], or are intended to provide error mitigated expectation values and therefore do not provide variable assignments, such as Zero Noise Extrapolation (ZNE) [118, 128–130]. For this direct comparison of QA and QAOA, we aim to obtain the variable assignments for the solutions to the combinatorial optimization problems, and therefore do not utilize quantum error mitigation algorithms that use classical post processing.

2.2.4 IBMQ Circuit Execution

The variable states for the optimization problems are either $\{+1, -1\}$, but the IBMQ circuit measurement states are either 0 or 1. Therefore once the measurements are made on the QAOA circuits, for each variable in each sample the variable states are mapped as follows: $0 \mapsto 1$, $1 \mapsto -1$. When executing circuits on the superconducting transom qubit `ibm_washington`, circuits are batched into *jobs* where each job is composed of a group of at most 250 circuits. The maximum number of circuits for a job on `ibm_washington` is 300, but we use 250 circuits per job in order to reduce backend job errors related to the size of jobs. Grouping circuits into jobs reduces the total amount of compute time required to prepare and measure each circuit. When submitting the circuits to the backend, they are all first locally transpiled using Qiskit [71] with `optimization_level=3` and targetting the exact hardware graph used to define the Ising models (see Figure 1). This transpilation adapts the gateset to the `ibm_washington` native gateset, and the transpiler optimization attempts to simplify the circuit where possible. The QAOA circuit execution on `ibm_washington` spanned several months, and therefore the backend software versions were not consistent. The backend software versions of `ibm_washington` that were used for all of the QAOA experiments are: `1.3.7`, `1.3.8`, `1.3.13`, `1.3.14`, `1.3.15`, `1.3.17`, `1.3.19`, `1.3.22`, `1.4.0`, `1.5.1`, `1.5.2`, `1.5.3`, `1.5.4`, `1.5.5`, `1.6.0`.

Example Qiskit QAOA circuit drawings are given in Figures 16, and 17 in Appendix A. When these QAOA circuits are compiled, the total number of instructions used (not including delay gates) is approximately $3,000$ depending on the compilation and β, γ angles used.

2.2.5 Classical QAOA simulation

Because the problem instances are defined on a relatively sparse graph, and because we run only 1 and 2 rounds, it is possible to classically simulate the *mean* QAOA energy landscape (for an arbitrary set of angles β, γ) for both *quadratic* and *cubic* problem instances. This is an improvement over ref. [33] where no classical simulation was performed on the *cubic* Ising models, but this method as we propose it does not provide a mechanism to compute a distribution of expectation values (e.g. with some shot noise), instead it provides only a mechanism to compute the mean expectation value. The key observation is that we can simulate portions of the overall QAOA circuit applied to a subset of the problem instances in order to compute the *mean* expectation value for a single term (e.g. a linear, quadratic, or cubic term). By linearity of expectation, you can compute the cumulative sum of the mean energy for each of the terms in the problem instance. This provides a mechanism to compute the overall mean energy for a problem instance, given some angles β, γ. Note however that this only provides the mean expectation values for any given β, γ, but it does not provide an objective function distribution and it does not provide variable assignment solutions for the given optimization problem. Therefore, this simulation method will allow us to verify and compare against the NISQ experimental results. For highly entangled problems though, this would still be intractable to simulate. However, for the heavy hex lattice problems it is only required that those qubits be simulated that interact with the specific term in question. More rounds leads to more interactions for each term, and variable interactions are defined by both quadratic and cubic terms. Figure 3 shows the subgraph of the problem instances, for a single *cubic* problem instance, which interact with the specific cubic term (which, in this example, is formed by the qubits $23, 22, 24$) for $p = 2$ QAOA. This lightcone of interacting terms contains 27 qubits, which is possible to simulate using HPC resources. However, if $p = 3$ QAOA was used or if the problem instance had denser long range interactions, then the simulation would be considerably harder and potentially intractable. $p = 2$ rounds applied to a *cubic* problem instance is the hardest of the problems to simulate; $p = 1$ is easier to simulate, in that it uses fewer qubits, as are simulations of the *quadratic* problems.

The procedure is to enumerate over all of the terms, either in a *quadratic* or *cubic*, and compute the neighborhood of qubits that interact with that term (this is dependent on how many rounds are used), extract all terms (linear quadratic, and optionally cubic depending on the problem) that are strictly within this neighborhood of qubits, and then execute the QAOA circuit construction algorithm (see Figure 2) in order to create a QAOA circuit specifically for that neighborhood of terms. Next, execute that full QAOA circuit using many shots (in this case $10,000$, measure only the states of the qubits for that specific term. The expectation value for each of those shots is

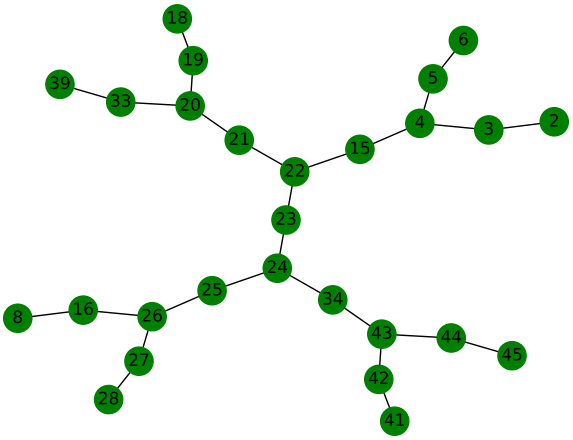

Figure 3: The largest heavy hex subgraph that is required to completely simulate a single cubic term (specifically qubits 23, 22, 24 on `ibm_washington`) for $p = 2$ QAOA. Since this subgraph contains 27 qubits (nodes), it is possible to directly classically simulate all of the linear, quadratic, and cubic term components for all of the problem instances by classically simulating the sub-circuit which contains all of the interacting terms in this 27 qubit "light-cone" (and not any of the other terms which are outside of this subgraphs).

computed, and the mean energy is recorded. This is then repeated for all other terms in the problem instance, thus giving a cumulative *mean* expectation value. Lastly, this entire procedure is iterated over for the discrete angle search space – specifically the exact same angle search space that is evaluated experimentally on `ibm_washington` and that is described in Section 2.2.

There are two optimizations to this method which we do not explore in this paper, but that could potentially be used for future improved simulations:

1. There are classical simulation methods, such as stabilizer decompositions [131], or tensor networks, which could be used to approximately simulate the full entanglement lightcone of these Ising models for rounds greater than 2.

2. The lightcone of entanglement for individual terms in these Ising models does not include all of the gate operations performed in the QAOA circuit subgraph, and in particular many gate operations within each Ising term entanglement lightcone could be removed in order to speed up classical computations of each sub-circuit.

2.3 Solving with Quantum Annealing (QA)

2.3.1 QA General Overview

Quantum annealing uses quantum fluctuations, such as quantum tunneling, in order to search for the ground state of a user programmed Hamiltonian. Quantum annealing, in the case of the transverse field Ising model as implemented on D-Wave hardware, is explicitly described by the system given in Eq. (4) below. The state begins at time zero purely in the transverse Hamiltonian state $\sum_i \sigma_i^x$, and then over the course of the anneal (parameterized by the *annealing time*) the user programmed Ising problem is applied according the function $B(s)$. Together, $A(s)$ and $B(s)$ define the anneal schedules of the annealing process, and s is referred to as the *anneal fraction*. The standard anneal schedule that is used is a linear interpolation between $s = 0$ and $s = 1$.

$$H = -\frac{A(s)}{2}\Big(\sum_i^n \sigma_i^x\Big) + \frac{B(s)}{2}\Big(H_{ising}\Big) \tag{4}$$

The adiabatic theorem states that if changes to the Hamiltonian of the system are sufficiently slow, then the system will remain in the ground state of problem Hamiltonian (up until the state is classical measured), thereby providing a computational mechanism for finding the ground state of combinatorial optimization problems. The user programmed Ising problem H_{ising}, acting on n qubits, is defined in Eq. (5). Combined, the quadratic terms

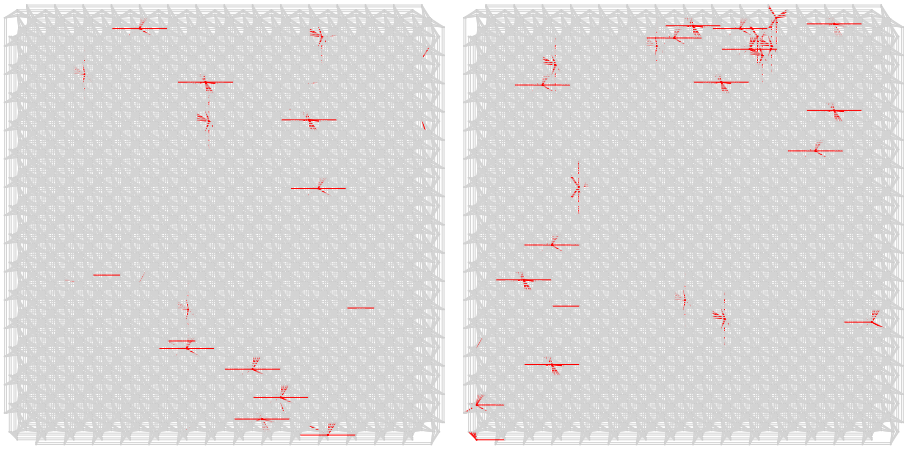

Figure 4: Pegasus graph (P_{16}) renderings with the missing hardware drawn in red nodes and edges for the `Advantage_system4.1` D-Wave chip (left) and the `Advantage_system6.1` D-Wave chip (right).

and the linear terms define the optimization problem instance that quantum annealing samples. As with QAOA, the objective of quantum annealing is to find the variable assignment vector z that minimizes the cost function that has the form of Eq. (5).

$$H_{ising} = \sum_{i}^{n} h_i \sigma_i^z + \sum_{i<j}^{n} J_{ij} \sigma_i^z \sigma_j^z \tag{5}$$

2.3.2 Implementing Ising Problems on D-Wave's Pegasus Topology

In order to execute the Ising models of Eq. (1) and Eq. (2) on D-Wave quantum annealers, the primary challenge is that the higher-order (i.e., cubic) terms will need to have order reduction techniques applied to them in order to transform the cubic terms into linear and quadratic terms [8, 50, 86–88]. This order reduction will result in using additional variables, usually called *auxiliary* or *slack* variables, in addition to additional edges (quadratic terms) that allow those new auxiliary variables and the existing discrete optimization problem variables to interact. Figure 5 shows the mapping of the Ising models onto a logical Pegasus P_{16} graph, in particular showing the order reduction procedure which makes use of two alternating order reductions using auxiliary variables with different connectivities. The alternating order reduction's are required in order to fit an entire problem instance Ising model (defined on a heavy hex graph) with cubic terms onto the D-Wave hardware graph(s). The order reduction procedure outlined in Figure 5 allows for direct embedding of the order reduced polynomials onto the hardware graph, regardless of whether the cubic term coefficient is $+1$ or -1. This order reduction ensures that the ground state(s) of the cubic term are also the ground states of the order reduced Ising model. Additionally, this order reduction ensures that for every excited state of the cubic term, there are no slack variable assignments which result in the original variables having an energy less than or equal to the ground state of the original cubic term. This order reduction procedure allows any problem in the form of Eq. (1) to be mapped natively to quantum annealing hardware which accepts problems with the form of Eq. (5). Importantly, this procedure does not require minor-embedding (e.g. use of chains of physical qubits to represent a logical variable), even including the auxiliary variables. Constructing the embeddings of the order reduced higher order terms onto Pegasus requires alternating between two valid cubic reductions (both of which are shown in Figure 5); we found this was not possible to do using only a single cubic order reduction formulation.

In order to mitigate local biases and errors that may occur on the QA chip, and in order to get more problem samples for the same QPU time, the other strategy that is employed is to embed multiple independent problem instances onto the hardware graph and thus be able to execute several instances in the same annealing cycle(s). This technique is referred to as *parallel quantum annealing* [88, 132, 133] or *tiling* [6]. Note that the concept of *parallel quantum annealing* has analogous ideas in the circuit model quantum computing paradigm, which are generally referred to as *multi-programming* [134–137], or *parallel circuit execution* [138, 139], or *circuit concurrency* [140]. Figure 5 (right) shows the parallel embeddings on a logical Pegasus graph. Because some of the logical embeddings

[6] `https://dwave-systemdocs.readthedocs.io/en/samplers/reference/composites/tiling.html`

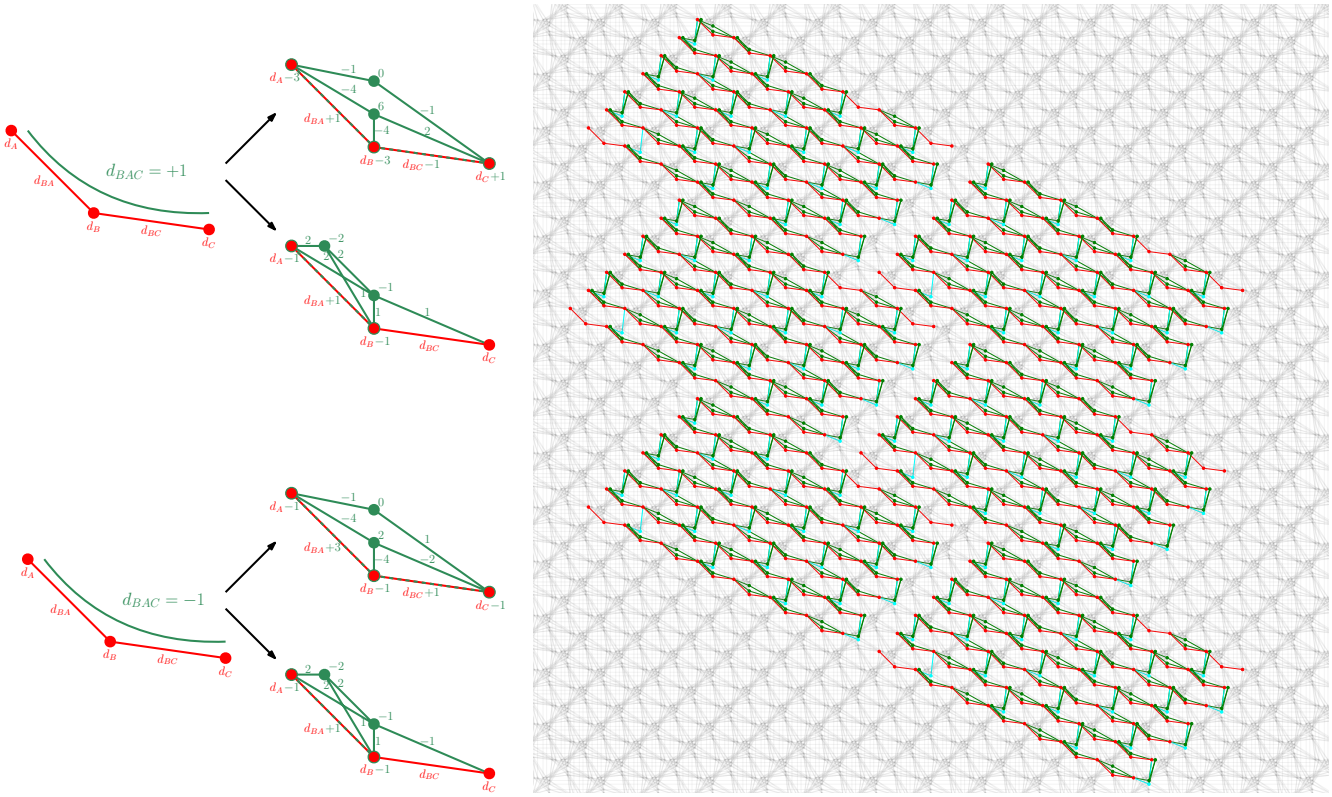

Figure 5: **(left)** Two different embeddings for cubic $+1/-1$ terms onto D-Wave's Pegasus topology. Each embedding needs two slack variable qubits, shown as green circles with the numbers indicating the weights d of the slack variables and their connections as well as changes to the original weights of the variables involved in the cubic term. Our overall embedding alternates between these two cubic term embeddings. Any embedding with only one slack variable needs a 4-clique between the slack and the three original variables, which is not possible to embed for consecutive cubic terms. **(right)** Embedding structures of the *cubic* problem instances embedded in parallel (independently) 6 times onto the logical Pegasus P_{16} graph. The view of this graph has been slightly partitioned so that not all of the outer parts of the Pegasus chip are drawn. The light grey qubits and couplers indicate unused hardware regions. The cyan coloring on nodes and edges denote the vertical qubits and CNOTs on the `ibm_washington` hardware graph (see Figure 1). The red coloring on nodes and edges denote the horizontal lines of qubits and CNOTs on `ibm_washington`. The green nodes and edges denote the order reduction auxiliary variables. Note that in problem tiling on Pegasus, the top right hand and lower left hand qubits are not present on the `ibm_washington` lattice, but for the purposes of generating the embeddings these extra qubits are filled in to complete the heavy hex lattice embedding. The embedding structure for *quadratic* problems is identical to this embedding, with the exception that there are no auxiliary (green) variables.

may use a qubit or coupler that is missing on the actual hardware, less than 6 parallel instances can be tiled onto the chips to be executed at the same time. For `Advantage_system4.1`, 2 independent embeddings of the *cubic* problem instances could be created without encountering missing hardware and and 3 independent embeddings of the *quadratic* problem instances could be created. For `Advantage_system6.1`, 3 independent embeddings of the *cubic* problem instances could be created and 5 independent embeddings of the *quadratic* problem instances could be created. The structure of the heavy-hex lattice onto Pegasus can be visually seen in Figure 5; the horizontal heavy-hex lines (Figure 1) are mapped to diagonal Pegasus qubit lines that run from top left to bottom right of the square Pegasus graph rendering. Then the vertical heavy-hexagonal qubits are mapped to QA qubits in between the diagonal qubit lines.

2.3.3 D-Wave Parameter Optimizations

In order to optimize the quantum annealing parameters, similar to the QAOA angle gridsearch, a gridsearch over forward anneal schedules with pauses is performed. Figure 6 visually shows all of forward annealing schedules

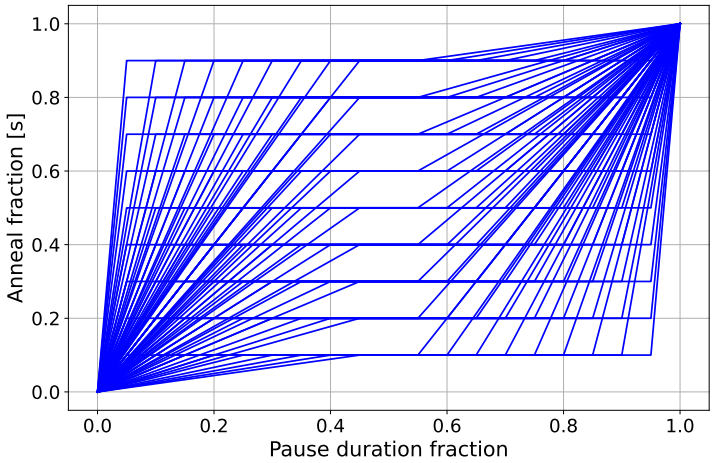

Figure 6: All modified (forward) quantum annealing schedules tested in order to find the best anneal schedule with a pause. The symmetric pause inserted into the normal linearly interpolated schedule defining the $A(s)$ and $B(s)$ functions can provide better ground state sampling probability. The anneal fraction at which this pause occurs is varied between 0.1 and 0.9 in steps of 0.1. The pause duration, as a fraction of the total annealing time, is also varied between 0.1 and 0.9 in steps of 0.1. Although not shown in this figure, the annealing times are also varied between 10, 100, 1000, and 2000 microseconds. Although not shown, linearly interpolated anneal schedules are also executed on both QA devices and all problem instances using an annealing time of $0.5, 1, 10, 20, 100, 200, 1000, 2000$ microseconds.

with pauses that are used for this schedule optimization. Pausing the anneal at the appropriate spot can provide higher chances of sampling the ground state [141–144]. Importantly the annealing times used in these schedule are also optimized for, but are not shown in the figure since the schedules can be scaled to different annealing times. The total number of QA parameters that are varied are 9 anneal fractions, 9 pause durations, and 4 different annealing times (10, 100, 1000, 2000 microseconds). Therefore, the total number of parameter combinations that are considered in the grid search is 324. 2000 microseconds is the longest annealing time available on the current D-Wave quantum annealers. The number of anneals sampled for each D-Wave job was either 500 a series of 250 anneal jobs (this was dependent on the maximum total QPU time that could be used per job). The annealing times and the anneal schedules were varied in a simple grid search. Readout and programming thermalization times are both set to 0 microseconds. All other parameters are set to default, with the exception of the modified annealing schedule.

2.4 Simulated Annealing and CPLEX implementations

In order to provide a reasonable basis of comparison, the 10 cubic instances and 10 quadratic instances are also solved exactly using CPLEX [145], and sampled using simulated annealing. Knowing the exact, optimum solutions tells us whether either QAOA or QA were able to find the optimal solutions. Simulated annealing is a well-known general purpose classical heuristic [146] that provides a good comparison point for heuristic quantum algorithms [20]. The simulated annealing implementation we use is the D-Wave systems SDK package [7], using all default settings and generating 1000 samples per problem instance.

Applying simulated annealing to the *quadratic* problem instances is straighforward since there are only linear and quadratic terms. The simulated annealing implementation we used does not natively handle higher order terms though, and therefore the *cubic* problem instances could not sampled natively. Instead, the *cubic* instances required an order reduction scheme - similar to the quantum annealing implementation (Section 2.3). Here for the order reduction, since the connectivity is not highly constrained we simply used an order reduction method available in the D-Wave systems SDK package *dimod* [8]. This method is called `make_quadratic`, which can take any higher spin or binary polynomial and transform it into linear and quadratic terms, at the cost of introducing additional

[7] `https://github.com/dwavesystems/dwave-neal`
[8] `https://github.com/dwavesystems/dimod`

auxiliary variables [9]. This method requires the specification of a `strength` term that defines the cost function penalty for breaking a product constraint in the order reduced Ising model. If the order reduction `strength` is too small, the ground states of the order reduced term problem will not match the ground states of the original problem. Typically, using a `strength` that is equal to the maximum of the absolute values of all problem coefficients produces order reduced problems that match the ground state of the original problem [88]. Because the maximum absolute value of all coefficients is 1, a penalty `strength` of 2 is used in order to construct order reduced Ising models in order to run simulated annealing on the *cubic* problem instances.

The CPLEX solver was used by converting the spin variables into binary variables so that it could be solved as a Mixed Integer Quadratic Programming (MIQP) problem. Adapting the linear terms h and quadratic terms J into this binary form is possible, the exact formulation is given in Eq. (6) where all x_i variables are binary. This is sufficient for adapting the *quadratic* problem instances to be solved exactly using CPLEX. However, for the *cubic* problem instances, CPLEX does not support higher order variable terms. Therefore, the same dimod order reduction method that was used for simulated annealing is also applied for all *cubic* problem instances. Eq. (6) allows CPLEX to evaluate the objective function of Eq. (2) (or Eq. (1) after order reduction) by specifying the variables in the domain of $\{0, 1\}$, but the objective function computation is evaluated correctly because the variables are first transformed into spins $\{+1, -1\}$.

$$C_{bin}(x) = \sum_{i \in h}(1 - 2x_i) + \sum_{(i,j) \in J}(1 - 2x_i) \cdot (1 - 2x_j) \tag{6}$$

After CPLEX has found the optimal variable assignment in terms of $\{0, 1\}$ variables, the variable states are mapped $1 \mapsto 1$ and $0 \mapsto -1$. This mapping then allows the original objective functions (Eq. (2) or Eq. (1)) to be evaluated using the optimal spin variable assignments.

3 Results

In the following sub-sections the results are presented as such. First, Section 3.1 details how QAOA and QA compare when their best parameters are chosen, and how close they get to reaching the optimal solution. As a baseline comparison, the energy distribution histograms in Section 3.1 contain the objective functions of random samples - these are computed with binomial probability of 0.5 for selecting $+1$ or -1. Next, Section 3.2 shows results for tuning the QA schedules and annealing times. Section 3.3 shows the $p = 1$ and $p = 2$ QAOA angle searches. Section 3.4 shows the exact QAOA $p = 1$ energy landscapes.

A notable comparison point that is not highlighted in the followings plots, is the total amount of computation time required to execute these experiments. Approximately 120 minutes of QPU access time was used for all D-Wave quantum annealing experiments. Approximately 16500 minutes of QPU time was used for all QAOA experiments. These estimates do not include queue times. The QPU time consumed for QAOA is entirely due to the high resolution parameter gridsearch that is performed, which used more parameters than the QA experiments.

3.1 Comparing QAOA, Quantum Annealing (QA), Simulated Annealing (SA), and Random Samples

3.1.1 Comparing Cubic Problem Performance

Figure 7 plots 10 histograms, one for each of the 10 minimization *cubic* problem instances. Each histogram contains a distribution showing $10,000$ random samples, the best angle choices found (experimentally) for $p = 1$ and $p = 2$ QAOA with and without digital dynamical decoupling, the QA results from two D-Wave quantum annealers, 1000 samples from simulated annealing, and the mean expectation values for the best angle choices of QAOA computed exactly. We make the following eight observations from these plots:

1. Figure 7 shows that the random samples are clearly separated from the QAOA energy distributions - although there is overlap.

2. The QA distribution is clearly distinct from the QAOA distribution, and performs much better.

3. The experimental QAOA distributions are roughly half-way between the exact QAOA simulation (which was computed classically) and random samples.

[9] `https://docs.ocean.dwavesys.com/en/stable/docs_dimod/reference/generated/dimod.make_quadratic.html`

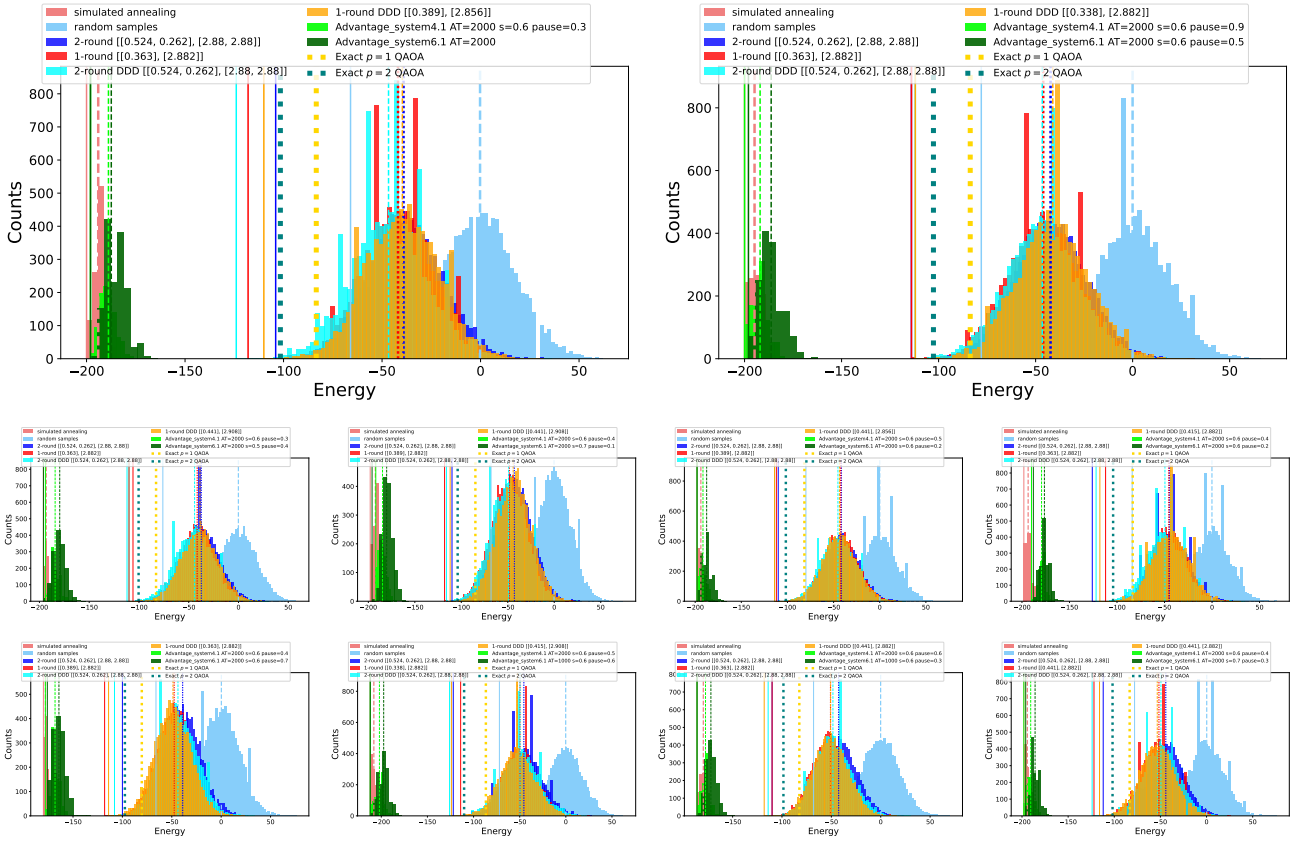

Figure 7: Direct energy histogram comparison of QA and QAOA results the ten minimization *cubic* problem instances. Here the energies being plotted are the full energy spectrum for the parameters that gave the minimum mean energy across the parameter grid searches performed across the QA and QAOA parameters. The optimal parameter combination is given in the figure legend. For QA parameters, the annealing time in microseconds, the forward anneal schedule (symmetric) pause fraction, and anneal fraction, are given in the legend. If the QA parameter in the legend is only an annealing time, that indicates that the optimal annealing schedule was the default linear interpolation for that specified annealing time. For the QAOA angle parameters, the format is $[\beta, \gamma]$, and are rounded to 3 decimal places. The mean for the energy distributions is marked with vertical dashed lines and the minimum energy found in each dataset is marked with solid vertical lines. If there are multiple parameter combinations that result in the same mean energy, one of those parameter combinations is chosen arbitrarily. The best mean energy found across the possible angle combinations from the exact (classical) QAOA simulations (described in Section 2.2.5) are marked with vertical thick dotted lines; those energy means show what QAOA could sample theoretically if there was no noise in the computation.

4. The four different QAOA implementations performed very similarly - their distributions have very high overlap and differences in the performance is marginal. In order to help show exactly which of the QAOA methods performed better across the 10 problems, Table 2 shows a confusion matrix type description of what methods had better mean energy distributions (for the best QAOA angle choice) compared to the other methods. In Table 2, we can see that digital dynamical decoupling improves the $p = 2$ QAOA computation noticeably.

5. The exact QAOA simulations show that a zero noise quantum computation of QAOA would not achieve the same mean energy found by the two quantum annealers. However, it could be the case that the lower energy tails of such a QAOA computation would be able to find the optimal solution or at least get close.

6. Although this will be observed in more detail in Section 3.3, the optimal angle choices (found experimentally) show parameter concentration - i.e. even though the 10 problems are different, the best QAOA angles across the problems are similar if not identical.

7. QA and simulated annealing are comparable but typically simulated annealing has a slightly better mean

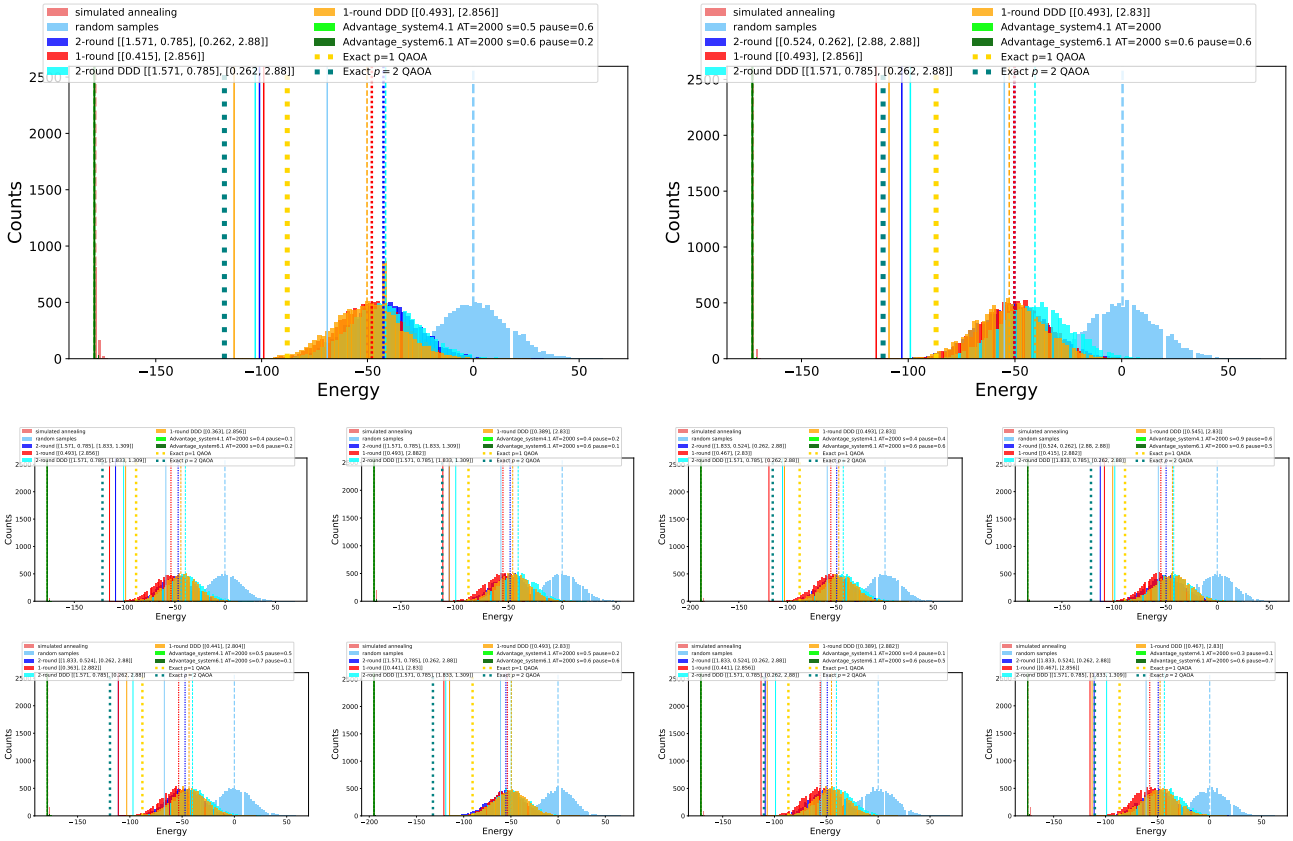

Figure 8: Direct energy histogram comparison of QA and QAOA results the ten minimization *quadratic* problem instances. Here the energies being plotted are the full energy spectrum for the parameters that gave the minimum mean energy across the parameter grid searches performed across the QA and QAOA parameters. The optimal parameter combination is given in the figure legend. For QA parameters, the annealing time in microseconds, the forward anneal schedule (symmetric) pause fraction, and anneal fraction, are given in the legend. For the QAOA angle parameters, the format is $[\beta, \gamma]$, and are rounded to 3 decimal places. The mean for each dataset is marked with vertical dashed lines and the minimum energy found in each dataset is marked with solid vertical lines. The best mean energy found across the possible angle combinations from the exact (classical) QAOA simulations (described in Section 2.2.5) are marked with vertical thick dotted lines; those energy means show what QAOA could sample theoretically if there was no noise in the computation.

energy.

8. `Advantage_system6.1` consistently has slightly worse mean energy than `Advantage_system4.1`.

3.1.2 Comparing Quadratic Problem Performance

Figure 8, similar to Figure 7, shows 10 histograms, but now for the 10 minimization *quadratic* problem instances that contain no higher order terms. The set of observations made about Figure 7 also apply to Figure 8 with a few exceptions. First, Table 3 shows the confusion matrix style representation of how the four QAOA methods compared against each other, but now for the *quadratic* problems. Table 3 shows that, as opposed to Table 2, digital dynamical decoupling does not help as much as it did for the *cubic* problems. This could be because the cubic problem instances utilized greater circuit depth, thus having more idle qubit time in the computation which dynamical decoupling can help to mitigate errors on. Second, these problems were considerably easier for both simulated annealing and quantum annealing to sample the optimal solution of. As a result, the energy distributions for both QA and SA are concentrated near the optimal solutions and there are not clear visual differences between the two distributions.

	$p = 1$	$p = 2$	$p = 1$ with DDD	$p = 2$ with DDD
$p = 1$ (no DDD) better than -	-	10/10	5/10	4/10
$p = 2$ (no DDD) better than -	0/10	-	2/10	0/10
$p = 1$ (with DDD) better than -	5/10	8/10	-	4/10
$p = 2$ (with DDD) better than -	6/10	10/10	6/10	-

Table 2: This table is a confusion matrix representation of how the four different QAOA implementations compare against each other when sampling the 10 *cubic* instances. Each cell is showing for how many of the 10 problem Ising models each method had a better mean objective value compared to the other three methods (given the best angle combination that was found in the massive angle gridsearch). This information is shown in the form of histograms in Figure 7, however there it is difficult to visually discern which methods. The order of performance of the four different QAOA implementations is as follows; $p = 2$ with no DDD performed the worse, $p = 1$ with DDD performed the next best, $p = 1$ with no DDD performed the next best, and $p = 2$ with DDD performed the best overall.

	$p = 1$	$p = 2$	$p = 1$ with DDD	$p = 2$ with DDD
$p = 1$ (no DDD) better than -	-	8/10	8/10	10/10
$p = 2$ (no DDD) better than -	2/10	-	8/10	10/10
$p = 1$ (with DDD) better than -	2/10	2/10	-	9/10
$p = 2$ (with DDD) better than -	0/10	0/10	1/10	-

Table 3: This table is a confusion matrix representation of how the four different QAOA implementations compare against each other when sampling the 10 *quadratic* instances. Each cell is showing for how many of the 10 Ising models each method had a better mean objective value compared to the other three methods (given the best angle combination that was found in the massive angle gridsearch). This information is shown in the form of histograms in Figure 8, however there it is difficult to visually discern which methods. The order of performance of the four different QAOA implementations is as follows; $p = 2$ with DDD performed the worse, $p = 1$ with DDD performed the next best, $p = 2$ with no DDD performed the next best, and $p = 1$ with no DDD performed the best overall.

3.1.3 Optimal Solutions

One important detail that is not fully represented in Figures 7 and 8 is whether any of the methods were able to sample the optimal solutions of the problems, and if so how frequently. In the case of QAOA, it is clear that it never sampled the optimal solution(s) because the plots have visual indicators for those minimum energies. Table 4 details the minimum energies found for each of the solver methods along with the optimal minimum energy, computed using CPLEX (see Section 2.4). CPLEX does not provide information on degenerate ground states, if they exist, and therefore here degenerate ground states or how those states are sampled is not considered - only whether the optimal energy was found or not. Table 4 shows that both simulated annealing and quantum annealing are able to sample the optimal solutions for all 10 quadratic problems. Simulated annealing is also able to sample the ground state solution for all 10 *cubic* problems, and quantum annealing is able to sample the ground state solution for 6 out of the 10 *cubic* problems.

Importantly, the implementations of quantum annealing, simulated annealing, and CPLEX, all required the use of order-reduction techniques for the *cubic* Ising models (as opposed to QAOA which can handle the higher order terms natively in the algorithm without auxiliary terms). This means the SA, QA, and CPLEX solutions all contain auxiliary variables in their implementation. When reporting the objective function evaluations (e.g. energy), these auxiliary terms are *not* included in the objective function computation. The objective function computations are always performed strictly on the variable assignments found for the original variables (not auxiliary variables from the order reduction), defined by Eq. (1) and Eq. (2).

3.2 Quantum Annealing Schedule Tuning

Figure 9 shows the QA pause anneal schedule energy landscapes for a *quadratic* problem and a *cubic* problem. It is clear that longer annealing times have a better energy (objective function value), and more uniform. Shorter annealing times, in particular 10 microseconds, shows a more pronounced difference across the energy landscape where it becomes clear that pauses at an anneal fraction of $s = 0.1$ and $s = 0.9$ produce equally poor solution

Cubic Problem index	Ground state energy (CPLEX)	SA min. energy	SA min. energy count	QA min. energy	QA min. energy count	QAOA min. energy	QAOA min. energy count
0	-200	-200	15	-198	11	-70	1
1	-200	-200	52	-200	48	-90	1
2	-196	-196	215	-196	2	-80	1
3	-198	-198	37	-196	2	-90	1
4	-198	-198	97	-198	97	-74	1
5	-198	-198	125	-190	6	-70	1
6	-180	-180	322	-178	5	-74	1
7	-212	-212	115	-212	37	-84	1
8	-186	-186	20	-186	24	-84	1
9	-198	-198	125	-198	2	-110	1
Quadratic Problem index	Ground state energy (CPLEX)	SA min. energy	SA min. energy count	QA min. energy	QA min. energy count	QAOA min. energy	QAOA min. energy count
0	-179	-179	815	-179	29169	-75	1
1	-173	-173	916	-173	31358	-105	2
2	-177	-177	934	-177	31720	-75	1
3	-175	-175	795	-175	28545	-69	3
4	-189	-189	948	-189	31807	-99	1
5	-183	-183	965	-183	31877	-79	1
6	-179	-179	840	-179	29850	-81	1
7	-195	-195	984	-195	32278	-79	1
8	-171	-171	910	-171	30440	-65	1
9	-175	-175	833	-175	29677	-87	1

Table 4: Minimum energies sampled for the 10 *cubic* problem instances (top 10 rows), and the 10 *quadratic* problem instances (lower 10 rows). In addition to the absolute minimum energies found using simulated annealing, quantum annealing, and QAOA, counts of how many samples found those minimum energies are also reported in order to show the stability of those minimum objective function measurements. The exact solutions were computed using CPLEX (see Section 2.4). For the quantum annealing results, the minimum energies were taken from the modified anneal schedule with pauses data and the forward annealing with no schedule modification results. The simulated annealing data is always out of 1000 samples. The QA data is out of all samples, including the annealing time parameter sweep with no schedule modifications and the full pause anneal schedule parameter sweep. Similarly, the QAOA data is out of all energies from $p = 1$, $p = 2$, with and without dynamical decoupling. QA was able to find the optimal solutions for 18 of the 20 problems; the two *cubic* problem instances 5 and 6 were not solved to optimality. Notice that the ground state energy for the *cubic* problems are necessarily even and the ground state energy for the *quadratic* problems are necessarily odd.

quality and pauses around $s = 0.5$ produce better solution quality.

Figure 10 (right) shows some energy distributions for one of the *cubic* problems across different annealing times. The smallest and largest annealing times available on the two D-Wave quantum annealers. This plot shows that there is consistent trend towards longer annealing times performing better, however there is also a diminishing return on energy improvement as a function of annealing time as the mean energy distribution approaches the optimal solution. Figure 10 (left) shows the same histogram distribution of energies as a function of annealing times for one of the *quadratic* problem instances. The energy distributions for the other 18 problems are very similar to these two arbitrarily chosen plots.

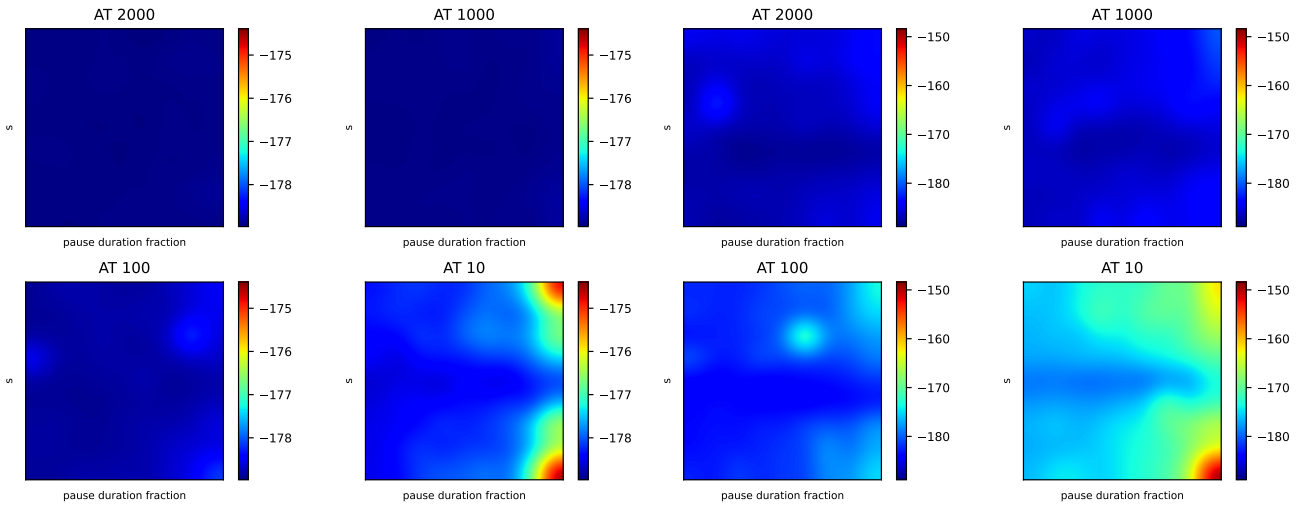

Figure 9: Mean energy landscape of the parameter search of QA anneal schedules with symmetric pauses. Each individual heatmap is plotting the mean sample energy found among the N anneals for that problem and device (N can vary, see Section 2.3) where the y-axis is the anneal fraction s at which the forward anneal pause occurs and the x-axis is the fraction of the anneal time that was spent in that pause at the anneal fraction s. Both the x and y-axis range is $0.1, 0.2, \ldots, 0.8, 0.9$. Each cluster of 4 heatmaps are performing the same grid search over anneal fraction and pause duration fraction, but repeated over the annealing times of 2000, 1000, 100, and 10 microseconds (shown the individual heatmap titles). The left hand four heatmaps are QA results for one *quadratic* problem instance, and the right hand four heatmaps are QA results for one *cubic* problem instance, both executed on `Advantage_system4.1`. The heatmaps show that there is a consistent low energy region slightly below $s = 0.5$ for any pause duration and for all annealing times. Longer annealing times clearly result in lower energy results.

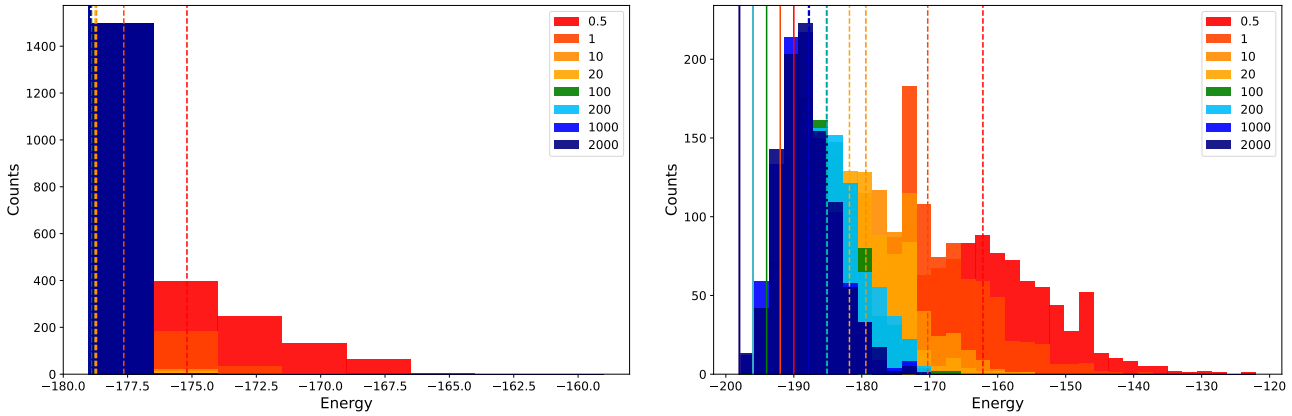

Figure 10: Histograms of the energy distributions from a *quadratic* instance (left) and a *cubic* instance (right) executed on `Advantage_system4.1` using forward quantum annealing execution (specifically, the schedule was not modified and so the default linearly interpolated schedule was applied). The annealing times were varied (annealing time in microseconds shown in legend) in order to show how the energy distribution changes as a function of annealing time. These histograms, while only the energies of a single one out of the 10 problem instances, are representative of the behavior of the other nine problems. Mean energy is marked with dashed vertical lines, and the minimum energy is marked with solid vertical lines. Notice that there is a clear improvement in solution quality as the anneal times get longer.

3.3 QAOA Angle Tuning

Figures 11, 12 show that the optimal QAOA angles found in the angle gridsearch across the 10 random problem instances are very similar, if not identical at the level of grid resolution we selected. This type of result has

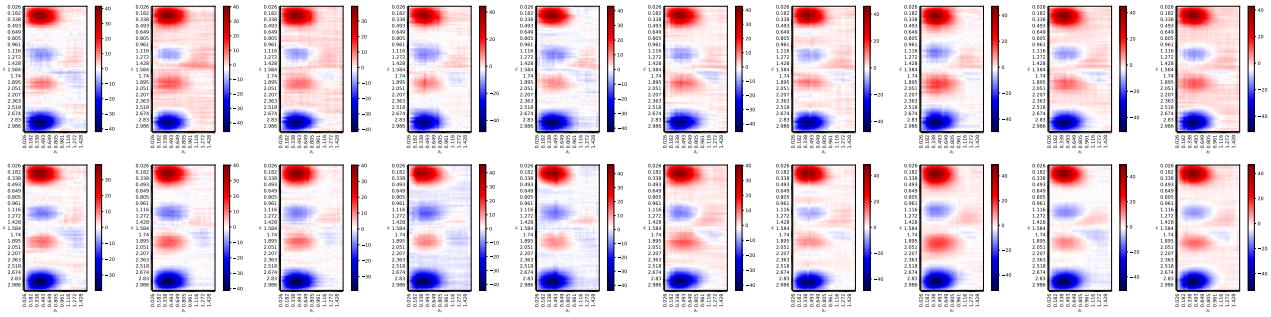

Figure 11: These plots show the mean energy landscape for the $p = 1$ QAOA angle grid search for all 10 *cubic* problems computed on `ibm_washington`, represented as the 10 columns. Top row is the bare QAOA circuit implementation. Bottom row is the QAOA circuit implementation with digital dynamical decoupling applied. The heatmap coloring for each β_0 and γ_0 corresponds to the mean sample energy out of the 10000 measurements made for that single circuit. Each heatmap uses bicubic interpolation.

been observed for other classes of random combinatorial optimization problems, such as maximum cut and the Sherrington-Kirkpatrick model [37, 49, 147–150]. If this behavior is consistent for large classes of combinatorial optimization problems then this could partially alleviate the hardness of the angle finding problem by allowing a single computation of optimal angles to be applied to any number of problems that fall into that class, and could potentially even help with angle finding for higher round QAOA.

Figures 13 and 14 show the $p = 2$ QAOA angle search spaces in the form of 3-d volumetric plots. At rounds greater than $p = 1$ it becomes difficult to meaningfully represent the angle search spaces; Figures 13 and 14 are a version of representing the objective function evaluations for this 4 dimensional search space. Examining such search spaces can be useful to discern patterns, such as parameter concentration, that could be used to infer good parameters at higher rounds or for similar problem instances. Although the full set of plots for $p = 2$ is not shown for all problem instances because of the number of plots, we observe parameter similar concentration across all problem instances for $p = 2$. The $p = 2$ parameter concentration can be seen in the legend of Figures 7 and 8 that show the best angle choices found in the experiments.

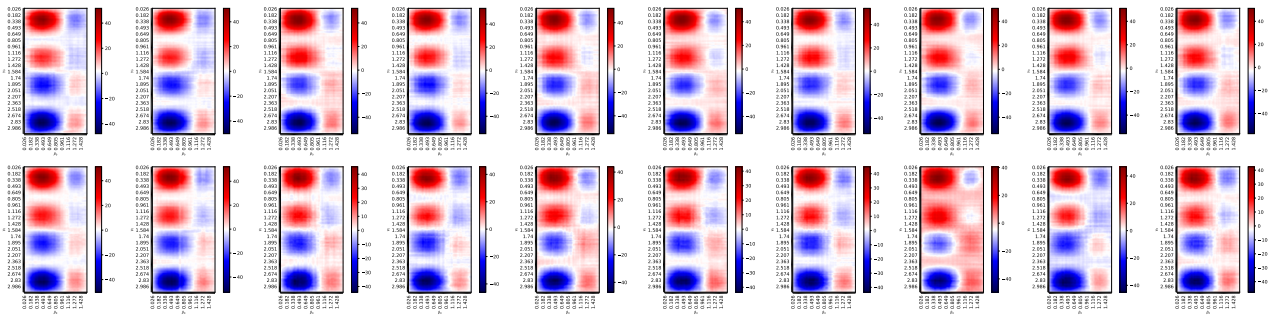

Figure 12: These plots show the mean energy landscape for the $p = 1$ QAOA angle grid search for all 10 *quadratic* problems computed on `ibm_washington`, represented as the 10 columns. Top row is the bare QAOA circuit implementation. Bottom row is the QAOA circuit implementation with digital dynamical decoupling applied. The heatmap coloring for each β_0 and γ_0 corresponds to the mean sample energy out of the 10000 measurements made for that single circuit. Each heatmap uses bicubic interpolation.

3.4 Classical Simulation of QAOA

As described in Section 2.2.5 it is possible for HPC resources to simulate the studied QAOA circuits for $p = 1, 2$. At least, it is possible to simulate the ideal *mean expectation values* for arbitrary angles. Figure 15 shows the classically computed $p = 1$ parameter search space for all of the *quadratic* and *cubic* problem instances. The mean energy for the best angle combinations, computed exactly, are also shown as part of the histograms in Figures 7 and 8. These plots should be compared against Figures 11 and 12, where two things are clear. First, the trends of

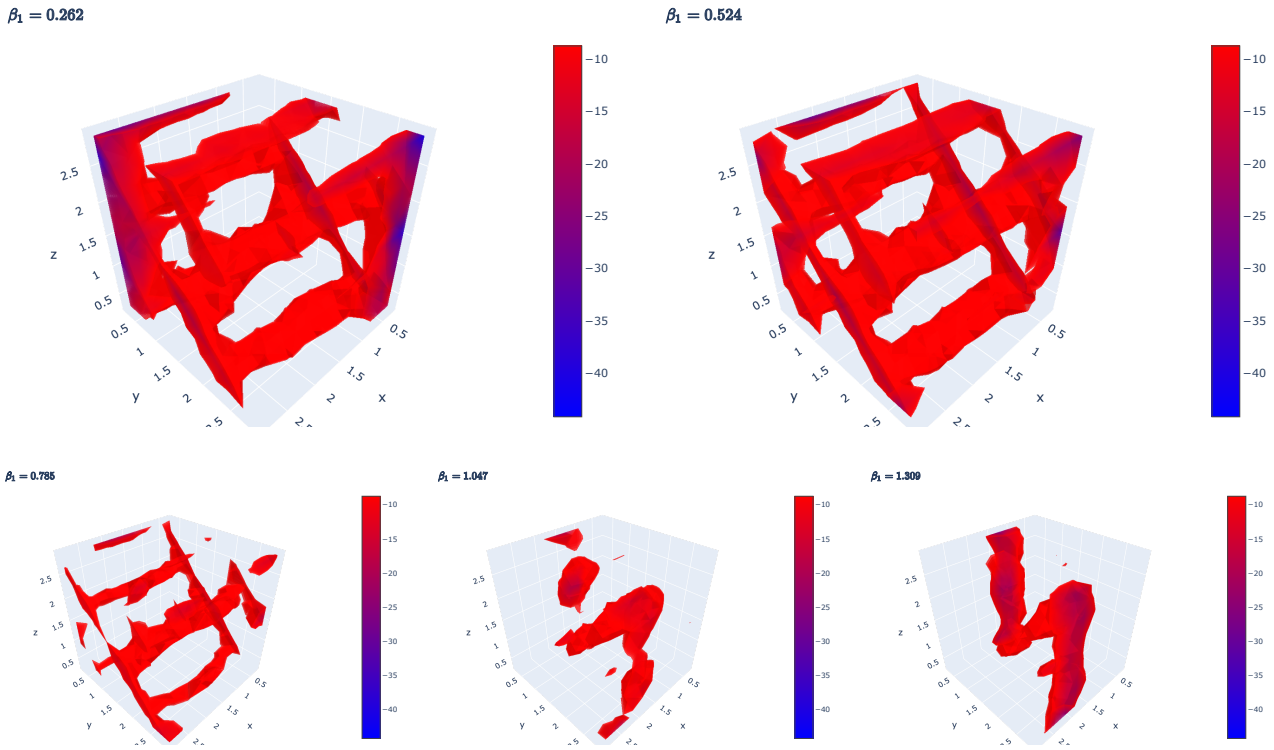

Figure 13: 2-round QAOA search space volumetric plots of the mean energy for each angle combination, with no dynamical decoupling, for one of the 10 *cubic* problem instances computed on `ibm_washington`. Here the heatmap coloring is consistent between all five sub-plots; blue regions are the lowest (mean) energy regions and the regions with no color volume being plotted are the highest (mean) energy regions. Each of the five sub-plots corresponds to each of the five β_1 angles that the grid search was performed on (the exact β_1 angles are shown in each figures title). The x, y, and z axis correspond to the γ_0, γ_1, and β_0 angles respectively. Therefore, the volumes being plotted in each figure contain $11^3 = 1331$ values (each of those values is the mean of the 10000 measurements made for that specific set of angles), of which only the 15th percentile of energy (or lower) is being plotted as solid volumes. Although the rest of the volumetric plots for the other 9 problems are not shown, the 2-round QAOA search spaces are very similar across the 10 problem instances.

the landscape computed on the NISQ computer are very similar to the trends in the exact energy landscape - e.g. there are no obvious biases or lost local minima/maxima not found by the quantum hardware results. Second, the optimal angle combinations result in a significantly worse mean expectation value (energy) on `ibm_washington`; compared to the exact QAOA simulation, the hardware energies get approximately 50% to the ideal computation. Interestingly, this is also true for the angle symmetries that result in the highest expectation values (which in this case we do not want since it is a minimization problem). Specifically, the experimental energies are symmetric and the highest expectation values are also approximately 50% of what the exact classical simulations show could be sampled for those angle choices.

An observation that has not been made before for QAOA sampling combinatorial optimization problems is that adding in higher order terms to the Ising models (e.g. going from the *quadratic* instances to the *cubic* instances) does not substantially change the parameter concentration. This can be seen especially clearly in Figure 15, and is also seen in the NISQ computer experiments from Figures 11 and 12. The exact shape of the energy landscape does change when the higher order terms are added, but the best parameter combination region is nearly identical. This gives evidence that parameter coenctration may be transferable across problem instances even when adding or removing higher order terms.

Figure 18 in Section B examines detailed (normalized) differences in the experimentally computed QAOA energy landscapes of Figures 11 and 12, compared to the ideal simulations of Figure 15. We note that such a comparison could be made for $p = 2$, but as with even representing the search space (such as in Figures 13 and 14) becomes difficult to meaningfully visually represent.

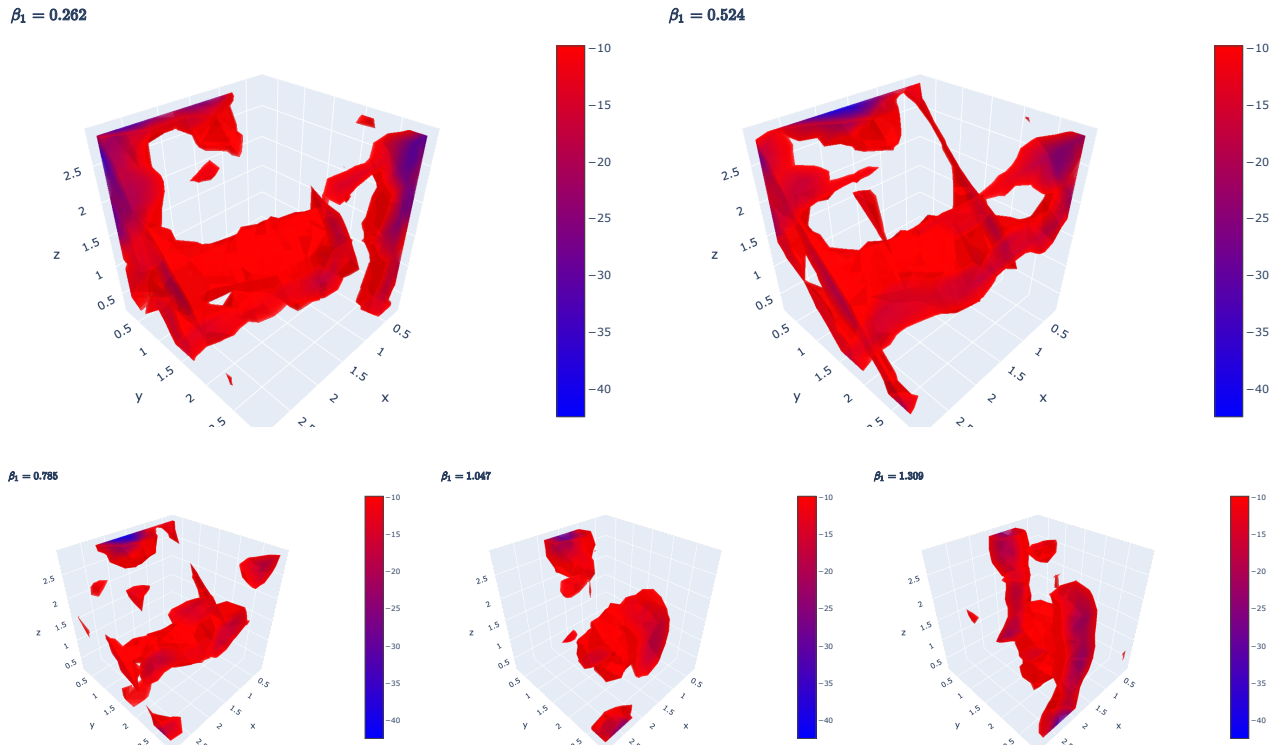

Figure 14: QAOA $p = 2$ round energy landscape, plotted as 5 separate 3-d volumetric plots, for one of the 10 *quadratic* problem instances. Computed on `ibm_washington` (with no dynamical decoupling). As in Figure 13, the x, y, and z axis correspond to the γ_0, γ_1, and β_0 angles respectively and the volumes being plotted in each figure contain $11^3 = 1331$ values.

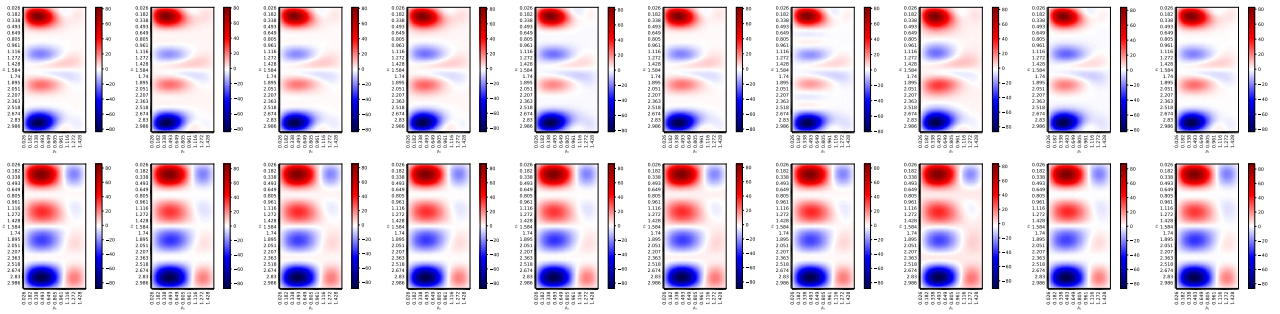

Figure 15: Exact classical simulation of the $p = 1$ QAOA energy landscape for the 10 *cubic* instances (top) and the 10 *quadratic* instances (bottom). Each point on the landscape is the mean expectation value for executing QAOA at that angle combination. Notice that there are slight variations of the energy landscape across the 10 problem instances, for both class of problems, but there is clear parameter concentration, as seen in Figures 11 and 12.

4 Discussion and Conclusion

This paper has presented state of the art QAOA results, in particular the largest QAOA experimental demonstration to date, which enabled a direct comparison against quantum annealing. This is also one of the largest quantum system simulations performed on a circuit model quantum computer to date, using approximately 3000 circuit instructions on 127 qubits. Specifically we found the following:

1. Quantum annealing finds lower energy solutions compared to QAOA, and is able to sample the optimal solution at least once for all 10 *quadratic* problems and 6 out of the 10 *cubic* problems.

2. QAOA samples all of the problem instances better than random sampling

3. The short depth QAOA circuit can be applied to heavy hexagonal lattices of any size, which means that as heavy hexagonal hardware improves these short depth QAOA circuits can be used.

4. Digital dynamical decoupling can help improve the quality of computation on NISQ computers, even for relatively short depth QAOA circuits. However, as observed in other empirical results the improvements are not uniform - and in particular for $p = 1$ dynamical decoupling did not improve the computation.

5. Consistent with empirical results on other problems, we observed parameter concentration for both $p = 1$ and $p = 2$ QAOA circuits. This is encouraging since it indicates that for some classes of similar problem instance it may be possible to develop good heuristics for angle selection for high-round QAOA. This result is consistent with other similar classes of random combinatorial optimization problems.

6. The exact classical simulation of QAOA showed that the experimental QAOA $p = 1$ energy landscapes computed on `ibm_washington` were not biased in particular regions, but overall performed significantly worse (roughly one half) than the theoretical performance.

Figures 7 and 8 hint at a possible future research direction involving the selection of good QAOA angles at higher rounds. This observation is that the best found β_0 and γ_0 in the gridsearch are usually similar - because the $p = 2$ gridsearch was not as granular as the $p = 1$ gridsearch, this similarity is not always highly apparent. This suggests that it may be possible to extrapolate good angle choices for high round QAOA, using the best angles found at $p-1$. This type of extrapolation technique has been used in other studies for studying the scaling of QAOA using classical simulation [27]. Additionally, we have seen evidence that the addition of the higher order terms does not substantially change the concentration of good QAOA angles. This then leads to the natural question of how far this extends to higher order terms beyond cubic terms, and whether QAOA parameter concentration holds for classes combinatorial optimization problems where higher order terms are added to a fixed underlying connectivity structure. Parameter concentration for QAOA on problems which contain higher order terms is very under-investigated, and would be an interesting topic of future research.

Because of the increasing availability of NISQ computers with increasing qubit count (now with hundreds of qubits), we encourage the algorithmic development of novel short depth circuits alongside experimental evaluation of these shallow circuits on full hardware lattices. Both the QAOA circuit construction algorithm and the QA embedding algorithm that we have outlined are scalable to larger lattice sizes – as are the problem instances that match those hardware graphs. This means that these algorithms can be used in the future as hardware improves and increases in scale.

Although here we used an angle gridsearch for the QAOA implementation, this is not scalable to high round QAOA. High round QAOA needs to be experimentally evaluated as the hardware and algorithms improve, and therefore an important future research topic is to utilize more scalable angle finding algorithms for high round QAOA on quantum computers.

The capability of QAOA to natively accept higher order terms seems to be under-utilized when evaluating QAOA capabilities, both numerically in ideal simulations, and on NISQ computers. As a future research direction, we encourage further testing of QAOA on problem types which contain higher order terms.

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. The research presented in this article was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20220656ER and 20210114ER. This research used resources provided by the Darwin testbed 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). We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team. We thank the IBM Quantum team for technical support on the IBM Quantum systems throughout this project.

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 technical report number LA-UR-23-22023.

A Detailed QAOA Circuit Drawings

Qiskit circuit drawings for $p = 1$ QAOA circuits, applied to both a *quadratic* and a *cubic* problem instance, are shown in Figure 16. Figure 16 shows that both the QAOA circuits for the *quadratic* and *cubic* problems have a maximum CNOT depth of 6. These two circuits use all 127 qubits on `ibm_washington`, and implicitly there are measurement operations performed on all of the qubits at the end of the computation, the results of which are stored in a classical register (not drawn to conserve the total space used). All angles β_0, γ_0 are set to $\frac{\pi}{4}$ in these two circuits for demonstration purposes.

Figure 17 shows a $p = 1$ QAOA circuit, applied to *cubic* problem instance, with X, X dynamical decoupling passes inserted on idle qubits, drawn as in a scheduled as late as possible (ALAP) timeline view. Note that `rz` gates are virtual gates [151], meaning that they are executed on the software level and have no error rate; the rz gates are included in this scheduled circuit diagram, but since they take up very little time they are not highly visible. The QAOA circuit in Figure 17 is composed of 1384 `rz` gates, 959 `sx` gates, 284 `cx` gates, 15 `x` gates, and 127 measurement instructions. The digital dynamical decoupling pass adds an additional 422 `x` gates, resulting in a total of 437 `x` gates. The exact number of instructions used in the QAOA circuits changes depending on what β, γ angles are used, as well as the exact compilation procedure used.

The instruction durations, in particular for CNOT gates, can vary. This can be seen in Figure 17. The `sx` and `x` gates all had gate times of 3.555×10^{-8} seconds. The measure operation uses 8.64×10^{-7} seconds. The `cx` gate instructions have a mean duration of $5.595\,15 \times 10^{-7}$ seconds with a standard deviation of $2.085\,531\,4 \times 10^{-7}$ seconds.

B QAOA $p = 1$ Normalized Energy Differences

Figure 18 shows detailed heatmaps of how the $p = 1$ QAOA energy landscapes computed experimentally compare against the ideal simulations of the QAOA circuits. This comparison shows where the normalized energy landscapes are in disagreement, e.g. it is showing regions where the normalized experimental results are inconsistent with the exact energy landscape (computed classically). If the experimental results were a uniformly scaled version of the classical distribution, Figure 18 would show no deviations above or below 0. This shows that the experimental results agree with the classical computations in the left regions of the energy landscape where the mean expectation values are more extreme. Conversely, the right hand region of the plots in Figure 18 show that there is more inconsistency where the mean expectation values are more closely centered at 0.

C Error Rates and Decoherence Times on ibm_washington

Figure 19 plots the T1 and T2 coherence times and calibration gate error rates on `ibm_washington` when executing all QAOA circuits. The T1 times have a mean of 0.000099 seconds with a standard deviation of 0.0000274 seconds. The T2 times have a mean of 0.0000948 seconds with a standard deviation of 0.0000497 seconds. The `rz` gate error rates are not in this figure because although they are part of the native gateset for IBMQ devices, they are virtual gates and therefore have an error rate of 0 [151]. The number of data points being plotted for single qubit metrics is 284480 [10], and 318080 [11] for cx gates. These numbers are computed in terms of jobs, instead of circuits, because all circuits in each job have the same calibrations.

Figure 19 shows that the error rates and coherence times vary significantly across the whole chip QAOA experiments. Because the QAOA circuits use all of the gate operations and qubits available on the device, these error rates are consistent with the QAOA computations in Figures 7 and Figures 8. Using the CNOT gate times from Section A, we can estimate the mean time used to execute the CNOT gate instructions for $p = 2$ QAOA circuits would be $6.714\,18 \times 10^{-6}$ seconds, which is smaller than the mean T1 and T2 coherence times in Figure 19. This suggests that qubit de-coherence is not the primary source of the errors in the QAOA computation.

[10] 284480 comes from 27 device jobs for the 1-round angle gridsearch, and 29 device jobs for 2-round angle gridsearch, 10 problems, with and without dynamical decoupling, *quadratic* and *qubic* problem instances, and 127 qubits: $27{\cdot}10{\cdot}127{\cdot}2{\cdot}2 = 137160$, $29{\cdot}10{\cdot}127 = 147320$, and $284480 = 137160 + 147320$

[11] 318080 comes from 27 device jobs for the 1-round angle gridsearch, and 29 device jobs for 2-round angle gridsearch, 10 problems, with and without dynamical decoupling, *quadratic* and *qubic* problem instances, and 142 cx gates: $27 \cdot 10 \cdot 142 \cdot 2 \cdot 2 = 153360$, $29 \cdot 10 \cdot 142 \cdot 2 \cdot 2 = 164720$, and $318080 = 153360 + 164720$

Figure 16: Logical low depth circuit drawings, in Qiskit [71], for 1-round QAOA on a *quadratic* instance (left), and 1-round QAOA a *cubic* problem instance (right). The logical circuit gateset is entirely composed of the gateset `cx, rz, rx, h`.

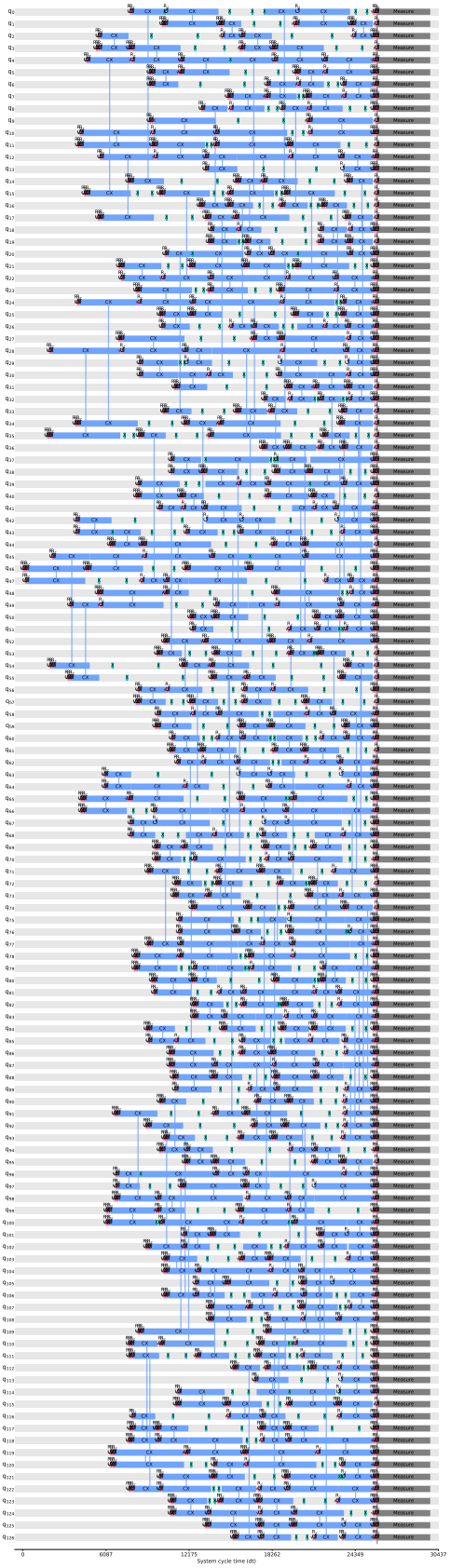

Figure 17: One round QAOA applied to a *cubic* problem instance (with arbitrarily chosen $\beta_0 = \frac{\pi}{5}, \gamma_0 = \frac{\pi}{5}$) with inserted X, X dynamical decoupling sequences, and scheduled using the real backend instruction times of `ibm_washington` with the instruction stop times being scheduled as late as possible. The basis gates for this circuit have been compiled to the native gateset for `ibm_washington`, which is `cx, rz, sx, x`.

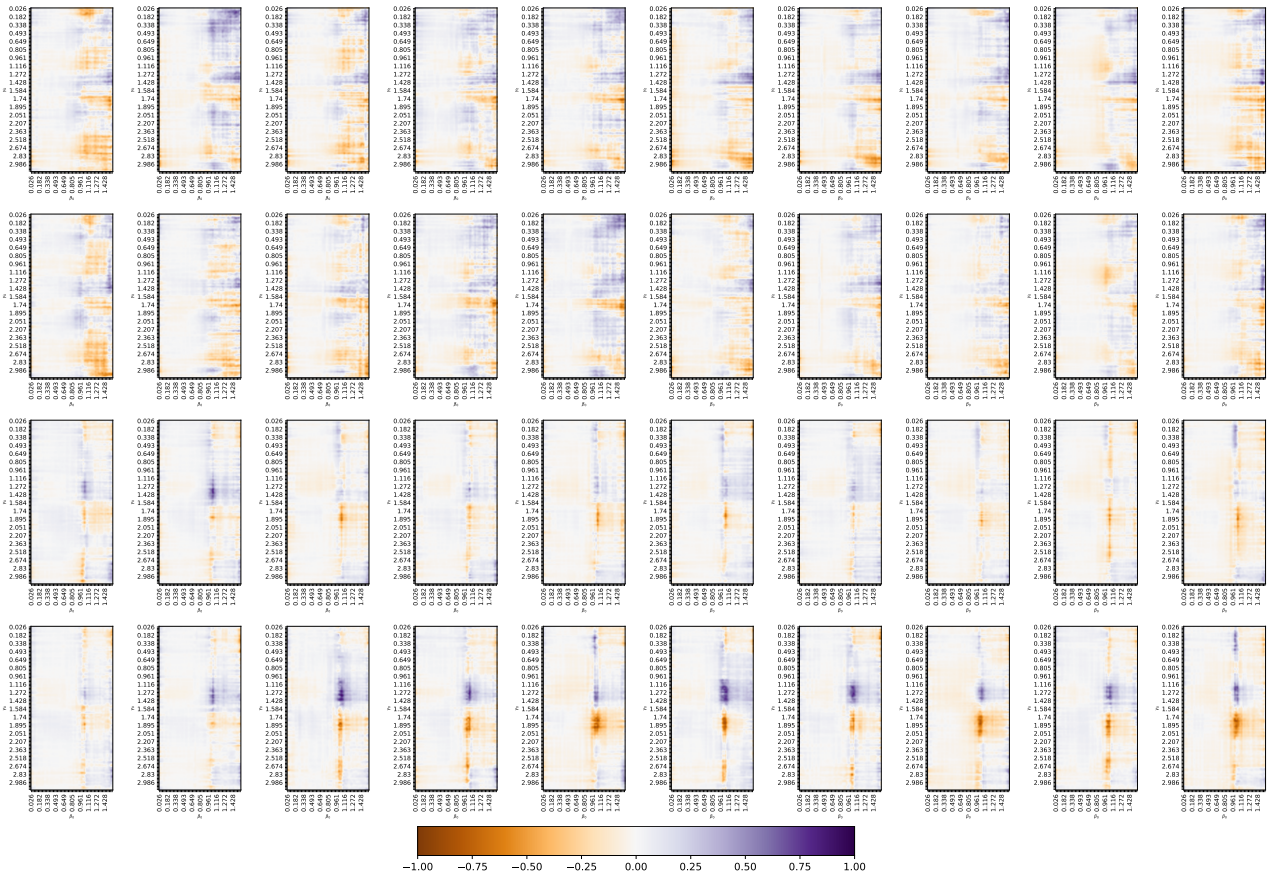

Figure 18: Normalized $p = 1$ QAOA energy differences between the ideal simulation and the experimental results. Each column corresponds to the 10 unique random problem instances. The top row are the *cubic* problem instances without dynamical decoupling, the second row down are the results from the *cubic* problem instances with dynamical decoupling. The third row down are *quadratic* problem instances without dynamical decoupling, and the last row are the results from the *quadratic* problem instances with dynamical decoupling. The goal of these plots are to show where the energy landscape trends of the experimental results were significantly different from the ideal computation, which was computed classically. In order to compute these plots the energy dataset for the experimental results were normalized into a range of $[0, 1]$, as was the energy dataset for the ideal classical simulation. Next, the normalized ideal energy data was subtracted from the experimental dataset. This means that the possible range of values being plotted is $(-1, +1)$ where points close to $+1$ or -1 show high disagreement between the ideal and experimental results (in terms of the normalized differences of the energy distributions, not the absolute value of the energies).

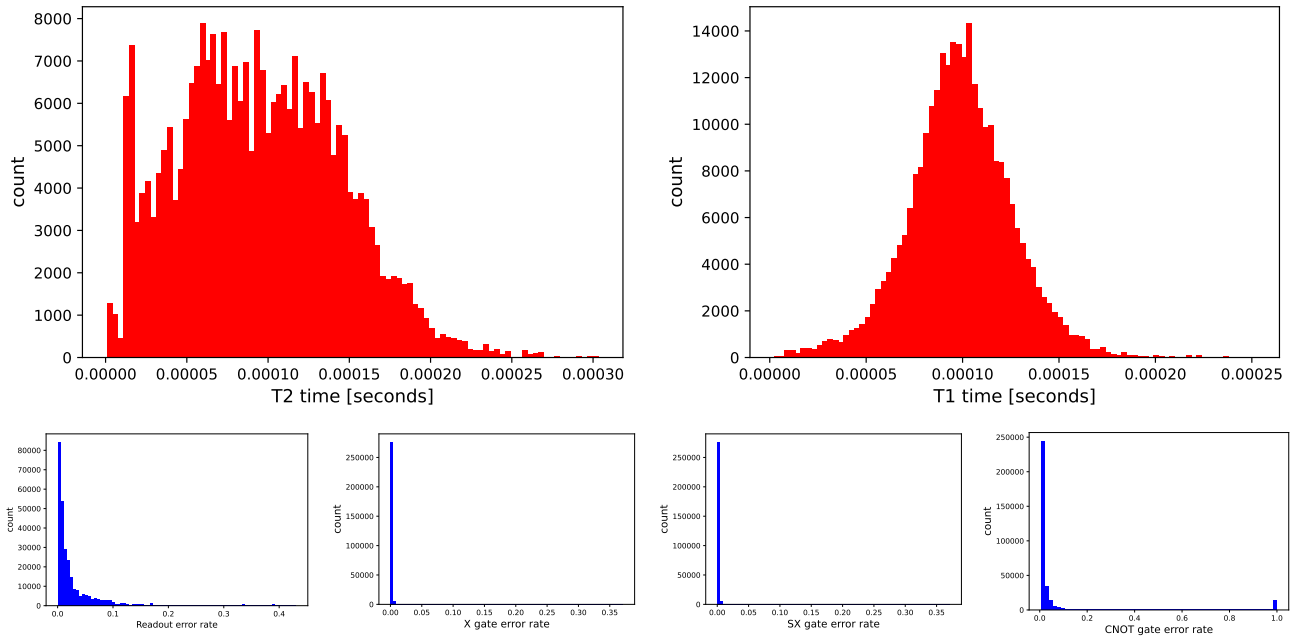

Figure 19: Distribution of qubit T1 and T2 coherence times, in seconds, (top row) and gate error rates (bottom row) for readout, `x`, `sx`, and `cx` gates, on `ibm_washington` read from left to right. These error rates and qubit T1 and T2 times are aggregated across all circuit executions for both $p = 1$, $p = 2$ QAOA on all 10 *quadratic* and *cubic* instances. These plots are the error rates on all single, two qubit, and readout errors from the latest device calibration data for each job id (where each job id contains 250 QAOA circuits). Notice that there are a number of CNOT error rates that are exactly 1; those error rates are not actually measured to be 1, rather those are the default values for gates whose error rate was not measured. Therefore, the error rate of 1 actually indicates that the true error rate is not known.

References

[1] Stuart Hadfield et al. "From the Quantum Approximate Optimization Algorithm to a Quantum Alternating Operator Ansatz". In: *Algorithms* 12.2 (2019), p. 34. DOI: 10.3390/a12020034. URL: https://doi.org/10.3390%2Fa12020034.

[2] Jeremy Cook, Stephan Eidenbenz, and Andreas Bärtschi. "The Quantum Alternating Operator Ansatz on Maximum k-Vertex Cover". In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2020, pp. 83–92. DOI: 10.1109/QCE49297.2020.00021.

[3] Zhihui Wang et al. "XY mixers: Analytical and numerical results for the quantum alternating operator ansatz". In: *Physical Review A* 101.1 (2020). DOI: 10.1103/physreva.101.012320. URL: https://doi.org/10.1103%2Fphysreva.101.012320.

[4] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. *A Quantum Approximate Optimization Algorithm*. 2014. DOI: 10.48550/ARXIV.1411.4028. URL: https://arxiv.org/abs/1411.4028.

[5] Edward Farhi et al. *Quantum Computation by Adiabatic Evolution*. 2000. DOI: 10.48550/ARXIV.QUANT-PH/0001106. URL: https://arxiv.org/abs/quant-ph/0001106.

[6] 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.

[7] 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.

[8] 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. URL: https://dx.doi.org/10.1088/1361-6633/ab85b8.

[9] Sheir Yarkoni et al. "Quantum annealing for industry applications: introduction and review". In: *Reports on Progress in Physics* 85.10 (2022), p. 104001. DOI: 10.1088/1361-6633/ac8c54. URL: https://doi.org/10.1088%2F1361-6633%2Fac8c54.

[10] 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.

[11] 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.

[12] Aleta Berk Finnila et al. "Quantum annealing: A new method for minimizing multidimensional functions". In: *Chemical physics letters* 219.5-6 (1994), pp. 343–348.

[13] Mark W Johnson et al. "Quantum annealing with manufactured spins". In: *Nature* 473.7346 (2011), pp. 194–198. DOI: 10.1038/nature10012.

[14] 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.

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

[16] 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.

[17] Byron Tasseff et al. *On the Emerging Potential of Quantum Annealing Hardware for Combinatorial Optimization*. 2022. DOI: 10.48550/ARXIV.2210.04291. URL: https://arxiv.org/abs/2210.04291.

[18] Yuval R. Sanders et al. "Compilation of Fault-Tolerant Quantum Heuristics for Combinatorial Optimization". In: *PRX Quantum* 1.2 (2020). DOI: 10.1103/prxquantum.1.020312. URL: https://doi.org/10.1103%2Fprxquantum.1.020312.

[19] Phillip C. Lotshaw et al. "Scaling quantum approximate optimization on near-term hardware". In: *Scientific Reports* 12.1 (2022). DOI: 10.1038/s41598-022-14767-w. URL: https://doi.org/10.1038%2Fs41598-022-14767-w.

[20] Tameem Albash and Daniel A. Lidar. "Demonstration of a Scaling Advantage for a Quantum Annealer over Simulated Annealing". In: *Phys. Rev. X* 8 (3 2018), p. 031016. DOI: 10.1103/PhysRevX.8.031016. URL: https://link.aps.org/doi/10.1103/PhysRevX.8.031016.

[21] 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.

[22] Edward Farhi and Aram W Harrow. *Quantum Supremacy through the Quantum Approximate Optimization Algorithm*. 2016. DOI: 10.48550/ARXIV.1602.07674. URL: https://arxiv.org/abs/1602.07674.

[23] Lucas T. Brady et al. "Optimal Protocols in Quantum Annealing and Quantum Approximate Optimization Algorithm Problems". In: *Phys. Rev. Lett.* 126 (7 2021), p. 070505. DOI: 10.1103/PhysRevLett.126.070505. URL: https://link.aps.org/doi/10.1103/PhysRevLett.126.070505.

[24] Madita Willsch et al. "Benchmarking the quantum approximate optimization algorithm". In: *Quantum Information Processing* 19.7 (2020), pp. 1–24. DOI: 10.1007/s11128-020-02692-8.

[25] Stefan H. Sack and Maksym Serbyn. "Quantum annealing initialization of the quantum approximate optimization algorithm". In: *Quantum* 5 (2021), p. 491. DOI: 10.22331/q-2021-07-01-491. URL: https://doi.org/10.22331%2Fq-2021-07-01-491.

[26] John Golden et al. *Evidence for Super-Polynomial Advantage of QAOA over Unstructured Search*. 2022. DOI: 10.48550/ARXIV.2202.00648. URL: https://arxiv.org/abs/2202.00648.

[27] John Golden et al. *The Quantum Alternating Operator Ansatz for Satisfiability Problems*. 2023. DOI: 10.48550/ARXIV.2301.11292. URL: https://arxiv.org/abs/2301.11292.

[28] Lennart Binkowski et al. *Elementary Proof of QAOA Convergence*. arXiv:2302.04968v1. 2023. eprint: 2302.04968.

[29] Thomas Lubinski et al. *Optimization Applications as Quantum Performance Benchmarks*. 2023. DOI: 10.48550/ARXIV.2302.02278. URL: https://arxiv.org/abs/2302.02278.

[30] Elijah Pelofske et al. "Sampling on NISQ Devices: "Who's the Fairest One of All?"". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2021. DOI: 10.1109/qce52317.2021.00038. URL: https://doi.org/10.1109%2Fqce52317.2021.00038.

[31] Hayato Ushijima-Mwesigwa et al. "Multilevel Combinatorial Optimization across Quantum Architectures". In: *ACM Transactions on Quantum Computing* 2.1 (2021). ISSN: 2643-6809. DOI: 10.1145/3425607. URL: https://doi.org/10.1145/3425607.

[32] Michael Streif and Martin Leib. *Comparison of QAOA with Quantum and Simulated Annealing*. 2019. DOI: 10.48550/ARXIV.1901.01903. URL: https://arxiv.org/abs/1901.01903.

[33] Elijah Pelofske, Andreas Bärtschi, and Stephan Eidenbenz. "Quantum Annealing vs. QAOA: 127 Qubit Higher-Order Ising Problems on NISQ Computers". In: *Lecture Notes in Computer Science*. Springer Nature Switzerland, 2023, pp. 240–258. DOI: 10.1007/978-3-031-32041-5_13. URL: https://doi.org/10.1007%2F978-3-031-32041-5_13.

[34] Adrien Suau et al. *Single-Qubit Cross Platform Comparison of Quantum Computing Hardware*. 2021. DOI: 10.48550/ARXIV.2108.11334. URL: https://arxiv.org/abs/2108.11334.

[35] Guido Pagano et al. "Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator". In: *Proceedings of the National Academy of Sciences* 117.41 (2020), pp. 25396–25401. DOI: 10.1073/pnas.2006373117. URL: https://doi.org/10.1073%2Fpnas.2006373117.

[36] Johannes Weidenfeller et al. "Scaling of the quantum approximate optimization algorithm on superconducting qubit based hardware". In: *Quantum* 6 (Dec. 2022), p. 870. ISSN: 2521-327X. DOI: 10.22331/q-2022-12-07-870. URL: https://doi.org/10.22331/q-2022-12-07-870.

[37] Matthew P. Harrigan et al. "Quantum approximate optimization of non-planar graph problems on a planar superconducting processor". In: *Nature Physics* 17.3 (2021), pp. 332–336. DOI: 10.1038/s41567-020-01105-y. URL: https://doi.org/10.1038%2Fs41567-020-01105-y.

[38] Dylan Herman et al. *Portfolio Optimization via Quantum Zeno Dynamics on a Quantum Processor*. 2022. DOI: 10.48550/ARXIV.2209.15024. URL: https://arxiv.org/abs/2209.15024.

[39] Pradeep Niroula et al. "Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer". In: *Scientific Reports* 12.1 (2022), pp. 1–14. DOI: 10.1038/s41598-022-20853-w.

[40] Leo Zhou et al. "Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices". In: *Phys. Rev. X* 10 (2 2020), p. 021067. DOI: 10.1103/PhysRevX.10.021067. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.021067.

[41] Joao Basso et al. "The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model". en. In: Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. DOI: 10.4230/LIPICS.TQC.2022.7. URL: https://drops.dagstuhl.de/opus/volltexte/2022/16514/.

[42] Zhihui Wang et al. "Quantum approximate optimization algorithm for MaxCut: A fermionic view". In: *Phys. Rev. A* 97 (2 2018), p. 022304. DOI: 10.1103/PhysRevA.97.022304. URL: https://link.aps.org/doi/10.1103/PhysRevA.97.022304.

[43] Gavin E. Crooks. *Performance of the Quantum Approximate Optimization Algorithm on the Maximum Cut Problem*. 2018. DOI: 10.48550/ARXIV.1811.08419. URL: https://arxiv.org/abs/1811.08419.

[44] Gian Giacomo Guerreschi and Anne Y Matsuura. "QAOA for Max-Cut requires hundreds of qubits for quantum speed-up". In: *Scientific reports* 9.1 (2019), pp. 1–7. DOI: 10.1038/s41598-019-43176-9.

[45] Kunal Marwaha. "Local classical MAX-CUT algorithm outperforms $p = 2$ QAOA on high-girth regular graphs". In: *Quantum* 5 (2021), p. 437. DOI: 10.22331/q-2021-04-20-437. URL: https://doi.org/10.22331%2Fq-2021-04-20-437.

[46] M. B. Hastings. *Classical and Quantum Bounded Depth Approximation Algorithms*. 2019. DOI: 10.48550/ARXIV.1905.07047. URL: https://arxiv.org/abs/1905.07047.

[47] Zain Hamid Saleem. "Max-independent set and the quantum alternating operator ansatz". In: *International Journal of Quantum Information* 18.04 (2020), p. 2050011. DOI: 10.1142/S0219749920500112.

[48] Pierre Dupuy de la Grand'rive and Jean-Francois Hullo. *Knapsack Problem variants of QAOA for battery revenue optimisation*. 2019. DOI: 10.48550/ARXIV.1908.02210. URL: https://arxiv.org/abs/1908.02210.

[49] Edward Farhi et al. "The Quantum Approximate Optimization Algorithm and the Sherrington-Kirkpatrick Model at Infinite Size". In: *Quantum* 6 (2022), p. 759. DOI: 10.22331/q-2022-07-07-759. URL: https://doi.org/10.22331%2Fq-2022-07-07-759.

[50] Shuxian Jiang et al. "Quantum annealing for prime factorization". In: *Scientific reports* 8.1 (2018), pp. 1–9. DOI: 10.1038/s41598-018-36058-z.

[51] Xiangmin Ji et al. "New advanced computing architecture for cryptography design and analysis by D-Wave quantum annealer". In: *Tsinghua Science and Technology* 27.4 (2022), pp. 751–759. DOI: 10.26599/TST.2021.9010022.

[52] Raouf Dridi and Hedayat Alghassi. "Prime factorization using quantum annealing and computational algebraic geometry". In: *Scientific reports* 7.1 (2017), pp. 1–10.

[53] WangChun Peng et al. "Factoring larger integers with fewer qubits via quantum annealing with optimized parameters". In: *SCIENCE CHINA Physics, Mechanics & Astronomy* 62.6 (2019), pp. 1–8.

[54] Richard H Warren. "Factoring on a quantum annealing computer". In: *Quantum Information & Computation* 19.3-4 (2019), pp. 252–261.

[55] Olawale Titiloye and Alan Crispin. "Quantum annealing of the graph coloring problem". In: *Discrete Optimization* 8.2 (2011), pp. 376–384. ISSN: 1572-5286. DOI: https://doi.org/10.1016/j.disopt.2010.12.001. URL: https://www.sciencedirect.com/science/article/pii/S1572528610000721.

[56] Julia Kwok and Kristen Pudenz. *Graph Coloring with Quantum Annealing*. 2020. DOI: 10.48550/ARXIV.2012.04470. URL: https://arxiv.org/abs/2012.04470.

[57] Vaibhaw Kumar et al. "Quantum annealing for combinatorial clustering". In: *Quantum Information Processing* 17.2 (2018), pp. 1–14. DOI: 10.1007/s11128-017-1809-2.

[58] 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.

[59] Erica Grant, Travis S. Humble, and Benjamin Stump. "Benchmarking Quantum Annealing Controls with Portfolio Optimization". In: *Phys. Rev. Applied* 15 (1 2021), p. 014012. DOI: 10.1103/PhysRevApplied.15.014012. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.15.014012.

[60] Gili Rosenberg et al. "Solving the Optimal Trading Trajectory Problem Using a Quantum Annealer". In: *IEEE Journal of Selected Topics in Signal Processing* 10.6 (2016), pp. 1053–1060. DOI: 10.1109/JSTSP.2016.2574703.

[61] Davide Venturelli and Alexei Kondratyev. "Reverse quantum annealing approach to portfolio optimization problems". In: *Quantum Machine Intelligence* 1.1-2 (2019), pp. 17–30. DOI: 10.1007/s42484-019-00001-w. URL: https://doi.org/10.1007%2Fs42484-019-00001-w.

[62] Phillip C Lotshaw et al. "Simulations of frustrated Ising Hamiltonians using quantum approximate optimization". In: *Philosophical Transactions of the Royal Society A* 381.2241 (2022), p. 20210414.

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

[64] Andrew D King et al. "Qubit spin ice". In: *Science* 373.6554 (2021), pp. 576–580. DOI: 10.1126/science.abe2824.

[65] Shiyu Zhou et al. "Experimental realization of classical $\mathbb{Z}_2$ spin liquids in a programmable quantum device". In: *Phys. Rev. B* 104 (8 2021), p. L081107. DOI: 10.1103/PhysRevB.104.L081107. URL: https://link.aps.org/doi/10.1103/PhysRevB.104.L081107.

[66] Andrew D. King et al. "Quantum Annealing Simulation of Out-of-Equilibrium Magnetization in a Spin-Chain Compound". In: *PRX Quantum* 2 (3 2021), p. 030317. DOI: 10.1103/PRXQuantum.2.030317. URL: https://link.aps.org/doi/10.1103/PRXQuantum.2.030317.

[67] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[68] Thomas A Caswell et al. *matplotlib/matplotlib*. Version v3.4.3. DOI: 10.5281/zenodo.5194481.

[69] J. D. Hunter. "Matplotlib: A 2D graphics environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55.

[70] Aric Hagberg, Pieter Swart, and Daniel S Chult. *Exploring network structure, dynamics, and function using NetworkX*. Tech. rep. Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[71] Qiskit contributors. *Qiskit: An Open-source Framework for Quantum Computing*. 2023. DOI: 10.5281/zenodo.2573505.

[72] Christopher Chamberland et al. "Topological and Subsystem Codes on Low-Degree Graphs with Flag Qubits". In: *Phys. Rev. X* 10 (1 2020), p. 011022. DOI: 10.1103/PhysRevX.10.011022. URL: https://link.aps.org/doi/10.1103/PhysRevX.10.011022.

[73] Stefanie Zbinden et al. "Embedding algorithms for quantum annealers with chimera and pegasus connection topologies". In: *International Conference on High Performance Computing*. Springer. 2020, pp. 187–206. DOI: 10.1007/978-3-030-50743-5_10.

[74] Tomas Boothby, Andrew D King, and Aidan Roy. "Fast clique minor generation in Chimera qubit connectivity graphs". In: *Quantum Information Processing* 15.1 (2016), pp. 495–508.

[75] Robert Patton, Catherine Schuman, Thomas Potok, et al. "Efficiently embedding QUBO problems on adiabatic quantum computers". In: *Quantum Information Processing* 18.4 (2019), pp. 1–31. DOI: 10.1007/s11128-019-2236-3.

[76] Nike Dattani, Szilard Szalay, and Nick Chancellor. *Pegasus: The second connectivity graph for large-scale quantum annealing hardware*. 2019. DOI: 10.48550/ARXIV.1901.07636. URL: https://arxiv.org/abs/1901.07636.

[77] Erica Grant and Travis S Humble. "Benchmarking embedded chain breaking in quantum annealing". In: *Quantum Science and Technology* 7.2 (2022), p. 025029.

[78] Jeffrey Marshall, Gianni Mossi, and Eleanor G. Rieffel. "Perils of embedding for quantum sampling". In: *Physical Review A* 105.2 (2022). DOI: 10.1103/physreva.105.022615. URL: https://doi.org/10.1103%2Fphysreva.105.022615.

[79] C. H. Tseng et al. "Quantum simulation of a three-body-interaction Hamiltonian on an NMR quantum computer". In: *Phys. Rev. A* 61 (1 1999), p. 012302. DOI: 10.1103/PhysRevA.61.012302. URL: https://link.aps.org/doi/10.1103/PhysRevA.61.012302.

[80] Nicholas Chancellor, Stefan Zohren, and Paul A Warburton. "Circuit design for multi-body interactions in superconducting quantum annealing systems with applications to a scalable architecture". In: *npj Quantum Information* 3.1 (2017), pp. 1–7. DOI: 10.1038/s41534-017-0022-6.

[81] Colin Campbell and Edward Dahl. "QAOA of the Highest Order". In: *2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C)*. 2022, pp. 141–146. DOI: 10.1109/ICSA-C54293.2022.00035.

[82] Joao Basso et al. "Performance and limitations of the QAOA at constant levels on large sparse hypergraphs and spin glass models". In: *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*. 2022, pp. 335–343. DOI: 10.1109/FOCS54457.2022.00039.

[83] RAMIN Fakhimi et al. *Quantum-inspired Formulations for the Max k-cut Problem*. Tech. rep. ISE Technical Report 21T-007 Lehigh University, 2021.

[84] Adam Glos, Aleksandra Krawiec, and Zoltán Zimborás. *Space-efficient binary optimization for variational computing*. 2020. arXiv: 2009.07309 [quant-ph].

[85] Zsolt Tabi et al. "Quantum Optimization for the Graph Coloring Problem with Space-Efficient Embedding". In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2020. DOI: 10.1109/qce49297.2020.00018. URL: https://doi.org/10.1109%2Fqce49297.2020.00018.

[86] Elisabetta Valiante et al. "Computational overhead of locality reduction in binary optimization problems". In: *Computer Physics Communications* 269 (2021), p. 108102. ISSN: 0010-4655. DOI: https://doi.org/10.1016/j.cpc.2021.108102. URL: https://www.sciencedirect.com/science/article/pii/S0010465521002149.

[87] Hiroshi Ishikawa. "Transformation of General Binary MRF Minimization to the First-Order Case". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.6 (2011), pp. 1234–1249. DOI: 10.1109/TPAMI.2010.91.

[88] Elijah Pelofske et al. "Quantum annealing algorithms for Boolean tensor networks". In: *Scientific Reports* 12.1 (2022). DOI: 10.1038/s41598-022-12611-9. URL: https://doi.org/10.1038%2Fs41598-022-12611-9.

[89] Elijah Pelofske et al. "Boolean hierarchical tucker networks on quantum annealers". In: *Large-Scale Scientific Computing: 13th International Conference, LSSC 2021, Sozopol, Bulgaria, June 7–11, 2021, Revised Selected Papers*. Springer. 2022, pp. 351–358. DOI: 10.1007/978-3-030-97549-4_40.

[90] Kyungtaek Jun and Hyunju Lee. "HUBO formulations for solving the eigenvalue problem". In: *Results in Control and Optimization* 11 (2023), p. 100222. DOI: 10.1016/j.rico.2023.100222.

[91] 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.

[92] 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.

[93] Andreas Bärtschi and Stephan Eidenbenz. "Grover Mixers for QAOA: Shifting Complexity from Mixer Design to State Preparation". In: *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2020, pp. 72–82. DOI: 10.1109/QCE49297.2020.00020.

[94] Jonathan Wurtz and Peter J. Love. "Classically Optimal Variational Quantum Algorithms". In: *IEEE Transactions on Quantum Engineering* 2 (2021), pp. 1–7. DOI: 10.1109/TQE.2021.3122568.

[95] Jonathan Wurtz and Peter J. Love. "Counterdiabaticity and the quantum approximate optimization algorithm". In: *Quantum* 6 (2022), p. 635. DOI: 10.22331/q-2022-01-27-635. URL: https://doi.org/10.22331%2Fq-2022-01-27-635.

[96] John Golden et al. "Threshold-Based Quantum Optimization". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 137–147. DOI: 10.1109/QCE52317.2021.00030.

[97] Sergey Bravyi et al. "Obstacles to Variational Quantum Optimization from Symmetry Protection". In: *Physical Review Letters* 125.26 (2020). DOI: 10.1103/physrevlett.125.260505. URL: https://doi.org/10.1103%2Fphysrevlett.125.260505.

[98] Daniel J. Egger, Jakub Mareč ek, and Stefan Woerner. "Warm-starting quantum optimization". In: *Quantum* 5 (2021), p. 479. DOI: 10.22331/q-2021-06-17-479. URL: https://doi.org/10.22331%2Fq-2021-06-17-479.

[99] Reuben Tate et al. *Bridging Classical and Quantum with SDP initialized warm-starts for QAOA*. 2020. DOI: 10.48550/ARXIV.2010.14021. URL: https://arxiv.org/abs/2010.14021.

[100] Madelyn Cain et al. *The QAOA gets stuck starting from a good classical string*. 2022. DOI: 10.48550/ARXIV.2207.05089. URL: https://arxiv.org/abs/2207.05089.

[101] Daniel Beaulieu and Anh Pham. *Max-cut Clustering Utilizing Warm-Start QAOA and IBM Runtime*. 2021. DOI: 10.48550/ARXIV.2108.13464. URL: https://arxiv.org/abs/2108.13464.

[102] Bence Bakó et al. *Near-optimal circuit design for variational quantum optimization*. 2022. DOI: 10.48550/ARXIV.2209.03386. URL: https://arxiv.org/abs/2209.03386.

[103] Takuya Yoshioka et al. *Fermionic Quantum Approximate Optimization Algorithm*. 2023. DOI: 10.48550/ARXIV.2301.10756. URL: https://arxiv.org/abs/2301.10756.

[104] Libor Caha, Alexander Kliesch, and Robert Koenig. "Twisted hybrid algorithms for combinatorial optimization". In: *Quantum Science and Technology* 7.4 (2022), p. 045013. DOI: 10.1088/2058-9565/ac7f4f. URL: https://doi.org/10.1088%2F2058-9565%2Fac7f4f.

[105] Junde Li, Mahabubul Alam, and Swaroop Ghosh. "Large-scale Quantum Approximate Optimization via Divide-and-Conquer". In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* (2022), pp. 1–1. DOI: 10.1109/TCAD.2022.3212196.

[106] Rebekah Herrman et al. "Multi-angle quantum approximate optimization algorithm". In: *Scientific Reports* 12.1 (2022), pp. 1–10. DOI: 10.1038/s41598-022-10555-8.

[107] Kaiyan Shi et al. *Multi-Angle QAOA Does Not Always Need All Its Angles*. 2022. DOI: 10.48550/ARXIV.2209.11839. URL: https://arxiv.org/abs/2209.11839.

[108] M. Cerezo et al. "Variational quantum algorithms". In: *Nature Reviews Physics* 3.9 (2021), pp. 625–644. DOI: 10.1038/s42254-021-00348-9. URL: https://doi.org/10.1038%2Fs42254-021-00348-9.

[109] Samson Wang et al. "Noise-induced barren plateaus in variational quantum algorithms". In: *Nature communications* 12.1 (2021), pp. 1–11. DOI: 10.1038/s41467-021-27045-6.

[110] Yingyue Zhu et al. "Multi-round QAOA and advanced mixers on a trapped-ion quantum computer". In: *Quantum Science and Technology* 8.1 (2022), p. 015007. DOI: 10.1088/2058-9565/ac91ef. URL: https://dx.doi.org/10.1088/2058-9565/ac91ef.

[111] Lorenzo Leone et al. *On the practical usefulness of the Hardware Efficient Ansatz*. 2022. arXiv: 2211.01477 [quant-ph].

[112] Kouhei Nakaji and Naoki Yamamoto. "Expressibility of the alternating layered ansatz for quantum computation". In: *Quantum* 5 (Apr. 2021), p. 434. ISSN: 2521-327X. DOI: 10.22331/q-2021-04-19-434. URL: https://doi.org/10.22331/q-2021-04-19-434.

[113] Abhinav Kandala et al. "Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets". In: *Nature* 549.7671 (2017), pp. 242–246. DOI: 10.1038/nature23879. URL: https://doi.org/10.1038%2Fnature23879.

[114] Lorenza Viola and Seth Lloyd. "Dynamical suppression of decoherence in two-state quantum systems". In: *Physical Review A* 58.4 (1998), pp. 2733–2744. DOI: 10.1103/physreva.58.2733. URL: https://doi.org/10.1103%2Fphysreva.58.2733.

[115] Dieter Suter and Gonzalo A. Álvarez. "Colloquium: Protecting quantum information against environmental noise". In: *Rev. Mod. Phys.* 88 (4 2016), p. 041001. DOI: 10.1103/RevModPhys.88.041001. URL: https://link.aps.org/doi/10.1103/RevModPhys.88.041001.

[116] Lorenza Viola, Emanuel Knill, and Seth Lloyd. "Dynamical Decoupling of Open Quantum Systems". In: *Phys. Rev. Lett.* 82 (12 1999), pp. 2417–2421. DOI: 10.1103/PhysRevLett.82.2417. URL: https://link.aps.org/doi/10.1103/PhysRevLett.82.2417.

[117] Mustafa Ahmed Ali Ahmed, Gonzalo A. Álvarez, and Dieter Suter. "Robustness of dynamical decoupling sequences". In: *Physical Review A* 87.4 (2013). DOI: 10.1103/physreva.87.042309. URL: https://doi.org/10.1103%2Fphysreva.87.042309.

[118] Ryan LaRose et al. "Mitiq: A software package for error mitigation on noisy quantum computers". In: *Quantum* 6 (2022), p. 774. DOI: 10.22331/q-2022-08-11-774. URL: https://doi.org/10.22331%2Fq-2022-08-11-774.

[119] Clement Charles et al. *Simulating $\mathbb{Z}_2$ lattice gauge theory on a quantum computer*. arXiv:2305.02361v1. 2023. eprint: 2305.02361.

[120] Siyuan Niu and Aida Todri-Sanial. "Effects of Dynamical Decoupling and Pulse-Level Optimizations on IBM Quantum Computers". In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–10. DOI: 10.1109/tqe.2022.3203153. URL: https://doi.org/10.1109%2Ftqe.2022.3203153.

[121] Nic Ezzell et al. *Dynamical decoupling for superconducting qubits: a performance survey*. 2022. DOI: 10.48550/ARXIV.2207.03670. URL: https://arxiv.org/abs/2207.03670.

[122] Bibek Pokharel et al. "Demonstration of Fidelity Improvement Using Dynamical Decoupling with Superconducting Qubits". In: *Phys. Rev. Lett.* 121 (22 2018), p. 220502. DOI: 10.1103/PhysRevLett.121.220502. URL: https://link.aps.org/doi/10.1103/PhysRevLett.121.220502.

[123] Bibek Pokharel and Daniel Lidar. *Better-than-classical Grover search via quantum error detection and suppression*. 2022. DOI: 10.48550/ARXIV.2211.04543. URL: https://arxiv.org/abs/2211.04543.

[124] Youngseok Kim et al. "Scalable error mitigation for noisy quantum circuits produces competitive expectation values". In: *Nature Physics* (2023). DOI: 10.1038/s41567-022-01914-3. URL: https://doi.org/10.1038%2Fs41567-022-01914-3.

[125] Petar Jurcevic et al. "Demonstration of quantum volume 64 on a superconducting quantum computing system". In: *Quantum Science and Technology* 6.2 (2021), p. 025020. DOI: 10.1088/2058-9565/abe519.

[126] Andrew W. Cross et al. *Open Quantum Assembly Language*. 2017. DOI: 10.48550/ARXIV.1707.03429. URL: https://arxiv.org/abs/1707.03429.

[127] Filip B. Maciejewski, Zoltá n Zimborás, and Michał Oszmaniec. "Mitigation of readout noise in near-term quantum devices by classical post-processing based on detector tomography". In: *Quantum* 4 (2020), p. 257. DOI: 10.22331/q-2020-04-24-257. URL: https://doi.org/10.22331%2Fq-2020-04-24-257.

[128] Ying Li and Simon C. Benjamin. "Efficient Variational Quantum Simulator Incorporating Active Error Minimization". In: *Phys. Rev. X* 7 (2 2017), p. 021050. DOI: 10.1103/PhysRevX.7.021050. URL: https://link.aps.org/doi/10.1103/PhysRevX.7.021050.

[129] Abhinav Kandala et al. "Error mitigation extends the computational reach of a noisy quantum processor". In: *Nature* 567.7749 (2019), pp. 491–495. DOI: 10.1038/s41586-019-1040-7.

[130] Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. "Error Mitigation for Short-Depth Quantum Circuits". In: *Phys. Rev. Lett.* 119 (18 2017), p. 180509. DOI: 10.1103/PhysRevLett.119.180509. URL: https://link.aps.org/doi/10.1103/PhysRevLett.119.180509.

[131] Sergey Bravyi et al. "Simulation of quantum circuits by low-rank stabilizer decompositions". In: *Quantum* 3 (2019), p. 181. DOI: 10.22331/q-2019-09-02-181. URL: https://doi.org/10.22331%2Fq-2019-09-02-181.

[132] Elijah Pelofske, Georg Hahn, and Hristo N. Djidjev. "Parallel quantum annealing". In: *Scientific Reports* 12.1 (2022). DOI: 10.1038/s41598-022-08394-8. URL: https://doi.org/10.1038%2Fs41598-022-08394-8.

[133] Elijah Pelofske, Georg Hahn, and Hristo N. Djidjev. "Solving larger maximum clique problems using parallel quantum annealing". In: *Quantum Information Processing* 22.5 (2023). DOI: 10.1007/s11128-023-03962-x. URL: https://doi.org/10.1007%2Fs11128-023-03962-x.

[134] Abdullah Ash-Saki, Mahabubul Alam, and Swaroop Ghosh. "Analysis of Crosstalk in NISQ Devices and Security Implications in Multi-Programming Regime". In: *Proceedings of the ACM/IEEE International Symposium on Low Power Electronics and Design*. ISLPED '20. Boston, Massachusetts: Association for Computing Machinery, 2020, 25–30. ISBN: 9781450370530. DOI: 10.1145/3370748.3406570. URL: https://doi.org/10.1145/3370748.3406570.

[135] Poulami Das et al. "A Case for Multi-Programming Quantum Computers". In: *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. MICRO '52. Columbus, OH, USA: Association for Computing Machinery, 2019, 291–303. ISBN: 9781450369381. DOI: 10.1145/3352460.3358287. URL: https://doi.org/10.1145/3352460.3358287.

[136] Siyuan Niu and Aida Todri-Sanial. *Multi-programming Cross Platform Benchmarking for Quantum Computing Hardware*. 2022. DOI: 10.48550/ARXIV.2206.03144. URL: https://arxiv.org/abs/2206.03144.

[137] Yasuhiro Ohkura, Takahiko Satoh, and Rodney Van Meter. "Simultaneous Execution of Quantum Circuits on Current and Near-Future NISQ Systems". In: *IEEE Transactions on Quantum Engineering* 3 (2022), pp. 1–10. DOI: 10.1109/TQE.2022.3164716.

[138] Siyuan Niu and Aida Todri-Sanial. "How Parallel Circuit Execution Can Be Useful for NISQ Computing?" In: *Proceedings of the 2022 Conference and Exhibition on Design, Automation & Test in Europe*. DATE '22. Antwerp, Belgium: European Design and Automation Association, 2022, 1065–1070. ISBN: 9783981926361.

[139] Lana Mineh and Ashley Montanaro. *Accelerating the variational quantum eigensolver using parallelism*. 2022. DOI: 10.48550/ARXIV.2209.03796. URL: https://arxiv.org/abs/2209.03796.

[140] Salonik Resch et al. *Accelerating Variational Quantum Algorithms Using Circuit Concurrency*. 2021. DOI: 10.48550/ARXIV.2109.01714. URL: https://arxiv.org/abs/2109.01714.

[141] Jeffrey Marshall et al. "Power of Pausing: Advancing Understanding of Thermalization in Experimental Quantum Annealers". In: *Phys. Rev. Appl.* 11 (4 2019), p. 044083. DOI: 10.1103/PhysRevApplied.11.044083. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.11.044083.

[142] G. Passarelli, V. Cataudella, and P. Lucignano. "Improving quantum annealing of the ferromagnetic p-spin model through pausing". In: *Phys. Rev. B* 100 (2 2019), p. 024302. DOI: 10.1103/PhysRevB.100.024302. URL: https://link.aps.org/doi/10.1103/PhysRevB.100.024302.

[143] Huo Chen and Daniel A. Lidar. "Why and When Pausing is Beneficial in Quantum Annealing". In: *Phys. Rev. Appl.* 14 (1 2020), p. 014100. DOI: 10.1103/PhysRevApplied.14.014100. URL: https://link.aps.org/doi/10.1103/PhysRevApplied.14.014100.

[144] Zoe Gonzalez Izquierdo et al. "Advantage of Pausing: Parameter Setting for Quantum Annealers". In: *Physical Review Applied* 18.5 (2022). DOI: 10.1103/physrevapplied.18.054056. URL: https://doi.org/10.1103%2Fphysrevapplied.18.054056.

[145] IBM ILOG Cplex. "V12.10.0 : User's Manual for CPLEX". In: *International Business Machines Corporation* 46.53 (2019), p. 157.

[146] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. "Optimization by simulated annealing". In: *science* 220.4598 (1983), pp. 671–680. DOI: 10.1126/science.220.4598.671.

[147] Xinwei Lee et al. "Parameters Fixing Strategy for Quantum Approximate Optimization Algorithm". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 10–16. DOI: 10.1109/QCE52317.2021.00016.

[148] Alexey Galda et al. "Transferability of optimal QAOA parameters between random graphs". In: *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. 2021, pp. 171–180. DOI: 10.1109/QCE52317.2021.00034.

[149] V. Akshay et al. "Parameter concentrations in quantum approximate optimization". In: *Phys. Rev. A* 104 (1 2021), p. L010401. DOI: 10.1103/PhysRevA.104.L010401. URL: https://link.aps.org/doi/10.1103/PhysRevA.104.L010401.

[150] Jonathan Wurtz and Danylo Lykov. *The fixed angle conjecture for QAOA on regular MaxCut graphs*. 2021. DOI: 10.48550/ARXIV.2107.00677. URL: https://arxiv.org/abs/2107.00677.

[151] David C. McKay et al. “Efficient Z-Gates for Quantum Computing”. In: *Physical Review A* 96.2 (2017). DOI: 10.1103/physreva.96.022330. URL: https://doi.org/10.1103%2Fphysreva.96.022330.