FrozenQubits: Boosting Fidelity of QAOA by Skipping Hotspot Nodes

Ramin Ayanzadeh*
Georgia Tech
Atlanta, USA

Narges Alavisamani
Georgia Tech
Atlanta, USA

Poulami Das
Georgia Tech
Atlanta, USA

Moinuddin Qureshi
Georgia Tech
Atlanta, USA

ABSTRACT

Quantum Approximate Optimization Algorithm (QAOA) is one of the leading candidates for demonstrating the quantum advantage using near-term quantum computers. Unfortunately, high device error rates limit us from reliably running QAOA circuits for problems with more than a few qubits. In QAOA, the problem graph is translated into a quantum circuit such that every edge corresponds to two 2-qubit CNOT operations in each layer of the circuit. As CNOTs are extremely error-prone, the fidelity of QAOA circuits is dictated by the number of edges in the problem graph.

We observe that the majority of graphs corresponding to real-world applications follow a "power-law" distribution, where some hotspot nodes have significantly higher number of connections. We leverage this insight and propose "FrozenQubits" that freezes the hotspot nodes or qubits and intelligently partitions the state-space of the given problem into several smaller sub-spaces, which are then solved independently. The corresponding QAOA sub-circuits are significantly less vulnerable to gate and decoherence errors due to the reduced number of CNOT operations in each sub-circuit. Unlike prior circuit-cutting approaches, FrozenQubits does not require any exponentially complex postprocessing step. Our evaluations with 5,300 QAOA circuits on eight different quantum computers from IBM show that FrozenQubits can improve the quality of solutions by 8.73x on average (and by up to 57x), albeit while utilizing 2x more quantum resources.

CCS CONCEPTS

• **Computer systems organization → Quantum computing**.

KEYWORDS

NISQ, QAOA, Quantum Computing

ACM Reference Format:
Ramin Ayanzadeh, Narges Alavisamani, Poulami Das, and Moinuddin Qureshi. 2023. FrozenQubits: Boosting Fidelity of QAOA by Skipping Hotspot Nodes . In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '23), March 25–29, 2023, Vancouver, BC, Canada.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3575693.3575741

*The corresponding author can be reached at ayanzadeh@gatech.edu.

ASPLOS '23, March 25–29, 2023, Vancouver, BC, Canada

ACM ISBN 978-1-4503-9916-6/23/03.
https://doi.org/10.1145/3575693.3575741

1 INTRODUCTION

Near-term quantum computers with few dozens of noisy qubits can already outperform supercomputers for certain tasks [11, 111]. Soon, we expect these Noisy Intermediate Scale Quantum (NISQ) [92] devices to provide computational advantages for real-world applications such as estimating the ground state energy of molecules [83, 92] and discrete optimizations [48]. Unfortunately, NISQ programs are vulnerable to errors due to noise and their fidelity is low. To leverage NISQ systems for practical problems, the application fidelity must be increased.

Quantum approximate optimization algorithm (QAOA) [24, 48] is regarded as a leading candidate for demonstration of quantum advantage on NISQ devices. It approximates the ground state or a configuration with the lowest energy value of a physical system, called *Hamiltonian*. QAOA promises computational advantages for various industry-scale applications [33, 35, 38, 104, 110], where the objective is to minimize or maximize a cost function. Solving a problem using QAOA is a two-step process, as shown in Figure 1(a). First, the problem is mapped into a p-layer parametric quantum circuit with $2p$ parameters and executed for thousands of trials. Next, the expectation value of the output distribution is used by a classical optimizer to adjust the parameters. The process is repeated until the optimal parameters of the circuit have been found. Unfortunately, noisy devices limit us from running QAOA problems of practical scale. For example, despite various optimizations for reducing the impact of hardware errors, we are unable to solve QAOA problems on 3-regular graphs with more than twenty-three qubits on state-of-the-art NISQ systems such as Google Sycamore [59].

The number of two-qubit CNOT operations in a QAOA circuit increases linearly with the number of edges in the problem graph, as each edge typically corresponds to two CNOTs, as shown in Figure 1(a). CNOT operations are highly error-prone and incur long latencies. For example, CNOTs have an average error-rate of 1% on Google Sycamore [5] and take 400ns on average on IBM devices (10x slower than single-qubit gates). The problem compounds as most NISQ devices have limited connectivity and compilers are forced to introduce SWAPs that further increase the number of CNOTs required and depth of circuits. Post-compilation, the number of CNOTs scales super-polynomial with the number of edges in the problem graph. The high error-rates of CNOT operations and inability of qubits to retain information beyond a few micro-seconds on near-term systems cause the infeasability of practical QAOA problems [57] to become vanishingly small as a result of large CNOT count and circuit depth.

We observe that although most QAOA studies focus on 3-regular or fully-connected graphs, real-world problems follow *power-law (or Pareto) distribution* [4, 34, 52, 53, 62, 78, 86]. This means only a small number of nodes have much higher connectivity than others.

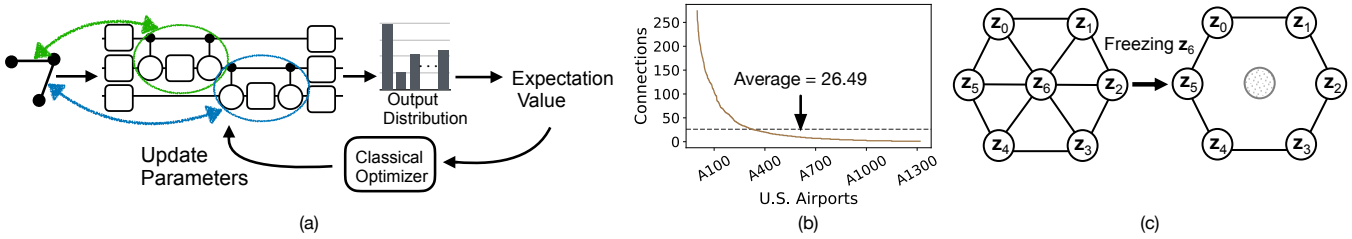


Figure 1: (a) Problem solving using QAOA. (b) Node degree of the U.S. airport traffic graph follows the power-law distribution where Hotspots or hubs have significantly higher connectivity than the average. (c) Freezing z_6 (the hotspot node) drops 6 edges.

For example, Figure 1(b) shows the number of connections for 1300 U.S. airports. We observe that some airports (known as hotspots or hubs) have a significantly larger number of connections or edges than others. The ten busiest airports have 10x connections compared to the average connectivity of all airports. From the QAOA perspective, these nodes with significantly higher connectivity than the average contribute to a greater number of CNOT operations in the parametric circuit than others. We leverage this insight and propose *FrozenQubits* to improve the fidelity of QAOA problems.

In FrozenQubits, we freeze some of the nodes with the highest degree of connectivity. This drops the edges connected to these nodes resulting in a circuit with fewer CNOTs and lower depth. We explain our design using the example shown in Figure 1(c). Here, node $\mathbf{z}_6$ is connected to six other nodes and contributes to 12 CNOTs in the original QAOA circuit. Note that this is under the assumption that the device topology is such that the compiler does not need to introduce any SWAPs. On real NISQ machines with limited connectivity, however, the CNOT count and depth will be even higher. The probability of introducing SWAPs for a node increases with the number of connections. Therefore, for QAOA circuits, hotspot nodes with significantly higher connectivity typically tend to have larger SWAP overheads compared to nodes with lower connectivity. Our proposed solution freezes node $\mathbf{z}_6$ and creates two sub-circuits, resulting in 50% reduction in the number of CNOTs in each sub-circuit. This also eliminates any SWAPs that would have been otherwise required for the CNOTs corresponding to the edges for node $\mathbf{z}_6$. The reduction in SWAP overheads will depend on the topology of the target device.

In FrozenQubits, we identify a hotspot node (which corresponds to qubits with the highest number of CNOTs), freeze it by substituting the value of each qubit with its two possibilities, -1 and +1. This partitions the state-space of the given problem into two smaller problems whose corresponding QAOA circuits are significantly less vulnerable to errors, and solve the resultant sub-circuits independently. In general, freezing m qubits will result in 2^m circuits. Note that, usually problem graphs have only a few hotspots, as shown in Figure 1(b). Thus, a small value of m is sufficient and our default design uses $m = 1, 2$. The challenge with FrozenQubits is that we do not know which sub-circuit includes the global optimum and therefore, we need to run all 2^m sub-circuits to find the solution. However, we observe that majority of these sub-circuits are pairwise symmetric whereby flipping all bits of any point in one sub-space is a point in the other sub-space with an identical cost. We leverage this insight to reduce the complexity of FrozenQubits by only solving one of these pairwise symmetric sub-circuits and then inferring the result of the other sub-circuit. For example, freezing $\mathbf{z}_6$ in Figure 1(c) results in two smaller circuits, one for $\mathbf{z}_6 = +1$ and one for $\mathbf{z}_6 = -1$, but we only evaluate one of them and postprocess its output distribution to evaluate the other circuit.

We note that FrozenQubits is significantly different than circuit cutting. For example, with CutQC [107], a circuit is decomposed into 2^c sub-circuits by cutting c edges from the circuit graph and the sub-circuits are executed separately. The output distribution is obtained using tensor products. While this approach is promising, it is not applicable to practical QAOA circuits because: (1) for real-world graphs, c must be greater than the number of hotspots, which is impractical as the number of sub-circuits and complexity of the post-processing step grow exponentially with c, (2) partitioning multi-layer QAOA circuits via cutting wires is nontrivial, and (3) exponentially complex post-processing can bottleneck the tuning of circuit parameters. On the contrary, FrozenQubits does not require evaluation of all the sub-circuits or involve any post-processing step that incurs exponential complexity. Similarly, edge cutting has been proposed for partitioning large graphs into smaller sub-graphs and running the sub-graphs on smaller quantum computers [71]. However, edge-cutting power-law graphs is nontrivial as hotspot nodes appear in all sub-graphs that degrades the accuracy of the output distribution estimation. Instead, we take an orthogonal approach in this paper to simplify the search space of QAOA.

Our evaluations with over 5,300 circuits on eight different real quantum computers from IBM show that FrozenQubits can obtain by 8.73x on average (and up to 57.14x) improvement in the quality of solutions for QAOA circuits compared to the baseline, albeit running 2x more circuits. FrozenQubits data and code are publicly available at https://doi.org/10.5281/zenodo.7278397.

Overall, this paper makes the following contributions:

(1) We propose a novel quantum divide-and-conquer approach, called *FrozenQubits*, that leverages the insight about most real-world applications having power-law graphs to boost the fidelity of QAOA applications;
(2) We observe that the search space of most QAOA problems have symmetric sub-spaces whereby flipping all bits of any point in one sub-space is a point in the other sub-space with an identical cost value;
(3) We leverage the symmetricity of the search space of most QAOA problems to subside the complexity of our design.

2 BACKGROUND

2.1 Quantum Approximate Optimization Algorithm

Quantum approximate optimization algorithm (QAOA) [24, 48, 49] is widely recognised as one of the leading candidates for demonstrating quantum advantage in the near-term. QAOA promises computational speed-up for optimization problems in various application domains. Solving real-world optimization problems using QAOA is a two-step process. First, representing the optimization problem in the form of Equation (1), which is known as ***Ising Hamiltonian***, where $\mathbf{z}_i \in \{-1, +1\}$ and $\mathbf{h}_i, J_{ij}, \mathit{offset} \in \mathbb{R}$ are given coefficients of the problem [12–17, 39, 59, 74, 85, 96, 100, 105].

$$H_Z := C(\mathbf{z}) = \sum_{i=0}^{N-1} \mathbf{h}_i \mathbf{z}_i + \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} J_{ij} \mathbf{z}_i \mathbf{z}_j + \text{offset} \qquad (1)$$

Second, tuning QAOA circuit parameters to find the optimum value of Equation (1). QAOA promises speed-up for many applications by accelerating this step, which is computationally expensive on conventional computers [48, 49].

For example, to represent the Max-Cut problem using Ising Hamiltonian, we add the $\mathbf{z}_i\mathbf{z}_j$ term to the Ising Hamiltonian for every edge between nodes i and j and use J_{ij} to indicate the weight of the edge between node i and j, as shown in Figure 2(a). If running the QAOA algorithm and finding optimum value yield $\mathbf{z}_i\mathbf{z}_j = -1$, it means that nodes i and j are in two separate cuts in the QAOA solution for that Max-Cut problem. Alternately, $\mathbf{z}_i\mathbf{z}_j = 1$ implies the nodes belong to the same cut. QAOA is computationally universal [73, 79]and can be used for tackling a wide range of real-world applications. However, formulating practical problems as minimizing Equation (1) in general is nontrivial [12–17, 74].

Solving Equation (1) on a quantum computer requires a QAOA circuit with (1) N qubits, (2) one single-qubit gate for each $\mathbf{z}_i$, and (3) two two-qubit gates, in addition to one single-qubit gate, for each $\mathbf{z}_i\mathbf{z}_j$ term. This comprises a single layer in the circuit. Thus, the number of single and two-qubit operations in a QAOA circuit scales with the number of nodes and edges in the problem graph, respectively. For example, Figure 2(b) shows the QAOA circuit for an Ising Hamiltonian that represents the graph in Figure 2(a). QAOA circuits can consist of multiple layers (p). In a QAOA circuit, $\mathbf{z}_i$ indicates the result of measuring qubit i. In quantum computing, when one measures qubits $|0\rangle$ and $|1\rangle$ on $\mathbf{z}$ basis, the outcomes are the eigenvalues of operator $\mathbf{z}$, which are +1 and −1, respectively.

QAOA circuits are parametric circuits, as shown in Figure 2(b), and involves searching for the optimal values of the parameters γ_1 and β_1 (which corresponds to the angles of the single-qubit rotation gates) using a classical optimizer. QAOA applications are run in a variational mode where the output distributions of the circuits are used by the classical optimizer to adjust the next round of parameters and this training process continues until the optimizer converges. The solution of the problem is inferred from the output distribution of the QAOA circuit using the optimal parameters. Demonstrating quantum advantage—outperforming the state-of-the-art classical optimization techniques—at a practical scale requires running QAOA circuits with (at least) some hundreds of qubits [57] and several layers [24].

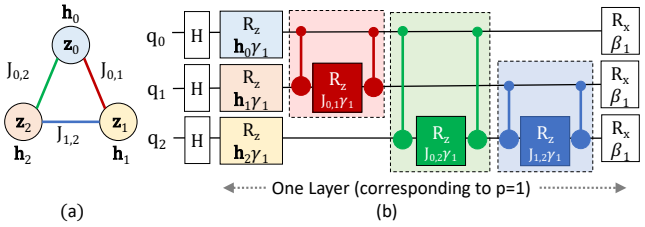


Figure 2: A QAOA example: (a) graph representation of a three-qubit Hamiltonian; (b) QAOA circuit with one layer.

2.2 Impact of Hardware Errors on QAOA Circuits

Near-term quantum devices are noisy and suffer from high error rates. For example, qubits do not retain their information beyond a few micro-seconds on most existing systems [5], a phenomenon referred to as *decoherence*. Also, imperfections in quantum gate and measurement operations cause programs to encounter computational errors when programs are executed on NISQ devices. In particular, CNOT operations are the dominant sources of errors on most existing NISQ systems with an average error-rate of about 1% [5]. They also incur long latencies (about 400 ns on existing IBMQ systems) and therefore, increases the probability of decoherence. This limits the fidelity of QAOA circuits as the problem size grows due to an increase in the number of two-qubit operations.

NISQ devices also suffer from limited connectivity where qubits are only connected to some of their nearest neighbors. To execute CNOT operations between unconnected qubits, NISQ compilers introduce SWAP operations. A SWAP is a sequence of three CNOT operations that interchanges the state of two qubits. Thus, they enable compilers to overcome the limited connectivity of NISQ devices by routing non-adjacent qubits next to each other. Unfortunately, SWAPs increase the total number of CNOT operations and depth of the circuits, making them even more vulnerable to errors. For example, Figure 3 shows that the number of CNOTs of a compiled program increases up-to 14X even for small programs. The problem compounds when QAOA circuits with multiple layers must be executed to accurately estimate the solution of a problem [24] as the number of CNOT operations and depth increase, exacerbating the impact of hardware errors [44, 58, 59].

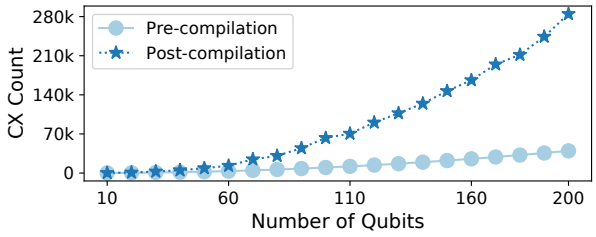


Figure 3: Impact of SWAPs on fully-connected QAOA graphs on a grid qubit architecture.

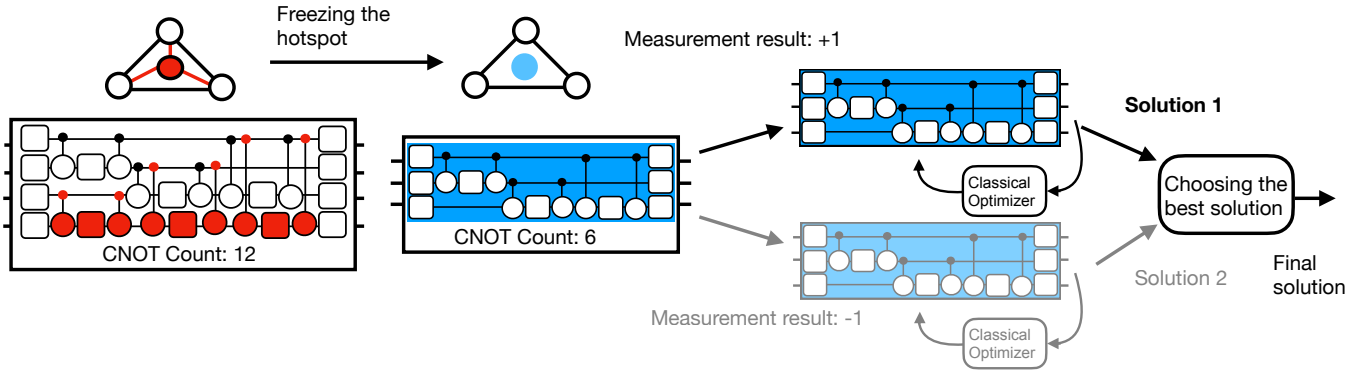


Figure 4: Overview of the FrozenQubits framework when one qubit is frozen. After freezing the hotspot qubit, we have two sub-problems with smaller quantum circuits. Due to the symmetricity of the search space of most QAOA problems, we can skip half of the sub-problems and find the best solution for only the remaining half sub-problems; see section 3.7.2.

3 FROZENQUBITS: FREEZE THE HOTSPOTS

We present *FrozenQubits*, an application-level software framework, to increase the fidelity of QAOA applications by *freezing* or skipping hotspot nodes and solving sub-problems with smaller more reliable quantum circuits. In this section, we provide our insight and explain the design of FrozenQubits.

3.1 Insight and Goal: Leverage Application Characteristics to Improve Fidelity of QAOA Circuits

The goal of this work is to decrease the impact of two-qubit gate errors, which are the main source of errors in QAOA circuits run on near-term quantum computers. To achieve this goal, we design a software framework that reduces the number of two-qubit gates of the QAOA circuit by leveraging application-level properties of real-world problem graphs.

Insight: In real-world problem graphs, not all nodes contribute equally to the QAOA circuit.

Power-Law Graphs: In most natural and artificial graphs, the degree of connectivity—number of edges connected to a node—of a small number of nodes are much higher than the rest of nodes. In other words, the degree distributions of most real-world application graphs follow ***Power-Law*** distribution [4, 20, 34, 52, 53, 62, 64, 78, 86]. For example, if we consider the graph for friendship relations on a social network, we observe that the majority of people in many friendship interactions have only a few connections to other people, whereas some people, such as influencers, are *hotspots* and are connected to many others in the network [2]. We make similar observations in many other real-world problem graphs. For example, Table 1 shows various power-law graphs in real-world problem domains along with the application of QAOA to solve them on near-term quantum computers.

In QAOA, the number of edges for a node determines the number of two-qubit CNOT gates, contributed by that node to the quantum circuit. Therefore, the hotspots contribute to a significantly larger number of CNOT gates in the QAOA circuits, compared to the other nodes, for real-world applications.

Table 1: Power-Law Graphs for Real-World Problems and The Usage of QAOA to Solve Them.

Domain	Sub-Domain	Power-Law Example	QAOA Application
Transportation	Vehicle Routing	[7, 26, 80]	[18, 25, 51]
	Supply Chain	[61, 106]	[1, 25]
Biology	Protein Folding	[76, 93, 99]	[47, 50, 97]
	DNA Sequences	[31, 37, 90]	[30, 98]
Finance and Economics	Portfolio Optimization	[6, 46, 113]	[19, 22, 27, 45]
	Auctions	[65]	[45]

3.2 Overview of FrozenQubits

The number of two-qubit gates in QAOA circuits scales with the number of edges. In real-world problem graphs, some nodes are hotspots and contribute more to the number of edges. Using these characteristics of graphs, we propose *FrozenQubits*, an application-level software framework that improves the fidelity of QAOA applications. FrozenQubits freezes the hotspot qubits meaning it removes those qubits and their associated two-qubit gates from the circuit. This divides the problem into smaller sub-problems of which the associated quantum circuits are less vulnerable to error. Figure 4 shows an overview of FrozenQubits scheme for freezing one qubit.

3.3 Freezing Qubits and Defining Sub-Problems

When one *freezes* a qubit, it means that the qubit, along with the gates connected to it, are eliminated from the circuit. By freezing the qubit we assume that we measure that qubit in z-basis. There are two possibilities for the result of measurements in z-basis, +1 or −1. For each of these possibilities, we need to define a sub-Hamiltonian. Let qubit k be the one frozen by the FrozenQubits. The two sub-Ising Hamiltonians $H_Z^{\mathbf{z}_k=+1}$ in Equation (2) and $H_Z^{\mathbf{z}_k=-1}$ in Equation

(3) are obtained by substituting $\mathbf{z}_k$ in the original Ising Hamiltonian H_Z with +1 and −1, respectively. Table 2 summarizes the notations used in defining the original Hamiltonian and sub-Hamiltonians.

$$H_Z^{\mathbf{z}_k=+1} = \sum_{\substack{i=0 \\ i\neq k}}^{N-1} \mathbf{h}_i^{\mathbf{z}_k=+1}\mathbf{z}_i + \sum_{\substack{i=0 \\ i\neq k}}^{N-1}\sum_{\substack{j=i+1 \\ j\neq k}}^{N-1} J_{ij}\mathbf{z}_i\mathbf{z}_j + \text{offset}^{\mathbf{z}_k=+1} \quad (2)$$

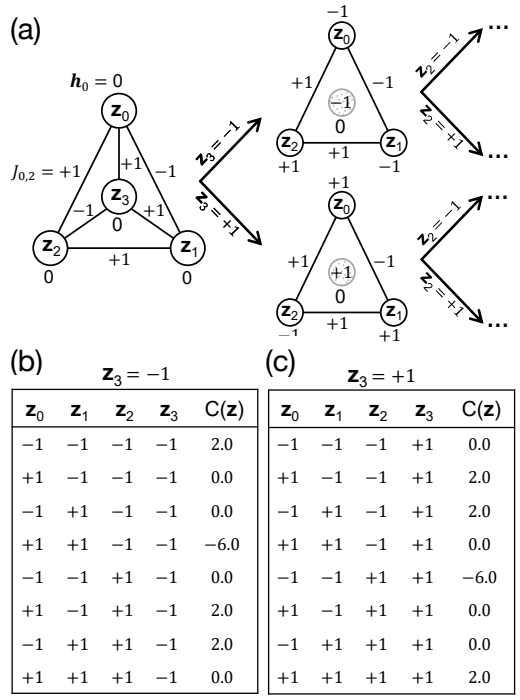
$$H_Z^{\mathbf{z}_k=-1} = \sum_{\substack{i=0 \\ i\neq k}}^{N-1} \mathbf{h}_i^{\mathbf{z}_k=-1}\mathbf{z}_i + \sum_{\substack{i=0 \\ i\neq k}}^{N-1}\sum_{\substack{j=i+1 \\ j\neq k}}^{N-1} J_{ij}\mathbf{z}_i\mathbf{z}_j + \text{offset}^{\mathbf{z}_k=-1} \quad (3)$$

As shown in Equations (2) and (3), each sub-problem corresponds to exactly one half of the state-space of H_Z—one sub-problem for substituting $\mathbf{z}_k$ with +1 and another sub-problem for substituting $\mathbf{z}_k$ with -1. Repeating the substituting process for remaining qubits on the resulting sub-problems will partition the state-space of H_Z into much smaller sub-spaces. More specifically, freezing m qubits will partition the state-space of H_Z into 2^m sub-spaces, and any of the resulting sub-problems will have $N-m$ variables and accordingly their associated sub-circuit will have $N-m$ qubits.

From a graph representation viewpoint, substituting $\mathbf{z}_k$ drops all edges that are connected to $\mathbf{z}_k$. Figure 5(a) illustrates the process of freezing qubits using the graph representation of an example problem with four qubits. Substituting $\mathbf{z}_3$ with +1 and -1 results in two sub-problems with three qubits. Figure 5(b) shows the state space of sub-problems when we substitute $\mathbf{z}_3$ with -1 and +1. The union of all sub-spaces is identical to the state-space of the original problem of interest.

Table 2: Notations of Ising Hamiltonians for a Given Problem and Sub-Problems After Freezing

Notation	Definition
H_Z	$\sum_{i=0}^{N-1}\mathbf{h}_i\mathbf{z}_i + \sum_{i=0}^{N-1}\sum_{j=i+1}^{N-1} J_{ij}\mathbf{z}_i\mathbf{z}_j + \mathit{offset}$, shows the Ising Hamiltonian of the original problem
$\mathbf{z}_i$	Result of measuring qubit i in z-basis; $\mathbf{z}_i \in \{-1,+1\}$
J_{ij}	Coefficient of quadratic term $\mathbf{z}_i\mathbf{z}_j$ in H_z. In graph representation, it indicates the weight of the edge between node i and node j.
$\mathbf{h}_i$	Coefficient of linear term $\mathbf{z}_i$ in H_z. In graph representation, it indicates the weight of the node i. In Max-Cut problem, the weight of all nodes are zero, $\mathbf{h}_i = 0, \forall i$.
$H_Z^{\mathbf{z}_k=+1}$	Ising Hamiltonian of a sub-problem for measurement result +1 after freezing qubit k
$H_Z^{\mathbf{z}_k=-1}$	Ising Hamiltonian of a sub-problem for measurement result −1 after freezing qubit k
$\mathbf{h}_i^{\mathbf{z}_k=+1}$	Linear coefficient in $H_Z^{\mathbf{z}_k=+1}$, after freezing qubit k, for node i which is equal to $\mathbf{h}_i + J_{k,i} + J_{i,k}$.
$\mathbf{h}_i^{\mathbf{z}_k=-1}$	Linear coefficient in $H_Z^{\mathbf{z}_k=-1}$, after freezing qubit k, for node i which is equal to $\mathbf{h}_i - J_{k,i} - J_{i,k}$.
$\text{offset}^{\mathbf{z}_k=+1}$	Offset of $H_Z^{\mathbf{z}_k=+1}$, after freezing qubit k, which is equal to $\mathit{offset} + \mathbf{h}_k$
$\text{offset}^{\mathbf{z}_k=-1}$	Offset of $H_Z^{\mathbf{z}_k=-1}$, after freezing qubit k, which is equal to $\mathit{offset} - \mathbf{h}_k$



$\mathbf{z}_0$	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	C(z)
−1	−1	−1	−1	2.0
+1	−1	−1	−1	0.0
−1	+1	−1	−1	0.0
+1	+1	−1	−1	−6.0
−1	−1	+1	−1	0.0
+1	−1	+1	−1	2.0
−1	+1	+1	−1	2.0
+1	+1	+1	−1	0.0

$\mathbf{z}_0$	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	C(z)
−1	−1	−1	+1	0.0
+1	−1	−1	+1	2.0
−1	+1	−1	+1	2.0
+1	+1	−1	+1	0.0
−1	−1	+1	+1	−6.0
+1	−1	+1	+1	0.0
−1	+1	+1	+1	0.0
+1	+1	+1	+1	2.0

Figure 5: Example of freezing a qubit for an Ising Hamiltonian with four qubits. Substituting $\mathbf{z}_3$ with +1 and -1 results in two sub-problems with three spin variables in each.

For each sub-Hamiltonian, we need to run the classical optimization step on each associated QAOA circuit. All the sub-Hamiltonians of the original problem have the same quadratic forms. However, they vary in terms of offsets and linear coefficients. Therefore, the general structure of the QAOA circuit for all sub-Hamiltonians is similar and they only differ in terms of angles of the rotation gates. These angles are trainable parameters learned during the optimization of QAOA.

Note that unlike the given problem where all linear coefficients were zero, the resulting sub-problems have non-zero linear coefficients (i.e., $\mathbf{h}_k \neq 0$). As shown in Figure 2, every linear term of H_Z (with a linear coefficient of $\mathbf{h}_k$) corresponds to an R_z gate in each layer of the QAOA circuit. However, R_z gates are software gates and do not impact the fidelity.

3.4 Optimizing Number of Qubits to Freeze: A Fidelity-Cost Trade-off

The performance of FrozenQubits depends on the number of qubits frozen. However, there exists a trade-off between the fidelity improvement and the quantum cost of freezing qubits. While freezing an increased number of qubits allow us to drop a larger number of CNOT operations and design smaller sub-circuits that execute with greater fidelity, the quantum cost of executing the sub-circuits grow exponentially. More specifically, the quantum cost of freezing m qubits is $O(2^m)$.

Finding the optimum number of qubits to freeze is non-trivial. To overcome this challenge, FrozenQubits leverages the insight that

for most real-world applications that follow Power-law distribution, the number of dropped edges per qubit decreases quickly for the hotspots (due to higher connectivity) but the pace of CNOT reduction decreases for additional nodes beyond the hotspots. Thus, freezing only a limited number of nodes is sufficient. We confirm our insights using experiments on real systems and observe that freezing additional nodes beyond a certain point has diminishing returns (we defer the discussion to Section 5.1.3). Thus, FrozenQubits can leverage circuit properties such as CNOT counts and depth to determine the number of qubits to freeze for a given application. As FrozenQubits is a scalable framework, we leave it up to the user to select the number of qubits to freeze. Our default design considers dropping up to two qubits.

3.5 Which Qubits to Freeze?

For a given QAOA problem, FrozenQubits can choose m qubits to freeze from ${}^{N}C_{m}$ possibilities. However, instead of randomly selecting qubits from all possibilities, FrozenQubits selects the m qubits corresponding to the hotspots in the problem graph. The insight is that freezing hotspots in real-world problem graphs allows FrozenQubits to drop the maximum number of CNOT operations in the QAOA circuit. Moreover, hotspots also contribute to a significantly larger number of SWAP operations compared to other nodes. Therefore, freezing hotspots allow FrozenQubits to also reduce the SWAP overheads to a much larger extent compared to other nodes.

3.6 Decoding Outcomes

After finding the optimum value of each sub-problem, we need to find the final solution of the original problem of interest. FrozenQubits partitions the state-space of the input problem into smaller sub-spaces and explores them independently via running multiple smaller QAOA programs. Every sub-problem corresponds to one of the sub-spaces while each includes $N - m$ qubits. Therefore, we can find the solutions with the best objective value, the lowest cost value, by just calculating the minimum of the solutions of the sub-problems. FrozenQubits, in contrast to previous works, has no postprocessing and no cost for finding the final solution after solving the sub-problems, except finding the minimum over the solutions of the sub-problems.

3.7 Tackling the Overheads of FrozenQubits

When we freeze m qubits, we will have 2^m smaller quantum circuits to train. This growth in the number of circuits increases the overheads because 1) we need to compile these circuits for execution on a quantum computer and 2) we need to run all circuits independently and infer their outputs to find the solution for the primary problem of interest. Here, we discuss how we tackle these overheads using the characteristics of sub-problem Hamiltonians.

3.7.1 Reducing the Compilation Overhead: Freezing m qubits results in 2^m separate QAOA sub-problems. To find the solution of each sub-problems, we run optimization steps on their associated QAOA circuits. However, these circuits only vary in terms of angles of rotation gates. This is because all the Hamiltonians of sub-problems have the same terms, and they only differ in terms of different coefficients and offset values. Therefore, we only compile one *template circuit*, and edit the resulting compiled circuit for generating executable circuits for all sub-problems. Editing the compiled circuit means embedding $\mathbf{h}_i$ and J_{ij} into the angles of the corresponding R_z rotations. This approach significantly reduces the compilation overhead of FrozenQubits.

3.7.2 Pruning Sub-Problems: When we freeze a qubit and create two sub-problems by substituting the frozen qubit $\mathbf{z}$ value in the Hamiltonian with -1 and $+1$, the state-space of these two sub-problems can be symmetric. For example in Figure 5(b) and Figure 5(c), values of $C(z)$ is symmetric with respect to $\mathbf{z}$. This means flipping all $\mathbf{z}$ values of any row from Figure 5(b)—i.e., $+1 \rightarrow -1$ and $-1 \rightarrow +1$—corresponds to a row in Figure 5(c), and vice versa. Here we demonstrate that this symmetricity appears in all Hamiltonians with all zero linear coefficients.

When all linear coefficients of an Ising Hamiltonian, shown in Eq. (1), is set to zero ($\mathbf{h}_i = 0$ for $i = 0, 1, \ldots, n-1$) we have

$$C(\mathbf{z}) = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} J_{ij} \mathbf{z}_i \mathbf{z}_j.$$

Note that we have omitted the "offset" term since (as a constant) it does not have any impact on the shape or structure of the problem landscape. Since $\mathbf{z}_i \in \{-1, +1\}$, $C(\mathbf{z})$ can be re-written as

$$C(\mathbf{z}) = \sum_{i=0}^{n-1} \sum_{j=i+1}^{n-1} \pm J_{ij}$$

where the sign of J_{ij} only depends on values of $\mathbf{z}_i$ and $\mathbf{z}_j$. In other words, when both $\mathbf{z}_i$ and $\mathbf{z}_j$ have the same value, their product will be +1. Otherwise, their product will be -1. While flipping all variables will change values of $\mathbf{z}_i$ and $\mathbf{z}_j$ individually, their product value will remain unchanged. Thus,

$$C(\mathbf{z}) = C(-\mathbf{z}).$$

If $\mathbf{z}^*$ is a global minimum of an Ising model with zero linear coefficients, $-\mathbf{z}^*$ is also a global minimum of the same Ising Hamiltonian. Moreover, we can conclude that the number of global minimums in an Ising model with zero linear coefficients is even.

FrozenQubits leverages this symmetricity to mitigate the quantum cost. When a qubit is frozen and all linear coefficients of the parent problem are zero, FrozenQubits executes QAOA steps on just one of the sub-problems. After training circuit parameters for this sub-problem, one flips the $\mathbf{z}$ values of the solution to construct the output distribution of the other sub-problem. This pruning significantly mitigates the quantum cost of FrozenQubits.

3.8 Scalability of FrozenQubits

Let m be the number of qubits to freeze, N be the number of qubits, s be the number of distinct outcomes in an output distribution, and $|J|$ be the number of quadratic terms in H_z.

Quantum complexity The quantum resource utilization in FrozenQubits scales exponentially with the number of skipping qubits, $\mathbf{O}(\mathbf{2^m})$. However, we can eliminate a significant number of sub-problems and substantially subside the quantum overhead of FrozenQubits, without compromising the performance of the primary QAOA application (we discussed it in Section 3.7.2). Note that m does not scale with N and for power-law graphs $m \ll N$.

Circuit compilation complexity: Assuming that all sub-problems are run on the same quantum computer, FrozenQubits only compiles one *template circuit*; accordingly, the compilation complexity of FrozenQubits is $\mathbf{O}(\mathbf{1})$. This takes significantly less time compared to compiling the QAOA circuit for the baseline.

Time complexity: The complexity of required tasks to be performed on a classical computer depends on the complexity of different components. The complexity for identifying the top m hotspot nodes is $O(N + m\log m)$, assuming that the adjacency list of the graph representing H_Z is available. The complexity order of forming the adjacency list in the worst case (i.e., fully connected graphs) is $O(N^2)$. A node is connected to at most $N-1$ other nodes; therefore, freezing m nodes scales with $O(mN)$, and forming sub-problems scales with $O(ms2^m)$. Decoding every outcome (whith $N-m$ bits) to the state-space of the original problem Hamiltonian is $O(m)$. Hence, inferring the final solution is $O(s2^m(m+N+|J|))$. For problems at a practical scale, $m \ll N \ll s$. Therefore, classical time complexity scales with the order of $\mathbf{O}(\mathbf{sN^2})$, excluding the circuit compilation time that is reduced significantly with increasing m.

Memory complexity: To identify and freeze hotspot qubits, FrozenQubits use the adjacency list representation of the input problem graph which has the space complexity of $O(N^2)$. Since sub-problems are independent, decoding the output distribution has the space complexity of $O(sN)$. For QAOA applications at a practical scale, we expect that $N \ll s$; thus, the overall space complexity of FrozenQubits is $\mathbf{O}(\mathbf{sN})$.

3.9 Comparison with Prior Works That Use Sub-Circuits

CutQC is a prior studies that divides a quantum circuit into smaller sub-circuits [29, 91, 107]. While CutQC is applicable to any quantum circuit, it works best when there are limited connections between qubits in the input quantum circuit. However, this is not true for QAOA and other variational quantum algorithms. Moreover, CutQC is bottle-necked by exponentially complex post-processing that scales with the number of qubits. On the other hand, FrozenQubits freezes only some of the hotspots (up to two in our default design) to create a limited number of circuits and does not incur such post-processing costs. Table 3 compares FrozenQubits with CutQC.

Table 3: Comparison of FrozenQubits and CutQC.

Design	Application	Overheads		
		Compile	Quantum	Post-process
CutQC	Generic	Linear	Linear	Exponential (in qubits)
FrozenQubits	QAOA	$O(1)$	Exponential* (in m)	Polynomial

**Our default FrozenQubits design only freezes up-to $m = 2$ qubits. For real-world applications, freezing a few hotspots is sufficient for FrozenQubits to be effective. Please see Section 5.1.3 for an analysis on this.*

4 METHODOLOGY

4.1 Benchmarks

We study FrozenQubits on three types of graphs: (1) Power-law, (2) 3-regular, and (3) fully-connected or SK-model graphs. While most real-world problems follow Power-law distribution, existing quantum computers cannot run problems at such scale as they involve hundreds of qubits. So instead, we generate smaller Power-law graphs that mimic the characteristics of real-world problems but can be run on real systems available today. To generate Power-law graphs, we use the widely accepted *Barabasi—Albert (BA)* algorithm [9, 20, 21, 56, 66, 75, 112, 115, 116]. BA graphs are associated with a preferential attachment factor d_{BA} that controls the the density of the graphs. We generate Power-law graphs using the BA algorithm for $d_{\text{BA}} = 1$ as prior studies show that $d_{\text{BA}} = 1$ can capture the dynamics of most real-world systems [34]. To study FrozenQubits on denser graphs, we use BA graphs corresponding to $d_{\text{BA}} = 2$ and 3. For all the graphs, the edge weights are randomly drawn from $\{-1, +1\}$, and all node coefficients ($\mathbf{h}_i$) are set to zero [39, 59]. Figure 6 shows five samples of the benchmark graphs used in this study.

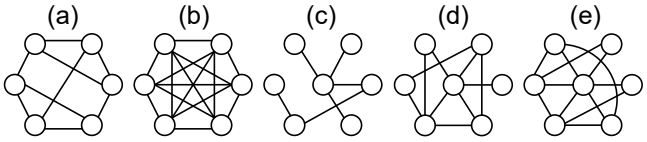


Figure 6: Random graphs: (a) 3-regular, (b) SK model, (c) BA ($d_{\text{BA}} = 1$), (d) BA ($d_{\text{BA}} = 2$), and (e) BA ($d_{\text{BA}} = 3$).

4.2 Baseline and Experimental Platform

Baseline: We run the QAOA circuits with the optimal circuit parameters (determined from simulations) on the NISQ hardware for 100K trials. This approach is consistent with prior works on compilers for QAOA [8]. Note that prior works show that additional trials do not improve application fidelity once the distribution saturates after a certain point [43]. To compile each circuit, we use IBM's Qiskit tool-chain with noise-adaptive routing and the highest optimization level 3.

Quantum hardware.: We use eight IBMQ systems with 27—127 qubits: Washington, Brooklyn, Montreal, Auckland, Toronto, Mumbai, Hanoi, and Cairo.

4.3 Figure of Merit

We evaluate the application fidelity of QAOA circuits using Approximation Ratio Gap (ARG) from prior works [8, 41, 60], as defined in Equation (4).

$$\text{ARG} = 100 \times \left| \frac{EV_{ideal} - EV_{real}}{EV_{ideal}} \right| \tag{4}$$

where EV_{ideal} and EV_{real} denote the expected values of the QAOA circuit on an ideal simulator and the real quantum machine, respectfully. ARG $\in [0, +\infty]$), and lower is better.

5 EVALUATION

5.1 FrozenQubits on Power-Law graphs

We evaluate FrozenQubits using Barabasi Albert (BA) graphs [20] that represent most power-law graphs [34].

5.1.1 Impact of Number of CNOTs and Circuit Depth. Figure 7(a) shows that FrozenQubits reduces the number of CNOTs by 3.13x on average when only one qubit is frozen. The CNOT count reduces by 7.19x on average when two qubits are frozen. Figure 7(b) shows the impact of FrozenQubits on circuit depth. Freezing a single qubit reduces circuit depth by 2.23x on average. Note that the CNOT count and circuit depth include the overheads from SWAP operations. Freezing two qubits reduces depth by 3.65x on average.

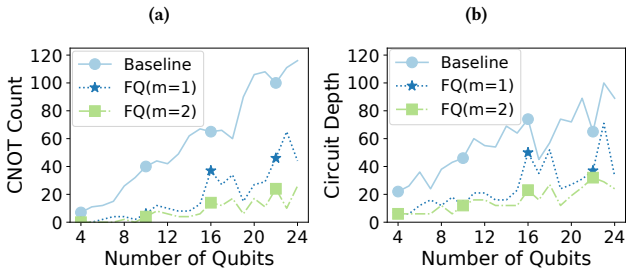


Figure 7: (a) CNOT counts and (b) Depth of QAOA and FrozenQubits (FQ) circuits, when $m = 1$ and 2 qubits are frozen.

5.1.2 Impact of FrozenQubits on Application Fidelity. Figure 8 shows that FrozenQubits improves the Approximation Ratio Gap (ARG) of the power-law (BA) QAOA circuits by 6.75x on average and by up-to 47.04x when $m = 1$ qubit is frozen. Freezing 2 qubits improves the ARG by 11.29x on average and by up-to 57.14x.

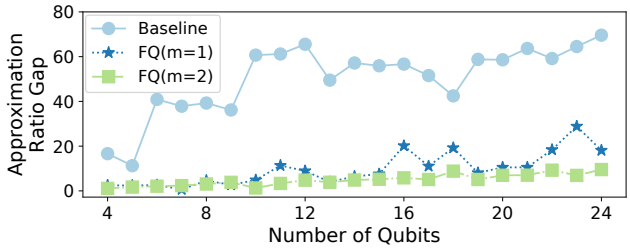


Figure 8: Approximation Ratio Gap (ARG) on IBM-Montreal using standard QAOA and FrozenQubits (FQ) for $m = 1$ and 2.

From Figure 8, we also observe that the fidelity of the baseline reduces (increasing ARG) rapidly with the circuit size. On the contrary, the ARG reduces at a much slower rate with increasing problem size for FrozenQubits.

As FrozenQubits exploits the symmetry of the search space, it does not incur any quantum cost when only a single node is frozen, and freezing two qubits requires twice the quantum resources. For more information regarding how leveraging the symmetry of search space reduces the overheads of FrozenQubits, see Section 3.7.2.

5.1.3 Cost and Fidelity Trade-off. There exists a trade-off between the fidelity improvement from freezing additional qubits and the quantum cost of running FrozenQubits. While freezing more qubits improves the fidelity of QAOA circuits, it simultaneously increases the quantum cost of FrozenQubits exponentially. However, our evaluations show that freezing additional qubits has diminishing returns after a certain point. For example, Figure 9(a) shows that the improvement in ARG saturates after freezing seven qubits.

Therefore, to make a trade-off, we must: (1) determine the quantum budget; and (2) roughly estimate the point (i.e., number of qubits to freeze) where the trend in improving the fidelity plateaus. The quantum budget is user-specific and depends on various factors such as the availability of quantum resources and specific requirements of the underlying applications. To estimate the number of qubits to be frozen, we can use circuit properties such as number of CNOTs and depth. For example, Figure 9(b) shows that these circuit parameters can accurately capture the application fidelity trends of the QAOA circuits.

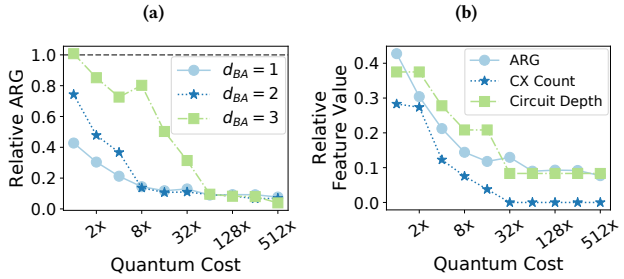


Figure 9: Trade-off between quantum cost and fidelity.

5.1.4 FrozenQubits on Dense Power-law (BA) Graphs. To study the impact of FrozenQubits on denser power-law graphs, we use BA graphs with $d_{\mathrm{BA}} = 2$and 3. A higher d_{BA} corresponds to a denser graph. Figure 10 shows that for dense graphs ($d_{\mathrm{BA}} = 2$), when one qubit is frozen, FrozenQubits improves ARG by 1.76x on average and by up-to 12.8x. Even for very dense graphs, corresponding to $d_{\mathrm{BA}} = 3$, FrozenQubits improves the ARG by 1.43x on average and by up-to 14.1x. Freezing two qubits enhances the performance even further, as shown in Figure 10.

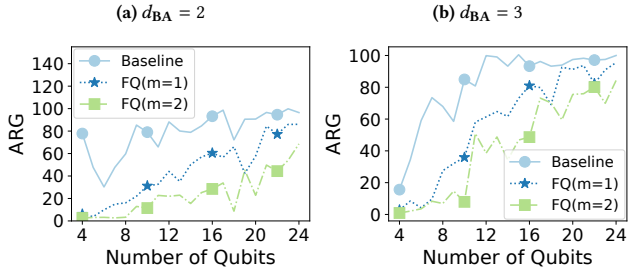


Figure 10: Approximation Ratio Gap (ARG) of denser BA graphs on IBM Montreal.

5.2 FrozenQubits on Regular Graphs

We study FrozenQubits on (1) 3-regular graphs and (2) fully connected graphs (or SK model) [24, 59, 100]. Figure 11 shows that FrozenQubits improves the ARG of QAOA for 3-regular graphs by 1.25x on average (and by up to 4.52x). For SK-model graphs, the ARG improves by 1.28x on average and by up to 3.79x when a single qubit is frozen. Freezing two qubits improves the effectiveness of FrozenQubits further.

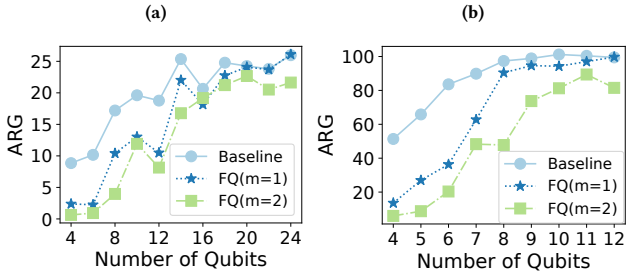


Figure 11: Approximation Ratio Gap (ARG) of (a) 3-regular graphs and (b) SK model on IBM-Montreal.

5.3 Impact of Resources on Training QAOA

Executing the Baseline for More Trials: The fidelity of the baseline does not improve once the output distribution saturates after few thousands of trials due to correlated errors [43]. Recent QAOA studies from Google uses 25K trials for programs using up-to 25 qubits, whereas our baseline is executed for 100K trials.

Executing the Baseline for More Iterations: The performance of QAOA depends on (1) the fidelity of the circuits and (2) the ability of the optimizer to tune the circuit parameters using the output of the quantum circuits. Unfortunately, noisy outputs impact the training as the circuits lose their sensitivity to parameters changes [59, 105].

To understand the impact of FrozenQubits on the training process of QAOA, we run the baseline and FrozenQubits for a 20-qubit power-law graph across a 50×50 optimization landscape and compute the approximation ratio (AR) as:

$$AR = \frac{\text{Expected Value}}{C_{\min}} \tag{5}$$

where $C_{\min}$ is the global minima. Each point denotes a unique combination of two circuit parameters (γ and β). AR $\in [-\infty, 1]$ and is maximum when all outcomes in the output distribution are global optima [59, 105]. Note that we perform this analysis on the grid to compare the baseline and FrozenQubits on the entire landscape as opposed to a specific path traversed by the optimizer.

Figure 12 shows that the landscape of the baseline is much more blurred due to noise, compared to FrozenQubits. Thus, even if the baseline is executed for more iterations, it may still not improve the quality of the solution as the sensitivity of the quantum circuits is lowered by noise. However, the reduced noise in FrozenQubits enables it to sharpen the gradients in the parameter landscape considerably. Therefore, circuits are more sensitive to changes in parameters in FrozenQubits, aiding the training of QAOA.

5.4 Sensitivity of FrozenQubits to Machines

Figure 13 shows that freezing one qubit across different machines improves the mean ARG of QAOA by 3.69x on average and by up to 5.20x. The ARG improves by 7.8x on average and by up to 13.16x when two qubits are frozen.

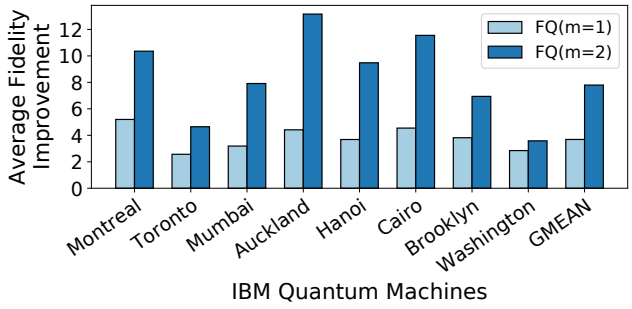


Figure 13: Average ARG improvement across IBMQ-systems.

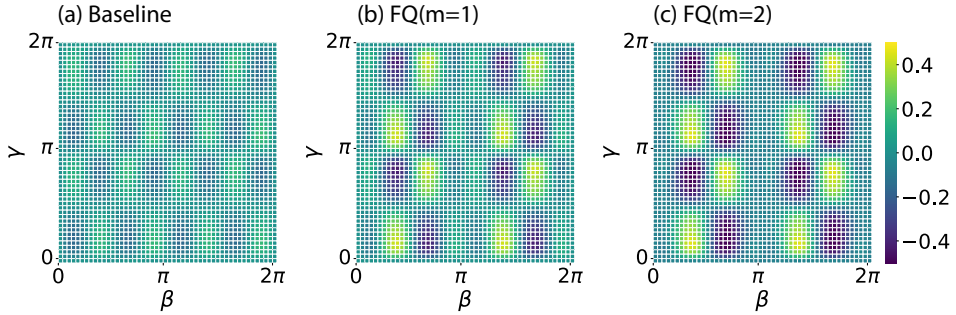


Figure 12: Classical optimizer landscape of AR in Equation (5) for (a) baseline, (b) FrozenQubits with $m = 1$, and (c) FrozenQubits with $m = 2$, on IBMQ-Auckland for a 20-qubit power-law graph. Note that the landscapes are not identical as the search space of the Hamiltonians in FQ$(m = 1)$ and FQ$(m = 2)$ correspond to half and quarter of the search space, respectively.

6 FROZENQUBITS AT PRACTICAL SCALE

Demonstration of quantum advantage using QAOA requires solving problems at the scale of several hundreds of qubits [57]. Unfortunately, we cannot run such applications on existing quantum computers. Even a state-of-the-art device such as 53-qubit Google Sycamore cannot run QAOA benchmarks beyond 23 qubits [59]. To evaluate the impact of FrozenQubits at practical scale, we study its performance using 500-qubit random power-law QAOA circuits on a 50×50 grid.

6.1 Impact on Number of CNOTs

Figure 14 shows the reduction in CNOT counts post-compilation for a 500-qubit circuit. Increasing the number of frozen qubits (m) reduces the number of CNOTs. For example, freezing ten qubits results in 65.94% CNOT reduction, at the cost of training 512 circuits independently. Further analysis shows that 91.47% of this reduction corresponds to reduced number of SWAP operations. Moreover, as FrozenQubits skip hotspot nodes, SWAP reduction on average provides 10.19x higher contribution to the total CNOT reduction, compared to reduction of CNOTs in the QAOA circuit due to the dropped edges corresponding to the skipped nodes (at the circuit before compilation). We make similar observations even on denser BA graphs, as shown in Figure 15(a).

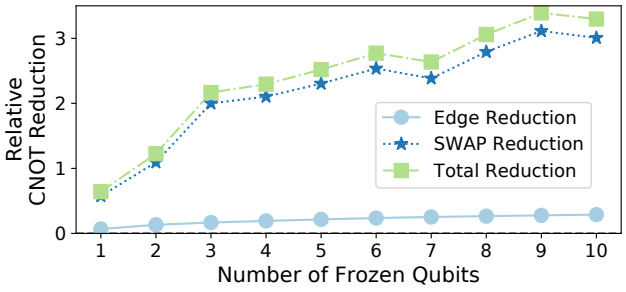


Figure 14: Relative CNOT reduction on BA benchmarks with $d_{\mathbf{BA}} = 1$. Higher is better.

6.2 Impact on Circuit Depth

Figure 15(b) shows that FrozenQubits reduces circuit depth of BA benchmarks by 1.47x–5.25x on average when the number of qubits frozen increase from one to ten.

6.3 Impact on Expected Probability of Success

Running such large problems on existing quantum computers to evaluate the performance of FrozenQubits is infeasible. Instead, we compute the *Expected Probability of Success (EPS)* using an optimistic error model, where we assume 0.1% CNOT error-rate and 0.5% readout error-rate. We also assume a 500μ seconds decoherence time. EPS is the probability that gate and measurement operations remain error-free and qubits remain free from decoherence. It is widely used in evaluating the performance of NISQ compilers on large programs [8, 43, 84, 108]. Figure 16 shows that FrozenQubits improves the EPS by 404x on average and by up to 515,900x.

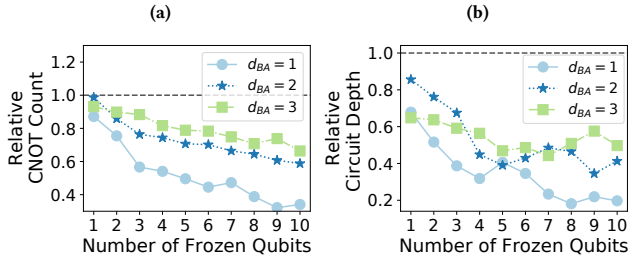


Figure 15: Relative (a) CNOT counts and (b) circuit depth, for BA benchmarks with $d_{\mathbf{BA}} = 1, 2$ and 3. Lower is better.

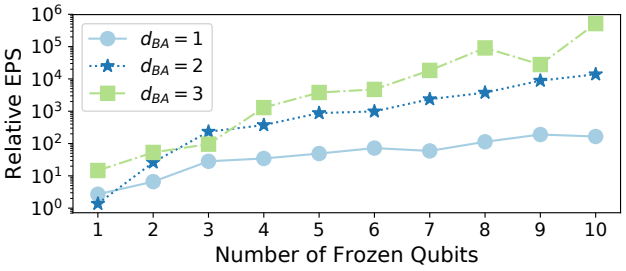


Figure 16: Expected Probability of Success (EPS) comparison.

6.4 Impact on Compilation Time

As the sub-circuits from FrozenQubits have fewer operations compared to the baseline and require fewer SWAPs, their compilation time is lower. But, FrozenQubits requires 2^m sub-circuits when m qubits are frozen and each of them must be compiled into an executable. However, our approach reduces the compilation overhead by compiling only one of the sub-circuits and generating the remaining executables by editing it (sequentially or in parallel). This approach reduces that compilation overheads of FrozenQubits. For example, Figure 17(a) shows that freezing ten qubits decreases the compilation time by 22.06%. This reduction is significantly higher than the overall time required to generate all the $O(2^m)$ executables, as shown in Figure 17(b).

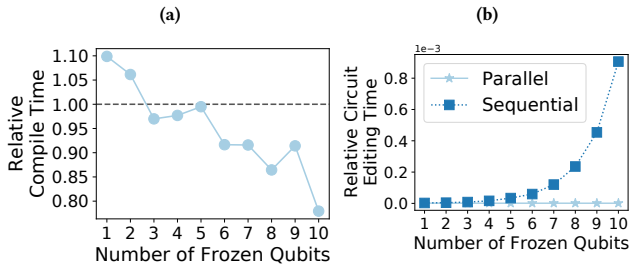


Figure 17: (a) Relative compilation time. (b) Time required to generate all executables for FrozenQubits, relative to the time required to compile the baseline.

6.5 End-to-End Workflow Runtime Analysis

Quantum computers are accessed via cloud services [3, 10, 32, 36, 77] and the overall runtime of a quantum circuit depends on several factors such as (1) queuing delays, (2) execution mode, and (3) execution time on the NISQ device. The cloud management software frameworks differ between providers and are evolving to provide flexibility to the users [42, 94, 95]. For example, users can access devices in *dedicated* or *shared* mode [10]. Similarly, some device providers allow launching multiple circuits simultaneously as part of a single cloud job, for instance, up to 900 circuits at once on IBMQ systems [36]. To capture the diverse execution models and fairly compare the runtime of the baseline and FrozenQubits, we use an analytical model described by Equation (6), where I is the number of QAOA iterations, τ is the number of trials, t_{NISQ} is the execution time for each trial, N_{batch} is the number of batches required, Δ_{cloud} is the cloud access latency, Δ_{opt} is the latency of the classical optimizer, $\delta_{compile}$ is the compilation latency, and δ_{pp} is post-processing time needed (if any).

$$T = \delta_{compile} + I \times N_{batch} \times \left(\tau \times t_{NISQ} + \Delta_{cloud}\right) + \delta_{opt} + \delta_{pp} \quad (6)$$

For our analysis, we assume (1) two execution modes: no-batching (such as Rigetti devices [10]) and batching up-to 900 circuits as on IBMQ systems [36]; and (2) two device access modes: shared with $\Delta_{cloud} = 30$ minutes and dedicated with $\Delta_{cloud} = 0$. By default, we assume both the baseline and FrozenQubits run $\tau = 25K$ trials [59] per circuit and it requires $t_{NISQ} = 1$ millisecond to run a trial. We also assume $\Delta_{opt} = 1$ minute is the optimizer latency for an iteration, and by default we need $I = 1000$ iterations per circuit. Both the baseline and FrozenQubits compile a single circuit only once and we assume a latency of $\delta_{compile} = 2$ hours. Lastly, we assume post-processing time of FrozenQubits, $\delta_{pp} = 1$ minute. Figure 18 compares the end-to-end runtime of the baseline, default FrozenQubits that freezes up to two qubits, and FrozenQubits freezing 10 qubits. Note that, freezing only some of the hotspots is sufficient for FrozenQubits to be effective.

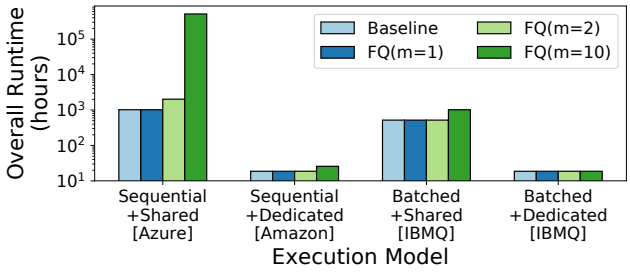


Figure 18: Overall Runtime comparison.

The end-to-end runtime for both the baseline and FrozenQubits depends on the execution model. The batching of circuits allow FrozenQubits to launch all sub-circuits in an iteration simultaneously, reducing the runtime. We use a simplified model to estimate the runtime and it will be lower if a user improves the throughput by multi-programming NISQ devices [42, 72] or running circuits on multiple devices [94, 103].

7 RELATED WORK

Software policies to improve the fidelity of NISQ applications is an active area of research. We can classify these techniques as generic and QAOA-specific policies. Generic policies do not use the domain knowledge of the underlying application, and try to improve the fidelity of NISQ computers by: (1) better compiling quantum circuits [23, 40, 41, 55, 67, 69, 81, 82, 87, 101, 102, 109, 114]; and (2) postprocessing output distributions [28, 63, 88, 108]. These policies are orthogonal to our proposed technique, and one may combine them with FrozenQubits. Approximating quantum circuits [89] and circuit cutting techniques [107] can improve the fidelity of NISQ computers. However, applying these techniques to QAOA circuits with hotspots is nontrivial.

QAOA-specific policies try to leverage the domain knowledge of the underlying problem for improving the fidelity of QAOA applications [68, 70]. Reordering Pauli terms of Hamiltonians can result in QAOA circuits with lower depth and fewer CNOTs [8]. However, increasing the number of CNOTs that are run in parallel can escalate the crosstalk [114]. Partial compilation of circuits in variational algorithms at the pulse level [54] can shorten the execution time of QAOA but has the overheads of dealing with custom pulses.

8 CONCLUSION

We propose *FrozenQubits*, an application-level software framework for boosting the fidelity of Quantum Approximate Optimization Algorithm (QAOA). It leverages the insight that most natural and artificial graphs follow Power-law distribution. FrozenQubits freezes hotspot qubits and intelligently partitions the state-space of the problem into several smaller sub-spaces such that the corresponding QAOA sub-circuits are significantly less robust to hardware errors on NISQ devices. To subside the quantum complexity of FrozenQubits, we define and prove a new theorem that eliminates running a considerable number of QAOA processes for sub-problems without losing the guarantee of the recovery of the exact solution. Our evaluations using 5,300 QAOA circuits on eight IBMQ computers show that FrozenQubits improves the fidelity of QAOA circuits by up to 57.14x compared to the baseline.

ACKNOWLEDGEMENTS

We thank Swamit Tannu and Nicolas Delfosse for helpful discussions and comments. We also thank Suhas Vittal for his inputs on the error modelling and editorial comments. Ramin Ayanzadeh was supported by the NSF Computing Innovation Fellows (CI-Fellows) program. This work was funded in part by EPiQC, an NSF Expedition in Computing, under grant CCF-1730449. Poulami Das was funded by the Microsoft Research PhD Fellowship. This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725. This research was supported in part through research cyberinfrastructure resources and services provided by the Partnership for an Advanced Computing Environment (PACE) at the Georgia Institute of Technology, Atlanta, Georgia, USA

REFERENCES

[1] [n. d.]. BMW's supply chain makes a quantum leap. https://www.cips.org/supply-management/analysis/2021/may/bmw-supply-chain-makes-a-quantum-leap/. Accessed: 2022-07-27.

[2] [n. d.]. How Network Math Can Help You Make Friends. https://www.quantamagazine.org/how-network-math-can-help-you-make-friends-20180820/. Accessed: 2022-07-27.

[3] 2021. Five years ago today, we put the first quantum computer on the cloud. Here's how we did it. https://research.ibm.com/blog/quantum-five-years. Accessed: 2022-07-27.

[4] Matthew T Agler, Jonas Ruhe, Samuel Kroll, Constanze Morhenn, Sang-Tae Kim, Detlef Weigel, and Eric M Kemen. 2016. Microbial hub taxa link host and abiotic factors to plant microbiome variation. *PLoS biology* 14, 1 (2016), e1002352.

[5] Google Quantum AI. Accessed: June 19, 2021. Quantum Computer Datasheet. https://quantumai.google/hardware/datasheet/weber.pdf.

[6] Yacine Aıt-Sahalia and Felix HA Matthys. 2015. Robust portfolio optimization with jumps.

[7] Meisam Akbarzadeh, Soroush Memarmontazerin, and Sheida Soleimani. 2018. Where to look for power Laws in urban road networks? *Applied Network Science* 3, 1 (2018), 1–11.

[8] Mahabubul Alam, Abdullah Ash-Saki, and Swaroop Ghosh. 2020. Circuit compilation methodologies for quantum approximate optimization algorithm. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*. IEEE, 215–228.

[9] Reka Albert. 2005. Scale-free networks in cell biology. *Journal of cell science* 118, 21 (2005), 4947–4957.

[10] Amazon. 2022. Amazon Braket - Explore and experiment with quantum computing:. https://aws.amazon.com/braket/. [Online; accessed 22-July-2021].

[11] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[12] Ramin Ayanzadeh, Poulami Das, Swamit S Tannu, and Moinuddin Qureshi. 2021. EQUAL: Improving the Fidelity of Quantum Annealers by Injecting Controlled Perturbations. *arXiv preprint arXiv:2108.10964* (2021).

[13] Ramin Ayanzadeh, John Dorband, Milton Halem, and Tim Finin. 2021. Multi-Qubit Correction for Quantum Annealers. *Scientific Reports* 11 (2021).

[14] Ramin Ayanzadeh, John Dorband, Milton Halem, and Tim Finin. 2022. Quantum-assisted greedy algorithms. In *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 4911–4914.

[15] Ramin Ayanzadeh, Milton Halem, and Tim Finin. 2020. An ensemble approach for compressive sensing with quantum annealers. In *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 3517–3520.

[16] Ramin Ayanzadeh, Milton Halem, and Tim Finin. 2020. Reinforcement Quantum Annealing: A Hybrid Quantum Learning Automata. *Scientific Reports* 10, 1 (2020), 1–11.

[17] Ramin Ayanzadeh, Seyedahmad Mousavi, Milton Halem, and Tim Finin. 2019. Quantum annealing based binary compressive sensing with matrix uncertainty. *arXiv preprint arXiv:1901.00088* (2019).

[18] Utkarsh Azad, Bikash K. Behera, Emad A. Ahmed, Prasanta K. Panigrahi, and Ahmed Farouk. 2022. Solving Vehicle Routing Problem Using Quantum Approximate Optimization Algorithm. *IEEE Transactions on Intelligent Transportation Systems* (2022), 1–10. https://doi.org/10.1109/TITS.2022.3172241

[19] Jack S Baker and Santosh Kumar Radha. 2022. Wasserstein Solution Quality and the Quantum Approximate Optimization Algorithm: A Portfolio Optimization Case Study. *arXiv preprint arXiv:2202.06782* (2022).

[20] Albert-László Barabási and Réka Albert. 1999. Emergence of scaling in random networks. *science* 286, 5439 (1999), 509–512.

[21] Albert-László Barabási, Réka Albert, and Hawoong Jeong. 2000. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: statistical mechanics and its applications* 281, 1-4 (2000), 69–77.

[22] Panagiotis Kl. Barkoutsos, Giacomo Nannicini, Anton Robert, Ivano Tavernelli, and Stefan Woerner. 2020. Improving Variational Quantum Optimization using CVaR. *Quantum* 4 (apr 2020), 256. https://doi.org/10.22331/q-2020-04-20-256

[23] George S Barron and Christopher J Wood. 2020. Measurement error mitigation for variational quantum algorithms. *arXiv preprint arXiv:2010.08520* (2020).

[24] Joao Basso, Edward Farhi, Kunal Marwaha, Benjamin Villalonga, and Leo Zhou. 2021. The Quantum Approximate Optimization Algorithm at High Depth for MaxCut on Large-Girth Regular Graphs and the Sherrington-Kirkpatrick Model. *arXiv preprint arXiv:2110.14206* (2021).

[25] Christopher DB Bentley, Samuel Marsh, André RR Carvalho, Philip Kilby, and Michael J Biercuk. 2022. Quantum computing for transport optimization. *arXiv preprint arXiv:2206.07313* (2022).

[26] Marcus Berliant and Axel H Watanabe. 2018. A scale-free transportation network explains the city-size distribution. *Quantitative Economics* 9, 3 (2018), 1419–1451.

[27] Sebastian Brandhofer, Daniel Braun, Vanessa Dehn, Gerhard Hellstern, Matthias Hüls, Yanjun Ji, Ilia Polian, Amandeep Singh Bhatia, and Thomas Wellens. 2022. Benchmarking the performance of portfolio optimization with QAOA. https://doi.org/10.48550/ARXIV.2207.10555

[28] Sergey Bravyi, Sarah Sheldon, Abhinav Kandala, David C Mckay, and Jay M Gambetta. 2020. Mitigating measurement errors in multi-qubit experiments. *arXiv preprint arXiv:2006.14044* (2020).

[29] Sergey Bravyi, Graeme Smith, and John A Smolin. 2016. Trading classical and quantum computational resources. *Physical Review X* 6, 2 (2016), 021043.

[30] Oscar Bulancea Lindvall. 2019. Quantum Methods for Sequence Alignment and Metagenomics.

[31] Sergey V Buldyrev. 2006. Power law correlations in DNA sequences. *Power laws, scale-free networks and genome biology* (2006), 123–164.

[32] Davide Castelvecchi. 2017. IBM's quantum cloud computer goes commercial. *Nature* 543, 7644 (2017).

[33] Jaeho Choi, Seunghyeok Oh, and Joongheon Kim. 2020. Quantum approximation for wireless scheduling. *Applied Sciences* 10, 20 (2020), 7116.

[34] Aaron Clauset, Ellen Tucker, and Matthias Sainz. 2016. The Colorado index of complex networks. *Retrieved July* 20, 2018 (2016), 22.

[35] Benjamin A Cordier, Nicolas PD Sawaya, Gian G Guerreschi, and Shannon K McWeeney. 2021. Biology and medicine in the landscape of quantum advantages. *arXiv preprint arXiv:2112.00760* (2021).

[36] International Business Machines Corporation. 2021. Universal Quantum Computer Development at IBM:. http://research.ibm.com/ibm-q/research/. [Online; accessed 22-July-2021].

[37] MO Costa, R Silva, DHAL Anselmo, and JRP Silva. 2019. Analysis of human DNA through power-law statistics. *Physical Review E* 99, 2 (2019), 022112.

[38] Constantin Dalyac, Loïc Henriet, Emmanuel Jeandel, Wolfgang Lechner, Simon Perdrix, Marc Porcheron, and Margarita Veshchezerova. 2021. Qualifying quantum approaches for hard industrial optimization problems. A case study in the field of smart-charging of electric vehicles. *EPJ Quantum Technology* 8, 1 (2021), 12.

[39] Arnab Das and Bikas K Chakrabarti. 2008. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics* 80, 3 (2008), 1061.

[40] Poulami Das, Swamit Tannu, Siddharth Dangwal, and Moinuddin Qureshi. 2021. ADAPT: Mitigating Idling Errors in Qubits via Adaptive Dynamical Decoupling. In *MICRO-54*. 950–962. https://doi.org/10.1145/3466752.3480059

[41] Poulami Das, Swamit Tannu, and Moinuddin Qureshi. 2021. JigSaw: Boosting Fidelity of NISQ Programs via Measurement Subsetting. In *MICRO-54*. 937–949. https://doi.org/10.1145/3466752.3480044

[42] Poulami Das, Swamit S Tannu, Prashant J Nair, and Moinuddin Qureshi. 2019. A case for multi-programming quantum computers. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. 291–303.

[43] Poulami Das, Swamit S Tannu, and Moinuddin Qureshi. 2021. JigSaw:Boosting Fidelity of NISQ Programs via Measurement Subsetting. In *MICRO*.

[44] Vladimir G Deı, Bettina Klinz, Gerhard J Woeginger, et al. 2006. Exact algorithms for the Hamiltonian cycle problem in planar graphs. *Operations Research Letters* 34, 3 (2006), 269–274.

[45] Daniel J Egger, Claudio Gambella, Jakub Marecek, Scott McFaddin, Martin Mevissen, Rudy Raymond, Andrea Simonetto, Stefan Woerner, and Elena Yndurain. 2020. Quantum computing for finance: State-of-the-art and future prospects. *IEEE Transactions on Quantum Engineering* 1 (2020), 1–24.

[46] Jonas Elmerraji. 2021. Optimal CVaR Portfolio Construction Under Power Law Stochastic Walks. *Available at SSRN 3959932* (2021).

[47] Prashant S Emani, Jonathan Warrell, Alan Anticevic, Stefan Bekiranov, Michael Gandal, Michael J McConnell, Guillermo Sapiro, Alán Aspuru-Guzik, Justin T Baker, Matteo Bastiani, et al. 2021. Quantum computing at the frontiers of biological sciences. *Nature Methods* 18, 7 (2021), 701–709.

[48] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).

[49] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A Quantum Approximate Optimization Algorithm Applied to a Bounded Occurrence Constraint Problem. *arXiv preprint arXiv:1412.6062* (2014).

[50] Mark Fingerhuth, Tomáš Babej, and Christopher Ing. 2018. A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding. https://doi.org/10.48550/ARXIV.1810.13411

[51] David Fitzek, Toheed Ghandriz, Leo Laine, Mats Granath, and Anton Frisk Kockum. 2021. Applying quantum approximate optimization to the heterogeneous vehicle routing problem. https://doi.org/10.48550/ARXIV.2110.06799

[52] D Gamermann, J Triana-Dopico, and R Jaime. 2019. A comprehensive statistical study of metabolic and protein–protein interaction network properties. *Physica A: Statistical Mechanics and its Applications* 534 (2019), 122204.

[53] Kwang-Il Goh, Eulsik Oh, Hawoong Jeong, Byungnam Kahng, and Doochul Kim. 2002. Classification of scale-free networks. *Proceedings of the National Academy of Sciences* 99, 20 (2002), 12583–12588.

[54] Pranav Gokhale, Yongshan Ding, Thomas Propson, Christopher Winkler, Nelson Leung, Yunong Shi, David I Schuster, Henry Hoffmann, and Frederic T Chong. 2019. Partial Compilation of Variational Algorithms for Noisy Intermediate-Scale Quantum Machines. In *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 266–278.

[55] Pranav Gokhale, Ali Javadi-Abhari, Nathan Earnest, Yunong Shi, and Frederic T Chong. 2020. Optimized Quantum Compilation for Near-Term Algorithms with OpenPulse. *arXiv preprint arXiv:2004.11205* (2020).

[56] Caitlin Gray, Lewis Mitchell, and Matthew Roughan. 2018. Super-blockers and the effect of network structure on information cascades. In *Companion Proceedings of the The Web Conference 2018*. 1435–1441.

[57] Gian Giacomo Guerreschi and Anne Y Matsuura. 2019. QAOA for Max-Cut requires hundreds of qubits for quantum speed-up. *Scientific reports* 9, 1 (2019), 1–7.

[58] Frank Hadlock. 1975. Finding a maximum cut of a planar graph in polynomial time. *SIAM J. Comput.* 4, 3 (1975), 221–225.

[59] Matthew P Harrigan, Kevin J Sung, Matthew Neeley, Kevin J Satzinger, Frank Arute, Kunal Arya, Juan Atalaya, Joseph C Bardin, Rami Barends, Sergio Boixo, et al. 2021. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nature Physics* 17, 3 (2021), 332–336.

[60] Rebekah Herrman, Phillip C Lotshaw, James Ostrowski, Travis S Humble, and George Siopsis. 2022. Multi-angle quantum approximate optimization algorithm. *Scientific Reports* 12, 1 (2022), 1–10.

[61] Yunzhang Hou, Xiaoling Wang, Yenchun Jim Wu, and Peixu He. 2018. How does the trust affect the topology of supply chain network and its resilience? An agent-based approach. *Transportation Research Part E: Logistics and Transportation Review* 116 (2018), 229–241.

[62] Thomas House, Jonathan M Read, Leon Danon, and Matthew J Keeling. 2015. Testing the hypothesis of preferential attachment in social network formation. *EPJ Data Science* 4, 1 (2015), 1–13.

[63] IBM. 2010. Measurement Error Mitigation. https://qiskit.org/textbook/ch-quantum-hardware/measurement-error-mitigation.html. [Online; accessed 26-July-2020].

[64] Hawoong Jeong, Sean P Mason, A-L Barabási, and Zoltan N Oltvai. 2001. Lethality and centrality in protein networks. *Nature* 411, 6833 (2001), 41–42.

[65] B Kahng, I Yang, H Jeong, and A-L Barabási. 2004. Emergence of power-law behaviors in online auctions. In *The Application of Econophysics*. Springer, 204–209.

[66] Sung-Soo Kim, Young-Min Kang, and Young-Kuk Kimt. 2022. Sparsity-Aware Reachability Computation for Massive Graphs. In *2022 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE, 157–160.

[67] Hyeokjea Kwon and Joonwoo Bae. 2020. A hybrid quantum-classical approach to mitigating measurement errors. *arXiv preprint arXiv:2003.12314* (2020).

[68] Lingling Lao and Dan E Browne. 2022. 2qan: A quantum compiler for 2-local qubit hamiltonian simulation algorithms. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 351–365.

[69] Gushu Li, Yufei Ding, and Yuan Xie. 2018. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. *arXiv preprint arXiv:1809.02573* (2018).

[70] Gushu Li, Anbang Wu, Yunong Shi, Ali Javadi-Abhari, Yufei Ding, and Yuan Xie. 2022. Paulihedral: a generalized block-wise compiler optimization framework for Quantum simulation kernels. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 554–569.

[71] Junde Li, Mahabubul Alam, and Swaroop Ghosh. 2021. Large-scale Quantum Approximate Optimization via Divide-and-Conquer. *arXiv preprint arXiv:2102.13288* (2021).

[72] Lei Liu and Xinglei Dou. 2021. QuCloud: A New Qubit Mapping Mechanism for Multi-programming Quantum Computing in Cloud Environment. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*. 167–178. https://doi.org/10.1109/HPCA51647.2021.00024

[73] Seth Lloyd. 2018. Quantum approximate optimization is computationally universal. *arXiv preprint arXiv:1812.11075* (2018).

[74] Andrew Lucas. 2014. Ising formulations of many NP problems. *Frontiers in physics* 2 (2014), 5.

[75] David Lusseau. 2003. The emergent properties of a dolphin social network. *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270, suppl_2 (2003), S186–S188.

[76] Abram Magner, Wojciech Szpankowski, and Daisuke Kihara. 2015. On the origin of protein superfamilies and superfolds. *Scientific reports* 5, 1 (2015), 1–7.

[77] Microsoft. 2022. Azure Quantum - Quantum Service | Microsoft Azure. https://azure.microsoft.com/en-us/services/quantum/#product-overview. [Online; accessed 22-July-2021].

[78] Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. 2007. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 29–42.

[79] Mauro ES Morales, Jacob D Biamonte, and Zoltán Zimborás. 2020. On the universality of the quantum approximate optimization algorithm. *Quantum Information Processing* 19, 9 (2020), 1–26.

[80] Tomoya Mori, Tony E Smith, and Wen-Tai Hsu. 2020. Common power laws for cities and spatial fractal structures. *Proceedings of the National Academy of Sciences* 117, 12 (2020), 6469–6475.

[81] Prakash Murali, Jonathan M Baker, Ali Javadi Abhari, Frederic T Chong, and Margaret Martonosi. 2019. Noise-Adaptive Compiler Mappings for Noisy Intermediate-Scale Quantum Computers. *arXiv preprint arXiv:1901.11054* (2019).

[82] Prakash Murali, David C McKay, Margaret Martonosi, and Ali Javadi-Abhari. 2020. Software Mitigation of Crosstalk on Noisy Intermediate-Scale Quantum Computers. *arXiv preprint arXiv:2001.02826* (2020).

[83] Yunseong Nam, Jwo-Sy Chen, Neal C Pisenti, Kenneth Wright, Conor Delaney, Dmitri Maslov, Kenneth R Brown, Stewart Allen, Jason M Amini, Joel Apisdorf, et al. 2020. Ground-state energy estimation of the water molecule on a trapped-ion quantum computer. *npj Quantum Information* 6, 1 (2020), 1–6.

[84] Shin Nishio, Yulu Pan, Takahiko Satoh, Hideharu Amano, and Rodney Van Meter. 2019. Extracting Success from IBM's 20-Qubit Machines Using Error-Aware Compilation. *arXiv preprint arXiv:1903.10963* (2019).

[85] Daniel O'Malley, Velimir V Vesselinov, Boian S Alexandrov, and Ludmil B Alexandrov. 2018. Nonnegative/binary matrix factorization with a D-Wave quantum annealer. *PloS one* 13, 12 (2018), e0206653.

[86] Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. 2015. Epidemic processes in complex networks. *Reviews of modern physics* 87, 3 (2015), 925.

[87] Tirthak Patel, Daniel Silver, and Devesh Tiwari. 2022. Geyser: a compilation framework for quantum computing with neutral atoms. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 383–395.

[88] Tirthak Patel and Devesh Tiwari. 2020. Veritas: accurately estimating the correct output on noisy intermediate-scale quantum computers. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 1–16.

[89] Tirthak Patel, Ed Younis, Costin Iancu, Wibe de Jong, and Devesh Tiwari. 2022. QUEST: systematically approximating Quantum circuits for higher output fidelity. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 514–528.

[90] C-K Peng, SV Buldyrev, AL Goldberger, S Havlin, RN Mantegna, M Simons, and HE Stanley. 1995. Statistical properties of DNA sequences. *Physica A: Statistical Mechanics and its Applications* 221, 1-3 (1995), 180–192.

[91] Tianyi Peng, Aram W Harrow, Maris Ozols, and Xiaodi Wu. 2020. Simulating large quantum circuits on a small quantum computer. *Physical Review Letters* 125, 15 (2020), 150504.

[92] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *arXiv preprint arXiv:1801.00862* (2018).

[93] Jiang Qian, Nicholas M Luscombe, and Mark Gerstein. 2001. Protein family and fold occurrence in genomes: power-law behaviour and evolutionary model. *Journal of molecular biology* 313, 4 (2001), 673–681.

[94] Gokul Subramanian Ravi, Kaitlin N Smith, Pranav Gokhale, and Frederic T Chong. 2021. Quantum Computing in the Cloud: Analyzing job and machine characteristics. In *2021 IEEE International Symposium on Workload Characterization (IISWC)*. IEEE, 39–50.

[95] Gokul Subramanian Ravi, Kaitlin N Smith, Prakash Murali, and Frederic T Chong. 2021. Adaptive job and resource management for the growing quantum cloud. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 301–312.

[96] Eleanor G Rieffel, Davide Venturelli, Bryan O'Gorman, Minh B Do, Elicia M Prystay, and Vadim N Smelyanskiy. 2015. A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing* 14, 1 (2015), 1–36.

[97] Anton Robert, Panagiotis Kl Barkoutsos, Stefan Woerner, and Ivano Tavernelli. 2021. Resource-efficient quantum algorithm for protein folding. *npj Quantum Information* 7, 1 (2021), 1–5.

[98] Aritra Sarkar, Zaid Al-Ars, and Koen Bertels. 2021. QuASeR: Quantum Accelerated de novo DNA sequence reconstruction. *Plos one* 16, 4 (2021), e0249850.

[99] Yoshito Sawada and Shinya Honda. 2006. Structural diversity of protein segments follows a power-law distribution. *Biophysical journal* 91, 4 (2006), 1213–1223.

[100] David Sherrington and Scott Kirkpatrick. 1975. Solvable model of a spin-glass. *Physical review letters* 35, 26 (1975), 1792.

[101] Yunong Shi, Nelson Leung, Pranav Gokhale, Zane Rossi, David I Schuster, Henry Hoffmann, and Frederic T Chong. 2019. Optimized compilation of aggregated instructions for realistic quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 1031–1044.

[102] Kaitlin N Smith, Gokul Subramanian Ravi, Prakash Murali, Jonathan M Baker, Nathan Earnest, Ali Javadi-Abhari, and Frederic T Chong. 2021. Error Mitigation in Quantum Computers through Instruction Scheduling. *arXiv preprint arXiv:2105.01760* (2021).

[103] Samuel Stein, Nathan Wiebe, Yufei Ding, Peng Bo, Karol Kowalski, Nathan Baker, James Ang, and Ang Li. 2022. EQC: ensembled quantum computing for variational quantum algorithms. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*. 59–71.

[104] Michael Streif, Sheir Yarkoni, Andrea Skolik, Florian Neukart, and Martin Leib. 2021. Beating classical heuristics for the binary paint shop problem with the

quantum approximate optimization algorithm. *Physical Review A* 104, 1 (2021), 012403.

[105] Kevin J Sung, Jiahao Yao, Matthew P Harrigan, Nicholas C Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R McClean. 2020. Using models to improve optimizers for variational quantum algorithms. *Quantum Science and Technology* 5, 4 (2020), 044008.

[106] Qi Suo, Jin-Li Guo, Shiwei Sun, and Han Liu. 2018. Exploring the evolutionary mechanism of complex supply chain systems using evolving hypergraphs. *Physica A: Statistical Mechanics and its Applications* 489 (2018), 141–148.

[107] Wei Tang, Teague Tomesh, Martin Suchara, Jeffrey Larson, and Margaret Martonosi. 2021. CutQC: using small quantum computers for large quantum circuit evaluations. In *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 473–486.

[108] Swamit S Tannu, Poulami Das, Ramin Ayanzadeh, and Moinuddin K Qureshi. 2022. HAMMER: Boosting Fidelity of Noisy Quantum Circuits by Exploiting Hamming Behavior of Erroneous Outcomes. In *Proceedings of the Twenty-Seventh International Conference on Architectural Support for Programming Languages and Operating Systems*. 529–540.

[109] Swamit S Tannu and Moinuddin K Qureshi. 2019. Not all qubits are created equal: a case for variability-aware policies for NISQ-era quantum computers. In *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. 987–999.

[110] Pontus Vikstål, Mattias Grönkvist, Marika Svensson, Martin Andersson, Göran Johansson, and Giulia Ferrini. 2020. Applying the quantum approximate optimization algorithm to the tail-assignment problem. *Physical Review Applied* 14, 3 (2020), 034009.

[111] Benjamin Villalonga, Dmitry Lyakh, Sergio Boixo, Hartmut Neven, Travis S Humble, Rupak Biswas, Eleanor G Rieffel, Alan Ho, and Salvatore Mandrà. 2020. Establishing the quantum supremacy frontier with a 281 pflop/s simulation. *Quantum Science and Technology* 5, 3 (2020), 034003.

[112] Hengbin Wang. 2019. *Complex Web-API Network Construction Based on Barabasi-Albert Model and Popularity-similarity Optimization Model.* Ph. D. Dissertation. Auckland University of Technology.

[113] Xu Wu, Linlin Zhang, Jia Li, and Ruzhen Yan. 2021. Fractal statistical measure and portfolio model optimization under power-law distribution. *The North American Journal of Economics and Finance* 58 (2021), 101496.

[114] Lei Xie, Jidong Zhai, ZhenXing Zhang, Jonathan Allcock, Shengyu Zhang, and Yi-Cong Zheng. 2022. Suppressing ZZ crosstalk of Quantum computers through pulse and scheduling co-optimization. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. 499–513.

[115] Vladimir Nikolaevich Zadorozhnyi and Evgenii Borisovich Yudin. 2012. Structural properties of the scale-free Barabasi-Albert graph. *Automation and Remote Control* 73, 4 (2012), 702–716.

[116] Stefanie Zbinden, Andreas Bärtschi, Hristo Djidjev, and Stephan Eidenbenz. 2020. Embedding algorithms for quantum annealers with chimera and pegasus connection topologies. In *International Conference on High Performance Computing*. Springer, 187–206.

Received 2022-07-07; accepted 2022-09-22