FrozenQubits:
A report on the methodology and its utility in sparse quantum programs

Baraa Alnassan
Technical University of Munich

Leo Schultheiß
Technical University of Munich

Abstract

FrozenQubits [1] is a technique that boosts the fidelity of Quantum Approximate Optimization Algorithm (QAOA) [3] by skipping hot-spot nodes. In FrozenQubits, a node with the highest degree of connectivity (hot-spot node) which corresponds to qubits with the highest number of CNOTs is frozen 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 improved. This article is intended to give an overview of the aforementioned method, as well as analyze its usefulness in more general quantum circuits.

1 Introduction

Near intermediate scale quantum computers (NISQ computers) pose an emerging field of research as the qubit count of state-of-the-art systems increases. Currently these qubits are still noisy, so larger quantum programs tend to get high error rates. Thus, current development is focused on utilizing present systems to the greatest extent. In this report we will present one of these methods called FrozenQubits and introduce the application of this method on sparse quantum circuits.

2 FrozenQubits

The method we will focus on in this report is called FrozenQubits. It provides a method to improve the execution of quantum approximate optimization algorithm (QAOA) programs on NISQ computers and proposes to do so by trading off smaller circuit size for a increased number of circuits needed to run to compute the result.

2.1 Background

This section will lay the groundwork for the specific optimization problem the original paper had worked on. It is known as the *Ising Hamiltonian*, which as a first step was represented as the QAOA Equation (1), where $z_i \in \{-1,+1\}$ and $h_i, J_{ij}, \text{offset} \in \mathbb{R}$ are given by the Ising statistical mathematical model of ferromagnetism [4].

$$H_Z := C(z) = \sum_{i=0}^{N-1} h_i z_i + \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} J_{ij} z_i z_j + \text{offset} \quad (1)$$

Then the circuit parameters were adjusted to find the optimum value for equation (1), which is computationally expensive on classical computers, but QAOA promises speed-up by accelerating this step.

2.2 Methodology

In this section we will explain, how FrozenQubits promises to speed up QAOA. The method makes use of the tendency for the degree of nodes in a connectivity graph to follow the power law. The real world example given in the paper referenced how just a few airports have flights going to almost every other airport, while most airports are only connected to a few other airports. Similarly most qubits in quantum programs tend to be only connected to a small amount of other qubits, with a handful of qubits being highly interconnected. FrozenQubits makes use of these highly connected qubits by, as the name suggests, "freezing" them. This is implemented by substituting the qubits with -1 and $+1$ in the circuit.
The z_i in the Hamiltonian (1) are represented by individual qubits, with z_k representing the frozen qubit k. As a result the equation gets split into two parts: one with $z_k = +1$ (2) and one with $z_k = -1$ (3).

$$H_Z^{z_k=+1} = \sum_{\substack{i=0 \\ i\neq k}}^{N-1} h_i^{z_k=+1} z_i + \sum_{\substack{i=0 \\ i\neq k}}^{N-1} \sum_{\substack{j=i+1 \\ j\neq k}}^{N-1} J_{ij} z_i z_j + \text{offset}^{z_k=+1} \quad (2)$$

$$H_Z^{z_k=-1} = \sum_{\substack{i=0 \\ i\neq k}}^{N-1} h_i^{z_k=-1} z_i + \sum_{\substack{i=0 \\ i\neq k}}^{N-1} \sum_{\substack{j=i+1 \\ j\neq k}}^{N-1} J_{ij} z_i z_j + \text{offset}^{z_k=-1} \quad (3)$$

As one can see, the most influential change to the Hamiltonian (1), is that every term with z_k was eliminated from the summations. Furthermore the h_i had to be adjusted, to account for the effect of z_k on its neighboring nodes. This can be more intuitively understood by looking at figure 1.

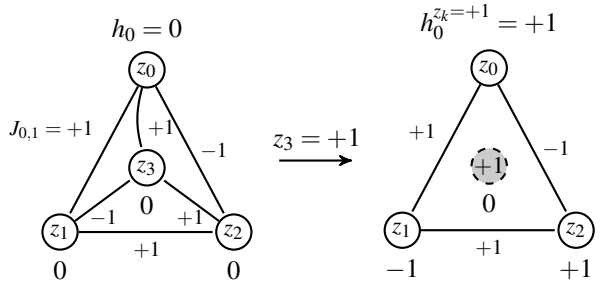

Figure 1: Depiction of substitution of node z_3 with $+1$ and resulting circuit graph with altered topology and h_i.

Each of these resulting equations, where z_k has been substituted, now represent a sub-problem of the original problem (1). Overall the size of each graph got reduced, but the number of graphs doubled. Thus, the quantum circuit size got reduced along with the number of CNOTs, however now both of these equations have to be computed separately and in the end the results need to be recombined. Therefore FrozenQubits creates a trade-off between circuit size and the number of circuits required to be computed.
By applying the method multiple times, thus "freezing" further qubits, the resulting circuit size can be further reduced, however this leads to a doubling of the number of graphs each time the method is applied. So by freezing m qubits 2^m equations are created. The problem graphs for QAOA usually only have very few nodes with a high degree of connectivity, which is why the paper usually chose $m = 1$ or $m = 2$.

2.3 Overhead

As we have shown in the previous paragraph freezing m qubits will result in 2^m sub-problem. This will increase the overhead due to several reasons, one of them being that each sub-graph will have to be compiled for execution on a quantum computer, which leads to an exponential compilation overhead. Another reason is each sub-problem has to be computed separately and its output has to be combined with all other sub-problems to find the solution to the optimization problem.

2.3.1 Compilation Cost

Due to the fact that the Hamiltonian of each sub-problem only varies in the coefficients and offset values FrozenQubits reduces its compilation overhead by only having to compile one circuit and adjusting this circuit to fit the other sub-problems. Therefore all sub-circuits for a specific problem only differ in the angles of their rotation gates. Thus, adjusting the first compiled circuit to fit all other sub-circuits is enough.

2.3.2 Trimming Sub-circuits

As we have seen in section 2.2, when we freeze a qubit, two sub-circuits will be created as a result by substituting the two possibilities $\{-1,+1\}$ in the Hamiltonian equation (1).
When all linear coefficients h_i in the equation are set to zero, the sign of J_{ij} depends solely on the value of the multiplication of z_i and z_j. By substituting these values we get $C(z) = \sum_{i=0}^{N-1} \sum_{j=i+1}^{N-1} {}_{-}^{+} J_{ij}$. From this above we can deduce that the state-space of all Hamiltonian with zero linear coefficients are symmetric, so that $C(z) = C(-z)$. This is also used to further speed up computation of the problem.

2.4 Results

In this section we will discuss the impact of FrozenQubits on the performance of power-law graphs.
In figure (2) the researchers present the results of their method on Barabasi-Albert graphs [2], which are a type of randomly generated power-law graphs, in order to determine its effectiveness. As can be seen in sub-figure (2a), FrozenQubits successfully reduces the number of CNOTs in the resulting quantum circuit with one qubits being frozen ($m = 1$) compared to the original circuit. This reduction continues with one more frozen qubit ($m = 2$). In sub-figure (2b) a similar trend was observed, where increasing the number of frozen qubits directly correlates to a lower circuit depth, compared to the original circuit the method was used on.

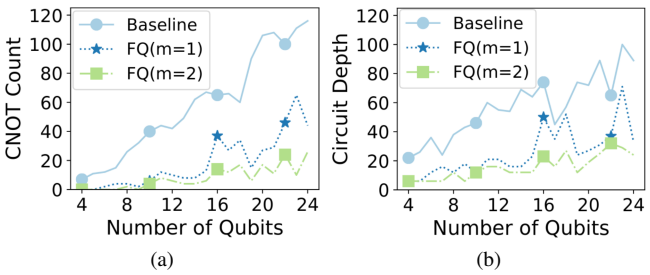

Figure 2: Impact of FrozenQubits on power-law Barabasi-Albert graphs [2] in relation to the Number of CNOTs in the circuit and the circuit depth in relation the number of qubits in the circuit

Furthermore FrozenQubits showed an increase in compilation time of 10% for $m = 1$ compared to the baseline. However, due to the graph construction shortcuts discussed in 2.3, for $m \geq 3$ relative compilation time went below 1, due to

the decreased graph size. The overall trend can be observed in figure (3).

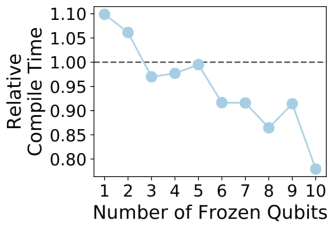

Figure 3: Relative impact on compile time of circuits with FrozenQubits compared to baseline

3 Research

In the following part we will present our own research building on FrozenQubits and how it affects quantum circuits with different topologies than power-law graphs, more specifically *sparse* quantum circuits. Because the method proposed by the paper presents a great opportunity to increase fidelity in power-law graphs it is our goal to generalize this method, to see how useful it might be in more diverse problems. If it would be generally applicable, this would mean that every program could be run at higher fidelity at the cost of more quantum processing. This could be beneficial for many applications requiring a higher degree of resolution in their results not yet reachable/hard to obtain with current systems. We used the number of CNOTs for measuring the impact of FrozenQubits, as they have a large impact on the error rate in circuits and are an easy metric to simulate. An empirical verification of our results in 3.2 on real hardware systems is pending.

3.1 Sparse quantum circuits

For now we will define, what kind of quantum circuits we tested FrozenQubits on. Sparse quantum circuits in this case refers to circuits, in which each qubit is only connected to few other nodes. Because the method presented in the paper was mainly tested on power-law graphs like Barabasi-Albert graphs [2] this is meant to determine the limit of the benefit in fidelity promised by FrozenQubits. Similar to the original work, we will generalize potential real-world problem graphs using two types of topologies. Examples for the types of graphs we used FrozenQubits on can be seen in figure (4) below. Firstly in (4a) we tested 2-regular graphs, as the original paper only covered 3-regular graphs. Additionally an example for randomly generated connected graph (RGCG$_e$) [5] can be seen below (4b), which were generated with a given edge count e and used to more closely resemble real world examples of sparse circuits.

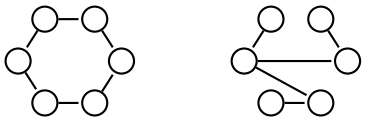

Figure 4: Sparse graph types the following research is based upon. These include 2-regular graphs (4a) and randomly generated connected graphs (4b) (RGCG$_e$ with $e = 5$ here)

3.2 Analysis

We applied FrozenQubits on different types of graphs that were not mentioned in the paper such as 2-regular graphs and randomly generated connected graphs (RGCG$_e$) as shown in figure (4). Similar to the previous work in FrozenQubits, in order to be able to study those graphs, we used $J_{i,j} \in \{-1,+1\}$ and $\forall i.h_i = 0$ and then applied FrozenQubits on them and studied how the number of CNOTs in the resulting graphs responded. The number of CNOTs in the circuits directly corresponded with the number of edges in each graph, with a factor of $\#CNOTs = 2 \times \#edges$. The data for the graphs shown in the following sections was generated using the code in this repository.

3.2.1 2-regular graphs

In the following figure (6), we show our results for analyzing the impact of FrozenQubits on the number of CNOTs in 2-regular graphs. As is expected and can be seen the number of CNOTs gets reduced for every frozen qubit. However, the impact is very reduced in comparison to power-law type graphs, since every qubit in a 2-regular circuit by definition only has two connections. Therefore the number of connections that can be left out with FrozenQubits stays constant, leading to no further speed up when freezing more nodes. Specifically, when freezing m qubits, the number of CNOTs gets reduced by $2 \times 2 \times m = 4m$.

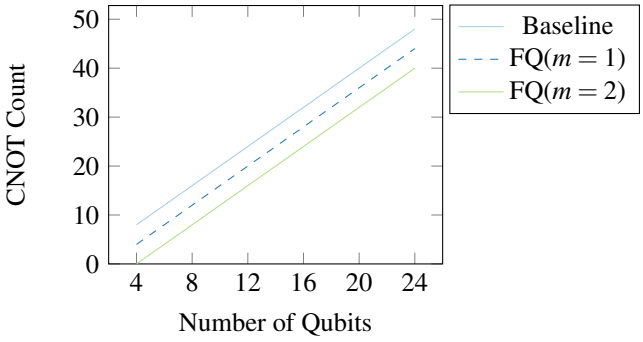

Figure 5: Impact of FrozenQubits on the total number of CNOTs in 2-regular graphs for $m \in \{1,2\}$ compared to the original graph

In figure (6) we visualize this trend again, where significant changes occur with lower qubit counts. One benefit of using FrozenQubits on 2-regular graphs however, may be the creation of disconnected nodes. For $m \geq 2$ it is possible for one or more nodes to be disconnected from their neighbors, leaving trivial circuits to be computed. These could lead to decreased computational requirements, however as these graphs are highly idealized it is reasonable to assume that this has few applications.

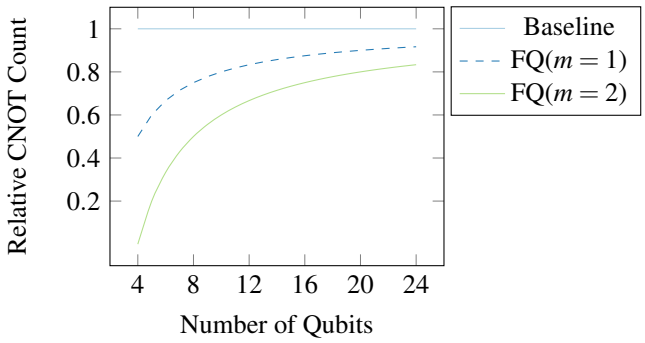

Figure 6: Relative impact of FrozenQubits on the number of CNOTs in 2-regular graphs

3.2.2 Randomly generated connected graphs

In randomly generated connected graphs we are more likely to encounter nodes with a higher degree of connectivity compared to the 2-regular graphs discussed in the last section. Thus, we can expect for FrozenQubits to have a greater impact on the number of CNOTs. This assumption is reaffirmed by figure (7) in that the reduction in the number of CNOTs has almost doubled compared to applying FrozenQubits to 2-regular graphs. However this is of course still far away from the original results in the paper, when applying the method to Barabasi-Albert graphs.

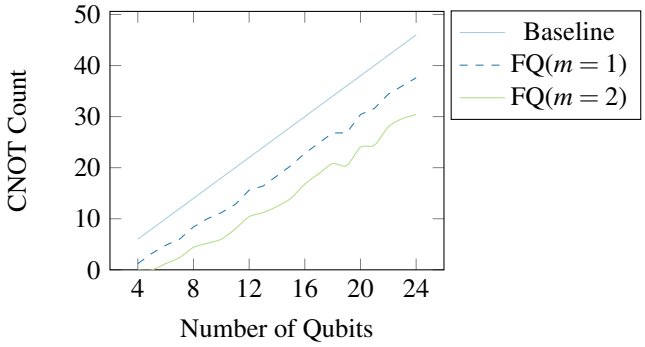

Figure 7: Impact of FrozenQubits on RGCG$_e$, averaged over 5 generated graphs with $e = \#nodes - 1$

In the next graph (8) we portray the relative amount of CNOTs in the graph to the original graph with the same data as in (7). The initial benefit of FrozenQubits stems from the fact that by eliminating one node in a graph with fewer nodes this also removes a larger fraction of the existing edges. However this benefit shrinks as qubit count increases, though not quite as extremely as with 2-regular graphs, because of the nodes with higher degree.

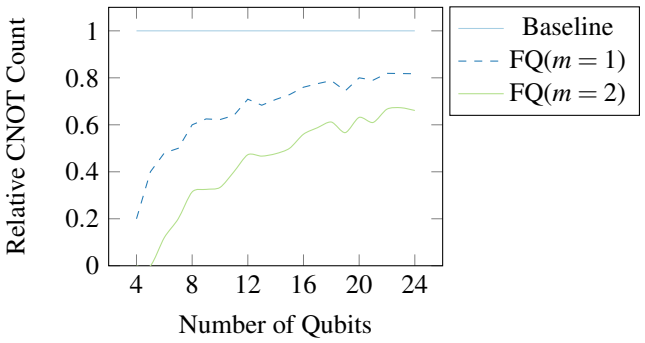

Figure 8: Relative impact of FrozenQubits on CNOT count in RGCG$_e$, averaged over 5 generated graphs with $e = \#nodes - 1$

So overall FrozenQubits can still lead to an improvement in fidelity in less densely connected graphs. However the benefits are a lot less when compared to dense graphs, and importantly they fall off for circuits with more qubits. Therefore FrozenQubits would prove impractical for most sparse circuits.

4 Conclusion

FrozenQubits was introduced as an application-level software framework to improve the fidelity of Quantum Approximation Optimization Algorithm (QAOA), it works by freezing a qubit or more by substituting the two possibilities in the problem, which partitions the state-space of the problem into smaller sub-spaces. multiple strategies were developed to subside the problem of running the newly created sub-problems individually, such as using the symmetry of the created sub-circuits. In this paper we discussed the impact of FrozenQubits on sparser circuits with lower average connectivity degrees than the ones discussed in the original paper. We discussed the impact of FrozenQubits on sparser circuits on the number of CNOTs. Overall the method produces significantly worse results with sparse graphs than with power-law type graphs leading to minor improvements in fidelity at the cost of significant computational overhead.

References

[1] Ramin Ayanzadeh, Narges Alavisamani, Poulami Das, and Moinuddin Qureshi. 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 2023, page 311–324, New York, NY, USA, 2023. Association for Computing Machinery.

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

[3] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[4] Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik*, 31(1):253–258, Feb 1925.

[5] David Bruce Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, page 296–303, New York, NY, USA, 1996. Association for Computing Machinery.