Bubbles in Quantum Devices

Seyed Sajad Kahani[1] and Dan Browne[1]

[1]Department of Physics and Astronomy, UCL, London WC1E 6BT, UK

July 28, 2024

Abstract

Optimizing qubit placement in quantum devices is crucial for efficient circuit execution. Current approaches often prioritize dense packing, neglecting the potential benefits of strategic ancilla qubit placement. This paper introduces the concept of "bubbles," where ancilla qubits are strategically positioned to reduce routing costs. We have developed an algorithm for optimally allocating qubits within a simplified cost model. Our results demonstrate that while the impact on NISQ devices can reach 8%, there is potential for significant cost reduction in larger, higher-dimensional architectures. This is also a meaningful result that demonstrates the importance of having bus qubits in quantum devices.

Keywords: Quantum Computing, Circuit Transformation, Qubit Allocation, Qubit Routing, Quantum Device Architecture

1 Introduction

As quantum computing advances towards practical applications, the need for optimized compilation processes becomes increasingly critical [5]. While many aspects of quantum compilation have reached maturity, the circuit transformation to adapt to device connectivity constraints remains a significant challenge due to the large search space and unclear objectives.

To address this challenge, researchers have defined various sub-problems. One such sub-problem is finding an initial placement of logical qubits onto physical qubits, also known as qubit allocation [15, 14, 13, 10, 7, 16]. A study suggests that the initial placement could affect the total cost of a circuit by 10% in realistic scenarios [14]. Moreover, by introducing additional concepts like partial placement and layers in the circuit, this problem will be useful for solving the entire circuit transformation problem [7, 4].

Despite the simplification, qubit allocation is an NP-hard problem [15]. Unsurprisingly, the most promising results come from heuristics that do not guarantee optimal placement. While these heuristics and algorithms differ, they share a common trait: they tend to pack qubit placements as densely as possible, leaving no ancilla qubits in between.

A closely related problem is implementing two-qubit gates between non-adjacent qubits, often referred to as routing. Recent research trends [5] aim to reduce routing costs by simplifying SWAP gates and exploring alternative routing methods. These techniques include embedding SWAP gates within other two-qubit gates [9], compiling certain circuit classes directly into CNOT gates [11, 8], and using Hamiltonians instead of SWAP gates [1], among others.

In this work, we present a novel approach to qubit allocation based on a simplification rule for routing over ancillas. We explore the concept of "bubbles," which are ancilla qubits surrounded by data qubits in a quantum device. We demonstrate that these bubbles have the potential to reduce the overall cost of the circuit. For simplicity, we frame this idea within a basic cost model, though the concept is applicable to various compilation frameworks. Based on this cost model, we propose a placement algorithm and evaluate its potential on different Noisy Intermediate-Scale Quantum (NISQ) devices. Our results show a cost reduction of 1-2% in real-world lattice-based NISQ devices, while achieving 5-8% in hierarchical devices, with the potential for more than 10% reduction in future higher-dimensional devices.

2 Background

To provide the necessary context for understanding the concepts and algorithms introduced in this paper, we begin with an overview of quantum device architectures and their constraints, and discuss the impact

of ancilla qubits on routing operations. After that, we will introduce relevant graph theoretical concepts.

2.1 Quantum Devices

Quantum devices have specific architectural constraints that impact qubit connectivity and operations. These constraints are often represented as a graph called the device graph.

Definition 1. *A quantum device graph $G = (N, E)$ is a graph where N is the set of nodes representing qubits and $(u, v) \in E$ if and only if we can directly perform a two-qubit gate between qubits u and v.*

To perform a two-qubit gate between two non-adjacent qubits, we often use a sequence of SWAP gates to move the qubits to adjacent positions. For two qubits with a distance of d in the device graph, the cost of swapping them is $6d-3$ CNOT gates. However, we can further optimize the cost of swapping non-adjacent qubits when ancilla qubits are present along the path. Figure 1 illustrates this optimization.

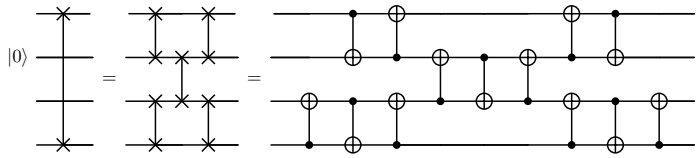

Figure 1: Optimization of swapping two qubits with an ancilla on the path.

Furthermore, this optimization can be generalized, as stated in the following proposition.

Proposition 1. *If the distance between two qubits on a device graph is d and the shortest path between them consists of k ancilla qubits, then the cost of swapping these two qubits is $6d - 2k - 3$ CNOT gates.*

This proposition can be proved by induction, though we omit the proof for brevity.

2.2 Graph Theoretical Concepts

To analyse the placement of qubits on quantum devices, we utilize several graph theoretical concepts. Two key measures are betweenness centrality and farness.

Betweenness centrality is a measure defined on each node of a graph, representing the number of shortest paths passing through it. This measure finds numerous applications in network analysis, revealing the impact of a node on information flow and other network dynamics [12].

The original definition of betweenness centrality [6] differs slightly from our current context. We have tailored our definition to account for scenarios where multiple shortest paths exist between two nodes, allowing for various interpretations of betweenness. We introduce a definition that aligns with the original definition for $\alpha = 1$. Additionally, in the extreme cases where α approaches infinity or zero, the definition respectively covers the count of pairs where all or some of their shortest paths pass through the node.

Definition 2. *Given a graph $G(N, E)$ and a set of occupied nodes $R \subseteq N$, α-betweenness centrality of a node v (written as $B_\alpha(v)$) is defined as*

$$B_\alpha(v) := \sum_{s,t \in R, s,t \neq v} \left(\frac{\sigma_{s,t|v}}{\sigma_{s,t}} \right)^\alpha, \tag{1}$$

where $\sigma_{s,t|v}$ is the number of shortest paths from s to t that go through v, and $\sigma_{s,t}$ is the total number of shortest paths from s to t.

2.2.1 Farness

Another useful concept is farness, which measures how "far" a node is from all other nodes in the graph.

Definition 3. *Given a graph $G(N, E)$ and a set of occupied nodes $R \subseteq N$, the farness of a node v (written as $F(v)$) is defined as*

$$F(v) := \sum_{u \in R, u \neq v} d_G(u, v), \tag{2}$$

where $d_G(u, v)$ is the shortest path distance between nodes u and v.

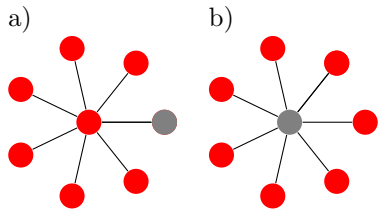

Figure 2: Two different placements of 8 qubits on a star graph. The red dots are the occupied qubits and the grey dots are the ancilla qubits.

3 Discussion

While the qubit allocation problem itself has a specific objective to minimize, which is the cost of the circuit, we introduce a simpler cost function in this work. This new cost function can be viewed as a special case of the general cost function for a circuit consisting of highly non-local operations. We have chosen this specific cost function to facilitate analysis and demonstrate the potential of bubbles in a simplified setting.

The cost function we introduce is the average swap cost of a placement and is defined as follows.

Definition 4 (Average Swap Cost)**.** *The average swap cost of a placement $\varphi : Q \to R \subseteq N$, where Q is the set of logical qubits of the circuit and N are the nodes of the device graph, is defined as the average complexity of implementing a SWAP gate between any two qubits in Q.*

$$\mathcal{C}[\varphi] := \frac{2}{|Q|(|Q|-1)} \sum_{q,q' \in Q} \textit{cost of swapping}[\varphi(q), \varphi(q')] \tag{3}$$

With this definition, we can examine examples where a non-trivial placement with ancillas surrounded by data qubits proves more beneficial. The first example is a star graph, described in the following and depicted in Figure 2.

Example 1 (Star Graph)**.** *Consider a star graph S_k as the device graph, where we aim to allocate $k-1$ qubits. By setting the central qubit as an ancilla, the average swap cost will be 7, while if we allocate the central qubit, which would be a more compact placement, the average swap cost will be $9 - \frac{12}{k-1}$, which is greater than 7 for $k > 7$.*

While this may seem to be an unrealistic device graph, a similar phenomenon can be observed in a hypercube graph. Assuming two placements for $|Q|$ logical qubits into a d-dimensional hypercube graph; first, mapping them densely, and second, having one ancilla per k^d qubits uniformly, resulting in a larger device graph. The average swap cost for the second placement, in comparison to the first one, will be reduced by a factor of $\left(1 - \frac{1}{3k}\right)\left(1 + \frac{1}{dk^d}\right)$.

Theorem 1 (Hypercube)**.** *For a d-dimensional hypercube with N qubits, the average swap cost of a full placement will be*

$$\mathcal{C}[\varphi] = 2N^{1/d} d \frac{1 - N^{-2/d}}{1 - N^{-1}} - 3. \tag{4}$$

For a d-dimensional hypercube with $N + \frac{N}{k^d}$ that has N logical qubits and $\frac{N}{k^d}$ ancilla qubits, uniformly distributed in the hypercube, the average swap cost will be

$$\mathcal{C}[\varphi'] \leq \left(1 - \frac{1}{3k}\right)\left(1 + \frac{1}{dk^d}\right)\mathcal{C}[\varphi] + 6kd. \tag{5}$$

This implies that for a sufficiently large hypercube, even within a reasonable dimension (2 or 3), we can achieve a 6% to 13% reduction in the average swap cost using this simple technique.

To visualize the effect of bubbles, we have depicted the distribution of costs for two different placements on a Chimera graph [3], a dense placement and another one with bubbles generated by the algorithm we will introduce in the next section in Figure 3. One can observe that by introducing bubbles, the distribution changes significantly, and the average cost is reduced.

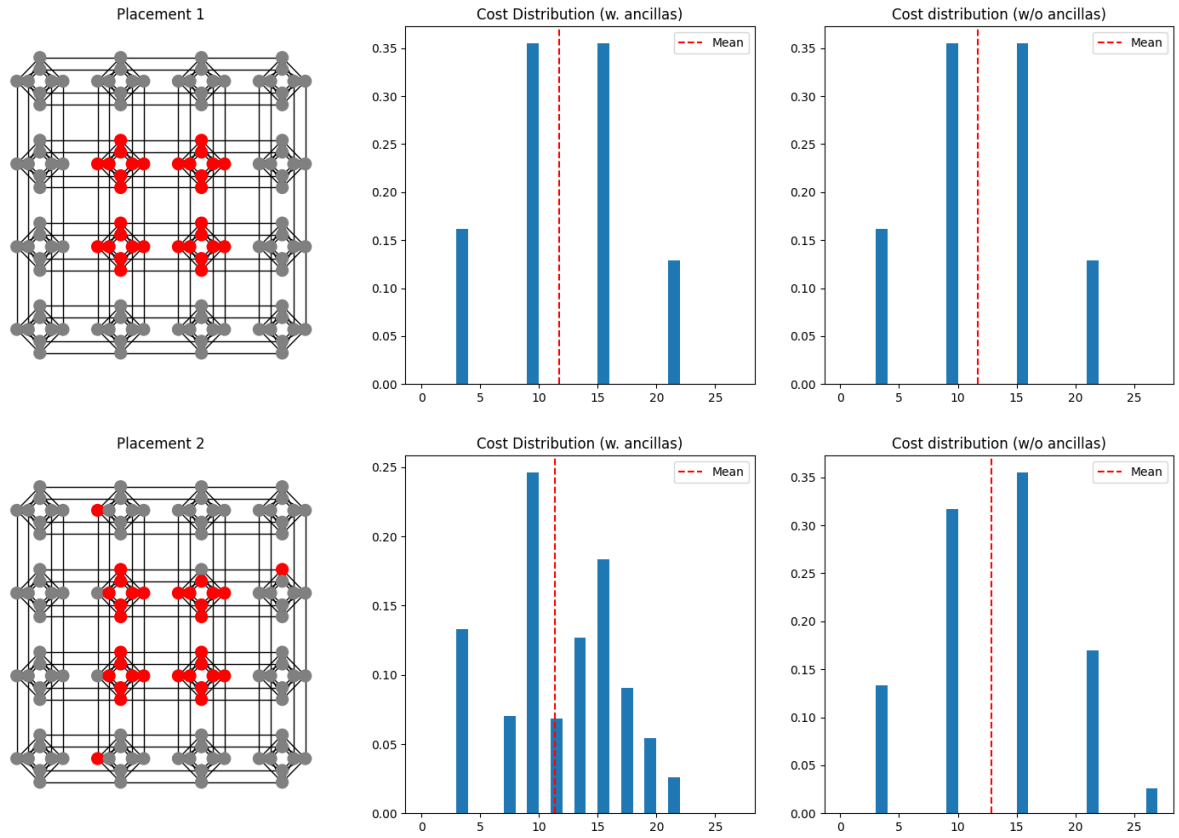

Figure 3: Swap cost distribution for different placements of the same number of qubits on a Chimera$(4, 4, 4)$ graph device.

Based on the improvement in the average swap cost, one might inquire about systematically generating these placements to achieve the optimal placement with the highest possible improvement. We will address this question in the subsequent section.

3.1 Algorithmic Generation of Bubbles

Before proceeding further, we need to define a weighted graph that represents the cost of swapping qubits in a quantum device.

Definition 5. *Given a device graph G and a set of occupied nodes R, $H_{G,R}$ is defined as a weighted graph, with the nodes and edges of G and weights defined as*

$$w_H(u,v) := \begin{cases} 6 & u,v \notin R \\ 5 & u \in R, v \notin R \text{ or } u \notin R, v \in R \\ 4 & u,v \in R \end{cases}. \tag{6}$$

As a consequence of this definition, the cost of swapping two qubits in a device graph, taking Proposition 1 into account, can be calculated as follows.

Proposition 2. *Given a device graph G and a placement $\varphi : Q \to R \subseteq N$, the cost of swapping q and q' is equal to $d_{H_{G,\varphi(Q)}}(\varphi(q), \varphi(q')) - 3$, where $\varphi(Q)$ is the image of Q under φ.*

Building upon this, we can prove a theorem that simplifies the calculation of the reduction in average swap cost when replacing a qubit with an ancilla.

Theorem 2. *Assume a device graph G and a placement $\varphi : Q \to N$ and then a replacement on top of it that leads to*

$$\varphi'(i) = \begin{cases} b & \text{if } \varphi(i) = a \\ \varphi(i) & \text{otherwise} \end{cases}. \tag{7}$$

Then, the difference between the average swap cost of φ and φ' is given by

$$\mathcal{C}[\varphi] - \mathcal{C}[\varphi'] = \frac{2}{|Q|(|Q|-1)}(2B_0(a) + F(a) - 2B'_\infty(b) - F'(b) - |Q| + 1) \tag{8}$$

where B_0 and F are the betweenness and farness for the cost graph $H = H_{G,\varphi(Q)}$, and B'_∞ and F' are the betweenness and farness for the cost graph $H' = H_{G,\varphi(Q)\setminus\{a\}}$.

This theorem can be used to design an algorithm that iteratively optimizes an initial placement. Algorithm 1 optimizes the initial placement one step further by a replacement and guarantees that the step is optimal. The complexity of this algorithm is $O(|Q|\|G\|^3)$ [2].

Algorithm 1 Step-by-step placement optimization

```
function ExactOptimization(G, φ, Q)
    Precalculate B_0, F and d_H for H_{G,φ(Q)}
    candidates ← {}
    for a ∈ φ(Q) do
        aTerms ← 2B_0(a) + F(a) − |Q| + 1
        R* ← φ(Q) \ {a}
        Precalculate B'_∞, F' for H_{G,R*}
        for b ∈ N \ φ(Q) do
            bTerms ← 2B'_∞(b) + F'(b)
            candidates ← candidates ∪ {(a, b, aTerms − bTerms)}
        end for
    end for
    (a, b, cost) ← arg max_{(a,b,c)∈candidates} c
    if cost > 0 then
        return φ with a replaced by b
    else
        return φ
    end if
end function
```

The proposed algorithm may not be practical for large circuits. In such cases, we can use the following lower bound formula that can be precalculated for any pair, making the algorithm run in $O(\|G\|^3)$, which is one order of magnitude faster, while sacrificing the guarantee of optimality.

Corollary 1. *Given two placements φ and φ' that differ in a single qubit replacement, the difference in the average swap cost can be bounded by the following formula*

$$\begin{aligned}\mathcal{C}[\varphi] - \mathcal{C}[\varphi'] \geq \frac{2}{|Q|(|Q|-1)}\big(&2B_0(a) + F(a) - 2B_0(b) - F(b)\\ &+ d_H(a,b) - |Q|\big),\end{aligned} \tag{9}$$

where B_0 and F are the betweenness and farness for the cost graph $H = H_{G,\varphi(Q)}$.

Proof. By bounding $B'_\infty(b)$ and $F'(b)$ from Theorem 2, we can prove this corollary.

First, because all the weights in H' are less than or equal to those in H,

$$\begin{aligned}F'(b) &= \sum_{u\in\varphi(Q)\setminus\{a\}} d_{H'}(u,b)\\ &\leq \sum_{u\in\varphi(Q)} d_H(u,b) - d_{H'}(a,b)\\ &\leq F(b) - d_H(a,b) + 1.\end{aligned} \tag{10}$$

Second, noting that for any pair (u, v) where all of their shortest paths in H' pass through b (counted in $B'_\infty(b)$), there certainly exists a shortest path in H that passes through b as well. This can be shown separately for the cases where all of their shortest paths in H' pass through a or not.

Formally,

$$B'_\infty(b) \leq B_0(b). \tag{11}$$

Combining these two inequalities, we obtain the desired result. □

Even further, by removing the term $d_H(a, b)$ from the lower bound, we can visualize it with two scores for removing and adding nodes to the occupied nodes. Figure 4 shows these scores for a placement on the Sycamore device.

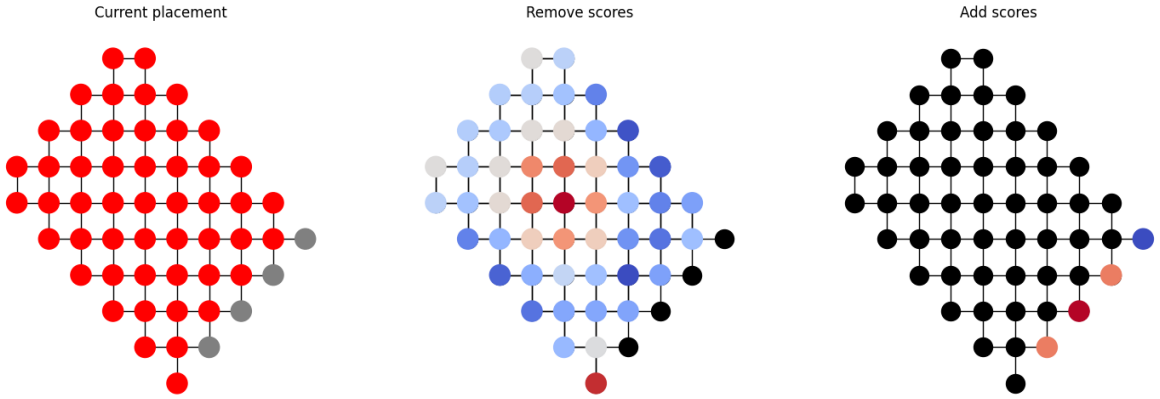

Figure 4: Scores of the nodes in current placement by $B(v) + F(v)$ and the nodes outside the placement by $-B(v) - F(v)$.

4 Results

We tested Algorithm 1 on Google's Sycamore, IBM's Eagle, and D-Wave's One quantum devices. While D-Wave is not a circuit-based quantum device, its device graph serves as a valuable example of a hierarchical architecture in quantum devices.

Starting with an initial placement generated by breadth-first search (BFS), the algorithm iteratively produces a sequence of placements. The improvement in average swap cost relative to the initial placement is depicted in the first column of Figure 5 for different numbers of qubits across different devices.

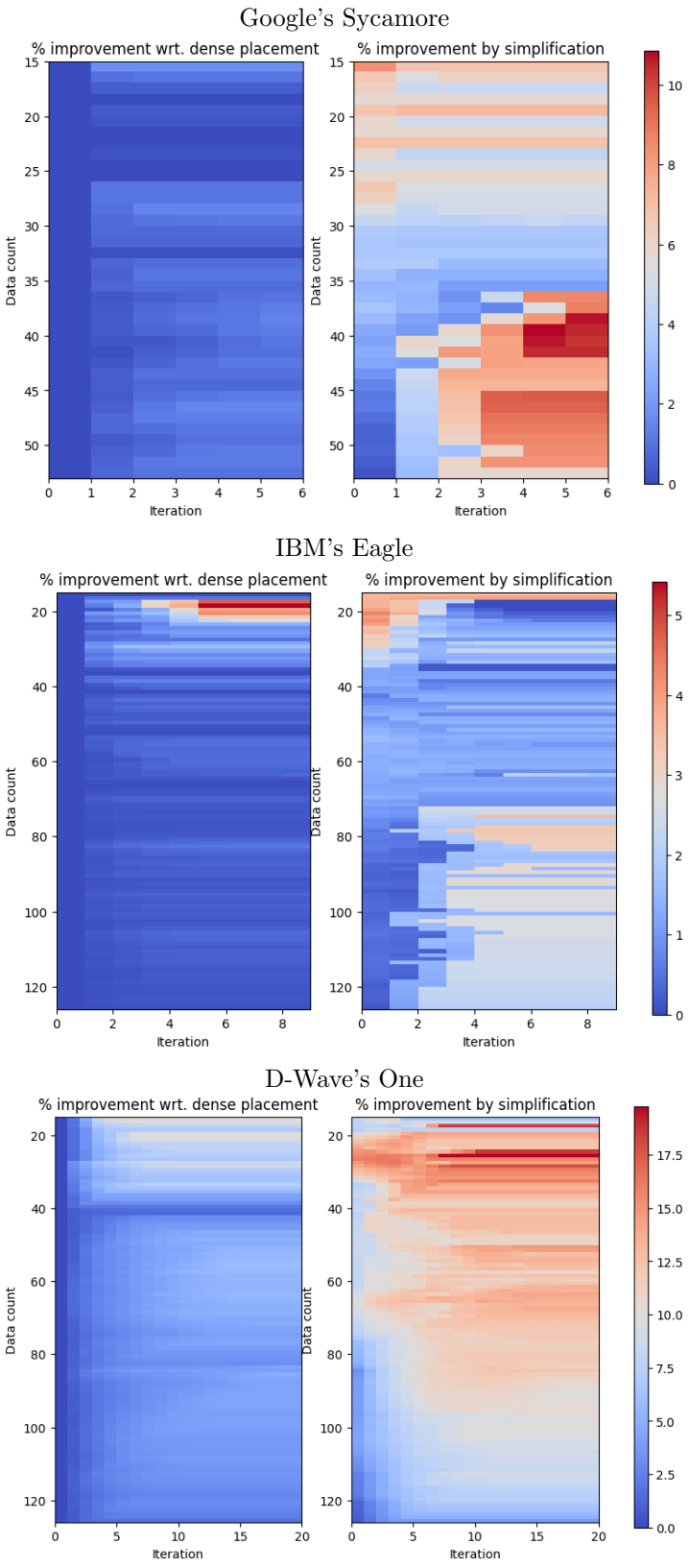

Figure 5: Left column: Improvement in average swap cost relative to the baseline BFS placement cost. Right column: Improvement in average swap cost of a placement relative to the same case without ancilla simplification.

To isolate the contribution of bubbles to the observed improvement, we compared the placement cost with and without ancilla simplification for each placement in the sequence. This analysis quantifies the

impact of bubbles on placement quality. The results are presented in the second column of Figure 5.

An improvement can be attributed to the existence of bubbles in the device graph if a corresponding increase is observed in the second column of Figure 5. This trend is evident in Google's Sycamore with 40-50 data qubits and IBM's Eagle with 50 or more data qubits. Due to its fundamentally different architecture, D-Wave's One exhibits significantly larger improvements attributable to bubbles, even with fewer than 20 data qubits.

Overall, we observed a 5-8% improvement in average swap cost for D-Wave's One, compared to approximately 1-2% for Google's Sycamore and IBM's Eagle.

5 Conclusion

In this paper, we introduced the novel concept of "bubbles" in quantum device qubit allocation - the strategic placement of ancilla qubits surrounded by data qubits to reduce overall circuit costs. Our analysis demonstrates that this approach can yield meaningful improvements in routing efficiency, particularly for larger and more complex quantum architectures.

Utilizing an algorithm we developed for optimizing qubit placement with bubbles, we have uncovered several key findings. For current lattice-based devices such as Google's Sycamore and IBM's Eagle, bubbles provide modest but measurable benefits, reducing average swap costs by 1-2%. More significant improvements of 5-8% were observed for hierarchical architectures like D-Wave's One, even with relatively few data qubits.

Furthermore, theoretical analysis suggests potential cost reductions of 13% for large 3D lattice architectures can be achieved with a simple allocation, indicating that bubbles may become increasingly beneficial as quantum devices scale up.

These results open up several promising avenues for future research. One direction is generalizing the simplified cost model based on swapping pairs into general permutations. This generalization may connect to the routing number [4], which is well-studied in the literature. Additionally, incorporating bubbles into broader frameworks for qubit allocation and circuit transformation presents an interesting area for further investigation.

Finally, this effect could also be interpreted as demonstrating the benefits of having bus qubits in quantum devices. This suggests that even though a qubit can be used for data storage and computation, it may be more beneficial in some cases to dedicate it solely to data transfer. This finding has implications for the design and optimization of future quantum architectures.

References

[1] Aniruddha Bapat et al. "Advantages and Limitations of Quantum Routing". In: *PRX Quantum* 4 (1 Feb. 2023), p. 010313. DOI: 10.1103/PRXQuantum.4.010313. URL: https://link.aps.org/doi/10.1103/PRXQuantum.4.010313.

[2] Ulrik Brandes. "A faster algorithm for betweenness centrality∗". In: *Journal of Mathematical Sociology* (June 2001). URL: https://www.tandfonline.com/doi/abs/10.1080/0022250X.2001.9990249.

[3] Jun Cai, William G. Macready, and Aidan Roy. *A practical heuristic for finding graph minors*. 2014. arXiv: 1406.2741 [quant-ph]. URL: https://arxiv.org/abs/1406.2741.

[4] Andrew M. Childs, Eddie Schoute, and Cem M. Unsal. "Circuit Transformations for Quantum Architectures". In: *arXiv* (Feb. 2019). DOI: 10.4230/LIPIcs.TQC.2019.3. eprint: 1902.09102.

[5] Jason Cong. "Lightning Talk: Scaling Up Quantum Compilation – Challenges and Opportunities". In: *2023 60th ACM/IEEE Design Automation Conference (DAC)*. 2023, pp. 1–2. DOI: 10.1109/DAC56929.2023.10247677.

[6] Linton C. Freeman. "A Set of Measures of Centrality Based on Betweenness". In: *Sociometry* 40.1 (1977), pp. 35–41. ISSN: 00380431. URL: http://www.jstor.org/stable/3033543 (visited on 04/11/2024).

[7] Toshinari Itoko et al. "Optimization of Quantum Circuit Mapping using Gate Transformation and Commutation". In: *arXiv* (July 2019). DOI: 10.48550/arXiv.1907.02686. eprint: 1907.02686.

[8] Aleks Kissinger and Arianne Meijer-van de Griend. *CNOT circuit extraction for topologically-constrained quantum memories*. arXiv:1904.00633. ZSCC: 0000062 type: article. arXiv, May 26, 2019. arXiv: 1904.00633[quant-ph]. URL: http://arxiv.org/abs/1904.00633 (visited on 07/19/2023).

[9] Lingling Lao and Dan E. Browne. *2QAN: A quantum compiler for 2-local qubit Hamiltonian simulation algorithms*. Nov. 7, 2021. arXiv: 2108.02099.

[10] Gushu Li, Yufei Ding, and Yuan Xie. "Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices". In: *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '19: Architectural Support for Programming Languages and Operating Systems. Providence RI USA: ACM, Apr. 4, 2019, pp. 1001–1014. ISBN: 978-1-4503-6240-5. DOI: 10.1145/3297858.3304023. (Visited on 11/11/2022).

[11] Beatrice Nash, Vlad Gheorghiu, and Michele Mosca. "Quantum circuit optimizations for NISQ architectures". In: *Quantum Science and Technology* 5.2 (Mar. 31, 2020). ZSCC: 0000081, p. 025010. ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab79b1. arXiv: 1904.01972[quant-ph]. URL: http://arxiv.org/abs/1904.01972 (visited on 03/24/2023).

[12] Mark Newman. *Networks: An Introduction*. Oxford University Press, Mar. 2010. ISBN: 9780199206650. DOI: 10.1093/acprof:oso/9780199206650.001.0001. URL: https://doi.org/10.1093/acprof:oso/9780199206650.001.0001.

[13] Siyuan Niu et al. "A Hardware-Aware Heuristic for the Qubit Mapping Problem in the NISQ Era". In: *IEEE Transactions on Quantum Engineering* 1 (2020), pp. 1–14. DOI: 10.1109/TQE.2020.3026544.

[14] Alexandru Paler. "On the Influence of Initial Qubit Placement During NISQ Circuit Compilation". In: *arXiv* (Nov. 2018). DOI: 10.48550/arXiv.1811.08985. eprint: 1811.08985.

[15] Marcos Yukio Siraichi et al. "Qubit allocation". In: *CGO 2018: Proceedings of the 2018 International Symposium on Code Generation and Optimization*. New York, NY, USA: Association for Computing Machinery, Feb. 2018, pp. 113–125. ISBN: 978-1-45035617-6. DOI: 10.1145/3168822.

[16] Chi Zhang et al. "Time-optimal Qubit mapping". In: *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. ASPLOS '21: 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems. Virtual USA: ACM, Apr. 19, 2021, pp. 360–374. ISBN: 978-1-4503-8317-2. DOI: 10.1145/3445814.3446706. (Visited on 11/11/2022).

Appendix A

Theorem 3 (Hypercube)**.** *For a d-dimensional hypercube with N qubits, the average swap cost of a full placement will be*

$$\mathcal{C}[\varphi] = 2N^{1/d} d \frac{1 - N^{-2/d}}{1 - N^{-1}} - 3. \tag{12}$$

For a d-dimensional hypercube with $N + \frac{N}{k^d}$ *that has N logical qubits and* $\frac{N}{k^d}$ *ancilla qubits, uniformly distributed in the hypercube, the average swap cost will be*

$$\mathcal{C}[\varphi'] \leq \left(1 - \frac{1}{3k}\right)\left(1 + \frac{1}{dk^d}\right)\mathcal{C}[\varphi] + 6kd. \tag{13}$$

Proof. We begin with the definition of the average swap cost

$$\mathcal{C}[\varphi] := \frac{2}{|Q||Q-1|} \sum_{q,q' \in Q} \text{cost of swapping}[\varphi(q), \varphi(q')], \tag{14}$$

applying Proposition 1, we obtain

$$\mathcal{C}[\varphi] = \frac{2}{|Q||Q-1|} \sum_{q,q' \in Q} 6 d_G(\varphi(q), \varphi(q')) - 3. \tag{15}$$

In the absence of ancilla qubits, the average distance between any two nodes in the graph is $\frac{1}{3}ld\frac{1-l^{-2}}{1-l^{-d}}$, where l is the length of the hypercube. This leads to

$$\mathcal{C}[\varphi] = 2N^{1/d}d\frac{1-N^{-2/d}}{1-N^{-1}} - 3. \tag{16}$$

For the case with ancilla qubits, we first note that the grid expands to a length l'

$$l' = \left\lceil \sqrt[d]{N + \frac{N}{k^d}} \right\rceil. \tag{17}$$

We partition the hypercube into k^d zones, each containing one ancilla qubit. We propose a path for each pair of qubits that passes through the ancilla qubits of their respective zones and all zones in between. This yields an upper bound for the average swap cost

$$\mathcal{C}[\varphi'] \leq \frac{2}{|Q||Q-1|}\sum_{q,q'} 6\big(d_G(q,a_q) + d_G\big(a_q,a'_q\big) + d_G\big(a'_q,q'\big)\big) - 2\bigg(\frac{d_G(a_q,a'_q)}{k}+1\bigg) - 3, \tag{18}$$

here, a_q and a'_q denote the ancilla qubits in the zones of q and q', respectively. The average distance from a node to its ancilla is $\frac{d}{2}(k-1)$, and the average distance between any two ancilla qubits is $\frac{1}{3}l'd\frac{1-\left(\frac{l'}{k}\right)^{-2}}{1-\left(\frac{l'}{k}\right)^{-d}}$. This leads to

$$\mathcal{C}[\varphi'] \leq 2\bigg(1-\frac{1}{3k}\bigg)l'd\frac{1-\left(\frac{l'}{k}\right)^{-2}}{1-\left(\frac{l'}{k}\right)^{-d}} + 6kd - 6d - 5. \tag{19}$$

Assuming $d > 1$ and $k > 1$, we can show that $\frac{l'}{k} < N^{1/d}$, which implies

$$\frac{1-\left(\frac{l'}{k}\right)^{-2}}{1-\left(\frac{l'}{k}\right)^{-d}} \leq \frac{1-N^{-2/d}}{1-N^{-1}}, \tag{20}$$

consequently

$$\mathcal{C}[\varphi'] \leq 2\bigg(1-\frac{1}{3k}\bigg)l'd\frac{1-N^{-2/d}}{1-N^{-1}} + 6kd - 6d - 5. \tag{21}$$

Given that

$$l' = N^{1/d}(1+\frac{1}{dk^d}) + 1, \tag{22}$$

We arrive at

$$\mathcal{C}[\varphi'] \leq \bigg(1-\frac{1}{3k}\bigg)(1+\frac{1}{dk^d})2N^{1/d}d\frac{1-N^{-2/d}}{1-N^{-1}} + 6kd - 4d - 5. \tag{23}$$

Finally, we obtain the desired upper bound:

$$\mathcal{C}[\varphi'] \leq \bigg(1-\frac{1}{3k}\bigg)\bigg(1+\frac{1}{dk^d}\bigg)\mathcal{C}[\varphi] + 6kd. \tag{24}$$

□

Appendix B

Theorem 4. *Assume a device graph G and a placement $\varphi : Q \to N$ and then a replacement on top of it that leads to*

$$\varphi'(i) = \begin{cases} b & \text{if } \varphi(i) = a \\ \varphi(i) & \text{otherwise} \end{cases}. \tag{25}$$

Then, the difference between the average swap cost of φ and φ' is given by

$$\mathcal{C}[\varphi] - \mathcal{C}[\varphi'] = \frac{2}{|Q|(|Q|-1)}(2B_0(a) + F(a) - 2B'_\infty(b) - F'(b) - |Q| + 1) \tag{26}$$

where B_0 and F are the betweenness and farness for the cost graph $H = H_{G,\varphi(Q)}$, and B'_∞ and F' are the betweenness and farness for the cost graph $H' = H_{G,\varphi(Q)\setminus\{a\}}$.

Proof. By defining H'' as $H_{G,\varphi'(Q)}$ we can write the average swap cost of φ and φ' as follows, using Proposition 2.

$$\mathcal{C}[\varphi] = -3 + \frac{2}{|Q|(|Q|-1)} \sum_{u,v \in \varphi(Q)} d_H(u,v), \tag{27}$$

$$\mathcal{C}[\varphi'] = -3 + \frac{2}{|Q|(|Q|-1)} \sum_{u,v \in \varphi'(Q)} d_{H''}(u,v). \tag{28}$$

Noting that

$$\varphi'(Q) \setminus \{b\} = \varphi(Q) \setminus \{a\} = \varphi(Q) \cap \varphi'(Q), \tag{29}$$

we can write the difference between them as

$$\begin{aligned} \mathcal{C}[\varphi] - \mathcal{C}[\varphi'] = \frac{2}{|Q|(|Q|-1)} \Bigg(& \sum_{u,v \in \varphi(Q) \cap \varphi'(Q)} d_H(u,v) - d_{H''}(u,v) \\ & + \sum_{v \in \varphi(Q)} d_H(v,a) - \sum_{v \in \varphi'(Q)} d_{H''}(v,b) \Bigg). \end{aligned} \tag{30}$$

To simplify the first summation, we will use these two following facts that hold for any $u, v \in \varphi(Q) \cap \varphi'(Q)$.

$$\begin{cases} d_H(u,v) = d_{H'}(u,v) - 2 \text{ if some of the shortest paths of } (u,v) \text{ in } H \text{ pass through } a \text{ ,} \\ d_H(u,v) = d_{H'}(u,v) \text{ otherwise .} \end{cases} \tag{31}$$

$$\begin{cases} d_{H''}(u,v) = d_{H'}(u,v) + 2 \text{ if all the shortest paths of } (u,v) \text{ in } H' \text{ pass through } b \text{ ,} \\ d_{H''}(u,v) = d_{H'}(u,v) \text{ otherwise .} \end{cases} \tag{32}$$

As a result, we can write the first summation as

$$\sum_{u,v \in \varphi(Q) \cap \varphi'(Q)} d_H(u,v) - d_{H''}(u,v) = 2B_0(a) - 2B'_\infty(b). \tag{33}$$

The second summation is simply equal to $F(a)$, and the third summation will be simplified to

$$\sum_{v \in \varphi'(Q)} d_{H''}(v,b) = \sum_{v \in \varphi'(Q)} (d_{H'}(v,b) + 1) = F'(b) + |Q| - 1. \tag{34}$$

Combining these results, we obtain the desired result. □