Towards Fidelity-Optimal Qubit Mapping on NISQ Computers

Sri Khandavilli*, Indu Palanisamy*, Manh V. Nguyen*, Thinh V. Le*, Tu N. Nguyen*, and Thang N. Dinh†
*Department of Computer Science, Kennesaw State University, Marietta, GA 30060, USA.
†Department of Computer Science, Virginia Commonwealth University, VA 23284 USA.

2023 IEEE International Conference on Quantum Computing and Engineering (QCE) | 979-8-3503-4323-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/QCE57702.2023.00019

***Abstract*—Quantum computing is gaining momentum in revolutionizing the way we approach complex problem-solving. However, the practical implementation of quantum algorithms remains a significant challenge due to the error-prone and hardware limits of near-term quantum devices. For instance, physical qubit connections are limited, which necessitates the use of quantum SWAP gates to dynamically transform the logical topology during execution. In addition, to optimize fidelity, it is essential to ensure that 1) the allocated hardware has a low error rate and 2) the number of SWAP gates injected into the circuit is minimized. To address these challenges, we propose a suite of algorithms: the Fidelity-aware Graph Extraction Algorithm (FGEA) is used to identify the hardware region with the lowest probability of error, the Frequency-based Mapping Algorithm (FMA) allocates logical-physical qubits that reduce the potential distance of topological transformation, and the Heuristic Routing Algorithm (HRA) searches for an optimal swapping injection strategy. We evaluate the proposed algorithms on the IBM-provided Noisy Intermediate-Scale Quantum (NISQ) computer, using a dataset consisting of 17 different quantum circuits of various sizes. The circuits are executed on the IBM Toronto Falcon processor. The three proposed algorithms outperform the existing SABRE algorithm in reducing the number of SWAP gates required. Therefore, our proposed algorithms hold significant promise in enhancing the fidelity and reducing the number of SWAP gates required in implementing quantum algorithms.**

***Index Terms*—Quantum computing, qubit mapping, SWAP reduction, calibration data, circuit fidelity, gate execution time.**

I. Introduction

Quantum computer. By harnessing the power of quantum mechanics [1], it is proven that quantum computers, in certain cases of computation, will be able to perform exponentially better than their classical counterparts in the foreseeable future. One theoretical demonstration of this prospect is the infamous paper written by Shor [2], in which he proposes a quantum prime factorization algorithm that executes in polynomial time. However, the current generation of quantum computers is still extremely limited in terms of (1) *noise and decoherence* susceptibility, which causes error in computation [3], and (2) scalability, where the *insufficient qubits and connections* hinder the execution of quantum algorithms [4]. Therefore, massive devotion is still needed on science and engineering fronts before the vision of quantum computation is realized.

We thank the anonymous reviewers for their suggestions and feedback. This research was in part supported by US NSF under Grants: AMPS-2229073, AMPS-2229075, and CNS-2103405. Corresponding author: Tu N. Nguyen. The source code is released at: https://github.com/NextCNS/QubitMapping.

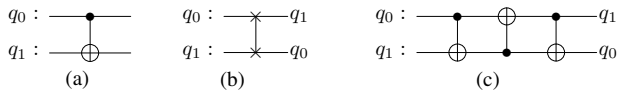

Fig. 1. Quantum circuit representation of (a) CNOT gate (b) SWAP gate, and (c) the composition of CNOT gates to build a SWAP gate.

Qubits and connections. Possessing two critical properties of a quantum particle: *superposition* and *entanglement*, the qubit is the heartbeat of all related studies regarding quantum algorithm design and execution [5]–[7]. While superposition allows the qubits to exist in multiple classical states simultaneously, an entanglement between two qubits correlates them in such a way that the state of one inherently depends on the other. To establish an entanglement in a quantum processor, there must be a *physical connection* between the two qubits [8], [9], and an effective quantum algorithm typically requires a series of entanglement pairing of a variety of qubits combinations. However, early-stage engineering of the quantum computer provides very *limited topologies of physical qubits connection* [10]. This highlights the necessity of designing quantum circuits in a manner that can tolerate potential errors and account for the *topological limits* of a quantum computer.

Entanglement and swap. The quantum circuit describes an ordered set of *quantum operators* (called gate) to be performed on a set of qubits in order to execute a quantum algorithm [9]. Within which, the entanglement operator is represented by the CNOT gate, illustrated in Fig. 1(a). In order to work around the *topological limitation* of the quantum processor, the SWAP gate, illustrated in Fig. 1(b), is employed. The SWAP gate has the ability to *swap the states* of the two target qubits [11], hence, it can be used to *virtually rearrange the physical qubits* while in execution. In simple term, it can "move" one logical qubit to a physical position that is adjacent to another logical qubit on the circuit.

Swapping in action. For instance, let's examine the first two CNOT gates depicted in Fig. 2(a). Suppose that the quantum processor is only equipped with limited connections, namely q_0-q_1, q_1-q_2, q_2-q_3. Thanks to their adjacency, the first entanglement operator between q_0-q_1 can be performed directly. However, this can not be the case for the second operator of CNOT(q_0, q_2) since q_0 and q_1 are not connected. Instead, as illustrated in Fig. 2(b), after the first CNOT, the SWAP(q_0, q_1) gate is applied to "*move q_0 downward and q_1 upward*". The CNOT(q_1, q_2) gate applied afterwards shall then be equivalent to the originally intended CNOT(q_0, q_2) gate.

979-8-3503-4323-6/23/$31.00 ©2023 IEEE
DOI 10.1109/QCE57702.2023.00019

Why qubit mapping? Even state-of-the-art quantum computers are highly susceptible to *noise and decoherence*, meaning that the addition of more gates to the circuit execution can result in increasingly error-prone outcomes. Furthermore, each SWAP gate is composed of three CNOT gates [9], [12], as illustrated in Fig. 1(c). Therefore, it is not only essential to find the *optimal swapping strategy*, but also to arrange the qubits in a manner that facilitates this strategy. In this work, we address the problem of designing a *logical-to-physical qubits mapping scheme* that is geared towards optimizing the *overall fidelity* of a quantum circuit execution on Noisy Intermediate-Scale Quantum (NISQ) systems.

Motivation. Although there exist algorithms for qubit mapping and swapping, only a few of studies take the fidelity of the logical circuit into account. Comprehensive overview of current literature is presented below.

- The authors in [13] propose an exact synthesis matching flow aimed to realize circuits for quantum architecture for LNN. Along the same lines, the authors in [14] address the problem of efficient movement of SWAP gates and proposed a NN optimization. However, these methods of constructing NN- compliant quantum circuits are proven to be NP-complete in [15] and hence these methods are not suitable when dealing with circuits of large instances.
- In the studies [10], [16], [17], authors propose heuristic, exact algorithms for qubit allocation by considering the physical topology of the IBM quantum processor. These methods are, however, not scalable for complex frameworks.
- In [18], [19], the authors propose a strategy to incorporate SWAP gates into the circuit. While the heuristic approach quickly searches for and identifies local SWAPs, the algorithm doesn't consider the initial mapping of the circuit. P. Zhu et al. [20] have incorporated the look-ahead" strategy into the heuristic cost function for additional optimization. But, this method misses the best SWAP operation needed for connectivity constraints in many instances.
- Stephen et al. [21] propose a method based on graph partitioning that can be used to identify optimal qubit topology and mapping flows for interaction graphs that are specific to the problem at hand. On the other hand, in [22] proposes a multi-tier approach for solving qubit mapping problems by considering the topology and gate fidelity constraints. If gate fidelity is non-uniform or errors are correlated, it becomes crucial to map the quantum program onto qubits in the architecture that exhibit high fidelity.

Our Contribution. The aim of "Towards Fidelity-Optimal Qubit Mapping on NISQ Computers" is to enhance the efficiency of mapping and allocating quantum circuits onto a physical quantum computer. This can reduce the number of qubits and computational resources required, which can improve the overall performance of the quantum computer. In short, the research focuses on reducing qubit swaps to enhance the efficiency of quantum circuits during mapping and allocation. We summarize key **innovation** and **contribution** of this work as follows:

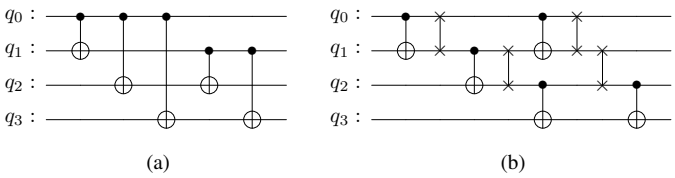

Fig. 2. (a) Logical circuit with five CNOT gates and (b) Logical circuit with nearest neighbor compliance.

- To guarantee reduced errors and to enhance the overall performance of the circuit we propose a *fidelity-aware graph extraction algorithm (FGEA)*.
- After choosing the graph, a *frequency-based mapping algorithm (FMA)* is proposed to provide an efficient initial mapping or allocation of logical qubits to physical qubits that reduces the addition of SWAP gates to a great extent.
- Thereafter, a *heuristic routing algorithm (HRA)* is put forth to minimize the number of SWAP gates added on the go. This results in reduced swap operations and fast execution times when compared to previous swap-based algorithms.
- We evaluate our results based on IBM's calibration data. We compare our results with existing algorithms based on the metrics CNOT gate count, fidelity, and execution times.

Organization. The rest of this paper is organized as follows. In section §II, we provide a background of NISQ systems, quantum circuits, how quantum mapping problems are solved. Section §III begins with a detailed discussion of the formulation of the Qubit mapping problem. We analyze the problem in section §III-B. In section §IV, we demonstrate algorithms used in our approach and analyze the complexity of the proposed solution in section §IV-D. Following that, we provide the evaluation of the results of experiments in section §V. Section §VI concludes this article.

II. Preliminaries

In the following sections, we provide a brief overview of NISQ systems, quantum circuits in section §II-A, and quantum mapping problems and provide the background of quantum gates and how it can be applied in solving the optimization of a mapping problem in section §II-B. In section §II-C we discuss the previous research attempts to solve the quantum mapping and allocation on the NISQ architectures.

A. NISQ, Quantum Circuits and IBMQ.

Noisy Intermediate-Scale Quantum. NISQ devices are quantum computers consisting of 20 to a few hundred qubits [23]. Quantum device applies the superposition and entanglement to the qubits to perform logical circuit operations. As quantum states are noise-sensitive, these systems are prone to quantum decoherence. Because of the qubit limit constraint, reaching fault-tolerance requires qubits to perform error corrections during the operation [18]. Two methods are described to decrease the number of qubits required in NISQ devices. One approach uses complete active space techniques,

which involve partitioning the molecular space into active and inactive regions [24]. Another is to employ optimized techniques for allocating qubits [10]. A critical problem with NISQ is the low connectivity of its coupling maps which the precise allocation of qubits can mitigate for better connectivity.

The importance of quantum circuit. Quantum circuits perform computations more quickly than classical circuits in certain cases [4]. Quantum circuits have potential applications in cryptography, material science, and drug discovery through new ways to solve problems that are currently intractable using classical computers [23]. The quantum circuit replicates the ideology of classical computers by leveraging the principles of quantum theory. This allows the quantum system to execute a sequence of processes to perform quantum computations like initialization of qubits, measurement, and quantum gates. Quantum gates are the basic building blocks of a quantum circuit and are used to manipulate qubits.

IBM-NISQ. Via cloud computing technology, IBM is able to provide access to quantum computation to the general population. Their commercialized NISQ system is called IBMQ, and it is widely adopted by researchers and companies across many fields of research to explore the potential of quantum computing and develop new applications and algorithms. IBM-NISQ is used as a platform for executing medium-sized quantum circuits which also serves as a testbed for showcasing advancements in performance and scalability before these improvements are implemented on larger quantum devices [23]. IBM's primary goal is to advance fault-tolerance quantum devices before commercializing. In this paper, we focus on the IBMQ Falcon processor for our research motivation because of the improved gate fidelity and reduced error rates compared to its previous generation topologies [25]. It has a relatively large number of qubits compared to some other NISQ devices, which makes it suitable for running certain quantum algorithms and simulations. In terms of performance, it shows promising performance in some benchmarking tests, indicating that it may be capable of outperforming other NISQ devices in certain scenarios.

Fig. 3 refers to the IBMQ Toronto processor. In general, IBMQ machines undergo calibration twice a day, with daily public postings of experimental measurements of key properties such as qubit relaxation time (T_1), coherence time (T_2), gate errors, readout errors. Only short programs can execute reliably on the machine, with programs exceeding 16 CNOT operations having less than 50% chance of executing correctly due to fluctuation in the above-mentioned properties [26].

B. Quantum Mapping and Allocation Problem.

Entanglement in quantum circuit. In the quantum circuit, a quantum logic gate is a basic quantum circuit operating on fewer qubits. The CNOT gate is essential for performing entangling operations between qubits and is a fundamental building block of many quantum algorithms, such as Shor's algorithm for factoring large numbers [2]. CNOT gate is a key resource in many quantum algorithms and protocols, including quantum teleportation, and quantum error correction

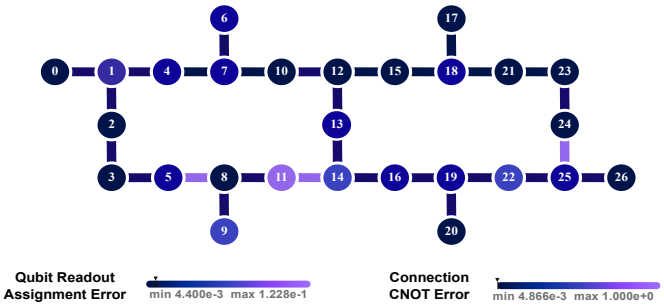

Fig. 3. IBM-Toronto Topology. The color of each qubit and connection reflects the degree of error. The lighter the color, the higher the error rate.

[27]. However, entanglement is also a delicate resource that can be easily destroyed by noise and other sources of error in the quantum hardware. Because of this, it poses a challenge for developing practical quantum computers.

Quantum connectivity limitation. In NISQ architectures, the qubits need to be isolated from the environment in order to maintain their coherence and prevent decoherence. Hence a limited number of qubits are physically connected to preserve their coherence. In addition to this limitation, the availability of physical space to place the qubits is another limitation. As the number of qubits in the quantum computer increases, the amount of physical space needed to place them also increases. This can be a practical challenge for building large-scale quantum computers. There are a variety of techniques for physically connecting qubits in its processors, depending on the specific architecture and technology being used. The qubits are connected by a series of superconducting wires, known as resonators, which allow for the transfer of quantum information between qubits. The resonators are designed to be highly tuned and matched to the qubits, in order to minimize noise and errors. Fig. 3 represents IBM Falcon processor which uses a heavy-hexagonal qubit while this reduces qubit connectivity slightly, it also minimizes frequency collisions and spectator qubit errors that can negatively impact the performance of quantum applications [26].

Mapping and swapping. To overcome the connectivity limitations of the quantum computer, quantum circuit mapping methods are put forward. (i) Initial mapping, when designing a quantum algorithm, logical qubits and gates are typically represented in an abstract way, without specifying physical qubits that will be used to implement them. However, when it comes to running the algorithm on a specific quantum processor, logical qubits must be mapped to physical qubits available on that processor. The objective of this process is to minimize the number of physical qubits required to implement logical qubits and gates of the algorithm, while also taking into account the connectivity constraints of the quantum processor [10]. (ii) Swapping allows operations to be performed on qubits that are not directly connected. However, swapping can also introduce additional errors and increase the complexity of quantum circuits, so it is important to use it judiciously [18].

C. Previous works and experiments.

In this subsection, we will review and discuss the relevant literature and experiments that have been conducted in the problem field. This will provide a foundation for understanding the current state of research and the context for our work.

1) Reducing C-Not count: Mapping quantum circuit onto a physical layer in a quantum computer has constraints. One solution is optimal mapping to reduce resources. CNOT gates typically use two logical qubits for processing. Vlad Gheorghiu and others address the problem by using Clifford+T gates, improving qubit mapping performance compared to other algorithms [12]. Results improve when coupled with methods for optimal initial mapping of qubits. A heuristic approach considers optimal mapping to reduce resources (CNOT) and the method is processed as pre-processing.

2) SAT approach for commuting gate: SWAP gates are needed in quantum circuits due to limited qubit connectivity. A pre-determined SWAP gate is effective for connecting two qubits, and an SAT-based approach [28] can find initial mappings for circuits with commuting gates to minimize SWAP gates. This approach is shown to reduce gate count and swap layers by 65% and 25% respectively, on a random 500-node three-regular graph. Swap strategies can also efficiently transpile circuits with blocks of commuting two-qubit gates to hardware, resulting in low-depth circuits. A good initial mapping can further reduce the required number of swap gates for program graphs that are not complete.

3) Exploiting Qubit Reuse through Mid-circuit Measurement and Reset: In this research paper [29], the authors discuss reducing qubit swaps in quantum circuit mapping that can improve efficiency by reducing errors and execution time. Mid-circuit measurement allows for qubit reuse and can significantly reduce the number of qubits and swaps needed, improving efficiency and fidelity. This technique is shown to reduce circuit resource usage by 60% and improve fidelity by 15%. By reducing the number of required qubits, the number of swap operations can be decreased, which in turn reduces execution time and improves reliability.

III. Problem Formulation

In this section, we provide the problem formulation for the qubit mapping problem, constraints, and their definitions. Following problem formulation we provide problem analysis along with examples that leads us to the proposed solution.

A. Problem formulation.

The objective of qubit mapping problem denoted by a function $\mathcal{I} : \mathcal{L} \rightarrow \mathcal{P}$, is to find a mapping $\mathcal{I}$ between a set of logical qubits $\mathcal{L}$ to physical qubits $\mathcal{P}$ that minimizes the total distance between connected qubits, taking into account their logical connection strength. Solving this formulation can provide an optimal or near-optimal mapping of logical qubits to physical qubits, which can then be used to implement the quantum circuit with fewer SWAP gates. The definitions given below need to understand before formulating the problem.

Definition 1 (Qubit interaction graph)**.** *Given a logical circuit with k qubits, a qubit interaction graph is defined as a graph whose edge weights represent the interaction count between the qubits. For easy understanding, this can also be shown in matrix form $\mathcal{C}(k \times k)$, where $\mathcal{C}_{ij}$ denotes the interaction rate between qubits q_i, q_j.*

Definition 2 (Coupling matrix)**.** *Given a topological graph with k qubits, the coupling matrix $\mathcal{D}(k \times k)$ denotes the minimum swaps required for a qubit to interact with another qubit in the topology graph. The distance can be calculated using the Floyd-Warshall algorithm [10].*

Formally, the problem of qubit mapping can be formulated as an integer linear program (ILP) as follows:

$$\min \sum_{i,j=0}^{n-1} \sum_{k,l=0}^{n-1} \mathcal{D}_{i,j} * \mathcal{C}_{k,l} * x_{i,k} * x_{j,l} \tag{1}$$

The variable x is a binary decision variable where

$$x_{ik} = \begin{cases} 1, & \text{logical qubit } k \text{ is mapped to physical qubit } i \\ 0, & \text{otherwise.} \end{cases}$$

subject to:

$$\sum_{k=0}^{n-1} x_{ik} = 1; \forall i = 0, 1, 2...n-1 \tag{2}$$

$$\sum_{i=0}^{n-1} x_{ik} = 1; \forall k = 0, 1, 2...n-1 \tag{3}$$

$$x_{ik} \cdot x_{jl} + x_{jk} \cdot x_{il} \leq 1; \forall i, j, k, l = 0, 1, 2...n-1 \tag{4}$$

where $\mathcal{D}_{ij}$ represents the minimum number of two-qubit swaps required to exchange the states of physical qubits i and j. Equation (2) states that each logical qubit i must be mapped to exactly one physical qubit k. The equation (3) states that each physical qubit k must be mapped to exactly one logical qubit i. Equation (4) means that for any two adjacent qubits i, j and k, l in the interaction matrix $\mathcal{C}$, at most one of the two pairs of physical qubits can be mapped to 1 but not both simultaneously. This constraint ensures that only adjacent logical qubits are mapped to adjacent physical qubits, as specified by the interaction matrix, which helps to reduce the overall cost of the mapping.

B. Problem Analysis.

In the context of considering the fidelity of quantum gates, previous research utilizes an Linear Nearest Neighbour circuit as a foundational framework (as seen in reference [22], [30]). Fig. 2(a) shows a quantum circuit composed of 5 CNOT gates with qubits $\{q_0, q_1, q_2, q_3\}$. It is mapped to IBM quantum processor *ibmq_toronto*, shown in Fig. 3. Based on the qubit count in logical circuit and physical topology, the initial step should involve choosing the subgraph onto which we wish to map the circuit and ensuring that the circuit is NN compliant.

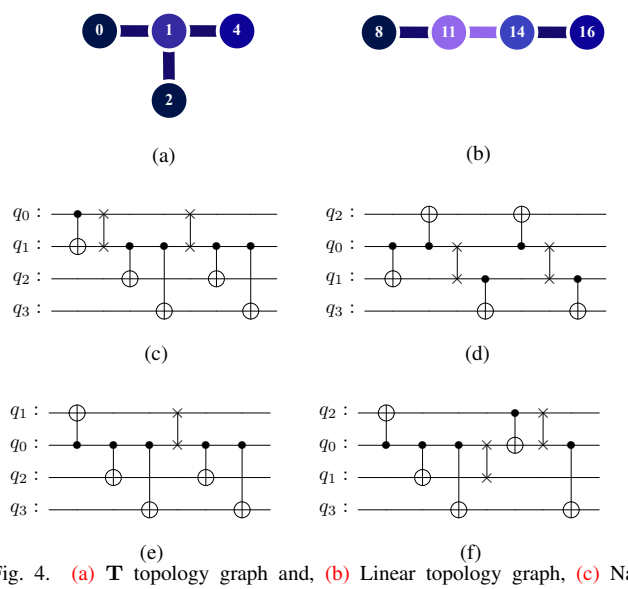

Fig. 4. (a) $\mathbf{T}$ topology graph and, (b) Linear topology graph, (c) Naive assignment of qubits in $\mathbf{T}$ topology (d) Dynamic assignment of qubits in Linear topology, (e) Dynamic assignment of qubits in $\mathbf{T}$ topology with SWAP gate added on (q_0, q_2) (f) Dynamic assignment of qubits in $\mathbf{T}$ topology with SWAP gate added on (q_1, q_0).

Graph selection. As per the topological graph in Fig. 3, the possible non-isomorphic subgraphs are as shown in Fig. 4(a), 4(b). The logical circuit is executed on any one of these subgraphs. But, due to the different topologies of the subgraphs, each subgraph has different nearest-neighbour constraints. When the logical circuit is executed on these subgraphs, each topology results in a unique final circuit as shown in Fig. 4(c), 2(b) on the basis of swap gates added.

For example, Consider $\mathbf{T}$ topology as shown in Fig. 4(a), and initial configuration on given logical circuit Fig. 2(a) as $\{q_0 \rightarrow 0, q_1 \rightarrow 1, q_2 \rightarrow 2, q_3 \rightarrow 3\}$. The NN compliant circuit obtained through this configuration is shown in Fig. 4(c). Similarly, for Linear topology as shown in Fig. 4(b), and initial configuration on given logical circuit Fig. 2(a) as $\{q_0 \rightarrow 0, q_1 \rightarrow 1, q_2 \rightarrow 2, q_3 \rightarrow 3\}$. The NN compliant circuit obtained through this configuration is shown in Fig. 2(b). Therefore, it is clear that the $\mathbf{T}$ topology subgraph needs fewer swap gates to execute the entire circuit than the linear topology subgraph. So we choose the $\mathbf{T}$ topology subgraph.

Fidelity-aware graph extraction. After choosing topology graph, examining accuracy of the gates used in logical circuits leads to varying levels of circuit fidelity. Each qubit in physical topology graph exhibits different error rates as shown in Tab. I. In general, single-qubit gate errors are one order lower than multi-qubit gate errors. Hence, choose qubits with better error rates to execute the logical circuit. $\mathbf{T}$ topology subgraph as shown in Fig. 4(a) can be extracted from *ibm_toronto* topology graph as shown in Fig. 3 in many ways i.e., the topology subgraph can be $\{0, 1, 2, 4\}$ or $\{22, 24, 25, 26\}$. These two topologies are similar but their fidelities are different.

Mapping. After selecting a higher fidelity subgraph, we have to map the logical qubits onto the physical qubits. Consider again the logical circuit in Fig. 2(a) and the $\mathbf{T}$ topology as shown in Fig. 4(a). This time instead of choosing the configuration as $\{q_0 \rightarrow 0, q_1 \rightarrow 1, q_2 \rightarrow 2, q_3 \rightarrow 4\}$, choose a different initial configuration. Suppose the initial configuration is $\{q_1 \rightarrow 0, q_0 \rightarrow 1, q_2 \rightarrow 2, q_3 \rightarrow 4\}$ the logical circuit after mapping is shown in Fig. 4(e). In the same way, consider linear topology as shown in Fig. 4(b), and the initial mapping as $\{q_2 \rightarrow 8, q_0 \rightarrow 11, q_1 \rightarrow 14, q_3 \rightarrow 16\}$ the logical circuit after mapping is shown in Fig. 4(d). It is evident from the figures that the SWAP gate count is decreased to a great extent. Hence, choosing a good initial mapping can impact the overall performance of the circuit.

IV. Proposed Solution

In this section, we discuss in detail the algorithms proposed. We propose (i) *Fidelity-aware graph extraction algorithm (FGEA)* with an objective to select a subgraph with better error rates to map logical qubits. Following this, we put forth (ii) *Frequency-based mapping algorithm (FMA)* with an objective to reduce the total distance between connected qubits, as well as (iii) *Heuristic routing algorithm (HRA)* that aims to minimize the SWAP gate count in the circuit.

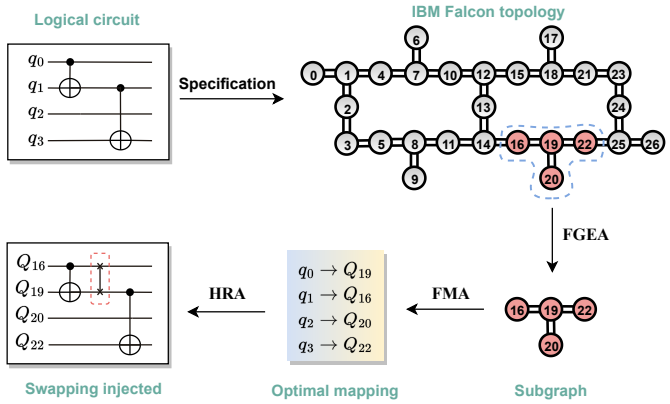

Fig. 5. Algorithm flow-map for optimal mapping.

The flowchart shown in Fig. 5 gives a basic idea of the proposed solution. The flow starts with the logical circuit and the IBM Falcon topology. The logical circuit provides IBM Falcon topology with the number of qubits required to extract a sub-topology from it while considering the qubit and gate status to mitigate the overall error rate circuit execution. From now on, we refer to sub-topology as a subgraph. With an aim to minimize the swap distance between connected qubits, an optimal initial mapping based on distance and interaction count of physical qubits to logical qubits is employed. To decrease the overload of SWAP gates in the circuit and to satisfy the connectivity constraint, cost-based swapping is used that takes into account the number of two-qubit gates.

A. Fidelity-aware graph extraction (FGEA)

The subgraph can be extracted based on criteria of better error rates and count of logical qubits. This is particularly important as physical qubits are prone to errors, and minimizing error rates is critical for the reliability of quantum computing systems. The pseudo-code of subgraph extraction with minimum error rates is given in Alg. 1 (*FGEA*).

Algorithm 1: Fidelity-aware graph extraction (FGEA)

Input: IBM qubits topology $\mathcal{G}$, number of logical qubits k

Output: The subgraph of lowest error rate $S_\Omega \in \mathcal{G}$

Let $\mathcal{S}$ be set of possible subgraphs of $\mathcal{G}$;
Let *error*(S) be the function to evaluate the overall error rate of the subgraph $S \in \mathcal{G}$ based on the status of individual gates and qubits;
Initialize the final result: $S_\Omega \leftarrow \emptyset$;
for $S \in \mathcal{S}$ **do**
  if $S_\Omega = \emptyset$ **or** *error*(S) $<$ *error*(S_Ω) **then**
    $S_\Omega \leftarrow S$;
  end
end
return S_Ω

Brute-forcing sub-topologies. The primary objective of *FGEA* is to extract a reliable subgraph. This can ultimately improve the overall performance of the circuit. Consider the topology graph $\mathcal{G}$, the number of qubits k in the logical circuit as inputs. Get the list of possible subgraphs with k nodes from the function *subgraph()* into the variable $\mathcal{S}$. This method is used to create a new graph that contains only a subset of the nodes and edges from an existing graph and returns a new graph object that contains only the nodes and edges from the original graph that are in the subset specified by nodes.

Error estimation. The qubit error on each node and gate errors on each edge is estimated based on IBM's calibration data [26], using the method *error()*. For more details on the calibration data, refer to Tab. I. This function takes a subgraph as a parameter from the available subgraph list and aggregates the error of the subgraph. The obtained error rate is compared with the error rate of the remaining subgraphs available in $\mathcal{S}$ and updates the lowest error rate subgraph in the variable $\mathcal{S}_\Omega$.

B. Frequency-based mapping (FMA)

By considering the mapping of logical qubits to physical qubits as an assignment problem, with the goal of minimizing the total distance between connected qubits. The problem formulation and pseudo-code for mapping are described in section §III and Alg. 2 (*FMA*), respectively.

High interaction → low distance. The main idea of *FMA* is that the logical qubits with highest number of interactions on the circuit are prioritized to be placed on the physical qubits that have a lower overall distance to other qubits. This can significantly minimize the overall addition of SWAP gates into the circuit and henceforth make the circuit reliable. *FMA* input contains the subgraph topology $\mathcal{S}_\Omega$ obtained from *FGEA* and the logical circuit $\mathcal{K}$, it outputs the initial map $\mathcal{I}$.

Logical/physical evaluation. The interaction count of each logical qubit in a variable $\mathcal{C}_i$ is updated from line $2-6$ by leveraging the *qubit_interaction()* function in the IBM Quantum circuit library which is used to check the interactions between two qubits in a quantum circuit. The method takes

Algorithm 2: Frequency-based mapping (FMA)

Input: Subgraph topology from Alg. 1 as $\mathcal{S}_\Omega$, logical circuit as $\mathcal{K}$

Output: Initial mapping result $\mathcal{I}$

Let n be the number of qubits to be mapped
for $i \leftarrow 0$ **to** $n-1$ **do**
  Let $\mathcal{C}_i$ be the number of interaction of qubit i needs to make when executing circuit $\mathcal{K}$:
  $\mathcal{C}_i \leftarrow 0$
  for $j \leftarrow i+1$ **to** $n-1$ **do**
    Let $c_{\langle i,j \rangle}$ be the number CNOT gate between qubit i and j in $\mathcal{K}$;
    $\mathcal{C}_i \leftarrow \mathcal{C}_i + c_{\langle i,j \rangle}$;
  end
  Let $\mathcal{D}_i$ be the shortest distance from physical qubit i to all vertices in the subgraph $\mathcal{S}_\Omega$ calculated via the Floyd-Warshall algorithm;
end
Let $\mathcal{L}$ be the set of logical qubits sorted by $\mathcal{C}$;
Let $\mathcal{P}$ be the set of physical qubits sorted by $\mathcal{D}$;
$\mathcal{I} \leftarrow \emptyset$; $j \leftarrow 0$;
for $i \leftarrow n-1$ **downto** 0 **do**
  Map the logical qubit $\mathcal{L}_i$ with highest number of interactions to physical qubit $\mathcal{P}_j$ with lowest Floyd-Warshall distance:
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{\mathcal{L}_i : \mathcal{P}_j\}$ & $j \leftarrow j+1$;
end
return $\mathcal{I}$

two integer parameters, *qubit1* and *qubit2*, which represent the qubit indices in the circuit and returns a gate object if the interaction between the two qubits is found. On the other hand, the variable $\mathcal{D}_i$ is updated by calculating the distance between each physical qubit to all the other qubits of the topology graph using Floyd-Warshall algorithm [31], that provides the shortest path between all pairs of vertices.

Sorting & mapping. Based on the values obtained from $\mathcal{C}_i$, $\mathcal{D}_i$, sorting is performed on the logical and physical qubits, and the sorted list is stored in variables $\mathcal{L}, \mathcal{P}$, respectively. Then, map the logical qubit that has more interactions with other logical qubits to the physical qubit in the coupling map which has the least distance. $\mathcal{I}$ holds the updated mapping data. This can lead to more efficient and reliable quantum circuits. In particular, mapping logical qubits that have more interactions with other qubits to physical qubits that have the least distance can help reduce the overall circuit depth and improve the circuit's error rate.

We describe an example of the logical circuit in Fig. 2(a) and the physical graph obtained by *FGEA* as shown in Fig. 4(a). The algorithm executes as follows. (i) Total number of interactions of logical qubits q_0 given by $\mathcal{C}_0$, is 3. Similarly, $\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3$ for qubits q_1, q_2, q_3 are $3, 2, 2$ respectively. The total distance $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$, $\mathcal{D}_3$ of physical qubits Q_0, Q_1, Q_2, Q_3 to

other qubits is $2, 1, 1, 2$ respectively. Sort the logical qubits in the descending order of their interaction count, the result would be $\mathcal{L} \leftarrow \{q_0, q_1, q_2, q_3\}$, the physical qubits in the ascending order of their distance, the result would be $\mathcal{P} \leftarrow \{Q_1, Q_2, Q_0, Q_3\}$. At the end, map the qubits $\mathcal{I} \leftarrow \{q_2 : Q_0, q_0 : Q_1, q_1 : Q_2, q_3 : Q_3\}$.

Analysis. We analyze the readout assignment error rate dataset of *ibm_toronto* to find the qubits with lower error rates and map them to the physical qubits. The algorithm aims to minimize the sum of swap distances between neighboring physical qubits in the final mapping, which is a measure of the total number of two-qubit swaps required to execute quantum circuits on the mapped logical qubits. After extracting a subgraph from the main topology graph, there may be multiple ways to assign the qubits to the vertices of the subgraph. This is referred to as a "configuration" and is defined in the section §III. Our approach involves determining the mapping of the NN-compliant circuit to the qubits in the quantum computer, taking into account both the qubit and gate error rates in order to minimize the overall error rate.

C. Heuristic routing (HRA)

In this section, we discuss heuristic routing Alg. 3 (*HRA*). The central idea of *HRA* is to transform a quantum circuit that can only be executed on a limited set of qubits, into a quantum circuit that can be executed on every set of qubits by inserting swap gates judiciously. *HRA* takes initial mapping $\mathcal{I}$ obtained from *FMA*, subgraph topology from *FGEA* $\mathcal{S}_\Omega$ and logical circuit $\mathcal{K}$ as inputs, and outputs final CNOT count c.

Potential swap pairs. *HRA* traverses through all gates available in logical circuit $\mathcal{O}_{i,j}$. In each iteration, check whether the gate is executable using method *is_executable()*. We leverage this method from IBM API and it returns $true$ if the gate is executable, $false$ otherwise. If the gate in the $\mathcal{O}_{i,j}$ is executable, it proceeds with next gate. If the gate is not executable, then insert swap gates as given in line 8. Variable $\mathcal{X}$ contains the list of all possible swaps between qubits. the function *swap()* finds the swaps associated with the qubits in $\mathcal{O}_{i,j}, \mathcal{I}$ and list them as candidate swaps.

Heuristic cost evaluation. The cost of each possible swap pair is calculated based on minimum sum of distances between two qubits dependent on initial mapping and number of swap gates inside the circuit using method *cost()*. The cost of all swap pairs is compared and the variable $\mathcal{X}_\Omega$ is updated with lease cost swap. The CNOT gate count provides useful performance metrics for evaluating the effectiveness of the algorithm. By efficiently calculating the optimal sequence of swap gates and minimizing the number of CNOT gates, the algorithm can help reduce the cost and improve the performance of quantum circuits executed on physical hardware. We provide an instance of a logical circuit as shown in Fig. 2(a), a physical graph as shown in Fig. 4(a), and initial configuration $\mathcal{I}$ obtained from *FMA* $\{q_2 : Q_0, q_0 : Q_1, q_1 : Q_2, q_3 : Q_3\}$. The circuit obtained after initial mapping would be as shown in Fig. 4(e). We need to add one swap gate to make the entire circuit executable on a quantum computer. Obtain the list of gates to be executed from the logical circuit. Gates CNOT(q_0, q_1), CNOT(q_0, q_2), CNOT(q_0, q_3) executes directly as the two qubits satisfy the nearest neighbor compliance. Gather the list of possible swap operations and put them in the list variable $\mathcal{X}$. Possible swap operations to execute the gate CNOT(q_1, q_2) are SWAP(q_0, q_2), SWAP(q_1, q_0). Calculate the cost associated with each SWAP. Consider adding the SWAP gate between qubits (q_0, q_2). The cost is 6 since 2 SWAP gates are added in the circuit (Fig. 4(e)). The next possible SWAP gate is (q_1, q_0). The cost of this swap is 3 (Fig. 4(f)).

Algorithm 3: Heuristic routing (HRA)

Input: Logical circuit as $\mathcal{K}$, initial mapping as $\mathcal{I}$, subgraph topology S_Ω from Alg. 1
Output: Final CNOT gate count c

1 Initialize $c \leftarrow 0$;
2 Let $\mathcal{X}$ be the set of possible swap operations that can be performed on two qubits;
3 Let $\mathcal{O}(i,j) \in \mathcal{K}$ be a CNOT gate in the logical circuit that requires the interaction between qubits i and j;
4 Let $\mathcal{X}_\Omega$ be the swap operation with minimum cost; $\mathcal{X}_\Omega \leftarrow \emptyset$;
5 **for** $\mathcal{O}(i,j) \in \mathcal{K}$ **do**
6 **if** (i,j) *are not adjacent in* S_Ω **then**
7 $\mathcal{X} \leftarrow \emptyset$ & $\mathcal{X} \leftarrow \mathcal{X} \cup$ *swap*$(\mathcal{O}, \mathcal{I})$
8 **for** $\mathbf{X} \in \mathcal{X}$ **do**
9 **if** $\mathcal{X}_\Omega = \emptyset$ **or** *cost*$(\mathbf{X}) <$ *cost*$(\mathcal{X}_\Omega)$ **then**
10 $\mathcal{X}_\Omega \leftarrow \mathbf{X}$;
11 **end**
12 **end**
13 **end**
14 **end**
15 Calculate the number of CNOT gates in the circuits;
16 $c \leftarrow$ *cost*$(\mathcal{X}_\Omega)$ + $|\mathcal{O}|$
17 **return** c

D. Complexity of algorithms.

Here we discuss the complexity of initial mapping and swapping algorithms. Understanding the complexity is crucial for designing efficient quantum circuits and optimizing their execution on quantum hardware.

Lemma 1. *Given the same number of logical qubits and physical qubits, the time complexity of FMA is* $O(|\mathcal{L}|^2)$.

Proof. Computing the logical qubit interactions has a complexity of $O(|\mathcal{L}|^2)$, and computing the distance between physical qubits has a time complexity of $O(|\mathcal{P}|)$ where n is the number of qubits in the logical circuit as well as in physical topology. Sorting the logical qubits, and physical qubits has a time complexity of $O(\mathcal{C}\log\mathcal{C})$, $O(\mathcal{D}\log\mathcal{D})$ respectively. Creating the initial mapping takes $O(|\mathcal{L}|)$ time. The total time complexity of the qubit mapping problem is $O(|\mathcal{L}|^2+|\mathcal{P}|+\mathcal{C}\log\mathcal{C}+\mathcal{D}\log\mathcal{D}+|\mathcal{L}|)$, which can be simplified to $O(|\mathcal{L}|^2)$ since $|\mathcal{L}|$, $|\mathcal{P}|$ are the same in our algorithm. □

Lemma 2. *The computational complexity of HRA is* $O(|\mathcal{O}|^2 + |\mathcal{X}| + |\mathcal{X}_\Omega|)$.

Proof. Computing the available two-qubit gates has a complexity of $O(|\mathcal{O}|^2)$. To obtain the possible swap operations, the time complexity is $O(|\mathcal{X}|)$. Calculating the cost of each swap operation requires the time complexity of $O(|\mathcal{X}_\Omega|)$. The overall time complexity of the algorithm is $O(|\mathcal{O}|^2 + |\mathcal{X}| + |\mathcal{X}_\Omega|)$. □

V. Experiment and Evaluation

In this section, we assess the performance of the *FGEA* based on the number of CNOT gates of a collection of benchmarks taken from previous works [10], [18] using the most recently reported hardware model *ibm_toronto* that employs superconducting circuit technology. We choose SABRE [18] as the baseline algorithm to evaluate our results.

Parameter	Range
Qubit	27
Qubit error rate	7.700e-3 to 2.192e-1
CNOT error rate	5.986e-3 to 4.922e-2
Gate execution time (ns)	214.778 to 860.444

TABLE I
IBM Toronto calibration data. The data has been taken on 4-4-2023 and the data changes over a period of time

The given table provides calibration data of *ibm_toronto*, which shows falcon qubit topology with their respective qubit error rates, CNOT error rates, and gate execution time. Calibration data of IBM processors refers to the set of parameters that are determined through calibration experiments and used to optimize the performance of the quantum processor. These parameters include values for gate durations, amplitudes, and frequencies, as well as crosstalk coefficients and other error mitigation techniques. This data has been used in our proposed solution to analyze the fidelity of the entire circuit.

It is important to note that the calibration data is not static, and may need to be updated periodically as the behavior of the processor changes over time. This may be due to changes in the environment, such as temperature fluctuations or electromagnetic interference, or to changes in the hardware itself, such as the introduction of new qubits or gate types. Therefore, continuous monitoring and calibration of the quantum processor is necessary to maintain optimal performance. The qubit error rate represents the probability that a qubit will experience an error during computation. The IBM Quantum Experience APIs [26] are utilized for conducting experiments on IBMQ16. Through these APIs, we can access the daily calibration data of the machine, which includes information such as the time required for single qubit gates, the coherence time of qubits (T2 time), CNOT gate durations, and error rates for single-qubit gates, CNOT gates, and measurement.

We evaluate our algorithm on a set of benchmark circuits with up to 20 logical qubits. Our results show that our algorithm achieves higher fidelity than the baseline algorithms in most cases, while also being scalable to larger circuits. For our experiment, the algorithms *FGEA*, *FMA*, *HRA* are implemented in Python and Qiskit version 0.16.3. The hardware configuration is AMD Ryzen 7 5825U with Radeon Graphics 2.00 GHz, 16GB memory. We use the 27-qubit NISQ architecture IBM Toronto as the target quantum platform. To evaluate the algorithms, we selected 10 benchmark quantum circuits of various qubits (from 4 to 15).

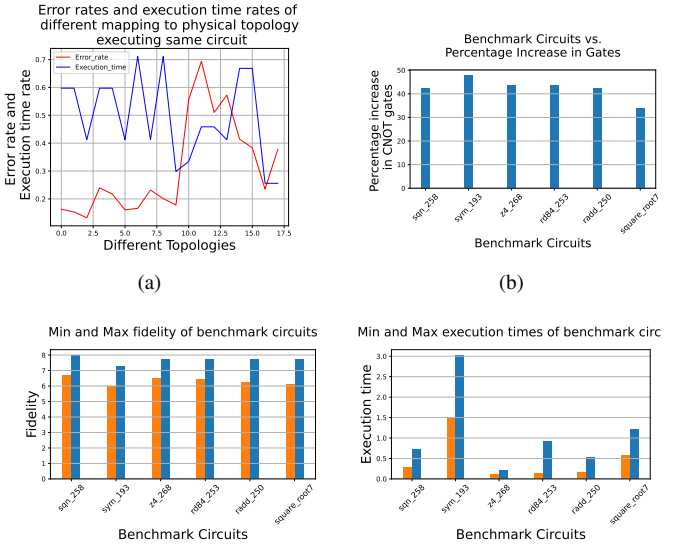

Fig. 6. (a) Error rate to Subgraph plot, (b) Percentage increase in CNOT count, (c) Percentage increase in fidelity, (d) Execution times of benchmarks

Table II clearly shows that our algorithm performs better for reducing CNOT gate count in the final circuit when compared with the SABRE algorithm. The %age gate reduction column demonstrates that on average the optimal mapping algorithm computes the circuit with 10.52% reduced CNOT gates than SABRE. Initial mapping using *FMA* is well-calculated, which reduces SWAP gate overload on the circuit. In contrast, SABRE uses a random initial mapping, which requires running the algorithm multiple times to obtain the optimal solution, consuming more time. However, our optimal mapping algorithm is deterministic with the same output consistently.

As it is evident from Fig. 6(c), by incorporating our error-aware qubit mapping technique, it is possible to generate an initial mapping that utilizes the most dependable qubits with the lowest cost. By adapting noises in the system, the fidelity of the overall circuit is increased by an average of 1.25 times and up to 1.33 times higher than the baseline algorithm.

The CNOT error rate represents the probability that a CNOT gate will experience an error during computation. Gate execution times mean the time a quantum gate takes to perform its operation. Both CNOT error rate and execution times are useful in evaluating the performance of quantum computing systems. This table gives an abstract for our understanding of how this information can be used to implement our approach to find the best suitable physical topology for the logical circuit.

Fig. 6(a) visually represents the error rates and execution times for different mappings to physical topology executing the 17 different benchmark quantum circuits. The lower the

TABLE II
EXPERIMENTS EVALUATION DATA TABLE

S.No	Benchmarks	n	Gates	SABRE gates	OM gates	%age gate reduction	f_min	f_max
1	bv_4	4	3	6	6	0	0.635	0.931
2	bv_6	6	5	20	14	30	0.686	0.896
3	bv_8	8	7	31	19	38.7	0.655	0.846
4	qft_10	10	90	429	399	6.99	0.67	0.796
5	sqn_258	10	4459	11044	10506	4.87	0.67	0.789
6	sym9_148	10	9408	25722	24344	5.36	0.669	0.791
7	sym9_193	11	15232	26531	25355	4.43	0.659	0.783
8	wim_266	11	427	1603	1306	18.52	0.652	0.772
9	z4_268	11	1343	4278	3859	9.78	0.651	0.769
10	cycle10_2_110	12	2648	6543	6167	5.75	0.66	0.774
11	rd84_253	12	5960	14719	14536	1.24	0.654	0.754
12	sym9_146	12	148	512	499	2.53	0.641	0.733
13	radd_250	13	1405	3122	2893	7.33	0.631	0.729
14	col14_215	15	7840	19181	17968	6.32	0.626	0.719
15	misex1_241	15	2100	6343	5004	21.11	0.621	0.717
16	square_root7	15	3089	8492	8138	4.16	0.622	0.71
17	ising_model_16	16	150	269	234	13.01	0.611	0.702

S.No: Serial number of the benchmark. Benchmarks: the name of the benchmark circuit. n: number of qubits in the benchmark. Gates: the number of CNOT gates in the benchmark. SABRE gates: the total number of gates in the final circuit obtained using the SABRE algorithm. OM gates: the total number of gates in the final circuit obtained using the Optimal mapping algorithm. %age gate reduction: the percentage reduction of gates using the Optimal mapping algorithm when compared with SABRE. f_min: the fidelity of the circuit without using the Optimal mapping algorithm. f_max: the fidelity of the circuit using the Optimal mapping algorithm.

error rates and execution times are, the better the overall performance of the circuit. Fig. 6(b) shows the percentage increase in CNOT gate count for each benchmark circuit with our qubit mapping and routing algorithms *FMA*, *HRA*. As shown in the graph, the average percentage increase in the CNOT gate count was observed to be around 40% with the lowest increase in the CNOT gates being 33% observed in the *square_root7* benchmark circuit making it the best performing one using our algorithm.

Fig. 6(c) shows the fidelity improvement achieved by using our qubit mapping algorithm against the SABRE algorithm. The fidelity of the output state differs in each benchmark circuit surpassing the SABRE method, as observed. Here, we used the state tomography data obtained by IBM to evaluate the output state, which is later used for verifying a quantum system in the desired state and detecting errors during quantum state preparation and manipulation. The graph demonstrates that the algorithm substantially reduced the potential error rate of all benchmark circuits, with an average improvement of 25%. The *square_root7* circuit had the most significant improvement, with an increase of over 33%.

Fig. 6(d) displays the execution duration achieved by our algorithm against the SABRE, which was calculated based on the gate execution time data. Although using actual durations of the gates could lead to more precise execution durations for each benchmark, it would increase the number of constraints in the optimization problem and prolong the compilation time. Even though using the real durations, each benchmark only requires a few seconds of compilation time. The graph indicates that the execution time varies for each benchmark depending on the number of gates to execute. On average, our method *HRA* took 2.42 times less time to execute the circuit compared to the general method. The benchmark *sym9_193*, with the largest number of gates at 15232, took $1.49s$ to execute using *HRA*.

VI. CONCLUSION AND FUTURE WORK

In summary, this paper introduces a suite of algorithms such as FGEA, FMA, and HRA for qubit mapping and allocation through static optimization in quantum computing. As quantum systems are prone to errors caused by noise, these algorithms address this major challenge by efficiently mapping logical qubits to physical qubits, which reduces the number of operations and communication needed. Utilizing optimization theory and calibration data for error-rate calculations, these algorithms enhance circuit fidelity by minimizing the number of SWAP gates required [32]. This increases circuit scalability while reducing the need for costly error corrections. Our experimental results demonstrate that the algorithms are effective in real-world hardware, making them a practical and efficient solution to the problem of qubit allocation and mapping compared to the SABRE method.

However, a more challenging dynamic approach can improve the circuit depth by swapping qubits on every circuit interaction though this may affect fidelity optimization. Overall, this work represents a significant advancement in quantum computing, and in the future, we plan to improve our comparisons with the latest mapping optimization algorithms and IBMQ compiler optimization processes with more complex computations and increased datasets. We believe that this research will pave the way for further developments in unlocking the full potential of quantum computing for solving complex problems.

REFERENCES

[1] J. Preskill, "Quantum computing 40 years later," 2023.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509, oct 1997. [Online]. Available: https://doi.org/10.1137%2Fs0097539795293172

[3] J. A. Bergou, M. Hillery, and M. Saffman, *Decoherence and Quantum Error Correction*. Cham: Springer International Publishing, 2021, pp. 161–178. [Online]. Available: https://doi.org/10.1007/978-3-030-75436-5_9

[4] J. Allcock, P. Yuan, and S. Zhang, "Does qubit connectivity impact quantum circuit complexity?" 2022.

[5] D.-S. Wang, "Superposition and entanglement from quantum scope," 2011.

[6] S. Halder and U. Sen, "Separability and entanglement in superpositions of quantum states," *Physical Review A*, vol. 107, no. 2, feb 2023. [Online]. Available: https://doi.org/10.1103%2Fphysreva.107.022413

[7] R. K. Bera and V. Menon, "A new interpretation of superposition, entanglement, and measurement in quantum mechanics," 2009.

[8] G. J. Mooney, C. D. Hill, and L. C. L. Hollenberg, "Entanglement in a 20-qubit superconducting quantum computer," *Scientific Reports*, vol. 9, no. 1, Sep 2019. [Online]. Available: http://dx.doi.org/10.1038/s41598-019-49805-7

[9] A. Saha, D. Saha, and A. Chakrabarti, "Moving quantum states without swap via intermediate higher-dimensional qudits," *Physical Review A*, vol. 106, no. 1, jul 2022. [Online]. Available: https://doi.org/10.1103%2Fphysreva.106.012429

[10] P. Zhu, X. Cheng, and Z. Guan, "An exact qubit allocation approach for nisq architectures," vol. 19, no. 11, 2020. [Online]. Available: https://doi.org/10.1007/s11128-020-02901-4

[11] G. M. J. E. L. F. B. M. P. J. R. Sigillito, A. J., "Coherent transfer of quantum information in a silicon double quantum dot using resonant swap gates," 11 2019.

[12] V. Gheorghiu, J. Huang, S. M. Li, M. Mosca, and P. Mukhopadhyay, "Reducing the CNOT count for cliffordt circuits on NISQ architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1–1, 2022. [Online]. Available: https://doi.org/10.1109%2Ftcad.2022.3213210

[13] M. Saeedi, R. Wille, and R. Drechsler, "Synthesis of quantum circuits for linear nearest neighbor architectures," *Quantum Information Processing*, vol. 10, no. 3, pp. 355–377, oct 2010. [Online]. Available: https://doi.org/10.1007%2Fs11128-010-0201-2

[14] R. Wille, O. Keszocze, M. Walter, P. Rohrs, A. Chattopadhyay, and R. Drechsler, "Look-ahead schemes for nearest neighbor optimization of 1d and 2d quantum circuits," in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2016, pp. 292–297.

[15] A. Chakrabarti, S. Sur-Kolay, and A. Chaudhury, "Linear nearest neighbor synthesis of reversible circuits by graph partitioning," 2012.

[16] M. Y. Siraichi, V. F. d. Santos, C. Collange, and F. M. Q. Pereira, "Qubit allocation," in *Proceedings of the 2018 International Symposium on Code Generation and Optimization*, ser. CGO 2018. New York, NY, USA: Association for Computing Machinery, 2018, p. 113–125. [Online]. Available: https://doi.org/10.1145/3168822

[17] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the ibm qx architectures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 7, pp. 1226–1236, 2019.

[18] G. Li, Y. Ding, and Y. Xie, "Tackling the qubit mapping problem for nisq-era quantum devices," 2019.

[19] A. Cowtan, S. Dilkes, R. Duncan, A. Krajenbrink, W. Simmons, and S. Sivarajah, "On the qubit routing problem," 2019. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2019/10397/

[20] P. Zhu, Z. Guan, and X. Cheng, "A dynamic look-ahead heuristic for the qubit mapping problem of nisq computers," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4721–4735, 2020.

[21] S. Brierley, "Efficient implementation of quantum circuits with limited qubit interactions," 2016.

[22] D. Bhattacharjee, A. A. Saki, M. Alam, A. Chattopadhyay, and S. Ghosh, "Muqut: Multi-constraint quantum circuit mapping on nisq computers: Invited paper," in *2019 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2019, pp. 1–7.

[23] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, aug 2018. [Online]. Available: https://doi.org/10.22331%2Fq-2018-08-06-79

[24] S. Gocho, H. Nakamura, S. Kanno, Q. Gao, T. Kobayashi, T. Inagaki, and M. Hatanaka, "Excited state calculations using variational quantum eigensolver with spin-restricted ansätze and automatically-adjusted constraints," 2022.

[25] A. C. Z. N. Paul Nation, Hanhee Paik, "The ibm quantum heavy hex lattice," vol. 0, 07 2021. [Online]. Available: https://research.ibm.com/blog/heavy-hex-lattice

[26] "Ibm processors," https://quantum-computing.ibm.com/services/resources?type=Falcon.

[27] T. Brun, I. Devetak, and M.-H. Hsieh, "Correcting quantum errors with entanglement," *Science (New York, N.Y.)*, vol. 314, pp. 436–9, 11 2006.

[28] A. Matsuo, S. Yamashita, and D. J. Egger, "A sat approach to the initial mapping problem in swap gate insertion for commuting gates," 2022.

[29] F. Hua, Y. Jin, Y. Chen, S. Vittal, K. Krsulich, L. S. Bishop, J. Lapeyre, A. Javadi-Abhari, and E. Z. Zhang, "Exploiting qubit reuse through mid-circuit measurement and reset," 2023.

[30] A. A. Saki, M. Alam, and S. Ghosh, "Qure: Qubit re-allocation in noisy intermediate-scale quantum computers," in *2019 56th ACM/IEEE Design Automation Conference (DAC)*, 2019, pp. 1–6.

[31] R. W. Floyd, "Algorithm 97: Shortest path," *Commun. ACM*, vol. 5, no. 6, p. 345, jun 1962. [Online]. Available: https://doi.org/10.1145/367766.368168

[32] Sri Khandavilli, Indu Palanisamy, Manh V. Nguyen, Thinh V. Le, Tu N. Nguyen, and Thang N. Dinh, "Towards fidelity-optimal qubit mapping on nisq computers," [GitHub; access 2023]. [Online]. Available: https://github.com/NextCNS/Qubit_mapping/blob/main/README.md