Exploiting the Extended Neighborhood of Hexagonal Qubit Architecture for Mapping Quantum Circuits

ABHOY KOLE, German Research Centre for Artificial Intelligence, Germany
KAMALIKA DATTA, University of Bremen, German Research Centre for Artificial Intelligence, Germany
INDRANIL SENGUPTA, Indian Institute of Technology Kharagpur, India
ROLF DRECHSLER, University of Bremen, German Research Centre for Artificial Intelligence, Germany

In this work mapping of quantum circuits to regular hexagonal grid with coupling degree of six has been investigated. Architectures involving superconducting qubits impose restrictions on 2-qubit gate operations to be carried out only between physically coupled qubits, also referred to as Nearest Neighbour (NN) Constraint. The noise introduced by the 2-qubit gates and the execution time greatly affect the computational reliability. Existing mapping techniques suffer either from the adopted approach to reduce gate overhead or from their inability to take advantage of such architectural regularity. We outlined three different qubit mapping approaches using Remote-CNOT templates, Swap gates and combination of both. We show the benefits of assigning the Cartesian coordinate system in hexagonal grid for runtime elevation and devised approaches for reduction in gate overheads. While the template-based approach gives a strict upper bound of additional gate overheads for a particular qubit mapping, the combined approach provides better result employing a larger lookahead window. Experiments on benchmark quantum circuits confirm that the proposed Swap-based method provides an average 25% improvement in gate overheads over a recent work and the combined approach contributes further 15% average improvement on the result at the expense of a little higher runtime.

CCS Concepts: • **Computer systems organization** → **Quantum computing**; • **Software and its engineering** → **Software notations and tools**; **Architectural mapping**.

Additional Key Words and Phrases: Quantum circuit, layout mapping, hexagonal architecture, hardware constraints, Swap gates, remote-CNOT template

ACM Reference Format:
Abhoy Kole, Kamalika Datta, Indranil Sengupta, and Rolf Drechsler. 2024. Exploiting the Extended Neighborhood of Hexagonal Qubit Architecture for Mapping Quantum Circuits. *ACM J. Emerg. Technol. Comput. Syst.* XX, X, Article XX (July 2024), 19 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Quantum computing aims to solve some computationally harder problems in significantly less time as compared to classical computing. Some of the well-known quantum algorithms include Shor's factorization [28], Grover's database search [11], quantum simulation and annealing [21], etc. Technological advancements have paved the way for demonstrable physical quantum computers based

Authors' addresses: Abhoy Kole, Department of Cyber-Physical Systems, German Research Centre for Artificial Intelligence, Bibliothekstrasse 5, 28359 Bremen, Germany, abhoy.kole@dfki.de; Kamalika Datta, Institute of Computer Science, University of Bremen, German Research Centre for Artificial Intelligence, Bibliothekstrasse 5, 28359 Bremen, Germany, kdatta@uni-bremen.de; Indranil Sengupta, Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur, Kharagpur, India, isg@iitkgp.ac.in; Rolf Drechsler, Institute of Computer Science, University of Bremen, German Research Centre for Artificial Intelligence, Bibliothekstrasse 5, 28359 Bremen, Germany, drechsler@uni-bremen.de.

1550-4832/2024/7-ARTXX $15.00
https://doi.org/XXXXXXX.XXXXXXX

on technologies like superconducting qubits [8], trapped ion [12], all optical qubits [23], etc. Qubits implemented using these technologies have limited coherence period and support noisy single- and 2-qubit elementary gates [1]. Algorithms written using large quantum gates need to be redescribed using certain decomposition techniques, e.g. [3, 14] in order to simulate then on such devices.

Superconducting qubits have further limitations on 2-qubit gate operations, where the operations are allowed only on physically coupled qubits. The architectures of such quantum computers can be classified as: (a) the way the qubits are physically coupled, and (b) the placement of the qubits on a 2-D plane. The degree of coupling directly influences the number of supported 2-qubit gate operations. For such architectures, coupling violations of 2-qubit operations are addressed using *Swap* gates [10] or *Remote-CNOT* templates [24] for NN-compliance. The additional gates included in the process introduce more noise and also increase the execution time.

Among the various architectural alternatives, the hexagonal qubit architecture [20, 30] holds much promise due to the extended qubit association and structural regularity. Existing mapping algorithms (e.g., [9, 13, 18, 34]) are not able to exploit the structural regularity of hexagonal grid and suffer from higher execution time. Recently Chang et al. [4] proposed a greedy approach to map quantum circuits to a hexagonal grid, considering axial coordinate system and heuristics like local and global ordering for mapping the quantum circuits. Experimental results show significant reduction in gate overhead as compared to mapping to a 2-D Cartesian grid.

However, there is ample scope for improvement to develop a better gate mapping approach. In this paper, we present a graph-based method for quantum circuit mapping in hexagonal grid with multiple choices for NN-compliance (viz., using Swap gates, using Remote-CNOT templates, or a combination). For hexagonal qubit structure the Cartesian coordinate system introduced earlier in [19] is considered for defining the cost function. With this imposed coordinate system a qubit numbering scheme is introduced for defining the cost metric in an elegant way. We start with a good qubit ordering (using an evolutionary algorithm), and then use a lookahead strategy for gate mapping using Swap gates and Remote-CNOT templates.

For a given circuit with a qubit ordering, the use of Remote-CNOT templates for NN-mapping gives a strict upper bound of gate overhead. On the other hand, the combined use of Swap gates and Remote-CNOT templates considers a 3 times larger solution space than the Swap-based method and demands a larger lookahead window for giving better results. Compared to a recent approach [4], while Swap-based approach provides an average improvement of 25%, the combined approach extends the average improvement further by 15%.

The rest of the paper is organized as follows. Section II discusses the general background of the work, section III discusses about the proposed hexagonal structure and the coordinate system, section IV discusses the proposed mapping strategy, and section V shows the experimental results. Finally section VI concludes the paper.

2 BACKGROUND

In this section we provide a brief review of quantum circuits and discuss about various qubit architectures. We also review various works for mapping quantum circuits to specific architectures.

2.1 Quantum Circuits

A quantum circuit consists of a set of qubits, on which a sequence of quantum gate operations are carried out. A qubit can exist in one of the basis states like $|0\rangle$ and $|1\rangle$, or in a state of superposition as represented by the state vector $\psi = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex coefficients or amplitudes such that $|\alpha|^2 + |\beta|^2 = 1$. We can also carry out *measurement* of the state ψ of a qubit, which results in the state to collapse into one of the basis states $|0\rangle$ or $|1\rangle$, with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively. This concept can also be extended to a multi-qubit system. The states of the qubits can be changed

in the sequence of time by performing quantum gate operations on them. In general, all quantum operations are inherently reversible only with the exception of measurement, and can be represented by unitary matrices.

One of the commonly used quantum gate libraries is *Clifford+T*, which is supported by many physical realizations of qubits [2]. The Clifford+T library comprise of the Hadamard gate H (which sets a qubit into superposition), NOT and CNOT gates (which flips the basis states $|0\rangle$ and $|1\rangle$), as well as the phase shift gates T ($\pi/4$ gate), $S = T^2$, and $Z = S^2$. A quantum circuit consisting of gates from the Clifford+T library is shown in Fig. 1. In this paper, we mainly focus on 2-qubit gate operations, and as such the proposed approaches are independent of the exact quantum gate library being used.

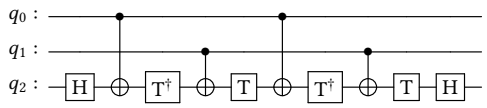


Fig. 1. An example quantum circuit.

2.2 Architectural Constraints and Nearest Neighbor

It has been experimentally found that when two qubits interact during a gate operation, some errors get introduced. The magnitude of this error has a direct correlation with the physical distance between the qubits. To limit the error during gate operations, researchers have proposed the *Nearest-Neighbor (NN) Constraint*, which requires the interacting qubits to be adjacent. This is typically achieved in one of two ways: (a) by introducing Swap gates (cascade of three back-to-back controlled-NOT gates) to exchange the states of the qubits, (b) by using CNOT templates, which consists of a cascade of CNOT gates, each of which operate on neighboring pair of qubits. Fig. 2(a) shows a quantum circuit where some of the 2-qubit gates operate on non-neighbor qubits. Fig. 2(b) and 2(c) the corresponding NN-compliant circuits using Swap gates and CNOT templates respectively. When we map a quantum circuit to some architecture, we must try to optimize in a way that the extra gate overhead is minimized.

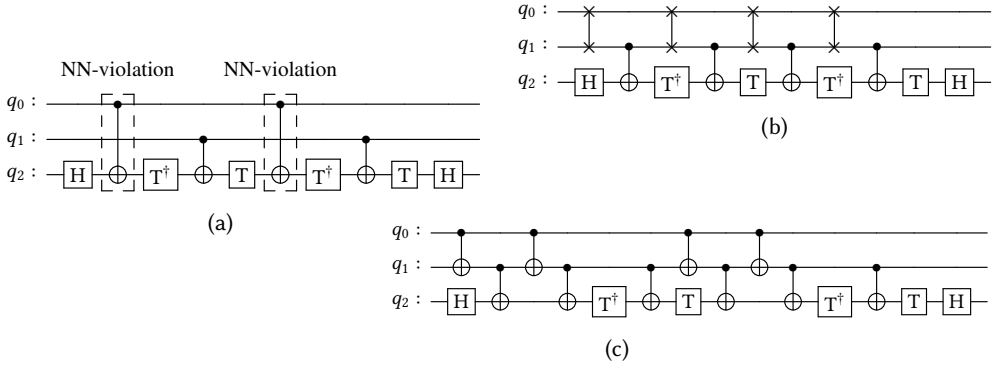


Fig. 2. (a) A given quantum circuit with non-neighbor qubit interactions, (b) NN-compliance using Swap gates, (c) NN-compliance using CNOT templates.

2.3 NN-based Qubit Mapping Methods

Many prior works have focused on the mapping of logical qubits from a given quantum circuit to hypothetical one-dimensional (1-D), two-dimensional (2-D) and three-dimensional (3-D) physical

qubit architectures [15, 16, 26, 27, 29, 33]. These methods explore various heuristics (e.g., local and global ordering) to reduce the cost of making the quantum circuit NN-compliant. A 2-D mapping approach using mixed integer programming is proposed in [27]. Some methods have also used lookahead strategies for inserting Swap gates in 1-D and 2-D architectures [15, 32]. Most of these methods rely on using Swap gates to bring interacting qubit pairs adjacent, while some of them use CNOT templates instead [7, 22, 24, 35, 36]. In [7] Almeida et al. have used CNOT templates to map circuits to the IBM QX architecture. But this method only shows better results for smaller qubit architectures. In [22] Neimann et al. have used combination of Swap and Remote-CNOT templates at the reversible gate level to reduce the mapping overhead. In all these methods the metric for comparison is the number of additional gates required.

Many qubit mapping approaches have been proposed in the literature for regular 2-D architectures; however, physical realizations of quantum computers have only been reported in recent years by various technological giants like IBM, Google, Microsoft etc [5, 25]. Several qubit mapping algorithms based on these architectures have also been proposed [7, 17, 37, 38]. In these architectures, the qubits are generally laid out on a 2-D plane with limited interconnections between them, which is governed by several factors like errors during computation, frequency assignment during fabrication, etc.

In a recent work [30], the authors have successfully demonstrated the interaction of qubits in a 2-D hexagonal grid. Motivated by this, a method for NN-compliant mapping of quantum circuits on a 2-D symmetrical hexagonal architecture has been proposed very recently [4]. Although this is the first work on 2-D hexagonal mapping, there is ample scope for improvement to further reduce the extra gate overhead. In this context, we introduce an improved graph based qubit mapping technique on a hexagonal grid. The method is found to be effective and shows better results as compared to [4]. The method is discussed in the following sections.

3 HEXAGONAL QUBIT STRUCTURE AND COORDINATE SYSTEM

In a two-dimensional hexagonal layout of qubits, each qubit can have a coupling with up to 6 adjacent qubits, as compared to a maximum of 4 for regular 2-D grid layout. This allows greater possibilities of native CNOT operations, and consequently more flexibility in quantum circuit mapping. Fig. 3 shows the proposed 2-D hexagonal layout of qubits and their coupling, which is arranged as a 7×5 layout (with width $W = 7$ and height $H = 5$).

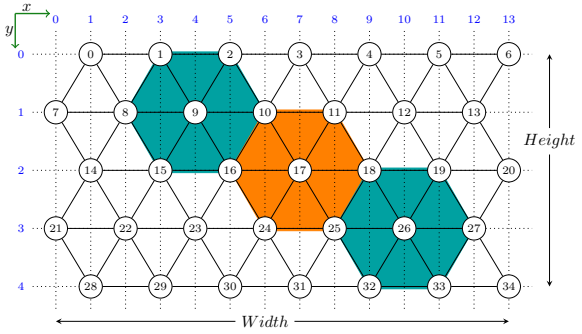


Fig. 3. A 2-D hexagonal layout of qubits and the coupling map (of width $W = 7$ and height $H = 5$).

As shown in Fig. 3, the qubits are labeled with some numbers in row-major order, i.e. in the left-to-right and top-to-bottom order. For a hexagonal array of dimension $W \times H$, the total number of qubits is $n = W.H$ which are numbered from q_0 to q_{n-1}. According to the coordinate convention

proposed in this paper, the coordinates (x_i, y_i) of qubit q_i, $0 \leq i < n$, is computed as:

$$\begin{aligned} x_i &= (1 - (y_i \,\%\, 2)) + 2\,(i \,\%\, W) \\ y_i &= i \div W \end{aligned} \tag{1}$$

where % and ÷ denote the modulus and integer division operator respectively.

Example 1. *For the* 7×5 *hexagonal qubit array depicted in Fig. 3, the coordinates of the qubit* q_9 *can be calculated as:*

$$\begin{aligned} x_9 &= (1 - (y_9 \,\%\, 2)) + 2(9 \,\%\, 7) = 4 \\ y_9 &= 9 \div 7 = 1 \end{aligned}$$

It is interesting to note that the proposed coordinate convention as per Eqn. (1) guarantees that the Manhattan Distance (MD) between any two coupled qubits is always 2. For any pair of qubits q_i and q_j, the Manhattan distance is defined as

$$MD(q_i, q_j) = |x_i - x_j| + |y_i - y_j| \tag{2}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of qubits q_i and q_j respectively. However, with this coordinate system for certain uncoupled qubits that differ only in their y coordinates, the MD can also be 2. The following example illustrates the MD of coupled and uncoupled qubits.

Example 2. *Consider again the qubit layout shown in Fig. 3. The MD between the mutually coupled qubits* q_1 *and* q_8 *with coordinates* $(3, 0)$ *and* $(2, 1)$ *is* $|3 - 2| + |0 - 1| = 2$. *Similarly, for the pair of uncoupled qubits, say* q_1 *and* q_{15} *with the corresponding coordinates* $(3, 0)$ *and* $(3, 2)$ *respectively, the MD is also* 2 *(*$= |3 - 3| + |0 - 2|$*).*

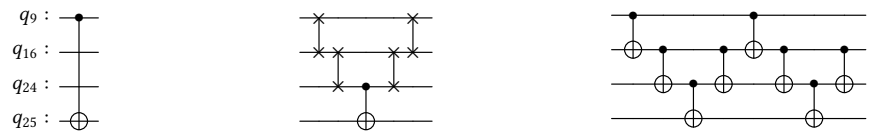
For some logical to physical qubit mapping, the pair of qubits (q_i, q_j) corresponding to a 2-qubit gate operation may be either adjacent (i.e. coupled) or non-adjacent (i.e. uncoupled). All non-native CNOT operations on uncoupled qubits can be evaluated only as a sequence of coupled gate operations. The possible alternatives include using Swap gates [33], using Remote-CNOT templates [24], or a combination of both [35]. The additional gate overhead incurred to evaluate such non-native CNOT operations is directly proportional to the distance $d_{i,j}$ between the uncoupled qubits q_i and q_j. For the proposed hexagonal qubit layout and the coordinate convention used, the distance between q_i and q_j can be computed as:

$$d_{i,j} = max\left(|y_i - y_j|, \frac{MD(q_i, q_j)}{2}\right) - 1 \tag{3}$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of qubits q_i and q_j respectively. The term $|y_i - y_j|$ in RHS of Eqn. (3) is introduced due to the inconsistency in MD between a pair of coupled or uncoupled qubits. The distance measure using Eqn. (3) is computationally more elegant than the one presented in [4, Eqn. (8)]. While the approach [4, Eqn. (8)] requires 4 addition/subtraction and two comparison operations, Eqn. (3) involves one comparison and one division-by-2 operations along with similar addition/subtraction operations, and a *division-by-2* operation which is a logical right shift of one place is less expensive than a comparison operation. The following example illustrates the calculation of distance between qubits.

Example 3. *Consider the qubits* q_9 *and* q_{25} *in the hexagonal qubit layout shown in Fig. 3, whose coordinates are* $(4, 1)$ *and* $(8, 3)$ *respectively as per Eqn.* (1)*. Using Eqn.* (2)*, we get* $MD(q_9, q_{25}) = 6$. *According to Eqn.* (3)*, the qubit distance will be*

$$d_{9,25} = max\left(|1 - 3|, \frac{6}{2}\right) - 1 = 2.$$



(a) (b) (c)

Fig. 4. (a) A non-native CNOT(q_9, q_{25}) operation, (b) realization using Swap gates, and (c) realization using Remote-CNOT operation.

The distance estimate computed using Eqn. (3) can be used to calculate the gate overhead in terms of additional CNOT operations required to make the operation NN-compliant. The calculation of the gate overheads using Swap gates and CNOT templates are discussed below.

a) Suppose that we use Swap gates to make any arbitrary 2-qubit gate operation NN-compliant, and assume that we restore the original qubit order after the gate operation. Since each Swap gate requires three CNOT gates for realization, the gate overhead (GS) will be:

$$GS(q_i, q_j) \quad = \quad 6 \times d_{i,j} + 1 \tag{4}$$

b) Suppose we use CNOT templates to make a 2-qubit gate operation NN-compliant. For this approach, the gate overhead (GR) will be:

$$GR(q_i, q_j) \quad = \quad 4 \times d_{i,j} \tag{5}$$

While either of Eqn. (4) or (5) can be used for estimating the gate overhead, Eqn. (5) provides strict *upper bound* compared to Eqn. (4). The following example illustrates the realization of non-native CNOT operation satisfying Eqns. (4) and (5).

Example 4. *Consider the non-native CNOT(q_9, q_{25}) operation on the hexagonal layout of Fig. 3, as shown in Fig. 4a. The estimated distance $d_{9,25} = 2$, and a possible implementation using Swap gates requires 12 ($= 6 \times 2$) additional CNOT gates as shown in Fig. 4b. Similarly, the equivalent Remote-CNOT realization using additional 7 ($= 4 \times 2 - 1$) CNOT operations is shown in Fig. 4c.*

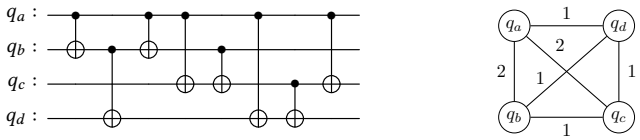
In the present work, we consider Eqn. (3) together with Eqn. (5) for estimating gate overhead in realizing non-native CNOT operations and propose the qubit mapping approach in the next section.

4 PROPOSED QUBIT MAPPING APPROACH

Consider a hexagonal layout of size $W \times H$, consisting of $n = W.H$ physical qubits. A quantum circuit with m logical qubits can be mapped to the physical qubits in $\binom{n}{m}$ ways. We propose a graph-based mapping approach that reduces the number of non-native CNOT operations, and hence also the additional gate overhead.

4.1 Physical Qubit Selection

For an m-qubit quantum circuit, we first create a weighted qubit interaction graph (QIG) where each node q_i, $0 \leq i < m$, represents a logical qubit, and an edge between a pair of nodes (q_i, q_j) with weight $w_{i,j}$ denotes the number of 2-qubit gates between them. Fig. 5 shows a quantum circuit and the corresponding QIG.



(a) (b)

Fig. 5. (a) A quantum circuit consisting of CNOT gates, and (b) the weighted qubit interaction graph.

For a given quantum circuit with m logical qubits $\{q_0^l, q_1^l, \ldots, q_{m-1}^l\}$, same number of physical qubits $\{q_0^p, q_1^p, \ldots, q_{m-1}^p\}$ must be selected during the mapping:

$$\pi : \{q_0^l, q_1^l, \ldots, q_{m-1}^l\} \xrightarrow[mapping]{physical} \{q_0^p, q_1^p, \ldots, q_{m-1}^p\}$$

such that it minimizes the mapping cost (MC) defined as

$$MC(\pi) = \sum_{q_i, q_j \in \{q_0^l, \ldots, q_{m-1}^l\}} GR(\pi(q_i), \pi(q_j)) \times w_{i,j} \tag{6}$$

where $\pi(q_i)$ denotes the mapping of i^{th} logical qubit, $GR(\pi(q_i), \pi(q_j))$ indicates the number of CNOT gates required for mapping as per Eqn. (5), and $w_{i,j}$ represents the weight of the edge (q_i, q_j) in the QIG.

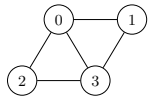
We start with the smallest $W \times H$ physical layout, such that $m \leq W.H$. Since an m-qubit circuit can be mapped to m selected physical qubits in $m!$ ways, we use an evolutionary approach (*genetic algorithm*) to find the best initial mapping (π_0) using the cost function defined in Eqn. (6). For a given netlist, we start with a randomly generated initial population $P = \{\pi^1, \pi^2, \ldots, \pi^n\}$ of size n, ordered by their respective mapping costs. From every generation, the next generation is obtained by (i) copying the m_1 best solutions from the present generation, (ii) performing crossover on m_2 pairs of solutions with some probability, and (iii) performing mutation operation on m_3 solutions with some probability, such that $n = m_1 + m_2 + m_3$. The process terminates after exploring a maximum predefined number of generations. The crossover between a solution pair is carried out by flipping the order of qubits that appear in the RHS of a randomly selected crossover point while the LHS are kept unchanged. The mutation on a solution is carried out by randomly performing operations like left-shift, right-shift, swap, etc. modifying the qubit order. The best solution from the last generation is taken as the initial mapping. The following example illustrates the process.

Example 5. *Consider a* 2×2 *hexagonal layout of qubits as shown in Fig. 6a, on which the quantum circuit shown in Fig. 5a is mapped. The logical qubits are* $\{q_a, q_b, q_c, q_d\}$*, while the physical qubits are* $\{q_0, q_1, q_2, q_3\}$*. The logical-to-physical mapping can be carried out in* 4! *possible ways,* $\{\pi^1, \pi^2, \ldots, \pi^{24}\}$*. Fig. 6b shows three such physical mappings (*π^1*,* π^2*, and* π^3*), and the corresponding mapping costs computed using Eqn.* (6)*.* π^1 *has the lowest mapping cost, and can be selected as the initial mapping (*π_0*).*

The initial mapping (π_0) as obtained may contain non-native CNOT operations, and three different methods are presented in the next subsection to realize them.

4.2 Realization of Non-native CNOT Operations

For a given initial qubit mapping (π_0) of a quantum circuit on a hexagonal layout, all the non-native CNOT operations can be realized using either of three approaches: (a) Remote-CNOT operations



(a)

π^i	q_a	q_b	q_c	q_d	MC
π^1	q_3	q_0	q_2	q_1	4
π^2	q_2	q_1	q_3	q_0	8
π^3	q_1	q_2	q_0	q_3	8

(b)

Fig. 6. (a) A 2×2 hexagonal qubit layout, and (b) mapping costs for three different logical-to-physical qubit mappings.

(called *Method-1*), (b) Swap operations (called *Method-2*), and (c) combination of both (called *Method-3*).

4.2.1 Method-1: Remote-CNOT Operations. We traverse the quantum circuit from left to right, and verify the NN-compliance of all the 2-qubit gates with respect to the initial mapping (π_0). Each 2-qubit gate that violates the coupling-map constraint of the hexagonal layout are replaced by the Remote-CNOT templates [24]. The following example illustrates the method.

Example 6. *Consider again the mapping of the quantum circuit of Fig. 5a on the 2×2 hexagonal layout shown in Fig. 6a. For the initial mapping $\pi_0 : \{q_a, q_b, q_c, q_d\} \rightarrow \{q_3, q_0, q_2, q_1\}$, there is a violation in the 7^{th} gate CNOT(q_c, q_d), as q_c and q_d are mapped to non-coupled physical qubits q_2 and q_1, respectively. Fig. 7 shows the netlist obtained using Method-1 by replacing the CNOT gate using a network of 4 CNOT gates (surrounded by dashed rectangle) implementing the Remote-CNOT(q_2, q_1) operation.*

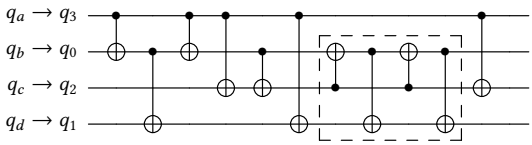


Fig. 7. Qubit mapping using Method-1 for initial mapping $\pi_0 : \{q_a, q_b, q_c, q_d\} \rightarrow \{q_3, q_0, q_2, q_1\}$.

The use of CNOT template for NN-violation of a gate does not provide any scope for exploiting qubit adjacency for subsequent gates. We shall see in the next subsection that the used of Swap or combination of Swap and CNOT template together can have a greater impact.

4.2.2 Method-2: Swap Operations. Here we use a k-gate lookahead while traversing the circuit, and insert Swap gates before each CNOT gate operating on non-coupled qubits for the current physical mapping. The method starts with the initial mapping π_0, and verifies each 2-qubit gate that violates the coupling map constraint. If such a violation exists at time-step t (≥ 0), Swap gates are inserted to mitigate the coupling violation updating the current mapping π_i to π_{i+1}. This is done in a way to reduce the gate overhead for subsequent 2-qubit gates that follow the gate G_t operating at t^{th} time-step as depicted in Fig. 8. Three alternate permutations π^1, π^2 and π^3 are shown, each requiring two Swap gates.

Among the alternate ways to insert the Swap gates before G_t, we select the one that minimizes the cost of mapping the k subsequent gates in the circuit, i.e. we set $\pi_{i+1} = \pi^l$ such that

$$MC(\pi^l) = Min\{MC(\pi^1), MC(\pi^2), \ldots, MC(\pi^m)\} \tag{7}$$

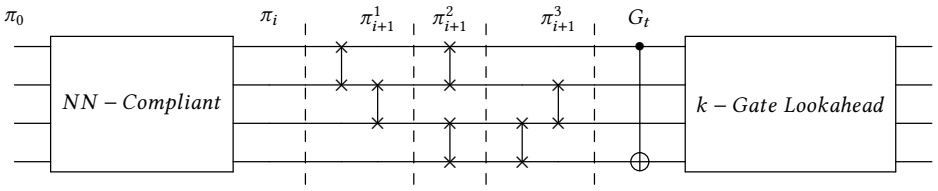


Fig. 8. Alternate permutations in lookahead approach.

where MC represents the mapping cost as per Eqn. (6), and m denotes the number of possible ways to insert Swap gates before G_t. For a CNOT(q_i, q_j) operation on uncoupled qubits q_i and q_j the value of m can be calculated as $m = d_{i,j} + 1$, where $d_{i,j}$ indicates the distance between qubits q_i and q_j as per Eqn. (3).

The number of possible paths to be considered for inserting the Swap gates increases on a 2-D plane based on the degree of coupling and the distance between the uncoupled qubit pair. For a pair of qubits q_i and q_j placed at coordinates (x_i, y_i) and (x_j, y_j) respectively on a hexagonal layout, the number of paths P_{ij} can be computed as

$$P_{ij} = \begin{cases} \frac{\left(\frac{|x_i - x_j| - |y_i - y_j|}{2} + |y_i - y_j|\right)!}{\left(\frac{|x_i - x_j| - |y_i - y_j|}{2}\right)! |y_i - y_j|!} & \text{if } x_i \neq x_j; \\ 2^{\left\lfloor \frac{|y_i - y_j|}{2} \right\rfloor} & \text{otherwise.} \end{cases} \tag{8}$$

An example is given below to illustrate this process.

Example 7. *Fig. 9 shows a* 3×3 *hexagonal qubit layout. The number of paths between* q_0 *and* q_8*, located at coordinates* $(1, 0)$ *and* $(5, 2)$ *respectively, according to Eqn. (8) is*

$$\frac{\left(\frac{|1-5| - |0-2|}{2} + |0-2|\right)!}{\left(\frac{|1-5| - |0-2|}{2}\right)! |0-2|!} = \frac{3!}{1!2!} = 3.$$

This can be verified from Fig. 9 by tracing the green colored directed lines. Similarly, the number of paths between the qubits q_1 *and* q_7 *at coordinates* $(3, 0)$ *and* $(3, 2)$*, according to Eqn. (8) is*

$$2^{\left\lfloor \frac{|0-2|}{2} \right\rfloor} = 2 \qquad (\textit{since } x_i = x_j).$$

These paths are highlighted in Fig.9 with red colored dashed directed lines.

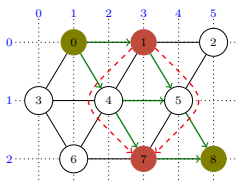


Fig. 9. Number of paths on a hexagonal layout of size 3×3.

In the present implementation of *Method-2*, all of these paths are examined to identify the best path for inserting Swap gates. The objective is to have maximum reduction in NN-violation over k subsequent 2-qubit gates.

4.2.3 Method-3: Combination of Remote-CNOT and Swap Gates. This is similar to *Method-2* where we consider the insertion of Swap gates before a 2-qubit gate operation $G_t(q_i, q_j)$ on two uncoupled qubits q_i and q_j evaluating all possible paths as obtained using Eqn. 8, based on the current permutation π_i by looking ahead k gates. For the current method the various possibilities that have been explored are:

i) Replace G_t using Remote-CNOT template as per *Method-1*.
ii) Insert Swap gate(s) before G_t in $d_{ij} + 1$ ways as per *Method-2*.
iii) Insert Swap gates and Remote-CNOT template for the gate G_t in $2(d_{ij} - 1)$ ways as shown below. Two alternatives π^1 and π^2 are shown in Fig. 10.

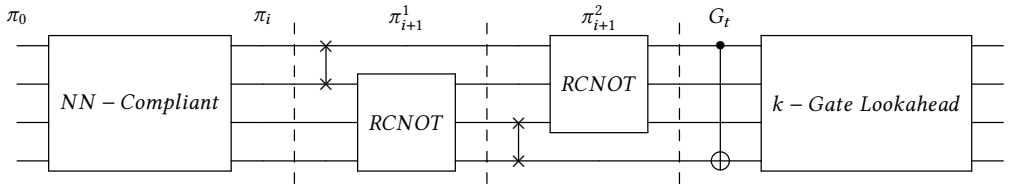


Fig. 10. Swap gates and Remote-CNOT template insertion.

While *Method-1* provides a *strict upper bound* of the number of CNOT gates required as overhead, both *Method-2* and *Method-3* try to optimize this upper bound by employing a lookahead strategy within the selected lookahead window at the expense of increased runtime. Depending on the number of qubits and 2-qubit gates, the mapping time also varies. A comparative study on this is presented next.

4.3 Complexity Anlysis

The proposed mapping approaches have two phases, which begin by finding an initial mapping (π_0) of the physical qubits ($q_i^l \rightarrow q_j^p$) followed by insertion of additional gates using either *Method-1* or *Method-2* or *Method-3* to mitigate the coupling constraint violation of 2-qubit gates due to π_0. Since the solution space for finding π_0 is $O(n!)$, an evolutionary approach is adopted to obtain a good solution for the first phase in reasonable time. It may be noted that the runtime of the evolutionary approach depends on the number of generations (G), population size (P), and member (solution) size (n) besides the complexity of the fitness function that is $O({}^nC_2)$ according to Eqn. (6). For a n-qubit circuit, the use of Genetic Algorithm reduces the search space to $O(GP\,{}^nC_2)$ in general.

In the second phase, the mapped netlist is obtained by traversing the entire circuit from the beginning by updating 2-qubit operations that do not adhere to the coupling constraints with respect to the current mapping π_i. In processing the circuit using *Method-1* π_0 remains unchanged, and the complexity of updating a 2-qubit operation using Remote-CNOT template depends on the distance between the operating qubit pair. In other words, the mapping time is upper bounded by $O(n_g d_{\max})$, where n_g denotes the number of 2-qubit gates in the original circuit and $d_{\max}(= W + \frac{H}{2} - 2)$ is the maximum distance between qubit pair on a $W \times H$ ($\geqslant n$) hexagonal layout.

Both *Method-2* and *Method-3* consider a lookahead window of size k, and may update the current mapping $\pi_i \rightarrow \pi_{i+1}$ after processing each 2-qubit gate. Since there are $d_{\max} + 1$ possible solutions according to Eqn. (7), and ${}^nC_2$ is the evaluation cost of each permutation owing to Eqn. (6), the worst-case complexity of *Method-2* is $O(n_g(d_{\max}+1)\,{}^nC_2)$. Similarly, *Method-3* that includes solutions from *Method-1* and *Method-2* with additional $2(d_{\max} - 1)$ hybrid solutions leads to an increased solution space of size $3 \times d_{\max}$ $(= 1 + d_{\max} + 1 + 2(d_{\max} - 1))$, i.e. runtime is upper bounded by $O(3n_g d_{\max}\,{}^nC_2)$.

5 EXPERIMENTAL EVALUATION

All the proposed approaches have been implemented in C++ and run on a system with an AMD Ryzen 7 PRO 5850U processor with Radeon Graphics running at 1.90GHz, 48GB RAM, 1TB SSD and Windows 10 Pro operating system. The implementations include the approaches for making a given quantum circuit NN-compliant, as well as the evolutionary approach (viz., genetic algorithm) for generating the initial global ordering of the qubits. Through experimental studies and parameter tuning, the parameters of the genetic algorithm are chosen as: population_size of 30, number of generation as 500, crossover rate of 0.7, and mutation rate of 0.1.

The proposed qubit mapping approaches have been evaluated with respect to various benchmark circuits. Two broad comparisons have been performed. Firstly, we have compared our proposed method with various qubit mapping approaches available in Qiskit. Secondly, we have compared the results with a very recent work by Chang et al. [4], using the same set of benchmarks from RevLib [31] as used by the authors. For a fair comparison, all the benchmarks (in reversible netlist format) have been transpiled using an approach similar to that used in [4]. The platform used for the purpose is Qiskit 0.19.0 [6] with Python 3.16.12, where the Qiskit parameters *basis_gates* and *optimization_levels* are set to $[cx, u3]$ and 0, respectively. The same classification of the benchmarks as used in [4] have been used, based on the number of CNOT operations (#CNOT) present in the circuit netlists.

$$\textit{Tiny} : 1 \leqslant \#\text{CNOT} \leqslant 50 \qquad \textit{Small} : 51 \leqslant \#\text{CNOT} \leqslant 200$$
$$\textit{Medium} : 201 \leqslant \#\text{CNOT} \leqslant 8000 \qquad \textit{Large} : \#\text{CNOT} \geqslant 8001$$

The number of benchmarks in the *Tiny*, *Small*, *Medium*, and *Large* categories considered for analysis are 62, 54, 61 and 62, respectively.

As discussed earlier, three alternate mapping approaches have been explored as part of this work: (a) *Method-1*: using only Remote-CNOT templates, (b) *Method-2*: using only Swap operations, (c) *Method-3*: using a combination of Remote-CNOT templates and Swap operations. For a given benchmark in the form of a quantum gate netlist, the initial qubit mapping (π_0) obtained using the global ordering approach is fed to each of the three methods. The NN-compliant mapped netlists are generated in this process. Fig. 11 shows the overall flow of the experimental evaluation. For fair comparison the parameters W and H, denoting the size $W \times H$ of the hexagonal grid, are chosen to be same as those used in [4]. In all these experiments the lookahead parameter (k) is set to 20 for compiling quantum circuits.

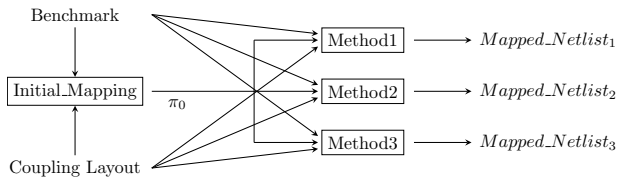


Fig. 11. Overall flow of the experiments.

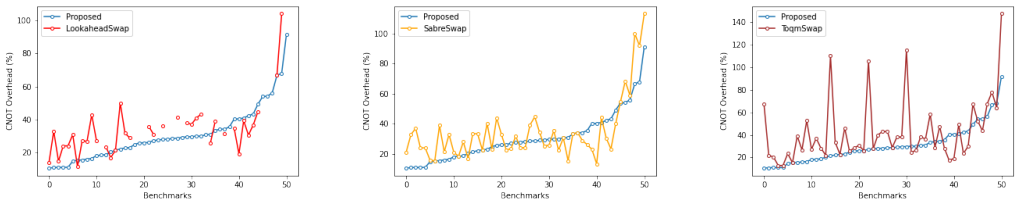
5.1 Comparison with Existing General Approaches

We now compare the performance of the proposed method with three existing approaches: Lookahead [13], Sabre [18] and TOQM [34]. We have considered the implemented versions of these three methods present in QISKIT [**?**] for compiling various benchmarks. For the purpose of comparison, we have chosen 51 small benchmarks consisting of a maximum of 200 2-input gates. The reason for

choosing only small benchmarks is due to fact that for medium and large benchmarks the lookahead method was not able to produce results within a preset threshold of 25 minutes. Fig. 12a, 12b, and 12c show the gate overheads of lookahead, Sabre and TOQM methods over our proposed method.

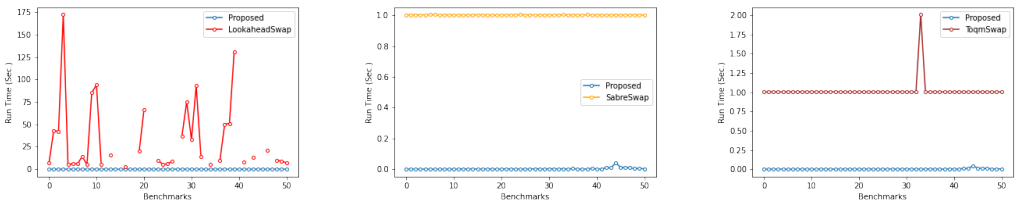
(1) For the lookahead approach, out of 51 benchmarks 10 gave better results, while the proposed method gave better results for 25 benchmarks. For two benchmarks, both the approaches gave the same result. For the remaining 14 benchmarks, the lookahead method could not generate any result within a preset 3 minutes timeout interval.
(2) For the Sabre approach, 17 out of the total 51 benchmarks gave better results, while the proposed method gave better results for 28 benchmarks. For the remaining 6 benchmarks, the results obtained were similar.
(3) For the TOQM approach, 11 out of the total 51 benchmarks gave better results, while the proposed method gave better results for 35 benchmarks. For the remaining 5 benchmarks, the results obtained were similar.

Fig. 13a, 13b, and 13c show the runtime of lookahead, Sabre and TOQM methods over our proposed method. From the figures it is evident that our method is better with respect to runtime compared to the three existing methods. This clearly shows why we require a coordinate system for the qubit mapping targeted to hexagonal grids, as existing methods are not able to exploit the structural regularity of such grids and also incur higher execution times.



(a) Lookahead (b) Sabre (c) TOQM

Fig. 12. Gate overhead comparison of proposed Method-2 with various QISKIT methods



(a) Lookahead (b) Sabre (c) TOQM

Fig. 13. Run-time comparison of proposed Method-2 with various QISKIT methods

5.2 Comparison with Existing Hexagonal Approaches

In this section we compare the results of our approach with a recent hexagonal based mapping approach [4]. As mentioned earlier we classify the benchmarks into four categories based on their

sizes, viz. *Tiny*, *Small*, *Medium*, and *Large*. For the purpose of fair comparison with Chang et al. [4], the benchmarks are mapped to 2-D hexagonal grids of similar sizes (i.e. same width and height).

Fig. 14a shows a summary of the results obtained with the proposed mapping methods for *Tiny* benchmarks, as well as those reported in [4]. The individual bars in the figure indicates the average percentage of additional CNOT gates required (μ) to satisfy the NN-constraints. The associated error bars indicate the corresponding one standard deviation (σ). From Fig. 14a it can be verified that the proposed *Method-2* and the approach [4] with [μ, σ] values as [17.70, 41.06] and [16.70, 34.92], respectively have marginal effect compared to *Method-3* with [14.18, 30.22] to bring down the upper bound of CNOT overhead set by *Method-1* with [17.94, 35.03] for *Tiny* benchmarks.

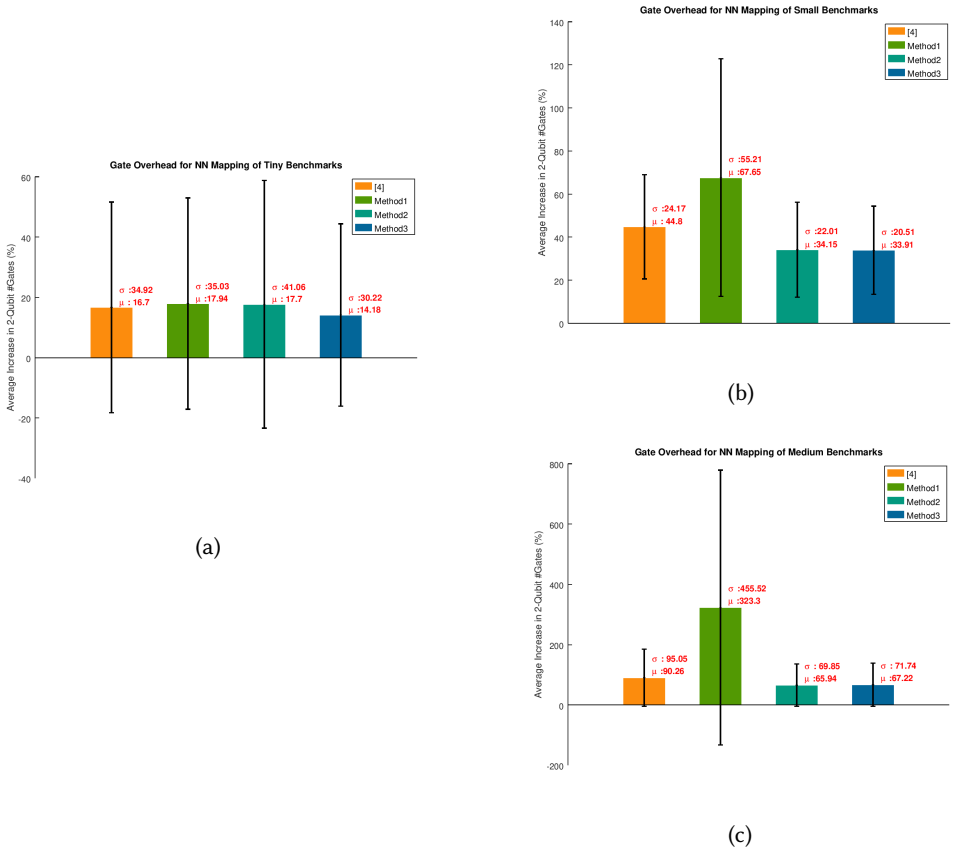


Fig. 14. Comparing gate overheads for (a) *tiny*, (b) *small* and (c) *medium* benchmarks.

Similarly, Fig. 14b shows the CNOT gate overheads (μ) and the error bars (σ) of respective approaches for *Small* benchmarks. From the diagram, it is clear that although the approach [4] which has [44.8, 24.17] curtails the gate overhead of *Method-1* with [67.65, 55.21], both *Method-2* and *Method-3* with [34.15, 22.01] and [33.91, 20.51] respectively gives much better realizations.For *Medium* benchmarks, the result summary of the [μ, σ] pairs for all the approaches are shown in Fig. 14c. Both *Method-2* with [65.94, 69.85] and *Method-3* with [67.22, 71.74] provide better realizations in terms of minimal additional CNOT gates compared to the approach [4] with [90.26, 95.05] over *Method-1* with [323.3, 455.52].

It can also be observed that the performance of *Method-3* is narrowly degraded compared to *Method-2* for medium size benchmarks with not more than 8000 CNOT gates. This is due to the fact that most of these medium size benchmarks contain large number of qubits (28 on the average), and *Method-3* that explores a 3 times larger solution space than *Method-2* requires a larger lookahead window ($k > 20$) for better results.

5.3 Evaluation of Scalability

In order to evaluate the scalability of the four NN-mapping approaches, the results obtained by running them on *Large* benchmarks are presented in Table 1. The first four columns of the table indicate the serial number, benchmark name, number of qubits (n), and number of CNOT operations in the original netlist (#CNOT), respectively. The column $W \times H$ indicates the dimension for hexagonal grid mapping. The next two columns represent the number of additional CNOT gates required for NN-mapping (#CNOT′) and the corresponding run time in seconds, $t(s)$. The value of #CNOT′ is estimated as $3 \times$ #Swap, where #Swap denotes the number of Swap gates required as per [4]. The next column indicates the run time of proposed global ordering approach in seconds to obtain initial mapping (π_0). The next three column pairs, $\big($#CNOT′, $t(s)\big)$ denote the additional CNOT gate overhead and runtime of the three proposed approaches. The next two columns indicate the improvements that are obtained, with the column Δ_m indicating the method with the least gate overhead, the column $\Delta_g(\%)$ indicates the corresponding percentage improvement over approach [4]. The last column $\Delta_{t(s)}$ denotes the time ratio of runtime observed for the best method over [4]. The time ratio is computed as

$$\Delta_{t(s)} \quad = \quad \frac{t(s)_{\pi_0} \; + \; t(s)_{Mi}}{t(s)_{[4]}} \tag{9}$$

It can be observed from Table 1 that *Method-2*, *Method-3* and the approach in [4] performs well to pull down the upper bound set by *Method-3* for *Large* benchmarks. *Method-3* provides better gate reduction compared to *Method-2* and [4] for the benchmarks mostly comprising of 16 or less qubits. For circuit with large number of qubits *Method-2* outperforms *Method-3* and [4]. On an average, 28.3% (maximum 43%) reduction in CNOT operations (column $\Delta_g(\%)$) is observed over [4]. For one benchmark *ryy6_256*, the proposed approaches fail to provide better result as compared to [4].

Both *Method-2* and *Method-3* experience an increase in runtime over [4] for 22 benchmarks, while for the remaining 40 benchmarks our method shows better runtime. Among these 22 benchmarks, our method suffers runtime overhead not exceeding 3.5 seconds for 18 benchmarks and maximum of 4 (approx.) minutes for the largest benchmark *misex3c_244* with 6 million (approx.) CNOT gates. Since the number of possibilities is directly proportional to the distance measure d_{ij} (computed using Eqn. (3)) between two qubits, q_i and q_j, for circuits with large number of qubits both *Method-2* and *Method-3* experience a little increase in runtime over [4].

5.4 Evaluation of Effective Lookahead Parameter

Another set of experiments is conducted over the *Large* benchmarks to analyze the effectiveness of the lookahead window size (k) for *Method-2* and *Method-3*, and the outcome of this experiment is presented in Table 2. In the first part, both the Swap-based methods, i.e. the proposed *Method-2* and the method from [4], are compared. As the authors in [4] consider a variable lookahead window in the range from 1 to 10 for reporting the best results, we have assigned a constant value 10 as the lookahead parameter for a fair comparison.

The first column presents the serial number of all the benchmarks considered previously for scalability experiment (see Table 1). The next three columns represent the mapping upper bound (UB), CNOT gate overhead (#CNOT), and runtime in seconds ($t(s)$) from [4]. Setting $k = 10$ in

Table 1. Comparison of NN-Mapping of Large Benchmarks to Hexagonal Architecture w.r.t. [4]

					[4]		π_0	Method-1 (M1)		Method-2 (M2)		Method-3 (M3)		Improvement		
Sl.No.	Benchmark	n	#CNOT	$W \times H$	#CNOT′	t(s)	t(s)	#CNOT′	t(s)	#CNOT′	t(s)	#CNOT′	t(s)	Δ_m	$\Delta_g(\%)$	$\Delta_t(s)$
1	9symml_195	10	39296	4 × 3	8916	2.72	0.70	23468	0.19	7323	0.79	7099	0.97	M3	20.38	0.61
2	add6_196	19	24745	5 × 4	10218	2.20	1.89	44759	0.20	6873	1.91	6323	2.38	M3	38.12	1.94
3	alu2_199	16	44982	4 × 4	12354	3.38	1.41	58874	0.30	9411	1.90	9065	2.54	M3	26.62	1.17
4	alu3_200	18	17787	5 × 4	5394	1.40	1.66	35461	0.15	4149	1.04	3823	1.30	M3	29.12	2.11
5	alu4_201	22	1858472	5 × 5	439200	150.96	5.67	4409910	17.01	395787	141.37	446213	233.03	M2	9.88	0.97
6	apex4_202	28	1249592	6 × 5	467376	117.73	5.42	5242414	17.65	346179	265.98	313290	267.22	M3	32.97	2.32
7	apla_203	22	26336	5 × 5	8496	2.30	2.46	76290	0.37	6849	3.16	6310	3.16	M3	25.73	2.44
8	clip_206	14	39082	4 × 4	11388	2.85	1.31	75087	0.33	9288	1.52	7700	1.69	M3	32.38	1.06
9	cm151a_211	28	12788	6 × 5	3336	1.24	2.50	35414	0.14	3054	1.91	3459	2.86	M2	8.45	3.57
10	cm85a_209	14	9558	4 × 4	3522	0.78	1.43	14516	0.08	2559	0.51	2214	0.56	M3	37.14	2.55
11	cmb_214	20	49136	5 × 4	10410	3.97	2.03	95740	0.36	9714	2.65	11206	4.40	M2	6.69	1.18
12	co14_215	15	344008	4 × 4	75294	26.05	2.14	321750	1.96	68307	12.82	107222	27.51	M2	9.28	0.57
13	cu_219	25	16396	5 × 5	3972	1.27	2.56	29608	0.15	3876	1.88	3256	2.13	M3	18.03	3.69
14	cycle10_2_110	12	9122	4 × 3	1878	0.63	1.04	13404	0.06	1815	0.25	2386	0.43	M2	3.35	2.07
15	cycle17_3_112	20	1376037	5 × 4	275616	114.71	4.62	3343793	11.92	249396	65.23	440948	187.31	M2	9.51	0.61
16	dist_223	13	37510	4 × 4	10698	2.79	1.11	64590	0.29	8751	1.71	7493	1.76	M3	29.96	1.03
17	dk17_224	21	13154	5 × 5	3990	1.08	2.12	35970	0.14	2832	0.87	3336	1.41	M2	29.02	2.77
18	ex1010_230	20	2165828	5 × 4	596298	174.21	6.02	6897736	25.21	491055	148.99	496186	217.51	M2	17.65	0.89
19	example2_231	16	44982	4 × 4	12354	3.36	1.37	54670	0.34	10326	2.87	9145	3.38	M3	25.98	1.41
20	f51m_233	22	1759626	5 × 5	406422	139.57	5.29	2856826	12.96	359547	125.98	507495	259.30	M2	11.53	0.94
21	hwb7_59	7	8423	3 × 3	3126	0.56	0.43	8637	0.06	1806	0.15	1917	0.20	M2	42.23	1.05
22	hwb8_113	8	29034	3 × 3	11856	2.17	0.66	41897	0.28	7254	0.86	7074	1.04	M3	40.33	0.79
23	hwb8_114	8	26169	3 × 3	9972	1.87	0.53	35537	0.18	6120	0.54	6046	0.65	M3	39.37	0.63
24	hwb8_116	8	12245	3 × 3	5310	0.99	0.53	15994	0.08	3177	0.28	3100	0.35	M3	41.62	0.89
25	hwb9_119	9	95787	3 × 3	38202	7.17	0.77	159399	0.79	24120	2.47	22967	3.10	M3	39.88	0.54
26	hwb9_121	9	95703	3 × 3	38118	7.21	0.74	158236	0.74	24267	2.39	23001	2.80	M3	39.66	0.49
27	hwb9_123	9	49777	3 × 3	19668	3.92	0.67	71532	0.35	12249	1.24	12025	1.54	M3	38.86	0.56
28	in0_235	26	1188570	6 × 5	281478	102.30	5.57	3903916	13.63	241503	117.15	330120	248.14	M2	14.20	1.20
29	in2_236	29	1042525	6 × 5	247038	90.90	6.55	4253388	14.35	218295	133.13	262098	250.87	M2	11.64	1.54
30	life_238	10	19936	4 × 3	5364	1.46	0.78	15384	0.16	3849	0.55	3699	0.54	M3	31.04	0.90
31	max46_240	10	22924	4 × 3	5868	1.51	0.80	15996	0.17	4512	0.66	4230	0.69	M3	27.91	0.99
32	misex3c_244	28	5678530	6 × 5	1382388	500.65	12.32	22667118	79.11	1220487	704.77	1530202	1380.63	M2	11.71	1.43
33	mlp4_245	16	16940	4 × 4	5676	1.38	1.36	27284	0.12	4113	1.19	3677	1.55	M3	35.22	2.11
34	plus127mod8192_162	13	526950	4 × 4	131688	40.66	2.72	693575	4.36	106953	16.58	104848	21.64	M3	20.38	0.60
35	plus63mod4096_163	12	163663	4 × 3	43878	11.81	1.51	224645	1.18	35580	4.85	35368	6.26	M3	19.39	0.66
36	plus63mod8192_164	13	330835	4 × 4	81210	25.42	1.73	454172	2.36	70626	10.70	62713	12.90	M3	22.78	0.58
37	rd84_253	12	9677	4 × 3	3120	0.76	1.03	10286	0.06	2526	0.35	1871	0.36	M3	40.03	1.83
38	root_255	13	17103	4 × 4	5190	1.28	1.10	27043	0.12	3894	0.60	3580	0.74	M3	31.02	1.43
39	ryy6_256	17	257389	5 × 4	57648	19.82	2.35	271052	1.55	59853	13.47	70012	22.73	M2	−3.82	0.80
40	sao2_257	14	76591	4 × 4	18108	5.56	1.41	143191	0.62	15243	2.54	15303	3.39	M2	15.82	0.71
41	sym10_262	11	99332	4 × 3	24108	7.09	1.00	82174	0.55	20241	2.41	18173	2.84	M3	24.62	0.54
42	sym9_193	10	39296	4 × 3	8916	2.70	0.76	23800	0.20	7176	0.79	7064	0.97	M3	20.77	0.64
43	tial_265	22	1963642	5 × 5	462780	162.31	5.80	3907526	16.20	424413	146.56	455134	237.22	M2	8.29	0.94
44	urf1_149	9	69324	3 × 3	56772	8.63	0.73	90164	0.61	37542	4.84	33975	4.48	M3	40.16	0.60
45	urf1_150	9	106039	3 × 3	40968	7.88	0.85	171326	0.83	25953	2.64	25482	3.15	M3	37.80	0.51
46	urf1_151	9	98477	3 × 3	38856	7.25	0.76	160514	0.81	24903	2.52	24156	3.07	M3	37.83	0.53
47	urf1_278	9	24916	3 × 3	22050	3.25	0.67	30980	0.17	14034	1.41	13035	1.69	M3	40.88	0.73
48	urf2_152	8	30180	3 × 3	21582	3.45	0.55	36064	0.22	15183	1.27	13725	1.46	M3	36.41	0.58
49	urf2_153	8	30013	3 × 3	12048	2.24	0.55	41066	0.21	7791	0.82	7167	0.83	M3	40.51	0.61
50	urf2_154	8	28403	3 × 3	11454	2.13	0.54	38570	0.21	6864	0.62	7071	0.92	M2	40.07	0.55
51	urf2_161	8	27460	3 × 3	18642	2.90	0.54	32546	0.21	11949	1.22	11409	1.29	M3	38.80	0.63
52	urf2_277	8	8873	3 × 3	8310	1.22	0.51	10085	0.05	5220	0.48	5132	0.58	M3	38.24	0.89
53	urf3_155	10	158808	4 × 3	134244	20.87	1.02	246548	1.42	85032	9.06	79233	11.26	M3	40.98	0.59
54	urf3_156	10	329500	4 × 3	114342	23.90	1.36	560178	2.66	76200	8.43	71026	10.30	M3	37.88	0.49
55	urf3_157	10	312746	4 × 3	109356	22.64	1.24	551112	2.50	71487	7.98	68380	9.51	M3	37.47	0.47
56	urf3_279	10	66435	4 × 3	51450	8.02	0.82	77604	0.43	30846	3.58	29220	4.26	M3	43.21	0.63
57	urf4_187	11	192024	4 × 3	131454	25.96	1.23	263392	1.47	83667	10.36	95514	15.25	M2	36.35	0.45
58	urf5_158	9	61656	3 × 3	48948	7.47	0.72	88292	0.47	31839	2.97	29276	3.77	M3	40.19	0.60
59	urf5_159	9	56096	3 × 3	18270	3.94	0.72	85998	0.43	12513	1.38	11604	1.43	M3	36.49	0.55
60	urf5_280	9	24200	3 × 3	17736	2.75	0.66	31583	0.17	11052	1.15	10620	1.43	M3	40.12	0.76
61	urf6_160	15	64440	4 × 4	67086	10.95	1.76	128864	0.67	44304	7.86	39401	9.72	M3	41.27	1.05
62	urf6_281	15	176567	4 × 4	57804	14.93	1.91	297425	1.35	47589	8.91	47302	12.59	M3	18.17	0.97

Method-2, the similar results in terms of the upper bound (UB), CNOT overhead (#CNOT) and runtime ($t(s)$) are presented in the next three columns, where like in [4], the upper bound for a given

circuit and for a given initial mapping π_0 is computed as

$$UB = 2 \sum_{q_i,q_j \in \{q^l_0,\ldots,q^l_{m-1}\}} d_{\pi_0(q_i),\pi_0(q_j)} \times w_{i,j} \tag{10}$$

where $d_{\pi_0(q_i),\pi_0(q_j)}$ represents the distance between physical qubits $\pi_0(q_i)$ and $\pi_0(q_i)$, and $w_{i,j}$ denotes the corresponding edge weight in the QIG. This is followed by a comparative analysis of upper bound ($\Delta_u(\%)$), gate overhead ($\Delta_g(\%)$) and runtime ratio ($\Delta_{t(s)} = t(s)_{Method-2}/t(s)_{[4]}$) between the approach [4] and *Method-2*. While the proposed evolutionary approach is unable to bring down the upper bound reasonably (on an average 2.4% reduction in UB and failing in 22 cases), a significant reduction in CNOT overhead is observed for almost all the benchmarks (on an average 25% reduction in CNOT overhead and failing in 3 cases) using *Method-2* over the approach [4] at the expense of a little increase in runtime (viz., average runtime ratio of 1.08).

In the second part of the experiment, the performances of both *Method-2* and *Method-3* are analyzed by running them for lookahead window size 10, 20, 30, 40 and 50, and picking the best results. The next six columns represent the best observed results from these 5 executions of both *Method-2* and *Method-3* in terms of CNOT overhead (#CNOT), best lookahead parameter (k_b) and average runtime ($t(s)$) in respective order. An improvement in terms of minimizing gate overhead ($\Delta_g(\%)$) and runtime ratio ($\Delta_{t(s)} = t(s)_{Method-3}/t(s)_{Method-2}$) are presented in the final two columns. It can be noticed that while *Method-2* provides better results for a smaller lookahead window (i.e. for $k = 10$ on average), *Method-3* gives better result when a larger lookahead window (i.e. for $k = 30$ on an average) is considered. Further *Method-3* provides better minimization of gate overhead (on an average 15.3% reduction in CNOT overhead and marginally failing in 8 cases) over *Method-2* with little higher runtime (i.e. 1.24 average runtime ratio).

Besides the lookahead parameter k that is used in both *Method-2* and *Method-3*, the parameters population size and number of generations considered for finding initial mapping (π_0) using evolutionary approach also affect the runtime as well as gate overhead. Setting these parameters with small values might reduce the runtime but gate overhead may increase — there is a trade off.

6 CONCLUSION

In this paper we have presented a graph-based approach for NN-compliant qubit mapping on a 2-D hexagonal grid. We have proposed an approach for distance measure between qubits based on hexagonal coordinate system, which is found to be more efficient compared to existing approaches. We show that the existing qubit mapping approaches are unable to exploit the structural regularity of the hexagonal grid and often suffer from higher run time, and incur extra gate overhead. Three alternate strategies have been explored for ensuring NN-compliance, viz. Remote-CNOT templates, Swap gates, and a combination of both. While the use of Remote-CNOT templates gives a strict upper bound of CNOT overhead for NN-compliance, the approach combining Swap gates and Remote-CNOT templates is found to generate better results as compared to Swap-based method. Due to assessing a larger solution space the adopted combined approach requires a larger lookahead window and suffers from marginally higher runtime to provide 15% better results on an average over Swap based approach that seems more effective over existing approach — an average improvement of 25% is observed in terms of CNOT overhead. Compared to the existing hexagonal approach our method suffers from a little higher runtime for benchmarks comprising of large number of qubits. The runtime can be improved by setting different parameters for the considered evolutionary and lookahead approach, which may also influence NN-transformation with an increase in CNOT overhead. As a future work, variability in the gate lookahead strategy can be incorporated to further improve the performance of the combined (Swap gates and Remote-CNOT templates) approach.

Table 2. Analysis of Effective Lookahead Window Size for both Method-2 and Method-3

	$k = 10$									$k = 10, 20, \cdots, 50$							
	[4]			Method-2			Improvement			Method-2			Method-3			Improvement	
Sl.No.	UB	#CNOT	t(s)	UB	#CNOT	t(s)	$\Delta_u(\%)$	$\Delta_g(\%)$	$\Delta_t(s)$	#CNOT	k_b	t(s)	#CNOT	k_b	t(s)	$\Delta_g(\%)$	$\Delta_t(s)$
1	14920	8916	2.72	14920	6771	2.13	0.00	24.06	0.78	6771	10	1.12	5022	30	1.19	25.83	1.06
2	29956	10218	2.20	26780	6972	5.32	10.60	31.77	2.42	6873	20	2.64	5870	30	3.31	14.59	1.26
3	31052	12354	3.38	33172	11097	4.97	−6.83	10.17	1.47	9411	20	2.78	7152	50	3.19	24.00	1.15
4	23378	5394	1.40	21890	4566	4.00	6.36	15.35	2.86	4149	20	1.58	3537	50	2.00	14.75	1.27
5	3778224	439200	150.96	2205040	407397	230.75	41.64	7.24	1.53	395787	20	193.65	265187	50	267.75	33.00	1.38
6	2868296	467376	117.73	2961240	334419	227.94	−3.24	28.45	1.94	334419	10	260.19	282156	30	325.00	15.63	1.25
7	42624	8496	2.30	45916	7053	6.78	−7.72	16.98	2.95	6531	30	3.68	4805	50	4.31	26.43	1.17
8	50010	11388	2.85	47182	8967	2.91	5.65	21.26	1.02	8967	10	2.14	6786	40	2.27	24.32	1.06
9	18636	3336	1.24	17268	3477	4.72	7.34	−4.23	3.81	3054	20	2.96	2159	50	4.64	29.31	1.56
10	11334	3522	0.78	9048	2733	1.87	20.17	22.40	2.40	2487	50	0.59	1979	30	0.68	20.43	1.15
11	81200	10410	3.97	55308	9843	5.45	31.89	5.45	1.37	9531	40	3.97	7877	30	5.86	17.35	1.48
12	132496	75294	26.05	178240	76083	15.89	−34.52	−1.05	0.61	65778	30	17.05	42255	50	26.05	35.76	1.53
13	16360	3972	1.27	16224	3486	4.29	0.83	12.24	3.38	3339	40	2.08	2742	50	2.78	17.88	1.34
14	8308	1878	0.63	8232	2067	1.33	0.91	−10.06	2.11	1815	20	0.38	1525	50	0.52	15.98	1.37
15	2237322	275616	114.71	1714810	250686	90.92	23.35	9.05	0.79	249396	20	116.41	228699	50	256.50	8.30	2.20
16	46146	10698	2.79	41220	8274	2.69	10.67	22.66	0.96	8274	10	1.99	6856	30	2.12	17.14	1.06
17	21396	3990	1.08	22500	3570	3.47	−5.16	10.53	3.21	2832	20	1.37	2250	50	1.61	20.55	1.18
18	4324872	596298	174.21	4227980	487650	170.62	2.24	18.22	0.98	487650	10	194.62	335517	50	249.04	31.20	1.28
19	31052	12354	3.36	31344	11292	4.52	−0.94	8.60	1.35	10116	40	3.43	7197	30	3.55	28.86	1.04
20	1685928	406422	139.57	1510510	382347	159.46	10.40	5.92	1.14	347079	30	164.23	235243	50	255.20	32.22	1.55
21	6228	3126	0.56	5758	1644	0.56	7.55	47.41	1.00	1644	10	0.26	1641	10	0.30	0.18	1.18
22	28236	11856	2.17	26716	6564	1.22	5.38	44.64	0.56	6564	10	1.08	7074	20	1.32	−7.77	1.23
23	23414	9972	1.87	22914	5709	1.09	2.14	42.75	0.58	5709	10	0.86	6046	20	0.96	−5.90	1.12
24	10750	5310	0.99	10016	3171	0.80	6.83	40.28	0.81	3171	10	0.45	2927	10	0.54	7.69	1.20
25	102556	38202	7.17	101418	23733	3.02	1.11	37.87	0.42	23733	10	3.44	22967	20	4.07	3.23	1.18
26	102536	38118	7.21	101136	22620	2.93	1.37	40.66	0.41	22620	10	3.45	23001	20	4.18	−1.68	1.21
27	44680	19668	3.92	45088	12237	2.02	−0.91	37.78	0.52	12237	10	1.82	11568	30	2.21	5.47	1.22
28	2474500	281478	102.30	2255630	259083	155.12	8.85	7.96	1.52	241503	20	156.54	170838	50	251.17	29.26	1.60
29	2488864	247038	90.90	2159370	229968	159.82	13.24	6.91	1.76	218295	20	170.85	140526	50	275.68	35.63	1.61
30	8756	5364	1.46	8880	3720	1.19	−1.42	30.65	0.82	3720	10	0.63	2799	30	0.63	24.76	1.01
31	9540	5868	1.51	10352	4299	1.26	−8.51	26.74	0.83	4299	10	0.70	3445	30	0.75	19.87	1.08
32	13123796	1382388	500.65	13334900	1230327	845.67	−1.61	11.00	1.69	1220487	20	861.95	815465	50	1484.89	33.19	1.72
33	20796	5676	1.38	17588	3939	2.53	15.43	30.60	1.83	3939	10	1.65	3616	50	1.96	8.20	1.19
34	436206	131688	40.66	434286	110181	20.88	0.44	16.33	0.51	106953	20	27.45	73849	50	29.69	30.95	1.08
35	132988	43878	11.81	142468	33672	5.96	−7.13	23.26	0.50	33672	10	7.69	26063	30	8.69	22.60	1.13
36	276350	81210	25.42	279322	67449	12.04	−1.08	16.94	0.47	67449	10	16.04	44439	50	18.09	34.11	1.13
37	5264	3120	0.76	6262	2121	1.38	−18.96	32.02	1.82	2121	10	0.57	1871	20	0.64	11.79	1.12
38	18468	5190	1.28	15686	3843	1.77	15.06	25.95	1.38	3843	10	1.00	3050	30	1.08	20.63	1.08
39	119164	57648	19.82	156316	54894	15.74	−31.18	4.78	0.79	43281	30	17.83	36289	50	28.27	16.15	1.59
40	109640	18108	5.56	85108	16053	4.44	22.38	11.35	0.80	15243	20	4.55	11399	50	4.97	25.22	1.09
41	39768	24108	7.09	45308	19401	3.85	−13.93	19.52	0.54	19401	10	4.11	13969	50	4.30	28.00	1.05
42	14920	8916	2.70	16380	6993	1.64	−9.79	21.57	0.61	6993	10	1.35	5055	30	1.43	27.71	1.06
43	3256364	462780	162.31	2578420	426627	179.17	20.82	7.81	1.10	410304	30	233.90	274521	50	258.68	33.09	1.11
44	67152	56772	8.63	59080	35724	3.46	12.02	37.07	0.40	35724	10	5.06	31740	10	5.79	11.15	1.15
45	111086	40968	7.88	109744	24918	2.92	1.21	39.18	0.37	24918	10	3.56	25482	20	4.22	−2.26	1.19
46	103608	38856	7.25	102686	23628	2.66	0.89	39.19	0.37	23628	10	3.51	24156	20	3.95	−2.23	1.13
47	19996	22050	3.25	19996	12774	1.65	0.00	42.07	0.51	12774	10	1.92	12416	10	2.34	2.80	1.22
48	23392	21582	3.45	23584	13932	1.49	−0.82	35.45	0.43	13932	10	1.71	12853	10	2.02	7.74	1.18
49	26508	12048	2.24	26508	7038	1.03	0.00	41.58	0.46	7038	10	0.99	7167	20	1.12	−1.83	1.14
50	24908	11454	2.13	24884	6642	0.99	0.10	42.01	0.46	6642	10	0.90	6855	10	1.10	−3.21	1.23
51	22132	18642	2.90	21184	11706	1.32	4.28	37.21	0.46	11706	10	1.44	10808	10	1.66	7.67	1.15
52	6958	8310	1.22	6650	4953	0.83	4.43	40.40	0.68	4953	10	0.68	4728	10	0.83	4.54	1.21
53	151720	134244	20.87	154568	82170	8.00	−1.88	38.79	0.38	82170	10	11.60	75907	10	14.39	7.62	1.24
54	360188	114342	23.90	356580	75330	8.23	1.00	34.12	0.34	75330	10	11.27	70092	30	12.64	6.95	1.12
55	344300	109356	22.64	343488	70110	7.66	0.24	35.89	0.34	70110	10	10.64	66458	30	12.15	5.21	1.14
56	52208	51450	8.02	49580	29871	3.59	5.03	41.94	0.45	29871	10	4.84	29220	20	5.81	2.18	1.20
57	135048	131454	25.96	141896	90249	10.19	−5.07	31.35	0.39	73122	40	14.42	70096	40	16.86	4.14	1.17
58	56456	48948	7.47	57312	31134	3.00	−1.52	36.39	0.40	31134	10	3.94	27815	10	4.68	10.66	1.19
59	55500	18270	3.94	54916	11358	1.62	1.05	37.83	0.41	11358	10	1.71	11604	20	1.93	−2.17	1.12
60	20536	17736	2.75	19824	10581	1.45	3.47	40.34	0.53	10581	10	1.52	10049	10	1.79	5.03	1.17
61	84368	67086	10.95	85968	42657	7.98	−1.90	36.41	0.73	42657	10	9.98	38995	10	13.08	8.58	1.31
62	148974	57804	14.93	165224	43782	9.19	−10.91	24.26	0.62	43782	10	10.36	33816	50	13.22	22.76	1.28

REFERENCES

[1] M. Ahsan, S. A. Z. Naqvi, and H. Anwer. 2022. Quantum circuit engineering for correcting coherent noise. *Phys. Rev. A* 105 (Feb. 2022), 022428. Issue 2. https://doi.org/10.1103/PhysRevA.105.022428

[2] M. Amy, D. Maslov, M. Mosca, and M. Roetteler. 2013. A Meet-in-the-Middle Algorithm for Fast Synthesis of Depth-Optimal Quantum Circuits. *IEEE Trans. Comput.-Aided Des. Integr. Syst.* 32, 6 (June 2013), 818–830.

[3] A. Barenco, C. H. Bennet, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. Smolin, and H. Weinfurter. 1995. Elementary gates for quantum computation. *Phys. Rev. A* 52, 5 (Nov. 1995), 3457–3467.

[4] K. Y. Chang and C. Y. Lee. 2021. Mapping Nearest Neighbor Compliant Quantum Circuits onto a 2-D Hexagonal Architecture. *IEEE Trans. Comput.-Aided Des. Integr. Syst.* (2021), 1–14.

[5] Google QAI Contributors. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (01 Oct. 2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[6] Qiskit Contributors. 2024. Quantum computing with Qiskit. *arXiv preprint arXiv:2405.08810* (2024).

[7] A. A. A. de Almeida, G. W. Dueck, and A. C. R. da Silva. 2019. CNOT gate mappings to Clifford+T circuits in IBM architectures. In *Int'l Symp. on Multiple-Valued Logic*. 7–12.

[8] R. D. Delaney, M. D. Urmey, S. Mittal, et al. 2022. Superconducting-qubit readout via low-backaction electro-optic transduction. *Nature* 606, 7914 (01 June 2022), 489–493. https://doi.org/10.1038/s41586-022-04720-2

[9] Haowei Deng, Yu Zhang, and Quanxi Li. 2020. Codar: A Contextual Duration-Aware Qubit Mapping for Various NISQ Devices. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*. 1–6. https://doi.org/10.1109/DAC18072.2020.9218561

[10] E. Fredkin and T. Toffoli. 1982. Conservative logic. *Int'l Journal of Theoretical Physics* 21 (April 1982), 219–253.

[11] L.K. Grover. 1996. A fast quantum mechanical algorithm for database search. In *ACM Symp. on Theory of computing*. 212–219.

[12] J. Hilder, D. Pijn, O. Onishchenko, et al. 2022. Fault-Tolerant Parity Readout on a Shuttling-Based Trapped-Ion Quantum Computer. *Phys. Rev. X* 12 (Feb. 2022), 011032. Issue 1. https://doi.org/10.1103/PhysRevX.12.011032

[13] S. Jandura. 2018. Improving a Quantum Compiler. https://medium.com/qiskit/improving-a-quantum-compiler-48410d7a7084.

[14] A. Kole and K. Datta. 2017. Improved NCV Gate Realization of Arbitrary Size Toffoli Gates. In *Int'l Conf. on VLSI Design and Embedded Systems*. 289–294.

[15] A. Kole, K. Datta, and I. Sengupta. 2016. A Heuristic for Linear Nearest Neighbor Realization of Quantum Circuits by SWAP Gate Insertion Using N-Gate Lookahead. *IEEE J. Emerg. Sel. Topics Circuits Syst.* 6, 1 (Feb. 2016), 62–72.

[16] A. Kole, K. Datta, and I. Sengupta. 2018. A New Heuristic for N-Dimensional Nearest Neighbor Realization of a Quantum Circuit. *IEEE Trans. Comput.-Aided Des. Integr. Syst.* 37, 1 (Jan. 2018), 182–192.

[17] A. Kole, S. Hillmich, K. Datta, R. Wille, and I. Sengupta. 2020. Improved Mapping of Quantum Circuits to IBM QX Architectures. *IEEE Trans. Comput.-Aided Des. Integr. Syst.* 39, 10 (2020), 2375–2383.

[18] Gushu Li, Yufei Ding, and Yuan Xie. 2019. Tackling the Qubit Mapping Problem for NISQ-Era Quantum Devices. In *Int'l Conf. on Architectural Support for Programming Languages and Operating Systems* (Providence, RI, USA) *(ASPLOS '19)*. Association for Computing Machinery, New York, NY, USA, 1001–1014. https://doi.org/10.1145/3297858.3304023

[19] Xiangguo Li. 2013. Storage and addressing scheme for practical hexagonal image processing. *Journal of Electronic Imaging* 22, 1 (2013), 010502. https://doi.org/10.1117/1.JEI.22.1.010502

[20] D. Litinski, M. S. Kesselring, J. Eisert, and F. von Oppen. 2017. Combining Topological Hardware and Topological Software: Color-Code Quantum Computing with Topological Superconductor Networks. *Physical Review* 7, 3 (2017), 031048.

[21] M. Nielsen and I. Chuang. 2000. *Quantum Computation and Quantum Information*. Cambridge Univ. Press.

[22] P. Niemann, C. Bandyopadhyay, and R. Drechsler. 2021. Combining SWAPs and remote Toffoli gates in the mapping to IBM QX architectures. In *Design Automation and Test in Europe*. 1–6.

[23] Srikrishna Omkar, Seok-Hyung Lee, Yong Siah Teo, Seung-Woo Lee, and Hyunseok Jeong. 2022. All-Photonic Architecture for Scalable Quantum Computing with Greenberger-Horne-Zeilinger States. *PRX Quantum* 3 (July 2022), 030309. Issue 3. https://doi.org/10.1103/PRXQuantum.3.030309

[24] Md. M. Rahman and G. W. Dueck. 2015. Synthesis of Linear Nearest Neighbor Quantum Circuits. *arXiv preprint arXiv:1508.05430* (2015).

[25] A. C. Santos. 2016. The IBM Quantum Computer and the IBM Quantum Experience. *arXiv preprint arXiv:1610.06980v2* (2016).

[26] A. Shafaei, M. Saeedi, and M. Pedram. 2013. Optimization of Quantum Circuits for Interaction Distance in Linear Nearest Neighbor Architectures. In *Design Automation Conf.* 41:1–41:6.

[27] A. Shafaei, M. Saeedi, and M. Pedram. 2014. Qubit placement to minimize communication overhead in 2D quantum architectures. In *Asia and South Pacific Design Automation Conf.* 495–500.

[28] P. W. Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Symp. on Foundations of Computer Science*. 124–134.
[29] R. R. Shrivastwa, K. Datta, and I. Sengupta. 2015. Fast Qubit Placement in 2D Architecture Using Nearest Neighbor Realization. In *Int'l Symp. on Nanoelectronic and Information Systems*. 95–100.
[30] H. Tang et al. 2018. Experimental quantum fast hitting on hexagonal graphs. *Nature Photonics* 12, 12 (2018), 754–758.
[31] R. Wille, D. Große, L. Teuber, G. W. Dueck, and R. Drechsler. 2008. RevLib: An Online Resource for Reversible Functions and Reversible Circuits. In *Proc. Intl. Symposium on Multiple-Valued Logic*. Texas, USA, 220–225.
[32] R. Wille, O. Keszocze, M. Walter, P. Rohrs, A. Chattopadhyay, and R. Drechsler. 2016. Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits. In *Asia and South Pacific Design Automation Conf.* 292–297.
[33] R. Wille, A. Lye, and R. Drechsler. 2014. Optimal SWAP Gate Insertion for Nearest Neighbor Quantum Circuits. In *Asia and South Pacific Design Automation Conf.* IEEE, Suntec, Singapore, 489–494.
[34] Chi Zhang, Ari B. Hayes, Longfei Qiu, Yuwei Jin, Yanhao Chen, and Eddy Z. Zhang. 2021. Time-Optimal Qubit Mapping *(ASPLOS '21)*. Association for Computing Machinery, New York, NY, USA, 360–374. https://doi.org/10.1145/3445814.3446706
[35] X. Zhou, S. Li, and Y. Feng. 2020. Quantum Circuit Transformation Based on Simulated Annealing and Heuristic Search. *tcad* 39, 12 (2020), 4683–4694. https://doi.org/10.1109/TCAD.2020.2969647
[36] X. Zhou, S. Li, and Y. Feng. 2020. Quantum circuit transformation based on simulated annealing and heuristic search. *IEEE Trans. Comput.-Aided Des. Integr. Syst.* 39, 12 (2020), 4683–4694.
[37] A. Zulehner, A. Paler, and R. Wille. 2019. An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures. *IEEE Trans. Comput.-Aided Des. Integr. Syst.* 38, 7 (2019), 1226–1236. http://iic.jku.at/eda/research/ibm_qx_mapping/
[38] A. Zulehner and R. Wille. 2019. Compiling SU (4) quantum circuits to IBM QX architectures. In *Asia and South Pacific Design Automation Conf.* 185–190.