

Geometric Refactoring of Quantum and Reversible Circuits Using Graph Algorithms

Martin LUKAC^{†a)}, Member, Saadat NURSULTAN^{††}, Georgiy KRYLOV^{†††}, Oliver KESZOCZE^{††††}, Abilmansur RAKHMETTULAYEV^{†††††}, Nonmembers, and Michitaka KAMEYAMA^{††††††}, Member

SUMMARY With the advent of gated quantum computers and the regular structures for qubit layout, methods for placement, routing, noise estimation, and logic to hardware mapping become imminently required. In this paper, we propose a method for quantum circuit layout that is intended to solve such problems when mapping a quantum circuit to a gated quantum computer. The proposed methodology starts by building a Circuit Interaction Graph (CIG) that represents the ideal hardware layout minimizing the distance and path length between the individual qubits. The CIG is also used to introduce a qubit noise model. Once constructed, the CIG is iteratively reduced to a given architecture (qubit coupling model) specifying the neighborhood, qubits, priority, and qubits noise. The introduced constraints allow us to additionally reduce the graph according to preferred weights of desired properties. We propose two different methods of reducing the CIG: iterative reduction or the iterative isomorphism search algorithm. The proposed method is verified and tested on a set of standard benchmarks with results showing improvement on certain functions while in average improving the cost of the implementation over the current state of the art methods.

key words: quantum circuits, qubit layout, graph algorithms

1. Introduction

The design of quantum circuits generally involves the selection of a set of quantum gates such as the $T/T^\dagger/CNOT$, CZ/CH , or Clifford- T gate sets and a particular set of constraints related to the synthesis process. These constraints can be related to the technology such as linear nearest neighbor constraints, 2D symmetric arrays of trapped ions or fault-tolerance in general.

The latest available technology showing most promises for the quantum computer is either the quantum gated computer or the fully connected ion-trap [1]. For gated quantum computers, the design of quantum circuits is focused on designing circuits under constraints with respect to planar interaction between qubits and the regular layout of qubits in the quantum processor [2]. Recently, the 2D arrays of Rydberg neutral atoms have also been popularized in [3] and require similar type of 2D nearest neighbor interactions. While new

technologies are still in development and are being experimented with, all of them are expected to have constraints strongly impacting the circuit design and optimization tools.

As a result, the design of quantum circuits and algorithms has been quite active since late nineties but the actual physical design and placement of quantum circuits is related to specific constraints of various quantum technologies. For instance, the design of NMR quantum computer [4] required a specific mapping technique from the representation to a chloroform molecule. In the one-way quantum computer [5], the target algorithm has to be translated to a series of measurement operations giving the overall realization a very specific spatial pattern. In addition, from the general point of view, the original logical universality was demonstrated in [6] using the set of quantum gates designated as $CNOT/CV$, while currently this gate set is not used due to its high cost of implementation. Currently in the logic design of quantum circuit the so called Clifford – T quantum gate set is used.

While the issue of technology mapping has been envisaged for various technologies, the mapping of logical and quantum circuits onto a quantum computer requires additional technological constraints. The first of such requirements is the limit on the size of a quantum gate that can be directly implemented in quantum computer [4], [6]. This means that every logical or quantum gate requiring more than two qubits must be broken into two qubit gates. Another constraint is the requirements of constructing quantum gates on physically adjacent qubits [4].

These constraints on placement and realization of the quantum physical architecture (called coupling map) have been used in various technologies. For instance, in [7], the layout of the one-way quantum computer [5] yields a NP-complete problem. The problem of creating nearest neighbor compliant circuits has already been considered in one or two dimensions (see, e.g., [8] or [9]). Recent studies [10]–[15] explored the application of the two-dimensional layout of quantum logical qubits to various existing realizations [16]–[18] of gated quantum computers.

The existing approaches have been exploiting various properties of logic, gate and implementation based local and global properties. In [10], the authors present a work where various quantum and reversible logic circuits are mapped specifically to individual architecture of the IBM Q quantum computers. The target of these studies is to demonstrate the maximum exploitation of the nearest neighbor requirements. In [9] the authors used a look-ahead method for

Manuscript received November 11, 2023.

Manuscript revised March 8, 2024.

Manuscript publicized June 24, 2024.

[†]Hiroshima City University, Hiroshima-shi, 731–3166 Japan.

^{††}Nazrbayev University, Astana, Kazakhstan.

^{†††}University of New Brunswick, Fredricton, Canada.

^{††††}Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Nürnberg, Germany.

^{†††††}Nazarbayev Intellectual School, Astana, Kazakhstan.

^{††††††}Tohoku University, Sendai-shi, 980–8577 Japan.

a) E-mail: malu@hiroshima-cu.ac.jp

DOI: 10.1587/transinf.2023L0P0011

placing qubits in a 1D and 2D quantum structure by estimating short term impact of qubit placement. In [8], the authors implemented a sifting method allowing to map a quantum circuit independently of hardware geometrical constraints to a Linear Nearest Neighbor (LNN) model by calculating the optimal cost of individual placement for each two consecutive quantum gates. A more specific constraints of a gated quantum computer was proposed in [11], [12]. The authors evaluated a method for reducing the number of inserted swap gates by replacing them with a less costly variant by inverting the CNOT gates required by some of the IBM Q architectures. In [19], the authors use a pre-processing of the quantum circuit that groups gates and places them using a direct, all-qubits-at-once mapping general layout heuristics. In [20], the authors propose a more complex method including qubits ordering combined with optimization of permutation between gates. The default algorithm of placement and layout of quantum circuits on the IBM Q computers uses a local graph arrangement algorithm that first generates sub-graphs linked by single edge and then places them optimally. In [15] the authors took a different approach where starting from an ideal circuit an iterative approach was implemented finding an optimal mapping to any given qubit coupling. These approaches are in general efficient but require the reformulation and application of local rules, which do not guarantee global minimization.

In parallel to these specific computer-architecture approaches there is a very large amount of work in general quantum circuit design and optimization. In [21] a method for optimization of quantum circuits with continuous parameters is presented, In [22] the authors minimize quantum circuits by a set of rules however they do not consider specific hardware constraints and in [23] a machine learning optimization of quantum circuits is proposed but the evaluation is limited to few circuits design. More general methodologies have been proposed using a variety of methods such as evolutionary approaches [24]–[26], logic minimization [8], [12], [27], [28] or heuristics [29]–[31].

To solve the problem of hardware constrained quantum circuit design, and unlike previous approaches that target specifics of hardware we take a different approach. Instead of designing a quantum circuit for a particular quantum chip and coupling map, we start by designing an ideal virtual hardware architecture that is iteratively refined to match the target real physical model. Given a quantum circuit on n qubits, we start by generating the most accommodating quantum physical model (allowing arbitrary qubit interaction) Υ , spanning the n dimensional space by representing the interactions between every pair of qubits. Once Υ is constructed, we apply iterative reductions allowing the mapping of Υ onto an existing qubit coupling model C of a quantum computer. This coupling is optimized by mapping the circuit onto the coupling map by minimizing the Dijkstra shortest distance on the qubit coupling. The reduction is driven by the minimization of the cost of the mapping, i.e. reducing the number of interactions between qubits. Alternatively, the model allows taking into account the noise model Ω (such as

correlated noise model [32] or [33]) of the qubits within the quantum chip. Combining the interaction model Υ and the noise model Ω permits the application of the multi-objective optimization and model-independent circuit placement optimization. Other work on exploring similar ideas exists (see, e.g. [10]), however, this research is uniquely characterized by the reduction rule set for the connectivity graph.

This paper is organized as follows: Sect. 2 introduces the necessary background to our method. Section 3 describes the proposed approach and Sect. 4 introduces the method of minimizing the proposed model. Section 5 describes the experiments performed and Sect. 6 concludes the paper.

2. Method Background

Let C be a quantum circuit operating with p gates on l qubits. Let the circuit C be a sequence $S_C = \langle s_1, \dots, s_p \rangle$ of quantum gates s_j selected from a gate set \mathbb{G} . The gate set Clifford- T , for example, is given by $\mathbb{G} = \{CNOT, T, T^\dagger, H\}$.

Also let, two qubit quantum gates such as $CNOT$, CV/CV^\dagger , CZ , I_{ZZ} , CH be referred to as Generalized Interaction (GI). Figure 1 shows three different circuit realizations of the Toffoli function. Each circuit uses a different quantum gate set and has a different number of GI gates.

Definition 1 (Circuit Skeleton (CS)). *A quantum circuit's skeleton is the structure that remains after removing all single qubit gates and any logic on the interaction gates.*

For instance the circuit skeleton of the circuit shown in Fig. 1 (c) is shown in Fig. 2 (a).

Definition 2 (Circuit Interaction Graph (CIG)). *The circuit interaction graph is a representation of a quantum circuit C*

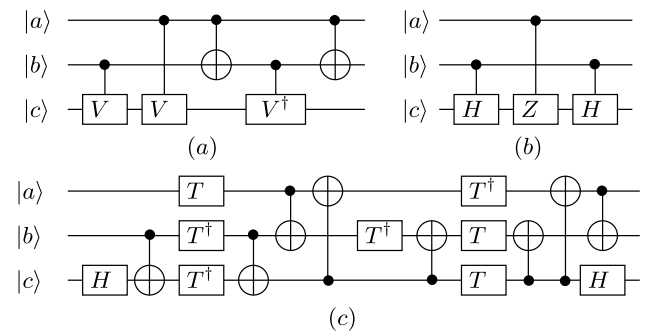


Fig. 1 Three examples of Toffoli function realizations using different gate sets: a) $CV/CV^\dagger/CNOT$, b) CH/CZ and c) Clifford- T gate sets.

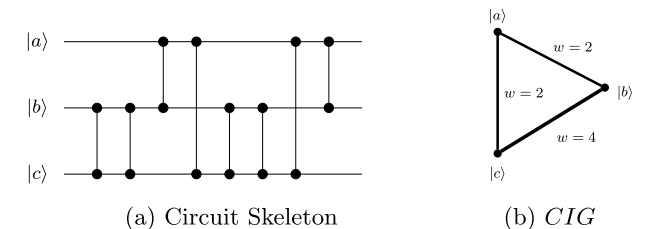


Fig. 2 Circuit Skeleton (a) and CIG (b) for circuit in Fig. 1 (c).

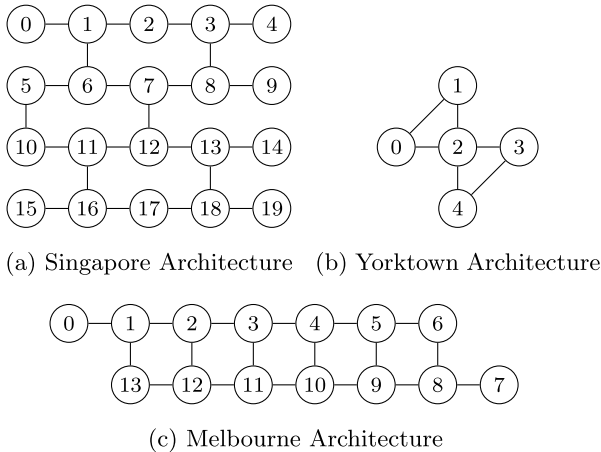


Fig. 3 Example of some of the available architectures and their qubit couplings available in the IBM Q initiative [2].

as a weighted graph $\Upsilon = (\mathbb{V}, \mathbb{E})$ such that:

- Each qubit i in the circuit C is represented by a vertex $v_i \in \mathbb{V}$
- There exists an edge e_{ij} between vertices v_i and v_j if there is a gate operating on qubits $|i\rangle$ and $|j\rangle$.
- Each edge e_{ij} is associated with a weight $w_{ij} = W(e_{ij})$. $W(e_{ij})$ is equal to the number of GI gates operating on qubits $|i\rangle$ and $|j\rangle$.

The corresponding CIG to the circuit from Fig. 1 (c) is shown in Fig. 2 (b).

Let a coupling map be a specification of an gated quantum computer physical interactions. The coupling map \mathbb{M} is a set $\{m_{0,1}, \dots, m_{k,l}\}$ of edges representing the physical connections between a set of physical qubits \mathbb{Q} . An element $m_{i,j}$ represents the existence of an physical connection between two physical qubits $|i\rangle$ and $|j\rangle$. We will denote a coupling map (un-directed graph) $QC = \{\mathbb{Q}, \mathbb{M}\}$ with $\mathbb{Q} = \{q_0, \dots, q_l\}$ being the set of physical qubits in the coupling. Examples of such architectures can be seen in Fig. 3. The coupling map indicates which qubits can be used to directly implement two qubit quantum operation. For instance qubits 0 and 1 in Fig. 3 (a) can interact directly but if qubits 0 and 2 must interact then their logical qubits must be moved to immediate proximity. This implies that for two qubits to interact they must be direct neighbors and be also connected by an edge in \mathbb{M} .

3. Circuit Interaction Graph Manipulation

To assist with representing the information about the graph, a notion of path is introduced. In this context, a path is a sequence of edges, connecting two vertices i and j in the graph.

Definition 3 (Path in CIG). A path ρ in a CIG and defined over (\mathbb{V}, \mathbb{E}) is an ordered set $\rho = \{e_{i,j} \leq \dots \leq e_{k,l}\}$ of edges, such that each edge $e_{i,j} \in \mathbb{E}$ is encountered only once, i.e. a simple path

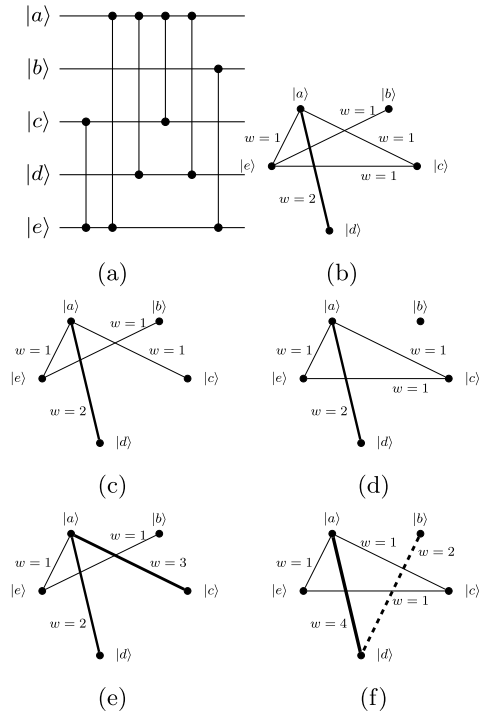


Fig. 4 (a) skeleton of quantum circuit, (b) corresponding CIG, (c) CIG after consistent ER $\mathcal{R}(e_{ce})$ and (d) CIG after inconsistent ER $\mathcal{R}(e_{be})$, (e) reconnecting $|c\rangle$ and $|e\rangle$ through $|a\rangle$ and (f) reconnecting $|b\rangle$ and $|e\rangle$ through $|d\rangle$ and $|a\rangle$

We will denote a path linking a start vertex v_{start} and an end vertex v_{end} by $\rho_{start,end}$. Evolving the definition further, the notion of a path ρ can be simply extended to the so called weighted path; a path that is associated with a single weight for each edge.

Definition 4 (Weighted Path). A weighted path γ in a CIG defined over (\mathbb{V}, \mathbb{E}) between two vertices i and l , is an ordered set $\gamma = \{\mathcal{W}(e_{i,j}) \leq \dots \leq \mathcal{W}(e_{k,l})\} = \{\gamma_0, \dots, \gamma_k\}$ with $\mathcal{W}(\cdot)$ being the weight function that returns the weight of any edge in the CIG.

As an example, the skeleton circuit from Fig. 4(a) is shown in the CIG in Fig. 4(b). Observe that the edges e_{ac} , e_{ae} , e_{be} and e_{ce} have weight $w = 1$ while the edge e_{ad} has a weight $w = 2$. The weights are obtained directly from the skeleton circuit in Fig. 4 (a).

Note that the weight based ordering removes information about the order of visitation of vertices by the quantum gates. However the purpose of this ordering is to provide a priority to highly visited paths when allocating the physical qubits.

Definition 5 (Cost of Path). A cost of path $cost(\rho)$ is defined as

$$cost(\rho) = \sum_{e \in \rho} \mathcal{W}(e) \tag{1}$$

To manipulate the CIG, a set of robust operations must be defined. The first operation that is required for minimizing

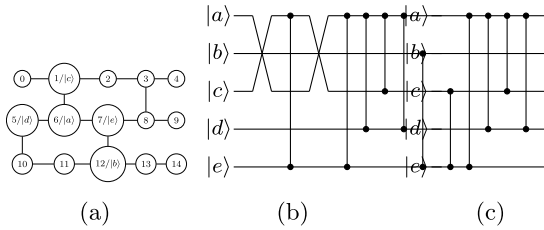


Fig. 5 (a) The logical to physical qubits mapping, (b) reconnecting $|c\rangle$ with $|e\rangle$ by exchanging the position of logical qubits $|a\rangle$ and $|c\rangle$, (c) an inconsistent ER resulting in $|b\rangle$ being an unconnected qubit.

the CIG is the removal of an individual edge.

Definition 6 (Edge Removal (ER)). *In a weighted graph $\Upsilon = (\mathbb{V}, \mathbb{E})$, ER is an operation $\mathcal{R}(e_{jk}) = \Upsilon(\mathbb{V}, \mathbb{E} \setminus \{e_{jk}\})$.*

The ER operation can create two different kind of graphs. Let $\Upsilon = (\mathbb{V}, \mathbb{E})$ and $\mathcal{R}(e_{jk})$: if applying ER the resulting graph is not connected any more, we call the ER *inconsistent*, if the resulting graph is still connected we refer to it as *consistent* ER.

For instance for the CIG in Fig. 4 (b) constructed from the skeleton in Fig. 4 (a), the result of a consistent and inconsistent ER are shown in Fig. 4 (c) and Fig. 4 (d) respectively.

The result of these two ER operations can be analyzed further. Assume that the logical qubits from the skeleton circuit in Fig. 4 (a) are mapped to the physical qubits of the coupling model from Fig. 3 (a) as follows: $|a\rangle \rightarrow 6$, $|b\rangle \rightarrow 12$, $|c\rangle \rightarrow 1$, $|d\rangle \rightarrow 5$, $|e\rangle \rightarrow 7$. As can be observed there is no available direct physical coupling between the $|c\rangle$ and $|e\rangle$ logical qubits located on physical qubits 1 and 7 respectively. The lack of physical coupling is represented in this case as a consistent ER operation resulting in the CIG from Fig. 4 (c). The result of the consistent ER is the disconnection of the qubit $|c\rangle$ and $|e\rangle$. It requires the re-routing of the logical qubit $|c\rangle$ and $|e\rangle$ to near proximity so that the GI gate can be applied. This can also be represented in terms of paths. In the original CIG the path $\rho_{ce} = \{e_{ce}\}$ and the \mathcal{R}_{ce} resulted in $\rho_{ce} = \{\}$, i.e. empty path. One possible solution is shown in Fig. 5 (a). In this case the logical qubits $|a\rangle$ and $|e\rangle$ were swapped using a SWAP gate, then the GI gate can be directly applied because the physical qubit $|a\rangle$ and $|c\rangle$ are directly linked. Finally, their respective location was restored using again a SWAP gate. As a result of these operations a new path $\rho'_{ce} = \{e_{ac}, e_{ce}\}$ was created.

Theorem 1. *In order to ensure the functionality of the original circuit, a consistent ER operation $\mathcal{R}(e_{ij})$ from a path ρ_{hk} , requires the insertion of a maximum of $2|\rho'_{hk}| - 1$ SWAP gates.*

A consistent ER operation therefore implies, that for all weights in the new path $\rho_{il} = \{e_{ij} \leq \dots \leq e_{kl}\}$, we have that each edge weight is increased by 2, i.e. for all $e_{ij} \in \rho_{il}$ we have $\mathcal{W}(e_{ij})^{new} = \mathcal{W}(e_{ij}) + 2$. For instance, if the edge e_{ce} in CIG from Fig. 4 (b) is removed and resulting in Fig. 4 (c), then two swap gates have to be inserted on the edge e_{ac} in order for $|e\rangle$ and $|c\rangle$ interact directly (Fig. 4 (e)). Note that one SWAP is to make the interaction possible and the other

to restore the original qubit placement. The corresponding circuit is depicted in Fig. 5 (b). Observe that in Fig. 5 (b) first the two logical qubits $|a\rangle$ and $|c\rangle$ have been swapped. This represents the swapping of the logical values between the physical qubits. Then the interaction between $|a\rangle$ and $|e\rangle$ is applied: despite that the interaction looks crossing more qubits it actually represents an existing physical connection between the physical qubit on top and on the bottom of the quantum circuit. Once the interaction is performed the logical qubits $|a\rangle$ and $|c\rangle$ are swapped back.

Figure 5 (c) illustrates the result of the inconsistent ER. The qubits $|b\rangle$ and $|e\rangle$ are disconnected and cannot be reconnected anymore because according to Fig. 4 (d) there are no more connections between logical qubit $|b\rangle$ (mapped to 12, Fig. 5 (a)) and other qubits of the circuit. To reconnect the logical qubits $|b\rangle$ and $|e\rangle$ either the logical qubit $|b\rangle$ should be placed on a different physical qubit or from the current physical qubit a new path to the logical qubit $|e\rangle$ should be found.

Let the qubit $|b\rangle$ be placed on physical qubit 2 instead of 12. The possible paths to connect $|b\rangle$ to $|e\rangle$ can be either by directly swapping the logical qubits $|b\rangle$ with $|c\rangle$ and then with $|a\rangle$ or to swap the logical qubit through a set of unused physical qubits 3 and 8. The second solution requires to move the logical qubit $|b\rangle$ from physical qubit 2 to 8 (by using four SWAP gates) and then interact with logical qubit $|e\rangle$.

Theorem 2. *An inconsistent ER operation $\mathcal{R}(e_{ij})$, requires the insertion of a maximum of $2(|\rho_{ik}| - 1) + 2$ SWAP gates into a path ρ_{ik} .*

In order to normalize the outcome of the edge removal operations, we impose the condition that any ER must preserve the logical structure of the circuit.

Definition 7 (Canonical Edge Removal (CER)). *A Canonical Edge Removal is an operation on a CIG that contains the following steps:*

1. a consistent or inconsistent edge removal
2. the insertion of appropriate number of SWAP gates to preserve the logic circuit structure

4. Υ Reduction

When a CIG is created, it represents the ideal or completely expanded geometrical model of a quantum circuit. Each qubit is connected in CIG to another qubit if an interaction or CNOT gate is applied to both of them. Once the coupling model is provided, the CIG of the quantum circuit is mapped to physical qubits and physical couplings. However, at this stage the CIG is oblivious to real hardware requirements (i.e., the coupling map) as with more qubits more edges will be generated, meanwhile a particular architecture will allow for a limited and constant connectivity between qubits. Therefore the mapping from the CIG to the quantum coupling is referred here to as Υ Reduction.

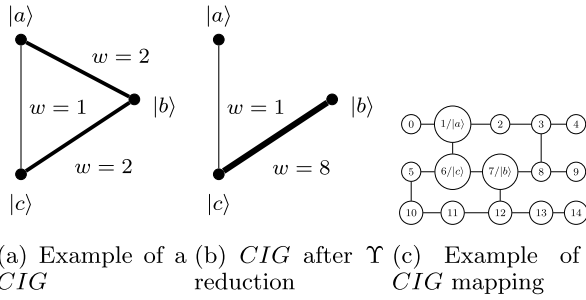


Fig. 6 Example of (a) CIG , (b) its Υ reduced form, (c) its mapping to the part of the Singapore architecture from Fig. 3 (a)

Definition 8 (Υ Reduction). *Let there be*

- $\forall v_i \in \mathbb{V}$ a unique mapping $\Sigma : v_i \rightarrow q_i$ such that $q_i \in \mathbb{Q}$ and
- $\forall e_{ij} \in \mathbb{E}$ a unique mapping $\Delta : e_{ij} \rightarrow m_{ij}$ such that $m_{ij} \in \mathbb{M}$

then, given a $CIG = (\mathbb{V}, \mathbb{E})$, a $QA = (\mathbb{M}, \mathbb{Q})$ and mappings $\Sigma : \mathbb{V} \rightarrow \mathbb{Q}$ and $\Delta : \mathbb{E} \rightarrow \mathbb{M}$, the Υ reduced CIG^Υ is implicitly defined by being isomorphic to $QA_{CIG} = (\Sigma(\mathbb{V}), \Delta(\mathbb{E}))$.

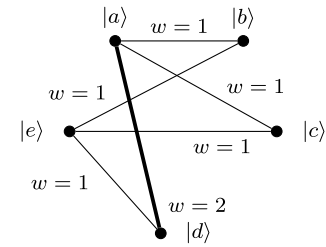
As an example the CIG from Fig. 2 (b) is Υ reduced to QA in Fig. 3 (b) but is not Υ reduced to QA in Fig. 3 (a). The CIG is isomorphic to the QA from Fig. 3 (b) for the mapping $|a\rangle \rightarrow 0$, $|b\rangle \rightarrow 1$ and $|c\rangle \rightarrow 2$ for instance. However as can be seen, this CIG cannot be mapped directly to Fig. 3 (a).

The Υ reduction of the CIG from Fig. 6 (a) to the CIG in Fig. 6 (b) results in the CIG from Fig. 6 (b) have the logical qubits directly to physical qubits of the QA in Fig. 6 (c): the CIG from Fig. 6 (b) is isomorphic to the QA in Fig. 6 (c) using the following mapping $|a\rangle \rightarrow 1$, $|b\rangle \rightarrow 7$ and $|c\rangle \rightarrow 6$. The resulting CIG is shown in Fig. 6 (b).

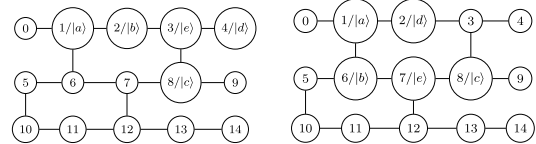
The Υ reductions represent a heuristic method for mapping a CIG to a QA by iterative application of consistent or inconsistent ER and recalculating the number of required of SWAP gates. However, in order not to search all possibilities to match CIG to a QA on all possible mappings Σ , several optimizations were implemented. The first step in all of the qubit mapping approaches start by first identifying qubits in QA that match the logical qubits that have edges with highest weight in the CIG . Then for each remaining logical qubit in the CIG a search for the closest match among the direct neighbors of all previously mapped physical qubits is performed. Another alternative in mapping the qubits is to map the CIG qubits in order of decreasing weight of edges connecting qubits. Note that such a mapping is an implicit Υ reduction: mapping qubits one after another one forces the removal of certain edges between qubits and thus requires the insertion of SWAP gates.

4.1 Qubit Σ Mapping

Once the CIG of a circuit is constructed, the algorithm searches for the best physical qubit in QA for each logical qubit in CIG . As already introduced, two orderings



(a) Example of a CIG



(b) Mapping of a CIG using s_w

Fig. 7 Example of (a) CIG , (b) its mapping using the s_w sequence and (c) its mapping using s_n sequence on a part of the Singapore architecture from Fig. 3

by which the qubits are mapped were explored: branching factor (direct edges) or direct neighbors. Searching for the next best candidate physical qubit for the next logical qubit will have different results based on the ordering. For instance let's consider the CIG from Fig. 7 (a) and the QA from Fig. 3 (a). The edge based ordering and the nearest neighbor ordering gives $s_w = \{|a\rangle, |e\rangle, |b\rangle, |c\rangle, |d\rangle\}$ and $s_n = \{|a\rangle, |b\rangle, |e\rangle, |c\rangle, |d\rangle\}$ sequence respectively. The first qubit is mapped as follows: $|a\rangle \rightarrow 1$.

Using the sequence of qubits s_w one possible mapping is $|e\rangle \rightarrow 3$, $|b\rangle \rightarrow 2$, $|c\rangle \rightarrow 8$ and $|d\rangle \rightarrow 4$ with a total cost of SWAP gates being 8. This is because the edges $v_{|b\rangle, |e\rangle}$, $v_{|c\rangle, |e\rangle}$ and $v_{|a\rangle, |b\rangle}$ are directly mapped to direct connections between physical qubits of QA . However the edges $v_{|a\rangle, |c\rangle}$ and $v_{|a\rangle, |d\rangle}$ require at least four SWAP gates each. Thus the total overhead is at least 8 (Fig. 7 (b)).

Using the sequence s_n a possible mapping is $|b\rangle \rightarrow 6$, $|e\rangle \rightarrow 7$, $|c\rangle \rightarrow 8$ and $|d\rangle \rightarrow 1$. The cost of required SWAP gates comes from the unsatisfied direct edges $v_{|a\rangle, |c\rangle}$ and $v_{|e\rangle, |d\rangle}$. Each edge requires four SWAP gates resulting in the total of 8 or more (Fig. 7 (c)).

In order to determine the best cost for placing individual qubits, we prepare a complete map of shortest distances between any two physical qubits in QA using Dijkstra algorithm. The placement of each qubit during the mapping is navigated so as to select the shortest distance between all currently placed qubits.

4.2 Weighted Mapping

The mapping so far is based simply on the distance between the individual qubits. However, to provide a more accurate estimation of SWAP gates required, the weights of the CIG are considered. Every weight $\mathcal{W}(v_{ij})$ represents the number of times two qubits have to interact in the quantum circuit. This implies that every time two qubits should interact, they

should be in adjacent position in QA . Therefore during the qubits mapping, the weights will be indicating which edges are preferred to be directly mapped to adjacent qubits.

The CIG from Fig. 7 (a) has five edges of cost 1 and one edge of cost 2. Taking into account the weights of individual edges in the CIG , the number of SWAP gates required for the mapping from Fig. 7 (b) now becomes 12 while the number of required gates for the mapping in Fig. 7 (c) remains the same at 8.

The inclusion of weights into the Υ reduction is represented by giving a priority to direct mapping of edges with high weight. Therefore independently of the sequence of placement of the qubits, the algorithm maps qubits with high weight edges first and in direct adjacency.

4.3 Additional Mapping Constraints

The constraints imposed so far on the CIG here are only partial. In addition to the connectivity, the placement and the cost (number of quantum gates), a noise model of each qubit as well as possible edge orientation requirements must be integrated for some of the specific requirements of the gated quantum computers.

Because the Υ reduction is a constraint satisfaction based algorithm, adding new constraints does not change the processing flow. The default solution is the result of fulfilling the constraint on qubit connections. Therefore, the Υ reduction and the mapping of logical onto physical qubits can easily be modified to take care of additional constraints while searching for the solution. This reformulation is taken care of by first merging all the constraints into a single expression and then by formulating a minimization function over the set of all constraints. Let C_{cost} , $C_{placement}$, C_{noise} and C_{edges} be the respective constraints describing the cost, the placement, the noise and the edges of the CIG , respectively.

- C_{cost} is a scalar constraint equal to the costs of a path as defined in Eq (1).
- The $C_{placement} \subseteq V$ is the set of vertices of the CIG that violate the placement constraints (the number of edges vs. the number of physical neighbors).
- C_{noise} is a constraint that can take several forms depending on the used noise model. It represents the circuit reliability obtained by simulating the circuitry on a set of allocated physical qubits on a quantum computer. It can also represent the additive noise per gate or per qubit added.
- C_{edges} is specific constraint indicating – if necessary – the properties of the physical connections between qubits. Example of such constraint is the direction of application of the control signal in a two qubit quantum gate.

First, let θ represent parameters of the CIG model. That is, θ is the set of physical qubits selected for the placement of the CIG with an associated layout scheme. Then, using the four constraints a minimization criterion can be formulated and is shown in Eq (2).

Algorithm 1 Quantum Layout Algorithm based on Graph Isomorphism

```

1:  $QA \leftarrow Quantum\ Coupling\ Map$ 
2:  $dDj \leftarrow Dijkstra(QA)$ 
3:  $Qubits \leftarrow \{\emptyset\}$ 
4:  $CIG^0 \leftarrow constructCIG(qCircuit)$ 
5:  $\mathbb{L} \leftarrow orderCIGQubits(CIG^0)$ 
6: while  $1 < |\mathbb{L}|$  do
7:   if  $Map(CIG^t, QA)$  then
8:     break
9:   else
10:     $t \leftarrow t + 1$ 
11:     $Qubits \leftarrow Qubits \cap \mathbb{L}_{-1}$ 
12:     $CIG^t \leftarrow CIG^{t-1} \setminus \mathbb{L}_{-1}$ 
13:     $\mathbb{L} \leftarrow \mathbb{L} \setminus \mathbb{L}_{-1}$ 
14:   end if
15: end while
16: while  $|Qubits| > 0$  do
17:    $CIG^{t+1} \leftarrow consistentER(CIG^t, Qubits_{-1})$ 
18:    $Qubits \leftarrow Qubits \setminus Qubits_{-1}$ 
19:    $t \leftarrow t + 1$ 
20: end while
21:  $\gamma \leftarrow orderCIGWeights(CIG^t)$ 
22:  $Cost \leftarrow \sum_{i=1}^j \gamma_i$ 

```

$$C(CIG, QA) = \min_{\theta} (C_{cost}, C_{placement}, C_{noise}, C_{edges}) \quad (2)$$

Observe that Eq (2) is a specification that can be applied to standard SAT solvers or any other CSP problem solver. However, due to the physical nature of the CIG to QA mapping several constraints can be relaxed by providing knowledge about QA ahead of time. For instance, one can order the qubits according to daily tomography calibration (such as provided by Rigetti [18]) in order to minimize the space search with respect to the least noisy location.

Another minimization can be provided by using the knowledge that the circuit scaling is noise-monotonous; adding more qubits always increases the noise of the whole system. Knowing the set of least noisy qubits, the relative distance between them, and increase in noise of the system caused by extra qubits, one can argue about optimal placement and number of qubits required for realization of a given circuit on the target machine.

4.4 Algorithm Implementation

Our software reads the gates from a file in RevLib format [34] to a format provided by Qiskit API. Then, it retrieves a coupling map, that is physical qubit connectivity, from available quantum computers. We define a default layout to map qubits in alphabetical order - i.e. logical qubit A is mapped to physical qubit a, B gets mapped to b, etc. The actual algorithm is shown in the pseudo code 1. It starts by finding a coupling map, a graph of the connectivity of logical qubits QA (line 1). Then it pre-computes distances between all physical qubits using Dijkstra algorithm (line 2) and then it builds a CIG of the circuit (line 4). Next it creates a ordered set \mathbb{L} of all logical qubits ordered by their branching factors (line 5). The algorithm then proceeds to iteratively match the current

Table 1 Results of the CIG based quantum layout method

Function	I	O	T	<i>CIGCost</i>	<i>CIGCost</i> ₁	<i>CIGCost</i> ₂	<i>CIGCost</i> ₃	<i>qCost</i>	<i>qCost</i> ₁	<i>qCost</i> ₂	<i>qCost</i> ₃	% Impr.
3.17	3	3	3	47	27	27	27	47	27	27	27	0
graycode6_47	6	6	6	17	15	15	21	17	15	15	21	0
decod24-v1_42	5	5	5	44	40	38	38	44	40	38	38	0
toffoli_double_3	4	4	4	44	18	18	20	44	18	18	20	0
4mod5-v1_24	4	1	5	58	262	61	80	185	180	178	165	64
4mod7-v0_96	4	3	7	352	589	406	427	599	552	552	543	35
4gt11_82	4	1	5	107	175	85	77	158	147	145	134	42
4gt12-v0_86	4	1	5	650	1150	658	738	1175	1114	1112	1090	40
4gt13_90	4	1	5	212	424	230	268	448	420	418	424	49
4gt4-v0_80	4	1	5	361	503	408	456	521	483	483	535	25
4gt5_77	4	1	5	273	407	273	324	400	369	369	403	26
alu-bdd_288	7	7	7	223	672	345	259	487	469	467	416	46
alu-v4_36	5	5	5	231	445	215	308	444	416	416	426	48
alu-v3_34	5	5	5	92	199	103	123	201	192	190	186	50
alu-v2_33	5	5	5	109	178	109	75	132	125	125	121	38
alu-v1_28	5	5	5	67	166	87	87	138	133	131	119	26
decod24	2	4	4	45	42	40	40	45	42	40	40	0
ex1	5	1	5	20	20	20	20	20	20	20	20	0
ham3_28	3	3	3	30	20	20	20	30	20	20	20	0
hwb4_49	4	4	4	319	177	175	180	319	177	175	180	0
hwb_52	4	4	4	117	85	85	95	117	85	85	94	0
hwb6	6	6	6	940	751	785	801	888	590	590	580	-29
sym6_63	6	1	6	689	564	519	446	689	519	519	446	0

CIG to any isomorphic sub-graph of the QA graph (line 7). If no such isomorphism is found the qubit with the smallest branching degree is removed from the CIG and from the \mathbb{L} set and is added to the $Qubits$ container. This process is iterated until the CIG is mapped to the QA (line 6 to 14). When this process is finished, then the qubits accumulated in $Qubits$ are mapped back to the QA using the *consistentER* operation one at a time (line 16 to 20). Finally the cost of the resulting mapping is calculated.

5. Experiments

To verify the proposed method, we use the Qiskit SDK [35]. The SDK provides an access to physical quantum computers through API called IBM Q Experience. For data, we use the RevLib [34] for both only reversible or truly quantum functions.

5.1 Assumptions

At this stage of experimentation, the following generalizations have been adopted:

- We minimize the whole circuit over the set of physical qubits without any edge direction: the direction of the coupling map is not taken into account
- The insertion of SWAP gates is bi-directional without any local optimization. This means that for a path ρ_{ik} between two vertices v_i and v_k in CIG, the number of inserted SWAP gates is equal to $2 \cdot cost(\rho_{ik})$.
- Where possible, the gates provided by Qiskit API were used, otherwise we provide a macro implementation using decomposition to available gates. this mainly concerns the decomposition of C^nMOT gates with $n >$

2: we use the decomposition introduced in [6].

Most of our experiments were conducted for the IBM Q QPU Melbourne 16 that has a qubit coupling map as shown in Fig. 3 (c). This machine provides access to 14 physical qubits.

5.2 Results

Table 1 shows the results of our applied graph-based synthesis method for the IBM Melbourne Quantum architecture (14 qubits).

The first four columns show the function name, the number of inputs, outputs and total number of qubits respectively. Columns five to eight show the result of CIG based algorithm and columns nine to twelve show the results using the IBM provided synthesis algorithm. The last column shows relative improvement of CIG vs IBM algorithm. The best result in each row is highlighted in bold.

The CIG-based approach was evaluated using the IBM circuit compiler using various levels of optimization. In Qiskit three levels of optimization are available from the weakest to the strongest. The results of not optimized design are entitled $qCost$ and the three levels of optimization using IBM native support are entitled $qCost_i$. Similarly the CIG approach was applied directly or has been combined with IBM optimization of all three levels.

On average the CIG improvement over the best IBM result is 24.6% with the lowest improvement being -29% for the function *hwb6* and the highest improvement was 64% for the function *4mod5-v124*. In addition it can also be seen that out of 22 displayed benchmark functions, the CIG algorithm without using the IBM optimization performs the best in 8 cases.

Table 2 Comparative results with similar approaches.

Benchmark	n	[20]	CIG	Change
max46	10	79236	82673	+4.33%
rd73	10	673	541	-19.6%
sqn	10	30172	28982	-2.9%
sys6-v0	10	641	433	-32.4%
urf3	10	413778	420800	+1.7%
9symml	11	104360	98634	-5.5%
dc1	11	5661	3725	-34.2%
life	11	67764	69232	+2.2%
sym9	11	62602	57206	-8.6%
wim	11	2816	1564	-44.5%
z4	11	9162	7860	-14.21%
cm152a	12	3401	3320	-2.4%
cycle10	12	18131	21317	+17.6%
rd84	12	1016	827	-18.6%
sqrt8	12	9149	9092	-0.6%
sym10	12	192500	201176	+4.5%
adr4	13	10358	7680	-25.9%
dist	13	116175	105407	-9.3%
radd	13	9637	11324	+17.5%
rd53	13	953	948	-0.52%
squar5	13	5887	4735	-19.6%
cm42a	14	5276	4410	-16.4%
cm82a	14	3649	1395	-61.8%
pm1	14	4956	4410	-11%
sao2	14	117695	134624	+14.4%
dc2	15	28711	28322	-1.4%
ham15	15	26590	24463	-8%
misex1	15	14549	18703	+28.5%
rd84	15	1016	872	-18.6%
cnt3-5	16	1380	873	-36.7%

In addition to the above experiments we also compared our results to the results from [20] that also use CIG based approach. However unlike in the proposed method, the work [20] is not reporting SWAP gates counts and in addition optimizes the gates. In order to evaluate our approach we also applied a set of optimizing transformations. The implemented transformations are single gate concatenation, gate movement and re-ordering. Each operation is by the rules below. The implemented transformations are similar to the transformations in [20] and from [36]. These optimizations combine multiple single-qubit quantum gates into one single qubit gate according to simple rules:

- two single qubit quantum gates can be approximated by a single quantum gate resulting from the multiplication of respective matrices of each quantum gate.
- Up to three distinct R quantum gates rotations can be combined by creating a single qubit gate with parameters on rotation about individual axes from the R rotations
- CNOT gates defined on same logical qubits can be eliminated if there is no any quantum gate in between their control terminals

The results of these experiments are shown in Table 2. The first column is the name of the evaluated function, the second column shows the number of total qubits in the quantum circuit. The third column shows the count of quantum gates in the circuit as proposed by [20] and in the fourth col-

umn are the results obtained by our proposed method. The last column shows the percentage of change.

The presented results show that on the evaluated benchmarks the proposed method using the isomorphism algorithm in average outperforms the similar method from [20]. The iterative method performed better on various circuits with up to 61.8% of improvement in the count of the gates for cm82a and with the worse result for misex1 circuit where our approach performed worse by 20%.

The reason for the improvement observed are most probably due to the following algorithmic differences. First, our algorithm starts by constructing an ideal CIG that is then reduced to the smallest graph matching the largest sub-graph on the quantum computer architecture. Second, the iterative approach allows to perform an incremental improvements by searching for better qubits location. Finally, the proposed approach allows to optimally reconnect and place the logical qubits using a gradient based method allowing for optimal placement.

6. Conclusion

In this paper we presented a general approach to solve the placement of quantum circuit on a two-dimensional regular structure quantum computer. The proposed method was applied to only existing architectures but is directly extensible to n -dimensional regular structure. We showed a heuristic method for placement of arbitrary quantum circuit taking into account the noise model and presented a heuristic algorithm allowing to place a given circuit based on the required constraints. Additionally, we proposed a formulation that allows the application of local search algorithms such as genetic algorithm or iterative optimization.

Our proposed method improved in some cases the default cost on the IBM Q approach. This is possibly due to the fact that our layout mapping uses guided global optimization and for a subset of circuits, does better job search for the qubit placements, accompanying the randomized qubits placement algorithms used in Qiskit API compilers. Another fact worth mentioning is that the improvement was achieved on the circuit that can be directly mapped to an IBMQ16 computer (with our assumptions considered). Therefore the use of an algorithm for optimizing circuits that can be directly mapped to existing methods is one of the areas for research and extension of the proposed method.

In the future we plan to provide a full set of functions integrated into a library that can be integrated in others algorithm solving the quantum layout problem. Our implementation could alternatively find its place as one of optimization layers for quantum object compilation. The next steps in the final developments are as follows: include the direction of coupling sensitive synthesis, noise related optimization, SAT solver for the optimal mapping, local minimization. Once all these features are implemented a testing and comparative analysis with other state of the art algorithms is to be performed.

References

- [1] N.M. Linke, D. Maslov, M. Roetteler, S. Debnath, C. Figgatt, K.A. Landsman, K. Wright, and C. Monroe, “Experimental comparison of two quantum computing architectures,” *Proceedings of the National Academy of Sciences*, vol.114, no.13, pp.3305–3310, 2017.
- [2] IBM, “IBM makes quantum computing available on IBM cloud to accelerate innovation,” 2016.
- [3] K. Wintersperger, F. Dommert, T. Ehmer, A. Houshanov, J. Klepsch, W. Maurer, G. Reuber, T. Strohm, M. Yin, and S. Luber, “Neutral atom quantum computing hardware: performance and end-user perspective,” *EPL Quantum Technology*, vol.10, no.1, Aug. 2023.
- [4] M.A. Nielsen and I.L. Chuang, *Quantum Computation and Quantum Information*, Cambridge University Press, 2010.
- [5] R. Raussendorf and H.J. Briegel, “A one-way quantum computer,” *Phys. Rev. Lett.*, vol.86, no.22, pp.5188–5191, May 2001.
- [6] A. Barenco, C.H. Bennett, R. Cleve, D.P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J.A. Smolin, and H. Weinfurter, “Elementary gates for quantum computation,” *Phys. Rev. A*, vol.52, no.5, pp.3457–3467, Nov. 1995.
- [7] Y. Lin, B. Yu, M. Li, and D.Z. Pan, “Layout synthesis for topological quantum circuits with 1-D and 2-D architectures,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.37, no.8, pp.1574–1587, Aug. 2018.
- [8] M. Lukac, P. Kerntopf, and M. Kameyama, “An analytic sifting approach to optimization of lnn reversible circuits,” 2017 International Conference on Information and Digital Technologies (IDT), pp.240–245, July 2017.
- [9] R. Wille, O. Keszöcze, M. Walter, P. Rohrs, A. Chattopadhyay, and R. Drechsler, “Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits,” *Asia and South Pacific Design Automation Conference*, pp.292–297, 2016.
- [10] M. Thornton and K.N. Smith, “Mustang-q: A technology dependent quantum logic synthesis and compilation tool,” *Design Automation for Quantum Computers Workshop, IEEE International Conference on Computer Aided Design (ICCAD-QCEDA)*, Nov. 2017.
- [11] A. Matsuo, W. Hattori, and S. Yamashita, “Reducing the overhead of mapping quantum circuits to IBM Q system,” 2019 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1–5, 2019.
- [12] G.W. Dueck, A. Pathak, M.M. Rahman, A. Shukla, and A. Banerjee, “Optimization of circuits for IBM’s five-qubit quantum computers,” *CoRR*, vol.abs/1810.00129, 2018.
- [13] A. Zulehner and R. Wille, “Compiling SU(4) quantum circuits to IBM QX architectures,” *Proceedings of the 24th Asia and South Pacific Design Automation Conference, ASPDAC ’19*, New York, NY, USA, pp.185–190, ACM, 2019.
- [14] A. Paler, I. Polian, K. Nemoto, and S.J. Devitt, “A fully fault-tolerant representation of quantum circuits,” *Reversible Computation*, ed. J. Krivine and J.B. Stefani, Cham, pp.139–154, Springer International Publishing, 2015.
- [15] M. Lukac, S. Nursultan, G. Krylov, and O. Keszöcze, “Geometric refactoring of quantum and reversible circuits: Quantum layout,” 23rd Euromicro Conference on Digital System Design, DSD 2020, Kranj, Slovenia, pp.428–435, IEEE, Aug. 2020.
- [16] J. Koch, T.M. Yu, J. Gambetta, A.A. Houck, D.I. Schuster, J. Majer, A. Blais, M.H. Devoret, S.M. Girvin, and R.J. Schoelkopf, “Charge-insensitive qubit design derived from the cooper pair box,” *Phys. Rev. A*, vol.76, no.4, p.042319, Oct. 2007.
- [17] J.A. Schreier, A.A. Houck, J. Koch, D.I. Schuster, B.R. Johnson, J.M. Chow, J.M. Gambetta, J. Majer, L. Frunzio, M.H. Devoret, S.M. Girvin, and R.J. Schoelkopf, “Suppressing charge noise decoherence in superconducting charge qubits,” *Phys. Rev. B*, vol.77, no.18, p.180502, May 2008.
- [18] C. Rigetti, J.M. Gambetta, S. Poletto, B.L.T. Plourde, J.M. Chow, A.D. Córcoles, J.A. Smolin, S.T. Merkel, J.R. Rozen, G.A. Keefe, M.B. Rothwell, M.B. Ketchen, and M. Steffen, “Superconducting qubit in a waveguide cavity with a coherence time approaching 0.1 ms,” *Phys. Rev. B*, vol.86, no.10, p.100506, Sept. 2012.
- [19] 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.
- [20] A. Kole, S. Hillmich, K. Datta, R. Wille, and I. Sengupta, “Improved mapping of quantum circuits to IBM QX architectures,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol.39, no.10, pp.2375–2383, 2020.
- [21] Y. Nam, N.J. Ross, Y. Su, A.M. Childs, and D. Maslov, “Automated optimization of large quantum circuits with continuous parameters,” *npj Quantum Information*, vol.4, no.1, p.23, May 2018.
- [22] T. Itoko, R. Raymond, T. Imamichi, and A. Matsuo, “Optimization of quantum circuit mapping using gate transformation and commutation,” *Integration*, vol.70, pp.43–50, 2020.
- [23] A. Paler, L. Sasu, A.-C. Florea, and R. Andonie, “Machine learning optimization of quantum circuit layouts,” *ACM Transactions on Quantum Computing*, vol.4, no.2, pp.1–25, Feb. 2023.
- [24] C.P. Williams and A.G. Gray, “Automated design of quantum circuits,” *Quantum Computing and Quantum Communications*, ed. C.P. Williams, Berlin, Heidelberg, pp.113–125, Springer Berlin Heidelberg, 1999.
- [25] F.Z. Hadjam and C. Moraga, “Rimep2: Evolutionary design of reversible digital circuits,” *ACM J. Emerg. Technol. Comput. Syst.*, vol.11, no.3, pp.27:1–27:23, 2014.
- [26] G. Krylov and M. Lukac, “Quantum encoded quantum evolutionary algorithm for the design of quantum circuits,” *Proceedings of the 16th ACM International Conference on Computing Frontiers, CF ’19*, New York, NY, USA, pp.220–225, Association for Computing Machinery, 2019.
- [27] C. Moraga, “Hybrid GF(2) – boolean expressions for quantum computing circuits,” *International Workshop on Reversible Computation*, 2011.
- [28] F.S. Khan and M. Perkowski, “Synthesis of multi-qubit hybrid and d-valued quantum logic circuits by decomposition,” *Theoretical Computer Science*, vol.367, no.3, pp.336–346, 2006.
- [29] D.M. Miller, D. Maslov, and G.W. Dueck, “A transformation based algorithm for reversible logic synthesis,” *Proceedings of the 40th Annual Design Automation Conference, DAC ’03*, New York, NY, USA, pp.318–323, Association for Computing Machinery, 2003.
- [30] W. Hattori and S. Yamashita, “Mapping a quantum circuit to 2D nearest neighbor architecture by changing the gate order,” *IEICE Transactions on Information and Systems*, vol.E102-D, no.11, pp.2127–2134, 2019.
- [31] R. Wille, S. Hillmich, and L. Burgholzer, “Decision diagrams for quantum computing,” *Design Automation of Quantum Computers*, pp.1–23, Springer International Publishing, Aug. 2022.
- [32] C.D. Wilen, S. Abdullah, N.A. Kurinsky, C. Stanford, L. Cardani, G. D’Imperio, C. Tomei, L. Faoro, L.B. Ioffe, C.H. Liu, A. Opremcak, B.G. Christensen, J.L. DuBois, and R. McDermott, “Correlated charge noise and relaxation errors in superconducting qubits,” *Nature*, vol.594, no.7863, pp.369–373, June 2021.
- [33] A. Agarwal, L.P. Lindoy, D. Lall, F. Jamet, and I. Rungger, “Modelling non-markovian noise in driven superconducting qubits,” vol.9, no.3, 2023.
- [34] R. Wille, D. Große, L. Teuber, G.W. Dueck, and R. Drechsler, “RevLib: An online resource for reversible functions and reversible circuits,” *Int’l Symp. on Multi-Valued Logic*, pp.220–225, 2008. RevLib is available at <http://www.revlib.org>
- [35] C.C. Moran, *Mastering Quantum Computing with IBM QX: Explore the world of quantum computing using the Quantum Composer and Qiskit*, Packt Publishing Ltd, 2019.
- [36] M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, C.H. Yu, K. Chung, H. Jeech, B.G. Kim, and Y.D. Kim, “Evolutionary approach to quantum and reversible circuits synthesis,” *Artificial Intelligence Review*, vol.20, no.3, pp.361–417, Dec. 2003.



Martin Lukac obtained his B.Sc. Degree in Biology from Universite Bordeaux I, his M.Sc. Degree in Cognitive Sciences from Ecole Polytechnique and University Paris Jussieu, France, and his Master's and Ph.D. Degree in Computer Engineering from Portland State University, USA in 2009. From 2009 till 2014 he was Assistant Professor at Tohoku University, Sendai, Japan where his main area of research was intelligent robotics and novel-concept algorithm development. From 2015 till 2016 he had a po-

sition of Assistant Professor and from 2016 till 2023 he was an associate professor in the department of Computer Science at Nazarbayev University, Kazakhstan. Currently, he is Associate Professor in the Department of Computer Networks and Architecture at Hiroshima City University. His interests are quantum computing, artificial intelligence, causal reasoning and neural networks optimization. He pioneered evolutionary quantum logic synthesis and currently he is working in the area of adaptive robotic vision, meta-learning, Human-Computer Interaction and Deep Neural Network Optimization with the target being real world problems. He has published more than 80 Journal and Conference articles obtaining along the way two best paper awards. Currently he is working on several projects related to neural cryptography, real-time human and sports prediction, high level neural network optimizations and the design of quantum algorithms.



Saadat Nursultan graduated from Computer Science at Nazarbayev University in Astana and continuing to Master's degree from the University of Padua in September 2023. She has experience in the design and optimization of quantum circuits, cloud computing and IoT.



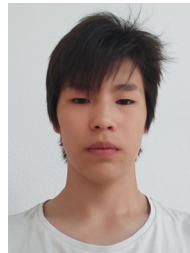
Georgiy Krylov is a PhD student at the University of New Brunswick, Canada. He has received his Bachelor's and Master's degree from Nazarbayev University, Astana, Kazakhstan in 2016 and 2018. He currently works on his research as part of IBM/UNB Centre for Advanced Studies – Atlantic (CASA). His research interests include, but are not limited to: Runtime Environments, Compilers Design, Logic Synthesis, Circuit Design (for both classical and quantum computing paradigms) and Computer Aided Circuit

Design. He has recently been involved with several open source projects, like Eclipse OMR and Verilog-to-Routing



Oliver Keszocze received a Diploma degree in applied mathematics and a B.Sc. in Computer Science from the University of Bremen in 2011. After a short career as a Software Engineer, he decided to pursue a Ph.D. in Computer Science in 2012 at his Alma Mater. Since 2014 he was also a Researcher with the German Research Center for Artificial Intelligence. In 2017, he received his doctoral degree (Dr. rer. nat) at the University of Bremen. Since 2018, he is with the Friedrich-Alexander-Universität

Erlangen-Nürnberg (FAU), Germany, where he is with the Chair for Hardware/Software-Co-Design. Prof. Keszocze's research interests include several aspects of Logic Synthesis and Computer Aided Design in various application domains and for different technologies. His current research focuses on Approximate Computing for both, ASICs and FPGAs. Prof. Keszocze serves as a TPC member for several conferences, including DATE, ICCAD, and ASP-DAC and is a reviewer for multiple IEEE journals. He is glad to be able to support young researchers by being part of DATE conference's PhD Forum TPC



Abilmansur Rakhmetulayev previously a student in NIS Physics and Mathematics Astana, Kazakhstan, is currently at Haileybury high-school after winning a scholarship. During his free time his research interests are quantum computing related to circuit optimization, and distinguishing sequences search using quantum computer.



Michitaka Kameyama his general research interests are intelligent integrated systems for real-world applications, advanced VLSI architecture, and new-concept VLSI including multiple-valued VLSI computing. He received the Outstanding Paper Awards at the 1984, 1985, 1987 and 1989 IEEE International Symposiums on Multiple-Valued Logic, the Technically Excellent Award from the Society of Instrument and Control Engineers of Japan in 1986, the Outstanding Transactions Paper Award from the

IEICE in 1989, the Technically Excellent Award from the Robotics Society of Japan in 1990, the Special Award at the 9th LSI Design of the Year in 2002, and the Outstanding Paper Award at the IADIS International Conference in 2013. He is Life Fellow of IEEE.