Nearest Neighbor Realization of Quantum Circuits in 2D Architecture

Sandeep Kumar Swain[1] and Anirban Bhattacharjee[2]

[1] *Kalinga Institute of Industrial Technology, Bhubaneswar, Odisha, India*
[2] *Kalinga Institute of Industrial Technology, Bhubaneswar, Odisha, India*

Abstract— Quantum computing has garnered significant attention among researchers due to its superiority over classical computing. However, the design and implementation of quantum circuits face challenges, particularly the Nearest Neighbor (NN) constraint, which requires qubits to interact only with their adjacent qubits. To satisfy this constraint, SWAP gates are employed, but this introduces another design issue—how to determine NN-compliant designs with minimal SWAP gate usage. In this work, we propose a heuristic approach for efficiently representing quantum circuits with NN constraints in a 2D space. Our technique consists of three stages: qubit selection, qubit placement, and SWAP gate insertion. We tested our approach on a wide range of benchmarks and observed significant reductions in cost parameters. Notably, we report improvements of over 38.798% and 24.015% in SWAP count and quantum cost, respectively, compared to 2D designs. This work contributes to the exploration of efficient circuit representations with NN constraints, offering promising results for optimizing quantum circuit design.

Keywords— Quantum circuits, Nearest Neighbor, Quantum gates, Qubit placement, SWAP gate insertion.

I. Introduction

The rise of Quantum Computing (QC) has brought forth a new paradigm for computation, offering promising advantages over classical computing in various domains, including quantum image processing, cryptography, machine learning, and database searching. Quantum algorithms are commonly represented using quantum circuit models, which are composed of elementary quantum gates. To ensure accurate computations, it has been observed experimentally that minimizing the distance between interacting qubits can help reduce errors [1]. Consequently, constructing quantum circuits requires satisfying the Nearest Neighbor (NN) restriction, where all interacting qubits must be physically adjacent.

While quantum circuits are typically designed independently of their physical implementation, achieving NN compliance often involves the insertion of SWAP gates. However, each SWAP operation incurs additional time and quantum cost, making it essential to minimize their usage. This optimization problem, known as the NN-compliant quantum circuit design, is known to be NP-complete. To address this challenge, heuristic approaches have been employed to find sub optimal solutions, as exact [2–4] optimization is impractical. Several scalable heuristic [5–9] search algorithms have been proposed in the literature.

In recent years, the two-dimensional (2D) architecture has gained attention due to its potential to reduce the number of required SWAP gates by allowing more adjacent neighbors for each qubit. Researchers have proposed various methods for efficiently realizing NN-compliant quantum circuits in the 2D architecture. Typically, this involves a two-step process: first, determining the initial qubit placements in a 2D grid, followed by the insertion of SWAP gates to achieve local ordering and NN compliance.

While [10–14]exact solutions for small-scale 2D quantum circuits have been proposed, their applicability is limited. Other approaches have focused on optimizing the interaction path between qubits using sliding window technique [15] or treating circuit mapping as a multi-objective optimization problem [16]. Some methods aim to minimize the number of SWAP gates for adjacent interactions, proposing algorithms that consider qubit allocation [17, 18] and utilize forward-looking strategies [19, 20].

In this work, we present a novel approach for NN-compliant quantum circuit design in 2D architectures. Our method involves an initial qubit placement technique and heuristic interaction routing strategies. We aim to minimize the number of SWAP gates required for achieving NN compliance. We evaluate the effectiveness of our proposed strategies on benchmark functions and compare them against existing methods.

The remainder of this paper is organized as follows: Section II provides background information on quantum circuits, followed by the description of our proposed qubit placement strategy in Section III and the SWAP gate insertion method in Section IV. Section V presents experimental results, and finally, we conclude in Section VI with remarks on the achieved outcomes.

II. Related Background

a. Qubits in Quantum Computing

In the realm of quantum computing, qubits serve as the fundamental units of information on which elementary quantum

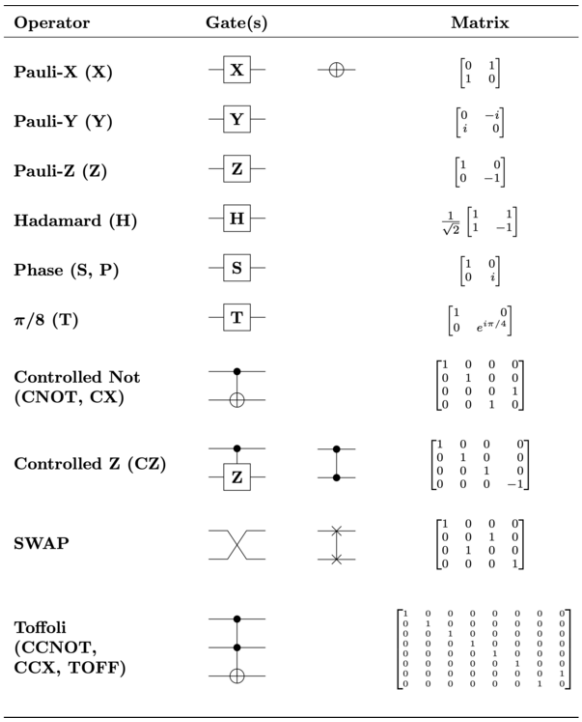

Fig. 1: Some well-known quantum gates and their schematic representations.

gates operate. These qubits, akin to classical bits, can exist in two basis states represented as $|1\rangle$ and $|0\rangle$, which correspond to the classical 1 and 0. However, qubits possess a unique characteristic that sets them apart from classical bits. They can also exist in a superposition state, which is a linear combination of the basis states $|1\rangle$ and $|0\rangle$. The state of a qubit, denoted as $|\psi\rangle$, can be expressed as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers known as probability amplitudes, subject to the constraint $\alpha^2 + \beta^2 = 1$.

b. Quantum Gates

Quantum gates play a crucial role in quantum circuits, enabling operations on qubits. These gates are responsible for the manipulation and transformation of quantum states during quantum computations. Some well-known quantum gates include the NOT gate, CNOT gate, V/V+ gate, and W/W+ gate. These gates form part of quantum gate libraries, such as NCV, NCVW, and $NCV_{|v_1\rangle}$, allowing the implementation of specific functions within gate-level descriptions.

SWAP gates are utilized in quantum circuits to exchange information between non-adjacent qubits by temporarily rearranging their positions. This involves using other gates, such as CNOT gates, to move one qubit to an adjacent location before performing the SWAP operation. However, the insertion of SWAP gates can impact circuit efficiency, as it adds extra gate operations and increases computation time.

c. Nearest Neighbor Property

While quantum gates can represent any desired function within a quantum circuit, practical implementation often necessitates certain constraints. One such significant criterion is the Nearest Neighbor (NN) property, which restricts the interaction of quantum gates to only adjacent qubits. This adjacency requirement arises from the need to mitigate errors introduced by noise sources in the quantum system. To enforce the NN property, SWAP gates are employed in quantum circuits. SWAP gates facilitate the exchange of qubit positions, ensuring that interacting qubits are physically adjacent.

d. Cost Metrics

The Nearest Neighbor Cost (NNC) quantifies the interaction distance between qubits for each two-qubit quantum gate (G). It is expressed as $NNC_G = |c - t| - 1$, where c represents the control line and t represents the target line of the gate. For a quantum circuit, the cumulative NNC (NNC_{CKT}) is obtained by summing the NNCs contributed by individual gates in the circuit, i.e., $NNC_{CKT} = \sum_G NNC_G$. By considering the NNC, we can evaluate the level of adjacency achieved in a quantum circuit. Number of quantum gates or the number of SWAP gates which are inserted can also be considered as cost metrics.

e. 2D Quantum Circuits

In addition to 1D representations, quantum circuits can also be represented in a 2D configuration. This involves utilizing a grid-structured planar graph (V,E), where each node $v \in V$ represents a qubit. The undirected edges $(u,v) \in E$ indicate interactions between qubits u and v, with $u,v \in V$. In a 2D representation, each node can have a maximum degree of four, allowing for more adjacent neighbors per qubit. Different grid configurations can be chosen, leading to diverse 2D representations of the same quantum circuit. These variations in arrangement affect the interaction distance between qubits within the circuit.

In this context, we aim to explore the design and optimization of 2D quantum circuits that satisfy the NN property. Our focus is on developing techniques to efficiently map quantum circuits onto a 2D grid structure, considering the placement of qubits and the insertion of SWAP gates. Through our approach, we seek to minimize the number of SWAP gates required for NN compliance, leading to improved circuit representations.

III. Qubit Placement Approach

a. Qubit Interaction Graph

To represent a quantum circuit with n quantum gates $\{g_1, g_2, \cdots, g_n\}$ and m quantum bits $\{q_1, q_2, \cdots, q_m\}$, a qubit interaction graph $IG = (V,E)$ can be constructed. The graph consists of m vertices, where each vertex $n_i \in V (1 \leq i \leq m)$ represents a qubit. The edges $e_{ij} \in E (i \neq j)$ signify the interactions between the vertices n_i and n_j. An adjacency matrix can be constructed from the interactions in the circuit is, where $Adj_{i,j} = \sum \frac{1}{t_{i,j}}$, $t_{i,j}$ is the time instant when q_i is the control qubit and q_j is the target qubit. Here, the adjacent matrix corresponding to Fig. 2 is shown as Fig. 3.

The weight of each edge w_{ij} in the graph is calculated as the sum of $Adj_{i,j}$ and $Adj_{j,i}$. The interaction graph corresponding to Fig. 2 is shown as Fig. 4.

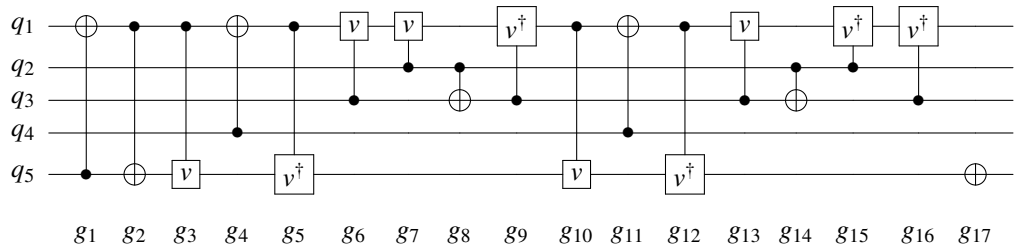

Fig. 2: Example Circuit

$$Adj = \begin{bmatrix} 0 & 0 & 0 & 0 & 1.22 \\ 0.21 & 0 & 0.2 & 0 & 0 \\ 0.42 & 0 & 0 & 0 & 0 \\ 0.34 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Fig. 3: Adjacency Matrix of the Example Circuit

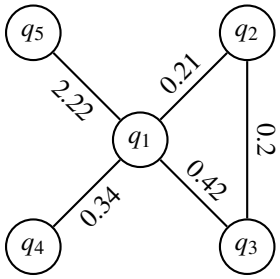

Fig. 4: Interaction Graph of the Example Circuit

b. Qubit Priority Selection

To establish qubit priorities, we employ a two-step procedure. Firstly, we identify the qubit with the highest number of connections, effectively measuring its degree of connectivity within the circuit. In cases of a tie, we further employ the Degree of Activity (DOA) metric, computed by summing the interaction weights of incoming and outgoing edges associated with a given qubit. The DOA metric serves as a tiebreaker and provides a measure of a qubit's overall engagement within the circuit.

Subsequently, employing a Depth-First Search (DFS) traversal strategy, we initiate the exploration of the quantum circuit from the qubit with the highest priority. During the DFS traversal, we consistently select the qubit that possesses the highest interaction weight on its outgoing edge to guide the circuit exploration. Furthermore, during backtracking within the DFS, we use the DOA values to guide the selection process when multiple qubits are pushed into the traversal queue.

In the example circuit provided, the qubit prioritization process unfolds as follows: Initially, qubit q_1 is selected as the starting point due to having the highest number of connections in the circuit. Subsequently, during the Depth-First Search (DFS) traversal, qubit q_5 is prioritized as it boasts the highest weight among its neighboring qubits. The traversal then backtracks to explore q_5's neighbors, leading to the selection of qubit q_3 based on its interaction factor among the pushed qubits. Continuing this heuristic approach, qubit q_2 is chosen from q_3's neighbor. Finally, the process concludes with the selection of qubit q_4. This systematic prioritization of qubits based on their connectivity and interaction weights optimizes the quantum circuit layout and gate sequencing, thereby enhancing error mitigation and overall quantum computing performance on noisy intermediate-scale quantum (NISQ) devices.

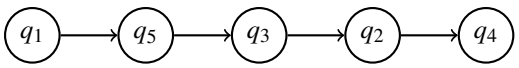

Fig. 5: Interaction Graph of the Example Circuit

c. Qubit Initial Placement

Finding the mapping method with the least interaction cost is proven to be an NP-complete problem, our approach focuses on minimizing the interaction cost between qubits to increase the number of quantum gates satisfying the NN (nearest-neighbor) constraint. By prioritizing qubits that frequently interact with each other and ensuring qubits have ample NN unoccupied positions, we can reduce the need for inserting SWAP gates and effectively decrease the quantum cost in quantum interactions.

The placement procedure starts with the selection of the first qubit from the priority queue. This initial qubit is then placed at the center position of the 2D grid, which serves as a reference point for subsequent placements. The algorithm then proceeds iteratively, placing each remaining qubit one by one, taking into account the already placed qubits to ensure nearest-neighbor unoccupied positions are identified for each new qubit.

For every new qubit placement, the algorithm evaluates a cost function that considers both the interaction factors and the number of SWAP gates required. The interaction factors reflect the strength of interactions between the current qubit and the qubits already placed in the grid. The SWAP gate count represents the effort needed to satisfy the nearest-neighbor constraint for the current qubit. The combination of these factors guides the algorithm in making optimal decisions regarding qubit placement.

The algorithm selects the position for placing the current qubit that yields the lowest overall cost. In case multiple positions offer the same minimum cost, the algorithm prioritizes the position that is closer to the center of the grid. This approach helps maintain a balanced distribution of qubits in the grid, mitigating potential imbalances that could lead to inefficiencies in quantum gate operations.

By repeating this iterative process for all qubits in the priority queue, the algorithm completes the qubit mapping in the 2D grid. The resulting layout is strategically designed to maximize the number of quantum gates satisfying the nearest-neighbor constraint while minimizing the need for SWAP gates, thus reducing the quantum cost associated with

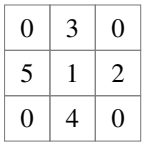

Fig. 6: Initial Qubit Placement

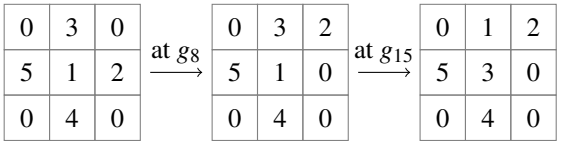

Fig. 7: Qubit Route for SWAP Gate insertion

quantum interactions. This heuristic placement approach provides an efficient and effective solution for qubit mapping in various quantum computing applications. The detailed flow of our proposed method is summarized in Algorithm 1.

Input : Qubit Priority pq and Adjacency Matrix $adjMat$
Output: 2D Grid $grid$ with qubit placements

Initialise the 2D Grid of size $M \times N$
$grid[center] \leftarrow pq$.pop()
while *pq not empty* **do**
  $qubit \leftarrow pq$.pop()
  $minCost \leftarrow \text{INF}, (minX, minY) \leftarrow (\text{INF}, \text{INF})$
  for ***each*** *neighbor* ***in*** *neighborList* **do**
    $cost \leftarrow 0$
    for ***each*** *placed* ***in*** *placedList* **do**
      $(i, j) \leftarrow (placed, neighbor)$
      $int\_factor \leftarrow adjMat[i][j] + adjMat[j][i]$
      $swaps \leftarrow \text{ManDist}(neighbor, placed)$
      $cost \leftarrow cost + swaps \times int\_factor$
    end
    if $cost < minCost$ **then**
      $minCost \leftarrow cost$
      $(minX, minY) \leftarrow (curX, curY)$
    end
    else if $cost is equal to minCost$ **then**
      if *(curX, curY) closer to grid*$[center]$ **then**
        $minCost \leftarrow cost$
        $(minX, minY) \leftarrow (curX, curY)$
      end
    end
  end
  $grid[minX][minY] \leftarrow qubit$
end
return *grid*

Algorithm 2: Qubit Placement in a 2D Grid

IV. Swap Gate Insertion Approach

For the initial qubit layout, the arrangement of qubits in a 2D grid may not always satisfy the nearest-neighbor (NN) constraints required for certain quantum gates. In such cases, we need to adjust the qubit layout by inserting SWAP gates between qubits to bring them into the NN state for proper interaction. The goal is to minimize the overall quantum cost of the quantum circuit, considering the impact of current NN operations on the interactions of subsequent quantum gates.

To achieve this, we define an "insertion path" as the set of possible shortest paths between two qubits that need to interact but are not in the NN configuration. The number of insertion paths between two qubit positions $p_1 = (x_1, y_1)$ and $p_2 = (x_2, y_2)$ is denoted as $nPath(p_1, p_2)$, and there is a recursive relation to compute this number i.e. $nPath(p_1, p) = nPath(p_1, (x-1, y)) + nPath(p_1, (x, y-1))$.

Additionally, we define an "insertion trace" as the sequence of SWAP gates inserted along an insertion path to bring the interacting qubits into the NN state. The number of insertion traces for an insertion path between two qubit positions is represented as $nTrace(p_1, p_2)$. For any given insertion path, there are $|x_2 - x_1| + |y_2 - y_1|$ possible insertion traces, and the number of inserted SWAP gates is $|x_2 - x_1| + |y_2 - y_1| - 1$.

Now, let's consider the interaction routing of a non-NN gate between two non-NN qubits (p_1 and p_2). For each non-NN gate, there are np insertion paths and nt insertion traces, where $np = nPath(p_1, p_2)$ and $nt = nTrace(p_1, p_2)$. This gives us $np \times nt$ possible ways to perform the interaction routing. To find the best routing policy, we perform all $np \times nt$ interaction routing policies in turn, based on the current qubit placement in the 2D grid. Each policy involves inserting the necessary SWAP gates to bring the interacting qubits into the NN state.

To illustrate this process with an example, suppose we have qubits $q_4, q_1, q_2, and q_3$ in the grid, and there exists a path $q_4 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3$. Let's consider the second qubit ($j = 2$) in this path. In the first round, we move q_4 to the jth position, i.e., $q_1 \rightarrow q_2 \rightarrow q_4$. This ensures that q_4 and q_3 become adjacent and satisfy the NN constraint hence no need of a second round, as q_4 and q_3 are already in the correct position to interact.

By following this algorithm and adjusting the qubit layout using SWAP gates strategically, we aim to minimize the quantum cost of the entire NN quantum circuit, ensuring that each non-NN operation has a positive impact on the interactions of subsequent quantum gates. This optimization process is essential for efficient and reliable quantum computations.

V. Experimental Results

The proposed approach has been implemented in a Jupyter Notebook and executed on a high-performance Intel i7 machine equipped with an octa-core CPU operating at 1.5 GHz and 16GB of RAM. To evaluate the effectiveness of the algorithm, extensive experimental assessments have been conducted using diverse benchmark suites sourced from the reputable RevLib benchmarks [21].

The performance evaluation of the algorithm is centered around two key metrics: the number of additional SWAP gates introduced (*nSWAP*) and the quantum cost (*qc*) of the quantum circuit after applying nearest-neighbor (NN) oper-

Input : Current Grid *grid* and positions (x_1, y_1) and (x_2, y_2) of the 2 interacting qubits
Output: List of updated grids *newGrids*

Calculate nt, np, and all possible shortest paths between the qubits
newGrids = []
for ***each*** *path* ***in*** *pathList* **do**
  $j \leftarrow 0$
  while $j < nt$ **do**
    newGrid $\leftarrow$ grid
    $k \leftarrow 0$
    while $k < j$ **do**
      $(x_1, y_1) \leftarrow$ path$[k]$
      $(x_2, y_2) \leftarrow$ path$[k+1]$
      *SWAP*newGrid$[x_1][y_1]$newGrid$[x_2][y_2]$
      $k \leftarrow k+1$
    end
    $k \leftarrow$ path.length -2
    while $k > j$ **do**
      $(x_1, y_1) \leftarrow$ path$[k]$
      $(x_2, y_2) \leftarrow$ path$[k+1]$
      *SWAP*newGrid$[x_1][y_1]$newGrid$[x_2][y_2]$
      $k \leftarrow k-1$
    end
    newGrids.append(*newGrid*)
    $j \leftarrow j+1$
  end
end
return *newGrids*

Algorithm 4: Qubit interaction Routing Algorithm

ations. The quantum cost (*qc*) is calculated as the sum of the original number of fundamental quantum gates (*ng*) in the circuit and three times the number of added SWAP gates ($3 \times nSWAP$).

The comparative analysis with previous works, including [18], [17], [20], [22], reveals a consistent improvement in the cost parameters achieved by the proposed approach. In the best-case scenario, the qubit mapping policy showcased an impressive enhancement of approximately 40.15% and 26.55% over the 2D approaches presented in [18] and [22], respectively in Table 1 which contains outcomes from small and medium-sized benchmarks. Table 2 summarizes the experimental data for larger benchmarks over the approaches presented in [17] and [20] which displays a 60.10% and 51.05% improvement on the cited works.

The experimental findings suggest that the proposed approach offers promising results in terms of reducing SWAP gates. These findings provide valuable insights into the efficacy of the algorithm for optimizing qubit layout, a critical aspect for improving the efficiency and performance of quantum computations. Overall, the rigorous evaluation, meticulous analysis, and compelling results presented in the research paper underscore the significance of the proposed qubit mapping approach in reducing SWAP gate overhead and enhancing the quantum cost for a wide range of quantum circuits.

VI. CONCLUSIONS

In conclusion, this study introduces a novel heuristic-driven qubit placement strategy aimed at transforming quantum circuits into designs compliant with nearest-neighbor (NN) constraints. The design process encompasses three pivotal stages: qubit prioritization, qubit positioning, and SWAP gate insertion. Through extensive testing across a diverse spectrum of benchmarks, the proposed approach has demonstrated notable improvements, particularly in achieving a substantial reduction in SWAP gate usage.

Comparisons with some of the best 2D representations available in the literature were performed, and the results were promising. Our heuristic-based approach consistently outperformed the 2D configurations, showcasing average reductions of approximately 38.798% and 24.015% in SWAP gate usage over the same cost parameters.

It is important to note that while our approach has yielded remarkable results, we acknowledge that the initial qubit mapping process still relies on the applied heuristic and does not guarantee an optimal solution in all cases. Therefore, for future enhancements, we aim to explore and develop more sophisticated and appropriate heuristics to further optimize the qubit mapping process.

Overall, this research lays a solid foundation for advancing qubit placement strategies for quantum circuits towards improved NN-compliance and reduced SWAP gate overhead. The achieved improvements in SWAP gate usage demonstrate the effectiveness of the proposed approach, and the potential for further optimization through enhanced heuristics opens avenues for continued research and development in this field.

TABLE 1: REDUCTIONS IN SWAP GATES AND QUANTUM COST COMPARED TO 2D TECHNIQUES FOR SMALL AND MEDIUM-SIZED BENCHMARK CIRCUITS

Benchmarks			Grid Size	[18]		[22]		Our Work		Improvements (in %)	
Name	nq	ng		nSwaps	qc	nSwaps	qc	nSwaps	qc	[18]	[22]
3_17_13	3	14	2x2	6	32	5	29	4	26	33.33	20
4_49_17	4	32	2x2	13	71	10	62	9	59	30.77	10
4gt10-v1_81	5	36	3x2	16	84	14	78	10	66	37.5	28.57
4gt11_84	5	7	2x3	2	13	2	13	2	13	0	0
4gt12-v1_89	6	55	3x2	19	112	20	115	15	100	21.05	25
4gt13-v1_93	5	17	3x3	-	-	3	26	2	23	-	33.33
4gt4-v0_80	6	43	2x3	17	94	16	91	12	79	29.41	25
4gt5_75	5	22	3x2	8	46	9	49	6	40	25	33.33
4mod5-v1_23	5	24	2x3	11	57	8	48	7	45	36.36	12.5
4mod7-v0_95	5	40	3x3	13	79	10	70	8	64	38.46	20
aj-e11_165	5	60	2x3	24	132	18	114	15	105	37.5	16.67
alu-v4_36	5	32	3x3	10	62	11	65	7	53	30	36.36
cnt3-5_180	16	125	3x6	69	332	54	287	39	242	43.48	27.78
ham7_104	7	87	3x3	48	231	38	201	23	156	52.08	39.47
hwb4_52	4	23	2x2	9	50	7	44	7	44	22.22	0
hwb5_55	5	109	3x2	45	244	38	223	31	202	31.11	18.42
hwb6_58	6	146	2x3	79	383	63	335	43	275	45.57	31.75
mod5adder_128	6	87	3x2	41	210	28	171	22	153	46.34	21.43
mod8-10_177	6	108	3x3	45	243	39	225	28	192	37.78	28.21
QFT10	10	45	3x4	53	204	33	144	23	114	56.60	30.30
QFT5	5	10	2x3	5	25	5	25	3	19	40	40
QFT6	6	15	2x3	6	33	5	30	5	30	16.67	0
QFT7	7	21	3x3	18	75	14	63	9	48	50	35.71
QFT8	8	28	4x2	18	82	18	82	12	64	33.33	33.33
QFT9	9	36	3x3	34	138	24	108	16	84	52.94	33.33
rd32-v0_67	4	8	2x2	2	14	2	14	2	14	0	0
rd53_135	7	78	3x3	39	195	29	165	23	147	41.03	20.69
rd73_140	10	76	4x3	37	187	28	160	22	142	40.54	21.43
rd84_142	15	112	4x4	54	274	48	256	36	220	33.33	25
sys6-v0_144	10	62	4x3	31	155	30	152	21	125	32.26	30
Total				772	3857	629	3445	462	2944	40.16	26.55

TABLE 2: REDUCTIONS IN SWAP GATES AND QUATUM COST COMPARED TO 2D TECHNIQUES FOR LARGE-SIZED BENCHMARK CIRCUITS

Benchmarks			Grid Size	[17]		[20]		Our Work		Improvements (in %)	
Name	nq	ng		nSwaps	qc	nSwaps	qc	nSwaps	qc	[17]	[20]
cycle10_2_110	12	1224	4x3	588	2988	598	3018	337	2235	42.69	43.65
ham15_108	15	458	5x3	280	1298	249	1205	132	854	52.86	46.99
hwb7_62	8	2683	3x3	1500	7183	1292	6559	715	4828	52.33	44.66
sym9_148	10	4452	4x3	2789	12819	2065	10647	872	7068	68.73	57.77
Total				5160	24306	4206	21444	2059	15003	60.10	51.05

REFERENCES

[1] H. Häffner, W. Hänsel, C. F. Roos, J. Benhelm, D. C. al kar, M. Chwalla, T. Körber, U. D. Rapol, M. Riebe, P. O. Schmidt, C. Becher, O. Gühne, W. Dür, and R. Blatt, "Scalable multiparticle entanglement of trapped ion," *Nature*, vol. 438, no. 7068, pp. 646–663, 2005.

[2] Y. Hirata, M. Nakanishi, S. Yamashita, and Y. Nakashima, "An efficient conversion of quantum circuits to a linear nearest neighbor architecture," *Quantum Info. Comput.*, vol. 11, no. 1, pp. 142–166, 2011.

[3] R. Wille, A. Lye, and R. Drechsler, "Exact reordering of circuit lines for nearest neighbor quantum architectures," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst*, vol. 33, no. 12, pp. 1818–1831, 2014.

[4] ——, "Optimal SWAP gate insertion for nearest neighbor quantum circuits," in *Proc. of the 19th Asian South Pacific Design Autom. Conf. (ASP-DAC)*, 2014, pp. 489–494.

[5] A. Shafaei, M. Saeedi, and M. Pedram, "Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures," in *Proc. 50th Annual Design Autom. Conf. DAC*, 2013, p. 41.

[6] M. AlFailakawi, L. AlTerkawi, I. Ahmad, and S. Hamdan, "Line ordering of reversible circuits for linear nearest neighbor realization," *Quantum Inf. Process.*, vol. 12, no. 10, pp. 3319–3339, 2013.

[7] A. Kole, K. Datta, and I. Sengupta, "A heuristic for linear nearest neighbor realization of quantum circuits by SWAP gate insertion using gate look-ahead," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 6, no. 1, pp. 62–72, 2016.

[8] X. Y. Cheng, G. Zhi-Jin, X. Hai, T. Ying-Ying, and L. Yang, "The nearest neighobr arrangement of quantum circuits based on MCT reversible circuits," *Aata Electronica Sinica*, vol. 46, no. 8, pp. 1891–1897, 2018.

[9] C. Xueyun, G. Zhijin, D. Weiping, and Z. Pengcheng, "Linear nearest neighbor quantum circuit synthesis and optimization based on the matrix," *Aata Electronica Sinica*, vol. 46, no. 3, pp. 688–694, 2018.

[10] M. Saeedi, R. Wille, and R. Drechsler, "Synthesis of quantum circuits for linear nearest neighbor architectures." *Quantum Info. Comput.*, vol. 10, no. 3, pp. 355–377, 2011.

[11] A. Lye, R. Wille, and R. Drechsler, "Determining the minimal number of swap gates for multi-dimensional nearest neighbor quantum circuits," in *Proc. ASP-DAC*, 2015, pp. 178–183.

[12] A. Zulehner, S. Gasser, and R. Wille, "Exact Global Reordering for Nearest Neighbor Quantum Circuits Using A*," *International Conference on Reversible Computation. Springer*, pp. 185–201, 2017.

[13] M. Y. Siraichi, F. M. Q. Pereira, V. D. Santos, and C. Collange, "Qubit allocation," in *Proc. Anais do Concurso de Teses e Dissertaçes da SBC (CTD-SBC)*, 2020, pp. 113–125.

[14] M. G. Alfailakawi, I. Ahmad, and S. Hamdan, "Harmony-search algorithm for 2D nearest neighbor quantum circuits realization," *Expert Systems with Applications*, vol. 6, no. 1, pp. 16–27, 2016.

[15] C. Lin, S. Sur-Kolay, and N. K. Jha, "PAQCS: Physical design-aware fault-tolerant quantum circuit synthesis," *IEEE Transactions on Very Large Scale Integrated (VLSI) Systems*, vol. 23, no. 7, pp. 1221–1234, 2015.

[16] D. Ruffinelli and B. Barán, "Linear nearest neighbor optimization in quantum circuits: A multiobjective perspective," *Quantum Inf. Process.*, vol. 16, no. 9, p. 220, 2017.

[17] R. R. Shrivastwa, K. Datta, and I. Sengupta, "Fast qubit placement in 2D architecture using nearest neighbor realization," in *Proc. IEEE Int. Symp. Nanoelectronic Inf. Syst.*, 2015, pp. 95–100.

[18] A. Shafaei, M. Saeedi, and M. Pedram, "Qubit placement to minimize communication overhead in 2d quantum architectures," in *Proc. ASP Design Autom. Conf*, 2014, pp. 495–500.

[19] R. Wille, O. Keszocze, M. Walter, P. Rohrs, A. Chattopadhyay, and R. Drechsler, "Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits," in *Proc. ASP Design Autom. Conf*, 2016, pp. 292–297.

[20] A. Kole, K. Datta, and I. Sengupta, "A new heuristic for N-dimensional nearest neighbor realization of a quantum circuit," *IEEE Transactions on Computer-Aided Design Integrated Circuits and Systems*, vol. 37, no. 1, pp. 182–192, 2018.

[21] R. Wille, D. Gro, L. Teuber, G. W. Dueck, and R. Drechsler, "RevLib: An online resource for reversible functions and reversible circuits," in *Proc. 38th Int. Symp. Multiple Valued Log. (ISMVL)*, 2008, pp. 220–225.

[22] A. Bhattacharjee, C. Bandyopadhyay, R. Wille, R. Drechsler, and H. Rahaman, "A Novel Approach for Nearest Neighbor Realization of 2D Quantum Circuits," *2018 IEEE Comp. Society Annual Symp. on VLSI*, 2018.