Nearest Neighbor Transformation of Quantum Circuits in 2D Architecture

XUEYUN CHENG, ZHIJIN GUAN, AND PENGCHENG ZHU, (Student Member, IEEE)
School of Information Science and Technology, Nantong University, Nantong 226019, China
Corresponding author: Zhijin Guan (guan.zj@ntu.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 62072259.

ABSTRACT In recent years, quantum computing has received extensive attention for its superior efficiency and application potential. For some physical architecture, one qubit can only interact with its adjacent qubits, SWAP gates are inserted to make a quantum circuit nearest neighbor compliant. The initial qubit placement algorithm in a 2D grid structure is proposed based on the constructed interaction cost metric model, and the method of dynamic grid decision is given. In order to obtain a better way of inserting SWAP gates, the gate level and circuit level interaction routing policies are put forward, and a heuristic pruning of interaction routing records is employed so as to reduce the runtime and additional quantum cost. Experimental results show that our proposed methods can achieve better performance than existing methods, and the average optimization rate of quantum cost is 21.76% and 17.23% respectively.

INDEX TERMS Quantum circuit, 2D grid structure, nearest neighbor, qubit placement, qubit interaction routing.

I. INTRODUCTION Quantum Computing (QC) is an alternative to classical computation owing to its potential advantages in various important applications, including quantum image processing [1], cryptography [2], [3], machine learning [4], database searching [5], and others. Quantum circuit model is a common representation of quantum algorithms, and it is cascaded with a series of elementary quantum gates. Experimental observations suggest that interaction between qubits close to each other can reduce computation errors [6]. Several physical systems such as ion traps [7], quantum dots [8] and superconducting [9] can be employed to construct quantum circuits, which require that the nearest neighbor (NN) restriction should be satisfied, i.e., all interacting qubits in a quantum circuit are physically adjacent to each other.

Quantum circuits are often designed without consideration of their physical implementation. Several strategies of NN conversion have been proposed, which convert a quantum circuit to its equivalent form complying with the NN restriction by inserting SWAP gates. Each SWAP operation increases the running time and quantum cost, so reduction the number of SWAP gates is important.

The design of an optimal NN-compliant quantum circuit by inserting the minimal number of SWAP gates is an NP-complete problem [10]. Researchers have solved this problem by means of heuristic approaches that do not guarantee optimality. The one-dimensional (1D) or linear nearest neighbor (LNN) architecture is the simplest where the qubits are ordered linearly, and each qubit interacts with the previous and next qubits. Exhaustive [11] and exact search [12], [13] approaches are well-studied for smaller quantum circuits. Several heuristic search approaches are reported in the literature [14]–[18] that are scalable.

The two-dimensional (2D) architecture allows more number of adjacent neighbors for each qubit, so it can reduce the number of SWAP gates required for NN-compliance. Recently several works have been reported for efficient NN-compliant realization of quantum circuits in the 2D architecture. The problem of synthesizing a 2D NN-compliant quantum circuit is typically approached as a two-step process. First, an initial mapping of qubits in a 2D grid is identified (global ordering). Next, SWAP gates are actually inserted in order to make the circuit NN-compliant (local ordering).

In [19] the exact solution for the NN optimization of 2D quantum circuits was proposed, which provided a measure of reference for other heuristic mapping algorithms, but it can only be applied to small-scale circuits. In [20] the authors optimized the interaction path between the qubits through the sliding window technique, and the performance of the algorithm is affected by the size of the window. In [21], authors regarded the mapping of the quantum circuit as a multi-objective optimization problem, which was solved by using an evolutionary algorithm, but there is still a large optimization space in the construction of the qubit placement. A simple and effective algorithm to minimize the number...
of SWAP gates for adjacent interactions was proposed in [22], and the time required for processing does not exceed one second even for larger benchmark circuits, however, the precise grid approach can be improved to get better results. A forward-looking algorithm to reduce the number of SWAP gates for NN optimization in the 2D architecture was put forward in [23]. In order to meet the connectivity constraints in the mapping process, an exact solution was given in [24] for the qubit allocation problem, furthermore, the optimization algorithm needs to be extended to a more complex architecture.

In the present work, an initial qubit placement method and heuristic interaction routing strategies have been proposed and evaluated on some benchmark functions. The optimization objective is to minimize the number of inserted SWAP gates. Experimental results show that the proposed strategies have achieved better results compared to existing methods.

The remainder of the paper is organized as follows. Section II presents fundamental knowledge on quantum circuits, followed by a description of the proposed 2D global and local ordering approach in Section III and Section IV; Section V discusses the experimental results. Finally, concluding remarks are presented in Section VI.

II. PRELIMINARIES

A. THE NEAREST NEIGHBOR QUANTUM CIRCUIT

In quantum computing, the qubit is used as a basic unit of information, and it is different from classical bit. One qubit can be in a linear combination of two basis states, $|0\rangle$ and $|1\rangle$, i.e. $\psi = \alpha |0\rangle + \beta |1\rangle$ where $\alpha^2 + \beta^2 = 1$. However, the classical bit can only represent 0 or 1 at a time.

An $n$-qubit quantum gate operates on $n$ qubits can be represented by a unitary matrix of size $2^n \times 2^n$, such as NCV gates, Clifford+T gates. In this paper, we consider quantum gates from NCV library [25] which includes NOT gate, CNOT gate, the controlled-V(CV) gate, and the controlled-V$^+$ (CV$^+$) gate.

A quantum circuit consists of some qubits and a sequence of quantum gates operating on these qubits. For example, a quantum circuit is depicted in Fig. 1.

![FIGURE 1. Example of a quantum circuit.](image)

Several quantum technologies restrict quantum gates to operate only on physically adjacent qubits, also known as nearest neighbor constraint. If the current two-qubit quantum gate satisfies the NN constraint, it is called the NN gate, otherwise it is called the non-NN gate. SWAP gates can be inserted before each non-NN quantum gate to make it satisfy the physical constraint in this regard. A SWAP gate can be realized using three CNOT gates operating on alternate qubits in sequence, as shown in Fig. 2.

![FIGURE 2. Equivalence relation of a SWAP gate.](image)

As a result, circuit transformation is required to make all non-NN gates in the original circuit turn into the NN gates, and the obtained circuit is called the NN quantum circuit.

B. COST METRICS

Some cost metrics have been proposed to estimate the cost of implementing the NN quantum circuits [23], the metrics associated with this paper are as follows.

1) Quantum cost ($qc$) is a generally used cost metric that is the number of elementary quantum gates.

2) SWAP count ($nSwap$) means the number of SWAP gates which are inserted before all non-NN gates.

3) Nearest neighbor cost (NNC) of a two-qubit quantum gate in 1D architecture is expressed as $|e - t| - 1$ where $|e\rangle$ is the order in which the control qubit is placed in the circuit, and $t$ is that of the target qubit. For the positions $(x_1, y_1)$ and $(x_2, y_2)$ of two qubits in a 2D architecture, the NNC is $|x_1 - x_2| + |y_1 - y_2| - 1$.

III. DYNAMIC GLOBAL ORDERING APPROACH

A. INTERACTION GRAPH AND DEGREE OF ACTIVITY

For a quantum circuits with $n$ quantum gates $\{g_1, g_2, \ldots, g_n\}$ and $m$ quantum bits $\{q_1, q_2, \ldots, q_m\}$, a directed qubit interaction graph $IG = (V, E)$ with $m$ vertices can be generated, where any vertex $n_i \in V(1 \leq i \leq m)$ represents each qubit, any directed edge $e_{ij} \in E(i \neq j)$ denotes the interaction between the vertex $n_i$ and $n_j$, and the direction of the directed edge is from the control qubit to the target qubit. The weight of each edge $w_{ij}$ is the times of interaction. For the quantum circuit shown in Fig. 1, the generated IG is shown in Fig. 3.

![FIGURE 3. Qubit interaction graph.](image)

The adjacency matrix corresponding to the interaction graph in Fig. 3 is shown in Fig. 4, where $\text{Adj}_{i,j} = w_{q_i \rightarrow q_j}$.

**Definition 1:** For a qubit, its degree of activity (DoA) is defined as the times the qubit participates in quantum interactions throughout a quantum circuit, that is, the times
the qubit is involved in the two-qubit gates. Suppose \( \text{Adj} \) is the adjacency matrix, and the DOA of qubit \( q_i \) is calculated by formula (1).

\[
\text{DoA}(q_i) = \sum_{k=0}^{m-1} \text{Adj}_{i,k} + \sum_{k=0}^{m-1} \text{Adj}_{k,i}
\]

The DOA corresponding to \( \{q_0, q_1, q_2, q_3, q_4, q_5\} \) in Fig.3 is \( \{2, 6, 4, 3, 3, 4\} \). The value of DOA determines the times of interactions between the qubit and other qubits, so the higher the DOA, the closer the qubits should be close to each other.

### B. INTERACTION COST

Two qubits \( q_i \) and \( q_j \) in the 2D grid, \( (x_i, y_i) \) and \( (x_j, y_j) \) are their positions, the Manhattan distance between them is as follows.

\[
\text{ManDist}(q_i, q_j) = |x_i - x_j| + |y_i - y_j| - 1
\]

The number of interactions between \( q_i \) and \( q_j \) is represented by \( \text{DoA}(q_i, q_j) \), and it is defined as the following formula.

\[
\text{DoA}(q_i, q_j) = \text{Adj}_{i,j} + \text{Adj}_{j,i}
\]

During qubit layout, the interaction cost of \( q_i \) placed at the current position is calculated as follows.

\[
\text{Cost}(q_i) = \sum_{j=0}^{k-1} \text{ManDist}(q_i, q_j) \times \text{DoA}(q_i, q_j)
\]

where \( k \) is the number of qubits which have already been placed into the grid. The function defines the interaction cost between the current qubit to be placed and all the qubits that have been placed. If the position selected when placing the current qubit is close to the qubits with which it often interacts, a lower cost can be obtained, and the less SWAP gates are needed.

### C. INITIAL QUBIT PLACEMENT AND DYNAMIC GRID DECISION

In order to solve the problem of grid size selection, a dynamic grid selection method is put forward in this paper. Finding the mapping method with the least interaction cost has been proven to be a NP complete problem. In order to minimize the interaction cost between the qubits and make more quantum gates satisfy the NN constraint at first, we must give priority to the qubits that often interact with each other and ensure the qubits have as many NN unoccupied positions as possible, so as to reduce the probability of inserting SWAP gates and decrease the quantum cost in quantum interaction. All qubits are sorted in terms of the value of interaction cost, and they are stored in the priority queue \( \text{PriorityQueue} \).

The initial qubits placement algorithm of 2D grid structure is given. The first qubit is popped up from the queue and it is placed at the center position of the grid. The second qubit is continued, and we find the NN unoccupied position based on the qubits that have been placed in the grid. Then, according to the cost function, the cost is calculated when placed in a NN position. After traversing all the possible positions, the position with the least cost is selected, the current qubit is placed into it, and the information of the placed qubits is updated. When all the qubits are placed, the qubits layout of the 2D grid structure can be obtained, and the size of the 2D grid can be determined. Because in the initial qubit layout, the qubit is arranged around the center of the grid, and owing to the characteristics of the cost function, there will be no imbalance in the qubit layout, so the size of the 2D grid established at this time is reasonable. The detailed flow is as Algorithm 1.

### D. EXAMPLE FOR INITIAL PLACEMENT

For the quantum circuit as Fig. 1, by sorting the DOA of each qubit, the priority queue shown in Fig.5 can be obtained.

\[
q_1 \rightarrow q_5 \rightarrow q_2 \rightarrow q_3 \rightarrow q_4 \rightarrow q_6
\]

### FIGURE 5. Qubit priority queue.

An unoccupied 2D grid of sufficient size is built, and the first qubit \( q_1 \) of the current priority queue is placed into the center position of the 2D grid, then the next qubit \( q_5 \) in the priority queue is taken out. A total of four positions with the NN can be selected. The cost functions of the four free positions are calculated respectively, and the values are equal, so the position closest to the center qubit is selected. Then qubit \( q_2 \) in the priority queue is carried out, and so on, the placement scheme with the smallest cost function is selected, and the grid shown in Fig.6 is obtained.

### FIGURE 6. The obtained initial qubit placement.

Finally, the minimized grid size needs to be generated according to the current qubit layout in the 2D grid. It can be seen that in this case, a 2D grid with \( 3 \times 3 \) is obtained, and three free positions are generated. The qubits with high interaction frequency have satisfied the NN constraint in the grid, and can interact directly, which is in line with the expected goal of the method.

### IV. PRIORITY-BASED LOCAL ORDERING APPROACH

For the initial qubit layout, the relationship between the qubits in the existing 2D grid cannot fully satisfy the NN constraints...
the interaction of the subsequent quantum gates. Therefore, during performing the interaction involved in the quantum gates in order, the layout state must be adjusted by inserting the SWAP gates to make the interacted two qubits in the NN state. For the current non-NN quantum gate, there are many ways to insert SWAP gates, the quantum cost caused by different ways is certain, it only depends on the NNC of the quantum gate, but it will cause the qubit layout state in the 2D grid to change, which affects the interaction of the subsequent quantum gates. Therefore, we must find a scheme to minimize the quantum cost of the whole NN quantum circuit, so it is the best that the current NN operation can have a positive impact on the interaction of the subsequent quantum gates.

A. GATE LEVEL INTERACTION ROUTING POLICY

1) INSERTION PATH

If two interacting qubits do not satisfy the NN constraint, they need to be moved closer to each other, one or more SWAP gates are inserted in between.

Definition 2: Let \( p_1 = (x_1, y_1) \) and \( p_2 = (x_2, y_2) \) are the positions of two qubits \( q_1, q_2 \), there exit several shortest paths between them to realize adjacent interaction, each of them is defined as insertion path.

The number of insertion paths between \( p_1 = (x_1, y_1) \) and \( p_2 = (x_2, y_2) \) is represented as \( nPath(p_1, p_2) \).

Theorem 1: The position of the upper left corner in the grid is \( p_1 = (x_1, y_1) \) and set \( nPath(p_1, p_1) = 1 \), for any other position \( p = (x, y) \), the recursion relation is satisfied: \( nPath(p_1, p) = nPath(p_1, (x - 1, y)) + nPath(p_1, (x, y - 1)) \).

Proof: Owing to the NN interaction and the least extra SWAP gates, there is only one path for \( (x_1 + 1, y_1) \), and it’s the same to position \( (x_1, y_1 + 1) \), i.e. \( nPath(p_1, (x_1 + 1, y_1)) = 1 \) and \( nPath(p_1, (x_1, y_1 + 1)) = 1 \). For \( (x_1 + 1, y_1 + 1) \), there are two shortest precursor paths, from \( (x_1 + 1, y_1) \) and \( (x_1, y_1 + 1) \), so the number of paths from \( (x_1, y_1) \) to \( (x_1 + 1, y_1 + 1) \) is the sum of \( nPath((x_1 + 1, y_1)) \) and \( nPath((x_1, y_1 + 1)) \), the rest may be deduced by analogy, so for any position \( (x, y) \) in the grid, \( nPath(p_1, p) = nPath(p_1, (x - 1, y)) + nPath(p_1, (x, y - 1)) \). □

2) INSERTION TRACE

Definition 3: In order to realize NN interaction of the non-NN gate, SWAP gates are inserted so that the control qubit and the target qubit are physically close to each other, the operation trace of inserting SWAP gates on an insertion path is defined as insertion trace.

The number of insertion traces between \( p_1 \) and \( p_2 \) for an insertion path is represented as \( nTrace(p_1, p_2) \).

Theorem 2: Two non-NN positions \( (x_1, y_1) \) and \( (x_2, y_2) \), there exist \(|x_2 - x_1| + |y_2 - y_1| \) insertion traces for any insertion path, and the number of inserted SWAP gates is \(|x_2 - x_1| + |y_2 - y_1| - 1 \).

Proof: Let \( path \) is an insertion path between them, there are \(|x_2 - x_1| + |y_2 - y_1| + 1 \) ends on this path, the optional position of the control qubit or the target qubit is \( |x_2 - x_1| + |y_2 - y_1| \), so there are \(|x_2 - x_1| + |y_2 - y_1| \) ways to insert SWAP gates. Although the qubit exchange mode in each way is different, the NN interaction of qubits is finally realized, and the number of inserted SWAP gates is equal to the sum of the number of movements of the control bit and target bit, i.e. \(|x_2 - x_1| + |y_2 - y_1| - 1 \). □

B. INTERACTION ROUTING OF A NON-NN GATE

Definition 4: Suppose \( p_1 = (x_1, y_1) \) and \( p_2 = (x_2, y_2) \) are the positions of two non-NN qubits, there exist \( np \) insertion
paths and \( nt \) insertion traces for each path, where \( np = nPath(p_1, p_2) \) and \( nt = nTrace(p_1, p_2) \), one of the \( np \times nt \) ways is defined as interaction routing of a non-NN Gate.

For a non-NN quantum gate, according to the current qubit placement \( grid[size][size] \), all \( np \times nt \) interaction routing policies are performed in turn, and the modified qubit placements \( PlacementList[np \times nt][size][size] \) corresponding to all policies are returned. The detailed process is as Algorithm 2.

For example, there exists a path \( q_4 \rightarrow q_1 \rightarrow q_2 \rightarrow q_3 \) in Fig.6, and \( j = 2 \) is considered. The first round moves \( q_4 \) to the \( j^{th} \) position, i.e. \( q_1 \rightarrow q_2 \rightarrow q_4 \), and no need for the second round, because \( q_4 \) and \( q_3 \) are adjacent.

C. CIRCUIT LEVEL INTERACTION ROUTING POLICY

1) INTERACTION ROUTING RECORDS

There is an exchange rule about two adjacent two-qubit gates \( g_i \) and \( g_j \), if \( g_i.c \neq g_j.t \) and \( g_i.t \neq g_j.c \), they can be exchanged and keep the circuit functional equivalent [26].

Definition 5: During quantum circuit mapping, each non-NN gate is dealt with in turn so that its control qubit and target qubit satisfy the NN constraint. The number of non-NN gates that have been processed is defined as number of NN operation, denoted by \( N\text{Ncount} \).

Definition 6: For the current non-NN gate \( g_i \), one of its NN interaction routing methods is used to make it satisfy the NN constraint, if the updated qubit placement make the subsequent \( k \) gates become NN gates, the \( (i + k + 1)^{th} \) gate is called the farthest reachable quantum gate, represented as \( Gfar \).

In order to record the current state in the circuit mapping process, the interaction routing record is introduced, and it contains: \( N\text{Ncount}, Gfar, QC \), that has been extra generated by inserting SWAP gates (\( QC_{extra} \)), and the detailed evolution of qubit placement. For all records, we construct a priority queue to reorder them in terms of the following rules.

First of all, the \( N\text{Ncount} \) is compared, and the smaller the \( N\text{Ncount} \), the more likely it is to reach the farther quantum gate, so it has a higher priority. For records with the same \( N\text{Ncount} \), the \( Gfar \) is considered, the farther the \( Gfar \) is, which indicates that the insertion of the SWAP gates in the record is more favorable to the subsequent quantum gate. If the \( Gfar \) is the same, the smaller the \( QC_{extra} \), which indicates that the insertion method of the SWAP gates can not only reach the farthest quantum gate, but also be realized by inserting the least SWAP gates, which is the best choice.

2) PREFERRED INTERACTION ROUTING RECORDS

In order to solve the problem of exponential increase of the algorithm in the process of qubit interaction routing selection, we pruned the routing records at a specific time, eliminated some records that are almost impossible to be locally optimal or globally optimal.

Definition 7: The minimum required value of \( Gfar \) for the current non-NN quantum gate is defined as preferred threshold.

Algorithm 2: Gate Level Interaction Routing Algorithm

**Input:** current qubit placement \( grid[size][size] \), positions of two interaction qubits \((x_1, y_1)\) and \((x_2, y_2)\)

**Output:** list of qubit placement corresponding to each NN interaction routing.

1. calculate \( np \) and \( nt \) and obtain all insertion paths \( PathList[np] \)
2. \( PlacementList[np \times nt][size][size] = [grid[size][size]] \)
3. \( i = 0 \)
4. while \( PathList \) is not empty do
   5. \( path \leftarrow PathList.pop_front() \)
   6. \( j \leftarrow 0 \)
   7. while \( j < nt \) do
      8. \( k \leftarrow 0 \)
      9. while \( k < j \) do
         10. \( \text{SWAP } Placement[i\times nt][path[k].x][path[k].y] \)
         11. \( \quad Placement[i\times nt][path[k+1].x][path[k+1].y] \)
         12. \( k \leftarrow k + 1 \)
      13. \( \text{end} \)
      14. \( k \leftarrow path.length-2 \)
      15. while \( k > j \) do
         16. \( \text{SWAP } Placement[i\times nt][path[k].x][path[k].y] \)
         17. \( \quad Placement[i\times nt][path[k+1].x][path[k+1].y] \)
         18. \( k \leftarrow k - 1 \)
      19. \( \text{end} \)
      20. \( j \leftarrow j + 1 \)
   21. \( \text{end} \)
22. \( i \leftarrow i + 1 \)
23. \( \text{end} \)
24. return \( PlacementList[np \times nt][size][size] \)

- i. \( path[nt] \) contains all position information that the path passes through
- ii. \( PathList[np] \) stores interaction paths between \((x_1, y_1)\) and \((x_2, y_2)\)
- iii. \( PlacementList[np \times nt][size][size] \) contains \( np \times nt \) qubit placements corresponding to all NN interaction routing policies
- iv. Two rounds of exchange to realize the nearest neighbor of the non-NN gate at the \( j^{th} \) and \((j+1)^{th} \) positions. The first round is the adjacent exchange between the starting and the \( j^{th} \) point, and the second round is the adjacent exchange between the end and the \((j+1)^{th} \) point.

If the value of \( Gfar \) is extremely small after several NN operations, the NN of the subsequent quantum gates cannot be well realized by this inserting method.
**Definition 8:** Preferred number defines the number of selected records that meet the preferred threshold.

For interaction routing records in the priority queue, we kept the first \( k \) preferred interaction routing records that satisfied the threshold \( v \), and the others were pruned. For interaction routing records in the priority queue, we kept the first \( k \) preferred interaction routing records that satisfied the threshold \( v \), and the others were pruned.

3) **NN REALIZATION OF A NON-NN QUANTUM CIRCUIT**

The NN realization of a non-NN quantum circuit inserts the SWAP gates by maintaining a priority queue, and the definition of the priority is the same as that in Algorithm 3: the less the number of \( NNcount \), the farther the reachable quantum gate and the smaller the quantum cost, the higher the priority of the interaction routing record.

**Algorithm 3:** Get Priority Queue of Interaction Routing Records for a Non-NN Gate

**Input:** non-NN gate \( g_i \), current qubit placement \( grid[siz][siz] \)

**Output:** priority queue of interaction routing records \( pq \)

1. \( np = nPath(pos(g_i, c), pos(g_i, t)) \), \( nt = nTrace(pos(g_i, c), pos(g_i, t)) \)
2. \( NNcount \leftarrow NNcount + 1 \)
3. \( QCextra \leftarrow QCextra + (nt-1) \times 3 \)
4. \( j \leftarrow 1 \)
5. while \( i + j < circuit.\ length \) do
6.   if \( g_i \) and \( g_{i+j} \) satisfy the exchange rule then
7.     \( g_{i+j} \) is marked executed
8.   else
9.     go to Line 13
10. end
11. \( j \leftarrow j+1 \)
12. end
13. calculate \( PlacementList(np \times nt)[siz][siz] \) of \( g_i \) by Algorithm 2
14. \( GfarArr[np \times nt]={} \)
15. \( k \leftarrow 0 \)
16. while \( k < np \times nt \) do
17.   \( GfarArr[i] \leftarrow Gfar \) for \( PlacementList[k][siz][siz] \)
18.   \( pq.push(\text{PlacementList}[k][siz][siz], GfarArr[k], QCextra, \text{PlacementList}[k][siz][siz]) \)
19.   \( k \leftarrow k+1 \)
20. end
21. return \( pq \)

**Algorithm 4:** Circuit Level Interaction Routing Algorithm

**Input:** initial qubit placement, quantum circuit, preferred threshold \( v \), preferred number \( k \)

**Output:** the minimum quantum cost

1. \( pq.push(\text{records for the first non-NN gate according to algorithm 3}) \)
2. while \( pq \) is not empty do
3.   \( rd \leftarrow pq.top() \)
4.   if \( rd.Gfar = \text{circuit.length} \) then
5.     go to Line 14
6.   end
7.   if \( rd.Gfar \geq v \) then
8.     \( pq \leftarrow \text{preferred interaction routing records} \)
9.   end
10. \( rd \leftarrow pq.pop() \)
11. \( pq.push(\text{all routing records of } rd.Gfar \text{ according to algorithm 3}) \)
12. end
13. return the minimum quantum cost of final NN circuit

i. Interaction routing records for the first non-NN gate are pushed into the priority queue \( pq \) at first
ii. If the record with the highest priority \( rd \) has completed all the quantum interaction, the optimal quantum cost is returned
iii. If \( rd.Gfar \geq v \), preferred interaction routing records are selected
iv. All routing records of \( rd.Gfar \) (next non-NN gate to be dealt with) are pushed into \( pq \)

Firstly, the initial qubit placement based on algorithm 1 is taken as the initial grid layout, and algorithm 3 is called to obtain all interaction routing for the first non-NN quantum gate, and the corresponding interaction routing records are generated and pushed to the priority queue. Then, as long as the priority queue is not empty, the highest priority record is taken out from the queue. It is judged whether all quantum interactions have been completed according to its reachable quantum gate. If so, the algorithm ends, and the interaction routing record that has completed all NN operations and has the minimum quantum cost in the priority queue is outputted. If not, it is determined whether the reachable quantum gate of the current record meets the preferred threshold, and if so, the interaction routing records are pruned. Then algorithm 3 is called to obtain the reachable quantum gate \( Gfar \) for the current record, thus new interaction routing records are generated and pushed to the priority queue. The above steps are repeated until the priority queue is empty or the first element
of queue has realized the NN interaction. The process is as Algorithm 4.

D. EXAMPLE FOR LOCAL INSERTION OF SWAP GATES

For the quantum circuit as Fig. 2, the initial qubit layout shown in Fig.6 is obtained. The gates \(g_0\) and \(g_1\) satisfy the NN constraint, \(g_2\) acts on the \(q_2\) and \(q_0\), so the NN operation is required. Call algorithm 2 to obtain all interaction routing information, because the Manhattan distance between \(q_0\) and \(q_2\) is 2, there are two insertion paths and two insertion traces for each path. Fig. 8 shows all interaction routing records generated by four ways.

For the first two ways, not only the NN operation of \(g_2\) is completed, but also this way can ensure that \(g_3\) and \(g_4\) meet the NN constraint. Therefore, we only need to consider the NN conversion from \(g_5\), whereas for the latter two ways, only \(g_2\) can satisfy the NN constraint, and the quantum gate NN problem needs to be considered from \(g_3\). In these four ways, all need to insert a SWAP gate, resulting in an extra 3 quantum cost, but apparently at the same quantum cost, the first two are able to accomplish more quantum interactions with more advantages.

The first record (1,5,3) in Fig. 8 is selected. For this strategy, the NN operation of \(g_5\) needs to be carried out. \(g_5\) acts on \(q_1\) and \(q_2\), algorithm 3 is called to generate interaction routing records. Fig. 9 lists all routing selection records to realize the NN of \(g_5\) gate in the circuit.

At this time, there are seven ways in the priority queue, the rest three cases in Fig. 6 plus four cases in Fig.9. Because there are several records with the NNcount 1 and 2 in the priority queue at the same time, the records with small number of NNcount are preferred. Due to space constraints, the placement history will no longer be shown in the priority queue below. Take out the second record with the NNcount
of 1, Gfar of 5 and call algorithm 2 to generate new routing records and push them to the priority queue, and the updated priority queue shown in Fig.10(a). The red shaded part is the extracted record, and the green part is the next routing record obtained by the quantum gate in the record after the NN operation is completed.

Then the record with Gfar as 3 can be reached by taking out NNcount of 1, and algorithm 2 is called to generate new routing selection records and push them to the priority queue. With this loop, the priority queue shown in Fig. 10(b)-(e) is obtained.

After a large number of interaction routing search and local optimal selection, all the routing selection records in the priority queue have completed 3 NN operations, and the green part records have completed all quantum interactions. Therefore, the minimum quantum cost in the green part record is selected as the optimal solution of the algorithm, the minimum quantum cost is 9, and a total of three SWAP gates are inserted.

V. EXPERIMENTAL RESULTS
The performance of the initial qubit placement algorithm and the local SWAP insertion strategies proposed in this paper were tested. The experiments are based on RevLib benchmark dataset [27]. The experimental environment CPU is Intel (R) Xeon (R) Gold 5115 CPU @ 2.40GHz, memory 64GB, Ubuntu 16.04 operating system, C++ programming language. The value of preferred number has been set to 10, indicating that the 10 records with the highest priority are retained in the priority queue. If the reachable quantum gate number Gfar of the interaction routing record is very small after several NN operations, it is shown that the insertion method of the SWAP gates cannot have positive effect on the subsequent quantum gates and it can be discarded.

In this paper, the performance of the algorithm is evaluated by the number of extra added SWAP gates(nSWAP) and the quantum cost(qc) of the quantum circuit after NN operations. nq represents the number of logical qubits, ng is the number of fundamental quantum gates in the original circuit, and the quantum cost is: \( qc = nq + 3 \times nSWAP \). The results compared to [22] and [23] are shown in TABLE 1.

A. IMPROVEMENTS IN SWAP GATES AND QUANTUM COST
A total of 26 benchmark circuits are tested, and the maximum number of gates up to 4452. The experimental results show...
that the number of SWAP gates of this method on most datasets is smaller than that of the method in [22] except on 4gt12-v1_89, and only half the number of SWAP gates need to be inserted on some test data, for example on aj-e11_165, cycle10_2_110, hwb5_55, hwb7_62, rd53_135. The last two columns give the optimization rate of the proposed method in terms of quantum cost, and the average optimization rate compared with [22] is 21.76%.

Compared with the method in [23], the results of the optimization have been obtained on almost all of the tested data, except for 4gt12-v1_89 and hwb4_52, they have equal results with our scheme. In this case, the average optimization rate is 17.23%.

The two sets of comparative data show that the proposed method can reduce the number of SWAP gates inserted and thus reduce the quantum cost of the NN circuit.

### B. SCALABILITY OF PROPOSED METHOD

Experiments are also carried on some larger benchmarks, such as cycle10_2_110, hwb7_62, sym9_148, the gate numbers are 1224, 2683 and 4452 respectively, so as to verify the scalability of the proposed method.

Compared with [22], the number of inserted SWAP gates on three circuits is decreased by 312, 782, 1957 respectively, and the optimization rate in quantum cost is 31.33%, 32.66% and 45.80%. Compared with [23], the number of inserted SWAP gates on three circuits is decreased by 322, 574, 1233 respectively, and the optimization rate in quantum cost is 32.01%, 26.25% and 34.74%. The experimental results show that the proposed method has achieved obvious improvement on these data, especially on sym9_148 with the largest number of gates, and the maximum optimization rate is up to 45.80%, which shows that our method has better applications on larger-scale circuits.

### VI. CONCLUSION

An initial qubit placement method based on the constructed interaction cost model is given. Moreover, gate level and circuit level interaction routing policies with efficient preferred technology are proposed. Experimental results show that the...
proposed methods have lower quantum cost on most test data than existing methods, especially on some large-scale benchmarks, which means that this method is meaningful in dealing with the mapping problem of non-NN quantum circuits. However, for some more large-scale quantum circuits, the runtime will be greatly increased, and the further optimization method also needs to be studied.

REFERENCES

[1] Y. Cai, X. Lu, and N. Jiang, “A survey on quantum image processing,” Chin. J. Electron., vol. 27, no. 4, pp. 718–727, Jul. 2018.
[2] T. Okamoto, X. B. Chen, D. Yun, and Y. X. Yang, “Quantum public-key cryptosystems,” Int. J. Theor. Phys., vol. 51, no. 3, pp. 912–924, Mar. 2011.
[3] J. Li, N. Li, Y. Zhang, S. Wen, W. Du, W. Chen, and W. Ma, “A survey on quantum cryptography,” Chin. J. Electron., vol. 27, no. 2, pp. 223–228, 2018.
[4] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” Nature, vol. 549, pp. 195–202, Sep. 2017.
[5] C. Zalka, “Grover’s quantum searching algorithm is optimal,” Phys. Rev. A, Gen. Phys., vol. 60, no. 4, pp. 2746–2751, Dec. 1999.
[6] H. Häffner, W. Hänsel, C. F. Roos, J. Bemhle, D. Chek-al-kar, M. Chwalla, T. Körber, U. D. Rapol, M. Kiebe, P. O. Schmidt, C. Becher, O. Gühne, W. Dürr, and R. Blatt, “Scalable multiparticle entanglement of trapped ions,” Nature, vol. 438, no. 7068, pp. 646–663, Dec. 2005.
[7] C. Monroe and J. Kim, “Scaling the ion trap quantum processor,” Science, vol. 339, no. 6124, pp. 1164–1169, 2013.
[8] S. N. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, “Layered architecture for quantum computing,” Phys. Rev. X, vol. 2, no. 3, Sep. 2012, Art. no. 031007.
[9] A. D. Córdcoes, E. Magesan, S. J. Srinivasan, A. W. Cross, M. Steffen, J. M. Gambetta, and J. M. Chow, “Demonstration of a quantum error detection code using a square lattice of four superconducting qubits,” Nature Commun., vol. 6, no. 1, p. 6979, Apr. 2015.
[10] M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, CA, USA: W. H. Freeman Co., 1979, pp. 17–38.
[11] Y. Hirata, M. Nakanishi, S. Yamasaki, 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, Jan. 2011.
[12] 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, Dec. 2014.
[13] R. Wille, A. Lye, and R. Drechsler, “Optimal SWAP gate insertion for nearest neighbor quantum circuits,” in Proc. 19th Asia South Pacific Design Autom. Conf. (ASP-DAC), Jan. 2014, pp. 489–494.
[14] 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, Oct. 2013.
[15] A. Kole, K. Datta, and I. Sengupta, “A heuristic for linear nearest neighbor realization of quantum circuits by SWAP gate insertion using -gate lookahead,” IEEE J. Emerg. Sel. Topics Circuits Syst., vol. 6, no. 1, pp. 62–72, Mar. 2016.
[16] A. Shafaei, M. Saeedi, and M. Pedram, “Optimization of quantum circuits for interaction distance in linear nearest neighbor architectures,” in Proc. 50th Annu. Design Autom. Conf. (DAC), May 2013, p. 41.
[17] X. Y. Cheng, G. Zhi-Jin, X. Hai, T. Ying-Ying, and L. Yang, “The nearest neighbor arrangement of quantum circuits based on MCT reversible circuits,” Acta Electronica Sinica, vol. 46, no. 8, pp. 1891–1897, 2018.
[18] C. H. N. G. Xueyun, G. U. A. N. Zhimin, D. I. N. G. Weiping, and Z. H. F. Pengcheng, “Linear nearest neighbor circuit quantum circuit synthesis and optimization based on the matrix,” Acta Electronica Sinica, vol. 46, no. 3, pp. 688–694, 2018.
[19] 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, Tokyo, Japan, Jan. 2015, pp. 178–183.
[20] C.-C. Lin, S. Sur-Kolay, and N. K. Jha, “PAQCS: Physical design-aware fault-tolerant quantum circuit synthesis,” IEEE Trans. Very Large Scale Integ. (VLSI) Syst., vol. 23, no. 7, pp. 1211–1234, Jul. 2015.
[21] D. Ruffinelli and B. Barán, “Linear nearest neighbor optimization in quantum circuits: A multiobjective perspective,” Quantum Inf. Process., vol. 16, no. 9, p. 220, Sep. 2017.
[22] 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., Dec. 2015, pp. 95–100.
[23] A. Kole, K. Datta, and I. Sengupta, “A new heuristic for N-dimensional nearest neighbor realization of a quantum circuit,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 37, no. 1, pp. 182–192, Jan. 2018.
[24] M. Y. Siraichi, F. M. Q. Pereira, V. Dos Santos, and C. Collange, “Qubit allocation,” in Proc. Anais do Concurso de Teses e Dissertações da SBC (CTD-SBC), Jan. 2020, pp. 3–125.
[25] M. Nielsen and I. Chuang, Quantum Computation and Quantum Information. Cambridge, U.K.: Cambridge Univ. Press, Oct. 2000.
[26] D. Maslov, G. W. Dueck, and D. M. Miller, “Simplification of toffoli networks via templates,” in Proc. 16th Symp. Integr. Circuits Syst. Design,SBCCI, Sep. 2003, pp. 53–58.
[27] 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), May 2008, pp. 220–225.

XUEYUN CHENG received the B.S. degree in computer science from Nanjing Normal University, Nanjing, China, in 2001, and the M.S. degree in computational mathematics from Nanjing Normal University, in 2007.

She is currently an Associate Professor with the School of Information Science and Technology, Nantong University, China. Her current research interests include reversible logic synthesis, quantum information, and quantum computation.

ZHIJUN GUAN received the B.S. degree in mathematics from Harbin Normal University, China, in 1986, and the Ph.D. degree in computer application technology from the Nanjing University of Aeronautics and Astronautics, China, in 2005.

He is currently a Professor with the School of Information Science and Technology, Nantong University, China. His current research interests include quantum circuit logic design, secure computing, information-aware and computer-aided intelligent manufacturing, and computer-aided design of integrated circuits and systems.

PENGCHENG ZHU (Student Member, IEEE) received the B.E. and M.E. degrees in computer science from Liaoning Technical University, Fuxin, China, in 2004 and 2007, respectively.

He is currently pursuing the Ph.D. degree with the Department of Information and Communication Engineering, Nantong University, China. He is currently a Lecturer with the Department of Computer Science and Technology, Suzhou College, China. His current research interests include reversible logic synthesis, quantum logic design, and computer-aided design of integrated circuits and systems.