Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2019, 7, 2796-2810
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

# The Mapping and Optimization Method of Quantum Circuits for Clifford + T Gate

**Xianya He[1], Zhijin Guan[1,2*], Fei Ding[1]**

[1]School of Information Science and Technology, Nantong University, Nantong, China
[2]Jiangsu Key Laboratory of Asic Design, Nantong University, Nantong, China
Email: *guan.zj@ntu.edu.cn

**How to cite this paper:** He, X.Y., Guan, Z.J. and Ding, F. (2019) The Mapping and Optimization Method of Quantum Circuits for Clifford + T Gate. *Journal of Applied Mathematics and Physics*, **7**, 2796-2810.
https://doi.org/10.4236/jamp.2019.711192

**Received:** October 15, 2019
**Accepted:** November 10, 2019
**Published:** November 13, 2019

## Abstract

In order to solve the fault tolerance and reliability problems of quantum circuit, a series of structural equivalence rules and optimization operation strategies of quantum circuit are proposed to minimize the number of T gates, increase T gate depth, minimize circuit level, reduce fault tolerance implementation costs and increase circuit reliability. In order to satisfy the nearest neighbor constraints of some quantum systems, a LNN (linear nearest neighbor) arrangement algorithm based on Clifford + T gate quantum circuit is presented. Experiments are done on some benchmarks of RevLib, the results show that the optimization rate of most functions and the running time of the algorithm are better than those of the existing literature.

## Keywords

Quantum Circuit, Clifford + T Circuit, Quantum Cost

## 1. Introduction

Quantum circuits are an important model of quantum computing. The integration and optimization of quantum circuits is of great significance [1] [2] [3]. In recent years, the Clifford + T gates [4] [5] [6] have been used in some typical quantum circuits. Due to the importance of fault tolerance in quantum computing [7] [8], and the fault-tolerant implementation cost of T gates may exceed the implementation cost of Clifford Gate by 100 times or more [4]. Therefore, minimizing the number of T gates is critical to optimizing the T depth of a quantum circuit.

Due to the limitation of quantum techniques, it is required that the control bit and target bit of the 2-qubit gates are physically adjacent, that is the Linear Nearest Neighbor (LNN) constraint required to be considered [9] [10] [11]. This

paper considers the mapping of NCV circuits to equivalent circuits composed of Clifford + T gates, especially optimizing T-count and T-depth, reducing circuit depth, satisfying the constraints of certain quantum architecture, reducing the cost of fault tolerance quantum circuit, and increasing circuit reliability.

## 2. Background

### 2.1. Quantum Gate and Quantum Circuit

The basic unit of operation in a quantum system is a qubit, which is similar to a bit in classical computer system. Qubits can represent states 0 and 1, represented by the symbols |0> and |1> respectively. Qubits can also represent an infinite number of state vectors $|\varphi\rangle$ (called quantum superposition states) between 0 and 1, expressed as:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where $\alpha$ and $\beta$ are complex numbers and satisfy the condition $|\alpha|^2 + |\beta|^2 = 1$.

The operation of the qubit is equivalent to superimposing a unitary matrix U on the state vector of the qubit. The logic gates that operate on qubits in quantum circuits are called quantum gates [12], and each quantum gate can be represented by a $2^n$-order unitary matrix, where $n$ represents the number of qubits.

A quantum circuit cascaded by quantum gates is called a quantum circuit. Some specific quantum gates that make up a quantum circuit are called quantum gate libraries [13]. The NCV gate library contains quantum gates such as NOT, CNOT, V, and $V^+$ [9]. The Clifford + T gate library includes quantum gates such as NOT, CNOT, H, S, $S^+$, T, and $T^+$ which is shown in **Table 1**. The circuit cascaded only by Clifford + T quantum gates is called Clifford + T circuit. The Clifford + T gate libraries are adopted by many quantum physics architectures [14].

The one-dimensional $n$-qubit circuit has $n$ horizontal lines, respectively representing $n$ quantum bit lines, which are sequentially recorded as $l = \{l_1, l_2, \ldots, l_n\}$ from top to bottom. The position of the left to right quantum gate in the line (can be regarded as a vertical line from left to right) indicates the time sequence of the line execution, which is recorded as $h = \{h_1, h_2, \ldots, h_m\}$. **Figure 1** is an example of representation of quantum circuit. There $l = \{1, 2, 3, 4\}$ and $h = \{1, 2, 3, \ldots, 21\}$.

### 2.2. Quantum Gate Decomposition

In general, quantum algorithms can be described by reversible circuits of MCT reversible logic gate cascades of multiple (single) control bit(s)/multiple (single) target bit(s). In order to map a reversible circuit to a quantum system for computation, it is necessary to decompose the logic gates in the reversible circuit. It can be seen from the literature [15] and [16] that a two control bits Toffoli gate is decomposed into a circuit composed of NCV gate library quantum gates as shown in **Figure 2**. AV gate can be decomposed into seven gates as shown in

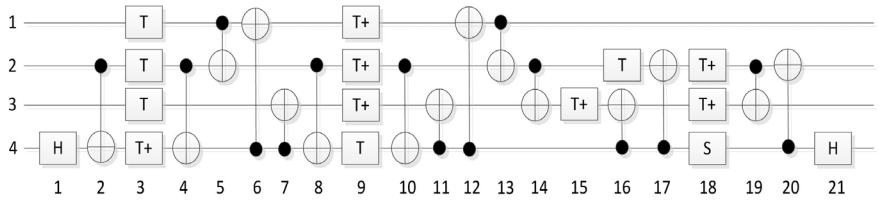


**Figure 1.** A example of representation of quantum circuit.

**Table 1.** Clifford + T gate library.

| type | symbol | graph | matrix |
|---|---|---|---|
| NOT | N | | $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ |
| CNOT | C | | $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$ |
| Hadamard | H | H | $\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ |
| T gate | T | T | $\begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{i\pi}{4}} \end{bmatrix}$ |
| T gate$^{-1}$ | $T^{+}$ | T+ | $\begin{bmatrix} 1 & 0 \\ 0 & e^{\frac{-i\pi}{4}} \end{bmatrix}$ |
| Phase | S | S | $\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$ |
| Phase$^{-1}$ | $S^{+}$ | S+ | $\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$ |

**Figure 3(a)**. A $V^{+}$ gate can be decomposed into seven gates as shown in **Figure 3(b)**, and the equivalent circuit can be obtained by further simplification as shown in **Figure 3(c)**.

## 2.3. Clifford + T Circuit Structure

**Definition 1:** In a one-dimensional quantum circuit, a sequence of quantum gates that can be operated in parallel is called a circuit level. If two or more quantum gates can be combined together in a circuit, their qubits can operate in parallel without disjoint, and these quantum gates are said to form a grouped.

**Definition 2:** In a one-dimensional circuit, the circuit depth is the number of levels in the circuit.

**Definition 3:** The $T$-depth of the Clifford + T circuit is the number of T or $T^{+}$ gates contained on different qubit lines in one level of the circuit.

**Definition 4:** A "CNOT + $T(T^{+})$ + CNOT" structure is a gate group by the consisting of two CNOT gates and one $T$ or $T^{+}$ gate, called the CTC-structure.

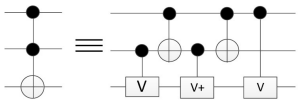


**Figure 2.** Decomposition of Toffoli gate.

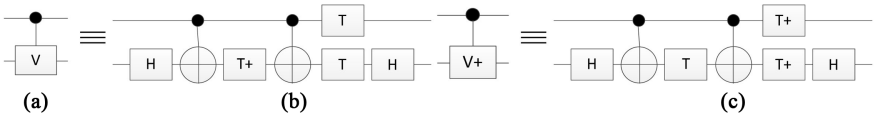


**Figure 3.** Decomposition of V gate and V$^+$ gate. (a) Decomposition of V gate; (b) Decomposition of V$^+$ gate; (c) Equivalent circuit for the (b) reduction.

The two control bits and the two target bits of the two CNOT gates are on the same qubit line, and the $T$ (or $T^+$) gate is between the two target bits.

**Figure 4(a)** and **Figure 4(b)** are represented as CTC ($T$) and CTC ($T^+$) respectively for ease of use. Where line $l_a$ is called the control bit line of the CTC-structure, and line $l_b$ is called the target bit line of CTC-structure. If the control bit line of the CTC-structure is line $i$, the target bit line is line $j$, and the two CNOT gates are located on $h_1$ and $h_2$ respectively ($h_1 = h_2 - 2$), then the CTC-structure can be expressed as $CTC_{h_1,h_2}(i,j)$.

**Definition 5:** The depth of the CTC-structure refers to the number of $T$ (or $T^+$) gates that can be operated in parallel in the CTC-structure.

**Definition 6:** If the depth of the CTC-structure is equal to the number of the circuit input/output qubits, then this CTC-structure is said to be full.

## 3. Decomposition and Optimization of the Quantum Circuits

In order to optimize quantum circuits, the quantum gates and related sub-line structures in the Clifford + T circuit are analyzed and discussed in this section.

### 3.1. Relevant Properties

The following properties 1 - 3 [5] can be verified by the matrix representation of the quantum gates, which are multiplied by matrices to obtain the results of their interactions.

**Property 1:** (a) Two adjacent $T$-gates are equivalent to a $S$-gate. Two adjacent $T^+$-gates are equivalent to a $S^+$-gate.

**Property 2:** A CNOT gate is equivalent to two consecutive $V$-gates which have same control bit line and target bit line. A CNOT gate is also equivalent to two consecutive $V^+$-gates which have same control bit line and target bit line.

**Property 3:** 1) Two adjacent CNOT gates with control bits on the same qubit line and target bits on the same qubit line can cancel each other out. 2) Two adjacent $H$-gates can cancel each other out. 3) Two adjacent $T$ and $T^+$-gates can cancel each other out. 4) Two adjacent $S$ and $S^+$-gates can cancel each other out.

**Property 4:** The combinations of two adjacent 2-qubit gates are equivalent, if the control bits are on the same qubit line, and their combinations are switched [5]. Analogously, the combinations of two adjacent 2-qubit gates are equivalent,

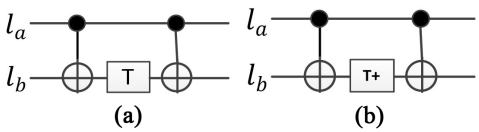


**Figure 4.** Two forms of CTC-structure. (a) CTC(T) structure; (b) CTC ($T^+$) structure.

if the target bits are on the same qubit line, and their combinations are switched.

**Property 5:** If the $T$-gate (or $T^+$-gate) and the 2-qubit gate control bit are on the same qubit line and adjacent to each other, then the combination of the $T$-gate (or $T^+$-gate) and the 2-qubit gate on the left and right sides of the control bit of the 2-qubit gate are equivalent [5].

**Property 6** [5]: For two CTC($T$) structures, if the control line of a CTC($T$) structure is the target line of the other CTC($T$) structure, then the two CTC($T$) structures are equivalent. Analogously, For two CTC($T^+$) structures, if the control line of a CTC($T^+$) structure is the target line of the other CTC($T^+$) structure, then the two CTC($T^+$) structures are equivalent.

**Property 7** [5]: The combination of the 2-qubit gate and the CTC($T$) structure are equivalent for the right and the left of CTC($T$) structure, if the control bit of the 2-qubit gate is on the target bit line of the CTC($T$) structure and the target bit of the 2-qubit gate is not on the line crossed by CTC($T$). The combination of the 2-qubit gate and the CTC($T^+$) structure are equivalent for the right and the left of CTC($T^+$) structure, if the control bit of the 2-qubit gate is on the target bit line of the CTC($T^+$) structure and the target bit of the 2-qubit gate are not on the line crossed by CTC($T^+$).

**Property 8:** The subcircuit of the CNOT gate combination is equivalent as shown in **Figures 5(a)-(c)**.

The equivalence of the above subcircuits can be easily verified by the truth table.

**Conclusion 1:** The combination of two adjacent $T$-gates and 2-qubit gate are equivalent on the right and the left of control bit of this 2-qubit gate.

The combination of two adjacent $T^+$-gates and 2-qubit gate are equivalent on the right and the left of control bit of 2-qubit gate.

The above inferences are readily available based on property 5.

Generally, the arbitrary combination of $m$ ($m \geq 1$) 2-qubit gates and $n$ ($n \geq 1$) $T$-gates (or $T^+$-gates) are equivalent, if the control bits of $m$ ($m \geq 1$) 2-qubit gates and $n$ ($n \geq 1$) $T$-gates (or $T^+$-gates) are on the same qubit line.

**Conclusion 2:** According to property 1, the S-gate (or $S^+$-gate) distributed on the left and right sides of the control bit is equivalent for a combination of an S-gate (or $S^+$-gate) on the control bit line of a two-qubit gate and the two-qubit gate. It can be seen from the conclusion 1.

**Conclusion 3:** The combination of single quantum gate ($T$, $T^+$, $S$, $S^+$) located on the left and right side of CTC structure is equivalent if the single quantum gate is on the same quantum bit line of CTC-structure.

**Theorem 1:** 1) Any combinations of the CTC-structure with the $T$-gates on

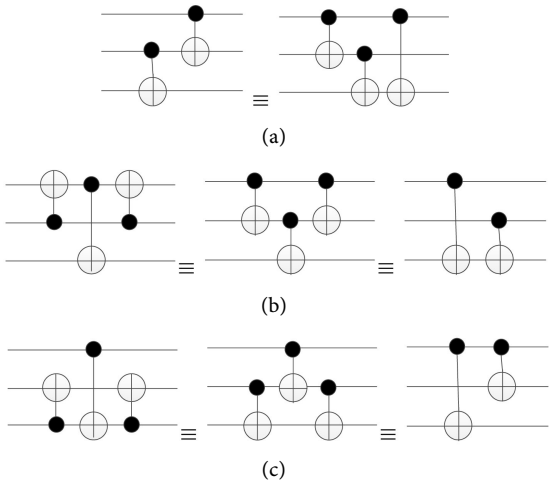

**Figure 5.** Equivalent variations of CNOT gate sequence. (a) The first equivalent case; (b) The second equivalent case; (c) The third equivalent case.

its control bit line and target bit line are equivalent. 2) Any combinations of the CTC-structure with the $T^+$-gates on its control bit line and target bit line are equivalent.

**Proof:** As long as proof 1), then 2) is available as the same.

1) According to the Property 6, the CTC-structure interchange control bit line with target bit line is equivalent, so the combination of a $T$ gate and a CTC-structure on the same qubit line is also equivalent. As show in **Figure 6** that **Figure 6(a)** is equivalent to **Figure 6(b)**;

2) According to the Property 5, the T gate of **Figure 6(b)** can be moved to the right side of the first CNOT gate. Similarly, it can be moved to the right side of the second CNOT gate to get **Figure 6(c)**. As show in **Figure 6** that **Figure 6(b)** is equivalent to **Figure 6(c)**;

3) According to the Property 6, the CTC-structure interchange control bit line with target bit line is equivalent, therefore, the combination of the CTC-structure and the T gate on the same bit line is also equivalent. As show in **Figure 6** that **Figure 6(c)** and **Figure 6(d)** are equivalent.

QED ■

**Corollary 1:** The combination of $S$-gate (or $S^+$-gate) and CTC-structure is equivalent when $S$-gate (or $S^+$-gate) is on the left or right sides of the target position of the CTC-structure according to property 1 and theorem 1.

**Theorem 2:** In an 1-dimensional quantum circuit, let $V' = \{V, V^+\}$, if there are consecutive $mV$-gates and $nV^+$-gates, the control bit line of $V'$ and the target bit line of $V'$ are same respectively, then the target bit line of the $m+nV'$-gates can decompose into sub-circuit structure and the number of quantum gates are:

$$\begin{cases} H+\Sigma+H, & |m-n| \text{ is even} \\ H+\Sigma+T'+H, & |m-n| \text{ is odd} \end{cases} \tag{2}$$

the control bit line of this $m+nV'$-gates can decompose into sub-circuit structure and the number of quantum gates are:

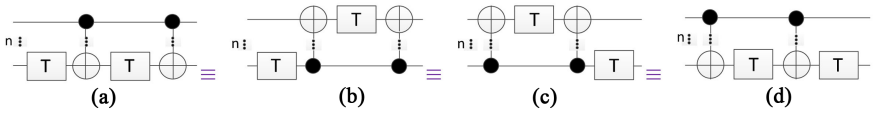


**Figure 6.** Equivalent exchange of $T$ and CTC-structures. (a) Original circuit; (b) Circuit after CTC-structure flip; (c) Circuit after $T$ gate movement; (d) Final circuit.

$$\begin{cases} \Sigma, & |m-n| \text{ is even} \\ \Sigma + T', & |m-n| \text{ is odd} \end{cases} \tag{3}$$

where $T' = \{T, T^+\}$, $S' = \{S, S^+\}$,

$$\Sigma = m * \mathrm{CTC}\left(T^+\right) + n * \mathrm{CTC}\left(T\right) + \left\lfloor \frac{|m-n|}{2} \right\rfloor * S'.$$

Proof: In an 1-dimensional quantum circuit, it is assumed that there are $k$ quantum bit lines $l$ (*i.e.* the input/output of the quantum line is $k$), that is, $L_1 = \{l_1, l_2, \ldots, l_k\}$. If there are consecutive $m$ target bits of $V$-gates and $n$ target bits of $V^+$-gates on the $i$-th quantum bit line, $i \in \{1, 2, \ldots, k\}$. Let $V' = \{V, V^+\} = < V_1', V_2', \ldots, V_m', V_{m+1}', \ldots, V_{m+n}' >$, then:

The target bit line of $V$-gate in $V'$ can be decomposed into:

$$H + \mathrm{CTC}\left(T^+\right) + T + H \tag{4}$$

The target bit line of $V^*$-gate in $V'$ can be decomposed into:

$$H + \mathrm{CTC}\left(T\right) + T^+ + H \tag{5}$$

It can be seen that when $m, n \geq 1$, continuous $m + n$ $V'$-gates are decomposed, and $m + n - 1$ pair of $H$-gates are adjacent on the target bit line of CTC-structure. In (1) and (2), the first $H$-gate and the last $H$-gate are the remaining, and there are $m + n$ CTC-structures and $m$ $T$-gates and $n T^*$-gates between this two $H$-gates. By the theorem 1, these $m T$-gates and $n T^*$-gates can be moved so that they are adjacent. So there have:

1) When $m > n$, there are $n$ $T$-gates and $n T^*$-gates are eliminated and $\lfloor (m - n)/2 \rfloor$ pairs of $T$-gates are replaced by $\lfloor (m - n)/2 \rfloor$ $S$-gates. If $m$-$n$ is an even number, all $T$ gates are replaced; if $m$-$n$ is an odd number, then remain a $T$-gate. The target bit line of $V'$-gate can eventually be decomposed into:

$$\begin{cases} H + m * \mathrm{CTC}\left(T^+\right) + n * \mathrm{CTC}\left(T\right) + \left\lfloor \frac{m-n}{2} \right\rfloor * S + H, & m-n \text{ is even} \\ H + m * \mathrm{CTC}\left(T^+\right) + n * \mathrm{CTC}\left(T\right) + \left\lfloor \frac{m-n}{2} \right\rfloor * S + T + H, & m-n \text{ is odd} \end{cases} \tag{6}$$

2) When $m = n$, all $T$-gates and $T^*$-gates are eliminated. The target bit line of $V'$-gate can eventually be decomposed into:

$$H + m * \mathrm{CTC}\left(T^+\right) + n * \mathrm{CTC}\left(T\right) + H \tag{7}$$

3) When $m < n$, there are $m$ $T$-gates and $m$ $T^*$-gates are eliminated, and $\lfloor (n-m)/2 \rfloor$ pairs of $T^*$-gates are replaced by $\lfloor (n-m)/2 \rfloor$ $S^*$-gates. If $n$-$m$ is an even number, all $T^*$-gates are replaced and if $n$-$m$ is an odd number then re-

maining a $T^+$-gate. The target bit line of the $V'$-gate can be eventually decomposed into:

$$\begin{cases} H + m * \mathrm{CTC}\left(T^+\right) + n * \mathrm{CTC}\left(T\right) + \left\lfloor \dfrac{n-m}{2} \right\rfloor * S^+ + H, & n-m \text{ is even} \\ H + m * \mathrm{CTC}\left(T^+\right) + n * \mathrm{CTC}\left(T\right) + \left\lfloor \dfrac{n-m}{2} \right\rfloor * S^+ + T^+ + H, & n-m \text{ is odd} \end{cases} \tag{8}$$

The control bit of V-gate and $V^+$-gate in $V'$ can be decomposed into the form of $\mathrm{CTC}\left(T^+\right)+T$ and $\mathrm{CTC}\left(T\right)+T^+$ respectively at the control bit line.

The control bit of $m$ $V$-gates and $n$ $V^+$-gates in $V'$ may be distributed on the $L_2 = \{l_1, \ldots, l_{i-1}, l_{i+1}, \ldots, l_k\}$. When $m$, $n \geq 1$, after successive $m+n$ $V'$gates are decomposed, there are a total of $m+n$ CTC-structures, $m$ $T$-gates and $n$ $T^+$-gates on all control bit lines. Among them, the $l_j (l_j \in L_2)$ qubit line has $m_j$ $T$-gates, $n_j T^+$-gates and $m_j + n_j$ CTC-structures, where $m = \sum_{j=1}^{k} m_j$, $n = \sum_{j=1}^{k} n_j$, $j \neq i$.

By theorem 1, these $m_j T$-gates and $n_j T^+$-gates can be moved to be adjacent, so:

1) When $m_j > n_j$, there are $n_j T$-gates and $n_j T^+$-gates are eliminated; $\lfloor (m_j - n_j)/2 \rfloor$ pairs of $T$-gates are replaced by $\lfloor (m_j - n_j)/2 \rfloor$ $S$-gates. If $m_j - n_j$ is an even number, all $T$-gates are replaced, and if $m_j - n_j$ is an odd number, then remain a $T$-gate. The control bit line of the $V'$-gate can be finally decomposed into:

$$\begin{cases} m_j * \mathrm{CTC}\left(T^+\right) + n_j * \mathrm{CTC}\left(T\right) + \left\lfloor \dfrac{m_j - n_j}{2} \right\rfloor * S, & m_j - n_j \text{ is even} \\ m_j * \mathrm{CTC}\left(T^+\right) + n_j * \mathrm{CTC}\left(T\right) + \left\lfloor \dfrac{m_j - n_j}{2} \right\rfloor * S + T, & m_j - n_j \text{ is odd} \end{cases} \tag{9}$$

2) When $m_j = n_j$, all $m_j T$-gates and $m_j T^+$-gates are eliminated. The control bit line of $V'$-gate can be finally decomposed into:

$$m_j * \mathrm{CTC}\left(T^+\right) + n_j * \mathrm{CTC}\left(T\right) \tag{10}$$

3) When $m_j < n_j$, $m_j T$ gates and $m_j T^+$-gates are eliminated, and $\lfloor (n_j - m_j)/2 \rfloor$ pairs of $T^+$-gates are replaced by $\lfloor (n_j - m_j)/2 \rfloor$ $S^+$-gates. If $n_j - m_j$ is an even number, all $T^+$-gates are replaced and if $n_j - m_j$ is an odd number, then remain a $T^+$-gate. The control bit line of the $V'$-gate can be finally decomposed into:

$$\begin{cases} m_j * \mathrm{CTC}\left(T^+\right) + n_j * \mathrm{CTC}\left(T\right) + \left\lfloor \dfrac{n_j - m_j}{2} \right\rfloor * S^+, & n-m \text{ is even} \\ m_j * \mathrm{CTC}\left(T^+\right) + n_j * \mathrm{CTC}\left(T\right) + \left\lfloor \dfrac{n_j - m_j}{2} \right\rfloor * S^+ + T^+, & n-m \text{ is odd} \end{cases} \tag{11}$$

QED■

As shown in **Figure 7(a)**, the circuit has eight 2-qubit gates ($G_1$ to $G_8$ from left to right), where $G_1$, $G_3$, $G_5$, $G_8$ are $V'$ ($V' \in \{V, V^+\}$) gates whose target bits are

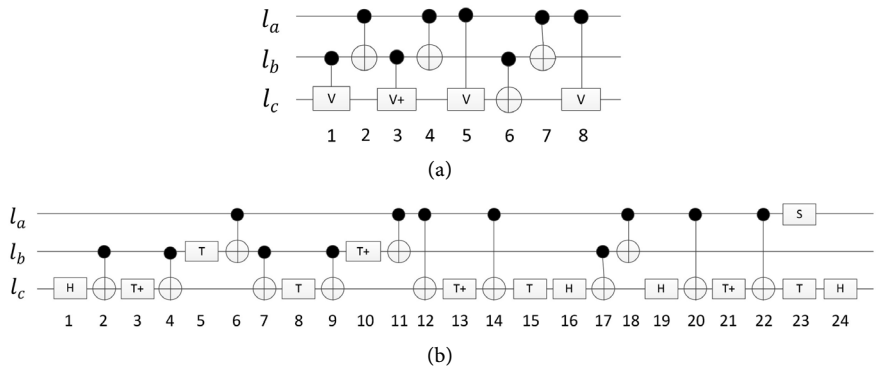


**Figure 7.** Quantum circuit Clifford + T gate decomposition example. (a) Original NCV circuit; (b) Decomposed Clifford + T circuit.

on the same qubit line $l_c$. The circuit after decomposition is shown in **Figure 7(b)** where the second H gate and $T_c$ gate generated by $G_1$ are eliminated respectively with the first H gate and $T_c^+$ gate generated by $G_3$, the second H gate and $T_c^+$ gate generated by $G_3$ are eliminated respectively with the first H gate and $T_c$ gate generated by $G_5$, while the H gate generated by $G_8$ cannot cancel out with the H gate generated by $G_5$ because of the blocking of the $G_6$ target bit. The $T_a$ generated by $G_5$ and $G_8$ respectively can be merged into $S_a$.

### 3.2. Algorithm for Decomposition of NCV Circuit

In order to obtain the quantum circuit composed of Clifford + T gate, using the equivalent circuit given in **Figure 3** and the related theory of theorem 2 to decompose NCV circuit.

The circuit is initially optimized with the relevant properties in the decomposition process. The decomposition algorithm is shown in **Algorithm 1**.

### 3.3. Depth Maximization of the CTC-Structure

In order to deepen the depth of the CTC-structure, reduce the depth of the circuit, improve the parallelism of the circuit, the decomposed Clifford + T circuit need to be structured and the main goal is to make the depth of each CTC-structure equal to the number of qubits (That is to say, make the CTC-structure to be fully occupied). The CTC-structure depth deepening algorithm is shown in **Algorithm 2**.

### 3.4. Depth Optimization of Circuit

After deepening the depth of the CTC-structure of the circuit, there will be many adjacent CNOT gates. The optimization of continuous CNOT gates can reduce the number of gates. **Algorithm 3** is CNOT optimization algorithm.

## 4. Quantum Circuit Synthesis Satisfying Constraints

### 4.1. Some Constraints on Quantum Circuits

In some practical quantum techniques, quantum bit interactions of quantum

*Input*: *NCV circuit*

*Output*: *Clifford* + *T circuit*

*Begin*:

*Step* 1: *Initialize* $count\_v[i] = 0(0 \le i < n), count\_V^+[i] = 0(0 \le i < n)$

*Step* 2: *Read* $G_1 = NCV(C_1, T_1)$ *from the far left of the circuit*

*Step* 3: *If* $G_1 = V$, *then* $count\_v[C_1] + +$, $count\_v[T_1] + +$;

*Step* 4: *If* $G_1 = V^+$, *then* $count\_V^+[C_1] + +$, $count\_V^+[T_1] + +$;

*Step* 5: *If* $G_1 = CNOT$ *or* $G_1 = NOT$, *according to theorem 2*, *add* $H$、$T$、$T^+$、$S$、$S^+$ *gates before CNOT or NOT, and make* $count\_V[T_1] = 0$, $count\_V^+[T_1] = 0$;

*Step* 6: *Determine if the circuit ends, if not, continue scanning circuit and back to step* 3

*Step* 7: *Add* $H$、$T$、$T^+$、$S$、$S^+$ *gates at the end of each line according to Theorem* 2, *and make* $count\_V[i] = 0(0 \le i < n)$, $count\_V^+[i] = 0(0 \le i < n)$.

*End*.

**Algorithm 1.** NCV decomposition algorithm.

*Input*: *Clifford* + *T circuit*

*Output*: *Optimized Clifford* + *T circuit*

*Begin:*

*Step* 1: *Starting from the leftmost side of the circuit, find the first CTC-structure* $CTC(l_C, l_T)$ *that is partial* occupied;

*Step* 2: *Determine whether* $CTC(l_C, l_T)$ *satisfies that there have no T* (*or* $T^+$ *or S or* $S^+$) *gate on line* $l_C$ *which can be moved to* $CTC(l_C, l_T)$ *but have T* (*or* $T^+$ *or S or* $S^+$) *gates on line* $l_T$ *which can be moved to* $CTC(l_C, l_T)$, *if satisfied then flip*;

*Step* 3: *Move the gate to* $CTC(l_C, l_T)$ *if it can be moved to or be cross-domain movement to* $CTC(l_C, l_T)$;

*Step* 4: *Determine whether have CTC-structures can flip over to CTC* $(l_C, l_T)$ *on the line* $l_T$, *if have such CTC-structures, and has no CTC-structures can flip over to CTC* $(l_C, l_T)$ *on line* $l_C$, *no gate* (*S, S+, T,* $T^+$) *can reach* $CTC(l_C, l_T)$ *by movement or cross-domain movement on both* $l_C$ *and* $l_T$, *then flip* $CTC(l_C, l_T)$.

*Step* 5: *Determine whether the* $l_i$ *qubit line of the* $CTC(l_C, l_T)$ *is empty, and if it is empty then add a CTC-structure to the* $CTC(l_C, l_T)$ *which can be moved or be flip moved*;

*Step* 6: *Determine whether the current gate is the last gate of the circuit. If not then continue to scan the circuit and to find the CTC-structure* $CTC(l_C, l_T)$ *that is partial occupied and then return to step* 2; *otherwise, the operation ends*.

*End*.

**Algorithm 2.** CTC-structure depth deepening algorithm.

*Input*: *Optimized Clifford + T circuit*
*Output*: *Further optimized Clifford + T circuit*
*Begin*:
*Step* 1: *Initialize* $temp\_cnot = \emptyset$, $the\_wire = \emptyset$, $new\_gate = \emptyset$.
*Step* 2: *Read from the leftmost side of the circuit.*
*Step* 3: *If there are consecutive* $CNOT_1, CNOT_2, CNOT_3$ *can be merged then make them merged.*
*Step* 4: *If there are continuous* $CNOT_1, CNOT_2$ *which can be decomposed, then put the* $CNOT_{new}(l_{C_{new}}, l_{T_{new}})$ *which generated by the decomposition into new_gate.*
*Step* 5: *If there has no continuous CNOT of step3, step4, but have* $CNOT_{now}((l_{C_{now}}, l_{T_{now}})$ *and in new_gate exists* $CNOT_{new}(l_{C_{now}}, l_{T_{now}})$ *(or* $CNOT_{new}(l_{T_{now}}, l_{C_{now}})$*), then find* $CNOT_1$、$CNOT_2$ *that produces* $CNOT_{new}$ *and decompose them according to* $CNOT_{now}$*. In the original circuit delete* $CNOT_1$、$CNOT_2$ *and insert* $CNOT_1'$、$CNOT_2'$ *which generated by decomposition. In the original circuit delete* $CNOT_{now}$*, and delete* $CNOT_{new}$ *in new_gate.*
*Step* 6: *Determine whether the current gate is the last gate of the circuit. If not then continue to scan the circuit and return to step* 3, *otherwise the operation ends.*
*End.*

**Algorithm 3.** CNOT optimization algorithm.

gates are required to satisfy linear nearest neighbor constraints. In a quantum logic circuit, to exchange the logic values of certain two circuits, it can generally be realized by inserting a SWAP-gate. The SWAP-gate also needs to satisfy the linear nearest neighbor in the nearest neighbor constrained quantum circuit, it is called the nearest neighbor SWAP gate (NNS gate) [7]. An NNS gate is equivalent to three cascades of CNOT gates that satisfy the linear nearest neighbor architecture, as shown in **Figure 8(a)**. Some quantum circuit physical structures require even stronger constraints. For example, IBM QX [17] [18] requires that the CONT gates in different directions be flipped through the *H* gate to be in the same direction (control bits and target bits are on two qubits), as shown in **Figure 8(b)**. Converting a non-LNN Clifford + T circuit into an LNN Clifford + T circuit is usually done by inserting several NNS gates, as shown in **Figure 9**. **Figure 9(a)** is a non-LNN circuit, **Figure 9(b)** is equivalent LNN circuit by insert SWAP gates, **Figure 9(c)** is equivalent form of NNS gate.

## 4.2. Synthesis Algorithm

It can be seen from the literature [1] that the linear nearest neighbor cost of the 2-qubit gate G in a quantum circuit is:

$$G_{nnc} = |l_C - l_T| - 1 \tag{12}$$

where $l_C$ and $l_T$ are the number of the control bit line and the target bit line of *G*, respectively, the nearest neighbor cost of a Clifford + T circuit is:

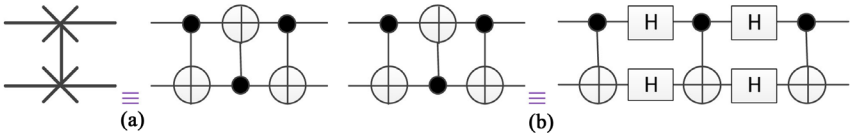


**Figure 8.** Equivalent form of NNS gate. (a) Equivalent form of a SWAP gate; (b) CNOT flips to the same direction.

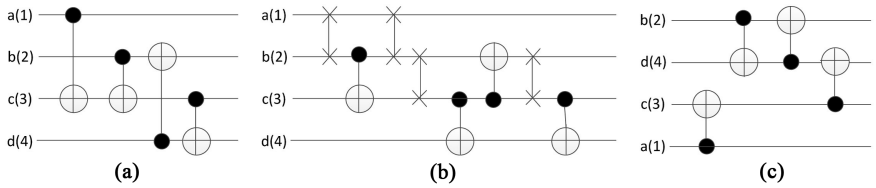


**Figure 9.** (a) Non-LNN circuit; (b) Equivalent LNN circuit by inserting SWAP gates; (c) Equivalent LNN circuit by rearranging the quantum lines.

$$CC' = \sum_{i=1}^{K} G_{nnc} \times \frac{N-i+1}{N}, (1 < K \le N) \tag{13}$$

where $N$ is the number of quantum gates in the Clifford + T circuit and $K$ is the linear nearest neighbor cost calculation coefficient of the circuit.

This paper presents an adaptive Clifford + T circuit neighborhood optimization algorithm, such as **Algorithm 4**. The optimized Clifford + T circuit is scanned from left to right, and all LNN schemes are listed for the first non-LNN 2-qubit gate is encountered. Calculate the $CC'$ of the remaining circuit after each scheme is executed, select the scheme with the smallest $CC'$, insert the NNS gates into the circuit one after the other, then continue to scan the remaining circuits, and iteratively execute the above steps until all the quantum gates reached LNN. The algorithm is executed for all coefficients $K$, and the scheme with the least number of inserted switching gates is recorded for output.

The time complexity of the algorithm is $O(L \times N \times K)$, where $L$ is the quantum number. For small-scale circuits, try all the coefficients $K$ to find the optimal solution. For large-scale circuits, $K$ can be reduced to a constant term in order to reduce algorithm runtime.

## 5. Experiment and Result Analysis

According to all the considerations and methods discussed above, based on RevLib [19] benchmark and the decomposition tool designed by the research team [6], use Intel (R) 64 - 3.2 GHz bit processor, 8 GB RAM, windows 10 operating environment, C++ programming language. Decompose NCV gate library circuit into basic quantum gate circuit of Clifford + T structure, then through neighboring and CNOT gate flipping the relevant constraints comparable to the literature [14] are satisfied, so use the experimental results to compare with the results in [14] (**Table 2**). In order to more fully evaluate the effectiveness of the proposed method, the benchmark function selected in this paper is more extensive. Since the literature [14] only targets the 10 - 16 qubit part of the benchmark

**Table 2.** Experimental results comparison.

| Name | *n* | Ref. [14] | | | Proposed approach | | | Gate optimization rate (%) | Depth optimization rate (%) |
|---|---|---|---|---|---|---|---|---|---|
| | | gate1 | deep1 | time | gate2 | deep2 | time | | |
| mini_alu_305 | 10 | 474 | 225 | 1.25 | 518 | 372 | 0.05 | −9.28 | −65.33 |
| rd73_141 | 10 | 656 | 301 | 1.52 | 478 | 328 | 0.04 | 27.13 | −8.97 |
| sys6−v0_144 | 10 | 613 | 250 | 1.36 | 476 | 326 | 0.04 | 22.35 | −30.4 |
| dc1_220 | 11 | 5946 | 3378 | 12.38 | 3460 | 2622 | 1.14 | 41.81 | 22.38 |
| wim_266 | 11 | 2985 | 1711 | 6.3 | 1738 | 1294 | 0.29 | 41.78 | 24.37 |
| sqrt8_260 | 12 | 9744 | 5501 | 19.66 | 5108 | 3909 | 2.52 | 47.58 | 28.94 |
| sym9_147 | 12 | 955 | 425 | 2.08 | 748 | 518 | 0.10 | 21.68 | −21.88 |
| adr4_197 | 13 | 11,301 | 6205 | 23.17 | 6332 | 4799 | 3.65 | 43.97 | 22.66 |
| squar5_261 | 13 | 6267 | 3448 | 12.96 | 3699 | 2764 | 1.63 | 40.98 | 19.84 |
| pm1_249 | 14 | 5431 | 3013 | 11.1 | 3221 | 2437 | 0.92 | 40.69 | 19.12 |
| 0410184_169 | 14 | 758 | 366 | 1.48 | 572 | 379 | 0.03 | 24.54 | −3.55 |
| cm42a_207 | 14 | 5431 | 3013 | 11.95 | 3221 | 2437 | 1.08 | 40.69 | 19.12 |
| sym6_316 | 14 | 852 | 456 | 1.84 | 1015 | 720 | 0.26 | −19.13 | −57.89 |
| ham15_108 | 15 | 28,310 | | 68.75 | 4756 | 3528 | 3.24 | 83.20 | 77.80 |
| misex1_241 | 15 | 15,185 | 8729 | 33.11 | 8103 | 6239 | 5.81 | 46.64 | 28.53 |
| average value | | 6327.2 | 3527.47 | | 2896.33 | 2178.13 | | 32.97 | 4.98 |

*Input*: *Non-LNN Clifford + T circuit*
*Output*: *LNN Clifford + T circuit*
*Step* 1: *Read the circuit and rearrange the qubits according to the qubit rearrangement rules*.
*Step* 2: *Set the value of K according to the circuit scale*.
*Step* 3: *Scan the circuit from left to right. For the first non-LNN quantum gate, list all neighboring schemes and calculate the* $CC'$ *of the remaining circuits of each scheme. Record the scheme with the smallest value of* $CC'$ *and then insert the relevant NNS gate*.
*Step* 4: *Repeat Step* 3 *until all quantum gates are adjacent*.
*Step* 5: *Repeat Step* 2 - 4 *until all K has been tried*.
*Step* 6: *Output the scheme with the least number of SWAP gates*.

**Algorithm 4.** LNN arrangement algorithm based on quantum weight.

function, the comparison results only compare the partial functions of 10 - 16 qubits. For each benchmark function, the function name (Name), the number of qubits (n), the number of quantum gates (g) of the circuit, the depth of the circuit (d), and the operation time (in seconds) in the corresponding operating environment are provided.

## 6. Conclusion

Due to the limitations of some quantum techniques, there are special requirements for the use of quantum gates in quantum circuits, and the linear nearest neighbor constraints are required for the physical positions of the control bits and target bits of the 2-qubit gates. The main work of this paper is to map the NCV circuit to the equivalent circuit composed of Clifford + T gate, optimize the quantum gate number and T depth, reduce the circuit depth, propose a series of circuit structure equivalence rules and optimization operation strategies. The CNOT gate neighbor algorithm of the Clifford + T gate quantum circuit satisfies the CNOT constraint imposed by the architecture. In the related properties and operation methods proposed in this paper, due to the limitation of H gate, the optimization of the circuit will have a great impact. How to lay out the position of the H gate is an important part of future work research.

## Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

## References

[1] Abdessaied, N. and Drechsler, R. (2018) Reversible and Quantum Circuits Optimization and Complexity Analysis. Springer Nature.

[2] Wille, R., Keszocze, O., Walter, M., Rohrs, P., Chattopadhyay, A. and Drechsler, R. (2016) Look-Ahead Schemes for Nearest Neighbor Optimization of 1d and 2d Quantum Circuits. *Asia and South Pacific Design Automation Conf.*, 292-297. https://doi.org/10.1109/ASPDAC.2016.7428026

[3] http://www.jku.at/iic/eda/ibm_qx_mapping

[4] Biswal, L., Bandyopadhyay, C., Wille, R., *et al.* (2016) Improving the Realization of Multiple-Control Toffoli Gates Using the NCVW Quantum Gate Library. *International Conference on VLSI Design & International Conference on Embedded Systems*. https://doi.org/10.1109/VLSID.2016.23

[5] Miller, D.M., Soeken, M. and Drechsler, R. (2014) Mapping NCV Circuits to Optimized Clifford + T Circuits. Reversible Computation. Springer International Publishing. https://doi.org/10.1007/978-3-319-08494-7_13

[6] Saeedi, M., Wille, R. and Drechsler, R. (2010) Synthesis of Quantum Circuits for Linear Nearest Neighbor Architectures. *Quantum Information Processing*. https://doi.org/10.1007/s11128-010-0201-2

[7] Xueyun, C., Zhijin, G. and Weiping, D. (2018) Mapping from Multiple-Control Toffoli Circuits to Linear Nearest Neighbor Quantum Circuits. *Quantum Information Processing*, **17**, 169. https://doi.org/10.1007/s11128-018-1908-8

[8] Abhari, A.J., Faruque, A., Dousti, M.J., Svec, L., Catu, O., Chakrabati, A., Chiang, C.-F., Vanderwilt, S., Black, J. and Chong, F. (2012) Scaffold: Quantum Programming Language. Technical Report, Princeton Univ Nj Dept of Computer Science.

[9] Wille, R., Lye, A. and Drechsler, R. (2014) Exact Reordering of Circuit Lines for Nearest Neighbor Quantum Architectures. *IEEE Trans. on CAD of Integrated Circuits and Systems*, **33**, 1818-1831. https://doi.org/10.1109/TCAD.2014.2356463

[10] Shafaei, A., Saeedi, M. and Pedram, M. (2013) Optimization of Quantum Circuits for Interaction Distance in Linear Nearest Neighbor Architectures. *Design Automation Conf.*, 41-46. https://doi.org/10.1145/2463209.2488785

[11] Cross, A.W., Bishop, L.S., Smolin, J.A. and Gambetta, J.M. (2017) Open Quantum Assembly Language. arXiv Preprint arXiv:1707.03429.

[12] Barenco, A., Bennett, C.H., Cleve, R., *et al*. (1995) Elementary Gates for Quantum Computation. *Physical Review A*. https://doi.org/10.1103/PhysRevA.52.3457

[13] Green, A.S., Lumsdaine, P.L., Ross, N.J., Selinger, P. and Valiron, B. (2013) Quipper: A Scalable Quantum Programming Language. *Conference on Programming Language Design and Implementation*, 333-342.

[14] Zulehner, A., Paler, A. and Wille, R. (2018) An Efficient Methodology for Mapping Quantum Circuits to the IBM QX Architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 1. https://doi.org/10.23919/DATE.2018.8342181

[15] Barenco, A., Bennett, C.H., Cleve, R., *et al*. (1995) Elementary Gates for Quantum Computation. *Physical Review A*, **52**, 3457. https://doi.org/10.1103/PhysRevA.52.3457

[16] Soeken, M., Miller, D.M. and Drechsler, R. (2013) Quantum Circuits Employing Roots of the Pauli Matrices. *Phys. Rev. A*, **88**, Article ID: 042322. https://doi.org/10.1103/PhysRevA.88.042322

[17] (2019) QISKIT Python SDK. https://github.com/QISKit/qiskit-sdk-py

[18] Zulehner, A., Paler, A. and Wille, R. (2018) Efficient Mapping of Quantum Circuits to the IBM QX Architectures. *Design, Automation and Test in Europe*. https://doi.org/10.23919/DATE.2018.8342181

[19] Tan, Y.Y., Cheng, X.Y., Guan, Z.J., *et al*. (2018) Multi-Strategy Based Quantum Cost Reduction of Linear Nearest-Neighbor Quantum Circuit. *Quantum Information Processing*, **17**, 61. https://doi.org/10.1007/s11128-018-1832-y