CNOT-Optimal Circuit Synthesis

Yifan Kang

Under the direction of
Henry Ma
Massachusetts Institute of Technology

Research Science Institute
August 1, 2023

Abstract

The Controlled-NOT (CNOT) gate is central to quantum circuit design due to its role in entanglement generation and information processing. The task of achieving CNOT-optimality becomes increasingly intricate when introducing additional connectivity limitations that restrict the permissible connections between qubits, which can be represented as a graph.

We first give a polynomial-time algorithm for checking if an n-qubit circuit is constructible on a given topological constraint. Our second result is an NP-hardness proof for the problem of finding CNOT-optimal circuits under directed topological constraints. We also make progress towards proving NP-hardness for the analogous problem which considers undirected constraints, which is relevant in real-life settings.

Summary

Computer science studies how easy or hard it is to solve computational problems with sequences of instructions known as algorithms. While some problems are known to have efficient algorithms, other problems are provably difficult to solve efficiently; these are known as NP-hardness results. In this paper, we focus on a specific type of quantum circuit, the CNOT circuit, and prove that optimizing the size of these circuits is a difficult task.

1 Introduction

Quantum circuit synthesis is the process of designing quantum circuits to implement specific quantum algorithms or operations using a set of basic quantum gates [SBM06, NC02]. Among the various quantum gates, the Controlled-NOT (CNOT) gate is central to quantum circuit design due to its role in entanglement generation and information processing.

In the context of circuit optimization, the pursuit of CNOT-optimality has become a fundamental challenge in quantum computing research [VMS04]. A CNOT-optimized circuit minimizes the number of CNOT gates needed to realize a quantum algorithm, thus reducing the overall execution time and improving the accuracy of computations. However, achieving CNOT-optimality becomes even more difficult when the connectivity between qubits is restricted.

Many researchers have focused on reducing the size/depth of CNOT circuits without considering connectivity. For size optimization, Patel *et al.* [PMH03] proved that each n-qubit CNOT circuit can be synthesized with $O(n^2/\log n)$ CNOT gates, and this bound is asymptotically tight. Some authors have also taken undirected topological constraints into consideration. For example, synthesis algorithms have been designed by Kissinger-de Griend [KvdG19] to build circuits of size $O(n^2)$. Generalizing the results of Patel *et al.*, Wu *et al.* have proposed an algorithm that achieves a worst-case bound of $O(\frac{n^2}{\log \delta})$ on any connected graph with minimum degree δ [WHY+23].

Moreover, there are also studies that focus on the hardness of optimizing CNOT circuits. Amy, Azimzadeh, and Mosca [AAM18] have shown that the problem of minimizing CNOT count in the presence of Z-basis rotation gates is NP-complete when all CNOT gates are required to have the same target. Jiang *et al.* [JST+22] have proven the NP-hardness of CNOT-optimality under directed topological constraints with ancilla qubits, which are additional qubits used in a CNOT circuit to help the computation. They have also shown the hardness of optimizing a sub-circuit of a CNOT circuit which has ancilla qubits.

In this paper, we consider the Minimum CNOT Circuit Size Problem under directed topological constraints without ancilla qubits. In Section 3, we propose an algorithm that can verify the constructibility of a CNOT circuit under any directed topological constraints. We then prove that the above problem is NP-hard by reducing the Vertex Cover Problem to it in Section 4. In Section 5, we prove partial results which give evidence of hardness for the analogous problem with undirected topological constraints.

2 Preliminaries

2.1 Basic notations

We use $[n]$ to denote the set $\{1, 2, 3, \dots, n\}$ and $\mathbb{F}_q$ to denote the finite field with q elements; for $q = 2$, we use $\oplus$ to denote xor, i.e. the addition under $\mathbb{F}_2$.

We use $GL(n, \mathbb{F}_2)$ to denote the set of all $n \times n$ invertible matrices over $\mathbb{F}_2$, I to denote the identity matrix, $M_{i,j}$ to denote the (i, j)-entry of matrix M, and $e_{i,j}$ to denote the matrix M where only $M_{i,j}$ equals 1.

2.2 CNOT Circuits

Definition 2.1. A **CNOT gate** with control qubit x_i and target qubit x_j maps (x_i, x_j) to $(x_i, x_i \oplus x_j)$. A **CNOT circuit** is a circuit that only contains CNOT gates. For our convenience, we will denote a CNOT gate with control qubit i and target qubit j as $CX(i, j)$.

Mathematically, it is the invertible linear map $\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ over $\mathbb{F}_2^2$. Thus, each n-qubit CNOT circuit C can be seen as an invertible linear map over $\mathbb{F}_2^n$, i.e. an invertible matrix $M \in GL(n, \mathbb{F}_2)$.

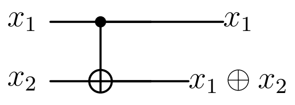


Figure 1: $CX(1, 2)$

2.3 Graph Theory

We use graph theory to describe the limited connectivity of quantum devices. All graphs we consider will have no multiple edges and no self-loops.

Definition 2.2. A **graph** $G = (V, E)$ consists of V, a set of **vertices**, and $E \subset \{(x, y) | x, y \in V \wedge x \neq y\}$, a set of **edges**. The graph is **directed** if E is a set of ordered pairs and **undirected** if E is a set of unordered pairs.

Definition 2.3. For graph $G = (V, E)$ and a subset $S \subseteq V$, the **induced subgraph** of S is $G[S] = (S, E')$, where $E' = \{(u, v) \mid u, v \in S\} \cap E$.

Definition 2.4. For a directed graph $G = (V, E)$, we say $t \in V$ is **reachable** from $s \in V$ if there exists a sequence of vertices:

$$v_0 = s, \quad v_1, \quad v_2, \quad \ldots, \quad v_k = t.$$

such that the edge (v_{i-1}, v_i) is in E for all $1 \leq i \leq k$. The vertex sequence is a **path** from s to t. If $s = t$, the vertex sequence is called a **cycle**.

Definition 2.5. A directed graph $G = (V, E)$ is said to be **strongly connected** if every vertex $t \in V$ is reachable from every other vertex $s \in V$.

Similarly, vertices u and v are said to be **strongly connected** to each other if there exists a path from u to v and a path from v to u.

We can see that the binary relation of being strongly connected is an equivalence relation:

Definition 2.6. For a graph $G = (V, E)$ and $u, v \in V$, $u \sim v$ if and only if u and v are strongly connected. The equivalence relation partitions the vertices into different sets, and the induced subgraph of each set is called a **strongly connected component** (SCC) of G.

Definition 2.7. A **directed acyclic graph** is a directed graph with no cycles.

2.4 Minimum CNOT Circuit Size Problem

We are interested in optimizing quantum circuits by minimizing the number of CNOT gates used.

Definition 2.8 (Minimum CNOT Circuit Size Problem)**.** The problem, denoted by `MCCSP`, is defined as follows:

- Input: An n-qubit CNOT circuit C and an integer k.
- Question: Does there exist an n-qubit CNOT circuit with at most k CNOT gates which is equivalent to C?

If we see each CNOT gate as an invertible linear map on $\mathbb{F}_2^n$, a CNOT circuit is essentially a composition of these maps. Hence, the problem can also be phrased in a linear algebra manner:

Definition 2.9 (Minimum Row Elimination Problem)**.**

- Input: an $n \times n$ invertible matrix M and an integer k.
- Question: Does there exist a sequence of elementary row-elimination matrices

  $$\mathbf{E_{k'}} \cdots \mathbf{E_2 E_1} = \mathbf{M}$$

  of length $k' \leq k$ whose composition is M?

  Here, each $\mathbf{E_i}$ satisfies $\mathbf{E_i} = I + e_{j,i}$ for some $i \neq j$. It is the matrix representation of the gate $CX(i,j)$.

In real life, however, it is not possible to place 2-qubit gates in arbitrary pairs of qubits. The limitations of the qubit connection can be represented as a directed graph, which is called the *topological constraint* of the problem.

Definition 2.10 (Topological Constraint)**.** A **topological constraint** is a set $S \subseteq \{(i,j) \in [n] \times [n] | i \neq j\}$. Any CNOT gate $CX(i,j)$ we construct in the circuit satisfies $(i,j) \in S$.

Example 2.11. For example, $S = \{(1,2),(2,3),(3,4)\}$, Figure 2 represents S:

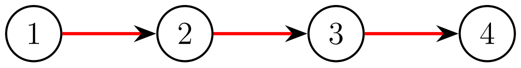
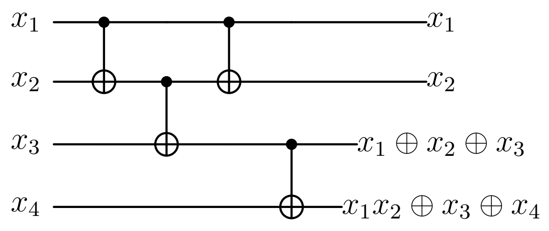
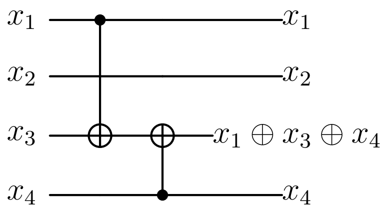


Figure 2: Directed Topological Constraint

Under the constraint S, the CNOT circuit shown in Figure 3a is valid, since it only uses $CX(1,2)$, $CX(2,3)$, and $CX(3,4)$. The circuit shown in Figure 3b is not, because neither $CX(1,3)$ nor $CX(4,3)$ is within our constraint.



(a) Valid Circuit on S



(b) Invalid Circuit on S

Figure 3: CNOT circuits under the constraint $S = \{(1,2),(2,3),(3,4)\}$

We let $\texttt{MCCSP}_{\texttt{DTC}}$ denote the variant of `MCCSP` which take a directed topological constraint S as an additional input, and asks whether the CNOT circuit C is constructible using at most k gates from S. $\texttt{MCCSP}_{\texttt{UDTC}}$ is defined analogously for undirected topological constraints.

3 Circuit Constructibility

What kind of circuits are constructible on a given topological constraint? Are we able to verify them using an efficient algorithm? In this section, we give a method to verify circuit constructibility efficiently for any constraints. We first show that it is possible to perform a CNOT operation between two qubits i and j if there exists a path in the topological constraint from i to j. We build on the construction that has been previously proposed by Nash *et al.* [NGM20] and prove that the construction is optimal. Finally, we apply this property and generalize it to an algorithm that can verify whether a circuit is constructible on any topological constraint.

Proposition 3.1. *For the topological constraint $S = \{(i, i+1) \mid 1 \le i < n\}$ and $n > 2$, there is a circuit with $4(n-2)$ CNOT gates which is constructs the operation $CX(1,n)$. Furthermore, this construction is optimal in the number of CNOT gates.*

Proof. We show the construction of the circuit here and the detailed proof for the lower bound is in Appendix A. First, we will give the construction $CX(1,n)$ using $4(n-2)$ CNOT gates. We perform the following gates in order:

1. $CX(1,2)$, $CX(2,3)$, $\dots$, $CX(n-1,n)$. This will transform the identity matrix to a lower triangular matrix with $M_{i,j} = 1$ for all $j \le i$.

2. $CX(n-2,n-1)$, $CX(n-3,n-2)$, $\dots$, $CX(1,2)$. Now, $M_{i,j} = 1$ for all (i,j) with $i = j$ or $i = n$.

3. $CX(2,3)$, $CX(3,4)$, $\dots$, $CX(n-1,n)$. We have $M_{1,1} = M_{1,n} = M_{n,n} = 1$ and $M_{i,j} = 1$ for all (i,j) with $1 < i,j < n$ and $i \le j$.

4. $CX(n-2,n-1)$, $CX(n-3,n-2)$, $\dots$, $CX(2,3)$. This gives us the desired matrix.

□

Example 3.2. We show an example of our construction when $n = 4$ in Figure 4.

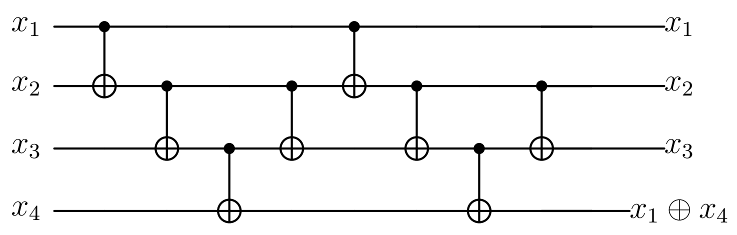


Figure 4: Construction when $n = 4$

If we represent the circuit as an 4×4 matrix, the construction works as follows:

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix} \Longrightarrow \begin{bmatrix}1&0&0&0\\1&1&0&0\\1&1&1&0\\1&1&1&1\end{bmatrix} \Longrightarrow \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\1&1&1&1\end{bmatrix} \Longrightarrow \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&1&1&0\\1&0&0&1\end{bmatrix} \Longrightarrow \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\1&0&0&1\end{bmatrix}$$

Corollary 3.3. *If the topological constraint S on n-qubits is strongly connected, then any CNOT circuit can be constructed on S.*

Proof. By the construction in Proposition 3.1, we can construct a CNOT circuit equivalent to $CX(u, v)$ if there exists a path from u to v. Since the graph S is strongly connected, every $CX(u, v)$ is constructible. It follows that any CNOT circuit will also be constructible. □

Theorem 3.4. *Given an n-qubit CNOT circuit in the form of an $n \times n$ invertible matrix M, and a topological constraint S, M can be constructed on S if and only if both:*

- *If $M_{i,j} = 1$, then i is reachable from j.*
- *For any SCC G of S, the submatrix of M formed by taking the indices in G is invertible.*

Proof. We start by showing the necessity of our conditions. Suppose M can be constructed on S, and label the strongly connected components of S as $G_1, G_2, \ldots, G_k$.

Suppose i is not reachable from j but $M_{i,j} = 1$, then there must exist a $CX(v_1, i)$ that adds the j-th entry to row i. If $v_1 \neq j$, there must also exist a $CX(v_2, v_1)$ that adds the j-th entry to row v_1. Since the circuit has finitely many CNOT gates, the process must stop at $v_t = j$. The sequence j, v_t, v_{t-1}, $\ldots$, v_1, i is a path on S. Contradiction! Hence the first condition must hold.

Moreover, the submatrix formed by the indices in G_i only changes when we perform $CX(u, v)$ such that $u, v \in G_i$. So, taking a CNOT circuit on S which constructs M and restricting to G_i gives a sequence of invertible CNOTs which constructs the submatrix. It follows that the submatrix is invertible.

Now, we want to show that these conditions are sufficient. Define a graph $S' = (V', E')$, where $V' = [k]$ and $(i, j) \in E'$ if and only if there exists $u \in G_i, v \in G_j$ such that $(u, v) \in E$. Since there does not exist u and v in different SCCs which are strongly connected, S' forms a DAG. Intuitively, we can think of each strongly connected component of S as a single node. Then S can be seen as a DAG of strongly connected components.

We can construct the invertible matrix M with operations from S, using the following algorithm which is inspired by topological sort.

1. On graph S', Let V_0 be the set of vertices v with no outgoing edges. If $V_0 = \varnothing$, we terminate the algorithm.

2. Each $v \in V_0$ represents an SCC $G_v \in S$. By assumption, the submatrix of M formed by the indices in G_v is invertible, so by Corollary 3.3, we can construct this submatrix using CNOTs in G_v. We add these gates to our circuit.

3. For all $M_{i,j} = 1$ that satisfies $i \in G_v$ for some $v \in V_0$ and $j \notin G_v$, we can build a CNOT circuit equivalent to $CX(j, i)$ since i is reachable from j by assumption, and we add these gates to our circuit.

4. Remove from S' the vertices in V_0 and their incident edges, then go back to step 1.

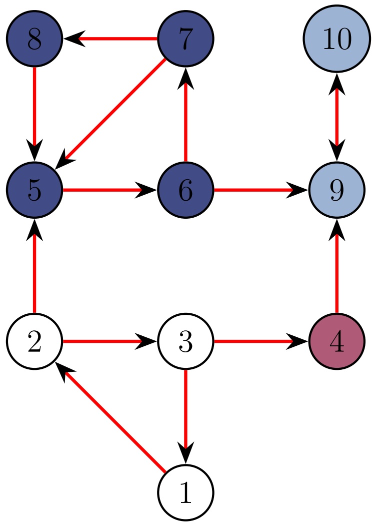
We can prove the correctness of the algorithm by induction on k, the number of SCCs. When $k = 1$, constructibility follows from Corollary 3.3. Now suppose the algorithm works for topological constraints with at most k SCCs. We want to show that it works for a topological constraint S with $k + 1$ SCCs.

In step 2, we construct a matrix M' with $M'_{i,j} = 1$ for all (i, j) with $M_{i,j} = 1$ and $i, j \in G_v$ for some $v \in V_0$. Let V' be the set of all indices $v \in G_i$ for all $i \in V_0$. V' is the set of indices included in the SCCs represented by V_0. In step 3, each $CX(j, i)$ adds row j to row i. Since $j \notin V'$, the only nonzero entry on row j is $M'_{j,j}$. Hence, we construct a matrix M' with $M'_{i,j} = 1$ for all (i, j) with $M_{i,j} = 1$ and $i \in V'$.

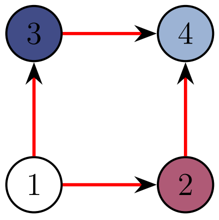
Since we complete all the rows i with $i \in V'$, we may delete the edges in S' that end at $v \in V_0$. Note that $|V \setminus V_0| \leq k$ and the graph S' is still a DAG after step 4, we are done by induction.

□

Example 3.5. Figure 5 illustrates an example of representing a directed topological constraint S as S', a DAG of SCCs on S.



(a) Topological Constraint S



(b) Directed Acyclic Graph S'

Figure 5: Transforming a directed graph into a DAG

Generally speaking, we decompose the matrix into several submatrices and construct them in topological order in S'. The following example illustrates our algorithm for a 10×10 invertible matrix M on the previous S:

Suppose we want to construct the matrix

$$M = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Initially, we have $M' = I$. In step 1, we take out $V_0 = \{4\}$ in S', which corresponds to

$V' = \{9, 10\}$ in the original graph S. In step 2, we construct the invertible 2×2 submatrix:

$$M' = \left[\begin{array}{cccccccccc}
1&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&0\\ \cline{9-10}
0&0&0&0&0&0&0&0&\multicolumn{1}{|c}{1}&\multicolumn{1}{c|}{1}\\
0&0&0&0&0&0&0&0&\multicolumn{1}{|c}{0}&\multicolumn{1}{c|}{1}\\ \cline{9-10}
\end{array}\right].$$

In step 3, we complete all the rows i for $i \in V'$:

$$M' = \left[\begin{array}{cccccccccc}
1&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0\\
0&0&0&1&0&0&0&0&0&0\\
0&0&0&0&1&0&0&0&0&0\\
0&0&0&0&0&1&0&0&0&0\\
0&0&0&0&0&0&1&0&0&0\\
0&0&0&0&0&0&0&1&0&0\\ \cline{1-8}
\multicolumn{1}{|c}{0}&1&1&1&1&1&1&\multicolumn{1}{c|}{1}&1&1\\
\multicolumn{1}{|c}{1}&0&0&0&0&1&1&\multicolumn{1}{c|}{1}&0&1\\ \cline{1-8}
\end{array}\right].$$

We go back to step 1. Now, $V_0 = \{2, 3\}$ and $V' = \{4, 5, 6, 7, 8\}$. In step 2, we construct the invertible submatrix for G_3 and for G_2 respectively:

$$M' = \left[\begin{array}{cccccccccc}
1&0&0&0&0&0&0&0&0&0\\
0&1&0&0&0&0&0&0&0&0\\
0&0&1&0&0&0&0&0&0&0\\
0&0&0&\boxed{1}&0&0&0&0&0&0\\ \cline{5-8}
0&0&0&0&\multicolumn{1}{|c}{0}&1&0&\multicolumn{1}{c|}{1}&0&0\\
0&0&0&0&\multicolumn{1}{|c}{0}&0&1&\multicolumn{1}{c|}{0}&0&0\\
0&0&0&0&\multicolumn{1}{|c}{1}&1&1&\multicolumn{1}{c|}{0}&0&0\\
0&0&0&0&\multicolumn{1}{|c}{0}&0&1&\multicolumn{1}{c|}{1}&0&0\\ \cline{5-8}
0&1&1&1&1&1&1&1&1&1\\
1&0&0&0&0&1&1&1&0&1
\end{array}\right].$$

In step 3, we complete all the rows i for $i \in V'$:

$$M' = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

We go back to step 1 and we have $V_0 = \{1\}$ and $V' = \{1, 2, 3\}$. We simply finish the algorithm by constructing the last invertible submatrix:

$$M' = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}.$$

Corollary 3.6. *For any n-qubit CNOT circuit and a given topological constraint S, we can verify if the circuit is constructible in polynomial time.*

Proof. We can use Tarjan's algorithm [Tar72] to find all the strongly connected components of the graph S in linear time, it takes polynomial time to go through all the elements $M_{i,j}$, and it takes polynomial time to check whether a submatrix of M is invertible. $\square$

4 NP-Hardness Results

In this section, we prove that Minimum CNOT Circuit Size Problem is NP-hard under directed topological constraints by reducing the vertex cover problem to it.

Definition 4.1 (Vertex Cover Problem)**.** A **vertex cover** V' of an undirected graph $G = (V, E)$ is a subset of $V' \subset V$, such that for any $(u, v) \in E$, $u \in V'$ or $v \in V'$, i.e. at least one endpoint of every edge is in the vertex cover V'.

The **vertex cover problem**, denoted by `VCP`, is defined as follows:

- Input: A graph $G = (V, E)$ and an integer k.

- Question: Does there exist a vertex cover $V' \subset V$ in G such that $|V'| \leq k$?

It is well-known that the vertex cover problem is NP-complete [Sip13].

Theorem 4.2. $\texttt{MCCSP}_{\texttt{DTC}}$ *is NP-hard.*

Proof. We want to construct an input for $\texttt{MCCSP}_{\texttt{DTC}}$ based on the given input $G = (V, E)$ and k for the vertex cover problem: Given input $G = (V, E)$ and k of the minimum vertex cover problem, we construct the input of $\texttt{MCCSP}_{\texttt{DTC}}$ as follows:

- Let $V = [p], E = \{(u_1, v_1), (u_2, v_2), \ldots, (u_q, v_q)\}$ for some $u_i, v_i \in [p]$. Set $n = p+q+1$.
- Set $S = \{(0, i) \mid 1 \leq i \leq p\} \cup \{(u_i, p+i) \mid i \in [q]\} \cup \{(v_i, p+i) \mid i \in [q]\}$.

The CNOT circuit in the input is constructed in the following way:

- $CX(u_i, p+i)$ for all $i \in [q]$. This adds x_{u_i} to qubits $p+i$ for all $i \in [q]$.
- $CX(0, v_i)$, $CX(v_i, p+i)$, and $CX(0, v_i)$ for all $i \in [q]$. This adds $x_{v_i} \oplus x_0$ to qubits $p+i$ for all $i \in [q]$ while keeping the other qubits unchanged.

If we denote the input as $|x_0\rangle |x_1 x_2 \cdots x_p\rangle |y_1 y_2 \cdots y_q\rangle$, then the topological constraints construct three layers of qubits representing a source node, the vertex set V in G, and the edge set E in G.

The batch of gates in our construction will add $x_{u_i} \oplus x_{v_i} \oplus x_0$ to y_is. Hence, the output of the circuit is:

$$|x_0\rangle |x_1 x_2 \cdots x_p\rangle |y_1 \oplus x_{u_1} \oplus x_{v_1} \oplus x_0\rangle \cdots |y_q \oplus x_{u_1} \oplus x_{v_1} \oplus x_0\rangle .$$

Finally, we ask if there exists an equivalent CNOT circuit with no more than $2k + 2q$ CNOT gates.

Claim 4.3. *If the size of the minimum vertex cover of $G = (V, E)$ is τ, the upper bound of the minimum circuit size is $2\tau + 2q$.*

Proof of Claim 4.3. We can prove the upper bound of the minimum size by providing the following construction of a circuit C:

1. Suppose V' is the minimum vertex cover of the graph G. For $i \in [q]$ such that $u_i, v_i \in V'$, we perform $CX(u_i, p+i)$.
2. We perform $CX(0, v)$ for all $v \in V'$.
3. For $i \in [q]$, we perform $CX(u_i, p+i)$ if we haven't done so in step 1, then we perform $CX(v_i, p+i)$.
4. We perform $CX(0, v)$ for all $v \in V'$.

It is not difficult to verify that the construction works. We have $2q$ gates in total in steps 1 and 3, and another 2τ gates in steps 2 and 4. □

We now establish the lower bound by proving the following claim:

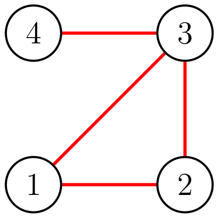
Claim 4.4. *The number of $CX(0, v)$ in C is at least 2τ, while the number of $CX(v, e)$ in C is at least $2q$, where $v \in [p]$ and $p + 1 \leq e \leq p + q$.*

Proof of Claim 4.4. Let V_0 be a subset of $[p]$ such that for any $CX(0, v)$ in the circuit, $v \in V_0$. If $|V_0| < \tau$, then there exists some $e_i \in E$ such that $u_i, v_i \notin V_0$, which contradicts with the final state of qubit $p + i$. Thus, $|V_0| \geq \tau$. Notice that for each $v \in V_0$, we need at least 2 $CX(0, v)$ since the final state for qubit v is $|x_v\rangle$. This implies the lower bound of the number of $CX(0, v)$ is 2τ.

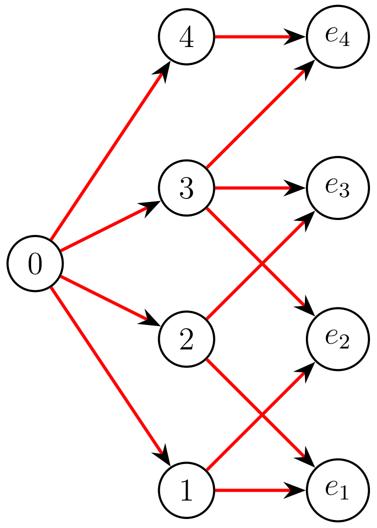
For each $i \in [q]$, $CX(u_i, p+i)$ and $CX(v_i, p+i)$ must be in C in order to have the state $|y_i \oplus x_{u_i} \oplus x_{v_i} \oplus x_0\rangle$. Hence, the number of $CX(v, e)$ is at least $2q$. □

Finally, we can combine *Claim* 4.3 and *Claim* 4.4 to get that the minimum size of the circuit C is exactly $2\tau + 2q$. Hence, deciding whether there exists a vertex cover of size at most k on the graph G is equivalent to finding a circuit of size at most $2k+2q$. This mapping implies the NP-hardness of $\texttt{MCCSP}_{\texttt{DTC}}$. □

Example 4.5. In Figure 6b, we have shown our construction for the graph G when $V = \{1, 2, 3, 4\}$ and $E = \{(1, 2), (1, 3), (2, 3), (3, 4)\}$:



(a) Original Graph G



(b) Topological Constraint S

Figure 6: Construction of S for NP-hardness Proof

5 Generalization of the NP-Hardness Results

We conjecture that it is possible to generalize the directed constraint construction to an undirected constraint.

Conjecture 5.1. $\texttt{MCCSP}_{\texttt{UDTC}}$ *is NP-hard.*

To prove the conjecture, we want to apply the previous construction of the topological constraint S and the circuit C in Section 4. The only difference is that all constraints in S are undirected, which means that $CX(i,j)$ and $CX(j,i)$ are both legal if $(i,j) \in S$. It suffices to prove the following conjecture:

Conjecture 5.2. *If the size of the minimum vertex cover of $G = (V, E)$ is τ, the minimum size of the circuit C is $2\tau + 2q$.*

For our convenience, we define some useful notations for our proof:

- Let $c(i,j)$ denote the number of $CX(i,j)$ in the circuit C.
- Let in_i denote the number of $CX(x,i)$ in the circuit C, out_i denote the number of $CX(i,x)$ in the circuit C. In other words,

$$\text{in}_i = \sum_x c(x,i), \quad \text{out}_i = \sum_x c(i,x).$$

Claim 5.3. *Let $V_0 = \{v \mid v \in [p] \text{ and } \text{in}_v > 0\}$,*

$$A = \sum_{v \in V_0} \text{in}_v \geq 2\tau.$$

Proof.

- If $|V_0| < \tau$, then there exists some $e_i \in E$ such that $u_i, v_i \notin V_0$, which contradicts with the final state of qubit $p + i$.
- For each $v \in V_0$, $\text{in}_v \geq 2$ since the qubit restores its original state $|x_v\rangle$.

Hence,

$$A \geq |V_0| \min_{v \in V_0} \text{in}_v \geq 2\tau.$$

□

Let

$$B = \sum_{e=p+1}^{p+q} \text{in}_e.$$

If $B \geq 2q$, then we have the desired lower bound in the conjecture 5.2.

Thus, we may assume that $B < 2q$. Let $E_0 = \{e \mid e \in [q] \text{ and } \text{in}_{e+p} = 1\}$, $|E_0| = t$.

We show some intuition on why the conjecture works by making the graph partially undirected. We prove the following lemma:

Lemma 5.4. *If $c(j,i) = 0$ for all $i \in [p]$ and $p + 1 \leq j \leq p + q$, then the minimum size is $2\tau + 2q$.*

Proof of lemma 5.4. WLOG, let $CX(u_i, i+p)$ be the only CNOT gate with target $i+p$ for all $i \in E_0$. Qubit u_i must have the state $x_{u_i} \oplus x_{v_i} \oplus x_0$ at some point.

We have to add x_{v_i} to qubit u_i in C. Since the only CNOT gate with u_i as a target is $CX(0, u_i)$ and we must add x_{v_i} to qubit u_i, for all $i \in E_0$, we have $c(v_i, 0) > 0$. Let $V_1 = \{v \mid v \in [p] \text{ and } c(v, 0) > 0\}$, then V_1 forms a vertex cover of the graph spanned by edges e_i where $i \in E_0$.

Assume for some $v \in V_1$ we have $c(v, 0) = 1$. Observe that the only qubit i with $0 \leq i \leq p$ and x_v in its state is qubit v. If $c(v, 0) = 1$, then the number of qubits i other than qubit v with $0 \leq i \leq p$ with x_v in its state is strictly greater than 0. Contradiction! Let $|V_1| = a$, we have

$$\sum_{v \in V_1} c(v, 0) \geq 2a.$$

Next, we claim that if we delete all the $CX(0, u_i)$ that turns the state of qubit u_i to $x_{u_i} \oplus x_{v_i} \oplus x_0$, V_0 still forms a vertex cover for the edge set with indices $i \in [q] \setminus E_0$. Suppose some edge i in the set is not covered, then:

- If no u_j with $j \in E_0$ are endpoints of e_i, then we are not able to add x_0 to y_i.
- If some u_j with $j \in E_0$ are endpoints of e_i, then we perform $CX(u_j, p+i)$ in the circuit to add $x_{u_j} \oplus x_{v_j} \oplus x_0$ to y_i. Suppose $e_i = (u_j, w)$, then we need some $CX(w, p+i)$ to add $x_{v_j} \oplus x_w$.

In both cases, we reach a contradiction. Thus, we have

$$A \geq t + b + |V_0|.$$

where b is the size of V_0 after we delete edges with indices in E_0 in graph G. Combine the inequalities, we get:

$$A + B \geq (2q - t) + 2a + t + b + |V_0| \geq 2q + 2a + b + \tau \geq 2m + 2\tau + a.$$

The last inequality applies the fact that two sets of vertices with size a and b can cover the graph G so $a + b \geq \tau$. □

6 Conclusion

In this paper, we present an efficient algorithm to check CNOT circuit constructibility on directed topological constraints. In addition, we show that CNOT-circuit optimization is NP-hard on directed topological constraints and attempt to generalize our results to undirected constraints. Some interesting questions in this field still remain open:

- Most NP-hardness results on the optimization of CNOT circuits heavily depend on multi-layer topological constraints. Is the problem still NP-hard on more general constraints, such as a $m_1 \times m_2 \times \cdots \times m_d$ grid?

- A relevant problem that introduces the idea of parallel computing is optimizing the depth of a CNOT circuit. [JST$^+$22] have provided constructions for many specific structures of CNOT circuits, such as a CNOT tree, in $O(\log^2 n)$ depth. It is also possible to verify efficiently if a circuit can be parallelized to depth 2. Is there a way to characterize all low-depth circuits?

7 Acknowledgments

I would like to thank my mentor, Henry Ma, for his illuminating insights and mentorship. I would like to thank my tutor Pesho Gaydarov, Dr. Tanya Khovanova, and Prof. David Jerison for their suggestions on my writing and presentation. I would also like to acknowledge the Research Science Institute, Center of Excellence in Education, and Massachusetts Institute of Technology for supporting my research and making it possible.

References

[AAM18] Matthew Amy, Parsiad Azimzadeh, and Michele Mosca. On the controlled-NOT complexity of controlled-NOT–phase circuits. *Quantum Science and Technology*, 4(1):015002, sep 2018.

[JST$^+$22] Jiaqing Jiang, Xiaoming Sun, Shang-Hua Teng, Bujiao Wu, Kewen Wu, and Jialin Zhang. Optimal space-depth trade-off of cnot circuits in quantum logic synthesis, 2022.

[KvdG19] Aleks Kissinger and Arianne Meijer van de Griend. Cnot circuit extraction for topologically-constrained quantum memories, 2019.

[NC02] Michael A Nielsen and Isaac Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, London, 2002.

[NGM20] Beatrice Nash, Vlad Gheorghiu, and Michele Mosca. Quantum circuit optimizations for NISQ architectures. *Quantum Science and Technology*, 5(2):025010, mar 2020.

[PMH03] K. N. Patel, I. L. Markov, and J. P. Hayes. Efficient synthesis of linear reversible circuits, 2003.

[SBM06] V.V. Shende, S.S. Bullock, and I.L. Markov. Synthesis of quantum-logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(6):1000–1010, jun 2006.

[Sip13] Michael Sipser. *Introduction to the Theory of Computation*. Course Technology, Boston, MA, third edition, 2013.

[Tar72] Robert Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2):146–160, 1972.

[VMS04] Juha J. Vartiainen, Mikko Möttönen, and Martti M. Salomaa. Efficient decomposition of quantum gates. *Physical Review Letters*, 92(17), apr 2004.

[WHY$^+$23] Bujiao Wu, Xiaoyu He, Shuai Yang, Lifu Shou, Guojing Tian, Jialin Zhang, and Xiaoming Sun. Optimization of CNOT circuits on limited-connectivity architecture. *Physical Review Research*, 5(1), jan 2023.

A Proof of Proposition 3.1

Proof. For $\mathbb{F}_2^n$, we define $\{x_1, x_2, \ldots, x_n\}$ as its orthogonal basis.

For our convenience, Denote the number of $CX(i, i+1)$ in our circuit C as a_i, and M as the resulting $n \times n$ matrix of the CNOT circuit.

First, we can see that $a_i > 0$ for all i in order to add x_1 to x_n. Furthermore, we can infer that $a_i \geq 2$ for $i < n-1$ since qubit 2, qubit 3, $\cdots$, and qubit $n-1$ have to stay the same after the circuit. This inspires us to establish the lower bound by bounding each a_i.

Claim A.1. *For all $i \in [n-1]$, a_i is even.*

Proof. Since we always have $M_{i,i} = 1$ for all $i \in [n]$, any $CX(i, i+1)$ will flip the parity of $M_{i+1,i}$. By $M_{i+1,i} = 0$, we know that $a_i \equiv 0 \pmod 2$ for all $i \in [n-1]$. $\square$

Combining the claim with $a_i > 0$, we immediately get $a_i \geq 2$ for all $i \in [n-1]$. We need a stronger statement to prove the proposition.

Claim A.2. *For all $2 \leq i \leq n-2$, we have $a_i > 2$.*

Proof. Suppose $a_i = 2$ for some $2 \leq i \leq n-2$, then the qubit $i+1$ has only one other state vector other than x_{i+1}. Let the state be $v \in \mathbb{F}_2^n$. We have

$$\text{span}(\{x_{i+1}, v\}) = \{x_{i+1}, x_{i+1} + v, v, 0\}.$$

Since $v \neq x_1$ and $v \neq x_{i+1} + x_1$, x_1 is not in the span.

However, we can prove by induction that the vector formed by the first $i+1$ entries of row n must be in the span of the vectors in row $i+1$, which contradicts with $M_{n,1} = 1$. Thus, $a_i > 2$ for all $2 \leq i \leq n-2$. $\square$

Applying *Claim* A.1 again, we have $a_i > 4$ for all $2 \leq i \leq n-2$. Hence, we established the lower bound of the size of our CNOT circuit to be

$$2 + 4 \times (n-3) + 2 = 4n - 8.$$

$\square$