On the complexity of CNOT-based circuit synthesis

Della Giustina Lorenzo
dellagiustina.lorenzo@spes.uniud.it
158854

March 28, 2025

Abstract

This thesis focuses on the study of the complexity of the **CNOT-based optimal circuit synthesis problem under directed topological constraints**. The problem is **proved to be NP-hard** to approximate by means of a prior known result about the variant with ancillae qubits [1].

Keywords Quantum Computing · Circuit Synthesis · CNOT · Complexity

1 Problem statement

Controlled NOT (or CNOT), is a quantum logic gate essential in the construction of quantum circuits. In particular, it is a 2-arity reversible operator (as all quantum gates) defined as:

$$CNOT(|x_1\rangle, |x_2\rangle) = (|x_1\rangle, |x_1 \oplus x_2\rangle) \tag{1}$$

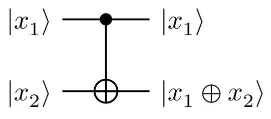


Figure 1: Graphical representation of the CNOT logic gate.

Here $|x_1\rangle$ and $|x_2\rangle$ are qubits. Each qubit $|x\rangle$ is a 2D complex vector ($|x\rangle \in \mathbb{C}^2$), i.e. a vector in the complex space spanned by the basis:

$$|0\rangle := \begin{pmatrix}1\\0\end{pmatrix}, |1\rangle := \begin{pmatrix}0\\1\end{pmatrix} \tag{2}$$

The **truth table**[1] of the CNOT gate is:

$\boldsymbol{|x_1\rangle}$	$\boldsymbol{|x_2\rangle}$	$\boldsymbol{|x_1\rangle}$	$\boldsymbol{|x_1\rangle \oplus |x_2\rangle}$
$|0\rangle$	$|0\rangle$	$|0\rangle$	$|0\rangle$
$|0\rangle$	$|1\rangle$	$|0\rangle$	$|1\rangle$
$|1\rangle$	$|0\rangle$	$|1\rangle$	$|1\rangle$
$|1\rangle$	$|1\rangle$	$|1\rangle$	$|0\rangle$

[1]Since each quantum gate is a linear operator, it can be unambiguously defined by only describing its behavior on the states of the basis. Note that throughout this thesis, reasoning will be based solely on the states of the basis, leveraging linearity.

Circuits based only on CNOT gates (as all quantum circuits) are **reversible**: the transformation of the input to the output is bijective (i.e. the output can be uniquely mapped back to the input).

This thesis concerns some complexity results about the **synthesis of optimal CNOT circuits**, i.e. the problem of minimizing the number of CNOT in a given circuit.

1.1 Matrix formulation (De Brugière et al. 2021)

The application of a CNOT gate on two input qubits $CNOT(x_i, x_j)$ can be written as a matrix in the $\mathbb{F}_2$ Galois field (i.e. where the sum corresponds to the $\oplus$ operation):

$$E_{ij} = I + e_{ij} \tag{3}$$

where:

- I is the identity matrix;
- e_{ij} the elementary matrix with all entries equal to 0 but the entry (i, j) whose value is 1.

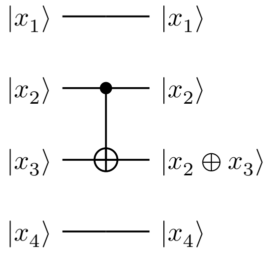


Figure 2: Exemplary CNOT gate with $i = 2, j = 3$.

$$E_{2,3} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4}$$

Figure 3: $E_{2,3}$ matrix that represents the circuit above.

Now, let us generalize and consider the synthesis of a whole CNOT circuit with input $x = |x_1, ..., x_m\rangle$ and output $y = |y_1, ..., y_m\rangle$ where each $y_i = \sum_{j \in Y_i} x_j$ (because CNOT gates can only $\oplus$ input values) with $i \in [m]$ and $Y_i \subseteq \{x_1, ..., x_m\}$. Define a matrix A such that $y = Ax$.

Finding a CNOT circuit (with n CNOT gates) implementing A[2] is therefore equivalent to finding a sequence of matrices $E_{i_k j_k}$ with $0 \leq i_k, j_k \leq n$ such that:

$$\prod_{k=1..n} E_{i_k j_k} = A \tag{5}$$

Since $E_{i_k j_k}^{-1} = E_{i_k j_k}$ it can be convenient to rewrite the synthesis as:

$$\left(\prod_{k=1..n} E_{i_k j_k} \right) A = I \tag{6}$$

where each multiplication with a matrix $E_{i_k j_k}$ corresponds to assigning to the i_k-th row the result of the $\oplus$ between the i_k-th and j_k-th rows:

$$r_{i_k} \leftarrow r_{i_k} \oplus r_{j_k} \tag{7}$$

where r_{i_k} is the i_k-th row.

[2]Note that only reversible A can be synthesized in CNOT circuits.

This shows that the **CNOT circuit synthesis**[3] **can be reformulated as the problem of finding a sequence of row operations that transforms the matrix A into the identity matrix I**. Therefore, the number of CNOT gates in the minimal circuit corresponds to the minimum number of row operations that transforms the matrix A into I.

1.2 Constrained Minimization Variant (Jiang et al. 2022)

In the current real-world quantum gates implementation, in many cases, it is impossible to turn on some desired interaction between pairs of qubits (DiVincenzo 2000). For this reason it is interesting to study the variant of the optimization problem with direct topological constraints, i.e. only certain gates are permitted.

Let $\mathscr{G}_n$ be the set of all possible CNOT gates over n qubits.

Definition 1.2.1 (Global Constrained Minimization): The problem, denoted by GCM, is defined for depth and size. This thesis concerns only the size variant GC(S)M which is defined as follows:

- Input: Positive integers n, s, non-negative integer m, directed topological constraints $S \subseteq \mathscr{G}_{n+m}$[4], and CNOT gates $g_1, ..., g_s \in S$ such that $C(g_1, ..., g_s)$ is an n-qubit m-ancilla CNOT circuit.
- Output GC(S)M (size version): the minimum integer $u \geq 0$ such that there exist CNOT gates $h_1, ..., h_u \in S$ and $C(g_1, ..., g_s) \cong C(h_1, ..., h_u)$.

A GCM problem where $m = 0$ (i.e. there are no ancillae qubits) is denoted by GCM(w/oA).

Following the notation from (Jiang et al. 2022), the problem is said to be *Global* because it concerns the minimization over the entire circuit. There exists a variant of the problem (which is not studied in this thesis) that focuses on local minimization.

2 Complexity results

A known and a novel result on the complexity of the *Global Constrained Size Minimization* are presented in this section.

2.1 Global Constrained Size Minimization (Jiang et al. 2022)

Let us introduce the ***r*-Bounded Set Cover problem**, which is a generalization of the Set Cover problem.

Definition 2.1.1 (r-Bounded Set Cover): For any integer r, the problem, denoted by ***r*BSC**, is defined as follows:

- *Input*:
  1. $[p]$: universe of elements;
  2. $W_1, ..., W_q \subseteq [p]$: q sets of cardinality $0 < |W_i| \leq r$ such that $\cup_{i=1}^{q} W_i = [p]$.
- *Output*: the minimum number of sets W_i whose union $= [p]$.

[3]Without ancillae qubits.

[4]I.e. applying a CNOT g_i is only allowed if $g_i \in S$.

Theorem 2.1.1 (rBSC is NP-hard): It is NP-hard to approximate rBSC to within a factor of $\ln r - O(\ln \ln r)$. (Feige 1998; Trevisan 2001)

Theorem 2.1.2 (GC(S)M is NP-hard): It is NP-hard to approximate the size version of GCM within any constant factor.

Example (rBSC $\preccurlyeq$ GC(S)M): The hardness of GC(S)M is proved with a reduction from the rBSC problem defined above. Before stating the proof, a reduction from a rBSC example instance is analyzed for explanatory purpose.

Let us consider the following instance of rBSC:
- elements: $p = 3$, $[p] = [3] = \{1, 2, 3\}$;
- sets: $q = 3$, $W_1 = \{1, 2\}$, $W_2 = \{2, 3\}$, $W_3 = \{1, 3\}$;
- bound: $r = 2$.

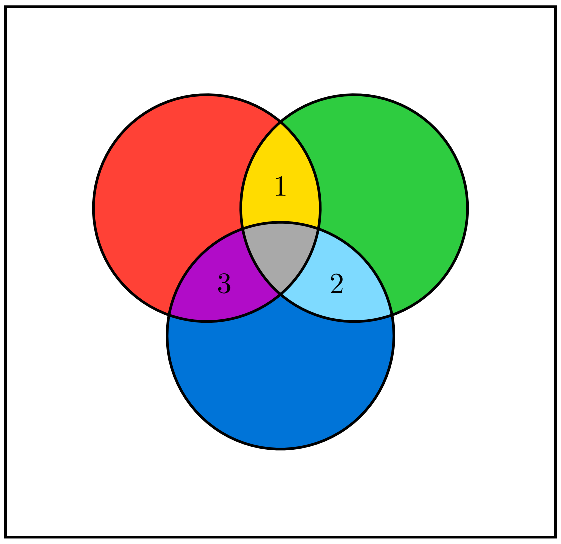


Figure 4: Graphical representation of the rBSC explanatory instance.

The optimal solution of the rBSC explanatory instance is $k = 2$ by choosing any pair of W_i, W_j with $i \neq j$.

From this rBSC instance, the aim is building a GC(S)M instance such that, solving GC(S)M would give the solution of the initial rBSC problem. The input circuit to be optimized by GC(S)M is constructed in this way:
- each element in $[p]$ is represented by a qubit $|x_i\rangle \forall i \in [p]$. In the following paragraphs, these qubits are called ***Input Qubits***;
- each set $W_j \forall j \in [q]$ is represented by ***Ancillae Qubits*** initialized to $|0\rangle$;
- additionaly, $c = p$ qubits are required.

The desired output is:
- each *Input Qubit* must be unchanged;
- each *Ancillae Qubit* must be unchanged;
- each *C Qubit* must be xored with each *Input Qubit*.

To obtain the desired output the input circuit is constructed in the following way:

- a CNOT gate is added from each *Input Qubit* x_i to the first *Ancillae Qubit* W_j such that $x_i \in W_j$. This means that for W_1 it is added a CNOT gate with all its elements, for W_2 with all its elements except the ones in W_1 and for W_3 with all its elements except the ones in W_1 and W_2;
- a CNOT is added from each *Ancillae Qubit* to each *C Qubit*.

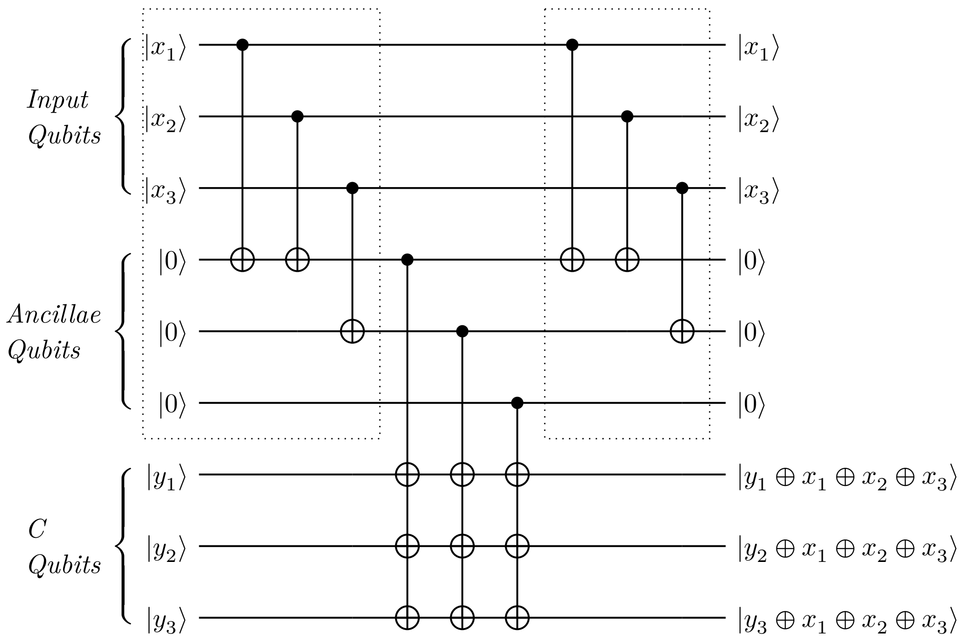


Figure 5: Graphical representation of the GC(S)M input circuit ($a = [p] = \{1, 2, 3\}$).

The set S in input to GC(S)M is constructed to permit only operations:

1. from *Input Qubit* x_i to *Ancillae Qubit* W_j (grouped in dotted line in the figure), only if $x_i \in W_j$;
2. from all *Ancillae Qubits* to all *C Qubits*.

Intuitively, the minimal number of operations of *type 1* is equal to the double of the number of elements $2p$:

- there must be at least a CNOT gate with control in each *Input Qubit* because otherwise qubits' parity $x_1 ... x_p$ couldn't be added to *C Qubit*'s output $y_1 ... y_p$;
- each one of these gates must appear an even number of times because *Ancillae Qubits* are restored to $|0\rangle$.

The minimum number that does not break these properties is $2p$.

On the other hand, the minimization of *type 2* operations represent the minimal choice of sets W_i such that their union is equal to the universe [p] (because the output of the *C Qubits* is set to contain all elements of the universe).

Therefore, minimizing this GC(S)M instance will solve the starting rBSC problem.

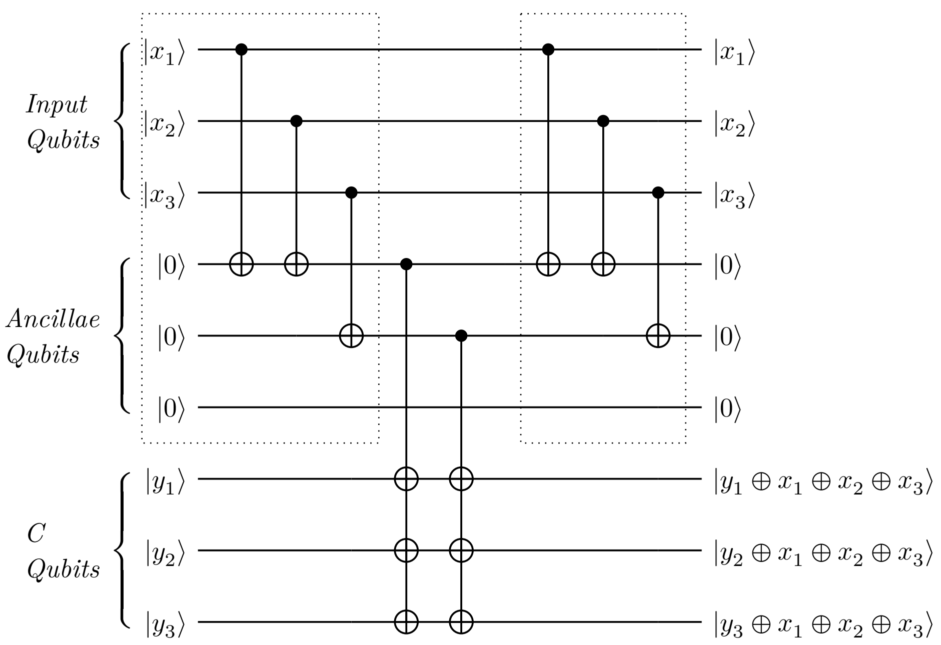


Figure 6: GC(S)M optimized circuit. Intuitively, the “choosen” sets are the once represented by the first two *Ancillae Qubits*.

The output of the GC(S)M instance defined above is $u = 2p + ck = 2p + pk = 6 + 3 * 2 = 12$.

If there exists a polynomial algorithm that approximates u within any constant factor, then using the above reduction also k can be approximated within any constant factor. This contradicts Theorem 2.1.1 (since r is an arbitrary constant in rBSC). Thus, such a polynomial algorithm cannot exist.

Below, a formalization of the reduction idea given in the above example.

Proof *Theorem 2.1.2*: The proof is by **reduction from *r*BSC**.

Given in input $r, p, q, W_1, ..., W_q$ of an instance of rBSC, an input instance (i.e. a circuit that needs to be minimized) of GC(S)M is constructed such that:

- the circuit has $p + c$ (c is a number that will be defined later) input qubits: p (*Input Qubits*) represent the elements of rBSC, and the remaining c (*C Qubits*) codify the set cover. Formally, $n = p + c$;
- the circuit has q *Ancillae Qubits*, each of one represents a set of rBSC. Formally, $m = q$;
- the circuit has $2p + cp$ CNOT gates. Formally, $s = 2p + cp$;
- constraints CNOT gates to be only:
  - from *Input Qubits* to *Ancillae Qubits* representing the set they belong to. Formally, $S_1 = \{(i, j + p) | i \in W_j, i \in [p], j \in [q]\}$;
  - from any *Ancillae Qubits* to *C Qubits*. Formally, $S_2 = \{(j + p, t + p + q) | j \in [q], t \in [c]\}$.

  Therefore, the constraints are $S = S_1 \cup S_2$;
- the circuit is constructed such that the input is

$$|x_1 ... x_p\rangle |0\rangle^{\times m} |y_1 ... y_c\rangle \tag{8}$$

and the output is

$$|x_1...x_p\rangle|0\rangle^{\times m}|y_1 \oplus x_1 \oplus ... \oplus x_p\rangle...|y_c \oplus x_1 \oplus ... \oplus x_p\rangle \tag{9}$$

More precisely, $\forall i \in [p]$ let $t = (c+2)(i-1)$, choose a $j \in [q]$ such that $i \in W_j$, and set $g_{t+1} = g_{t+p+2} = (i, j+p)$. Then $\forall t' \in [p]$ set $g_{t'+t+1} = (j+p, t'+p+q)$.

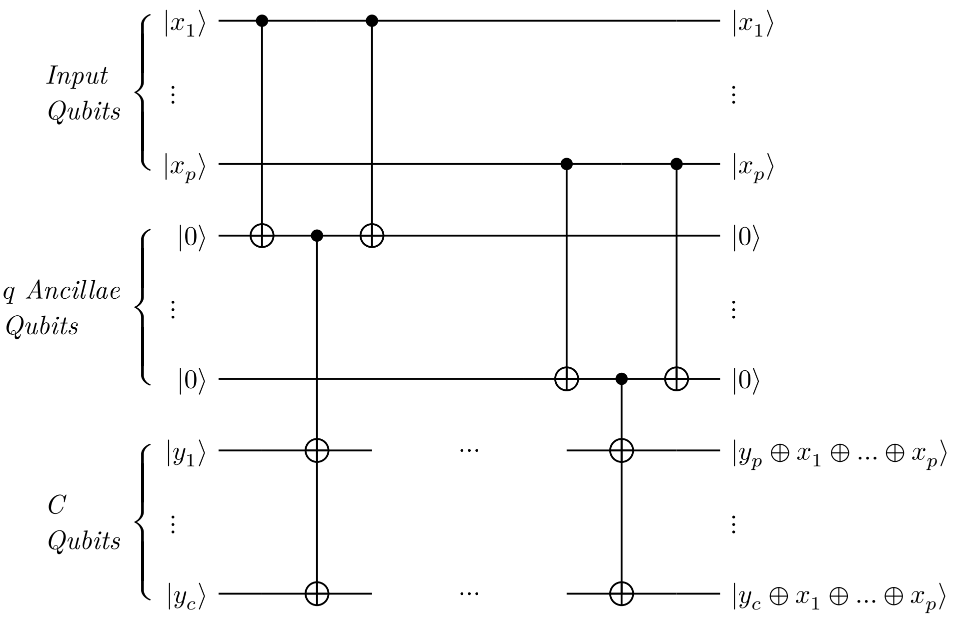


Figure 7: GC(S)M input circuit.

Let u be the output of this GC(S)M instance.

The claim is $\boldsymbol{u = kc + 2p}$:

- Upper bound $\boldsymbol{u \leq kc + 2p}$: assume without loss of generality that $\cup_{i=1}^{k} W_i = [p]$ and define:

$$T_i = W_i \setminus \left(\cup_{j=1}^{i-1} W_j\right) = \left\{t_1^i, ..., t_{|T_i|}^i\right\} \tag{10}$$

  Then we construct a circuit in the following way:
  - p gates prepare the i-th ancilla as the parity of each $x \in T_i$;
  - kc gates add the parity of first k ancillae (representing a set) to $y_i \forall i \in [c]$;
  - p gates restore the ancillae to their initial value.

  The constructed circuit is equivalent to the one in input and its total number of gates is $kp + 2p$, thus $u \leq kp + 2p$.
- Lower bound $\boldsymbol{u \geq kc + 2p}$: assume C is the smallest desired circuit. $\forall j \in [c]$, let V_j be the gates in C from *Ancillae Qubits* to y_j (in particular V_j contains the indexes of the *Ancillae Qubits* that have a gate to y_j). Note that each x_i can be added to y_j only through these gates (because of constraints S) and thus $x_i \in \cup_{z \in V_j} W_z$. Hence $[p] = \cup_{z \in V_j} W_z$. By the definition of k, $\forall j \ |V_j| \geq k$. Since each x_i must be added to *Ancillae Qubits* first in order to appear in $y_1, ..., y_c$, it must be added again to restore *Ancillae Qubits*, which contributes to at least $2p$ gates. Thus, $u \geq \sum_{j=1}^{c} |V_j| + 2p \geq kc + 2p$.

Now, suppose there exists a polynomial algorithm that approximates u within any constant factor. Let us call u' an approximation of u^* (the true minimum) within a constant factor obtained with a polynomial algorithm:

$$\frac{u^*}{\alpha} \leq u' \leq \alpha u^* \tag{11}$$

From u' an approximation k' of the optimum of rBSC k^* can be obtained:

$$k' = \frac{u' - 2p}{c} \tag{12}$$

The following bounds would be valid for the approximation:

$$\begin{gathered} 2p + ck' = u' \geq \frac{u^*}{\alpha} = \frac{2p + ck^*}{\alpha} \\ k^* \leq \alpha k' + 2\alpha\frac{p}{c} - 2\frac{p}{c} \end{gathered} \tag{13}$$

and

$$\begin{gathered} 2p + ck' = u' \leq \alpha u^* = \alpha(2p + ck^*) \\ k^* \geq \frac{k'}{\alpha} + \frac{2}{\alpha}\frac{p}{c} - 2\frac{p}{c} \end{gathered} \tag{14}$$

To show that k^* can be approximated within a constant factor, c it is set $c = p$:

$$\begin{gathered} \frac{k'}{\alpha} + \frac{2}{\alpha} - 2 \leq k^* \leq \alpha k' + 2\alpha - 2 \\ \Theta\left(\frac{k'}{\alpha}\right) \leq k^* \leq \Theta(\alpha k') \end{gathered} \tag{15}$$

This would mean that there exists a constant factor approximation for k for any constant factor α.

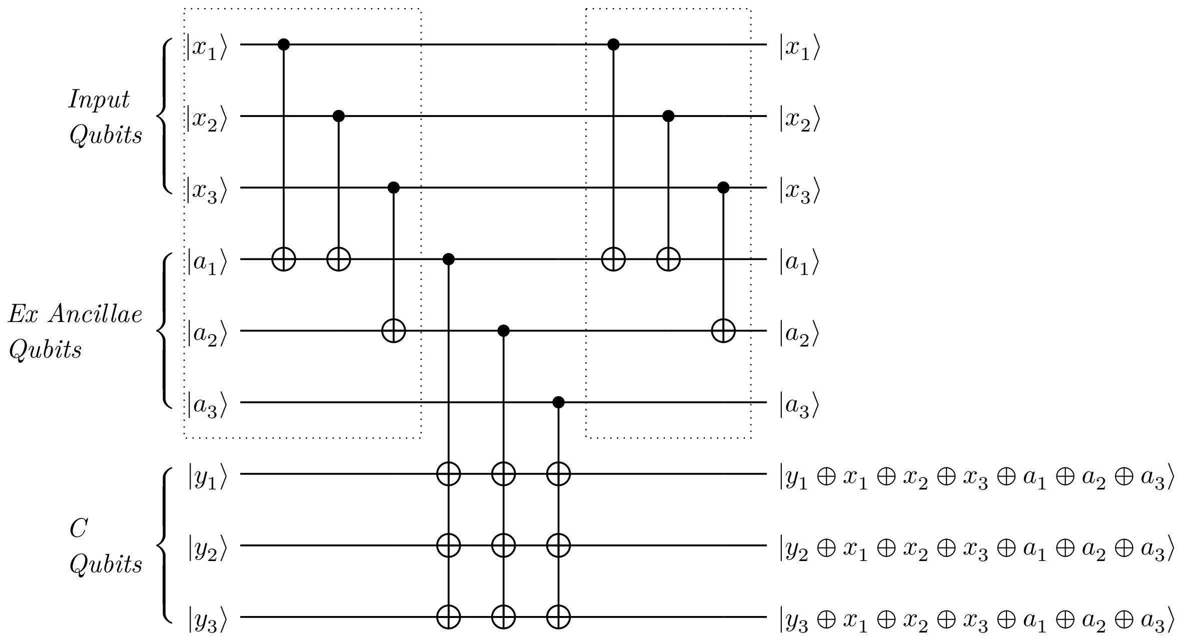
Since r is an arbitrary constant in rBSC, this approximation would allow to approximate the optimum of rBSC within a factor of $\ln r - O(\ln \ln r)$.

This violates Theorem 2.1.1. Thus, such a polynomial algorithm cannot exist. □

2.2 Global Constrained Size Minimization without Ancillae

Noticing that ancilla qubits are restored to their initial value, one could ask whether the problem without ancillae qubits is still NP-hard.

Example (rBSC $\preccurlyeq$ GC(S)M(w/oA)): Let us consider the same circuit introduced in the example above, but replacing ancillae qubits with input qubits.



Input Qubits
Ex Ancillae Qubits
C Qubits
$|x_1\rangle$
$|x_2\rangle$
$|x_3\rangle$
$|a_1\rangle$
$|a_2\rangle$
$|a_3\rangle$
$|y_1\rangle$
$|y_2\rangle$
$|y_3\rangle$
$|y_1 \oplus x_1 \oplus x_2 \oplus x_3 \oplus a_1 \oplus a_2 \oplus a_3\rangle$
$|y_2 \oplus x_1 \oplus x_2 \oplus x_3 \oplus a_1 \oplus a_2 \oplus a_3\rangle$
$|y_3 \oplus x_1 \oplus x_2 \oplus x_3 \oplus a_1 \oplus a_2 \oplus a_3\rangle$

Figure 8: Erroneous GC(S)M(w/oA) input circuit.

The fact that the *Ex Ancillae Qubits* are now input qubits changes the output of the circuit. The optimization of this circuit would not correspond to the initial rBSC problem because would require the chosen sets to be already coded in the output of the circuit.

To restore the desired output, the input circuit can be adjusted in the following way:

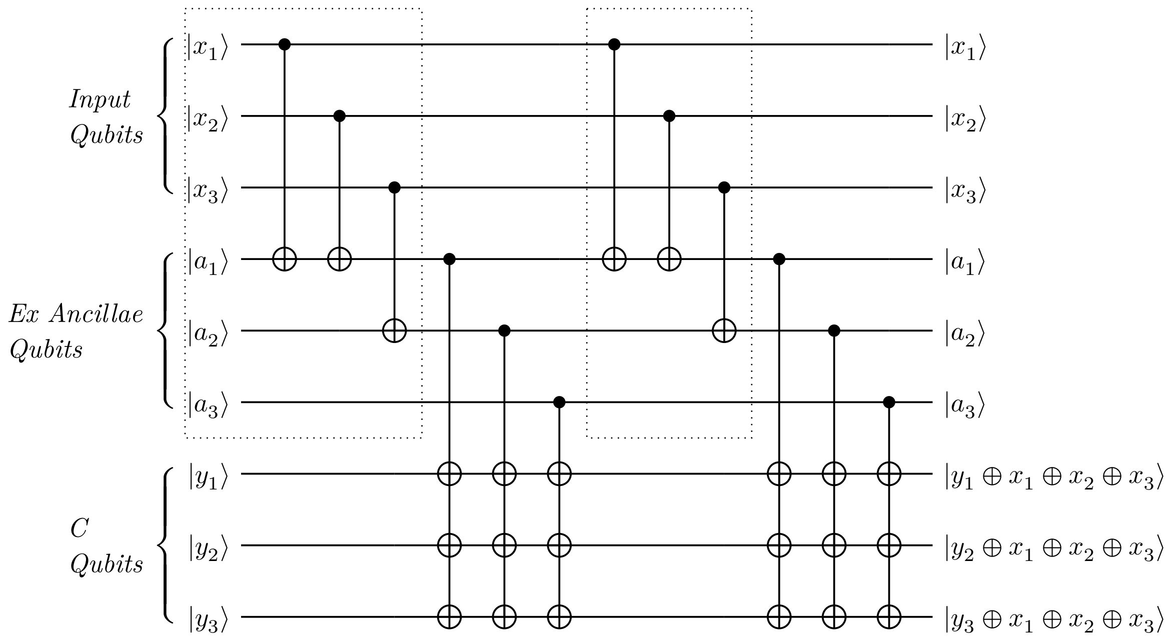


Figure 9: GC(S)M(w/oA) input circuit.

Following the reasoning of the last example, minimizing this circuit would result in:

$$u = 2kp + 2p \tag{16}$$

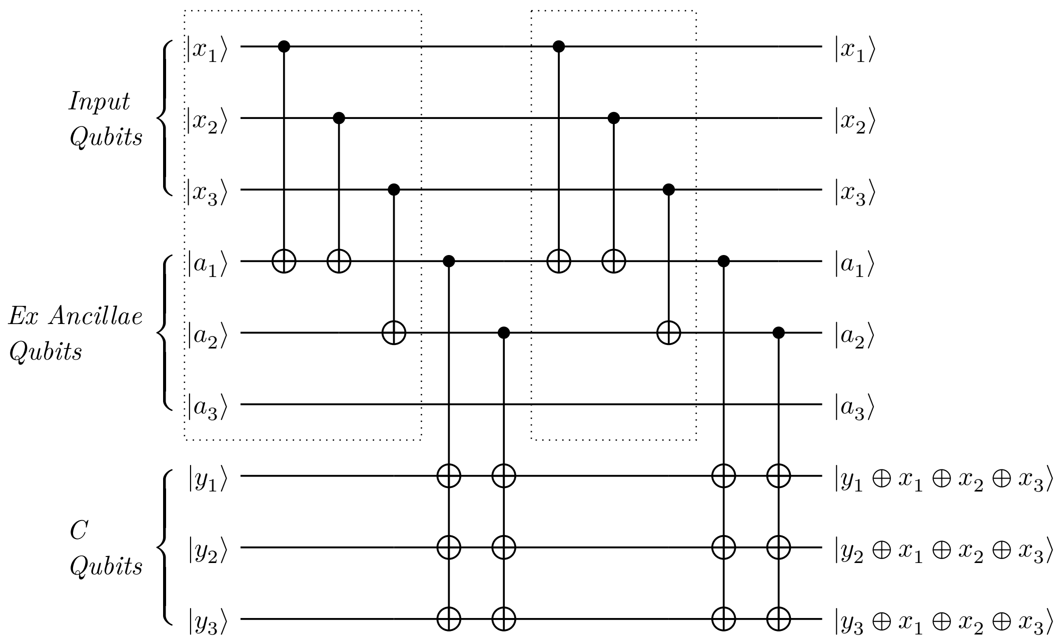


Figure 10: Optimized GC(S)M(woA) circuit.

Theorem 2.2.1 (GC(S)M(w/oA) is NP-hard): It is NP-hard to approximate the size version of GCM without *Ancillae Qubits* within any constant factor.

Proof *Theorem 2.2.1*: The proof is by **reduction from rBSC**.

Given in input $r, p, q, W_1, ..., W_q$ of an instance of rBSC, an input instance (i.e. a circuit that needs to be minimized) of GC(S)M is constructed such that:

- the circuit has $2p + q$ input qubits: p (*Input Qubits*) represent the elements of rBSC, p (*C Qubits*) codify the set cover, and the remaining q (*Ex Ancillae Qubits*) represent sets of rBSC. Formally, $n = 2p + q$;
- the circuit has $2p + 2p^2$ CNOT gates. Formally, $s = 2p + 2p^2$;
- constraints CNOT gates to be only:
  - from *Input Qubits* to *Ex Ancillae Qubits* representing the set they belong to. Formally, $S_1 = \{(x_i, a_j) | i \in W_j, j \in [q]\}$;
  - from any *Ex Ancillae Qubits* to any *C Qubit*. Formally, $S_2 = \{(a_j, y_z) | j \in [q], z \in [p]\}$.

  Therefore, the constraints are $S = S_1 \cup S_2$;
- the circuit is constructed such that the input is

$$|x_1...x_p\rangle|a_1...a_q\rangle^{\times m}|y_1...y_p\rangle \tag{17}$$

  and the output is

$$|x_1...x_p\rangle|a_1...a_q\rangle^{\times m}|y_1 \oplus x_1 \oplus ... \oplus x_p\rangle...|y_p \oplus x_1 \oplus ... \oplus x_p\rangle \tag{18}$$

  More precisely, $\forall i \in [p]$ let $t = (2p + 2)(i - 1)$, choose a $j \in [q]$ such that $i \in W_j$, and set $g_{t+1} = g_{t+p+2} = (x_i, a_j)$. Then $\forall z \in [p]$ set $g_{z+t+1} = g_{z+p+t+2} = (a_j, y_z)$.

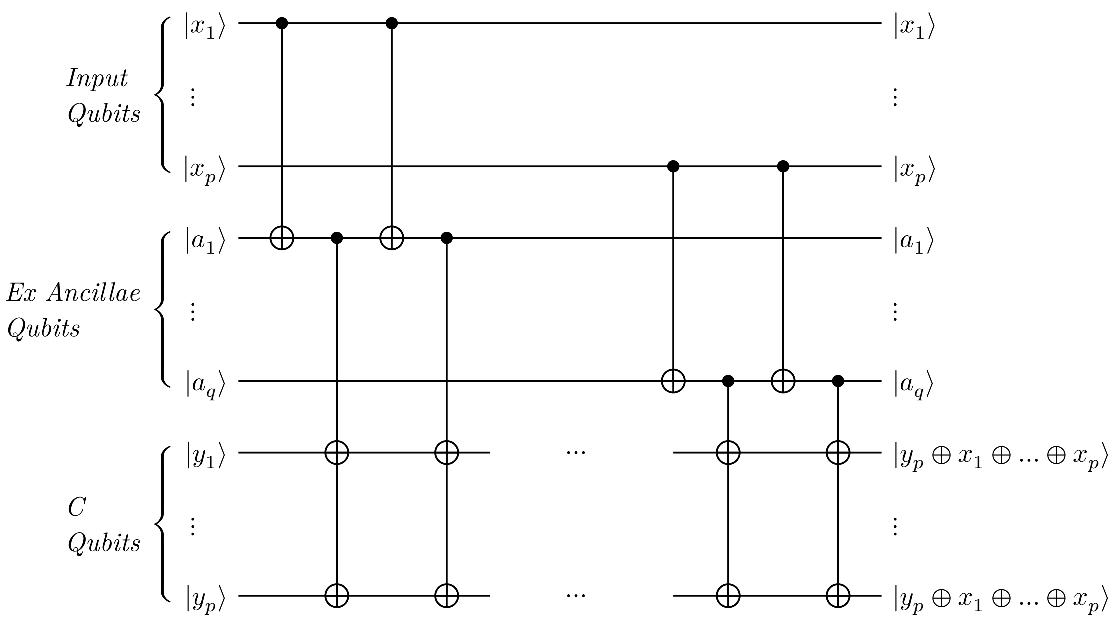


Figure 11: GC(S)M(w/oA) input circuit.

Let u be the output of this GC(S)M instance.

The claim is $\boldsymbol{u = 2kp + 2p}$:

- Upper bound $\boldsymbol{u \leq 2kp + 2p}$: assume without loss of generality that $\cup_{i=1}^{k} = W_i = [p]$ and define:

$$T_i = W_i \setminus \left(\cup_{j=1}^{i-1} W_j\right) = \left\{t_1^i, ..., t_{|T_i|}^i\right\} \tag{19}$$

  Then we construct a circuit in the following way:
  - p gates prepare the i-th *Ex Ancilla* as the parity of each $x \in T_i$;
  - kp gates add the parity of first k *Ex Ancillae* (representing a set) to $y_i \forall i \in [p]$;
  - p gates restore the *Ex Ancillae* to their initial value;
  - kp gates remove the parity of the first k *Ex Ancillae* from $y_i \forall i \in [p]$.

  The constructed circuit is equivalent to the one in input and its total number of gates is $2kp + 2p$, thus $u \leq 2kp + 2p$.
- Lower bound $\boldsymbol{u \geq 2kp + 2p}$. Assume C is the smallest desired circuit:
  - $\boldsymbol{2kp}$: $\forall j \in [p]$, let V_j be the gates in C from *Ex Ancillae Qubits* to y_j (in particular V_j contains the indexes of the *Ex Ancillae Qubits* that have at least a gate to y_j). Note that each x_i can be added to y_j only through these gates (because of constraints S) and thus $i \in \cup_{z \in V_j} W_z$. Hence $[p] = \cup_{z \in V_j} W_z$. By the definition of k, $\forall j\ |V_j| \geq k$. Since the output of the *C Qubits* does not contain the value of the *Ex Ancillae Qubits*, each gate from *Ex Ancillae Qubits* to *C Qubits* must be repeated an even number of times (i.e. there are at least 2 $|V_j|$ gates);
  - $\boldsymbol{2p}$: since each x_i must be added to *Ex Ancillae Qubits* first in order to appear in $y_1, ..., y_c$, it must be added again to restore *Ex Ancillae Qubits*, which contributes at least $2p$ gates.

  Thus, $u \geq \sum_{j=1}^{p} 2|V_j| + 2p \geq 2kp + 2p$.

Since $u = 2kp + 2p$ and r is a constant factor, can be proven that GC(S)M(w/oA) is NP-hard to approximate by following the same reasoning of Theorem 2.1.2. □

Observation 2.2.1: At a first glance seems that input qubits are more "powerful" then ancilla qubit. This is because in the proof of Theorem 2.1.2 ancillae qubits are restored (which in general isn't required). However, this is not the case: the problem of synthesizing a circuit with and without ancillae are two distinct problems.

Let us consider a circuit without ancillae qubits where:

- the input is:

$$|x_1, x_2, ..., x_n\rangle \tag{20}$$

- the output is:

$$|x_1 \oplus x_2, x_1 \oplus x_2, ..., x_n\rangle \tag{21}$$

Such a circuit cannot exist. It can be proved by looking at the matrix formulation of the problem:

$$\begin{pmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ 0 & 0 & & & \\ \vdots & \vdots & & I^{(n-2)\times(n-2)} & \\ 0 & 0 & & & \end{pmatrix} \tag{22}$$

Since it contains two linear dependent rows it cannot be turned into $I^{n\times n}$ by $\oplus$ row operations.

But as soon as an input qubit is replaced with an anicllae qubit the synthetization becomes trivial:

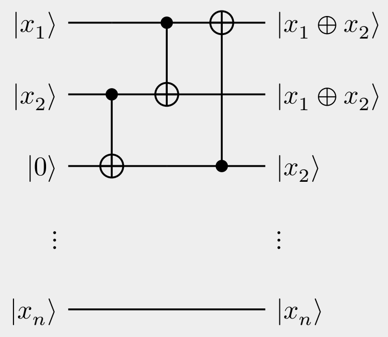
- the input becomes:

$$|x_1, x_2, ..., x_{n-1}\rangle|0\rangle \tag{23}$$

- the output becomes:

$$|x_1 \oplus x_2, x_1 \oplus x_2, ..., x_{n-1}\rangle| -\rangle \tag{24}$$

The synthesized circuit looks like this:



2.2.1 Matrix formulation

The problem variant GC(S)M(w/oA) can be expressed in matrix form by following the reasoning in Section 1.1.

Example (Matrix formulation): Let us express the reversible function expressed in the circuit of the previous example by constructing the matrix A such that it maps the input qubits $i = |x_1 ... x_p\rangle|0\rangle^{\times m}|y_1 ... y_p\rangle$ to the output $o = |x_1 ... x_p\rangle|0\rangle^{\times m}|y_1 \oplus x_1 \oplus ... \oplus x_p\rangle ... |y_p \oplus x_1 \oplus ... \oplus x_p\rangle$.

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{25}$$

Figure 12: GC(S)M(w/oA) circuit function in matrix form. Red rows are *Input Qubits*; blue rows are *Ex Ancillae Qubits*; green rows are *C Qubits*.

$$o = Ax \tag{26}$$

To complete the definition of the problem, the constraints' set S allows only $\oplus$ operations between rows that correspond to *type 1* and *type 2* operations.

The goal is finding the minimum number of allowed row operations that transforms A into the identity matrix I.

In general, any GC(S)M(w/oA) circuit function can be expressed as a matrix A with row constraints S.

$$A = \begin{pmatrix} I^{p\times p} & 0^{p\times q} & 0^{p\times p} \\ 0^{q\times p} & I^{q\times q} & 0^{p\times p} \\ 1^{p\times p} & 0^{p\times q} & I^{p\times p} \end{pmatrix}$$

where:

- $I^{i\times i}$ is the identity matrix with i rows and i columns;
- $0^{i\times j}$ is the matrix with all 0s with i rows and j columns;
- $1^{i\times j}$ is the matrix with all 1s with i rows and j columns.

2.3 Vertex cover $\preccurlyeq$ (Kang 2023)

A relaxation of the result Theorem 2.2.1 was already proven in (Kang 2023) where it is shown GC(S)M(w/oA) is NP-hard. Below, a revisited version that uses matrix formulation is presented.

Definition 2.3.1 (Vertex cover): A vertex cover V' of an undirected graph $G = (V, E)$ is a subset $V' \subset V$, 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;
- output: whether there exist a vertex cover $V' \subset V$ in G such that $|V'| \leq k$.

Theorem 2.3.1 (VCP is NP-complete): The vertex cover problem is NP-complete (Sipser 2013).

Theorem 2.3.2 (GC(S)M(w/oA) is NP-hard): Global constrained size minimization of circuits without ancillae qubits is NP-hard.

Proof: The proof is done by reduction VCP $\preccurlyeq$ GC(S)M(w/oA). Let us consider an input graph $G(V, E)$ of VCP such that:

- $V = [p]$;
- $E = \{(u_1, v_1), (u_2, v_2), ..., (u_q, v_q)\}$ for some $u_i, v_i \in [p]$[5].

Let $n = p + q + 1$. A GC(S)M(w/oA) instance with function matrix $n \times n$ A and $\oplus$ row operation constraints S is built from any VCP in the following way.

Let M be the matrix $p \times q$:

$$M = 0^{p\times q} + \sum_{i=1}^{q} e_{i,u_i,v_i} \tag{27}$$

where $e_{r,i,j}$ is the matrix $p \times q$ with all cells $= 0$ except the element on row r and column i, and the element on row r and column j that are $= 1$. Intuitively, every row of M represents an edge with its nodes set to 1.

$$A = \begin{pmatrix} 1 & & \\ 0 & & \\ \vdots & I^{p\times p} & 0^{p\times q} \\ 0 & & \\ 1 & & \\ \vdots & M & I^{q\times q} \\ 1 & & \end{pmatrix}$$

S allows only the following $\oplus$ row operations:

1. from the first row (red box) to any row in the blue box[6];
2. from any blue row, that represents a node in VCP, to the green rows that represent edges connected to it.

The claim is that the solution of this GC(S)M(w/oA) is $\boldsymbol{u = 2\tau + 2q}$ where τ is the solution of the starting VCP:

- Upper bound $\boldsymbol{u \leq 2\tau + 2q}$. Let V' be the minimum vertex cover of $G(V, E)$, let us define a reduction of A to I:
  1. for each edge (u_i, v_i)[5], $i \in [q]$ with both $u_i, v_i \in V'$, perform the $\oplus$ row operation $r_{1+p+i} \leftarrow r_{1+p+i} \oplus r_{1+u_i}$. Each of these operations remove a "1" in M;
  2. for each $u_i \in V'$ perform the $\oplus$ row operation $r_{1+u_i} \leftarrow r_{1+u_i} \oplus r_1$. These operations add a "1" in the first column of the blue box. In total these are τ operations;
  3. for each edge (u_i, v_i)[5], $i \in [q]$, perform any $\oplus$ row operation $r_{1+p+i} \leftarrow r_{1+p+i} \oplus r_{1+u_i}$ and $r_{1+p+i} \leftarrow r_{1+p+i} \oplus r_{1+v_i}$ that was not already performed in *step 1*. Each of these

[5]The couples (u_i, v_i) are unordered; i.e. if $(u_i, v_i) \in E$ then $(v_i, u_i) \notin E$.
[6]The phrase "$\oplus$ row operation from r_i to r_j" denotes the assignment $r_j \leftarrow r_i \oplus r_j$.

operations remove a "1" in M and a "1" in the first column. In total, combined with *step 1*, these are $2q$ operations (two for each edge);
4. repeat *step 2* to remove the "1"s in the first column of the blue box.

Correctness of this circuit (i.e. it maps the input to the expected output):

- the sub-matrix M becomes $0^{p \times q}$ because of *step 1* and *step 2* operations. Each couple operation (on each edge (u_i, v_i)) removes the parity of each e_{i,u_i,v_i} with $i \in [q]$;
- the first column of "1"s in the green box is removed because for each edge, the parity of the first row r_1 (red box) is added exactly once. To prove it, two cases are distinguished:
  - an edge (u_i, v_i) has both $u_i, v_i \in V'$. Then, the parity is added exactly once because $r_{1+p+i} \leftarrow r_{1+p+i} \oplus r_{1+u_i}$ is performed before $r_{1+u_i} \leftarrow r_{1+u_i} \oplus r_1$, while $r_{1+p+i} \leftarrow r_{1+p+i} \oplus r_{1+v_i}$ is performed after $r_{1+v_i} \leftarrow r_{1+u_i} \oplus r_1$;
  - an edge (u_i, v_i) has only $u_i \in V' \oplus v_i \in V'$. Without loss of generality let us consider the case $u_i \in V' \wedge v_i \notin V'$. Then, the parity is added exactly once because $r_{1+p+i} \leftarrow r_{1+p+i} \oplus r_{1+u_i}$ is performed after $r_{1+u_i} \leftarrow r_{1+u_i} \oplus r_1$, while $r_{1+v_i} \leftarrow r_{1+v_i} \oplus r_1$ is never performed.

  Note that an edge (u_i, v_i) with $u_i, v_i \notin V'$ cannot exist for how V' is defined.

The total number of $\oplus$ row operations is $2\tau + 2q$. Thus, it must holds that $u \leq 2\tau + 2q$.

- Lower bound $\boldsymbol{u \geq 2\tau + 2q}$:
  - let V_0 be the set containing the indexes of all target nodes of the row operations performed from the red row r_1. If $|V_0| < \tau$ then there exists some edge $i \in [q]$ such that $u_i, v_i \notin V_0$. This would mean that the first "1" in the i-th green row could not be removed: it is a contradiction. Thus, at least τ operations to **different** blue rows are needed. Since blue rows must be restored each operation must be repeated an even number of times. Therefore the total number of operations from red to blue rows must be $\geq 2\tau$;
  - the matrix M contains exactly $2q$ "1"s that needs to be removed. The only option to remove them is via $\geq 2q$ operations from blue to green rows.

  Thus, in total, at least $u \geq 2\tau + 2q$ operations are needed.

Therefore $u = 2\tau + 2q$, and GC(S)M(w/oA) is NP-hard. □

3 Conclusion

This thesis reviewed prior results on the complexity of the size minimization problem for CNOT circuits. Additionally, a novel result was proposed: it is NP-hard to approximate the size minimization problem of CNOT circuits (without ancillae qubits) under directed topological constraints. However, several fundamental questions in this area remain unanswered:

- Does the problem remain NP-hard when no topological constraints are imposed?
- Is the problem still NP-hard under undirected topological constraints? A conjecture regarding the NP-hardness of this case was proposed in (Kang 2023), but it remains an open problem.

Further research is needed to address these questions and deepen the understanding of the computational complexity of CNOT circuit optimization.

Bibliography

[1] J. Jiang, X. Sun, S.-H. Teng, B. Wu, K. Wu, and J. Zhang, "Optimal Space-Depth Trade-Off of CNOT Circuits in Quantum Logic Synthesis." [Online]. Available: https://arxiv.org/abs/1907.05087

[2] T. G. De Brugière, M. Baboulin, B. Valiron, S. Martiel, and C. Allouche, "Gaussian Elimination versus Greedy Methods for the Synthesis of Linear Reversible Circuits," *ACM Transactions on Quantum Computing*, vol. 2, no. 3, pp. 1–26, Sep. 2021, doi: 10.1145/3474226.

[3] D. P. DiVincenzo, "The Physical Implementation of Quantum Computation," *Fortschritte der Physik*, vol. 48, no. 9–11, pp. 771–783, Sep. 2000, doi: 10.1002/1521-3978(200009)48:9/11<771::aid-prop771>3.0.co;2-e.

[4] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, Jul. 1998, doi: 10.1145/285055.285059.

[5] L. Trevisan, "Non-approximability results for optimization problems on bounded degree instances," in *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, in STOC '01. Hersonissos, Greece: Association for Computing Machinery, 2001, pp. 453–461. doi: 10.1145/380752.380839.

[6] Y. Kang, "CNOT-Optimal Circuit Synthesis," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:260836008

[7] M. Sipser, *Introduction to the Theory of Computation*, Third. Boston, MA: Course Technology, 2013.