Systematic Optimization of Quantum Circuits via Toffoli Permutation and Local Search

G. Papakonstantinou

School of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece
papakon@cs.ntua.gr

***Abstract*—The synthesis of efficient reversible logic circuits is critical for fault-tolerant quantum computing. While standard libraries often treat the Toffoli gate as an atomic unit, its realization in the NCV library requires decomposition into 5 elementary gates. In this work, we propose a novel, systematic four-phase optimization methodology. First, we generate an optimal circuit in the strict NCT library (NOT, CNOT, Toffoli) using automated synthesis. Second, we replace each Toffoli gate with a specific NCV decomposition, permuting the control lines such that the final internal CNOT gate matches the subsequent linear gate in the circuit, allowing us to apply a cancellation rule ($X \cdot X = I$). Finally, post-synthesis local searches are applied using Mixed-Integer Non-Linear Programming (MINLP) techniques to further reduce the total gate count. Applying this method to the MIG, TSG, and MKG gates, we achieve Quantum Costs of 7, 14, and 12 respectively, matching or beating the best-known literature benchmarks.**

***Index Terms*—Reversible Logic, Gate Cancellation, NCV Library, Toffoli Decomposition, MIG Gate, TSG Gate, MKG Gate.**

I. Introduction

Reversible logic synthesis aims to minimize Quantum Cost (QC). While the NCT library is ideal for high-level synthesis, physical implementation often relies on the NCV library ($V, V^\dagger, CNOT$).

A standard Toffoli gate has a cost of 5 in the NCV library. However, standard substitutions often miss optimization opportunities at the boundaries between non-linear (Toffoli) and linear (CNOT) stages. We introduce a new methodology that exploits the permutability of Toffoli control lines and applies post-synthesis local search to reduce the effective cost of complex gates. This local search is based on the MINLP tool proposed in our previous work [8], integrating a new extension to include V-gates under certain conditions. Applying this methodology to the MIG, TSG, and MKG gates, we achieved Quantum Costs of 7, 14, and 12 respectively, matching or beating previously established literature benchmarks [7].

Reversible logic synthesis has been extensively studied. Barenco et al. [1] showed that universal quantum computation can be achieved using single-qubit gates and CNOTs. Maslov, Dueck, and Miller [2] developed template-based rewriting approaches for reversible circuit optimization, though these rely on static, predefined rules. More recently, automated exact synthesis has leveraged Mixed-Integer Programming (MIP) to find provably optimal quantum circuits [3]. However, pure exact methods face severe scalability limitations due to combinatorial explosion, rendering the global synthesis of complex, deep primitives (e.g., 10-15 gates) computationally intractable. To overcome this limitation, we propose a hybrid structural-optimization approach. Instead of attempting global synthesis, we combine, use of global synthesis only for the small NCT base, then apply human-like structural rules (Toffoli permutation/cancellation), then use of our previous solver [8], and finally a proposed extended solver on a Sliding Window of up to 6 gates. This allows us to dynamically generate optimal configurations for complex circuits while maintaining tractable computation times.

II. Methodology

The proposed optimization framework consists of four distinct phases.

A. Phase 1: Automated NCT Synthesis

The first step is to generate an optimal circuit topology within the **Strict NCT Library**. We utilize the MINLP-based computational synthesis framework described in our previous work [8]. This ensures that the base topology uses the minimum number of non-linear interactions (Toffoli gates) and positions them optimally relative to the linear gates, as shown in Fig. 2. It is noted that this solver can give many equivalent solutions.

B. Phase 2: Toffoli Permutation and Cancellation

We utilize a specific decomposition of the Toffoli gate ($C_1, C_2 \rightarrow T$) shown in Fig. 1. Crucially, this decomposition ends with CNOT($C_1 \rightarrow C_2$). The algorithm scans the NCT circuit for the pattern: Toffoli($x, y \rightarrow z$) followed by CNOT($x \rightarrow y$). By assigning the internal Toffoli controls $C_1 = x, C_2 = y$, the final internal CNOT cancels with the external CNOT ($X \cdot X = I$), reducing the Toffoli cost from 5 to 4.

C. Phase 3: Local Optimization 1

After Phase 2, the circuit consists of V-gates and CNOTs. We define a sliding window around the merged areas containing only CNOT gates (of any number of consecutive CNOT

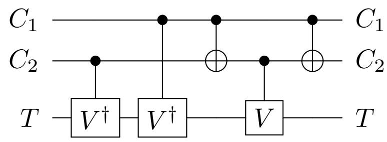


Fig. 1. The specific Toffoli decomposition used in Phase 2 ($C_1, C_2 \to T$). Note the final gate is CNOT($C_1 \to C_2$).

gates), and feed this sub-circuit back into the MINLP solver. This allows us to identify non-standard reductions that static template matching cannot find. The logical inputs and outputs of the sub-circuit boundaries are predefined by Phase 1.

D. Phase 4: Local Optimization 2

We define a sliding window (e.g., up to 5–6 gates) around the remaining merged areas and feed this sub-circuit into an extended version of the MINLP solver [8] (detailed in Section III). This phase identifies complex reductions, such as compressing a 6-gate block into 5 gates. A necessary condition for this reduction is that the relative phase between the first and last gates of the sliding window must be 0. This is enforced by imposing additional constraints (i.e., setting the imaginary part to 0) on the boundaries of the corresponding sub-circuit. Furthermore, the control lines must carry strictly binary values (0 or 1), and exactly one control line is permitted per gate. Due to the high computational complexity of the extended solver, the window length is practically limited up to 5–6 gates.

III. The Extended MINLP Solver

The NCV library consists of the following elementary gates:

- NOT: the Pauli-X operator,
- CNOT: controlled-X,
- V: the square root of X, satisfying $V^2 = X$,
- $V^\dagger$: the adjoint of V, satisfying $VV^\dagger = I$.

The matrix definitions are:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, V = \frac{1}{2}\begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix}, V^\dagger = \frac{1}{2}\begin{bmatrix} 1-i & 1+i \\ 1+i & 1-i \end{bmatrix}$$

A controlled version of a gate applies the above transformation only when the control value is $c = 1$.

The tool used to generate the decompositions is based on the framework introduced in [8], with several extensions making it suitable for NCV applications. First, each qubit line is represented by two real-valued variables (one complex number). This representation is an optimization embedding, not a physical interpretation. If the input to a (possibly controlled) single-qubit gate is

$$t = a + jb$$

and the output is

$$t' = a' + jb',$$

then the effect of the control is modeled as

$$t' = (1 - c)\, t + c\, y(t),$$

where $c \in \{0, 1\}$ is the control value and $y(t)$ is the complex output of the gate when active.

For the NCV gates, the complex functions $y(t)$ are defined:

$$X(t) = 1 - t,$$
$$V(t) = \frac{1-j}{2} + j\, t,$$
$$V^\dagger(t) = \frac{1+j}{2} - j\, t.$$

Substituting these expressions into the controlled update rule yields explicit real-valued propagation formulas for a' and b':

$$a' = \Re((1 - c)t + c\, y(t)), \qquad b' = \Im((1 - c)t + c\, y(t)).$$

These real-valued update equations are encoded directly in the MINLP formulation, allowing the solver to track the exact evolution of amplitudes through cascaded NCV gates. The resulting optimization problem is solved using a global MINLP engine [9], [10], guaranteeing the correctness of the obtained solutions for the different sub-circuits. Important constraints of the tool are that all inputs and outputs of the circuit or sub-circuit must have an imaginary part of 0 ($b = 0$), and the control lines must be strictly binary (0 or 1).

IV. Experimental Results

A. The MIG Gate (Cost 7)

The MIG gate is defined as: $P = A, Q = A \oplus B, R = AB \oplus C, S = AB' \oplus D$ [4].

Phase 1 (NCT): Our software generated the topology in Fig. 2(a), using 1 Toffoli and 4 CNOTs, achieving a strict NCT Cost of 9.

Phase 2 (Cancellation): The sequence Toffoli$(1, 2 \to 3)$ followed by CNOT$(1 \to 2)$ allows for perfect cancellation.

- Base Cost: 5 (1 Toffoli) + 4 (CNOTs) = 9.
- Reduction: 2 gates.
- **Final Cost: 7.** (See Fig. 3a).

B. The TSG Gate (Cost 14)

The TSG gate is defined as: $P = A$, $Q = A'C' \oplus B'$, $R = (A'C' \oplus B') \oplus D$, $S = (A'C' \oplus B')D \oplus (AB \oplus C)$ [5]. The general TSG gate ($D \neq 0$) is known for high complexity. Standard literature utilizing V-gates cites a cost of 17.

Phase 1 (NCT): The generated base circuit (Fig. 2b) uses 2 Toffoli gates and 7 CNOT gates, achieving a strict NCT Cost of 17.

Phase 2 (Cancellation): The sequence Toffoli$(1, 2 \to 4)$ followed by CNOT$(2 \to 1)$ allows for perfect cancellation after permuting lines 1 and 2 of the Toffoli gate (based on Fig. 1), reducing the cost by 2.

Phase 3 (Local Optimization 1): The linear block consisting of the last CNOT gate of the first Toffoli and the subsequent five CNOT gates was optimized down to 5 CNOT gates using the MINLP solver, reducing the cost by 1.

- Base Cost: 10 (2 Toffolis) + 7 (CNOTs) = 17.
- Reduction: $2 + 1 = 3$.
- **Final Cost: 14.** (See Fig. 3b).

C. The MKG Gate (Cost 12)

The MKG gate is defined as: $P = A$, $Q = C$, $R = (A'D' \oplus B') \oplus C$, $S = (A'D' \oplus B')C \oplus (AB \oplus D)$ [6]. The MKG gate is known for high complexity. Standard literature utilizing V-gates cites a cost of 17.

Phase 1 (NCT): The generated circuit (Fig. 2c) uses 2 Toffoli gates and 6 CNOT gates, achieving a strict NCT Cost of 16.

Phase 2 (Cancellation): The first Toffoli is followed immediately by a matching CNOT, reducing the cost by 2.

Phase 3 (Local Optimization 1): The linear block consisting of the last CNOT gate of the second Toffoli and the subsequent 3 CNOT gates was optimized to 3 CNOT gates using the MINLP solver, reducing the cost by 1.

Phase 4 (Local Optimization 2): The first block of 6 gates obtained after Phase 3 was optimized using the extended MINLP solver down to 5 gates, reducing the cost by 1.

- Base Cost: 10 (2 Toffolis) + 6 (CNOTs) = 16.
- Reduction: $2 + 1 + 1 = 4$.
- **Final Cost: 12.** (See Fig. 3c).

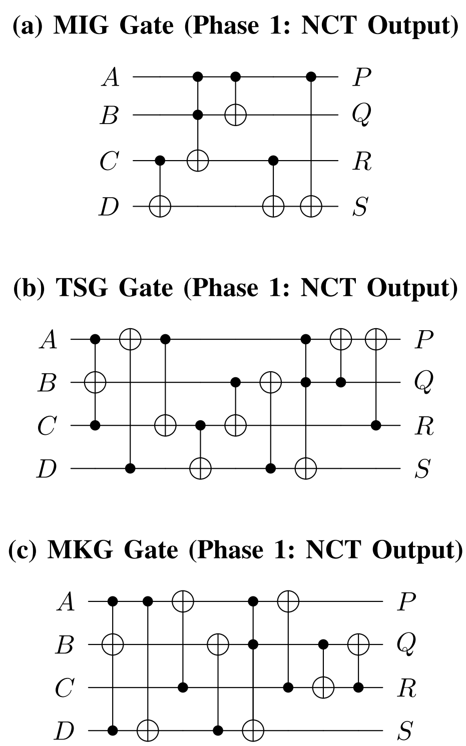
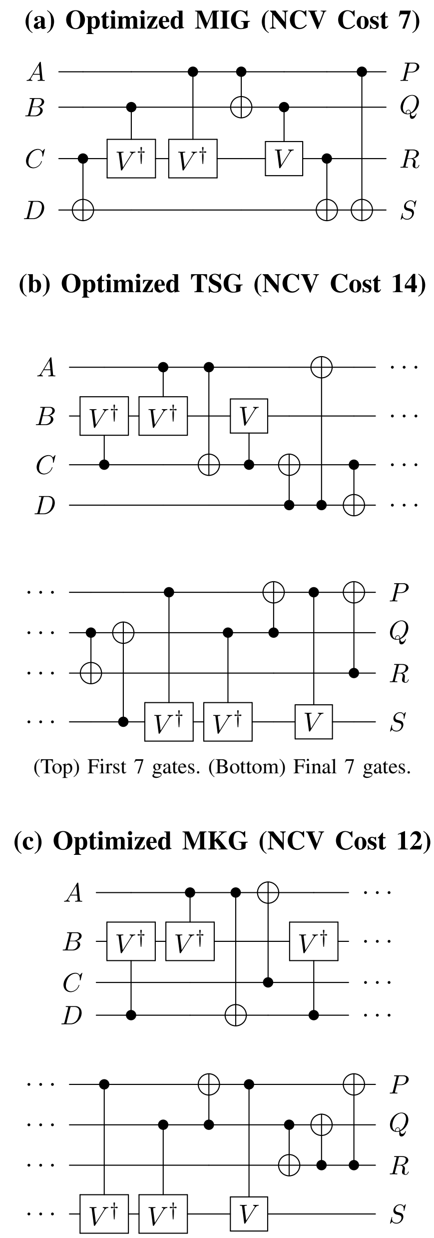


Fig. 2. The base NCT circuits generated by automated synthesis. These topologies expose specific Toffoli-CNOT adjacencies amenable to cancellation.



(Top) Simplified Toffoli + next linear. (Bottom) Reduced Toffoli + 3 Linear.

Fig. 3. NCV decompositions. (a) MIG Cost 7. (b) TSG Cost 14. (c) MKG Cost 12.

V. Comparison

Table I compares our results against standard literature values.

TABLE I
COMPARISON OF QUANTUM COSTS (NCV METRIC)

Gate	Standard Cost [7]	Proposed Cost	Improvement
MIG	7	**7**	Optimal
TSG	17	**14**	**17.6%**
MKG	17	**12**	**29.4%**

VI. Conclusion

This paper presented a generalized computational synthesis framework for reversible logic. By formulating the problem as an MINLP model in GAMS, we successfully automated the discovery of optimal gate realizations. The experimental results on several standard benchmarks confirm that the proposed algorithm systematically finds topologies that are comparable to or outperform human designs. By extending the optimization to the NCV library via a combination of Toffoli permutation, gate cancellation, and local search, we achieved a Quantum Cost of 7 for MIG, 14 for TSG, and 12 for MKG, significantly improving upon widely cited benchmarks.

REFERENCES

[1] A. Barenco et al., "Elementary gates for quantum computation," *Phys. Rev. A*, vol. 52, no. 5, pp. 3457-3467, 1995.

[2] D. Maslov, D. M. Miller, and G. W. Dueck, "Techniques for the synthesis of reversible Toffoli networks," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 8, no. 3, pp. 322-340, 2003.

[3] H. Nagarajan and Z. Szabó, "Provably Optimal Quantum Circuits with Mixed-Integer Programming," *arXiv preprint https://arxiv.org/abs/2510.00649v1*.

[4] H. M. H. Babu, et al., "Synthesis of a novel reversible fault tolerant reversible full adder," *IEEE Trans. VLSI*, 2006.

[5] H. Thapliyal and M. B. Srinivas, "Novel reversible TSG gate and its application for designing reversible carry look-ahead and other adder architectures," *ACSAC*, 2005.

[6] M. Haghparast and K. Navi, "A Novel Reversible Full Adder Circuit for Nanotechnology Based Systems," *J. Appl. Sci.*, vol. 7, no. 24, pp. 3995-4000, 2007.

[7] M. Surekha, "Efficient Approaches for Designing Quantum Costs of Various Reversible Gates," *Int. J. Eng. Studies*, Vol. 9, No. 1 (2017), pp. 57-78.

[8] G. Papakonstantinou, "Optimal Design of Reversible Circuits in the Strict NCT Library, *https://https://doi.org/10.36227/techrxiv.176827280.05224615/v2*.

[9] Gropp, W. and More, J. J. 1997, Optimization Environments and the NEOS Server, *Approximation Theory and Optimization*, M. D. Buhmann and A. Iserles, eds., Cambridge University Press, pages 167-182.

[10] M. Tawarmalani and N. V. Sahinidis, A polyhedral branch-and-cut approach to global optimization, *Mathematical Programming* **103**(2) (2005) 225-249.