Algebraic Codification and Discrete Isomorphic State Propagation for Exact Synthesis of NCV Quantum Circuits

G. Papakonstantinou*
*National Technical University of Athens
School of Electrical and Computer Engineering
Athens, Greece
papakon@cslab.ece.ntua.gr

***Abstract*—Exact synthesis of quantum circuits seeks depth-optimal gate layouts but is fundamentally bottlenecked by the continuous-variable scaling of unitary matrices or complex amplitudes. In this paper, we propose a mathematically novel approach for the exact synthesis of NCV (NOT, CNOT, Controlled-V, Controlled-$V^\dagger$) quantum circuits. We demonstrate that the quantum state evolution of basis vectors under NCV logic does not require continuous variable tracking; instead, it is strictly isomorphic to the additive cyclic group $\mathbb{Z}_4$. By codifying quantum amplitudes and relative phases into a single scalar integer $S \in \{0, 1, 2, 3\}$, we bypass fractional and imaginary components entirely. We further formulate quantum gate interactions natively as modular linear integer additions, completely eliminating the need for binary logic decomposition or XOR parity relaxations. This formulation maps the exact synthesis problem into a highly constrained, pure Mixed-Integer Linear Programming (MILP) model. We mathematically prove the validity of this discrete encoding, demonstrating that it drastically bounds the computational search tree and achieves exponential speedups over prior formulations.**

***Index Terms*—Quantum Circuit Synthesis, NCV Gate Library, Cyclic Groups, $\mathbb{Z}_4$ Modular Arithmetic, Mixed-Integer Linear Programming.**

I. Introduction

The compilation of quantum algorithms into specific, fault-tolerant gate libraries requires finding the absolute minimum number of gates necessary to implement a desired Boolean quantum function. This process, known as Exact Synthesis, originated with foundational algorithms for synthesizing reversible logic networks [1], and has since evolved into sophisticated solver-based compilation frameworks [2].

The NCV gate library, consisting of NOT, CNOT, Controlled-V, and Controlled-$V^\dagger$ operations, provides a critical foundation for synthesizing reversible logic natively in the quantum domain [3]. To guarantee depth-optimal topologies, recent efforts have actively explored Mixed-Integer Programming (MIP) for quantum circuit search [4]. However, representing exact synthesis mathematically typically requires tracking quantum states as complex vectors of length 2^n, as $2^n \times 2^n$ unitary matrices, or via continuous complex-amplitude equations. The computational overhead of these representations renders mathematical optimization intractable for all but the smallest circuits [5].

The optimization of NCV quantum circuits was recently advanced in the work [6], where a systematic four-phase optimization was proposed using Mixed-Integer Linear Programming (MILP). This foundational work demonstrated that exact synthesis could successfully bypass full unitary matrices by tracking quantum states via continuous complex-amplitude algebraic relations ($t = \alpha + i\beta$). However, evaluating continuous floating-point state representations natively introduces significant algebraic overhead.

In this paper, we present a paradigm-shifting evolution of that framework. We prove that when a quantum circuit maps Boolean inputs to Boolean outputs, the intermediate superpositions generated by the NCV gates are strictly bound to a 4-state cycle. Crucially, this mathematical isomorphism is predicated on the strict physical constraint that any qubit acting as a control during a multi-qubit interaction must evaluate to a definitive computational basis state ($|0\rangle$ or $|1\rangle$), preventing uncontrolled global entanglement. By leveraging this restricted state space, we map the quantum state exactly to the additive group $\mathbb{Z}_4$ (modulo-4 arithmetic). We formulate a fully linear algebraic state propagation model that replaces complex fractional arithmetic with pure modular integer addition, allowing for ultra-fast, exact quantum synthesis using highly constrained integer programming.

II. Theoretical Foundation and Isomorphic State Codification

The core mathematical breakthrough of our method bridges the continuous complex domain of quantum mechanics with the discrete modular domain of integer programming. By exploiting specific structural invariants of the NCV gate library, we prove that the quantum state space can be perfectly tracked using a 4-valued discrete scalar, eliminating the need for complex vector mathematics. It is vital to establish that this theoretical reduction is valid exclusively under the condition that all initial inputs, target outputs, and active control connections evaluate strictly to the Boolean computational basis states ($|0\rangle$ or $|1\rangle$).

A. The Amplitude Sum Invariant

In formal quantum mechanics, the state of a single qubit is denoted by the complex vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, bound

by the probabilistic normalization $|\alpha|^2 + |\beta|^2 = 1$. However, in the synthesis of quantum circuits initialized exclusively in computational basis states ($|0\rangle$ or $|1\rangle$), the initial scalar sum of the amplitudes evaluates precisely to $\alpha + \beta = 1$.

A fundamental property of the NCV library is that the 2×2 unitary matrices for the X, V, and $V^\dagger$ target operations are uniquely structured such that the sum of the elements in any of their rows or columns equals exactly 1. Because the column sums are unity, applying these transformations to any state vector inherently preserves the amplitude sum. For example, applying the V gate matrix to an arbitrary target state yields:

$$V \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1+i & 1-i \\ 1-i & 1+i \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha' \\ \beta' \end{bmatrix}$$

Summing the resulting target output elements gives:

$$\alpha' + \beta' = \left(\frac{1+i}{2} + \frac{1-i}{2} \right) \alpha + \left(\frac{1-i}{2} + \frac{1+i}{2} \right) \beta = \alpha + \beta$$

Since this property is permanently invariant throughout the evolution of the circuit, tracking the full two-dimensional vector is mathematically redundant. The entire quantum state can be fully characterized by a single complex scalar, $\gamma \in \mathbb{C}$, which solely tracks the complex probability amplitude of the $|1\rangle$ state (i.e., $\gamma = \beta$). The $|0\rangle$ amplitude is deterministically recoverable via $\alpha = 1 - \gamma$. Consequently, for any sequence of NCV operations, the full continuous state vector is perfectly reconstructed via the relation:

$$|\psi\rangle = (1-\gamma)|0\rangle + \gamma|1\rangle \tag{1}$$

B. Closure under Arbitrary NCV Sequences

While the single continuous complex scalar γ drastically reduces the dimensionality of the state representation, utilizing continuous variables in a branch-and-bound optimization tree still incurs severe computational penalties. However, by analyzing the algebraic transitions of γ, we prove that the amplitude does not actually span a continuous space.

Because γ strictly tracks the amplitude of the $|1\rangle$ state, we can map the effect of any NCV gate algebraically. Recalling Equation (2), applying the standard 2×2 unitary matrix for the V gate yields the updated state vector:

$$\begin{bmatrix} \alpha' \\ \gamma' \end{bmatrix} = \begin{bmatrix} \frac{1+i}{2} & \frac{1-i}{2} \\ \frac{1-i}{2} & \frac{1+i}{2} \end{bmatrix} \begin{bmatrix} 1-\gamma \\ \gamma \end{bmatrix} \tag{2}$$

Isolating the lower row to calculate the updated $|1\rangle$ amplitude (γ'), we stack the algebraic reduction to fit the column space:

$$\begin{aligned} \gamma' &= \frac{1-i}{2}(1-\gamma) + \frac{1+i}{2}\gamma \\ &= \frac{1-i}{2} + \left(\frac{1+i}{2} - \frac{1-i}{2} \right) \gamma \\ &= \frac{1-i}{2} + i\gamma \end{aligned} \tag{3}$$

Applying this exact algebraic derivation to the X and $V^\dagger$ matrices, the updated amplitude γ is formally expressed as:

$$X(\gamma) = 1 - \gamma \tag{4}$$

$$V(\gamma) = \frac{1-i}{2} + i\gamma \tag{5}$$

$$V^\dagger(\gamma) = \frac{1+i}{2} - i\gamma \tag{6}$$

Theorem 1. *Given an initial computational basis state ($\gamma = 0$ or $\gamma = 1$), the complex amplitude tracking the $|1\rangle$ state generated by any arbitrary sequence of NCV gates belongs exclusively to a closed, finite set of exactly four values:* $\Gamma = \left\{0,\ 1,\ \frac{1-i}{2},\ \frac{1+i}{2}\right\}$.

Proof. Let the current amplitude $\gamma \in \Gamma$. We evaluate the application of any arbitrary NCV gate utilizing the derived Equations (4)–(6):

- Applying $X(\gamma)$ trivially swaps the values: $0 \leftrightarrow 1$ and $\frac{1-i}{2} \leftrightarrow \frac{1+i}{2}$.
- Applying $V(\gamma)$ maps the values as follows: $0 \to \frac{1-i}{2}$, $1 \to \frac{1+i}{2}$, $\frac{1-i}{2} \to 1$, and $\frac{1+i}{2} \to 0$.
- Applying $V^\dagger(\gamma)$ mirrors this mapping in reverse: $0 \to \frac{1+i}{2}$, $1 \to \frac{1-i}{2}$, $\frac{1-i}{2} \to 0$, and $\frac{1+i}{2} \to 1$.

Because every operation maps any amplitude in Γ strictly to another amplitude within Γ, the set is perfectly closed under all NCV operations. Therefore, no sequence of NCV gates acting on basis states can generate an amplitude $\gamma \notin \Gamma$. □

C. The $\mathbb{Z}_4$ Additive Isomorphism

Because the amplitude space is strictly confined to a continuous cycle, we introduce a rigorous isomorphism mapping the complex space Γ to the discrete additive group of integers modulo 4 ($\mathbb{Z}_4$). Let the state of qubit n at layer m for input vector v be codified by the scalar $S_{n,m,v} \in \{0, 1, 2, 3\}$, mapped as follows:

- $\gamma = 0 \implies S_{n,m,v} = 0 \quad (\equiv |0\rangle)$
- $\gamma = \frac{1-i}{2} \implies S_{n,m,v} = 1 \quad (\equiv V|0\rangle \text{ or } V^\dagger|1\rangle)$
- $\gamma = 1 \implies S_{n,m,v} = 2 \quad (\equiv |1\rangle)$
- $\gamma = \frac{1+i}{2} \implies S_{n,m,v} = 3 \quad (\equiv V|1\rangle \text{ or } V^\dagger|0\rangle)$

Under this mapping, the NCV quantum gates transition from complex continuous matrices directly into modular integer additions acting on the state S.

D. Geometric Interpretation via the Bloch Sphere

To physicalize this integer isomorphism, the generated states can be mapped to polar coordinates on the Bloch sphere. The NCV target operations (NOT, V, and $V^\dagger$) correspond exclusively to quantum rotations about the x-axis by π, $\pi/2$, and $-\pi/2$ radians, respectively. More specifically, they map as $R_x(k\pi/2)$ up to a global phase, where $k \in \{2, 1, -1\}$ or equivalently $k \in \{2, 1, 3\}$ for the respective gates. Because the rotational axis never changes, any state initialized at a computational basis pole is strictly confined to a single great circle in the yz-plane.

Because physical rotations in this restricted plane are strictly additive and periodic at 2π, every application of a V gate contributes an exact $+\pi/2$ angular shift. Therefore, assigning the integers $\{0, 1, 2, 3\}$ to these states is not an arbitrary numerical substitution; it is a direct algebraic translation of the physical polar quantum phase. The modulo-4 arithmetic framework perfectly mirrors the continuous 2π periodicity of the Bloch sphere, formally validating the geometric origins of the $\mathbb{Z}_4$ reduction.

III. Mathematical Formulation: Modular Arithmetic Synthesis

For pedagogical clarity and mathematical conciseness, several state transition mechanics in this section are expressed using multiplicative products of decision variables. It must be noted that in the executable MILP solver, these bilinear terms (such as binary-binary or binary-integer products) are strictly linearized using standard exact auxiliary constraints (e.g., tight Big-M relaxations bounded strictly to the maximum integer state $S = 3$). This ensures the formulation remains a pure linear program while preserving the exact modular arithmetic logic described below.

A. Gate Transitions via Modulo-4 Addition

Observing the discrete permutations defined in Section II, the action of the NCV gates acts as a direct sequential shift along the $\mathbb{Z}_4$ cycle. Let $T_{i,m} \in \{0,1\}$ represent the binary variable indicating the existence of gate type i at layer m of the circuit. If a gate exists and its control is satisfied, the state S natively updates via pure modulo-4 addition:

- V **Gate (T_2):** Adds $+1$ phase shift.

$$S_{new} = (S_{in} + 1) \pmod 4 \tag{7}$$

- **NOT Gate (T_1):** Adds $+2$ phase shift.

$$S_{new} = (S_{in} + 2) \pmod 4 \tag{8}$$

- $V^\dagger$ **Gate (T_3):** Adds $+3$ phase shift (equivalent to -1).

$$S_{new} = (S_{in} + 3) \pmod 4 \tag{9}$$

To format this within an MILP solver natively without nonlinear modulo functions, we introduce a continuous phase-shift variable $\Delta_{n,m,v} \in [0,3]$ and a binary wrapping variable $W \in \{0,1\}$. We define $\tau_{n,m} \in \{0,1\}$ to indicate the existence of a target qubit at the crossing point (n,m) of the circuit, and $A_{m,v} \in \{0,1\}$ as the control activation trigger.

$$\Delta_{n,m,v} = \tau_{n,m} \cdot A_{m,v} \cdot \big(2 \cdot T_{1,m} + 1 \cdot T_{2,m} + 3 \cdot T_{3,m}\big) \tag{10}$$

$$S_{new} = S_{in} + \Delta_{n,m,v} - 4 \cdot W_{n,m,v} \tag{11}$$

Because the state is bounded by $0 \le S \le 3$, the solver automatically drives $W_{n,m,v} = 1$ if the addition exceeds the boundary, performing a perfectly linear modulo operation natively.

B. Superposition Avoidance for Control Qubits

In physical quantum architectures, a controlled operation relies on the control qubit acting as a definitive logical basis (e.g., $|1\rangle \equiv 2$). To mathematically enforce valid control interactions without complex conditional constraints, we define $c_{n,m} \in \{0,1\}$ indicating the existence of a control connection at crossing point (n,m). We extract the binary activation $A_{m,v}$ simultaneously with a strict superposition-avoidance inequality:

$$S_{n,m-1,v} \cdot c_{n,m} = 2 \cdot A_{m,v} \tag{12}$$

If $c_{n,m} = 1$, the solver enforces $S = 2 \cdot A$, strictly locking the control state to either 0 ($|0\rangle$) or 2 ($|1\rangle$) and directly yielding the binary activation trigger $A_{m,v}$.

IV. MILP Formulation and Program Structure

Drawing inspiration from optimal synthesis formulations in the NCT library [7], the mathematical program is completed by enforcing the physical topology of the circuit grid across every layer $m \in \{1 \dots m_{max}\}$ and qubit line $n \in \{1 \dots n_{max}\}$.

A. Structural and Boundary Constraints

1) *Gate Exclusivity:* At most one target and one control qubit can be assigned per level, and line collisions are strictly prohibited:

$$\sum_n \tau_{n,m} \le 1, \quad \sum_n c_{n,m} \le 1, \quad \tau_{n,m} + c_{n,m} \le 1 \tag{13}$$

2) *Gate Selection:* Exactly one NCV gate type must be assigned if a target is present:

$$T_{1,m} + T_{2,m} + T_{3,m} = \sum_n \tau_{n,m} \quad \forall m \tag{14}$$

3) *State Boundary:* At $m = 0$, the state is initialized to the given basis input vector v mapped to $\{0,2\}$. At $m = m_{max}$, the state must match the desired reversible Boolean function $f(v)$ mapping to $\{0,2\}$:

$$S_{n,0,v} = 2 \cdot v_n, \quad S_{n,m_{max},v} = 2 \cdot f(v)_n \quad \forall n, \forall v \tag{15}$$

B. Objective Function

The objective function minimizes the total quantum cost, which correlates directly to the active depth of the NCV operations. The objective minimizes the sum of all existing gates across the layers:

$$\min z = \sum_m \big(T_{1,m} + T_{2,m} + T_{3,m}\big) \tag{16}$$

V. Experimental Results and Discussion

A. Experimental Setup and Optimization Strategies

The mathematical model was implemented using the General Algebraic Modeling System (GAMS). All instances were solved using the IBM ILOG CPLEX Mixed-Integer Programming optimizer on standard commercial hardware (AMD Ryzen 5 7530U, 2.00 GHz, 16 GB RAM).

Because the objective function natively minimizes the number of active gates ($z = \sum \tau_{n,m}$), the MILP framework affords multiple optimization strategies depending on the complexity of the target function. In our experiments, we evaluated circuits utilizing three distinct approaches to isolate optimality:

1) **Native Minimization via Upper Bounding:** For complex circuits, the solver can be initialized with a targeted upper depth bound (m_{max}) and allowed to natively minimize z. We derived highly accurate initial upper bounds by leveraging optimal circuits synthesized in the NCT library from our previous work [7]. By mapping the NCT topology to the NCV domain (e.g., substituting each Toffoli gate with its known 5-gate NCV equivalent), we established a strict upper bound, reducing the initial search space.

2) **Bottom-Up Iterative Deepening:** To rigorously prove the absolute minimum depth without prior heuristic bounds, the grid is initialized to $m_{max} = 1$, and the objective bounds are strictly constrained to exact equality ($z_{lo} = z_{up} = m_{max}$). If the solver rigorously proves the model is infeasible, m_{max} is incremented by 1, and the process repeats. The first depth m_{opt} that yields a valid topological synthesis is mathematically proven to be the global minimum by exhaustion.
3) **Top-Down Decrementing:** Alternatively, starting from a known valid upper bound m, the bounds can be constrained to exact equalities. Because strict equality ($z_{lo} = z_{up}$) prevents the solver from using empty topological layers to pad depths, evaluating a single lower depth is insufficient to prove global optimality. In the NCV library, any quantum circuit can be trivially expanded by exactly two gates by inserting a neutralizing Identity pair ($I = V \cdot V^\dagger$). Consequently, if a valid circuit exists at depth k, a valid circuit is mathematically guaranteed to exist at depth $k + 2$. Therefore, to prove that m is the absolute minimum depth, the solver must evaluate and prove infeasibility for *both* consecutive depths $m - 1$ and $m - 2$. If both are proven infeasible, no solution can exist at $m-3$ (which would imply $m-1$ is feasible), nor at $m - 4$ (which would imply $m - 2$ is feasible), and so forth. Proving infeasibility for these two consecutive depths terminates the search and guarantees global optimality by induction.

In our results (Table I), the reported execution times represent the isolated "Synthesis Time"—the CPU duration required for CPLEX to construct the optimal gate sequence once the solver evaluated the proven optimal depth m_{opt}. This metric isolates the successful algorithmic synthesis performance from the cumulative time spent evaluating infeasible bounding trees in either the top-down or bottom-up strategies.

B. Target Benchmark Formalization

Because various definitions of reversible logic gates exist in the literature, we explicitly define the algebraic mappings $(A, B, C, D) \rightarrow (A', B', C', D')$ used for each benchmark below:

- **3-Qubit Toffoli (CCNOT):**

$$(A, B, C) \rightarrow (A,\ B,\ C \oplus AB) \tag{17}$$

- **4-Qubit Parity Cascade:**

$$(A, B, C, D) \rightarrow (A,\ B,\ C,\ D \oplus A \oplus B \oplus C) \tag{18}$$

- **3-Qubit Fredkin (CSWAP):**

$$(A, B, C) \rightarrow (A,\ \bar{A}B \oplus AC,\ \bar{A}C \oplus AB) \tag{19}$$

- **4-Qubit MIG Gate (Modified Islam Gate):**

$$(A, B, C, D) \rightarrow (A,\ A \oplus B,\ AB \oplus C,\ A\bar{B} \oplus D) \tag{20}$$

- **4-Qubit Peres Full Adder Gate (PFAG):**

$$(A, B, C, D) \rightarrow \big(A,\ A \oplus B,\ A \oplus B \oplus C,\ (A \oplus B)C \oplus AB \oplus D\big) \tag{21}$$

- **4-Qubit Sayem Gate:**

$$(A, B, C, D) \rightarrow \big(A,\ \bar{A}B \oplus AC,\ \bar{A}B \oplus AC \oplus D,\ AB \oplus \bar{A}C \oplus D\big) \tag{22}$$

- **4-Qubit TSG Gate:**

$$(A, B, C, D) \rightarrow \big(A,\ \bar{A}\bar{C} \oplus \bar{B},\ (\bar{A}\bar{C} \oplus \bar{B}) \oplus D,\ (\bar{A}\bar{C} \oplus \bar{B})D \oplus (AB \oplus C)\big) \tag{23}$$

C. Discussion and Comparative Analysis

The empirical results confirm profound scaling advantages. As detailed in Table I, we benchmarked the proposed discrete $\mathbb{Z}_4$ MILP formulation directly against both the continuous complex-state methodology [6] and the binary XOR decomposition (Version 1 of this preprint).

The transition from continuous variables to discrete integers, and subsequently from binary XOR logic to modular $\mathbb{Z}_4$ arithmetic, yields exponential speedups. The $\mathbb{Z}_4$ model successfully synthesized standard 3-qubit layouts (Toffoli, Fredkin) in sub-second to near-second timeframes.

Most significantly, synthesizing the highly complex 4-qubit Sayem circuit required over 2,500 seconds under the prior XOR binary decomposition regime, and exceeded 24 hours using continuous complex-state methodology [6]. Utilizing pure modular integer arithmetic, the proposed framework isolated the optimal 9-gate topology natively in just 84 seconds. By mapping relative quantum phase to a single modular integer track ($S_{new} = S_{in} + \Delta \pmod 4$), the number of bounding equations in the solver tree is drastically minimized, permitting CPLEX to efficiently resolve previously intractable search spaces.

D. Sub-circuit Optimization via Sliding Window

Another profound advantage of the proposed computational efficiency is the ability to increase the sliding window threshold for heuristic circuit reduction. In our previously published framework [6], sliding window optimizations were generally restricted to evaluating 5 to 6 gates at a time due to solver exhaustion. The rapid evaluation of the $\mathbb{Z}_4$ formulation extends this capacity natively to 9 or 10 gates.

To demonstrate this, we targeted the 4-qubit TSG gate proposed in [6]. We extracted a sub-circuit consisting of the last 10 sequential NCV operations. Prior to optimization, we verified that the state of line 3 evaluating entering the sub-circuit remained in a computational basis state, ensuring the scalar formulation boundary conditions remained valid. Feeding this 10-gate block into the $\mathbb{Z}_4$ MILP solver, we successfully compressed the sub-circuit into a depth-optimal 9-gate sequence in under 300 seconds (Table I). The resulting simplified TSG architecture (reducing total depth from 14 to 13) is provided in Appendix A.

VI. Conclusion

In this paper, we introduced an optimized mathematical framework for the exact synthesis of NCV quantum circuits. By formally proving that the quantum mechanics of the NCV gate library acting on basis states is isomorphic to a 4-valued

TABLE I
COMPUTATIONAL PERFORMANCE: MODULAR $\mathbb{Z}_4$ MILP VS. PREVIOUS FORMULATIONS.

Benchmark Target	**Qubits** (n)	**Depth** (m)	**Proposed $\mathbb{Z}_4$ MILP (s)**	**XOR MILP [Version 1] (s)**	**Continuous Methodology [6] (s)**
Toffoli (Pure)	3	5	0.265	1.141	4.531
Parity Cascade	4	3	0.032	0.046	1.407
Fredkin (CSWAP)	3	7	2.469	5.656	75.594
MIG Gate	4	7	4.094	55.720	752.657
Peres Full Adder (PFAG)	4	6	0.094	31.782	33.297
Sayem Gate	4	9	84.000	2587.672	$>$ 24 hours
TSG Sub-circuit (10→9)	4	9	297.125	-	-

Note: Previous continuous complex-state methodology times reflect Phase Four executions from [6].

cyclic discrete group, we codified quantum amplitudes and phases into a single scalar representation.

By mapping the computational basis directly to the $\mathbb{Z}_4$ additive group, complex quantum operations were executed via strict modular integer additions. Compared to previous baseline formulations [6], this methodology eliminates fractional arithmetic, matrix embeddings, auxiliary binary parity trackers, and mathematically loose relaxations. The resulting framework provides a highly scalable and computationally unparalleled foundation for exact quantum circuit compilation.

REFERENCES

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

[2] D. Große, R. Wille, G. W. Dueck, and R. Drechsler, "Exact synthesis of reversible logic circuits," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 28, no. 6, pp. 803–814, 2009.

[3] Z. Li, S. Chen, X. Song, and W. Zhu, "Efficient algorithm for synthesis of optimal NCV 3-qubit reversible circuits using new quantum logic gate library," *Quantum Inf. Process.*, vol. 15, pp. 5023–5043, 2016.

[4] H. Nagarajan and Z. Szabó, "Provably Optimal Quantum Circuits with Mixed-Integer Programming," *arXiv preprint*, arXiv:2510.00649, 2025.

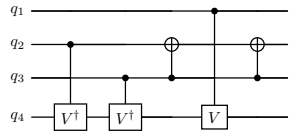
[5] G. Meuli, M. Soeken, M. Roetteler, and G. De Micheli, "Exact synthesis of multiqubit Clifford+T circuits," in *2019 IEEE/ACM Int. Conf. on Computer-Aided Design (ICCAD)*, 2019.

[6] G. Papakonstantinou, "Systematic Synthesis and Optimization of Reversible Quantum Circuits via MINLP, Toffoli Permutation, and Local Search," *Quantum Rep.* 2026, 8, 79. https://doi.org/10.3390/quantum8030079

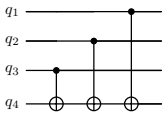
[7] G. Papakonstantinou, "Optimal Design of Reversible Circuits in the Strict NCT Library," *TechRxiv Preprint*, DOI: 10.36227/techrxiv.176827280.05224615/v2, 2026.

APPENDIX A
SYNTHESIZED OPTIMAL CIRCUITS

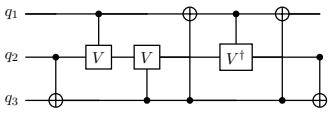
The optimal topological synthesis of the examined benchmark gates is provided in Figure 1. In addition, Figure 2 details the successful sliding-window reduction of the TSG gate architecture.



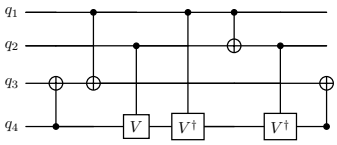
(a) 3-Qubit Toffoli ($m = 5$)



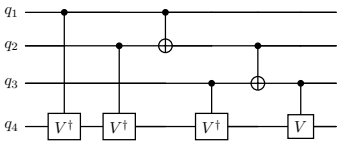
(b) 4-Qubit Parity ($m = 3$)



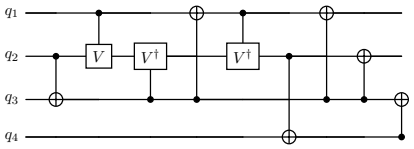
(c) 3-Qubit Fredkin ($m = 7$)



(d) 4-Qubit MIG ($m = 7$)

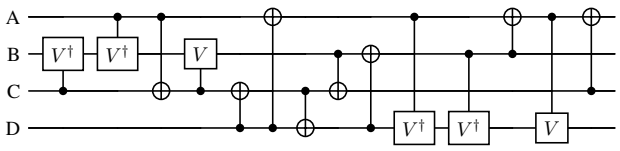


(e) 4-Qubit PFAG ($m = 6$)

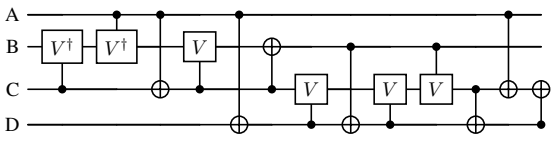


(f) 4-Qubit Sayem ($m = 9$)

Fig. 1. Exact topological synthesis of benchmark quantum circuits. By employing the discrete integer MILP formulation, complex NCV architectures are synthesized completely natively without continuous variable branching. (a) Toffoli gate at depth $m = 5$. (b) Parity cascade at depth $m = 3$. (c) Fredkin gate at depth $m = 7$. (d) Majority Inverter Graph (MIG) at depth $m = 7$. (e) Peres Full Adder Gate (PFAG) at depth $m = 6$. (f) Sayem gate at depth $m = 9$.



(a) Original 14-Gate TSG Circuit [6].



(b) Optimized 13-Gate TSG Circuit. The sub-circuit comprising the final 10 operations of (a) was isolated and mathematically proven to be reducible to a 9-gate architecture natively utilizing the $\mathbb{Z}_4$ MILP solver.

Fig. 2. Sliding window evaluation and exact compilation of the 4-qubit TSG circuit.