

Quantum State Preparation Circuit Optimization Exploiting Don't Cares

Hanyu Wang, Daniel Bochen Tan, Jason Cong
University of California, Los Angeles
{hanyuwang, bctan, cong}@cs.ucla.edu

ABSTRACT

Quantum state preparation initializes the quantum registers and is essential for running quantum algorithms. Designing state preparation circuits that entangle qubits efficiently with fewer two-qubit gates enhances accuracy and alleviates coupling constraints on devices. Existing methods synthesize an initial circuit and leverage compilers to reduce the circuit's gate count while preserving the unitary equivalency. In this study, we identify numerous conditions within the quantum circuit where breaking local unitary equivalences does not alter the overall outcome of the state preparation (i.e., don't cares). We introduce a peephole optimization algorithm that identifies such unitaries for replacement in the original circuit. Exploiting these don't care conditions, our algorithm achieves a 36% reduction in the number of two-qubit gates compared to prior methods.

1 INTRODUCTION

Quantum states preparation (QSP) is indispensable in quantum computing for initializing the state of quantum registers. The initial state to prepare is determined by the specific quantum algorithm and application. Examples of such states include GHZ states, which are instrumental in entanglement experiments [1]; W states [2] and Dicke states [3], which are useful in quantum metrology and act as generators of complex symmetric states [4]; VBS states [5], valuable for modeling interacting [6] quantum spin models. Additionally, certain algorithms encode the problem inputs directly into the initial states [7]. Consequently, developing automated algorithms to design the state preparation circuit is necessary.

Reducing the gate count, particularly the number of two-qubit gates in the circuit, is essential for improving the performance of noisy intermediate-scale quantum computing. To generate the initial QSP circuit, Boolean methods are proposed utilizing decision diagrams to prepare general n -qubit states employing $O(2^n)$ two-qubit gates [8–10]. Besides, specialized algorithms for sparse state preparation are developed to prepare n -qubit states with m nonzero amplitudes using $O(mn)$ two-qubit gates [11–13]. These methods efficiently search for a feasible circuit by decomposing the quantum state using divide-and-conquer techniques.

To reduce the gate count and optimize the initial circuit, synthesis algorithms partition the circuits into blocks with manageable unitary matrix dimensions [14] and apply *unitary synthesis* approaches that search for replacement circuits with fewer gates to implement the given unitary [15]. These unitary synthesis algorithms can effectively simplify the unitary matrices of systems up to six qubits [16–18]. Consequently, these algorithms are often employed as peephole optimizations within larger workflows that utilize iterative methods or design space exploration techniques to enhance scalability [19, 20].

Due to the inherent complexity, existing quantum circuit optimization algorithms assume the unitary is fixed for a given circuit, which is not necessarily true in many quantum applications. In the quantum state preparation problem, for instance, all qubits are initialized to the ground state. As a result, only the first column of the unitary matrix corresponding to the ground state affects the circuit outcome, while the entries in all other columns are flexible. Szasz et al. generalize the flexibilities in applications as a multi-set preparation problem and point out various flexible unitaries in Hamiltonian simulation circuits, preparation of general quantum channels, and circuits with ancillary qubits [21]. Exploiting these flexibilities can significantly improve the circuit performance, as the algorithms are allowed to replace the unitary with a more promising one with a smaller circuit size. However, encoding this flexibility in the optimization algorithm results in an exponentially growing complexity and, thus, does not apply to larger entangled systems [22].

In this paper, we propose a novel scalable algorithm to optimize quantum state preparation circuits. Our method formulates the flexibility in modifying unitary operators as don't cares and allows for simpler circuit configurations. Given an initial QSP circuit, we first partition the circuit into *single-target segments* whose functionality can be efficiently expressed. Then, we propagate and derive the don't-care conditions inside these segments. Finally, our resynthesis algorithm identifies and utilizes don't-cares to simplify the segments' implementation. Experimental results show that our approach enhances the optimization of the QSP circuit and reduces the CNOT count by 36% compared with existing algorithms without employing don't cares.

In the rest of the paper, we present background and related work in Section 2. Then, we give an example of don't care-based circuit optimization in Section 3 to motivate our work. We illustrate our methodology in Section 4. In Section 5, we demonstrate experimental results and evaluate our method.

2 BACKGROUND

In this section, we provide background for quantum states, gates, and circuits. For clarity and space constraints, we refer readers to established sources for formal definitions of notations [23] and quantum gates [24].

2.1 Quantum States and Quantum Gates

We express the n -qubit *quantum state* as a linear combination of 2^n orthonormal basis vectors.

$$|\psi\rangle = \sum_{x \in \{0,1\}^n} c_x |x\rangle, \text{ and } \sum_{x \in \{0,1\}^n} |c_x|^2 = 1,$$

where $|x\rangle \in \{0,1\}^n$ are the *basis states*, $c_x \in \mathbb{C}$ are *amplitudes* that indicate the probability of observing $|x\rangle$ after a measurement.

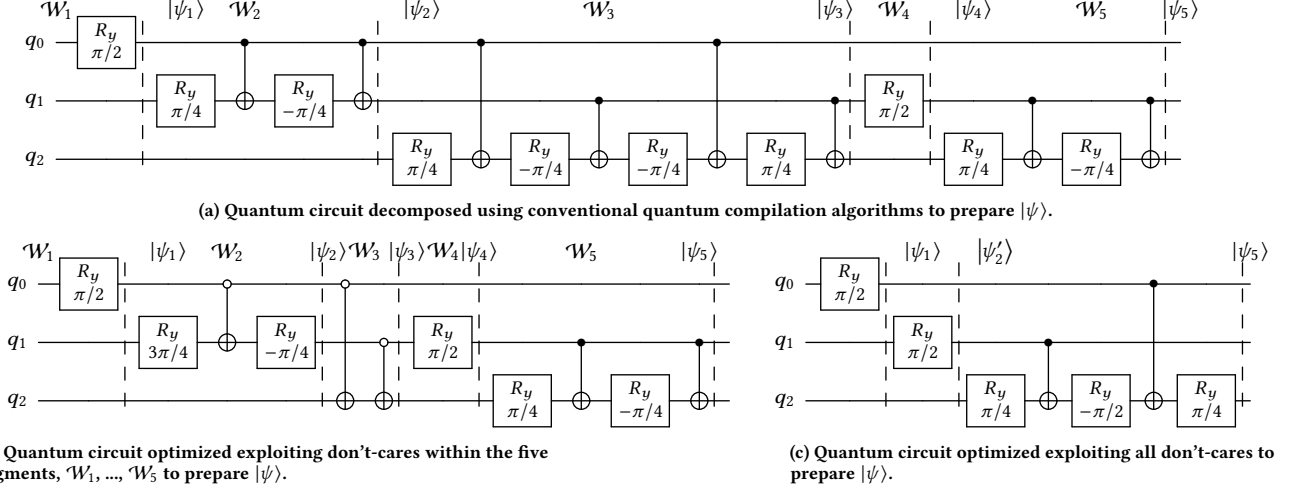


Figure 1: Three quantum circuits with different numbers of gates to prepare the same state $|\psi\rangle$ in Equation (1). This example establishes that the gate counts can be reduced by using don't cares in the quantum circuit.

Quantum gates, or *operators*, denoted by U , are unitary matrices representing transitions between quantum states. $\mathcal{U}(2^n)$ stands for the set of all n -qubit gates. Y rotations, R_y , are single-qubit unitaries $U \in \mathcal{U}(2)$ that redistribute the amplitude between $|0\rangle$ and $|1\rangle$. Z rotations, R_z , introduce the relative phase shift between $|0\rangle$ and $|1\rangle$.

$$R_y(\theta) = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, \quad R_z(\theta) = \begin{pmatrix} \exp(-i\frac{\theta}{2}) & 0 \\ 0 & \exp(i\frac{\theta}{2}) \end{pmatrix},$$

where θ is the *rotation angle*.

Pauli-X, σ_x , is a single-qubit operator that maps $\alpha|0\rangle + \beta|1\rangle$ to $\beta|0\rangle + \alpha|1\rangle$. CNOT is a two-qubit gate that transitions $\alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$ to $\alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle + \delta|10\rangle$.

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \text{CNOT} = \mathbb{1} \oplus \sigma_x = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

All operators in $\mathcal{U}(2^n)$ can be decomposed into gates in the set $\{\text{CNOT}, \mathcal{U}(2)\}$ [24]. A single-qubit gate $\mathcal{U}(2)$ can be further decomposed into R_z and R_y deterministically [25].

Two operators, U_1 and U_2 , *commute* if exchanging their order does not affect the functionality, i.e., $U_1 U_2 = U_2 U_1$.

2.2 Quantum State Preparation

Given a state $|\psi\rangle$, the *quantum state preparation* (QSP) finds a quantum circuit comprising l gates $U_1, U_2, \dots, U_l$ such that these gates transition the ground state $|0\rangle$ to the target state $|\psi\rangle$, i.e., $|\psi\rangle = U_l \dots U_2 U_1 |0\rangle$. For *noisy intermediate-scale quantum* (NISQ) computers, CNOTs produce far more noise than single-qubit gates. Besides, CNOTs require the two involved qubits to be coupled and generate connectivity constraints in the layout synthesis [26]. Therefore, the objective of the QSP circuit optimization in this paper is to minimize the number of CNOTs in the circuit after being decomposed into gates in $\{\text{CNOT}, \mathcal{U}(2)\}$.

Additionally, this paper studies states with real amplitudes due to their predominance in quantum algorithms. To prepare arbitrary

states with complex amplitudes, our method adapts through the controlled phase gates [27].

3 MOTIVATING EXAMPLE

In this section, we illustrate the limitations of applying conventional quantum compilation algorithms to optimize the quantum state preparation circuit. Then, we demonstrate the advantages of our optimization approach, which leverages don't-care conditions and resynthesis subcircuits to achieve a significant reduction in the CNOT count and the error rate.

Consider the target state $|\psi\rangle$. Each index encodes the value of three qubits in the order of $|q_0 q_1 q_2\rangle$.

$$|\psi\rangle = \sqrt{\frac{2}{8}}|000\rangle - \sqrt{\frac{1}{8}}|100\rangle + \sqrt{\frac{1}{8}}|010\rangle + \sqrt{\frac{1}{8}}|101\rangle + \sqrt{\frac{1}{8}}|011\rangle + \sqrt{\frac{2}{8}}|111\rangle \quad (1)$$

Figure 1a depicts a feasible initial QSP circuit of $|\psi\rangle$. This initial circuit comprises five segments, $\mathcal{W}_1, \dots, \mathcal{W}_5$. These segments are generated by decomposing unitary operators, including two single-qubit Y rotations ($\mathcal{W}_1$ and $\mathcal{W}_4$), two single-controlled Y rotations ($\mathcal{W}_2$ and $\mathcal{W}_5$), and one double-controlled Y rotation ($\mathcal{W}_3$). For a rotation with n control qubits, the configuration requires 2^n CNOTs after mapping to $\{\mathcal{U}(2), \text{CNOT}\}$ [25]. Therefore, The final circuit needs 10 single-qubit gates and 8 CNOT gates to prepare $|\psi\rangle$.

Existing optimization algorithms rigidly preserve the unitary equivalency of the quantum circuits [16, 18, 28]. Let U and U' be the unitaries of the initial and optimized circuit. These algorithms assert the optimized circuit must behave the same as the initial circuit for all possible states in the three-qubit system, i.e.,

$$U\varphi = U'\varphi, \quad \forall \varphi \in \mathcal{H}^{\otimes 3}. \quad (2)$$

This constraint is over-conservative for QSP circuit optimization and hinders conventional algorithms from finding more effective circuits with fewer gates. Figure 1b displays a smaller circuit that prepares $|\psi\rangle$ using 5 CNOT gates. Let U_{w_i} and U'_{w_i} be the unitaries of the i th initial and optimized segments $\mathcal{W}_i$, respectively. Although

Table 1: Intermediate quantum states in the example.

$$\begin{aligned}
|\psi_1\rangle &= \sqrt{\frac{1}{2}}|000\rangle + \sqrt{\frac{1}{2}}|100\rangle \\
|\psi_2\rangle &= \sqrt{\frac{2}{4}}|000\rangle + \sqrt{\frac{1}{4}}|100\rangle + \sqrt{\frac{1}{4}}|110\rangle \\
|\psi'_2\rangle &= \sqrt{\frac{1}{4}}|000\rangle + \sqrt{\frac{1}{4}}|010\rangle + \sqrt{\frac{1}{4}}|100\rangle + \sqrt{\frac{1}{4}}|110\rangle \\
|\psi_3\rangle &= \sqrt{\frac{2}{4}}|000\rangle + \sqrt{\frac{1}{4}}|101\rangle + \sqrt{\frac{1}{4}}|110\rangle \\
|\psi_4\rangle &= \sqrt{\frac{2}{8}}|000\rangle - \sqrt{\frac{1}{8}}|100\rangle + \sqrt{\frac{1}{8}}|111\rangle + \sqrt{\frac{1}{8}}|101\rangle + \\
&\quad \sqrt{\frac{1}{8}}|110\rangle + \sqrt{\frac{2}{8}}|010\rangle \\
|\psi_5\rangle &= \sqrt{\frac{2}{8}}|000\rangle - \sqrt{\frac{1}{8}}|100\rangle + \sqrt{\frac{1}{8}}|010\rangle + \sqrt{\frac{1}{8}}|101\rangle + \\
&\quad \sqrt{\frac{1}{8}}|011\rangle + \sqrt{\frac{2}{8}}|111\rangle
\end{aligned}$$

the unitaries of their subcircuits are different, e.g., $U_{w_2} \neq U'_{w_2}$, the outcome states are the same, i.e., $U_{w_2}|\psi_1\rangle = U'_{w_2}|\psi_1\rangle = |\psi_2\rangle$. It can be observed by expressing the matrix representation in terms of the subspace generated by q_0 and q_1 (q_2 is separable in $\mathcal{W}_2$'s state transition).

$$U_{w_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0.7 & -0.7 \\ 0 & 0 & 0.7 & 0.7 \end{pmatrix}, \quad U'_{w_2} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0.7 & -0.7 \\ 0 & 0 & 0.7 & 0.7 \end{pmatrix}, \quad |\psi_1\rangle = \begin{pmatrix} 0.7 \\ 0 \\ 0.7 \\ 0 \end{pmatrix}.$$

The constraint in Equation (2) allows the optimization algorithm to modify the functionality and simplify the circuit more aggressively than Figure 1a, which reduces the gate count.

Nevertheless, the optimization in Figure 1b is again conservative because preserving the equivalency of intermediate states is also unnecessary. An even more aggressive optimization is revealed in Figure 1c. Since a circuit U' is feasible if the outcome of the overall circuit is the target state $|\psi\rangle$, i.e., $U|0\rangle = U'|0\rangle = |\psi\rangle$. Therefore, this algorithm allows altering the intermediate state from $|\psi_2\rangle$ to $|\psi'_2\rangle$. This modification facilitates a significantly simplified circuit configuration to prepare $|\psi\rangle$ employing only 2 CNOT gates.

Table 2 shows the performance on a real quantum computer of the initial circuit in Figure 1a, circuit optimized by Qiskit [28], and circuit in Figure 1c. Results show that employing don't-care conditions allows the optimized circuit to prepare the same state

Table 2: Error rate compared to the ideal value, i.e., $|c_x|^2$, when running on the quantum device ibm-osaka. We display the actual probability and the absolute error (in parenthesis). Optimizing the QSP circuit using don't cares results in better state preparation fidelity¹.

	Ideal	Initial	Opt. w/o DC	Opt. with DC
000⟩	0.250	0.233 (1.7%)	0.226 (2.4%)	0.265 (1.5%)
001⟩	0.125	0.115 (0.9%)	0.116 (0.8%)	0.119 (0.5%)
010⟩	0.125	0.093 (3.2%)	0.223 (9.8%)	0.105 (2.0%)
011⟩	0.000	0.039 (3.8%)	0.022 (2.2%)	0.007 (0.7%)
100⟩	0.000	0.023 (2.3%)	0.013 (1.3%)	0.004 (0.3%)
101⟩	0.125	0.230 (10.%)	0.074 (5.0%)	0.146 (2.0%)
110⟩	0.125	0.061 (6.4%)	0.167 (4.2%)	0.130 (0.4%)
111⟩	0.250	0.206 (4.4%)	0.158 (9.2%)	0.224 (2.5%)
Avg. error		4.18%	4.40%	1.29%

with higher fidelity and reduce the average error compared to the ideal state from 4.18% to 1.29%.

4 METHODOLOGY

As established in Section 3, allowing modifications in the unitary matrices significantly enhances the optimization flexibility of the QSP circuit. However, this introduced flexibility also expands the solution space, potentially increasing the complexity if not managed properly. The main contribution of this paper is the development of a scalable algorithm that exploits these don't care conditions effectively while keeping the complexity under control.

Our quantum circuit optimization workflow is based on the peephole optimization, as presented in Algorithm 1. Starting with an initial QSP circuit $\mathcal{G}$, we sequentially traverse the gates and extract segments that target the same qubit. For each extracted segment, denoted as $\mathcal{W}$, we perform *resynthesis* that searches for a replacement circuit $\mathcal{W}'$ that transitions the quantum states identically while requiring fewer gates. The optimized circuit is then produced by aggregating these improvements across all segments.

In the rest of this section, we will explain the three main components of our algorithm: segment extraction, rotation table representation, don't care condition propagation, and segment resynthesis.

4.1 Single-Target Segment Extraction

Segment extraction is a widely used technique to simplify the complexity of circuit optimization. By focusing the optimization process on a localized segment of the quantum circuit, we can represent the unitary functions more efficiently and significantly reduce the problem size. However, local optimization lacks the comprehensive visibility of feasible solutions, resulting in missed optimization opportunities. Therefore, it is crucial for a partitioning algorithm to strike a balance between efficiency and effectiveness.

Our algorithm extracts single-target segments. A segment is *single-target* if all CNOTs and single-qubit gates in it target the same qubit. Let U_w be the unitary of the segment $\mathcal{W}$, then $\mathcal{W}$ is single-target if and only if U_w can be represented as a multi-controlled single-target operator

$$U_w = \sum_{x \in \{0,1\}^{n-1}} U_{w,x} \otimes |x\rangle\langle x|, \quad (3)$$

where $U_x \in \mathcal{U}(2)$ is unitary matrix that indicates the effect $\mathcal{W}$ casts on the target qubit if the other $n-1$ qubits are in state $|x\rangle$.

To extract single-target segments, we first group adjacent gates with common targets in the original circuit sequence and then extend the boundary of each segment utilizing commute operators. As shown in Algorithm 1 from lines 8 to 14, we traverse all the gates U_j sequentially after i and collect as many gates with the common target ($U_{i,\text{target}}$) to $\mathcal{W}$ as possible. To achieve this, we keep track of the operator U_{ext} , which represents the traversed unitaries with different target qubits and excluded from $\mathcal{W}$. A gate U_j with the same target qubit can be reordered before U_{ext} and inserted into the segment $\mathcal{W}$ if U_j and U_{ext} commute. Therefore, by allowing the reordering of commutable operators in the sequence, we can gather

¹We transpile all three circuits using qiskit with an optimization level of 3 and run 4096 shots for each experiment.

Algorithm 1: QSP Circuit Resynthesis

input : An initial QSP circuit $\mathcal{G}$ comprises a sequence of quantum gates, $U_1, U_2, \dots, U_l$, where $U_i \in \{\mathcal{U}(2), \text{CNOT}\}$.

output : An optimized circuit $\mathcal{G}'$.

```

1   $W \leftarrow \emptyset$ 
2   $|\varphi_t\rangle \leftarrow |0\rangle$ 
3  for  $i = 1, 2, \dots, l$  do
4       $|\varphi_s\rangle \leftarrow |\varphi_t\rangle$ 
5       $U_{ext} \leftarrow \mathbb{1}$ 
6       $\mathcal{W} \leftarrow \emptyset$ 
7      for  $j = i, \dots, l$  do
8          if  $U_j.target = U_i.target$  and  $isCommute(U_j, U_{ext})$  then
9               $\mathcal{W}.push(U_j)$ 
10              $|\varphi_t\rangle = U_j |\varphi_t\rangle$ 
11          else
12              if  $U_j.control = U_i.target$  then
13                  break
14               $U_{ext} \leftarrow U_j U_{ext}$ 
15           $W.push(\mathcal{W})$ 
16  propagateDontCares( $W$ )
17   $\mathcal{G}' \leftarrow \emptyset$ 
18  for  $\mathcal{W} \in W$  do
19       $\mathcal{W}' \leftarrow resynthesis(\mathcal{W}, |\varphi_s\rangle, |\varphi_t\rangle)$ 
20       $\mathcal{G}' \leftarrow \mathcal{G}' \cup \mathcal{W}'$ 
21  return  $\mathcal{G}'$ .
```

more gates into each single-target segment, thereby increasing the potential for optimizing gate count.

Observe that we can terminate the check for commute operators if the target qubit $t = U_i.target$ serves as a control qubit in U_{ext} (line 12). In this case, U_{ext} is not likely to commute with any U_j that targets t , and we exit the loop to accelerate the algorithm.

4.2 Rotation Table Representation

The advantage of single-target segment partitioning is that the local unitaries can be expressed efficiently. Instead of complex matrix multiplications, the functionality can be represented using additions and subtraction of rotation angles. This simplification will noticeably enhance the efficiency of simulating state transitions in the segment and facilitates the implementation of resynthesis algorithms.

THEOREM 4.1. *Given an arbitrary state ψ_s with real amplitudes and a single-target segment $\mathcal{W}$ comprises real unitaries from $\{\mathcal{U}(2), \text{CNOT}\}$, there exists a unitary operator U'_w ,*

$$U'_w = \sum_{x \in \{0,1\}^{n-1}} R_y(\theta_{w,x}) \otimes |x\rangle \langle x|, \quad (4)$$

where $-\pi \leq \theta_{w,x} < \pi$, such that $U'_w |\psi_s\rangle = U_w |\psi_s\rangle$.

PROOF. We give a proof based on induction. Assume this property holds true for all segments with no more than k gates, $k \geq 1$.

Consider a single-target segment $\mathcal{W}$ with $k+1$ gates. We can partition it into two single-target segment $\mathcal{W}_1$ and $\mathcal{W}_2$ with gate counts $k_1, k_2 \leq k$, such that there exists:

$$\begin{aligned} U_w |\psi_s\rangle &= U_{w_1} U_{w_2} |\psi_s\rangle \\ &= (\sum_x R_y(\theta_{w_1,x}) \otimes |x\rangle \langle x|) (\sum_x R_y(\theta_{w_2,x}) \otimes |x\rangle \langle x|) |\psi_s\rangle \\ &= (\sum_x R_y(\theta_{w_1,x} + \theta_{w_2,x}) \otimes |x\rangle \langle x|) |\psi_s\rangle, \end{aligned} \quad (5)$$

which implies that the property also applies to segments with $k+1$ gates.

To complete the proof, it suffices to show that a single $\mathcal{U}(2)$ or CNOT can be replaced by an R_y operator. It holds true because the initial and final state, $|\psi_s\rangle$ and $U_w |\psi_s\rangle$ are given. Let $\alpha_{s,x} |0\rangle + \beta_{s,x} |1\rangle$ and $\alpha_{t,x} |0\rangle + \beta_{t,x} |1\rangle$ be the initial and final states of the target qubit for a control state $|x\rangle$. Then this transition can be accomplished by a Y rotation of θ_x ,

$$\theta_x = 2 \cdot \text{atan2}(\beta_{t,x}, \alpha_{t,x}) - 2 \cdot \text{atan2}(\beta_{s,x}, \alpha_{s,x}), \quad (6)$$

where atan2 is applied instead of $\arctan$ to determine the correct quadrant of the angle. Finally, we can find θ_x to replace $\mathcal{U}(2)$ and CNOTs given the initial state ψ_s . $\square$

We use *rotation tables* to indicate the mapping between the control qubit states, $|x\rangle$, and the corresponding rotation angles, θ_x . This mapping is formalized as a function $\varphi : \{0,1\}^{n-1} \rightarrow [-2\pi, 2\pi)$. Table 3 displays the rotation table of states listed in Table 1 with respect to qubit q_2 . The first column represents the states of the control qubits, and the entries are rotation angles ranging from -2π to 2π . For example, the angle of $|10\rangle$ in $|\psi_5\rangle$ corresponds to two terms, $|100\rangle$ and $|101\rangle$, with coefficients $\alpha = -\sqrt{\frac{1}{8}}$ and $\beta = \sqrt{\frac{1}{8}}$. Therefore, the corresponding rotation angle is $2 \cdot \text{atan2}(\sqrt{\frac{1}{8}}, -\sqrt{\frac{1}{8}}) = \frac{3\pi}{2}$.

Table 3: Rotation tables of q_2 correspond to the state transition from $|\psi'_2\rangle$ to $|\psi_5\rangle$ in Figure 1c. $|\psi'_2\rangle = \sqrt{\frac{1}{4}}|000\rangle + \sqrt{\frac{1}{4}}|010\rangle + \sqrt{\frac{1}{4}}|100\rangle + \sqrt{\frac{1}{4}}|110\rangle$ and $|\psi_5\rangle = \sqrt{\frac{2}{8}}|000\rangle - \sqrt{\frac{1}{8}}|100\rangle + \sqrt{\frac{1}{8}}|010\rangle + \sqrt{\frac{1}{8}}|101\rangle + \sqrt{\frac{1}{8}}|011\rangle + \sqrt{\frac{2}{8}}|111\rangle$.

Control $ x\rangle$	$ \psi'_2\rangle$	$ \psi_5\rangle$	θ_x
$ 00\rangle$	0	0	0
$ 01\rangle$	0	$\pi/2$	$\pi/2$
$ 10\rangle$	0	$3\pi/2$	$3\pi/2$
$ 11\rangle$	0	π	π

4.3 Don't Care Condition Propagation

After partitioning the circuit into segments, line 16 in Algorithm 1 identifies the conditions within each segment where modifying its functionality does not impact the final outcome. The derivation and exploitation of these *don't-care* conditions are well-established in the field of logic synthesis, where they significantly contribute to the simplification of Boolean logic circuits [29–31]. This paper adheres to the established terminologies in logic synthesis, extends the concept of don't-cares, and applies Boolean methods to optimize quantum circuits.

All the don't care conditions utilized in this paper originate from the incompletely specified mappings between basis states. Let $V = \{v_1, \dots, v_{2^n}\}$ and $W = \{w_1, \dots, w_{2^n}\}$ be two sets of orthonormal

basis of $\mathcal{H}^{\otimes n}$. Given m specified mappings, $m \in \{1, \dots, 2^n\}$, the circuit unitary can be expressed as:

$$U = \sum_{i \leq m} |w_i\rangle \langle v_i| + \sum_{i > m} |w_{f(i)}\rangle \langle v_i|, \quad (7)$$

where f randomly permutes the mappings of the rest $2^n - m$ vectors between V and W . Szasz et al. demonstrate that this formulation generalizes the flexibilities in various quantum applications, including Hamiltonian simulation circuit synthesis and quantum channel preparation [21].

We define *external don't cares* of operator U as the set of vectors, $\{v_{m+1}, \dots, v_{2^n}\}$, with unspecified final states after applying U , and we define the corresponding *care set* as the set $C_{ext} = \{v_1, \dots, v_m\}$. Since the QSP circuit usually transitions the quantum states from the ground state $|0\rangle$ to the target state $|\psi\rangle$, the external care set $C_{ext} = \{|0\rangle\}$ contains only the ground state. All other $2^n - 1$ basis vectors $x \in \{0, 1\}^n \setminus C_{ext}$ are considered external don't cares.

To utilize don't cares in our scalable workflow, we need to propagate external don't cares to the single-target segments. According to their properties, we categorized the don't-cares of a segment into *controllability don't cares* (CDC) and *observability don't cares* (ODC).

4.3.1 Controllability Don't Cares (CDC). Controllability don't care is the set of states that will not occur (i.e., have zero amplitude) when entering a segment $\mathcal{W}$. Consequently, the behavior of the target qubits corresponding to non-existent control states does not affect the outcome. Let ψ_s represent the initial state of the segment $\mathcal{W}$, and U_s be the unitary of the circuits prior to $\mathcal{W}$, $|\psi_s\rangle = U_s |0\rangle$. Then, the care set of function $U_{\mathcal{W}}$ is:

$$C_{\mathcal{W}} = \left\{ |x\rangle \otimes \left(\cos\left(\frac{\varphi_{s,x}}{2}\right) |0\rangle + \sin\left(\frac{\varphi_{s,x}}{2}\right) |1\rangle \right) : x \in S(\psi_s) \right\}, \quad (8)$$

where $\varphi_{s,x}$ represents the rotation angle corresponds to control state $|x\rangle$ and $S(\psi_s)$ is the index set of ψ_s that consists of the basis vectors with non-zero amplitudes.

CDC can be efficiently captured by simulating the state transitions from the input ground state on the fly, as displayed in lines 4 and 10 in Algorithm 1. For instance, the segment $\mathcal{W}_2$ in Figure 1a targeting q_1 transition the states from ψ_1 to ψ_2 , whose rotation tables can be derived during simulation. The control states $|0*1\rangle$ and $|1*1\rangle$ do not exist in both $|\psi_1\rangle = \frac{\sqrt{2}}{2} |000\rangle + \frac{\sqrt{2}}{2} |100\rangle$ and $|\psi_2\rangle = \frac{\sqrt{2}}{4} |000\rangle + \frac{\sqrt{1}}{4} |100\rangle + \frac{\sqrt{1}}{4} |110\rangle$, thus, are CDC. The care set, according to Equation (8) is $\{|000\rangle, |100\rangle\}$ and has a cardinality 2 out of all $2^3 = 8$ basis vectors, as shown in Table 4.

Note that the cardinality of the care set $C_{\mathcal{W}}$ is upper-bounded:

$$|C_{\mathcal{W}}| \leq \min(2^{n-1}, |S(\psi_s)|),$$

Table 4: Rotation tables of q_1 corresponds to the state transitions from $|\psi_1\rangle$ to $|\psi_2\rangle$ and $|\psi'_2\rangle$ in Table 1. “X” represents controllability don't cares and “X*” is the observability don't care

Control state	$ \psi_1\rangle$	$ \psi_2\rangle$	$ \psi'_2\rangle$
$ 0*0\rangle$	0	0	X*
$ 0*1\rangle$	X	X	X
$ 1*0\rangle$	0	$\pi/2$	$\pi/2$
$ 1*1\rangle$	X	X	X

where n is the number of qubits and $|S(\psi_s)|$ represents the cardinality of the input state. This is because we need to specify at most one mapping for each control state $|x\rangle$, $x \leq 2^{n-1}$. If the control state $|x\rangle$ does not exist in $S(\psi_s)$, the outcome vector is in the don't care set.

4.3.2 Observability Don't Cares (ODC). Observability don't care is the set of control states whose output state that does not affect the final outcome of the QSP circuit. Let ψ_i be the final state of a segment $\mathcal{W}_i$ targets q_i , U_j be the initial unitary of the next segment $\mathcal{W}_j$ targets q_j , $j \neq i$. Then the rotation angle of a control state $|x\rangle \otimes |*\rangle_i$ is in $\mathcal{W}_i$'s ODC if the projections of U_j on $|x\rangle \otimes |0\rangle_i$ and $|x\rangle \otimes |1\rangle_i$ are the same, i.e.:

$$\langle 0|_i \langle x| U_j |x\rangle |0\rangle_i = \langle 1|_i \langle x| U_j |x\rangle |1\rangle_i. \quad (9)$$

Unlike CDC, which are derived from the predecessors of a segment, observability don't cares are generated and propagated from the opposite direction, from the successors. For example, the rotation angle for the control state $|0*0\rangle$ in Table 4 is an observability don't care of U_{w_2} generated by $\mathcal{W}_3$ from Figure 1a. Note that the eight gates in $\mathcal{W}_3$ compose a controlled Y rotation that is activated only when q_0 is $|1\rangle$ and q_1 is $|0\rangle$. Therefore, when q_0 is $|0\rangle$, U_{w_3} does not modify q_2 regardless of the value of q_1 . In other words, $|000\rangle$ and $|010\rangle$ are equivalent for $\mathcal{W}_2$. Consequently, although the control state $|0*0\rangle$ has non-zero amplitude, its rotation angle does not affect the feasibility of the circuit and, thus, is ODC.

Exploiting ODC can further simplify the circuits in the segment. As illustrated in Table 4, after setting the angle in the first row of $|\psi'_2\rangle$ from 0 to $\frac{\pi}{2}$, the original segment can be replaced by one single R_y gate. After traversing and optimizing the entire circuit, we derive the circuit in Figure 1c from Figure 1a utilizing both CDC and ODC.

It is worth mentioning that although the rotation of the segment $\mathcal{W}_3$ is not affected in the previous example, changing the angle from 0 to $\frac{\pi}{2}$ in Table 4 produces unintended rotations on q_1 . To mitigate the cost of reverting these rotations after $\mathcal{W}_j$, we only exploit observability don't care if the segment $\mathcal{W}_{j+1}$ after $\mathcal{W}_j$ targets the same qubit as $\mathcal{W}_{j-1}$, which introduced unintended rotation. In this case, we can simply adjust the rotation table of $\mathcal{W}_{j+1}$ to completely recover the functionality, which does not necessarily increase the gate count of $\mathcal{W}_{j+1}$.

4.4 Segment Resynthesis

As displayed in line 19 in Algorithm 1, the segment resynthesis is the final step of our method that finds the replacement of the initial segment with fewer gates that accomplishes the same state transition from ψ_s to ψ_t . This step serves as the core of our workflow, significantly decreasing gate count and optimizing the overall circuit.

The workflow of the resynthesis algorithm is illustrated in Algorithm 2. Given a target rotation table, our algorithm:

- (1) defines a hyper-parameter K as the number of CNOTs in the template and initializes it to $K = 0$.
- (2) constructs an equality system to check if the rotation table can be implemented if using K CNOTs.

Algorithm 2: Segment resynthesis

input : The initial and final rotation tables φ_s and φ_t with n control qubits.
output : A sequence of quantum operators to prepare the rotation table with K CNOTs.

```

1  $K \leftarrow 0$ 
2  $\text{solution} \leftarrow \emptyset$ 
3 while  $\text{solution}$  is  $\emptyset$  and  $K \leq K_{\max}$  do
4   for  $\{S_k\}$  in  $\mathbb{Z}_n^{\otimes K}$  do
5      $M \leftarrow \text{Construct equality system } (\varphi_s, \varphi_t, \{S_k\})$ 
6      $\text{solution} \leftarrow \text{Solve}(M)$ 
7    $K \leftarrow K + 1$ 
8 return  $\text{solution}$ .
```

(3) analyzes the result. If a set of solutions is found, we return the circuit with K CNOTs. Otherwise, we increase K by one and repeat step (2).

All utilized variables are listed in Table 5. In the rest of this section, we will demonstrate the construction of the equality system from line 5 in Algorithm 2.

4.4.1 Fixed-target CNOT template. Since all the CNOT gates in the circuit target the same qubit, the control qubit alone suffices to represent each CNOT gate. We define CNOT variables, $S_{k,i}$, as binary values indicating whether q_i serves as the control qubit in k th CNOT. Given that each CNOT has only one control qubit, $\sum_{i \in \mathbb{Z}_n} S_{k,i} = 1$ for all $k \in \mathbb{Z}_K$, where n represents the number of candidate control qubits. Then, we define a set of variables $R_{k,x}$, where $R_{k,x} = -1$ if the rotation at index $|x\rangle$ is enabled by the k th CNOT, and $R_{k,x} = 1$ otherwise. These $R_{k,x}$ variables are derived from the given $S_{k,i}$ variables, as the control qubit i and its phase determines whether an index x is affected by the CNOT gate.

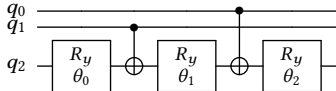


Figure 2: CNOT template with $K = 2$, $S_{1,1} = 1$ and $S_{2,0} = 1$.

4.4.2 Rotation angle variables and constraints. We use $\varphi_{k,x}$ to represent the rotation angle corresponding to index $|x\rangle$ after the k th CNOT gate. The values of the initial and final angles, $\varphi_{0,x} = \varphi_{s,x}$ and $\varphi_{K,x} = \varphi_{t,x}$, are given by the rotation tables. Then, we define

rotation angle constraints as shown in Equation (10).

$$\varphi_{k+1,x} = \frac{\pi}{2} + \theta_k + R_{k,x} \cdot (\varphi_{k,x} - \frac{\pi}{2}). \quad (10)$$

This constraint describes the evolution of the rotation table before and after the k th CNOT gate. If the index $|x\rangle$ is not enabled by the CNOT, then $R_{k,x} = 1$, and $\varphi_{k+1,x} = \varphi_{k,x} + \theta_k$ because the R_y gate after CNOT introduces a rotation of θ_k . Otherwise, if $|x\rangle$ is enabled, $R_{k,x} = -1$, then $\varphi_{k+1,x} = \pi - \varphi_{k,x} + \theta_k$ because the CNOT gate “reflects” the previous angle $\varphi_{k,x}$ along $\varphi = \frac{\pi}{2}$, as illustrated in the proof of Theorem 4.1, before applying the rotation of θ_k .

EXAMPLE 4.2. Consider the segment between $|\psi'_2\rangle$ and $|\psi_5\rangle$, whose initial and final angles are provided in Table 3. Assume we are at the iteration where $K = 2$, then we can assign the boundary values to the rotation angle variables:

$$\begin{aligned} \varphi_{0,|00\rangle} &= \varphi_{0,|01\rangle} = \varphi_{0,|10\rangle} = \varphi_{0,|11\rangle} = \theta_0, \\ \varphi_{2,|00\rangle} &= 0, \varphi_{2,|01\rangle} = \frac{\pi}{2}, \varphi_{2,|10\rangle} = \frac{3\pi}{2}, \varphi_{2,|11\rangle} = \pi, \end{aligned}$$

where θ_0 is the offset introduced by the first R_y gate.

The target is to find an assignment to $\theta_0, \theta_1, \theta_2, S_{1,i}$, and $S_{2,i}$, such that the equality system is satisfied. For clarity, we assume the control qubits of the two CNOTs are $S_{1,1} = 1$ and $S_{2,0} = 1$. This corresponds to the template in Figure 2, where $R_{1,|00\rangle} = R_{1,|01\rangle} = R_{2,|00\rangle} = R_{2,|01\rangle} = 1$ and $R_{1,|10\rangle} = R_{2,|10\rangle} = R_{1,|11\rangle} = R_{2,|11\rangle} = -1$. We can streamline the equality system by plugging in the known variables and canceling intermediate variables corresponding to the same x as shown in Equation (11).

$$\begin{cases} \varphi_{1,|00\rangle} = \varphi_{0,|00\rangle} + \theta_1 \\ \varphi_{1,|01\rangle} = \pi - \varphi_{0,|01\rangle} + \theta_1 \\ \varphi_{1,|10\rangle} = \varphi_{0,|10\rangle} + \theta_1 \\ \varphi_{1,|11\rangle} = \pi - \varphi_{0,|11\rangle} + \theta_1 \\ \varphi_{2,|00\rangle} = \varphi_{1,|00\rangle} + \theta_2 \\ \varphi_{2,|01\rangle} = \varphi_{1,|01\rangle} + \theta_2 \\ \varphi_{2,|10\rangle} = \pi - \varphi_{1,|10\rangle} + \theta_2 \\ \varphi_{2,|11\rangle} = \pi - \varphi_{1,|11\rangle} + \theta_2 \end{cases} \rightarrow \begin{cases} 0 = \theta_0 + \theta_1 + \theta_2 \\ \frac{\pi}{2} = (\pi - \theta_0 + \theta_1) + \theta_2 \\ \frac{3\pi}{2} = \pi - (\theta_0 + \theta_1) + \theta_2 \\ \pi = \theta_0 - \theta_1 + \theta_2 \end{cases} \quad (11)$$

The solution $\theta_0 = \theta_2 = \frac{\pi}{4}, \theta_1 = -\frac{\pi}{2}$ is feasible. Therefore, we can find a circuit using $K = 2$ CNOTs to implement this rotation table as seen in Figure 1c.

We can verify $K = 1$ has no feasible solution by watching the corresponding equality system in Equation (12).

$$\begin{cases} \varphi_{1,|00\rangle} = \varphi_{0,|00\rangle} + \theta_1 \\ \varphi_{1,|01\rangle} = \varphi_{0,|01\rangle} + \theta_1 \\ \varphi_{1,|10\rangle} = \pi - \varphi_{0,|10\rangle} + \theta_1 \\ \varphi_{1,|11\rangle} = \pi - \varphi_{0,|11\rangle} + \theta_1 \end{cases} \rightarrow \begin{cases} 0 = \theta_0 + \theta_1 \\ -\frac{\pi}{2} = \theta_0 + \theta_1 \\ \frac{\pi}{2} = \pi - \theta_0 + \theta_1 \\ \pi = \pi - \theta_0 + \theta_1 \end{cases} \quad (12)$$

Similarly, $K = 0$ is also infeasible, and we return the 2-CNOT circuit as the optimal solution for this segment synthesis problem.

4.5 Complexity Analysis

The overall resynthesis method is based on a peephole algorithm and is therefore scalable and efficient, running in polynomial time.

- (1) Window extraction in Algorithm 1. This step requires $O(N^2)$ quantum operator simulations, where N denotes the number of gates in the initial circuit.

Table 5: Variable declaration for the segment resynthesis problem.

Input:		
$S_{k,i}$	$\{0, 1\}$	Whether q_i is the control qubit in the k th CNOT.
$R_{k,x}$	$\{-1, 1\}$	$R_{k,x} = -1$ if $ x\rangle$ is activated by the k th CNOT.
$\varphi_{s,x}, \varphi_{t,x}$	$[-2\pi, 2\pi)$	Initial and final angles corresponds to $ x\rangle$.
Internal:		
$\varphi_{k,x}$	$[-2\pi, 2\pi)$	The angle corresponds to $ x\rangle$ after the k th CNOT.
Output:		
θ_k	$[0, 4\pi)$	The rotation angle of R_y after the k th CNOT.

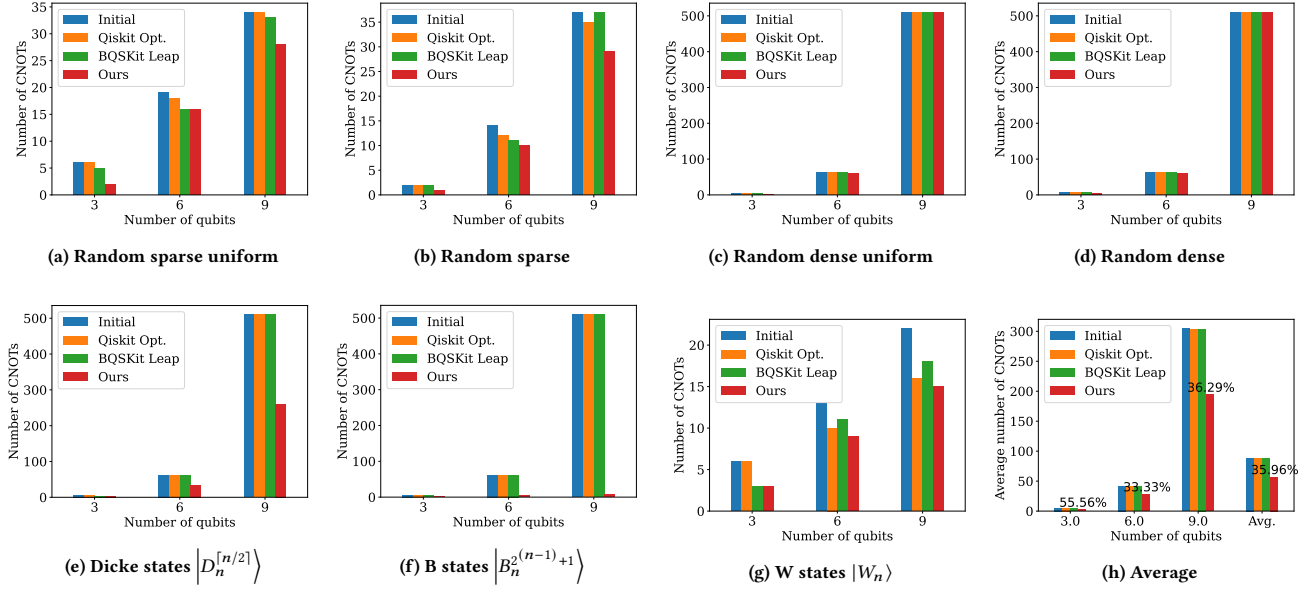


Figure 3: CNOT count comparison between three optimization algorithms.

- (2) Resynthesis algorithm in Algorithm 2. For each segment, the time complexity is $O(n^{K_{\max}} m^2 K_{\max})$, where n is the number of qubits and m is the cardinality of the care set in the extracted segment. This is because we need to enumerate $n^{K_{\max}}$ CNOT templates, and each template has a linear system with $K_{\max} + 1$ columns and m rows. We fix $K_{\max} = 2$ to upper-bound the complexity to $O(n^2 m^2)$.

It is worth mentioning that we accelerate the quantum operator simulation to speed up our algorithm further. Although each operator corresponds to a unitary matrix with a dimension of 2^n , the complexity, in practice, is lower because the implementation leverages sparsity and skips the multiplications of indices with zero amplitudes. Furthermore, all operators in the segment target the same qubits and do not affect the values of control qubits in the basis vectors.

5 EXPERIMENTAL RESULTS

In this section, we present results to evaluate the effectiveness and efficiency of the proposed algorithm.

5.1 Benchmarks and Workflow

The benchmark suite used in our experiments consists of various states with different numbers of qubits, different levels of symmetry, and different sparsity. For each category of quantum states, we apply the most advantageous algorithm to generate the initial QSP circuit [9, 11, 32].

- (1) $|B_n^k\rangle$ are quantum states with uniform amplitudes on the first $k + 1$ basis states. $|B_n^k\rangle = \frac{1}{\sqrt{k+1}} \sum_{x=0}^k |x\rangle$.
- (2) Dicke state $|D_n^k\rangle$ are highly symmetric n -qubit states. Indices of Dicke states have exactly k ones and $n - k$ zeros [3].
- (3) W state $|W_n\rangle$ is a special case of Dicke states when $k = 1$ [2].

- (4) Sparse random states have cardinality $m = n$, and dense random states have $m = 2^{n-1}$.

We implement the proposed method using Python and apply one iteration of the resynthesis algorithm introduced in Section 4 to these initial QSP circuits. We compare the gate count reduction of our method with two circuit optimization methods as baselines, including the unitary synthesis pass in Qiskit (with an optimization level of 3) [28] and a powerful circuit optimization tool based on numerical methods [16].

We map the optimized circuit from ours and baseline methods to $\{\mathcal{U}(2), \text{CNOT}\}$ and verify the correctness of the QSP circuits by running `qiskit` simulation [28]. All experiments are conducted on a computer equipped with a 3.7GHz AMD Ryzen 9 5900X processor and 64GB RAM.

Although our algorithm only applies to real unitary synthesis while baseline methods handle more general complex unitaries, we could adapt through phase oracle to prepare states with complex amplitudes. Besides, many interesting states, such as the benchmarks above, are real. Therefore, their state preparation circuits can be improved directly using our method.

5.2 CNOT Count Comparison

Figure 3 displays the CNOT count comparison on seven categories of quantum states after running the three optimization algorithms. The average CNOT count shown in Figure 3h demonstrates the advantages of our method in quantum circuit optimization. Leveraging the don't cares in the QSP circuit, we further lower the CNOT count of the initial circuit by 36%. Meanwhile, the two baseline algorithms do not significantly optimize the circuit because of the constraints to completely preserve the circuit unitary.

In general, our algorithm is more effective on sparse states than on dense ones. This is because the number of don't cares decreases as the cardinality of the index set m increases in our formulation.

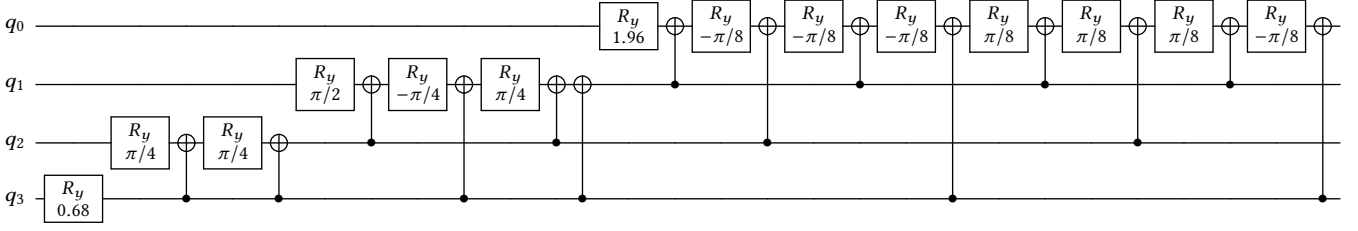


Figure 4: 4-qubit B state preparation using 14 single-qubit gates and 14 CNOTs.

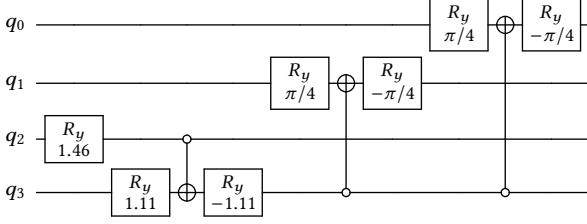


Figure 5: 4-qubit B state preparation utilizing 7 single-qubit gates and 3 CNOTs.

Note that our method simplifies the circuit by identifying an alternative unitary allowed by the don't-care conditions. Consequently, the greater the circuit's flexibility, the higher the likelihood of finding a feasible replacement that minimizes the gate count.

A particularly interesting result is the optimization of $|B_n^{2^{(n-1)}+1}\rangle$ state preparation circuits. Figure 3f exhibits that our resynthesis algorithm exponentially reduces the gate count required to prepare these states. To illustrate the optimization, we display the initial QSP circuit for $|B_4^9\rangle$ in Figure 4 and our optimized circuit in Figure 5.

Notice that although both circuits subsequently entangle qubits into the system, the initial circuit in Figure 4 needs 2, 4, and 8 CNOTs

to entangle q_2, q_3 and q_4 , respectively, while Figure 5 uses 1 CNOT for each qubit. Since the segments to entangle these qubits are all single-target, the result confirmed that our algorithm successfully extracts the rotation tables of these segments and identifies the most efficient CNOT configuration to accomplish these rotations.

5.3 Scalability Analysis

The principal contribution of this paper is the introduction of a scalable resynthesis algorithm. This algorithm partitions the quantum circuit into single-target segments and uses the propagation of don't care conditions to derive a rotation table. By converting the complex problem of unitary synthesis, which typically involves matrix multiplications with high-order terms, into simpler linear equalities, our approach not only simplifies the process but also utilizes don't care conditions more efficiently, enhancing practical utility.

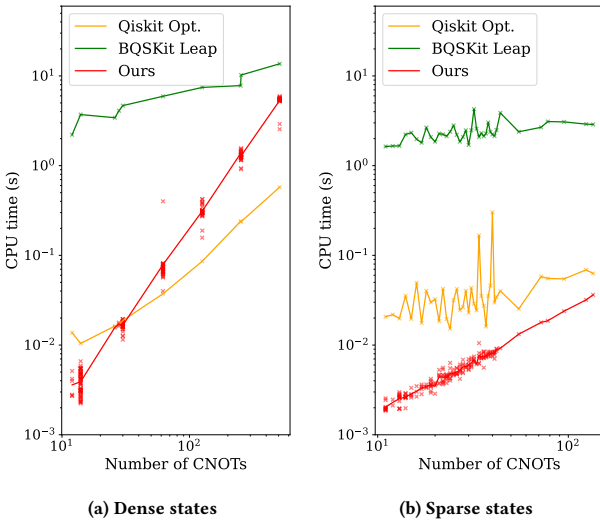
Figure 6 depicts the relationship between the CPU time required by three optimization algorithms and the number of CNOTs in the initial circuit. The runtime exhibits linear growth on a log-log plot, suggesting that our algorithm operates with polynomial complexity when limiting the value of $K_{\max}$, which is 2 in this paper. Furthermore, our resynthesis algorithm completes circuits encompassing up to 10 qubits and several hundred CNOT gates within a few seconds, thereby proving both the efficiency and practical utility of our method.

6 CONCLUSION

In the noisy intermediate-scale quantum computing era, minimizing the count of two-qubit gates within quantum circuits is crucial. Due to the inherent complexity, existing algorithms typically assume a fixed unitary to optimize the circuit. However, modifying the unitary operator does not necessarily affect the functional correctness of the circuit in several quantum applications, and leveraging the flexibility can lead to significant circuit simplification. This paper introduces a scalable workflow that employs don't-care conditions to encode such flexibilities within the unitary matrix and develops an efficient resynthesis algorithm to optimize gate counts. Our experimental results demonstrate a significant reduction in CNOT gate usage. By permitting adjustments to the unitary during the optimization process, our method achieves an average decrease of 36% in CNOT count when applied to the optimization of state preparation circuits.

ACKNOWLEDGEMENTS

This work is funded by NSF grant 2313083. The authors would like to thank Marci Baun for proofreading the paper.


 Figure 6: Scalability analysis. We demonstrate the relationship between CPU time on the y -axis and the number of CNOTs in the initial circuit on the x -axis.

REFERENCES

- [1] Daniel M Greenberger et al. Going beyond Bell's theorem. In *Bell's Theorem, Quantum Theory and Conceptions of the Universe*, pages 69–72. Springer, 1989.
- [2] Wolfgang Dür et al. Three qubits can be entangled in two inequivalent ways. *Physical Review A*, 62(6):062314, 2000.
- [3] Robert H Dicke. Coherence in spontaneous radiation processes. *Physical Review*, 93(1):99, 1954.
- [4] Géza Tóth. Multipartite entanglement and high-precision metrology. *Physical Review A*, 85(2):022322, 2012.
- [5] Bruno Murta, Pedro MQ Cruz, and Joaquín Fernández-Rossier. Preparing valence-bond-solid states on noisy intermediate-scale quantum computers. *Physical Review Research*, 5(1):013190, 2023.
- [6] N Read and Subir Sachdev. Valence-bond and spin-Peierls ground states of low-dimensional quantum antiferromagnets. *Physical review letters*, 62(14):1694, 1989.
- [7] Sahel Ashhab. Quantum state preparation protocol for encoding classical data into the amplitudes of a quantum information processing register's wave function. *Physical Review Research*, 4(1):013091, 2022.
- [8] Israel F Araujo et al. A divide-and-conquer algorithm for quantum state preparation. *Scientific Reports*, 11(1):6329, 2021.
- [9] Fereshte Mozafari, Mathias Soeken, and Giovanni De Micheli. Preparation of uniform quantum states utilizing Boolean functions. In *28th International Workshop on Logic Synthesis (IWLS)*, 2019.
- [10] Philipp Niemann, Rhitam Datta, and Robert Wille. Logic synthesis for quantum state generation. In *2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)*, pages 247–252. IEEE, 2016.
- [11] Niels Gleinig and Torsten Hoeffler. An efficient algorithm for sparse quantum state preparation. In *2021 58th ACM/IEEE Design Automation Conference (DAC)*, pages 433–438. IEEE, 2021.
- [12] Fereshte Mozafari et al. Efficient deterministic preparation of quantum states using decision diagrams. *Physical Review A*, 106(2):022617, 2022.
- [13] Emanuel Malvetti et al. Quantum circuits for sparse isometries. *Quantum*, 5:412, 2021.
- [14] Xin-Chuan Wu, Marc Grau Davis, Frederic T Chong, and Costin Iancu. QGo: Scalable quantum circuit optimization using automated synthesis. *arXiv preprint arXiv:2012.09835*, 2020.
- [15] Marc Grau Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. Heuristics for quantum compiling with a continuous gate set. *arXiv preprint arXiv:1912.02727*, 2019.
- [16] Ethan Smith, Marc Grau Davis, Jeffrey Larson, Ed Younis, Lindsay Bassman Otfelie, Wim Lavrijsen, and Costin Iancu. LEAP: Scaling numerical optimization based synthesis using an incremental approach. *ACM Transactions on Quantum Computing*, 4(1):1–23, 2023.
- [17] Marc G Davis, Ethan Smith, Ana Tudor, Koushik Sen, Irfan Siddiqi, and Costin Iancu. Towards optimal topology aware quantum circuit synthesis. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 223–234. IEEE, 2020.
- [18] Ed Younis, Koushik Sen, Katherine Yelick, and Costin Iancu. Qfast: Conflating search and numerical optimization for scalable quantum circuit synthesis. In *2021 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 232–243, 2021.
- [19] Aleks Kissinger and John van de Wetering. PyZX: Large scale automated diagrammatic reasoning. *arXiv preprint arXiv:1904.04735*, 2019.
- [20] Korbinian Staudacher. *Optimization Approaches for Quantum Circuits using ZX-calculus*. PhD thesis, Master's thesis, Ludwig-Maximilians-Universität, München., 2021.
- [21] Aaron Szasz, Ed Younis, and Wibe De Jong. Numerical circuit synthesis and compilation for multi-state preparation. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, pages 768–778. IEEE, 2023.
- [22] Sahel Ashhab, Fumiki Yoshihara, Miwako Tsuji, Mitsuhiro Sato, and Kouichi Semba. Quantum circuit synthesis via a random combinatorial search. *Phys. Rev. A*, 109:052605, May 2024.
- [23] Michael A Nielsen and Isaac L Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2010.
- [24] Adriano Barenco et al. Elementary gates for quantum computation. *Physical Review A*, 52(5):3457, 1995.
- [25] Anna M Krol, Aritra Sarkar, Imran Ashraf, Zaid Al-Ars, and Koen Bertels. Efficient decomposition of unitary matrices in quantum circuit compilers. *Applied Sciences*, 12(2):759, 2022.
- [26] Bochen Tan and Jason Cong. Optimal layout synthesis for quantum computing. In *Proceedings of the 39th IEEE/ACM International Conference on Computer-Aided Design, ICCAD '20*, Virtual Event, USA, July 2020. Association for Computing Machinery.
- [27] Matthew Amy et al. On the controlled-not complexity of controlled-not-phase circuits. *Quantum Science and Technology*, 4(1):015002, September.
- [28] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.
- [29] Hamid Savoj and Robert K Brayton. The use of observability and external don't cares for the simplification of multi-level networks. In *Proceedings of the 27th ACM/IEEE Design Automation Conference*, pages 297–301, 1991.
- [30] Karen A Bartlett, Robert K Brayton, Gary D Hachtel, Reily M Jacoby, Christopher R Morrison, Richard L Rudell, Alberto Sangiovanni-Vincentelli, and A Wang. Multi-level logic minimization using implicit don't cares. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 7(6):723–740, 1988.
- [31] Maurizio Damiani and Giovanni De Micheli. Observability don't care sets and boolean relations. In *ICCAD*, volume 90, pages 502–505, 1990.
- [32] Hanyu Wang, Jason Cong, and Giovanni De Micheli. Quantum state preparation using an exact cnot synthesis formulation. In *Design, Automation and Test in Europe Conference (DATE)*, pages 1–6, 2024.