Quantum circuit synthesis with SQiSW

Jialiang Tang[1,2], Jialin Zhang[1,2], and Xiaoming Sun[1,2]

[1]State Key Lab of Processors, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China

The primary objective of quantum circuit synthesis is to efficiently and accurately realize specific quantum algorithms or operations utilizing a predefined set of quantum gates, while also optimizing the circuit size. It holds a pivotal position in Noisy Intermediate-Scale Quantum (NISQ) computation. Historically, most synthesis efforts have predominantly utilized CNOT or CZ gates as the 2-qubit gates. However, the SQiSW gate, also known as the square root of iSWAP gate, has garnered considerable attention due to its outstanding experimental performance with low error rates and high efficiency in 2-qubit gate synthesis. In this paper, we investigate the potential of the SQiSW gate in various synthesis problems by utilizing only the SQiSW gate along with arbitrary single-qubit gates, while optimizing the overall circuit size. For exact synthesis, the upper bound of SQiSW gates to synthesize arbitrary 3-qubit and n-qubit gates are 24 and $\frac{139}{192}4^n(1+o(1))$ respectively, which relies on the properties of SQiSW gate in Lie theory and Quantum Shannon Decomposition. We also introduce an exact synthesis scheme for Toffoli gate using only 8 SQiSW gates, which is grounded in numerical observation. More generally, with respect to numerical approximations, we provide a theoretical analysis of a pruning algorithm to reduce the size of the searching space in numerical experiment to $\frac{1}{12}+o(1)$ of previous size, helping us reach the result that 11 SQiSW gates are enough in arbitrary 3-qubit gates synthesis up to an acceptable numerical error.

Jialiang Tang: tangjialiang20@mails.ucas.ac.cn
Jialin Zhang: zhangjialin@ict.ac.cn
Xiaoming Sun: sunxiaoming@ict.ac.cn

1 Introduction

Quantum circuit synthesis is crucial to the implement of quantum algorithm on physical devices. It involves constructing a quantum circuit to realize the target unitary operator, while optimizing the circuit size or depth with respect to a given gate set [1–3].

A lot of research has been done on quantum circuit synthesis, most of which are about exact synthesis and focus on CNOT (Controlled-NOT) gate at the early stage [1, 2, 4–12]. The first synthesis algorithm using CNOT and arbitrary single-qubit gates was designed by Barenco et al. in 1995 [4], giving an upper bound of $O(n^3 4^n)$ to synthesize arbitrary n-qubit gates. The upper bound was improved to $O(n4^n)$ in the same year [10]. During the next decade, circuit transform techniques and Gray code were adopted in circuit synthesis to reduce the upper bound [5, 13, 14]. Actually, the lower bound of arbitrary n-qubit gate was proved to be $\lfloor \frac{1}{4}(4^n-3n-1) \rfloor$ by parameter counting [15]. And the state-of-the-art result of upper bound is $\frac{23}{48}4^n - \frac{3}{2}2^n + \frac{4}{3}$ by making use of a brand new method called Quantum Shannon Decomposition [1]. This upper bound is already no more than 2 times of the lower bound. On Toffoli gate, the CNOT cost is 6, which is a tight result [16, 17]. There are also several works to optimize CNOT-cost of multi-qubit Toffoli gate [18]. Recently, Chen et al. [19] looked into the universal synthesis using arbitrary 2-qubit gates. They proposed a "rule-them-all" exact synthesis scheme AshN, pointing out that the upper bound of circuit size of 2-qubit gates to synthesize n-qubit gate is $\frac{23}{64}4^n(1+o(1))$.

There are also also several studies on how to *numerically approximate* the target unitary matrix, that is, how to optimize the circuit size required to approximate the target circuits within an acceptable numerical error [20–25]. A study in 2019 [20] explored how to get a optimal syn-

thesis of circuits numerically, which uses Mølmer Sørensen (MS) gate. Since MS gate entangles all the qubits in the circuit, the whole structure of circuit is fixed and determining the continuous parameters of different gates is the only thing. In 2022, Ashhab et al. [24] designed a numerical optimization algorithm using CNOT gate and arbitrary single-qubit gates, achieving good performance in 2-qubit quantum state preparation, 3-qubit quantum state preparation and synthesis of n-qubit Toffoli gate. The number of CNOT gates to reach high fidelity is even close to the theoretical lower bound.

In addition, some works have proposed more general circuit synthesis and optimization frameworks that can support arbitrary set of gates. Anouk et al. raised Synthetiq [26], a fast and versatile quantum circuit synthesis framework based on Simulated Annealing. Addressing the limitations of existing tools, Synthetiq synthesizes circuits over arbitrary finite gate sets and supports partial specifications. Also, researchers from Berkeley developed a universal circuit compilation framework called BQSKit [27, 28], which is a comprehensive and powerful circuit compilation tool. It is dedicated to providing efficient and scalable quantum circuit optimization and synthesis solutions. Its core goal is to reduce the depth of quantum circuits through innovative algorithms, improve operational reliability and computational efficiency, and is suitable for fields such as subroutine compilation, gate group conversion, and algorithm exploration.

However, other quantum gates besides CNOT gate are rarely studied in quantum circuit synthesis. A problem has then emerged that whether there exists a kind of quantum gate that has a more powerful synthesis ability and also shows off lower experimental error at the same time.

The SQiSW gate, known as the square root of iSWAP gate, is a 2-qubit quantum gate experimentally realized in earlier works [29–31]. Recently it has been proved to potentially realize the dream [32]. On one hand, SQiSW gate has a shorter gate time and lower error than CNOT gate on superconducting quantum processor of Alibaba. This new type of quantum gate set has excellent performance in experiments [32]: The gate fidelity measured on single SQiSW gate is up to 99.72%, with an average of 99.31%. For any 2-qubit gate synthesis problem, it can achieve an average fidelity of 96.38% on the random samples. Compared to using iSWAP gates on the same processor, the former reduces the average error by 41%, while the latter reduces it by 50%. On the other hand, SQiSW gate has an improved ability in synthesis of arbitrary 2-qubit gates than CNOT gate. Specifically, the upper bound of SQiSW gates and CNOT gates to synthesize arbitrary 2-qubit gates are both 3, and about 79% 2-qubit gates can be synthesized by at most 2 SQiSW gates while the gates generated by 2 CNOT gates constitute a zero-measure set (see supplemental material of [32]). These results rely on KAK decomposition and Weyl chamber, which are Lie algebra mathematical tools to depict general properties of $SU(4)$.

Due to the superior properties of SQiSW gate over common CNOT gate, it's significant to look deeper into the potential of SQiSW gate in circuit synthesis, which hopefully help implement more efficient quantum circuits on superconducting quantum computers.

In this paper, we focus on the exact synthesis and numerical optimization with SQiSW circuits. We utilize only the SQiSW gate along with arbitrary single-qubit gates, aiming at the optimization of number of SQiSW gates. Our main results are presented as follows.

Theorem 1. *An arbitrary 3-qubit gate can be synthesized using a maximum of 24 SQiSW gates.*

Theorem 2. *An arbitrary n-qubit gate can be synthesized using a maximum of $\frac{139}{192} \times 4^n - 3 \times 2^n + \frac{5}{3}$ SQiSW gates.*

Our approach to deriving Theorem 1 is that we firstly decompose arbitrary 3-qubit gates into arbitrary 2-qubit gates, and subsequently utilizing the properties of SQiSW gates in synthesis of arbitrary 2-qubit gates. To prove Theorem 2, we employ Quantum Shannon Decomposition recursively in which the base case is Theorem 1, with several circuit optimization techniques.

Algorithm 1 and algorithm 2 are numerical optimization algorithms for Toffoli gate and arbitrary 3-qubit gates respectively. Algorithm 3 is a pruning algorithm using two pruning techniques, which are qubit re-arrangement and circuit reversion [24, 26], to speed up our numerical optimization algorithms.

The numerical results indicate that we only need 8 and 11 SQiSW gates in circuit to synthesize Toffoli gate and arbitrary 3-qubit gates respectively under acceptable numerical error, which is intuitively shown in Fig. 5(a) and Fig. 5(b). Theorem 3 is about the efficiency analysis of algorithm 1 and algorithm 2 with application of algorithm 3 on them.

Theorem 3. *The pruning algorithm 3 reduces the size of circuit structure space (with 3 qubits and N quantum gates) in algorithm 1 and algorithm 2 to $\frac{1}{12}\times 3^N + 3^{\lfloor \frac{N-1}{2} \rfloor} + \frac{1}{4} = \frac{1}{12}3^N(1+o(1))$.*

Theorem 4. *Toffoli gate can be exactly synthesized by 8 SQiSW gates, a scheme shown in Fig. 6.*

Theorem 4 is proved in a numerical-aided way. It is the numerical optimization on Toffoli gate that provides the observation of pattern of parameters in circuit. By repeatedly guessing, fixing and retraining, we get to have a circuit with only few parameters, which approximates Toffoli gate well. Then we prove it can be an exact synthesis scheme by assigning appropriate values to the parameters.

The rest of this paper is organized as follows. Section 2 is about preliminaries. In section 3, we prove Theorem 1 and recursively induce Theorem 2, our road map contains some decomposition schemes and circuit optimization techniques. In section 4, we show the numerical optimization algorithm and pruning algorithm. Then we prove Theorem 3 and show the numerical results. In section 5, we prove Theorem 4, which is derived by numerical observation. Section 6 makes a summary.

2 Preliminaries

Given a matrix U, we denote its (i,j)-th entry as U_{ij}. We also denote the conjugate transpose of U as $U^\dagger$. And we use $\otimes$ to denote the Kronecker tensor product over two matrices.

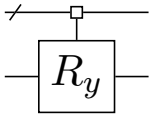
We denote the unitary group $U_n(\mathbb{C})$ as $U(n)$ and specialized unitary group $SU_n(\mathbb{C})$ as $SU(n)$. Here $\mathbb{C}$ is the field of complex numbers.

2.1 Quantum multiplexor

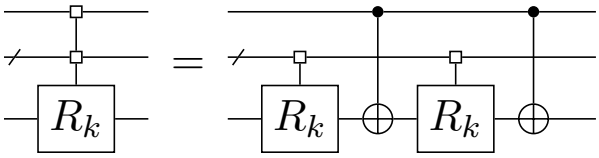
Definition 1 (Quantum multiplexor [1])**.** *A quantum multiplexor with s controlled qubits (positioned at the highest levels) and d target qubits is a block-diagonal unitary matrix comprising 2^s unitary matrices $U_i \in U(2^d)$, $1 \le i \le 2^s$:*

$$U = \begin{pmatrix} U_1 & & \\ & \ddots & \\ & & U_{2^s} \end{pmatrix}.$$

Here is a typical example: the multiplexor-R_y, which is a multiplexor with 1 data qubit acted by R_y gate with different angles depending to the state of the controlled qubits. In the circuit diagram, we mark each controlled qubit of U in quantum circuits with a "□" symbol.



Lemma 1 (Decomposition of multiplexor rotation gates [1, 2, 33])**.** *For multiplexor-R_k and $k = y, z$, it holds that*



and any n-qubit multiplexor-R_k can be synthesized by at most 2^{n-1} CNOT gates.

2.2 Local equivalence and Weyl chamber

Since single-qubit gates contribute little to the cost of a quantum circuit, it is essential to study the equivalence between quantum gates that only differ in single-qubit gates on sides. Fortunately, KAK decomposition and Weyl chamber provide mathematical tools to depict 2-qubit gates up to single-qubit gates [32, 34–38].

Lemma 2 (KAK decomposition [34])**.** *$\forall U \in SU(4)$, there exists a unique $\vec{k} = (x, y, z)$, $\frac{\pi}{4} \ge x \ge y \ge |z|$, single-qubit gates $A_0, A_1, B_0, B_1 \in SU(2)$ and $g \in \{1, i\}$ s.t.*

$$U = g \cdot (A_0 \otimes A_1) exp\{i\vec{k} \cdot \vec{\sigma}\}(B_0 \otimes B_1),$$

in which $\vec{\sigma} \equiv [X \otimes X, Y \otimes Y, Z \otimes Z]$.

The vector $\vec{k}$ is called *interaction coefficients*, which characterize the equivalent class of an arbitrary $U \in SU(4)$, denoted as $k(U)$. The 3-dimension area related to $\vec{k}$ is known as *Weyl*

chamber [32, 34, 37]. The Weyl chamber W is defined as $W \equiv \{\frac{\pi}{4} \geq x \geq y \geq z \quad \text{and} \quad z \geq 0 \quad \text{if} \quad x = \frac{\pi}{4} \mid (x, y, z) \in \mathbb{R}^3\}$,

shown in Fig. 1 with some common gates and their interaction coefficients marked.

Definition 2 (Local equivalence [32])**.** *Two unitary matrices U and V are said to be locally equivalent, if and only if U and V share the same interaction coefficients in Weyl chamber.*

The following properties are of utmost importance when addressing interaction coefficients.

Property 1 (Transformation of interaction coefficients [34])**.** *The local equivalent class of a unitary matrix U remains the same under such transformation on its interaction coefficients:*

- *Add or minus one element of (x, y, z) by $\pi/2$. For example, add x by $\pi/2$ while keep y and z:*
$$(x, y, z) \mapsto (x + \pi/2, y, z).$$
- *Take the opposite number of two elements of (x, y, z). For example, take the opposite number of x and y, while keep z:*
$$(x, y, z) \mapsto (-x, -y, z).$$
- *Exchange two elements of (x, y, z). For example, exchange x and y, while keep z:*
$$(x, y, z) \mapsto (y, x, z).$$

In fact, if we take the equivalence under transformation as $\approx$, then [34, 38]

$$W = \mathbb{R}^3 / \approx .$$

So we can easily round the interaction coefficients of any arbitrary $U \in SU(4)$ to one and only one position in Weyl chamber.

2.3 SQiSW gate and its properties

SQiSW gate is defined as the matrix square-root of iSWAP gate, that is,

$$\begin{aligned} \text{SQiSW} &\equiv \sqrt{\text{iSWAP}} \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{1}{\sqrt{2}} & \frac{i}{\sqrt{2}} & 0 \\ 0 & \frac{i}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

Besides, SQiSW has such mathematical properties which are useful in the rest part of this paper. Here we skip the proof since it's easy to verify them.

Property 2 (Mathematical Properties of SQiSW)**.** *SQiSW gate has such properties:*

- *SQiSW gate is commutative with $Z \otimes Z$, thus commutative with $R_z(\theta) \otimes R_z(\theta)$.*
- *SQiSW gate is qubit symmetrical. It satisfies that $SQiSW = SWAP \cdot SQiSW \cdot SWAP$.*
- *$SQiSW^\dagger$ is locally equivalent to SQiSW.*

It's been proved that any arbitrary $U \in SU(4)$ can be synthesized by at most 3 SQiSW gates [32], and the region that can be spanned by 2 SQiSW gates is stained red in Fig. 2. The rest part needs at least 3 SQiSW gates. We rewrite it as the Lemma 3 below.

Lemma 3 (Synthesis of 2-qubit gates using SQiSW [32])**.** *Any arbitrary 2-qubit gate $U \in SU(4)$ can be exactly synthesized by at most 3 SQiSW gates, up to single qubit gates. U can be synthesized by 1 SQiSW gate, if and only if $k(U)$ is $(\frac{\pi}{8}, \frac{\pi}{8}, 0)$. U can be synthesized by at most 2 SQiSW gates, if and only if $k(U)$ is in W', where $W' \equiv \{\frac{\pi}{4} \geq x \geq y \geq |z|$ and $x \geq y + |z| \mid (x, y, z) \in W\}$.*

This indicates that 3 SQiSW gates can span the whole Weyl chamber, and 2 SQiSW gates span the red area of Weyl chamber in Fig. 2.

3 Exact synthesis of arbitrary quantum gates

In this section, we prove that the upper bound of arbitrary 3-qubit synthesis is 24, then recursively derive the upper bound of arbitrary n-qubit synthesis, which is $\frac{139}{192} 4^n (1 + o(1))$.

3.1 Exact synthesis of arbitrary 3-qubit gates

We first introduce two lemmas, namely Lemma 4 and Lemma 5, as the foundation for the proof of Theorem 1.

Lemma 4 (Decomposition of arbitrary 3-qubit gates [19])**.** *Any 3-qubit gate can be synthesized by at most 11 2-qubit gates, a scheme shown as below.*

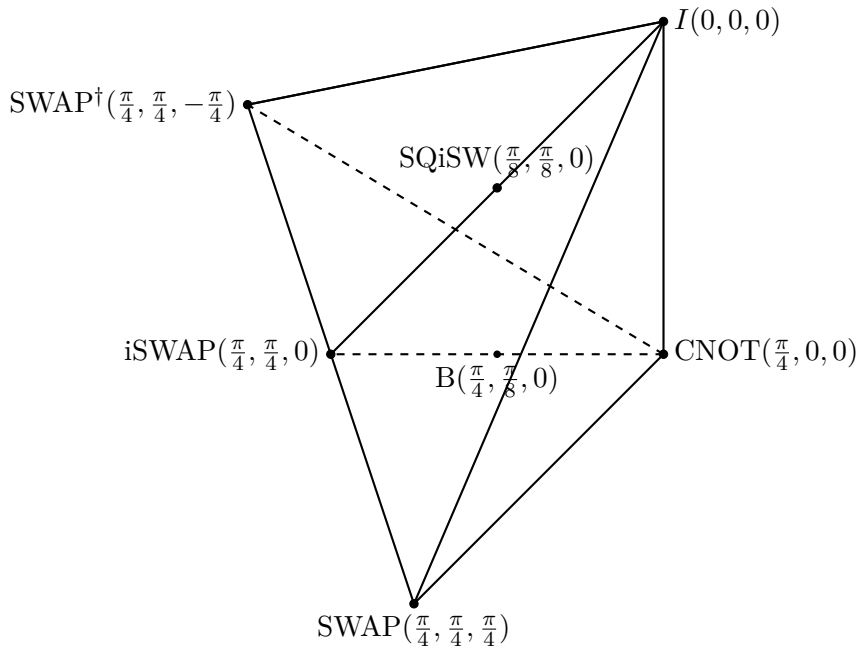


Figure 1: Weyl chamber in $\mathbb{R}^3$, with some common gates and their interaction coefficients. Note that there are many ways to draw Weyl chamber, and here we adopt that in [32].

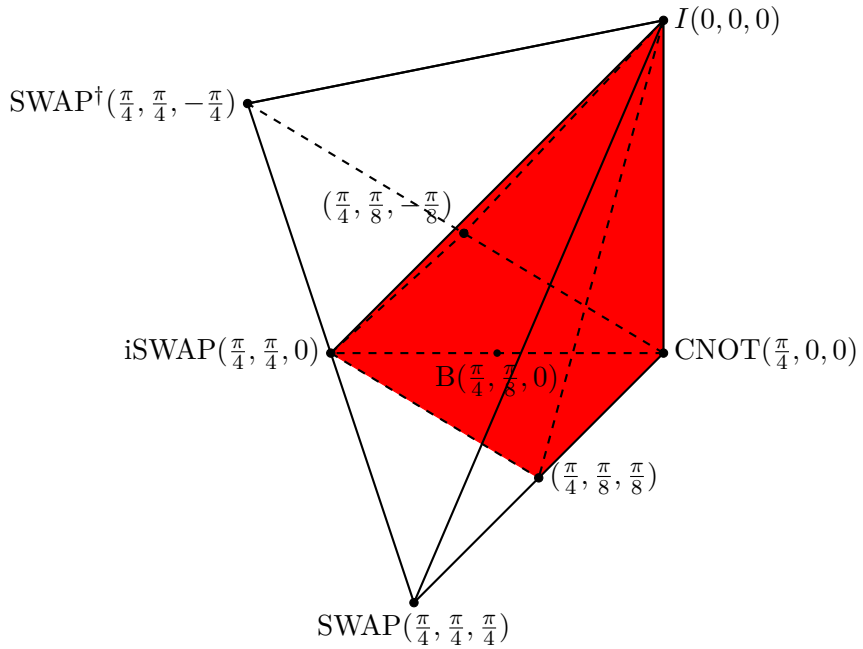
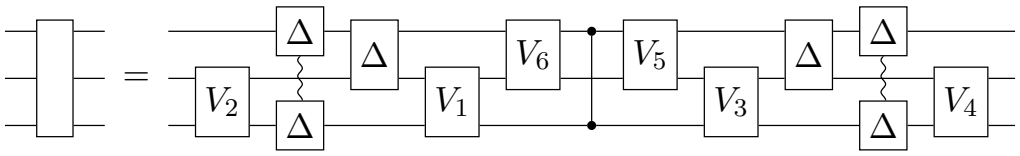
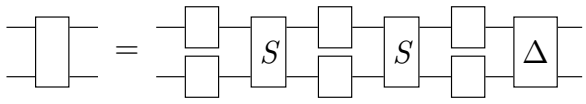


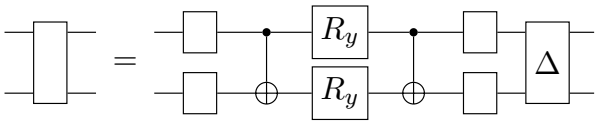
Figure 2: Region in Weyl chamber that can be spanned by 2 SQiSW gates, filled in red [32].



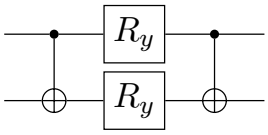
Lemma 5. *Any arbitrary 2-qubit gate can be synthesized using 2 SQiSW gates and 1 diagonal 2-qubit gate. The synthesis scheme is shown below, where the SQiSW gate is denoted as S.*



Proof. Firstly, the arbitrary 2-qubit gates can be synthesized by 2 CNOT gates and 1 diagonal 2-qubit gate, by [1]. So we have



Inside such scheme, at least 2 elements of interaction coefficients of the sub-circuit



are zeros (by the general calculation process for interaction coefficients in section 3.2 of [35]). So the sub-circuit above can be synthesized by 2 SQiSW gates. □

Now we have all the fragment contributing to Theorem 1. We now give the complete Theorem 1.

Theorem 1. *An arbitrary 3-qubit gate can be synthesized using a maximum of 24 SQiSW gates.*

Proof. First, we aim to prove that all diagonal gates in the scheme of Lemma 4 can be synthesized by at most 2 SQiSW gates.

Any diagonal matrix is locally equivalent to a R_{zz} gate, which is defined as

$$R_{zz}(\theta) = R_z(\theta) \otimes R_z(-\theta),$$

so we have

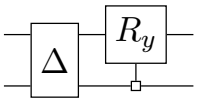
$$R_{zz}(\theta) = \begin{pmatrix} e^{-i\frac{\theta}{2}} & & & \\ & e^{i\frac{\theta}{2}} & & \\ & & e^{i\frac{\theta}{2}} & \\ & & & e^{-i\frac{\theta}{2}} \end{pmatrix}.$$

It's easy to see

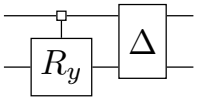
$$R_{zz}(\theta) = exp(-i\frac{\theta}{2} Z \otimes Z),$$

which tells us that the interaction coefficients of R_{zz} gate is $(0, 0, -\frac{\theta}{2})$. By Lemma 3, R_{zz} can be synthesized by 2 SQiSW gates. Thus all arbitrary diagonal gates can be synthesized by 2 SQiSW gates.

Next, we prove V_5 and V_6 can be synthesized by at most 2 SQiSW gates. In the original proof of Lemma 4 (see Theorem 12 in [19]), we get to know that V_5 and V_6 have the form like:



Rearrange qubits and reverse the circuit, we only need to prove the circuit



can be synthesized by at most 2 SQiSW gates. That's due to the second and third points of Property 2.

Since an arbitrary diagonal gate is locally equivalent to R_{zz} gate up to single qubit gates on the leftmost side or rightmost side, we only need to prove the interaction coefficients of unitary

$$\begin{pmatrix} R_z(\theta) & \\ & R_z(-\theta) \end{pmatrix} \begin{pmatrix} R_y(\theta_1) & \\ & R_y(\theta_2) \end{pmatrix}$$

locates in W', which is spanned by 2 SQiSW gates.

To simplify the calculation, we consider the unitary locally equivalent to it. By applying R_z

and R_y gates to two sides of it, we have

$$\begin{aligned}
&\begin{pmatrix} R_z(\theta) & \\ & R_z(-\theta) \end{pmatrix} \begin{pmatrix} R_y(\theta_1) & \\ & R_y(\theta_2) \end{pmatrix} \\
&\sim \begin{pmatrix} I & \\ & R_z(-2\theta) \end{pmatrix} \begin{pmatrix} R_y(\theta_1) & \\ & R_y(\theta_2) \end{pmatrix} \\
&\sim \begin{pmatrix} I & \\ & R_z(-2\theta) \end{pmatrix} \begin{pmatrix} I & \\ & R_y(\theta_2 - \theta_1) \end{pmatrix} \\
&= \begin{pmatrix} I & \\ & R_z(-2\theta) R_y(\theta_2 - \theta_1) \end{pmatrix},
\end{aligned}$$

where $\sim$ means being locally equivalent to. For simplicity's sake, we assume

$$U = \begin{pmatrix} I & \\ & R_z(\theta_1) R_y(\theta_2) \end{pmatrix}.$$

The interaction coefficients of U has 2 elements being zero. So U lies on the x-axis in Weyl chamber, thus U can be synthesized by at most 2 SQiSW gates. So V_5 and V_6 can all be synthesized by at most 2 SQiSW gates.

Then, we make use of Lemma 5 to optimize the circuit. Synthesizing V_1 and V_3 by the scheme in Lemma 5 allows us to move the diagonal gates from V_1 and V_3 across other diagonal gates in Lemma 4. Then they can be finally absorbed into V_4 and V_6.

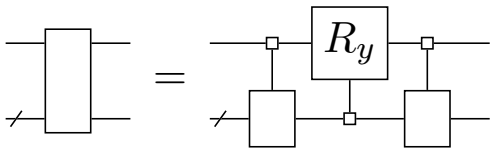
Also, We can synthesize the middle CNOT gate in circuit of Lemma 4 by 2 SQiSW gates, which is shown in Fig. 2. To conclude, we can synthesize any arbitrary 3-qubit gate by at most 24 SQiSW gates. □

Recall that the upper bound of synthesis of arbitrary 3-qubit gates using CNOT gate is 20 [1]. Our result is 24 for SQiSW gate, 40% better than trivially replacing each CNOT gate by 2 SQiSW gates.

3.2 Exact synthesis of arbitrary n-qubit gates

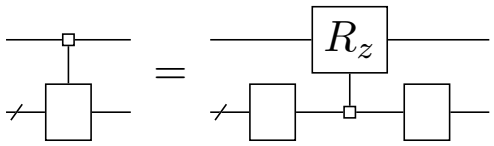
We already have the upper bound of synthesis of arbitrary 3-qubit gates, now we take Theorem 1 as the base case and prove Theorem 2, using Quantum Shannon Decomposition recursively and some circuit optimization techniques.

Lemma 6 (Cosine-Sine Decomposition [1, 2, 39])**.** *Any arbitrary n-qubit gate can be exactly synthesized as below.*



where $n \geq 2$.

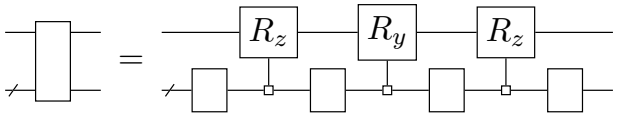
Lemma 7 (Demultiplexing of multiplexor [1])**.** *Any arbitrary n-qubit multiplexor with 1 controlled qubit can be exactly synthesized as below.*



where $n \geq 2$.

We can get a synthesis scheme from the combination of Lemma 6 and 7, which is called Quantum Shannon Decomposition.

Lemma 8 (Quantum Shannon Decomposition [1])**.** *Any arbitrary n-qubit gate can be exactly synthesized as below.*



where $n \geq 2$.

Quantum Shannon Decomposition gives a recursive decomposition scheme for arbitrary n-qubit gates, thus we have enough evidence to prove Theorem 2.

Denote c_l as the upper bound of number of SQiSW gates to synthesize any arbitrary l-qubit gate. Now we state the proof of Theorem 2 below.

Theorem 2. *An arbitrary n-qubit gate can be synthesized using a maximum of $\frac{139}{192} \times 4^n - 3 \times 2^n + \frac{5}{3}$ SQiSW gates.*

Proof. According to Lemma 8 and Lemma 1, we have

$$c_j \leq 4c_{j-1} + 3 \times 2^j, \ j \geq 2,$$

by replacing each CNOT gate by 2 SQiSW gates.

Then

$$c_n \leq 4^{n-l}(c_l + 3 \times 2^l) - 3 \times 2^n.$$

Now we introduce two circuit optimization techniques to decrease the upper bound.

First, notice that CZ gate is locally equivalent to CNOT gate (see Fig. 3), and Lemma 1 still holds after replacing all CNOT gates by CZ gates.

So the multiplexor-R_y gates in Lemma 6 can be synthesized by CZ gates with one CZ gate located on the leftmost side or rightmost side. Since CZ gate is diagonal, it can be absorbed into the adjacent multiplexor, saving $(4^{n-l}-1)/3$ CNOT (or CZ) gates in total, thereby saving $2(4^{n-l}-1)/3$ SQiSW gates.

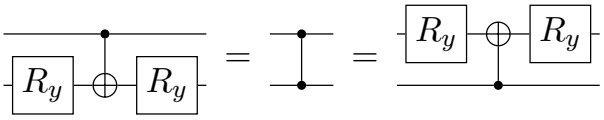


Figure 3: CNOT gate is locally equivalent to CZ gate, and a CZ gate is equal to a CNOT gate with R_y gates on both sides of its target qubit.

Second, stop recursion at $l = 3$ and consider Theorem 1. We can synthesize V_2 or V_4 by Lemma 5 and get to have a synthesis scheme for 3-qubit gates with a diagonal gate on the leftmost or rightmost side. All these diagonal gates are commutative with multiplexor gates through controlled qubits in Lemma 8. Then they can be absorbed into the neighboring 3-qubit gate. We get to save $4^{n-3}-1$ SQiSW gates in total.

Since we already have $c_3 = 24$, such two optimization techniques above help achieve an upper bound that

$$\begin{aligned} c_n &\le 4^{n-3}(c_3 + 3\times 2^3) - 3\times 2^n - 5(4^{n-3}-1)/3 \\ &= \frac{139}{192}\times 4^n - 3\times 2^n + \frac{5}{3}. \end{aligned}$$

Here, we have finished the proof of Theorem 2. □

Recall that the upper bound of synthesis of arbitrary n-qubit gates using CNOT gate is $\frac{23}{48}4^n(1+o(1))$ [1]. Our result is $\frac{139}{192}4^n(1+o(1))$ using SQiSW gate, 24% better than trivially replacing each CNOT gate by 2 SQiSW gates when n is large.

4 Numerical optimization and its analysis

Different from exact synthesis, numerical optimization aims to find a circuit numerically approximating the target circuit in synthesis problems. In this section, we run numerical optimization to synthesize arbitrary 3-qubit gates and Toffoli gate, studying how many SQiSW gates are sufficient to reach low enough error in the two tasks. Additionally, we give a reliable pruning algorithm and analyze its performance which is shown as Theorem 3.

In this section we denote n as the number of qubits, N as the number of SQiSW gates in the circuit. For the target function of optimization, we employ the known standard: the red"distance" or error between two circuits U and V is defined as

$$E(U,V) = 1 - \frac{|tr(U^\dagger V)|}{2^n}.$$

The stopping threshold of error in the optimization algorithm is set to 10^{-6}, which is nothing compared to the experimental error of realizing a quantum circuit.

To better describe the algorithm, we first introduce the concept of circuit structure.

Definition 3 (Circuit structure [24])**.** *The circuit structure of a quantum circuit is a sequence of positions to show the configuration of multi-qubit gates in the circuit.*

In this paper, we use $0 \sim n-1$ to show the position in a circuit with n qubits and in chronological order. For example, structure of the circuit in Fig. 4 is $(0,1),(1,2),(0,2)$.

4.1 Numerical optimization algorithm

We should look for the minimum number of SQiSW gates used in the circuit to numerically approximate target circuits. But we may have lots of structures to consider, and there may be a huge gap between performances of different structures.

We adopt the frame of two searching spaces, which are mentioned in [24]. More specifically, given the number of SQiSW gates, for each circuit structure (circuit structure space), we optimize the parameters in the circuit (parameter space) to approximate target circuit. When programming, they are actually two nested "for" loops. And we utilize qfactor package in python [40] to help the parameter optimization, which is a parameter learning tool. In the rest part of

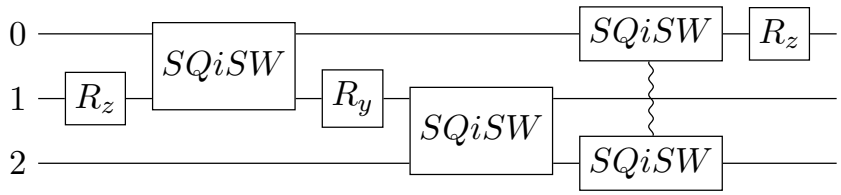


Figure 4: A quantum circuit with structure $(0,1),(1,2),(0,2)$.

this paper, we take it as a black box. The algorithms for Toffoli gate and arbitrary 3-qubit gates are shown in algorithm 1 and algorithm 2. Note that SQiSW gate is a deterministic gate, we only need to optimize parameters of single-qubit gates. In algorithm 2, we sample unitary matrices uniformly under Haar measure [41] and study the average error.

Algorithm 1: Numerical optimization for Toffoli gate.

Output: Deterministic circuit C numerically approximating Toffoli gate.

for *num of SQiSW gates* N **:**
  for *circuit structure* S **:**
    for i *in range(repetition_time)* **:**
      randomize($\vec{\theta}$);
      $\vec{\theta}$ = Learn(S, $\vec{\theta}$, Toffoli, learning parameters);
      E = dist($S(\vec{\theta})$, Toffoli);
      if $E \leq \epsilon$ **:**
        Return $S(\vec{\theta})$;

Algorithm 2: Numerical optimization for arbitrary 3-qubit gates.

Output: Circuit structure S numerically approximating arbitrary 3-qubit gates under acceptable average error.

for *num of SQiSW gates* N **:**
  for *circuit structure* S **:**
    lst_U = sample($SU(8)$,100);
    error = [ ];
    for U *in lst_U* **:**
      min_d = $+\infty$;
      for i *in range(repetition_time)* **:**
        randomize($\vec{\theta}$);
        $\vec{\theta}$ = Learn(S, $\vec{\theta}$, Toffoli, learning parameters);
        if *dist*$(S(\vec{\theta}), U)$ $<$*min_d* **:**
          min_d = dist$(S(\vec{\theta}), U)$;
      error.append(min_d);
    average = avg(error);
    if *average* $\leq \epsilon$ **:**
      Return S;

4.2 Pruning techniques for speed-up

In a n-qubit system, each SQiSW gate has $n(n-1)/2$ possible positions, which are $(0,1)$, $(0,2)$ and $(1,2)$ when $n = 3$. Recall SQiSW gate is qubit symmetrical, the qubit order of SQiSW gate doesn't matter.

That means we would have a structure space of 3^N size, leading to huge difficulties when N is large, which is intuitively shown in table 1.

The idea is to find these circuit structures that are "equivalent" under numerical optimization, thus we only need to keep one element of each class with little influence on the performance of our numerical optimization algorithms.

Table 1: The size of circuit structure space under different number of qubits n and number of SQiSW gates N

n \ N	5	10	15	50
2	1	1	1	1
3	243	59049	14348907	$> 10^{23}$
4	7776	60466176	$\sim 10^{11}$	$> 10^{36}$
5	10^5	10^{10}	10^{15}	10^{50}

We adopt two pruning techniques which are usually used in numerical optimization: qubit rearrangement and circuit reversion [24, 26], and give an explicit theoretical analysis of pruning efficiency when both of them are used in our synthesis tasks. To make our pruning algorithm theoretically reliable, we are supposed to make sure that such techniques make little influence on the numerical results. Here we define numerical equivalence and equivalence closure to help our proof of Theorem 3.

Definition 4 (Numerical equivalence)**.** *Given circuit structures C_1 and C_2, if*

$$\forall T \in \mathcal{T},\ \exists\ \vec{\theta}\ s.t.\ (C_1, \vec{\theta}) \to T$$

is equivalent to

$$\forall T \in \mathcal{T},\ \exists\ \vec{\theta'}\ s.t.\ (C_2, \vec{\theta'}) \to T,$$

where $\mathcal{T}$ is our target gate set, then we say C_1 and C_2 are numerically equivalent (or C_1 and C_2 are in the same equivalent class under numerical optimization).

Our target gate set in algorithm 1 and algorithm 2 are {Toffoli} and $SU(8)$ respectively. To better depict the equivalent class under more than one equivalence relation, we define equivalent closure.

Definition 5 (Equivalent closure)**.** *An equivalent closure $\mathcal{C}$ under operation set R is a maximal set s.t. $\forall\ e_1, e_2 \in \mathcal{C}$, e_1 can be generated by applying operations in R on e_2 finite times. Here R must guarantee that e_1 and e_2 are numerically equivalent.*

It can be verified that in algorithm 1 and algorithm 2, the circuit structures before and after being transformed by R are numerically equivalent. Here R contains "qubit rearrangement" and "circuit reversion", in which qubit rearrangement means rearranging the qubits order of the circuit structure and circuit reversion means reversing the gate order of the circuit.

So the whole circuit structure space in both algorithm 1 and algorithm 2 can be divided into different equivalent closures under R. Now we state algorithm 3, which successfully keeps exact one structure remained but kicks other structures out in each equivalent closure under R when it is applied to algorithm 1 and algorithm 2.

Algorithm 3: Pruning of circuit structure space [24, 26]

Input: Set of all possible circuit structures S.

Output: Set S with circuit structures remained after qubit rearrangement and circuit reversion.

1 **for** *circuit structure* $C \in S$ **:**
2 $S_1 \leftarrow$ circuit structures equivalent to C under qubit rearrangement;
3 $S_2 \leftarrow$ circuit structures equivalent to C under circuit reversion;
4 Remove circuits different from C in $S_1 \cup S_2$ from S;

Previously, there was no theoretical analysis conducted. Below, we will conduct a asymptotical analysis using both pruning methods simultaneously in numerical optimization.

To prove Theorem 3, we need to give explicit analysis of these closures, stated as below. Denote that f_{inv} is a function mapping a circuit structure space to its reversed structure, and f_{arr} is a function mapping a circuit structure to the set of circuit structures that only differ in qubit arrangement.

Theorem 3. *The pruning algorithm 3 reduces the size of circuit structure space (with 3 qubits and N quantum gates) in algorithm 1 and algorithm 2 to $\frac{1}{12} \times 3^N + 3^{\lfloor \frac{N-1}{2} \rfloor} + \frac{1}{4} = \frac{1}{12} 3^N (1 + o(1))$.*

Proof. Firstly we prove that there are only equivalent closures with 3 or 6 or 12 elements under R.

For circuit structure with gates having only one position, its closure only has 3 elements. For circuit structure with at least 2 positions of gates, it has 5 other different elements under qubit re-

arrangement. If a structure C' generated by reversing a circuit in rearrangement of original circuit C is equal to a structure in rearrangement of original circuit C, then the rearrangement of C' is exact the arrangement of C.

So either the reversion of arrangement of C is the same as arrangement of C or it has no same element with arrangement of C. In the first case, we have a equivalent closure with 6 elements, and in the other case we have a equivalent closure with 12 elements. They are the maximal set since we cannot generate a new element by apply any equivalent relation on any element of it.

The rest part of proof is on the calculation of number of each kind of equivalent closure. We only have one 3-element closure. The question is, how many circuits C are there satisfying

$$f_{inv}(C) \in f_{arr}(C),$$

where C has at least two different positions (that is, C is not an all-$(0,1)$ or all-$(0,2)$ or all-$(0,3)$ pattern).

The rearrangement function set is isomorphic to the permutation group

$$S_6 = \{f_1, f_2, f_3, \tau_1, \tau_2, \tau_3\},$$

in which

$$f_1 = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 1 & 2 \end{pmatrix},\ f_2 = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 0 & 1 \end{pmatrix},\ f_3 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 2 & 0 \end{pmatrix},$$

and

$$\tau_1 = \begin{pmatrix} 0 & 1 & 2 \\ 1 & 0 & 2 \end{pmatrix},\ \tau_2 = \begin{pmatrix} 0 & 1 & 2 \\ 2 & 1 & 0 \end{pmatrix},\ \tau_3 = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 2 & 1 \end{pmatrix}.$$

Without causing confusion, we take S_6 to represent the rearrangement function set, and each element in S_6 refers to the corresponding qubit rearrangement operation.

Then we calculate number of circuit C satisfying

$$f_{inv}(C) = f(C),\ f \in S_6. \tag{1}$$

We go into two situations:

Case 1. Here N is even. Assume $N = 2n$ and $C = a_1 a_2 \dots a_{2n}$ (according to the reading order of a quantum circuit), each a_i means a position of entangled gate and can be $(0,1)$, $(0,2)$ or $(1,2)$.

If $f = f_1$, then $inv(C) = C$, there are $3^n - 3$ circuits, since each $a_i = a_{2n+1-i}$ has 3 options and we exclude 3 circuits in the 3-element closure.

If $f = f_2$, we have

$$a_{2n+1-i} = f(a_i),$$
$$a_i = f(a_{2n+1-i}),$$

for $1 \le i \le 2n$. That is,

$$a_i = f(f(a_i)),$$

for $1 \le i \le 2n$. But it doesn't hold for any a_i. So there's no circuit satisfying (1).

If $f = f_3$, similarly there's no circuit satisfying (1).

If $f = \tau_1$, then

$$(a_i, a_{2n+1-i}) = ((0,1),(0,1)),$$
$$or\ (a_i, a_{2n+1-i}) = ((0,2),(1,2)),$$
$$or\ (a_i, a_{2n+1-i}) = ((1,2),(0,2)),$$

for $1 \le i \le n$. We have $3^n - 1$ circuits, excluding the circuit with only $(0,1)$.

If $f = \tau_2$, then

$$(a_i, a_{2n+1-i}) = ((0,2),(0,2)),$$
$$or\ (a_i, a_{2n+1-i}) = ((0,1),(1,2)),$$
$$or\ (a_i, a_{2n+1-i}) = ((1,2),(0,1)),$$

for $1 \le i \le n$. We have $3^n - 1$ circuits, excluding the circuit with only $(0,2)$.

If $f = \tau_3$, then

$$(a_i, a_{2n+1-i}) = ((1,2),(1,2)),$$
$$or\ (a_i, a_{2n+1-i}) = ((0,1),(0,2)),$$
$$or\ (a_i, a_{2n+1-i}) = ((0,2),(0,1)),$$

for $1 \le i \le n$. We have $3^n - 1$ circuits, excluding the circuit with only $(1,2)$.

So we have 1 closure with 3 elements, $\frac{1}{6}(4 \times 3^n - 6)$ closures with 6 elements, and $\frac{11}{12}[3^{2n} - (4 \times 3^n - 6) - 3]$ closures with 12 elements. Algorithm 3 successfully reduces the size of space to

$$\frac{1}{12} \times 3^N + 3^{\frac{N}{2}-1} + \frac{1}{4}.$$

Case 2. Here N is odd. Assume $N = 2n + 1$ and $C = a_1 a_2 \dots a_{2n+1}$.

If $f = f_1$, then

$$(a_i, a_{2n+2-i}) = ((0,1),(0,1)),$$
$$or\ (a_i, a_{2n+2-i}) = ((0,2),(0,2)),$$
$$or\ (a_i, a_{2n+2-i}) = ((1,2),(1,2)),$$

for $1 \le i \le n$. And a_{n+1} can be arbitrarily chosen. Thus we have $3^{n+1} - 3$ circuits. $f = f_2$ and

$f = f_3$ are all similar to corresponding situations that N is even.

If $f = \tau_1$, then

$$\begin{aligned}(a_i, a_{2n+1-i}) &= ((0,1),(0,1)),\\ or\ (a_i, a_{2n+1-i}) &= ((0,2),(1,2)),\\ or\ (a_i, a_{2n+1-i}) &= ((1,2),(0,2)),\end{aligned}$$

for $1 \leq i \leq n$. And $a_{n+1} = (0,1)$. We have $3^n - 1$ circuits, excluding the circuit with only $(0,1)$.

If $f = \tau_2$, then

$$\begin{aligned}(a_i, a_{2n+1-i}) &= ((0,2),(0,2)),\\ or\ (a_i, a_{2n+1-i}) &= ((0,1),(1,2)),\\ or\ (a_i, a_{2n+1-i}) &= ((1,2),(0,1)),\end{aligned}$$

for $1 \leq i \leq n$. And $a_{n+1} = (0,2)$. We have $3^n - 1$ circuits, excluding the circuit with only $(0,2)$.

If $f = \tau_3$, then

$$\begin{aligned}(a_i, a_{2n+1-i}) &= ((1,2),(1,2)),\\ or\ (a_i, a_{2n+1-i}) &= ((0,1),(0,2)),\\ or\ (a_i, a_{2n+1-i}) &= ((0,2),(0,1)),\end{aligned}$$

for $1 \leq i \leq n$. And $a_{n+1} = (1,2)$. We have $3^n - 1$ circuits, excluding the circuit with only $(1,2)$.

So we have 1 closure with 3 elements, $\frac{1}{6}(2 \times 3^{n+1} - 6)$ closures with 6 elements, and $\frac{11}{12}[3^{2n} - (2 \times 3^{n+1} - 6) - 3]$ closures with 12 elements. Algorithm 3 successfully reduces the size of space to

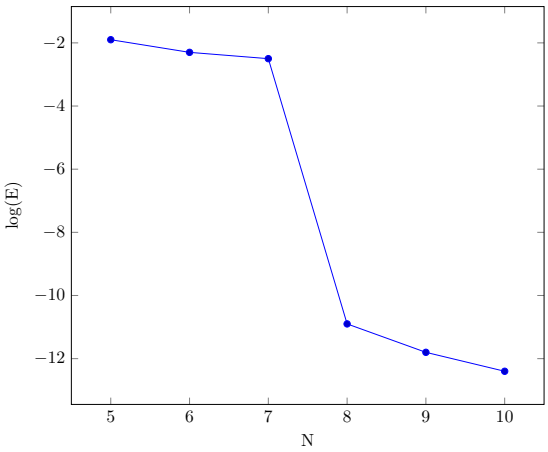
$$\frac{1}{12} \times 3^N + \frac{1}{2} \times 3^{\lfloor \frac{N}{2} \rfloor} + \frac{1}{4}.$$

Combining the two cases, we have finished the proof of Theorem 3. □

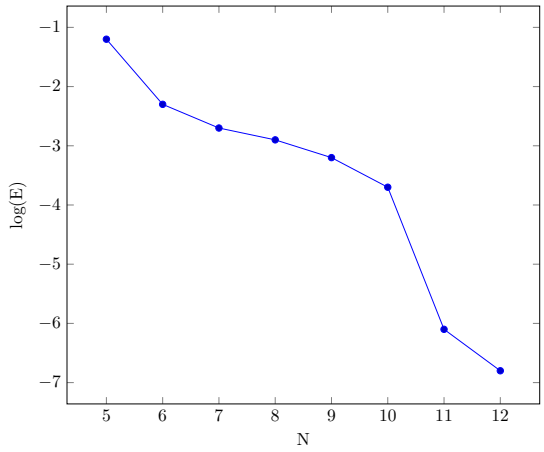
Theorem 3 tells that the pruning algorithm 3 reduces the size of circuit structure space to $\frac{1}{12} + o(1)$ of previous size, significantly increasing the efficiency of algorithm 1 and algorithm 2.

4.3 Our results and analysis

We apply the pruning algorithm and set the repetition time to 10 in both Algorithm 1 and Algorithm 2. Then, we present the results of the numerical optimization in a line chart, shown in Figure 5(a) and Figure 5(b). It is worth noting that the reason we start with $N = 5$ in algorithm 1 and algorithm 2 is that a Toffoli gate needs at least 5 2-qubit gates [42] and lower bound of 2-qubit gates synthesizing any arbitrary 3-qubit gate is 6 [15, 43].



(a) best error to approximate Toffoli gate



(b) best average error to approximate arbitrary 3-qubit gates

Figure 5: The logarithm of best error E in numerical optimization, using N SQiSW gates in the circuit.

Fig. 5(a) and Fig. 5(b) indicate that 8 and 11 SQiSW gates are sufficient to numerically approximate Toffoli gate and arbitrary 3-qubit gates respectively.

As a comparison, we only need 11 SQiSW gates to numerically approximate arbitrary 3-qubit gates, and the number for CNOT gates is 14 [19]. Thus SQiSW gate works better with aspect to circuit size in numerical optimization of arbitrary 3-qubit gates. For Toffoli gate, though we need 2 more entangled gates compared to CNOT gate [24], a single SQiSW gate only contributes to half of the experimental error of a single CNOT gate [32]. On the other hand, the numerical result can even help us derive an exactly synthesis scheme for Toffoli gate using 8 SQiSW gates, which is introduced in detail in section 5.

5 Exact synthesis of Toffoli gate

In this section, we prove that 8 SQiSW gates are sufficient to exactly synthesize Toffoli gate and give a feasible synthesis scheme, which is based on the observation of parameters in numerical optimization.

The idea is repeatedly rounding and retraining the rest parameters. For the rounding, it's all human work to select the suitable value. We round the numerical parameters to a near constant number. As an example, we round 3.14158 to π, 1.57106 to $\pi/2$ and fix them and retrain the other parameters without obvious pattern at this stage.

Sometimes it's not enough to just round the parameter to a constant number. We also try to observe and utilize the inner relation between different parameters. For instance, if parameters x and y satisfy $y = x + 3.14267$ throughout the training, then we believe $y = x + \pi$. This is another rounding strategy.

After such process, we finally derive the circuit with only one free parameter, shown in Fig. 6. Such a scheme can reach an error of even 10^{-16} in numerical optimization to approximate Toffoli gate, which is a strong evidence for the exact synthesis of Toffoli gate with 8 SQiSW gates.

Now we prove Theorem 4, by assigning appropriate value to the parameter remained and prove the existence of solution of the circuit equations.

Theorem 4. *Toffoli gate can be exactly synthesized by 8 SQiSW gates, a scheme shown in Fig. 6.*

Proof. Replace θ_1 by x. Calculate the matrix representation U of the circuit in Fig. 6, we have

$$
\begin{aligned}
U_{11} = U_{22} &= (-\frac{1}{4} + \frac{1}{4}i)(-1)^{\frac{3}{8}}(\cos\frac{x}{2} + \sin\frac{x}{2})(-2 + i\sqrt{2} + (2+\sqrt{2})\sin x),\\
U_{33} = U_{44} &= \frac{1}{2}(-1)^{\frac{7}{8}}(\cos\frac{x}{2} + \sin\frac{x}{2})(-i - \sqrt{2} + (1+\sqrt{2})\sin x),\\
U_{55} = U_{66} &= -\frac{(-1)^{\frac{1}{8}}(\cos\frac{x}{2} + \sin\frac{x}{2})((5-i) - (4-i)\sqrt{2} + (-1-i+\sqrt{2})\sin x)}{2(-1+(-1)^{\frac{1}{4}})^3},\\
U_{87} = U_{78} &= -\frac{1}{2}(-1)^{\frac{1}{8}}(\cos\frac{x}{2} + \sin\frac{x}{2})(i - \sqrt{2} + (1+\sqrt{2})\sin x).
\end{aligned}
$$

The rest elements of U are either 0 or a multiple of $(\cos\frac{x}{2} - \sin\frac{x}{2})(1 + (1+\sqrt{2})\sin x)$. So we derive equations by making U equal to the matrix representation of Toffoli gate:

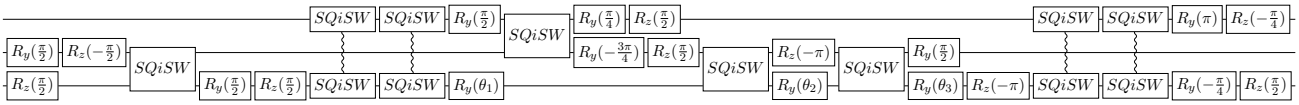


Figure 6: A synthesis scheme for Toffoli gate, using 8 SQiSW gates and several single-qubit gates, where $\theta_3 = \theta_2 + \pi/2 = \theta_1 + \pi$.

$$\begin{cases} (-\frac{1}{4}+\frac{1}{4}i)(-1)^{\frac{3}{8}}(\cos\frac{x}{2}+\sin\frac{x}{2})(-2+i\sqrt{2}+(2+\sqrt{2})\sin x) & = 1 \\ \frac{1}{2}(-1)^{\frac{7}{8}}(\cos\frac{x}{2}+\sin\frac{x}{2})(-i-\sqrt{2}+(1+\sqrt{2})\sin x) & = 1 \\ -\frac{(-1)^{\frac{1}{8}}(\cos\frac{x}{2}+\sin\frac{x}{2})((5-i)-(4-i)\sqrt{2}+(-1-i+\sqrt{2})\sin x)}{2(-1+(-1)^{\frac{1}{4}})^3} & = 1 \\ -\frac{1}{2}(-1)^{\frac{1}{8}}(\cos\frac{x}{2}+\sin\frac{x}{2})(i-\sqrt{2}+(1+\sqrt{2})\sin x) & = 1 \\ \cos\frac{x}{2}=\sin\frac{x}{2} \quad or \quad \sin x = 1-\sqrt{2}. \end{cases} \tag{2}$$

It can be verified that when the second equation of (2) holds, and the solution of it is in $(-\frac{\pi}{2}, \pi)$, we have

$$\begin{cases} U_{11} * U_{87} & = 1 \\ U_{33} * U_{87} & = 1 \\ U_{33} * U_{55} & = 1 \\ U_{87} & = 1. \end{cases}$$

So the circuit equations have an analytical solution, and a feasible solution is $x = \arcsin\left(1-\sqrt{2}\right)$, where $-\pi/2 < x < 0$.

Thus, the circuit in Fig. 6 is exactly equal to Toffoli gate, when

$$\theta_3 = \theta_2 + \frac{\pi}{2} = \theta_1 + \pi = \arcsin\left(1-\sqrt{2}\right) + \pi.$$

□

6 Summary

In this paper we propose an exact synthesis scheme for arbitrary n-qubit gates using only SQiSW gates and single-qubit gates. We utilize the algebraic properties of SQiSW gate to optimize the number of SQiSW gates required for synthesizing of an arbitrary 3-qubit gate, achieving a result of 24. Recursively, we provide an upper bound for the synthesis of an n-qubit gate, which is $\frac{139}{192} \times 4^n(1+o(1))$. Also, we state and analyze a pruning algorithm in numerical optimization of the synthesis problems using SQiSW gates, reducing the searching size to $\frac{1}{12}$ of previous size asymptotically. We conduct numerical experiment and find 8 and 11 SQiSW gates are enough for synthesis of Toffoli and arbitrary 3-qubit gates under acceptable error. Finally, we derive an exact synthesis scheme for Toffoli gate using only 8 SQiSW gates from numerical optimization, by observation of circuit parameters.

It remains open whether 8 is a tight result for synthesis of Toffoli gate. That's to say, whether it requires *at least* 8 SQiSW gates to synthesize Toffoli gate. We've already provided numerical evidence in this paper for the problem but the rigorous proof is still absent. Another interesting problem is, whether we can derive exact synthesis schemes in similar numerical-aided way in more complicate synthesis tasks. For example, can we raise an exact synthesis scheme for arbitrary 3-qubit gates using 11, or just less than 24 SQiSW gates by numerical observation? The essential difficulty is that Toffoli gate is a deterministic gate but "arbitrary" synthesis task has lots of free parameters in its target circuits. So it's hard to fix parameters explicitly and we must reserve enough free parameters in the final scheme. Also, how the optimization techniques used to achieve the bounds can be generalized is a question worth exploring. For example, one can try to use Weyl

chamber to discover more decomposition schemes with special structures (such as the circuit with "diagonal edge" in Lemma 5) that are advantageous for optimization.

7 Acknowledgment

The authors thank Longcheng Li, Jiadong Zhu, Ziheng Chen, Qi Ye and Jianxin Chen for helpful discussions. This work was supported in part by the National Natural Science Foundation of China Grants No. 62325210, 92465202, 62272441, and the Strategic Priority Research Program of Chinese Academy of Sciences Grant No. XDB28000000.

References

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

[2] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. "Quantum circuits for general multiqubit gates". Phys. Rev. Lett. **93**, 130502 (2004).

[3] Jiaqing Jiang, Xiaoming Sun, Shang-Hua Teng, Bujiao Wu, Kewen Wu, and Jialin Zhang. "Optimal space-depth trade-off of CNOT circuits in quantum logic synthesis". In Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA). Pages 213–229. (2020).

[4] Adriano Barenco, Charles H. Bennett, Richard Cleve, David P. DiVincenzo, Norman Margolus, Peter Shor, Tycho Sleator, John A. Smolin, and Harald Weinfurter. "Elementary gates for quantum computation". Phys. Rev. A **52**, 3457–3467 (1995).

[5] Alfred V. Aho and Krysta M. Svore. "Compiling quantum circuits using the palindrome transform" (2003). arXiv:quant-ph/0311008.

[6] Matthew Amy, Parsiad Azimzadeh, and Michele Mosca. "On the controlled-not complexity of controlled-not–phase circuits". Quantum Science and Technology **4**, 015002 (2018).

[7] Shuai Yang, Guojing Tian, Jialin Zhang, and Xiaoming Sun. "Quantum circuit synthesis on noisy intermediate-scale quantum devices". Phys. Rev. A **109**, 012602 (2024).

[8] G. Cybenko. "Reducing quantum computations to elementary unitary operations". Computing in Science & Engineering **3**, 27–32 (2001).

[9] Farrokh Vatan and Colin P. Williams. "Realization of a general three-qubit quantum gate" (2004). arXiv:quant-ph/0401178.

[10] E. Knill. "Bounds for approximation in total variation distance by quantum circuits" (1995). arXiv:quant-ph/9508007.

[11] Wei Zi, Junhong Nie, and Xiaoming Sun. "Shallow quantum circuit implementation of symmetric functions with limited ancillary qubits". IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **44**, 3060–3072 (2025).

[12] S.-B. Wang, P. Wang, G.-H. Li, et al. "Variational quantum eigensolver with linear depth problem-inspired ansatz for solving portfolio optimization in finance". Sci. China Inf. Sci. **68**, 180504:1–180504:11 (2025).

[13] E. Knill. "Approximation by quantum circuits" (1995). arXiv:quant-ph/9508006.

[14] Juha J. Vartiainen, Mikko Möttönen, and Martti M. Salomaa. "Efficient decomposition of quantum gates". Phys. Rev. Lett. **92**, 177902 (2004).

[15] V.V. Shende, I.L. Markov, and S.S. Bullock. "Smaller two-qubit circuits for quantum communication and computation". In Proceedings Design, Automation and Test in Europe Conference and Exhibition. Volume 2, pages 980–985. (2004).

[16] Vivek V. Shende and Igor L. Markov. "On the cnot-cost of toffoli gates" (2008). arXiv:0803.2316.

[17] Michael A. Nielsen and Isaac L. Chuang. "Quantum computation and quantum information: 10th anniversary edition". Cambridge University Press. USA (2011). 10th edition.

[18] Junhong Nie, Wei Zi, and Xiaoming Sun. "Quantum circuit for multi-qubit toffoli gate with optimal resource" (2024). arXiv:2402.05053.

[19] Jianxin Chen, Dawei Ding, Weiyuan Gong, Cupjin Huang, and Qi Ye. "One gate scheme to rule them all: Introducing a com-

plex yet reduced instruction set for quantum computing". In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2. Page 779–796. ASPLOS '24. Association for Computing Machinery (2024).

[20] Timothée Goubault de Brugière, Marc Baboulin, Benoît Valiron, and Cyril Allouche. "Synthesizing quantum circuits via numerical optimization". Page 3–16. Springer International Publishing. (2019).

[21] Esteban A Martinez, Thomas Monz, Daniel Nigg, Philipp Schindler, and Rainer Blatt. "Compiling quantum algorithms for architectures with multi-qubit gates". New Journal of Physics **18**, 063029 (2016).

[22] M. Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J. Coles. "Variational quantum state eigensolver". npj Quantum Information**8** (2022).

[23] Tomonori Shirakawa, Hiroshi Ueda, and Seiji Yunoki. "Automatic quantum circuit encoding of a given arbitrary quantum state". Phys. Rev. Res. **6**, 043008 (2024).

[24] Sahel Ashhab, Naoki Yamamoto, Fumiki Yoshihara, and Kouichi Semba. "Numerical analysis of quantum circuits for state preparation and unitary operator synthesis". Phys. Rev. A **106**, 022426 (2022).

[25] Sahel Ashhab, Fumiki Yoshihara, Miwako Tsuji, Mitsuhisa Sato, and Kouichi Semba. "Quantum circuit synthesis via a random combinatorial search". Phys. Rev. A **109**, 052605 (2024).

[26] Anouk Paradis, Jasper Dekoninck, Benjamin Bichsel, and Martin Vechev. "Synthetiq: Fast and versatile quantum circuit synthesis". Proceedings of the ACM on Programming Languages **8**, 55–82 (2024).

[27] Siyuan Niu, Akel Hashim, Costin Iancu, Wibe Albert De Jong, and Ed Younis. "Effective quantum resource optimization via circuit resizing in bqskit". In Proceedings of the 61st ACM/IEEE Design Automation Conference. DAC '24. Association for Computing Machinery (2024).

[28] Ed Younis and Emma Smith. "Bqskit github home page". `https://github.com/BQSKit`.

[29] Norbert Schuch and Jens Siewert. "Natural two-qubit gate for quantum computation using the XY interaction". Phys. Rev. A **67**, 032301 (2003).

[30] R. C. Bialczak, M. Ansmann, M. Hofheinz, E. Lucero, M. Neeley, A. D. O'Connell, D. Sank, H. Wang, J. Wenner, M. Steffen, A. N. Cleland, and J. M. Martinis. "Quantum process tomography of a universal entangling gate implemented with josephson phase qubits". Nature Physics **6**, 409–413 (2010).

[31] Deanna M. Abrams, Nicolas Didier, Blake R. Johnson, Marcus P. da Silva, and Colm A. Ryan. "Implementation of xy entangling gates with a single calibrated pulse". Nature Electronics **3**, 744–750 (2020).

[32] Cupjin Huang, Tenghui Wang, Feng Wu, Dawei Ding, Qi Ye, Linghang Kong, Fang Zhang, Xiaotong Ni, Zhijun Song, Yaoyun Shi, Hui-Hai Zhao, Chunqing Deng, and Jianxin Chen. "Quantum instruction set design for performance". Phys. Rev. Lett. **130**, 070601 (2023).

[33] Stephen S. Bullock and Igor L. Markov. "Smaller circuits for arbitrary n-qubit diagonal computations" (2003). arXiv:quant-ph/0303039.

[34] Robert R. Tucci. "An introduction to cartan's kak decomposition for qc programmers" (2005). arXiv:quant-ph/0507171.

[35] Byron Drury and Peter Love. "Constructive quantum shannon decomposition from cartan involutions". Journal of Physics A: Mathematical and Theoretical **41**, 395305 (2008).

[36] Navin Khaneja and Steffen J. Glaser. "Cartan decomposition of su(2n) and control of spin systems". Chemical Physics **267**, 11–23 (2001).

[37] Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation, and Jay M. Gambetta. "Validating quantum computers using randomized model circuits". Phys. Rev. A **100**, 032328 (2019).

[38] Jun Zhang, Jiri Vala, Shankar Sastry, and K. Birgitta Whaley. "Geometric theory of nonlocal two-qubit operations". Phys. Rev. A **67**, 042313 (2003).

[39] Robert R. Tucci. "A rudimentary quantum compiler(2cnd ed.)" (1999). arXiv:quant-ph/9902062.

[40] Alon Kukliansky, Ed Younis, Lukasz Cincio, and Costin Iancu. "Qfactor: A domain-specific optimizer for quantum circuit instantiation". In 2023 IEEE International Conference on Quantum Computing and Engineering (QCE). Page 814–824. IEEE (2023).

[41] Francesco Mezzadri. "How to generate random matrices from the classical compact groups" (2007). arXiv:math-ph/0609050.

[42] Nengkun Yu, Runyao Duan, and Mingsheng Ying. "Five two-qubit gates are necessary for implementing the toffoli gate". Phys. Rev. A **88**, 010304 (2013).

[43] Vivek V. Shende, Igor L. Markov, and Stephen S. Bullock. "Minimal universal two-qubit controlled-not-based circuits". Phys. Rev. A **69**, 062321 (2004).