1-2020

An Extended Approach for Generating Unitary Matrices for Quantum Circuits

Zhiqiang Li
Yangzhou University

Wei Zhang
Yangzhou University

Gaoman Zhang
Yangzhou University

Juan Dai
Yangzhou University

Jiajia Hu
Yangzhou University

See next page for additional authors

Citation Details

Authors

Zhiqiang Li, Wei Zhang, Gaoman Zhang, Juan Dai, Jiajia Hu, Marek Perkowski, and Xiaoyu Song

An Extended Approach for Generating Unitary Matrices for Quantum Circuits

Zhiqiang Li[1, *], Wei Zhang[1], Gaoman Zhang[1], Juan Dai[1], Jiajia Hu[1], Marek Perkowski[2] and Xiaoyu Song[2]

Abstract: In this paper, we do research on generating unitary matrices for quantum circuits automatically. We consider that quantum circuits are divided into six types, and the unitary operator expressions for each type are offered. Based on this, we propose an algorithm for computing the circuit unitary matrices in detail. Then, for quantum logic circuits composed of quantum logic gates, a faster method to compute unitary matrices of quantum circuits with truth table is introduced as a supplement. Finally, we apply the proposed algorithm to different reversible benchmark circuits based on NCT library (including NOT gate, Controlled-NOT gate, Toffoli gate) and generalized Toffoli (GT) library and provide our experimental results.

Keywords: Quantum circuit, unitary matrix, quantum logic gate, reversible circuit, truth table.

1 Introduction

Quantum computation has become a very active and promising field of research and machine learning has been transformed a lot through the introduction of quantum computing [Liu, Gao, Wang et al. (2019)]. Quantum computation has gained a lot of advances in experimental physics and engineering which have made possible to build the quantum computer prototypes. A quantum computer is built from a quantum circuit containing wires and elementary quantum gates to carry around and manipulate the quantum information. In this respect, quantum gates and quantum circuits play an essential role in quantum computation. Usually, we utilize a unitary matrix to describe the quantum gate. The unitary matrix is a mathematical model and precisely reflect the mathematical properties of the quantum gates. This applies to quantum circuits as well because quantum circuits are cascaded by elementary quantum logic gates according to a certain logical relationship. On the basis of it, we propose an algorithm for calculating unitary matrices corresponding to quantum circuits automatically.

Reversible circuits are capable to realize the reversible function using the reversible gates. All the quantum operations are necessarily reversible in nature and mapping the reversible circuit to its equivalent quantum circuit efficiently is the area of keen interest [Handique and Sonkar (2018)]. In this paper, quantum circuits are divided into six types

---

[1] College of Information Engineering, Yangzhou University, Yangzhou, 225000, China.

[2] Department of Electrical & Computer Engineering, Portland State University, Portland, Oregon, USA.

[*] Corresponding Author: Zhiqiang Li. Email: yzqqlzq@163.com.

and we consider the reversible circuits of NCT and GT library as benchmark circuits with the .tfc file [Maslov, Dueck and Scott (2009)] as the storage for circuits.

The remainder of this paper is structured as follows. The next section briefly reviews the basics of reversible and quantum circuits. Later in this section, it gives the information about the machine-readable formatted .tfc file and the six types of quantum circuits. Section 3 introduces the proposed algorithm and mentions the truth table method as a supplement to this algorithm. Afterwards, we apply the proposed algorithm to different reversible benchmark circuits and provide our experimental results in Section 4. Finally, conclusions are drawn in Section 5.

2 Preliminaries

This section describes the background necessary for this paper. Readers interested in more detail should consult the literature [Nielsen and Chuang (2000)].

2.1 Reversible gates and circuits

The function is reversible if it has the same number of inputs and outputs and maps each unique input of the function to a unique output and vice versa. A reversible function can be realized by a circuit comprised of a cascade of reversible gates with no fan-out and feedback [Nielsen and Chuang (2000)]. Some of the fundamental reversible gates have been introduced including NOT gate, Controlled-NOT gate (CNOT, also known as Feynman gate), Toffoli gate [Toffoli (1980)] and the family of these gates is called NCT library.

A multiple-control Toffoli (MCT) gate has a target line and several control lines. The target line is inverted if all the controls have value 1 (e.g., in Fig. 1(c), the target line t is inverted, if and only if all control lines c1 to c2 are equal to 1.), otherwise the value on the target line is passed through unchanged. The values on the control lines and unconnected lines always pass through the gate unchanged. An MCT gate with no control lines acts like the NOT gate and with one and two control line(s), the MCT gate is called CNOT and Toffoli gate respectively as shown in Fig. 1. The family of MCT gate is called generalized Toffoli (GT) library [Maslov, Dueck and Scott (2009)]. MCT gates are universal in that all reversible functions can be realized using this gate type alone [Toffoli (1980)]. The Fredkin et al. [Fredkin and Toffoli (1982)] and the Peres gate [Thapliyal and Ranganathan (2010); Peres (1985)] can, for example, be realized using MCT gates [Sasanian, Wille and Miller (2012)].
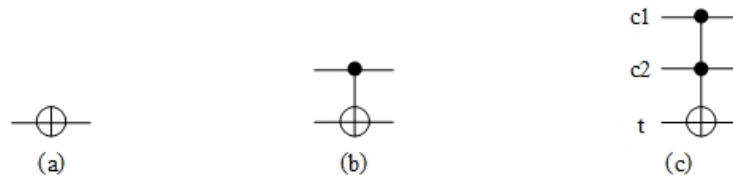
Figure 1: (a) NOT gate, (b) CNOT gate, (c) Toffoli gate

2.2 Quantum gates and circuits

Quantum gates are the basic unit of quantum circuits and a cascade of quantum gates constitutes the quantum circuit. There are a potentially infinite number of distinct quantum gates. However, in practice, researchers consider circuits composed of a small number of gate types [Sasanian, Wille and Miller (2012)].

The NCV library Li et al. [Li, Chen, Song et al. (2017)] is the fundamental library introduced by Barenco et al. [Barenco, Bennett, Cleve et al. (1995)] in quantum circuits. The NCV library can realize any reversible Boolean function. Therefore, the NCV library is the universal quantum logic gate library that contains the following set of quantum gates:

1) NOT gate: It inverts a single qubit described by the unitary matrix $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$.

2) CNOT gate: The target qubit is inverted if the control qubit is 1.

3) Controlled-V gate: As shown in Fig. 2(a), it performs a V operation on the target qubit if the control qubit is 1. The operation is described by the unitary matrix $V = \frac{1+i}{2} \begin{bmatrix} 1 & -i \\ -i & 1 \end{bmatrix}$.

4) Controlled-$V^\dagger$ gate: As shown in Fig. 2(b), it performs a $V^\dagger$ operation on the target qubit if the control qubit is 1 and unitary matrix describes as $V^\dagger = \frac{1-i}{2} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix}$.

**Figure 2:** (a) Controlled-V gate, (b) Controlled-$V^\dagger$ gate

Although the NCV gate library is universal, other libraries are of interest as they can lead to better circuits, e.g., fewer gates. In Sasanian et al. [Sasanian, Wille and Miller (2012)], a modification to the NCV gate library is introduced based on the concept of multi-valued logic [Muthukrishnan and Stroud (2000)]. However, in this work we do not consider the multivalued logic gates i.e., the NCV-$|v_1\rangle$ gate library mentioned in Sasanian et al. [Sasanian, Wille and Miller (2012)].

2.3 Machine-readable format: .tfc file

The .tfc file Maslov et al. [Maslov, Dueck and Scott (2009)] is the machine-readable format for reversible circuits. It contains detailed information about reversible circuits such as input lines, output lines, gate netlist, garbage lines and constant inputs. In short, the .tfc file provides a unified, formatted file storage method for reversible circuits that allows us to quickly read and identify reversible circuits. The detailed concept of the .tfc file can be found at Maslov et al. [Maslov, Dueck and Scott (2009); Handique and Sonkar (2018)].

2.4 Classification of quantum circuits

In this section, we divide quantum circuits into six basic types in Fig. 3 [Samoladas (2008)]. Here, U is an n-qubit unitary operator. When U is below the quantum circuit, according to whether there is a control point above the gate, the circuit can be divided into two types, as shown in Figs. 3(a) and 3(b). When U is above the quantum circuit, there are two types circuits similarly, as shown in Figs. 3(d) and 3(e). If the target qubit is flipped when the control qubit is zero, we draw a hollow circle. We can get two more types of circuits as shown in Figs. 3(c) and 3(f). In addition, U can be any of the six circuits mentioned above, which implements a nested way to describe general quantum circuits.
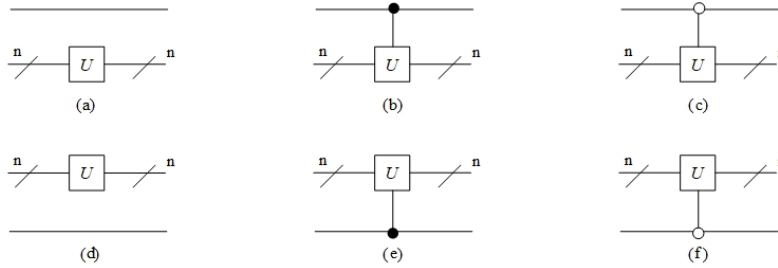
Figure 3: Six constructions based on U and corresponding circuits

1) In Fig. 3 the expression for (a) is:

$$U_A = I_1 \otimes U_n = \begin{bmatrix} U_n & 0 \\ 0 & U_n \end{bmatrix} \tag{1}$$

2) In Fig. 3 the expression for (b) is slightly more complex than (a):

$$U_B = |0\rangle\langle 0| \otimes I_n + |1\rangle\langle 1| \otimes U_n = \begin{bmatrix} I_n & 0 \\ 0 & U_n \end{bmatrix} \tag{2}$$

3) The expression for (c) in Fig. 3 is:

$$U_C = |1\rangle\langle 1| \otimes I_n + |0\rangle\langle 0| \otimes U_n = \begin{bmatrix} U_n & 0 \\ 0 & I_n \end{bmatrix} \tag{3}$$

4) The expression for (d) in Fig. 3 can be obtained by exchanging two operands of $\otimes$ operation of (a):

$$U_D = U_n \otimes I_1 \tag{4}$$

5) Similarly, the expression for (e) in Fig. 3 is:

$$U_E = I_n \otimes |0\rangle\langle 0| + U_n \otimes |1\rangle\langle 1| \tag{5}$$

6) The expression for (f) in Fig. 3 is:

$$U_F = I_n \otimes |1\rangle\langle 1| + U_n \otimes |0\rangle\langle 0| \tag{6}$$

When n=1, the matrices of U_B and U_D are separately presented as below [Samoladas (2008)]:

$$controlled\text{-}U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & u_{11} & u_{12} \\ 0 & 0 & u_{21} & u_{22} \end{bmatrix}, \quad upside\text{-}down\text{-}controlled\text{-}U = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & u_{11} & 0 & u_{12} \\ 0 & 0 & 1 & 0 \\ 0 & u_{21} & 0 & u_{22} \end{bmatrix}, \text{ where}$$

$U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$ is an arbitrary 1-qubit gate. I_n and U_n respectively represent the $2^n \times 2^n$ identity matrix and the $2^n \times 2^n$ unitary matrix.

3 Proposed method

The proposed method is able to generate the unitary matrix of the quantum circuit. We use a non-recursive approach with great simplification of this algorithm. Besides we use NumPy library and SymPy library in Python. NumPy is the fundamental package for scientific computing and SymPy is a Python library for symbolic mathematics. This method considers quantum circuits composed of NCT gates, MCT gates, GT gates, NCV gates, one-qubit gates and the gate whose target qubit is a one-qubit gate. Now we describe the key content of the proposed method as shown in Tab. 1.

Table 1: The key of the proposed algorithm

Input: the quantum circuit
Output: the unitary matrix of the quantum circuit

1	gates[]=the quantum gates in the circuit
2	n=the number of quantum lines in the circuit
3	circuit_matrix=the $2^n \times 2^n$ identity matrix
4	**for** i=0 **to** length(gates)-1 **do**
5	circuit_matrix=GateMatrix (gates[i], n)$\times$ circuit_matrix
6	**end for**
7	**return** circuit_matrix
8	**function** GateMatrix (mygate, n)
9	v[]=all the points in mygate
10	idx=-1
11	**for** i=0 **to** n-1
12	**if** v[i] is the target point or a single qubit gate **then**
13	idx=i
14	**end for**
15	**if** idx=-1 **then return** null
16	gate_matrix=the matrix of v[idx]
17	m=0

18   **for** i=idx-1 **to** 0 **do**

19   m=m+1

20   **if** v[i] is the 1-controlled point **then**

21   gate_matrix=$|0\rangle\langle0|\otimes I_m +|1\rangle\langle1|\otimes$ gate _ matrix

22   **else if** v[i] is the 0-controlled point **then**

23   gate_matrix=$|1\rangle\langle1|\otimes I_m +|0\rangle\langle0|\otimes$ gate _ matrix

24   **else**

25   gate_matrix=$I_1\otimes$ gate _ matrix

26   **end if**

27   **end for**

28   **for** i=idx+1 **to** n-1 **do**

29   m=m+1

30   **if** v[i] is the 1-controlled point **then**

31   gate_matrix=$I_m\otimes|0\rangle\langle0|+$ gate _ matrix $\otimes|1\rangle\langle1|$

32   **else if** v[i] is the 0-controlled point **then**

33   gate_matrix=$I_m\otimes|1\rangle\langle1|+$ gate _ matrix $\otimes|0\rangle\langle0|$

34   **else**

35   gate_matrix= gate _ matrix $\otimes I_1$

36   **end if**

37   **end for**

38   **return** gate_matrix

39 **end function**

The proposed algorithm can calculate unitary matrices for complex quantum circuits such as Fig. 4 [Garcia-Martin and Sierra (2017)].

Figure 4: A Toffoli gate implemented as the product of 1-qubit gates and CNOTs
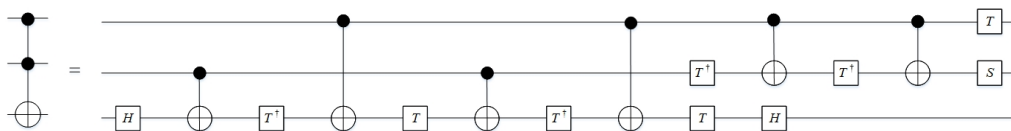
In addition, the truth table method can be used to calculate unitary matrices of reversible logic circuits. The quantum circuit being computed should be able to be described by a truth table. Although the truth table method is faster than proposed method, it is only suitable for quantum reversible logic circuit.

In the truth table, the input value of a quantum circuit ranges from 0 to $2^n -1$. The input vector corresponding to the input value i is x_i. In the x_i, the i-th number (starting from

zero) is 1 and the rest are 0. Suppose the matrix of the quantum circuit is A, under the action of quantum circuits, the output vector is y_i. Clearly, y_i is the i-th column of A, as expressed in the following equations. So we can calculate unitary matrices for quantum circuits by input and output logic values.

$$Ax_i = \begin{bmatrix} a_{00} & \cdots & a_{0i} & \cdots & \cdots & a_{0N} \\ \vdots & & \vdots & & & \vdots \\ \vdots & & a_{ji} & & & \vdots \\ \vdots & & \vdots & & & \vdots \\ \vdots & & \vdots & & & \vdots \\ a_{N0} & \cdots & a_{Ni} & \cdots & \cdots & a_{NN} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}_i = \begin{bmatrix} a_{0i} \\ \vdots \\ a_{ji} \\ \vdots \\ \vdots \\ a_{Ni} \end{bmatrix} = y_i \tag{7}$$

The following is a specific description of the truth table method, taking the quantum circuit in Fig. 5(a) as an example which is equivalent to a SWAP gate [Garcia-Martin and Sierra (2017)]. It is enough to look at what happens to each state 00, 01, 10, and 11. Let us consider 01. The first gate CNOT does nothing since the control is 0. The second gate reversed CNOT flips the first qubit, so we have 11. Finally, the last CNOT flips the second qubit and we get 10. The other cases can be worked out similarly. Tab. 2 shows the truth table of this quantum circuit. The tensor product of the output states corresponds to each column of the matrix and according to the expression (7) above, the unitary matrix can be obtained i.e., (y_0, y_1, y_2, y_3) as shown in Fig. 5(b).

Table 2: The truth table of the equivalent SWAP gate

Input		Input vector	Output		Output vector
0	0	$x_0=(1, 0, 0, 0)^{-1}$	0	0	$y_0=(1, 0, 0, 0)^{-1}$
0	1	$x_1=(0, 1, 0, 0)^{-1}$	1	0	$y_1=(0, 0, 1, 0)^{-1}$
1	0	$x_2=(0, 0, 1, 0)^{-1}$	0	1	$y_2=(0, 1, 0, 0)^{-1}$
1	1	$x_3=(0, 0, 0, 1)^{-1}$	1	1	$y_3=(0, 0, 0, 1)^{-1}$
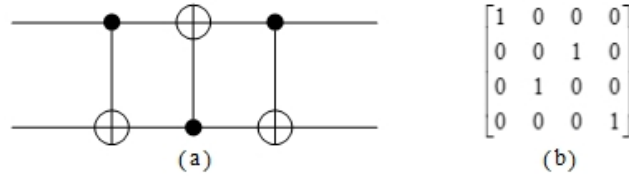
Figure 5: Equivalent SWAP gate and corresponding matrix

4 Experimental results

The proposed algorithm is implemented in Python (Version 3.7) on an Intel Core i3 with 6 GB RAM, clock speed 2.4 GHz, 256 GB SSD. The proposed algorithm has been applied to different reversible benchmark circuit based on NCT and GT libraries as

shown in Tab. 3. From the results, when the number of circuit lines is above 7, there is a significant increase in calculation time and using the truth table method, the optimal computing time we get is only 12 ms in the process of calculating the quantum circuit hwb8_637.

Table 3: Time of generating unitary matrices for reversible benchmark circuits

Benchmark Circuits	Gate Count	Quantum Cost	Gate Library	Time/ms
4b15g_1	15	47	GT	234
mspk_4b15g_1	15	39	GT	233
4b15g_2	15	61	GT	236
mspk_4b15g_2	15	31	NCT	235
4b15g_3	15	53	GT	231
mspk_4b15g_3	15	33	NCT	233
4b15g_4	15	47	GT	228
mspk_4b15g_4	15	35	GT	232
4b15g_5	15	43	NCT	232
mspk_4b15g_5	15	29	NCT	229
ham7-23-81	23	81	GT	173268
ham7-21-65	21	65	GT	167965
ham7-25-49	25	49	NCT	191774
hwb4-11-23	11	23	NCT	174
hwb4-11-21	11	21	NCT	173
mspk-hwb4-12	12	20	NCT	188
mspk-hwb4-13	13	19	NCT	204
hwb8_637	637	16522	GT	42324557

5 Conclusions

In this paper, quantum circuits are divided into six basic types and we discuss an approach for generating unitary matrices of quantum circuits. The algorithm by the aforementioned method is not fully optimized, and further optimization is possible. Our future work, we will concentrate on the improvement of the algorithm, how to incorporate the constraints into the truth table method and the simulation experiments based on IBM Q [Liu, Gao, Yu et al. (2018)].

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Barenco, A.; Bennett, C. H.; Cleve, R.; Divincenzo, D. P.; Margolus, N. et al. (1995): Elementary gates for quantum computation. *Physical Review A*, vol. 52, no. 5, pp. 3457-3467.

Fredkin, E.; Toffoli, T. (1982): Conservative logic. *International Journal of Theoretical Physics*, vol. 21, no. 3, pp. 219-253.

Garcia-Martin, D.; Sierra, G. (2017): Five experimental tests on the 5-qubit IBM quantum computer. *Journal of Applied Mathematics and Physics*, vol. 6, no. 7, pp. 1460-1475.

Handique, M.; Sonkar, A. (2018): An extended approach for mapping reversible circuits to quantum circuits using NCV-$|v_1\rangle$ library. *Procedia Computer Science*, vol. 125, pp. 832-839.

Li, Z.; Chen, S.; Song, X.; Perkowski, M.; Chen, H. et al. (2017): Quantum circuit synthesis using a new quantum logic gate library of NCV quantum gates. *International Journal of Theoretical Physics*, vol. 56, no. 4, pp. 1023-1038.

Liu, W.; Gao, P.; Wang, Y.; Yu, W.; Zhang, M. (2019): A unitary weights based one-iteration quantum perceptron algorithm for non-ideal training sets. *IEEE Access*, vol. 7, pp. 36854-36865.

Liu, W.; Gao, P.; Yu, W.; Qu, Z.; Yang, C. (2018): Quantum relief algorithm. *Quantum Information Processing*, vol. 17, no. 10, pp. 280.

Maslov, D.; Dueck, G.; Scott, N. (2009): Reversible logic synthesis benchmark. http://webhome.cs.uvic.ca/~dmaslov/.

Muthukrishnan, A.; Stroud, C. R. (2000): Multivalued logic gates for quantum computation. *Physical Review A*, vol. 62, no. 5, 052309.

Nielsen, M. A.; Chuang, I. L. (2000): *Quantum Computation and Quantum Information*. Cambridge University Press, USA.

Peres, A. (1985): Reversible logic and quantum computers. *Physical Review A*, vol. 32, no. 6, pp. 3266-3276.

Samoladas, V. (2008): Improved BDD algorithms for the simulation of quantum circuits. *European Symposium on Algorithms*, vol. 5193, pp. 720-731.

Sasanian, Z.; Wille, R.; Miller, D. M. (2012): Realizing reversible circuits using a new class of quantum gates. *Proceedings of the 49th Annual Design Automation Conference, DAC '12*, pp. 36-41.

Thapliyal, H.; Ranganathan, N. (2010): Design of reversible sequential circuits optimizing quantum cost, delay, and garbage outputs. *ACM Journal on Emerging Technologies in Computing Systems*, vol. 6, no. 4, pp. 1-31.

Toffoli, T. (1980): Reversible computing. *International Colloquium on Automata, Languages, and Programming*, vol. 85, pp. 632-644.

Williams, C. P. (2010): *Explorations in Quantum Computing*. Springer London Ltd., London.