Quantum Inf Process (2018) 17:61
https://doi.org/10.1007/s11128-018-1832-y

Multi-strategy based quantum cost reduction of linear nearest-neighbor quantum circuit

Ying-ying Tan[1] · Xue-yun Cheng[1,2] · Zhi-jin Guan[1] · Yang Liu[1] · Haiying Ma[1]

Received: 4 July 2017 / Accepted: 19 January 2018 / Published online: 31 January 2018

Abstract With the development of reversible and quantum computing, study of reversible and quantum circuits has also developed rapidly. Due to physical constraints, most quantum circuits require quantum gates to interact on adjacent quantum bits. However, many existing quantum circuits nearest-neighbor have large quantum cost. Therefore, how to effectively reduce quantum cost is becoming a popular research topic. In this paper, we proposed multiple optimization strategies to reduce the quantum cost of the circuit, that is, we reduce quantum cost from MCT gates decomposition, nearest neighbor and circuit simplification, respectively. The experimental results show that the proposed strategies can effectively reduce the quantum cost, and the maximum optimization rate is 30.61% compared to the corresponding results.

Keywords Quantum circuit · Linear nearest neighbor · Quantum cost · MCT decomposition

1 Introduction

In recently years, due to the density of power dissipation, the goal to further minimize the size of transistor becomes obstacle. And with the promotion of two decade principles of Landauer [1] and Bennett [2], people become interested in reversible and quantum computation, and synthesis of reversible and quantum circuits has also become an active research area. The main study of synthesis of reversible circuits is constructing the optimal circuits with the least quantum cost under the limits of given

✉ Zhi-jin Guan
guan_g617@163.com

1 College of Computer Science and Technology, Nantong University, Nantong 226019, China

2 College of Electronic Information, Nantong University, Nantong 226019, China

reversible gates and constraint conditions. The exact method can get the optimal circuit, but the time and space complexity is very high. So the method is mainly suitable for small-scale circuit [3]. The heuristic method generally adopts the two-stage synthesis [4,5] way. At first, some heuristic methods such as truth table and decision graph are used to realize the reversible circuit from the corresponded reversible function. And then, ways of making circuit reorganization, replacement and logic gate simplification are applied to reduce the cost of reversible circuit without changing the function of the reversible circuit.

However, most of the existing synthesis methods do not consider some of technological constraints, like ion-traps [6], requiring that all interactions take place within adjacent qubits only. This has led researchers to explore new methods for synthesis and optimization under the nearest-neighbor constraints.

Methods for realizing the nearest-neighbor circuit are mostly based on modifying the arrangement of qubits [7–12], what can be broadly classified into two categories. The first category, known as global ordering [7–9], the arrangement of qubits is modified globally. The second category, known as local ordering [10–12], considers smaller parts of circuits and tries to locally insert SWAP gates. In addition to realizing the nearest-neighbor circuit, quantum circuit optimization, which reduces the quantum cost of circuit is also popular in recent years. In [13], the authors reduce quantum cost by decreasing the number of the common control lines. In [14], template matching method is used as well. And in [15], the method for reducing quantum cost is changing the order of gates in a circuit. However, these methods are not based on the nearest-neighbor and have some drawbacks, such as uncompleted templates and high time complexity.

Therefore, the paper proposed three optimization strategies for linear nearest-neighbor circuit to reduce quantum cost in three aspects, respectively, including MCT gates decomposition, making circuit nearest neighbor and simplifying circuit.

2 Background

A Boolean function F is reversible if and only if it is bijective. In other words, every input vector is uniquely mapped to an output vector and vice versa. A reversible circuit f consists of a cascade of reversible gate without fanout or feedback [16].

In conventional computing, two-valued bits, 0 and 1 are used. However, in quantum computing, qubits are used in computing. The state of a qubit can be represented as a linear combination of computational basis states $|0>$ and $|1>$ as a sum of two complex numbers, i.e.,

$$|\varphi>= \alpha|0> +\beta|1>$$

where α and β are complex numbers, and $|\alpha|^2 + |\beta|^2 = 1$. That is, a qubit can exhibit superposition of basis states, what is one of the differences between conventional and quantum computing.

There are many different kinds of quantum gate libraries in quantum logic synthesis, such as NCV and NCVW. These optimization strategies we proposed are suitable for

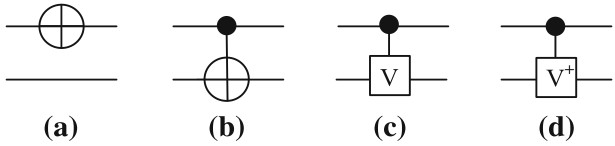
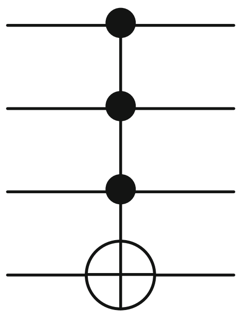


Fig. 1 NCV library

Fig. 2 Multiple-control Toffoli gate



arbitrary gate libraries. However, in the paper, we mainly use MCT and NCV gate library as the descriptions and examples.

In the so-called NCV library, four unitary operations are defined: NOT, controlled-NOT, controlled-V and controlled-V^{+} as shown in Fig. 1. Here, the control bits are denoted by $\cdot$, while the target bits are denoted by $\oplus$, a V box, or a V^{+} box.

NOT, controlled-NOT, Toffoli (control bits are two) and multiple-control Toffoli gate (control bits generally are over two), as shown in Fig. 2, are included in MCT gate library. It has already been very mature to synthesize the reversible function into a reversible MCT circuit [3,17–19]. When a reversible MCT circuit is converted to a quantum circuit, the MCT gate is needed to decompose into some basic quantum logic gates at first.

A quantum circuit is a circuit cascaded by a bunch of quantum gates. The cost of realizing a quantum circuit is typically expressed as the number of quantum gates required, also called as quantum cost [19]. It is noteworthy that these quantum gates only include the elementary gates. If there are non-elementary gates, we need to decompose these non-elementary gates firstly.

In quantum circuit, a quantum gate is usually represented by $U_n(c, t, k)$, where c is the control bit of the quantum gate, t is the target bit, and k is the position of the gate in the quantum circuit from left to right. $|c - t| - 1$ is called the nearest-neighbor cost (NNC) of a quantum gate, and the sum of the nearest-neighbors cost of gates in the circuit is called the nearest-neighbor cost (NNC) of the circuit. If the nearest-neighbor cost of a quantum circuit is 0, that is, each quantum gate in a circuit has $|c - t| = 1$. This quantum circuit is called the linear nearest-neighbor circuit.

As can be seen from the above, one of the common ways to obtain linear nearest neighbor circuit is inserting SWAP gates locally. The SWAP gate can modify the arrangement of qubits locally, and it is equivalent to three CNOT gates, so the quan-

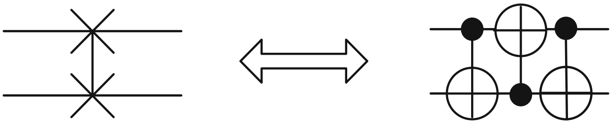


Fig. 3 SWAP gate equivalence

tum cost of SWAP gate is three, as shown in Fig. 3. Here the target bit is denoted by $\times$.

3 Proposed method

In general, if a reversible MCT circuit is converted to LNN quantum circuit, there are three stages in total: decomposing MCT gates, making circuit nearest neighbor and simplifying circuits. So, three optimization strategies of reducing quantum cost by optimizing these three aspects, respectively, have been proposed here.

3.1 MCT gates decomposition

3.1.1 Basic decomposition method

According to [19], basic decomposition methods are proposed as following: n represents the number of qubits in the circuit, and m tokens the number of control bits of a MCT gate.

When applying these methods, the first step that needs to be taken is to judge whether the circuit has free lines. The free line means there is no control bits or target bits in this line. If there is no free line, we need to add an additional free line at first. So, the first step to decompose a MCT circuit is to satisfy premise below.

Premise If $m = n - 1$, an additional auxiliary line is added to the original circuit.

For a MCT circuit, it can consist of MCT gates, NCV gates and Toffoli gates. So, decomposing a MCT circuit is equivalent to decompose a signal MCT gate and Toffoli gate. From the reference [19], Lemma 7.2 and Lemma 7.3, we can classify MCT gates into two cases: the number of control bits is over $\lceil n/2 \rceil$ and less than $\lceil n/2 \rceil$. $\lceil x \rceil$ here is a ceiling function mapping a real number to an integer, and $\lceil x \rceil$ is the least integer than or equal to x.

These two cases correspond to the following two different decomposition methods, rule 1 and rule 2.

Rule 1 If $n \geq 5, m \in \{3, 4 \ldots, \lceil n/2 \rceil\}$, m control bits MCT gate can be decomposed into a circuit cascaded by $4(m - 2)$ Toffoli gates, as shown in Fig. 4.

Rule 2 If $n \geq 5$, $m > \lceil n/2 \rceil$, m control bits MCT gate can be decomposed into a circuit consisting of $\lceil n/2 \rceil$ control bits and $n - 1 - \lceil n/2 \rceil$ control bits MCT gates, as shown in Fig. 5.

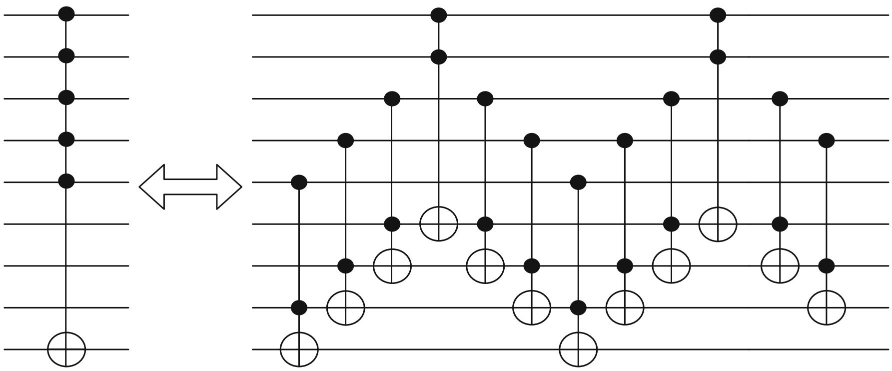


Fig. 4 Decomposition rule 1

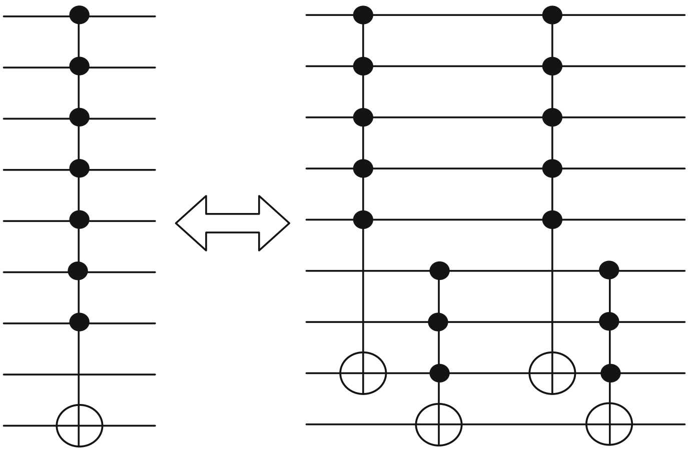


Fig. 5 Decomposition rule 2

For rule 1 and rule 2, n is all more than 5. Because the number of control bits of a MCT gate control bit is more than 3, n in the circuit is at least 4. However, for a circuit with $n = 4$, it is necessary to add an auxiliary line according to the premise, and n becomes 5. Therefore, rule 1 and rule 2 cover all the cases of decomposing MCT gates.

At the same time, if we need to get a circuit consisting of elementary gate, we also need to decompose Toffoli gate. From Lemma 6.1 [19], rule 3 can be applied to decompose the Toffoli gates.

Rule 3 Toffoli gate can be decomposed into the following circuit as shown in Fig. 6.

3.1.2 Decomposition optimization

Given a cascade of reversible gates $G_1 G_2 \ldots G_k$ realizing the reversible function F, the cascade $G_k^{-1} \ldots G_2^{-1} G_1^{-1}$ realizes the function F^{-1}, where G_i^{-1} is the inverse

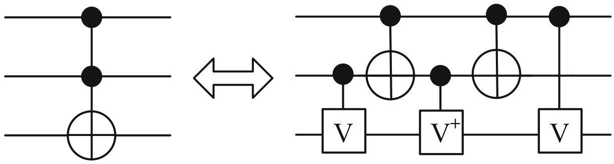


Fig. 6 Decomposition rule 3

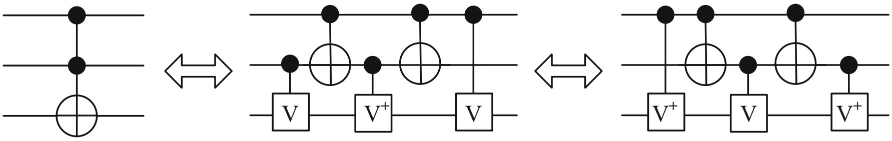
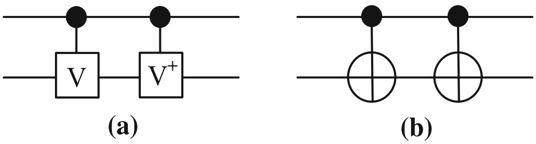


Fig. 7 Two forms of decomposing Toffoli gate

Fig. 8 Deleting rules



gate for G_i [20]. At the same time, V and V^+ gates can be exchanged with each other without affecting the function of circuits [20]. So, Toffoli gate can also be decomposed into the other form, as shown in Fig. 7.

From Figs. 4 and 5, we can obviously find that the decomposed circuit exhibits symmetry, that is, the composition of the left and right sides is same, and the difference is the order of gates. So, we can decompose two symmetry Toffoli gates with same control and target bits in two different forms.

As we all know, logic gates NCV libraries, such as NOT and CNOT, are self-inverse. And, two adjacent gates (or moved to adjacent) having the same control and target bits yield the identity mapping. So, if these two gates are deleted, the function of circuit will not be affected. We can have the following deleting rules.

Deleting rule 1 If two gates V and V^+ are adjacent (or moved adjacent), as shown in Fig. 8a, these two gates can be deleted in the circuit.

Deleting rule 2 If two NCV gates are adjacent (or moved adjacent), as shown in Fig. 8b, these two gates can be deleted in the circuit.

Therefore, applying deleting rules after decomposing can reduce the quantum cost of the circuit. The explicit applying process is following:

```
Strategy 1. Applying in the process of MCT decomposing
Input:
      MCT circuit G = G_1, G_2, G_3, ... G_S
      Index of the current gate p
      the number of qubits of circuit n
Output:
      NCV circuit G'
begin
   for p to s do
      begin
         m is the number of control bits of G_p
         if m == n - 1
         begin
            add an auxiliary line
         end
         else
            begin
               if m ∈ {3,4 ..., ⌈n/2⌉}      apply rule 1
               else if m > ⌈n/2⌉      apply rule 2
            end
            begin
               apply different forms to decompose Toffoli gate depend on symmetry
               apply deleting rules
            end
             p++
      end
end
```

Example Decompose a MCT gate, with $n = 4$, $m = 3$, as shown in Fig 9a.

First, check whether the circuit requires an auxiliary line. According to premise, n minus one is m, so an auxiliary line is added to the circuit, as shown in Fig 9b. Then, the circuit as shown in Fig. 9c can be got by using rule 1 and Fig. 10 can be realized after decomposing Toffoli gates. At the end, we apply deleting rules to get the final circuit in Fig. 11. Comparing Figs. 10, 11, quantum cost is decreased from 20 to 14.

3.2 Nearest neighbor of quantum circuit

3.2.1 Gate rearrangement

In general, the order of the gates in the circuit cannot be changed arbitrarily in that will affect the function of the circuit. However, the order of some gates can be changed when they satisfy some particular condition. So, in the processing of realizing linear

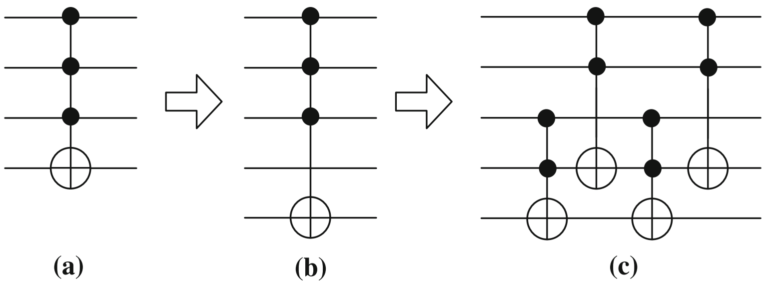


Fig. 9 Add auxiliary line, decomposition

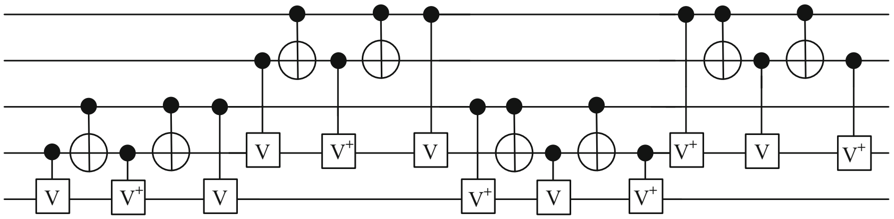


Fig. 10 The original circuit after decomposing

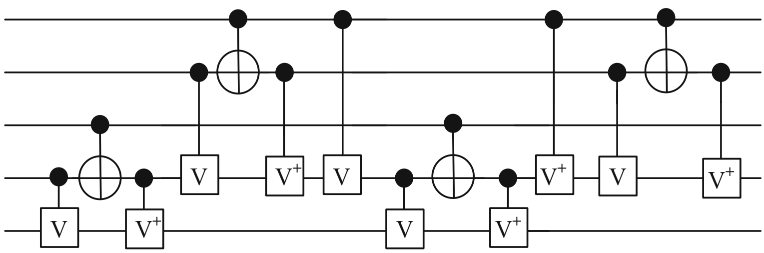


Fig. 11 Optimized circuits after decomposing

nearest-neighbor circuits, we can let some nearest-neighbor gate move ahead to reduce some quantum cost of circuits.

Exchange rule Two adjacent gates g_1 and g_2, and their control bits and target bits are c_1, c_2 and t_1, t_2, respectively. These two gates can be interchanged if $c_1 \cap t_2 = \phi$ and $c_2 \cap t_1 = \phi$ [14].

As shown in Fig. 12, according to exchange rule, gate A and B cannot be interchanged, and gate D and E can be interchanged. Besides, from Fig. 12, the first three gates and the last gate are all nearest neighbor. So, if moving gate E ahead of D before inserting SWAP gates before gate D, the quantum cost of circuit will be lower. Therefore, we can apply this optimized strategy to actual nearest-neighbor method that can further reduce quantum cost.

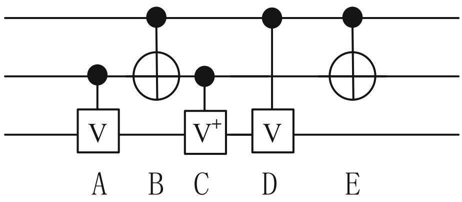


Fig. 12 Gates interchanged

3.2.2 Nearest-neighbor optimization

The optimization strategy, respectively, acts on two methods to realize the nearest-neighbor circuit in order to verify the effect of optimized strategy in the paper.

One is a heuristic method [12]. The main idea of this method is following: a non-nearest-neighbor gate g_n is found by traversing the circuit, and by scanning N gates that followed determines how many SWAP gates to insert in front of g_n and how to insert (move the target to control, move the control to target or both move target and control). The process is repeated until all gates in the circuit are all nearest neighbor.

The other is also a local look-ahead method [21]. By traversing the circuit, a non-nearest-neighbor gate g_n can be found. Then consider all possible qubits arrangements to make g_n nearest neighbor and calculate the sum of NNC of all the gates following the g_n with different qubit arrangement. From the available options, the one is chosen to insert SWAP gate which has the least negative effect to the circuit, that is the sum of NNC is minimal. At the same, the process is repeated until all gates in the circuit are all nearest neighbor.

Our optimization strategy in this stage is combining methods of realizing LNN with the above exchange rule. In other words, we judge whether the current gate and the next gate satisfy exchange rule before inserting SWAP gates in front of the current gate. If they are, and the next gate is the nearest neighbor, two gates are interchanged at first and make the current gate nearest neighbor later.

Strategy 2. Applying in the process of realizing nearest- neighbor circuit

```
Input:
    NCV circuit  G' = G'_1, G'_2, G'_3, ... G'_S
    Index of the current gate  p
Output:
    Nearest-neighbor circuit  G''
begin
  for p to s do
    n = p + 1
    if G'_P is the the nearest-neighbor gate
      begin  p++  end
    else
      begin
        if < G'_P , G'_n > satisfies exchange rule && G'_P is nearest-neighbor gate
          change the position of < G'_P , G'_n >
      end
      begin
        else  make G'_P nearest neighbor by adapting some methods
      end
   p++
end
```

3.3 Simplify LNN quantum circuits

For SWAP gates, there is also have deleting rule.

Deleting rule 3 If two SWAP gates are adjacent (or moved adjacent), as shown in Fig. 13, these two gates can be deleted in the circuit.

To achieve LNN circuits, the circuits consist of NCV gates and SWAP gates. In order to get lower quantum cost, the above deleting rules and exchange rule can be applied one more time.

The strategy 3 is the specific operation.

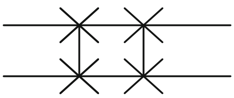


Fig. 13 Deletion rules

Strategy 3. Applying in the process of simplifying LNN circuits

Input:

LNN circuit $G'' = G''_1, G''_2, G''_3, \ldots G''_s$

Index of the current gate p

Output:

Final optimized circuit G‴

begin

for p to s do

begin

find gate that can yield the identity mapping with G''_p by applying deleting rules and exchange rule, and delete them from the circuit.

end

$p++$

end

4 Experiment

The proposed strategies are implemented in standard C++. The experimental environment is Intel (R) Core (TM) i5-4200H CPU @ 2.80 GHz, 4.00 GB memory and 64-bit Windows 8.1 operating system. All the experimental data are from RevLib, and the circuit size is up to 15 lines. Quantum cost is compared to verify whether the proposed methods are effective.

4.1 Experiment one

The experiment one is to verify the effective of decomposing optimized strategy. Assuming the circuits are all single MCT gates, the graph shown in Fig. 14 can be obtained.

In Fig. 14, abscissa m represents the number of MCT gate control bits, and ordinate qc tokens the quantum cost of quantum circuit. The curve labeled 280 represents the quantum cost curve before optimization, and the curve labeled 176 tokens the quantum cost after optimizing. From Fig. 14, with m gradually increased, the distance between two lines will gradually increases. It can be seen that with the size of MCT gate in the circuit becoming larger, the effect of the optimized decomposition method is better.

4.2 Experiment two

The second experiment verifies the effectiveness of proposed optimization strategies by, respectively, acting on two methods mentioned above of realizing LNN circuit. And the experimental results obtained are, respectively, compared with the best results of paper [12] and [21], as shown in Table 1.

In Table 1, the first three columns indicate the name of the circuit (benchmark), the initially decomposed using the decomposition method from [19] and the corresponding

Table 1 Comparison table

Benchmark	qc	n	[12]		[21]		Optimization strategies				Improvement(%)	
			$swap_1$	qc_1	$swap_2$	qc_2	$swap_3$	qc_3	$swap_4$	qc_4	[12] (%)	[21] (%)
3_17_13	14	3	5	29	6	32	4	26	3	23	10.34	28.13
4_49_17	32	4	15	77	–	–	15	75	12	68	2.60	–
4gt10-v1_81	48	5	22	114	25	123	21	99	21	99	13.16	19.51
4gt11_84	7	5	4	19	–	–	3	16	3	16	15.79	–
4gt12-v1_89	73	5	32	169	–	–	44	185	45	190	−9.47	–
4gt13-v1_93	23	5	9	50	–	–	11	50	8	41	0.00	–
4gt4-v0_80	58	5	33	157	–	–	27	121	27	125	22.93	–
4gt5_75	28	5	13	67	–	–	17	73	13	61	−8.96	–
4mod5-v1_23	24	5	13	63	–	–	18	78	9	51	−23.81	–
aj-e11_165	52	5	29	139	33	151	26	122	20	106	12.23	29.80
alu-v4_36	38	5	20	98	–	–	12	68	13	71	30.61	–
ham15_108	642	15	492	2118	531	2235	432	1738	451	1811	17.94	18.97
ham7_104	111	7	67	312	72	327	55	244	64	279	21.79	14.68
hwb4_52	23	4	9	50	–	–	14	63	10	53	−26.00	–
hwb5_55	139	5	65	334	66	337	52	265	58	283	20.66	16.02
hwb6_58	170	6	108	494	111	503	95	429	92	420	13.16	16.50
mod5adder_128	111	6	53	270	46	249	64	277	49	234	−2.59	6.02
mod8-10_177	143	5	71	356	–	–	112	443	65	302	−24.44	–
rd53_135	98	7	64	290	66	296	62	258	64	268	11.03	9.46
rd73_140	76	10	44	208	–	–	60	244	54	226	−17.31	–
rd84_142	112	15	76	340	–	–	94	380	81	341	−11.76	–
cycle10_2_110	1912	12	1128	5296	966	4810	1102	4454	764	3516	15.90	26.90

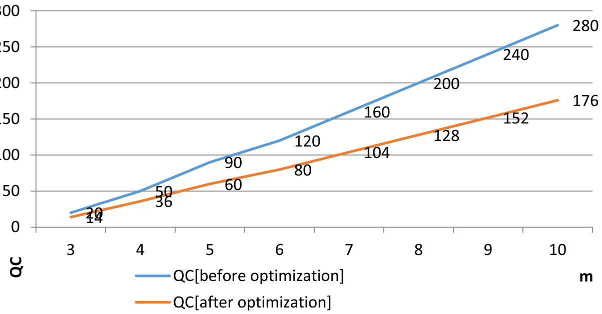


Fig. 14 Using algorithm 1 for single MCT gate

quantum cost (qc), and the size of circuit (n). The following two columns represent the best number of SWAP gates ($swap_1$) and the quantum cost of the circuits (qc_1) in [12]. The next two $swap_2$ and qc_2 are the same meaning in [21] (The benchmark with no result in [21] is represented by ——). It should be noted that the data of qc_1 or qc_2 come from the sum of qc and the numbers of $swap_1$ or $swap_2$. Specifically, qc_1 or qc_2 equals qc plus $swap_1$*3 or $swap_2$*3 (because the quantum cost of SWAP gate is three). The next four columns token the number of SWAP gates ($swap_3$, $swap_4$) and the quantum cost of the circuit (qc_3, qc_4) after applying the algorithm of optimization strategies. The last two columns show optimization rate compared with [12] and [21], respectively, that is, the reduction rate of quantum cost.

It can be seen from Table 1, after applying optimization strategies proposed in this paper, most of the results of both of them have lower quantum cost comparing with [12] and [21], and the maximum rate of optimization is 30.61%. That is, these strategies have obvious effect on reducing the quantum cost of the circuit, and can be acted on many existing methods.

5 Conclusion

In this paper, three strategies are proposed for LNN quantum circuits. In other words, the quantum circuit is optimized in three aspects: MCT gate decomposition, quantum circuit nearest-neighbor realization and quantum circuit simplification. It can be seen from the experimental results that the proposed strategies have a certain advantage in reducing the quantum cost and can be acted on many methods to optimize methods. The maximum optimization rate is up to 30.61%. What's more, we only use NCV library as descriptions in this paper, and these strategies can also fit for acting on arbitrary gate libraries not just NCV library. At the same time, MCT decomposition method has a higher effect on more control bits MCT gates. So, with the increase in the size of the circuit quantum gate, the optimization algorithm can get better results. In the future, we will consider to use parallel mechanism to let the optimization algorithm suitable for larger-scale circuits.

Acknowledgements The authors thank the financial supports from the National Nature Science Foundation of China (60873069), General Project of Natural Science Research of Colleges and Universities of Jiangsu

Province, China (14KJB520033), Natural Science Foundation of Jiangsu Province, China (BK20151274) and Postgraduate Research & Practice Innovation Program of Jiangsu Province (KYCX17_1916).

References

1. Landauer, R.: Irreversibility and heat generation in the computing process. IBM J. Res. Dev. **44**(12), 261–269 (2000)
2. Bennett, C.H.: Logical reversibility of computation. IBM J. Res. Dev. **17**(6), 525–532 (1973)
3. Golubitsky, O., Maslov, D.: A study of optimal 4-bit reversible Toffoli circuits and their synthesis. IEEE Trans. Comput. **61**(9), 1341–1353 (2012)
4. Wan, S., Chen, H., Cao, R.: A novel transformation-based algorithm for reversible logic synthesis. In: Proceedings of the 4th International Symposium on Intelligence Computation and Applications (ISICA), vol. 5821, pp. 70–81 (2009)
5. Cheng, X., Guan, Z.: Linear nearest neighbor quantum circuit synthesis based on valid Boolean matrix. Chin. J. Quantum Electron.(量子电子学报) **33**(6), 743–750 (2016) (**in Chinese**)
6. Cirac, J.I., Zoller, P.: Quantum computations with cold trapped ions. Phys. Rev. Lett. **74**(20), 4091–4094 (1995)
7. Wille, R., Quetschlich, N., Inoue, Y., Yasuda, N., Minato, S.I.: Using π DDs for Nearest Neighbor Optimization of Quantum Circuits. Reversible Computation. Springer International Publishing, New York (2016)
8. Alfailakawi, M., Alterkawi, L., Ahmad, I., et al.: Line ordering of reversible circuits for linear nearest neighbor realization. Quantum Inf. Process. **12**(10), 3319–3339 (2013)
9. Saeedi, M., Wille, R., Drechsler, R.: Synthesis of quantum circuits for linear nearest neighbor architectures. Quantum Inf. Process. **10**(3), 355–377 (2011)
10. Rahman, M.M., Dueck, G.W., Chattopadhyay, A., Wille, R.: Integrated synthesis of linear nearest neighbor Ancilla-free MCT circuits. In: IEEE, International Symposium on Multiple-Valued Logic, pp. 144–149. IEEE (2016)
11. Wille, R., Lye, A., Drechsler, R.: Exact reordering of circuit lines for nearest neighbor quantum architectures. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **33**(12), 1818–1831 (2014)
12. Kole, A., Datta, K., Sengupta, I.: A heuristic for linear nearest neighbor realization of quantum circuits by swap gate insertion using N-gate lookahead. IEEE J. Emerg. Sel. Top. Circuits Syst. **6**(1), 62–72 (2016)
13. Deb, A., Wille, R., Drechsler, R., Das, D.K.: An efficient reduction of common control lines for reversible circuit optimization. In: IEEE International Symposium on Multiple-Valued Logic, pp. 14–19. IEEE (2015)
14. Ali, M.B., Hirayama, T., Yamanaka, K., Nishitani, Y.: Quantum cost reduction of reversible circuits using new Toffoli decomposition techniques. In: International Conference on Computational Science and Computational Intelligence, pp. 59–64. IEEE (2016)
15. Miller, D.M., Sasanian, Z.: Lowering the quantum gate cost of reversible circuits. In: IEEE International Midwest Symposium on Circuits and Systems, pp. 260–263. IEEE (2010)
16. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, Cambridge (2011)
17. Hung, W.N.N., Song, X., Yang, G., et al.: Optimal synthesis of multiple output Boolean functions using a set of quantum gates by symbolic reachability analysis. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **25**(9), 1652–1663 (2006)
18. Drechsler, R., Wille, R.: From Truth Tables to Programming Languages: Progress in the Design of Reversible Circuits. **4**(10):78-85 (2011)
19. Barenco, A., et al.: Elementary gates for quantum computation. Phys. Rev. A **52**, 3457–3467 (1995)
20. Miller, D.M., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control Toffoli gates. IEEE International Symposium on Multiple-Valued Logic, vol. 47, pp. 288–293. IEEE (2011)
21. Wille, R., Keszocze, O., Walter M., et al.: Look-ahead schemes for nearest neighbor optimization of 1D and 2D quantum circuits. In: Asia and South Pacific Design Automation Conference. IEEE, vol. 2001, pp. 292–297