

Considering nearest neighbor constraints of quantum circuits at the reversible circuit level

Robert Wille · Aaron Lye · Rolf Drechsler

Received: 4 July 2013 / Accepted: 5 September 2013 / Published online: 24 September 2013
© Springer Science+Business Media New York 2013

Abstract Since many underlying quantum algorithms include a Boolean component, synthesis of the respective circuits is often conducted by a two-stage procedure: First, a reversible circuit realizing the Boolean component is generated. Afterwards, this circuit is mapped into a respective quantum gate cascade. In addition, recent physical accomplishments have led to further issues to be considered, e.g. nearest neighbor constraints. However, due to the lack of proper metrics, these constraints usually have been addressed at the quantum circuit level only. In this paper, we present an approach that allows the consideration of nearest neighbor constraints already at the reversible circuit level. For this purpose, a recently introduced gate library is assumed for which a proper metric is proposed. By means of an optimization approach, the applicability of the proposed scheme is illustrated.

Keywords Synthesis · Reversible circuits · Quantum circuits · Nearest neighbor

1 Introduction

Quantum circuits [25] represent a promising alternative to conventional circuit technologies allowing for solving many important problems (e.g. database search, factorization, graph problems) significantly faster (see e.g. [7, 9, 33]). Information is thereby stored in terms of qubits. In contrast to conventional bits, qubits do not only allow to represent Boolean 0's and Boolean 1's, but also the superposition of both. The

R. Wille (✉) · R. Drechsler
University of Bremen and DFKI Bremen, 28359 Bremen, Germany
e-mail: rwille@informatik.uni-bremen.de

A. Lye
University of Bremen, 28359 Bremen, Germany

states of the respective qubits are thereby modified by quantum operations which can be represented by unitary matrices. That is, each quantum computation is inherently reversible, but manipulates qubits rather than pure logic values. Since most of the solutions include a significant Boolean component to be realized (e.g. the Oracle function in the Deutsch Algorithm or Grover's database search as well as the modular exponentiation in Shor's factorization algorithm), synthesis of quantum circuits is often conducted by a two-stage procedure: First, a reversible circuit is designed using established reversible gate libraries (containing e.g. Toffoli gates [36]). To this end, several synthesis approaches have been introduced in the past (see e.g. [20,28,32,35,37,39]). Then, the resulting (reversible) circuits are mapped into the respective quantum circuits. Here schemes as originally introduced by Barenco [2] or its recently optimized versions (see e.g. [19,21]) are applied. By this, the Boolean parts of a quantum circuit to be realized are synthesized. This is reviewed in detail later in Sect. 3.

However, recent physical accomplishments have led to further constraints to be considered during quantum circuit synthesis. In particular, so called *nearest neighbor constraints* have been introduced. They are justified by the fact that many physical realizations of quantum circuits assume that computations are only performed between adjacent (i.e. nearest neighbor) signals. This was originally motivated by physical realizations based e.g. on trapped ions (see e.g. [10]), liquid nuclear magnetic resonance (see e.g. [18]), and architectures based on the original Kane model [15]. Nowadays, ion traps are no longer appropriately described as universally nearest neighbor architectures, liquid nuclear magnetic resonance is acknowledged as not scalable, and the original Kane proposal has been superseded by [13]. Nevertheless, nearest neighbor architectures are still an issue in recently proposed technologies including proposals for ion traps [1,17,24], nitrogen-vacancy centers in diamonds [5,40], quantum dots emitting linear cluster states linked by linear optics [11], laser manipulated quantum dots in a cavity [14], and superconducting qubits [6,26].

Motivated by that, synthesis methods have been introduced which realize circuits satisfying this condition (see e.g. [4,12,16,22,29]). A major problem is thereby that methods for nearest neighbor optimization usually work at the quantum circuit level only. Indeed, some previous work (e.g. [3]) considered nearest neighbor constraints already at the reversible circuit level. But due to the lack of proper metrics, only adjacency of Toffoli gates has been achieved so far, while the mapping to its corresponding quantum circuit realization may introduce further non-adjacent gates. In fact, nearest neighbor conditions cannot efficiently be addressed at the reversible circuit level yet. This is discussed in detail later in Sect. 4.

In this paper, we present an approach that allows the consideration of nearest neighbor constraints already at the reversible circuit level. For this purpose, a quantum gate library is applied which theoretically and conceptionally has been discussed in [23,31]. We provide a precise metric of the nearest neighbor constraints that even can be used at the reversible circuit level. We illustrate how this metric can be applied to optimize reversible circuits with respect to these constraints.

The remainder of this paper is structured as follows. First, the required background on reversible as well as quantum circuits is provided and the mapping from reversible circuits to quantum circuits is reviewed in Sects. 2 and 3, respectively. Afterwards, how to consider nearest neighbor constraints is discussed in Sect. 4. This includes the

introduction of metrics that even can be applied at the reversible circuit level. The application of this is then illustrated in Sect. 5 and evaluated in Sect. 6. Finally, the paper is concluded in Sect. 7.

2 Preliminaries

In order to keep the remainder of this paper self-contained, preliminaries on reversible and quantum circuits are briefly reviewed in this section.

2.1 Reversible gates & circuits

A logic function $f : \mathbb{B}^n \rightarrow \mathbb{B}^m$ over inputs $X = \{x_1, \dots, x_n\}$ is *reversible* if and only if

- its number of inputs is equal to its number of outputs (i.e. $n = m$) and
- it maps each input pattern to a unique output pattern.

Otherwise, the function is termed *irreversible*. In other words, a reversible function represents a bijection.

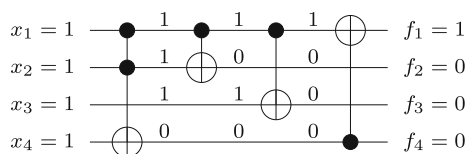
A reversible function can be realized by a circuit $G = g_1 g_2 \dots g_d$ comprised of a cascade of reversible gates g_i , where d is the number of gates. Fanouts and feedback are not directly allowed [25]. Several different reversible gates have been introduced including the Toffoli gate [36], the Fredkin gate [8], and the Peres gate [27]. In the following, we focus on Toffoli gates which are universal gates, i.e. all reversible functions can be realized by means of this gate type alone [36].

A *multiple control Toffoli gate* has a *target line* x_j and *control lines* $\{x_{i_1}, x_{i_2}, \dots, x_{i_k}\}$. This gate maps $(x_1, x_2, \dots, x_j, \dots, x_n)$ to $(x_1, x_2, \dots, x_{i_1} x_{i_2} \dots x_{i_k} \oplus x_j, \dots, x_n)$. That is, the target line is inverted if all control lines are set to 1; otherwise the value of the target line is passed through unchanged. A Toffoli gate with no control lines always inverts the target line and is a *NOT gate*. A Toffoli gate with a single control line is called a *controlled-NOT gate* (also known as the CNOT gate). The case of two control lines is the original gate defined by Toffoli. For brevity, we refer to a multiple-control Toffoli gate as a Toffoli gate.

In the following, a Toffoli gate is denoted by the tuple $T(C, t)$ where $C \subset X$ is the possibly empty set of control lines and $t \in X \setminus C$ is the target line. Note that the control lines and unconnected lines pass through a gate unchanged. For drawing circuits, we follow the established convention of using the symbol \oplus to denote the target line and solid black circles to indicate control connections for the gate.

Example 1 Figure 1 shows a reversible circuit composed of $n = 4$ circuit lines and $d = 4$ Toffoli gates. This circuit maps e.g. the input pattern 1111 to the output pattern 1000

Fig. 1 Reversible circuit



(as shown in Fig. 1). Inherently, every computation can be performed in both directions (i.e. computations towards the outputs *and* towards the inputs can be performed).

2.2 Quantum gates & circuits

The basic building block for a quantum computer is the qubit. A qubit is a two level quantum system, described by a two dimensional complex Hilbert space. The two orthogonal quantum states $|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ are used to represent the values 0 and 1. Any state of a qubit may be written as $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β are complex numbers with the following condition $|\alpha|^2 + |\beta|^2 = 1$. The quantum state of a single qubit is denoted by the vector $\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. The state of a quantum system with $n > 1$ qubits is given by an element of the tensor product of the single state spaces and can be represented as a normalized vector of length 2^n , called the state vector. The state vector is changed through multiplication of appropriate $2^n \times 2^n$ unitary matrices [25].

Since this allows an infinite number of qubit-values and corresponding operations, researchers defined proper models and gate libraries in order to realize Boolean functions in quantum logic. In this work, the following two libraries are considered.

2.2.1 NCV library

The quantum gate library introduced by Barenco et al. [2] is mostly applied in the development of design methods for quantum circuits. Here, the following set of quantum gates is considered:

- NOT gate $T(\emptyset, t)$: A single qubit t is inverted.
- Controlled NOT (CNOT) gate $T(\{c\}, t)$: The target qubit t is inverted if the control qubit c is 1.
- Controlled V gate $V(\{c\}, t)$: A V operation is performed on the target qubit t if the control qubit c is 1. The V operation is also known as the square root of NOT, since two consecutive V operations are equivalent to an inversion.
- Controlled V^\dagger gate $V^\dagger(\{c\}, t)$: A V^\dagger operation is performed on the target qubit t if the control qubit c is 1. The V^\dagger gate performs the inverse operation of the V gate, i.e. $V^\dagger = V^{-1}$.

More precisely, these gates transform the target qubit t as specified by the unitary matrices

$$NOT = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad V = \frac{1+i}{2} \begin{pmatrix} 1 & -i \\ -i & 1 \end{pmatrix}, \quad \text{and} \quad V^\dagger = \frac{1-i}{2} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}.$$

If the input signals and all control lines are restricted to Boolean values, a 4-valued logic results where each qubit may represent one value of $\{0, 1, v_0, v_1\}$ with $v_0 = \frac{1+i}{2} \begin{pmatrix} 1 \\ -i \end{pmatrix}$ and $v_1 = \frac{1+i}{2} \begin{pmatrix} -i \\ 1 \end{pmatrix}$. Figure 2 shows the resulting transitions with respect to the possible NOT, V, and V^\dagger operations. This is sufficient to realize every reversible function as a quantum circuit [2]. Furthermore, this keeps the gate library simple enough to become physically realizable. In the following, this model is called *NCV gate library*.

Fig. 2 State transitions for NOT, CNOT, V, and V^\dagger operations

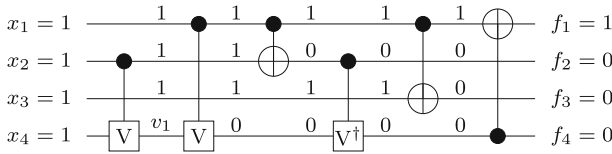
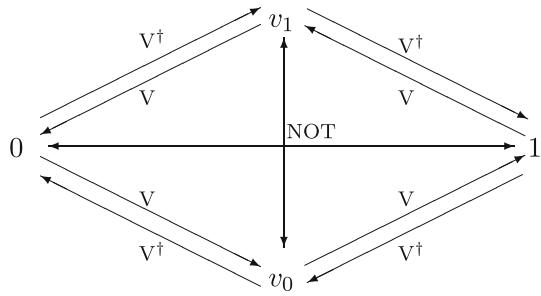


Fig. 3 Quantum circuit using the NCV gate library

Example 2 Figure 3 shows a quantum circuit composed of $n = 4$ circuit lines and $d = 6$ quantum gates. This circuit again maps e.g. the input pattern 1111 to the output pattern 1000, but in contrast to the circuit from Fig. 1 quantum values and quantum operations are utilized for this purpose.

2.2.2 NCV- $|v_1\rangle$ library

Although the NCV library is universal, i.e. every Boolean function can be represented by it [2], extensions of it have been introduced recently (see e.g. [30,31]). In this work, we additionally consider the quantum gate library introduced in [31] which is based on the theoretical discussion on physical realizations from [23]. Here, *qudits* instead of qubits are assumed, i.e. a basic building block which does not rely on a two level quantum system, but a (multiple-valued) d -level quantum system is assumed. Any state of a qudit may be written as $|\Psi\rangle = c_0|0\rangle + c_1|1\rangle + \dots + c_{d-1}|d-1\rangle$ where c_i for all $i = 0, \dots, d-1$ are complex numbers such that $|c_0|^2 + |c_1|^2 + \dots + |c_{d-1}|^2 = 1$. Similar to qubits, the states of a qudit are represented by a state vector. The state vector is changed through multiplication of appropriate unitary matrices. In the case of an uncontrolled transformation the dimension of the unitary matrices is $d \times d$, in the case of an controlled transformation the dimension is $d^2 \times d^2$. However, in contrast to a qubit, the controlled gates for qudits perform the respective operation not when the control line is $|1\rangle$, but rather when the control line is set to the value $|d-1\rangle$.

In [23], a corresponding gate library for qudits has been presented and discussed. In [31], this model is adopted with a 4-level logic, i.e. $d = 4$. The basic states are, in that order, 0, v_0 , 1, and v_1 . As already explained, the controlled gates are only transforming the target qudit if the value of the control line is set to $|d-1\rangle \equiv v_1$. We emphasize this fact by labeling the control connections for the respective gates with v_1 .

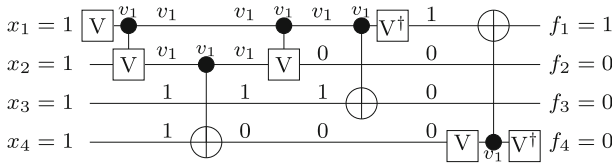


Fig. 4 Quantum circuit NCV- $|v_1\rangle$ gate library

The library is composed of the three unitary gates (i.e. gates without a control line) performing the NOT, V, and V^\dagger operation as well as single-control versions of these gates. More precisely, these gates transform the target qubit t as specified by the unitary matrices

$$NOT = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, V = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, V^\dagger = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{pmatrix}.$$

Again, restricting the inputs to Boolean values still allows for the realization of any arbitrary reversible functions. In the following, this model is called NCV- $|v_1\rangle$ gate library.

Example 3 Figure 4 shows a quantum circuit composed of $n = 4$ circuit lines and $d = 5$ quantum gates. This circuit performs the same computation as the circuits from Figs. 1 and 3, but is composed of gates from the NCV- $|v_1\rangle$ library introduced in [31].

3 Mapping reversible circuits to quantum circuits

Since any quantum operation can be represented by a unitary matrix [25], each quantum circuit is inherently reversible. Consequently, every reversible circuit can be transformed to a quantum circuit. Motivated by this, synthesis of Boolean components of quantum circuits is usually conducted in two steps: First, the desired logic is synthesized as reversible circuit. Afterwards, each gate of the resulting circuit is mapped to a corresponding cascade of quantum gates. To this end and depending on the addressed gate library, different mapping schemes have been proposed.

3.1 Mapping to gates from the NCV library

Mapping of reversible gates to NCV gates has intensely been considered in the past. Originally, Barenco et al. proposed the initial mappings in [2]. Afterwards, further improvements have been introduced e.g. in [19] and, more recently, in [21].

Example 4 Consider a Toffoli gate with two control lines as shown in the left-hand side of Fig. 5a. A functionally equivalent realization in terms of gates from the NCV library is depicted in the right-hand side of Fig. 5a.

Similar mappings exist for Toffoli gates with more than two control lines. But with increasing number of control lines, the resulting quantum circuits become more expensive, i.e. require more quantum gates. Furthermore, also the number of the *ancillary*

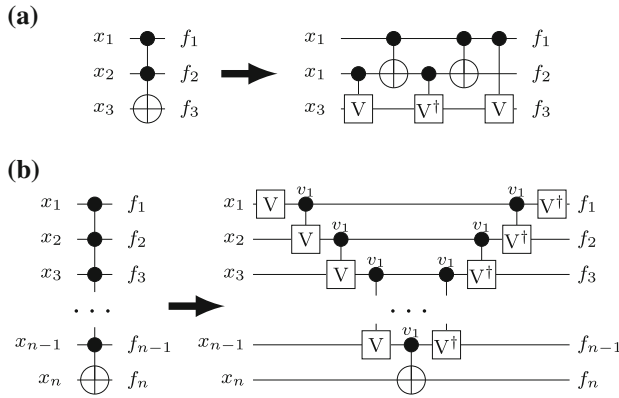


Fig. 5 Mapping reversible circuits to quantum circuits. **a** Mapping to gates from the NCV library. **b** Mapping to gates from the NCV- $|v_1\rangle$ library

lines, i.e. the number of circuit lines which neither are a control line nor a target line, affect the size of the resulting quantum circuit. To provide some numbers, Table 1a lists the respective number of quantum gates for different Toffoli gate configurations according to the current state-of-the-art NCV mapping scheme introduced in [21].

3.2 Mapping to gates from the NCV- $|v_1\rangle$ library

With the NCV- $|v_1\rangle$ library, also a corresponding mapping scheme has been introduced in [31]. This scheme fully exploits the $|v_1\rangle$ -sensitivity of the control lines which enables a more efficient mapping of reversible gates than the mapping to gates from the NCV library. The general principle of this mapping is illustrated by means of the following example.

Example 5 Consider a Toffoli gate with an arbitrary number of control lines as shown in the left-hand side of Fig. 5b. A functionally equivalent realization in terms of gates from the NCV- $|v_1\rangle$ library is depicted in the right-hand side of Fig. 5b. First, all control lines are $|v_1\rangle$ -sensitized, i.e. (v_1 -controlled) V -gates are applied setting the values of the control lines to v_1 iff they all have initially been set to 1. By this, a v_1 -controlled NOT gate ensures that the value of the target line is only flipped iff all control lines have been set to 1. Afterwards, (v_1 -controlled) V^\dagger -gates are applied to de-sensitize the control lines.

Using this scheme, every Toffoli gate $T(C, x_j)$ with $x_j \in X$ and $C \subset X \setminus \{x_j\}$ can be mapped to an equivalent cascade of $2 \cdot |C| + 1$ NCV- $|v_1\rangle$ quantum gates [31]. In comparison to the mapping to gates from the NCV library, this is (1) significantly more compact and (2) does not even require ancillary lines. Table 1 provides a more precise comparison between these two mappings. Note that the physical costs of the respective gates may differ in the respective libraries. However, comparing the number of elementary gates seems to be an acceptable abstraction until the physical realizations eventually advanced.

Table 1 Number of quantum gates

(a) Using NCV library							(b) Using NCV- $ v_1\rangle$ library		
Number of control lines	Number of ancillary lines						Number of control lines	Gates	Δ (%)
	1	2	3	4	5	6			
1	1						1	1	0
2	5						2	3	40
3	14						3	5	64
4	20						4	7	65
5	32						5	9	72
6	44						6	11	75
7	64	56					7	13	77–80
8	76	68					8	15	78–80
9	96	88	80				9	17	79–82
10	108	100	92				10	19	79–82
11	132	120	112	104			11	21	80–84
12	156	132	124	116			12	23	80–85
13	180	156	148	136	128		13	25	80–86
14	204	180	172	148	140		14	27	81–87
15	228	204	198	172	160	152	15	29	81–87

4 Consideration of nearest neighbor constraints

As discussed in [31] and briefly reviewed in the previous section, the NCV- $|v_1\rangle$ library enables a much more compact mapping of reversible circuits to quantum circuits with respect of gates. Beyond that, the NCV- $|v_1\rangle$ mapping additionally allows to consider nearest neighbor constraints of quantum circuits already at the reversible circuit level.

Nearest neighbor constraints have been introduced as many physical realizations of quantum circuits assume that computations are only performed between adjacent, i.e. nearest neighbor, signals (see Sect. 1). Accordingly, synthesis methods have been introduced which realize circuits satisfying this condition (see e.g. [4, 12, 16, 22, 29]).

However, as shown in the first part of this section, these methods usually work on the quantum circuit level. In the second part of this section, we illustrate how this can be lifted to the reversible circuit level.

4.1 In the NCV library

The quantum circuits resulting by using the NCV library and the state-of-the-art mappings reviewed in Sect. 3.1, often do not satisfy the nearest neighbor constraint. For example, already the mapping illustrated in Fig. 5a leads to a non-adjacent gate.

A naive approach to address that, is to add *SWAP* gates to the circuit. *SWAP* gates $S(x_i, x_j)$ have two target lines $x_i, x_j \in X$ and map $(x_1 \dots x_i \dots x_j \dots x_n)$ to

$(x_1 \dots x_j \dots x_i \dots x_n)$, i.e. the values of these target lines are interchanged. In order to satisfy the nearest neighbor condition of quantum circuits, SWAP gates can be added in front of each gate g with non-adjacent control and target lines to “move” the control (target) line of g towards the target (control) line until they become adjacent. Afterwards, SWAP gates are added to restore the original order of circuit lines. Since SWAP gates can easily be represented by a cascade of three controlled NOT-gates, the resulting quantum circuit is still composed of gates from the supported library. But obviously this procedure increases the number of gates of the circuit.

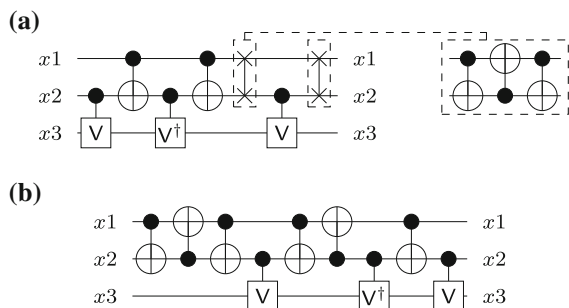
Example 6 Consider again the Toffoli gate with two control lines and its corresponding NCV mapping as shown in Fig. 5a. As can be seen, the last gate is non-adjacent. Thus, to satisfy the nearest neighbor constraint, SWAP gates in front and after the last gate are inserted as shown in Fig. 6a. Since each SWAP gate requires three controlled NOT gates, this increases the total number of NCV-gates from 5 to 11. Even a minimal nearest neighbor-aware mapping of this Toffoli gate (as shown in Fig. 6b; taken from [29]) requires a total of 9 NCV-gates.

Already this simple example illustrates the effects a consideration of nearest neighbor constraints has on the size of the resulting circuits. Moreover, any further configuration of Toffoli gates (e.g. a different position of the target line or a larger number of control lines) leads to other non-adjacent gates for which other SWAP gates have to be added. In fact, no metric exists covering all the possible mappings of arbitrary Toffoli gate configurations to quantum circuits satisfying the nearest neighbor conditions. As a consequence, nearest neighbor conditions cannot efficiently be addressed at the reversible circuit level. Instead, the established mapping from a reversible circuit to a quantum circuit is conducted first and adjacency of the resulting quantum gates is ensured by a post-synthesis process afterwards. To this end, methods introduced e.g. in [4, 12, 16, 22, 29] are applied.

4.2 In the NCV- $|v_1\rangle$ library

Using the NCV- $|v_1\rangle$ library reviewed in Sect. 3.2, nearest neighbor constraints can be satisfied much easier. In fact, the mapping shown in Fig. 5b already satisfies it since only adjacent gates are included here. However, this does not hold for all configurations: Applying the same scheme for a Toffoli gate as shown in the left-hand side of

Fig. 6 Consideration of nearest neighbor constraints in the NCV library. **a** Naive mapping. **b** Minimal mapping



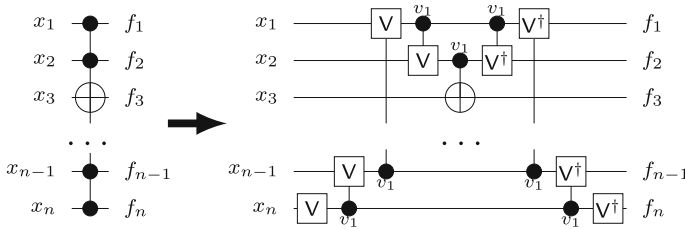


Fig. 7 Nearest neighbor-aware mapping of reversible gates

Fig. 7, a quantum circuit as shown in the right-hand side of Fig. 7 results still including two non-adjacent gates. In contrast to the NCV library, the number of SWAP gates to be applied is linear and can always be derived from the configuration of the considered reversible gate.

More precisely, let $T(C, x_j)$ be a Toffoli gate with $x_j \in X$ and $C \subset X/\{x_j\}$. Assume that all control and target lines in $T(C, x_j)$ are already adjacent. Furthermore, w.l.o.g. assume that $x_1, x_2, \dots \in X$ are the top lines and $\dots, x_{n-1}, x_n \in X$ are the bottom lines of the circuit, respectively. Then, this Toffoli gate can be mapped to a cascade of adjacent quantum gates with respect to the following two cases:

1. The target line x_j is “either on the top or the bottom”, i.e. either $\forall x_i \in C : i < j$ or $\forall x_i \in C : i > j$ holds. In this case, the mapping already introduced in Fig. 5b can be used. This already satisfies the nearest neighbor condition, i.e. no additional SWAPs are needed and a total of $2 \cdot |C| + 1$ gates are required.
2. The target line x_j is “between the control lines”, i.e. $\exists x_i \in C : i < j$ and $\exists x_k \in C : k > j$ holds. In this case, the Toffoli gate can be represented by a quantum gate cascade as shown in Fig. 7. This requires $\min(2 \cdot (j - 1 + 1), 2(k - j + 1))$ SWAP gates in order to satisfy the nearest neighbor condition (assuming that both, x_i and x_k are the control lines at the top and the bottom, respectively, i.e. $\nexists x_l \in C : l < i$ and $\nexists x_m \in C : m > k$). That is, a total of $(2 \cdot |C| + 1) + 3 \cdot \min(2 \cdot (j - 1 + 1), 2(k - j + 1))$ gates are required.

Hence, if a Toffoli gate $T(C, x_j)$ is adjacent, a linear mapping to a quantum circuit satisfying nearest neighbor constraints can be derived. In contrast to the mapping for the NCV library, this provides a precise metric for the nearest neighbor constraints that can even be used at the reversible circuit level. By this, existing synthesis procedures for reversible circuits can be extended in order to directly support nearest neighbor architectures right from the beginning of the design process.

5 Nearest neighbor-aware optimization of reversible circuits

Using the new metric provided above, this section introduces an approach that exemplarily illustrates the consideration of nearest neighbor constraints at the reversible circuit level. Therefore, an optimization method is proposed that is inspired by the work from [29]. Here a re-ordering scheme was applied to quantum circuits in order to reduce the number of non-adjacent quantum gates. The application of this scheme to the reversible circuit level is illustrated by means of an example.

Example 7 Consider the reversible circuit shown in Fig. 8a. Using the metric from Sect. 4.2, it can be calculated that a quantum circuit satisfying the nearest neighbor constraints would be composed of 50 gates, i.e. $5 + 5 + 5 + 5 = 20$ gates resulting from the mapping plus $3 \cdot 5 = 15$ gates to swap the control and target lines of the first, third, and fourth reversible gate plus $3 \cdot 5 = 15$ gates to undo this swapping. However, it can be calculated that, by slightly reordering the lines as shown in Fig. 8b, the total number of quantum gates necessary can be reduced to 32. Due to the absence of a precise metric, previous methods were not able to precisely determine such optimizations at the reversible circuit level.

Motivated by this, a reversible circuit can be optimized with respect to nearest neighbor constraints as follows: First, the “contribution” of each line to the total number of needed SWAPs is calculated. More precisely, for each gate g with control lines C and target line x_j , the number of SWAPs needed to make C and x_j adjacent is determined. This number is added to variables $prop_i$ with $i \in C \cup \{x_j\}$ which are used to save the proportion of a circuit line i on the total number of SWAPs. Next, the line i with the highest value of $prop_i$ is chosen for reordering and placed at the middle of the circuit (i.e. line i is swapped with the middle line). If the selected line is the middle line itself, a line with the next larger value is selected. This procedure is repeated until no improvements are achieved.

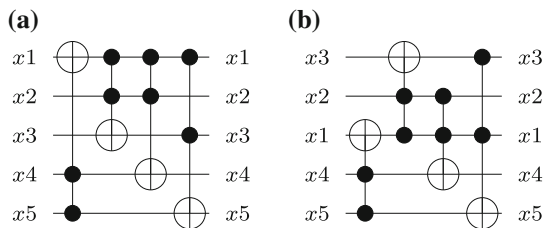
Example 8 Applying the proposed scheme to the circuit shown in Fig. 8, values $prop_{x_1} = 5$, $prop_{x_2} = 1$, $prop_{x_3} = 2$, $prop_{x_4} = 3$, and $prop_{x_5} = 4$ result. Thus, the lines x_1 (largest proportion) and x_3 (middle line) are swapped. Since further swaps do not lead to further improvements, the reordering terminates leading to the circuit shown in Fig. 8b.

Note that this only exemplarily illustrates the consideration of nearest neighbor constraints at the reversible circuit level. In fact, thanks to the metric from Sect. 4.2 other existing synthesis and optimization approaches for reversible circuits can be adjusted accordingly. However, the experimental evaluation summarized in next section confirms that already the this rather simple re-ordering scheme leads to significant improvements.

6 Experimental evaluation

The optimization approach presented in the previous section has been implemented in C++ on top of *RevKit* [34] and applied to *all* benchmark circuits available at *RevLib* [38]. Thanks to the rather heuristic nature of the approach, all circuits have

Fig. 8 Nearest neighbor-aware optimization of a reversible circuits. **a** Original circuit. **b** Optimized circuit



been processed with negligible run-time (i.e. within a couple of minutes) on an Intel Core i5-3210M machine with 2.5 GHz and 4 GB of memory.

The evaluation showed that, over all considered circuits, a reduction in the number of gates of the resulting quantum gate cascades of approx. 8% can be achieved on average. This includes many small circuits for which no improvement at all has been obtained. In contrast, for larger circuits improvements of up to 68% are possible.

Table 2 shows the best improvements which have been observed during our evaluation. The first columns give thereby the name, the number n of lines, and the number g

Table 2 Experimental evaluation

CIRCUIT	n	g	NO. OF QUANTUM GATES		IMPR. (%)
			W/O OPT.	W/ OPT.	
plus127mod8192_308.real	25	54	1936	610	68
plus63mod8192_310.real	25	53	1864	676	64
rd53_311.real	13	34	872	344	61
e64-bdd_295.real	195	387	149492	68174	54
ham15_298.real	45	153	11706	5820	50
mini_alu_305.real	10	20	467	245	48
4mod5-v0_20.real	5	5	26	14	46
hwb7_302.real	73	281	46853	25487	46
ham7_299.real	21	61	2545	1429	44
4mod5-v1_23.real	5	8	98	56	43
con1_216.real	9	21	359	209	42
rd73_312.real	25	73	3648	2148	41
cycle10_293.real	39	78	5435	3275	40
plus63mod4096_309.real	23	49	1592	974	39
one-two-three-v2_100.real	5	8	62	38	39
rd84_313.real	34	104	6589	4057	38
sym9_317.real	27	62	3564	2214	38
hwb6_301.real	46	159	16194	10254	37
hwb5_300.real	28	88	5668	3592	37
mod5adder_306.real	32	96	7532	4796	36
4mod5-v0_21.real	5	6	17	11	35
rd53_136.real	7	15	256	166	35
bw_291.real	87	307	57753	37665	35
urf4_187.real	11	32004	831648	542844	35
4gt12-v1_89.real	5	5	52	34	35
4mod5-v1_24.real	5	5	52	34	35
mod10_171.real	4	10	87	57	34
4gt11_84.real	5	3	35	23	34
rd32_270.real	5	9	70	46	34
rd32_271.real	5	9	70	46	34
hwb8_303.real	112	449	113585	74951	34

of gates of the respective reversible circuits. Afterwards, the number of quantum gates are reported for the original circuits (w/o optimization) and for the improved circuits (w/ optimization). In both cases, of course the metric from Sect. 4.2 has been applied, i.e. the nearest neighbor condition was considered. The total improvement is provided by the last column.

7 Conclusions

In this paper, we presented an approach that allows for the consideration of nearest neighbor constraints already at the reversible circuit level. By this, a gap in today's design flows for corresponding quantum circuits has been closed. So far, only adjacency of Toffoli gates has been achieved at the reversible circuit level, while the mapping to its corresponding quantum cascades may have introduced further non-adjacent gates. Using the gate library proposed in [23, 31], we were able to provide a metric where such cases are covered. By this, nearest neighbor constraints can directly be addressed in the reversible circuit. Exemplarily, this has been illustrated by means of an optimization approach.

Future work consists of two major pillars: First, this work as well as the previous contributions in [23, 31] just provided a theoretical and conceptual discussion on the applicability of the NCV- $|v_1\rangle$ library. Hence, physical realizations and concepts on how to realize this library is subject to future work. This includes (1) direct realizations of the qudits, (2) the emulation of qudits e.g. by existing qubit-realizations, and (3) the compatibility to existing fault-tolerant quantum error correction protocols. Second, the optimization approach presented here was applied to illustrate the applicability of the proposed metric and, hence, is rather simple. More detailed investigations on synthesis and optimization approaches for reversible circuits exploiting these concepts are left for future work, too.

Acknowledgments The authors sincerely thank the reviewers for their thorough consideration of the manuscript and many comments that helped to improve this work.

References

1. Amini, J.M., Uys, H., Wesenberg, J.H., Seidelin, S., Britton, J., Bollinger, J.J., Leibfried, D., Ospelkaus, C., VanDevender, A.P., Wineland, D.J.: Toward scalable ion traps for quantum information processing. *New J. Phys.* **12**(3), 033,031 (2010) <http://stacks.iop.org/1367-2630/12/i=3/a=033031>
2. Barenco, A., Bennett, C.H., Cleve, R., DiVincenzo, D., Margolus, N., Shor, P., Sleator, T., Smolin, J., Weinfurter, H.: Elementary gates for quantum computation. *Am. Phys. Soc.* **52**, 3457–3467 (1995)
3. Chakrabarti, A., Sur-Kolay, S., Chaudhury, A.: Linear nearest neighbor synthesis of reversible circuits by graph partitioning. *CoRR* (2011)
4. Chakrabarti, A., Sur-Kolay, S.: Nearest neighbour based synthesis of quantum boolean circuits. *Eng. Lett.* **15**, 356–361 (2007)
5. Devitt, S.J., Fowler, A.G., Stephens, A.M., Greentree, A.D., Hollenberg, L.C.L., Munro, W.J., Nemoto, K.: Architectural design for a topological cluster state quantum computer. *New J. Phys.* **11**(8), 083,032 (2009) <http://stacks.iop.org/1367-2630/11/i=8/a=083032>
6. DiVincenzo, D.P., Solgun, F.: Multi-qubit parity measurement in circuit quantum electrodynamics. *New J. Phys.* **15**(7), 075,001 (2013). <http://stacks.iop.org/1367-2630/15/i=7/a=075001>
7. Dürr, C., Heiligman, M., Hoyer, P., Mhalla, M.: Quantum query complexity of some graph problems. *SIAM J. Comput.* **35**, 1310–1328 (2006)

8. Fredkin, E.F., Toffoli, T.: Conservative logic. *Int. J. Theor. Phys.* **21**(3/4), 219–253 (1982)
9. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: *Theory of Computing*, pp. 212–219 (1996)
10. Häffner, H., Hänsel, W., Roos, C.F., Benhelm, J., al kar, D.C., Chwalla, M., Körber, T., Rapol, U.D., Riebe, M., Schmidt, P.O., Becher, C., Gühne, O., Dür, W., Blatt, R.: Scalable multiparticle entanglement of trapped ions. *Nature* **438**, 643–646 (2005)
11. Herrera-Marti, D.A., Fowler, A.G., Jennings, D., Rudolph, T.: Photonic implementation for the topological cluster-state quantum computer. *Phys. Rev. A* **82**, 032,332 (2010). doi:[10.1103/PhysRevA.82.032332](https://doi.org/10.1103/PhysRevA.82.032332)
12. Hirata, Y., Nakanishi, M., Yamashita, S., Nakashima, Y.: An efficient method to convert arbitrary quantum circuits to ones on a linear nearest neighbor architecture. In: *International Conference on Quantum, Nano and Micro Technologies*, pp. 26–33. IEEE Computer Society, Washington, DC, USA (2009). doi:[10.1109/ICQNM.2009.25](https://doi.org/10.1109/ICQNM.2009.25)
13. Hollenberg, L.C.L., Greentree, A.D., Fowler, A.G., Wellard, C.J.: Two-dimensional architectures for donor-based quantum computing. *Phys. Rev. B* **74**, 045,311 (2006). doi:[10.1103/PhysRevB.74.045311](https://doi.org/10.1103/PhysRevB.74.045311)
14. Jones, N.C., Van Meter, R., Fowler, A.G., McMahon, P.L., Kim, J., Ladd, T.D., Yamamoto, Y.: Layered architecture for quantum computing. *Phys. Rev. X* **2**, 031,007 (2012). doi:[10.1103/PhysRevX.2.031007](https://doi.org/10.1103/PhysRevX.2.031007)
15. Kane, B.: A silicon-based nuclear spin quantum computer. *Nature* **393**, 133–137 (1998)
16. Khan, M.H.A.: Cost reduction in nearest neighbour based synthesis of quantum boolean circuits. *Eng. Lett.* **16**, 1–5 (2008)
17. Kumph, M., Brownutt, M., Blatt, R.: Two-dimensional arrays of radio-frequency ion traps with addressable interactions. *New J. Phys.* **13**(7), 073,043 (2011). <http://stacks.iop.org/1367-2630/13/i=7/a=073043>
18. Laforest, M., Simon, D., Boileau, J.C., Baugh, J., Ditty, M., Laflamme, R.: Using error correction to determine the noise model. *Phys. Rev. A* **75**, 133–137 (2007)
19. Maslov, D., Young, C., Dueck, G.W., Miller, D.M.: Quantum circuit simplification using templates. In: *Design, Automation and Test in Europe*, pp. 1208–1213 (2005)
20. Miller, D.M., Maslov, D., Dueck, G.W.: A transformation based algorithm for reversible logic synthesis. In: *Design Automation Conference*, pp. 318–323 (2003)
21. Miller, D.M., Wille, R., Sasanian, Z.: Elementary quantum gate realizations for multiple-control toffoli gates. In: *International Symposium on Multi-Valued Logic*, pp. 288–293 (2011)
22. Mottonen, M., Vartiainen, J.J.: Decompositions of general quantum gates. Ch. 7 in *Trends in Quantum Computing Research*, NOVA Publishers, New York (2006). <http://www.citebase.org/abstract?id=oai:arXiv.org:quant-ph/05%04100>
23. Muthukrishnan, A., Stroud, C.R.: Multivalued logic gates for quantum computation. *Phys. Rev. A* **62**, 052,309 (2000)
24. Nickerson, N.H., Li, Y., Benjamin, S.C.: Topological quantum computing with a very noisy network and local error rates approaching one percent. *Nat Commun.* **4**, 1756 (2013)
25. Nielsen, M., Chuang, I.: *Quantum Computation and Quantum Information*. Cambridge Univ Press, Cambridge (2000)
26. Ohliger, M., Eisert, J.: Efficient measurement-based quantum computing with continuous-variable systems. *Phys. Rev. A* **85**, 062,318 (2012). doi:[10.1103/PhysRevA.85.062318](https://doi.org/10.1103/PhysRevA.85.062318)
27. Peres, A.: Reversible logic and quantum computers. *Phys. Rev. A* **32**, 3266–3276 (1985)
28. Saeedi, M., Sedighi, M., Zamani, M.S.: A novel synthesis algorithm for reversible circuits. In: *International Conference on CAD*, pp. 65–68 (2007)
29. Saeedi, M., Wille, R., Drechsler, R.: Synthesis of quantum circuits for linear nearest neighbor architectures. *Quantum Inf. Process.* **10**(3), 355–377 (2011)
30. Sasanian, Z., Miller, D.M.: Transforming MCT circuits to NCVW circuits. In: *Reversible Computation 2011. Lecture Notes in Computer Science* vol. 7165, pp. 77–88 (2012)
31. Sasanian, Z., Wille, R., Miller, D.M.: Realizing reversible circuits using a new class of quantum gates. In: *Design Automation Conference*, pp. 36–41 (2012)
32. Shende, V.V., Prasad, A.K., Markov, I.L., Hayes, J.P.: Synthesis of reversible logic circuits. *IEEE Trans. CAD* **22**(6), 710–722 (2003)
33. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. *Found. Comput. Sci.* pp. 124–134 (1994)

34. Soeken, M., Frehse, S., Wille, R., Drechsler, R.: RevKit: An Open Source Toolkit for the Design of Reversible Circuits. In: Reversible Computation 2011, Lecture Notes in Computer Science, vol. 7165, pp. 64–76 (2012). RevKit is available at www.revkit.org
35. Soeken, M., Wille, R., Hilken, C., Przigoda, N., Drechsler, R.: Synthesis of reversible circuits with minimal lines for large functions. In: ASP Design Automation Conference, pp. 85–92 (2012)
36. Toffoli, T.: Reversible computing. In: de Bakker, W., van Leeuwen, J. (eds.) Automata, Languages and Programming, p. 632. Springer, Technical Memo MIT/LCS/TM-151. MIT Lab. for Comput. Sci (1980)
37. Wille, R., Drechsler, R.: BDD-based synthesis of reversible logic for large functions. In: Design Automation Conference, pp. 270–275 (2009)
38. Wille, R., Große, D., Teuber, L., Dueck, G.W., Drechsler, R.: RevLib: an online resource for reversible functions and reversible circuits. In: International Symposium on Multi-Valued Logic, pp. 220–225 (2008). RevLib is available at <http://www.revlib.org>
39. Wille, R., Offermann, S., Drechsler, R.: SyReC: a programming language for synthesis of reversible circuits. In: Forum on Specification and Design Languages, pp. 184–189 (2010)
40. Yao, N.Y., Gong, Z.X., Laumann, C.R., Bennett, S.D., Duan, L.M., Lukin, M.D., Jiang, L., Gorshkov, A.V.: Quantum logic between remote quantum registers. *Phys. Rev. A* **87**, 022,306 (2013). doi:[10.1103/PhysRevA.87.022306](https://doi.org/10.1103/PhysRevA.87.022306)