Adjustable-depth quantum circuit for position-dependent coin operators of discrete-time quantum walks

Ugo Nzongani, Pablo Arnault

Adjustable-depth quantum circuit for position-dependent coin operators of discrete-time quantum walks

Ugo Nzongani[1,*] and Pablo Arnault[1,†]

[1] *Université Paris-Saclay, CNRS, ENS Paris-Saclay, INRIA, Laboratoire Méthodes Formelles, 91190 Gif-sur-Yvette, France*

Discrete-time quantum walks with position-dependent coin operators have numerous applications. For a position dependence that is sufficiently smooth, it has been provided in Ref. [1] an approximate quantum-circuit implementation of the coin operator that is efficient. If we want the quantum-circuit implementation to be exact (e.g., either, in the case of a smooth position dependence, to have a perfect precision, or in order to treat a non-smooth position dependence), but the depth of the circuit not to scale exponentially, then we can use the linear-depth circuit of Ref. [1], which achieves a depth that is linear at the cost of introducing an exponential number of ancillas. In this paper, we provide an adjustable-depth quantum circuit for the exact implementation of the position-dependent coin operator. This adjustable-depth circuit consists in (i) applying in parallel, with a linear-depth circuit, only certain packs of coin operators (rather than all of them as in the original linear-depth circuit [1]), each pack contributing linearly to the depth, and in (ii) applying sequentially these packs, which contributes exponentially to the depth.

I. INTRODUCTION

Quantum walks are models of quantum transport on graphs [2, 3]. They exist both in continuous time [4] and in discrete time [5–7]. In terms of computer science, quantum walks are a model of computation, which has been shown to be universal, both in the continuous- [8, 9] and in the discrete-time case [10], that is, any quantum algorithm can be written in terms of a quantum walk; moreover, many algorithms solving a variety of tasks have been conceived with quantum walks [11–15]. In terms of physics, quantum walks are particularly suited to simulate quantum partial differential equations such as the Schrödinger equation or the Dirac equation [16–24] – the latter being the equation of motion for matter particles which are both quantum and relativistic –, or models of solid-state physics [25].

Discrete-time quantum walks (DQWs), in particular, are discretizations of the Dirac equation which respect both unitarity – as continuous-time quantum walks (CQWs) – and strict locality of the transport (contrary to CQWs), that is, concerning the latter point, they preserve relativistic locality on the lattice [23, 26–28]. These DQWs combine, as basic ingredients, shifts on the lattice which depend on the internal state of the particle, together with internal-state rotations, called coin operators, that "reshuffle the cards" regarding whether one goes in one direction or another. The Dirac equation coupled to a variety of gauge fields has been shown to be simulatable with DQWs having a coin operator which depends on the position of the walker on the graph [29–38]. Position-dependent coin operators also arise when considering randomly chosen coin operators, which are a model of noise in DQWs [39–44].

In Ref. [1], we have presented different quantum circuits that achieve the implementation of a DQW on the line with such a position-dependent coin operator. In this paper, we propose a family of quantum circuits with adjustable depth, parametrized by a parameter $m \in \mathbb{N}$, the extremes of which are (i) the naive circuit of Ref. [1] for $m = 0$, which means that all coin operators are implemented sequentially, and (ii) the linear-depth circuit of Ref. [1] for $m = n$ (where n is the number of qubits used to encode the position of the walker in base 2), which means that all coin operators are implemented in parallel. A higher (lower) m means that more (less) coin operators are implemented in parallel, so that one can choose m as best suited for the experimental platform, knowing that a higher m, and hence a smaller depth, requires more ancillary qubits.

In Sec. II, we recall the system on which we work, which is that of Ref. [1], namely, a DQW on the cycle with $N = 2^n$ nodes, $n \in \mathbb{N}$. Such a DQW is made of two operations: a coin rotation $C^{(n)}$, and a coin-dependent shift operation $S^{(n)}$. Still in Sec. II, we recall how to implement $S^{(n)}$ with a quantum circuit. In Sec. III, we introduce our adjustable-depth quantum circuit for the implementation of the position-dependent coin rotation $C^{(n)}$. The idea of this circuit is to (i) apply in parallel, with a linear-depth circuit such as that introduced in Ref. [1], only certain packs of coin operators (rather than all of them as in the original linear-depth circuit [1]), each pack contributing linearly to the depth, and to (ii) apply sequentially these packs, which contributes exponentially to the depth. In Sec. IV, we implement our adjustable-depth quantum circuit on IBM's QASM, the classical simulator of IBM's quantum processors. In Sec. V, we conclude and discuss our results.

* ugo.nzongani@universite-paris-saclay.fr
† pablo.arnault@inria.fr

II. FRAMEWORK

A. The walk

The system we consider is the same as that of Ref. [1], namely, a DQW on a cycle with $N = 2^n$ nodes, $n \in \mathbb{N}$. Let us briefly recall the features of this system. Each node, labeled as $k = 0, \dots, N-1$, is associated to a position quantum state $|k\rangle$. Let $\mathcal{H}_{\text{pos.}}$ be the 2^n-dimensional Hilbert space spanned by the position basis $\{|k\rangle\}_{k=0,\dots,N-1}$. The quantum state of the DQW has an additional, "internal" degree of freedom, which is called the coin. Such a coin belongs to a two-dimensional Hilbert space $\mathcal{H}_0$, the basis of which is $(|\uparrow\rangle, |\downarrow\rangle) \equiv (|0\rangle, |1\rangle) \equiv ((1,0)^\top, (0,1)^\top)$, where $\top$ denotes the transposition. The total Hilbert space to which the quantum state belongs is therefore $\mathcal{H} := \mathcal{H}_{\text{pos.}} \otimes \mathcal{H}_0$. Such a quantum state at time $j \in \mathbb{N}$ decomposes as follows on the basis of $\mathcal{H}$,

$$|\psi_j\rangle = \sum_{k=0}^{N-1} \left(\psi_{j,k}^{\uparrow} |k\rangle |0\rangle + \psi_{j,k}^{\downarrow} |k\rangle |1\rangle \right) , \tag{1}$$

where the complex numbers $\psi_{j,k}^{\uparrow}$ and $\psi_{j,k}^{\downarrow}$ are the coefficients of the decomposition.

The evolution of the quantum state, Eq. (1), is governed by the following dynamics,

$$|\psi_{j+1}\rangle = W^{(n)} |\psi_j\rangle , \tag{2}$$

where the *walk operator* $W^{(n)}$ is composed of two operations,

$$W^{(n)} := S^{(n)} C^{(n)} . \tag{3}$$

The first operation is a possibly position-dependent total *coin operator*,

$$C^{(n)} := \sum_{k=0}^{N-1} |k\rangle\langle k| \otimes C_k , \tag{4}$$

where each C_k is a coin operator, that is, here, a 2×2 complex matrix acting on $\mathcal{H}_0$. The second and last operation is a *coin-dependent shift operator*,

$$\begin{aligned} S^{(n)} := \sum_{k=0}^{N-1} \Big(& |k-1 \bmod N\rangle\langle k| \otimes |0\rangle\langle 0| \\ & + |k+1 \bmod N\rangle\langle k| \otimes |1\rangle\langle 1| \Big) . \end{aligned} \tag{5}$$

B. The encoding in base 2

What is the minimum number of entangled qubits, i.e., of wires, that we need in our quantum circuit in order to encode the $N = 2^n$ nodes of the cycle: this minimum number is n, and we call these qubits *position qubits*. This naturally provides a base-2 encoding of the position of the walker. Let $|k_2\rangle$ be the writing of $|k\rangle$ in base 2, that is:

$$|k\rangle \equiv |k_2\rangle \equiv |b_{n-1} \dots b_0\rangle , \tag{6}$$

where $b_p = 0$ or 1 with $p = 0, \dots, n-1$, such that $k = \sum_{p=0}^{n-1} b_p \times 2^p$. One can thus rewrite Eqs. (4) and (5) as

$$C^{(n)} \equiv \sum_{k=0}^{2^n-1} |k_2\rangle\langle k_2| \otimes \tilde{C}_{k_2} , \tag{7}$$

where

$$\tilde{C}_{k_2} := C_k , \tag{8}$$

and,

$$\begin{aligned} S^{(n)} \equiv \sum_{k=0}^{2^n-1} \Big(& |(k-1 \bmod N)_2\rangle\langle k_2| \otimes |0\rangle\langle 0| \\ & + |(k+1 \bmod N)_2\rangle\langle k_2| \otimes |1\rangle\langle 1| \Big) . \end{aligned} \tag{9}$$

C. Final remarks

In order to be able to implement suck a walk $W^{(n)}$ with a quantum circuit, one has to be able to implement the two operators $C^{(n)}$ and $S^{(n)}$. There are different ways of implementing the coin-dependent shift operator $S^{(n)}$ using a quantum circuit, as recalled in Ref. [1]. The aim of this paper is the quantum-circuit implementation of a position-dependent total coin operator $C^{(n)}$ with a circuit having a depth that can be adjusted at will, which is the subject of the next section.

III. ADJUSTABLE-DEPTH CIRCUIT

A. General idea

1. The idea

As mentioned in the introduction, the general idea of the adjustable-depth circuit we introduce in this paper is to (i) apply in parallel, with a linear-depth circuit such as that introduced in Ref. [1], only certain packs of coin operators $\tilde{C}_{k_2}$ (rather than all of them as in the original linear-depth circuit [1]), each pack contributing linearly to the depth, and to (ii) apply sequentially these packs, which contributes exponentially to the depth.

The total number of coin operators $\tilde{C}_{k_2}$ is 2^n. The size of the packs, i.e., the number of coin operators that we apply in parallel, is the tunable parameter of our model, and we write it as a power of 2 to simplify the discussion, that is, we write it $M = 2^m$, $m \in \mathbb{N}$. The number of packs is thus $2^n/2^m = 2^{n-m}$. Let us call $U_i^{(n,m)}$, $i =$

$0, ..., 2^{n-m}-1$, the circuit that implements the ith pack of coin operators $\tilde{C}_{k_2}$ in parallel. The total circuit, which we are going to show implements the coin operator $C^{(n)}$, thus reads

$$U^{(n,m)} := \prod_{i=0}^{2^{n-m}-1}{}^{L} U_i^{(n,m)} , \tag{10}$$

where the superscript L means that the terms are multiplied in increasing index order from right to left. The number i is called the stage number.

2. *The ancillae, and more precisions*

The number of ancillae necessary to apply each $U_i^{(n,m)}$ is: 2^m ancillary position states, and 2^m-1 ancillary coin states. We depict in Fig. 1 the different registers used to implement our circuit $U^{(n,m)}$. In Fig. 2, we illustrate Eq. (10).

As in Eq. (26) of Ref. [1], the fact that $U^{(n,m)}$ does the job, i.e., implements $C^{(n)}$, means that it coincides with $C^{(n)}$ on the Hilbert space spanned by the position qubits plus the coin qubit, provided that we have correctly initialized the ancillary qubits. We are going to detail this in the next paragraph.

Including the ancillae means extending the total Hilbert space $\mathcal{H}$ introduced in Sec. II into $\mathcal{H}' := \mathcal{H} \otimes \mathcal{H}'_{\text{coins}} \otimes \mathcal{H}'_{\text{pos.}}$. The last two Hilbert spaces contain respectively the quantum states of the coin and position ancillary qubits. A correctly initialized quantum state $|S\rangle \in \mathcal{H}'$ is a state that is arbitrary on $\mathcal{H}$, but has to be equal to $|s'=0\rangle\,|b'=0\rangle$ on $\mathcal{H}'_{\text{coins}} \otimes \mathcal{H}'_{\text{pos.}}$, that is,

$$|S\rangle := \left(\sum_{k=0}^{2^n-1} \sum_{s_0=0,1} \alpha_{k,s_0}\,|k_2\rangle\,|s_0\rangle \right) |s'=0\rangle\,|b'=0\rangle \ , \tag{11}$$

with the α_{k,s_0}'s being complex numbers such that $\sum_{k=0}^{2^n-1}\sum_{s_0=0,1}|\alpha_{k,s_0}|^2 = 1$. As we said above, that $U^{(n,m)}$ does the job, i.e., implements $C^{(n)}$, means the following,

$$U^{(n,m)}\,|S\rangle = \left(C^{(n)} \otimes I_{2^{(2^m-1)}} \otimes I_{2^{(2^m)}} \right) |S\rangle \ . \tag{12}$$

Notice that in Eq. (11) we have chosen to represent, in that order: the state of the position qubits $|k_2\rangle$, the principal coin $|s_0\rangle$, the ancillary coins $|s'\rangle$, and finally the ancillary position $|b'\rangle$. This choice has been made for a clearer formulation of the equations. In contrast, in the diagrammatic representations of our circuits, see Figs. 1 and 2, the principal coin was placed under the ancillary coins. This choice has been made for a better visual understanding of the functioning of the circuits.

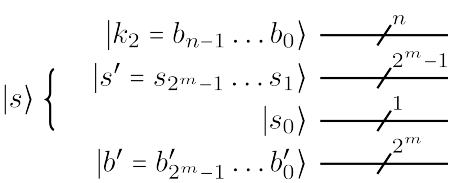

Figure 1: Registers necessary for the implementation of $U^{(n,m)}$.

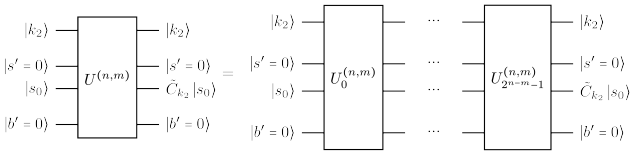

Figure 2: Decomposition of $U^{(n,m)}$ in 2^{n-m} packs $U_i^{(n,m)}$, as written in Eq. (10).

B. General structure of $U_i^{(n,m)}$: that of $U_{\text{lin.}}^{(n)}$ of Ref. [1]

As the circuit $U_{\text{lin.}}^{(n)}$ in Ref. [1], each $U_i^{(n,m)}$ is made of several operations; more precisely, it reads

$$U_i^{(n,m)} := Q_{1,i}^{(n,m)\dagger} Q_2^{(n,m)\dagger} Q_{0,i}^{(n,m)} Q_2^{(n,m)} Q_{1,i}^{(n,m)} \ . \tag{13}$$

Let us first briefly recall the operating principle of $U_{\text{lin.}}^{(n)}$, which is also that of each $U_i^{(n,m)}$: we first encode the ancillary position with $Q_{1,i}^{(n,m)}$; we then swap the state of the principal coin $|s_0\rangle$ onto the ancillary coins via $Q_2^{(n,m)}$; we then apply the running, pack i of coin operators in parallel, via $Q_{0,i}^{(n,m)}$; finally, the ancillary coin states are reset via $Q_2^{(n,m)\dagger}$, and the ancillary position states are reset via $Q_{1,i}^{(n,m)\dagger}$. Equation (13) is illustrated in Fig. 3.

Now, the central operation, $Q_{0,i}^{(n,m)}$, is the same as $Q_0^{(n)}$ in $U_{\text{lin.}}^{(n)}$ of Ref. [1], except that we only apply 2^m coin operators $\tilde{C}_{k_2}$ in parallel instead of 2^n. More precisely, $Q_{0,i}^{(n,m)}$ reads

$$Q_{0,i}^{(n,m)} := I_{2^n} \otimes \left(\bigotimes_{k=0}^{2^m-1}{}^{L} K_{b'_k,s_k}(C_{i2^m+k}) \right) , \tag{14}$$

where I_{2^n} is applied on the position qubits, and where $K_{a,b}(C)$ corresponds to applying the one-qubit gate C on qubit $|b\rangle$ while controlling it on qubit $|a\rangle$ (we apply C only if $a=1$). In Fig. 4, we illustrate the definition of $Q_{0,i}^{(n,m)}$ in Eq. (14). For $m=n$, we have a single operator $U_0^{n,m=n} = U_{\text{lin.}}^{(n)}$.

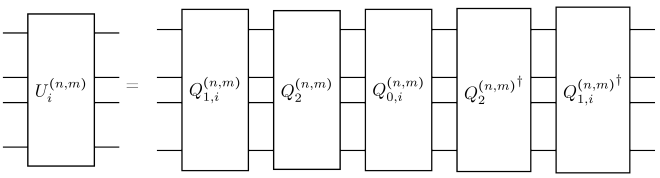

Figure 3: Decomposition of each $U_i^{(n,m)}$, written in Eq. (13).

C. New ingredient: only in the operator $Q_{1,i}^{(n,m)}$

1. Introduction

Apart from the fact that we have less ancillae for $m < n$ than for $m = n$, and that $Q_{0,i}^{(n,m)}$ only applies 2^m coin operators $\tilde{C}_{k_2}$ in parallel, the only difference between $U_i^{(n,m)}$ and the circuit $U_{\text{lin.}}^{(n)}$ of Ref. [1], is in the operation that initializes the ancillary positions, namely, $Q_{1,i}^{(n,m)}$. Let us explain this difference.

Previously, the operator $Q_1^{(n)}$ of Ref. [1] encoded the ancillary position $|b'\rangle$ for any position $|k_2\rangle$. Now, as we apply in parallel packs of 2^m coin operators only, one must only encode the ancillary position for 2^m position states only, not all of them.

We call *current position state at stage* i a position state $|k_2^{(i)}\rangle$ such that after stage i the coin operator $\tilde{C}_{k_2^{(i)}}$ has been applied to the principal coin state $|s_0\rangle$. There are thus 2^m current position states at each stage i, namely, those for which $k^{(i)} = i2^m, ..., (i+1)2^m - 1$ (see Eq. (14) and Fig. 4).

In Ref. [1], the ancillary position is encoded for every position state thanks to the fact that the qubit which encodes the least significant bit of the ancillary position, namely $|b'_0\rangle$, is flipped using a NOT gate, before the application of the series of controlled-SWAP operations (see the operation Q_{11} in Fig. 15 of Ref. [1]).

2. Main explanations

To encode only the current position states at each stage i, one has to flip the same qubit $|b'_0\rangle$ but only for these position states. Let us explain how to do that. Let $|k_2\rangle = |b_{n-1}...b_0\rangle$ be the input position state. The current position states at any stage i have their last $n-m$ bits in common in their binary writing, which is of the form $k_2^{(i)} := b_{n-1}^{(i)}...b_0^{(i)}$. More specifically, it turns out that these $n-m$ bits in common actually code for the binary writing i_2 of i, that is, we have

$$i_2 := h_{n-m-1}...h_0 \tag{15a}$$

$$:= b_{n-1}^{(i)}...b_m^{(i)} . \tag{15b}$$

Hence, flipping $|b'_0\rangle$ only for the current position states at stage i can be done by controlling the NOT gate with positive (i.e., on 1) and/or negative (i.e., on 0) controls on the first $n-m$ position qubits from top to bottom, starting from $|b_{n-1}\rangle$, such that the NOT gate is activated if and only if

$$b_{n-1}...b_m = i_2 . \tag{16}$$

This corresponds to applying a certain generalized $(n-m)$-Toffoli gate, where "generalized" means with positive and/or negative controls. As a reminder, an n-Toffoli gate is a NOT gate controlled positively by n qubits. Note that a 0-Toffoli gate thus denotes a NOT gate. The encoding performed by $Q_{1,i}^{(n,m)}$ is then $|b' = 0\rangle \to |b' = (2^{k^{(i)}} - i2^m)_2\rangle$ for the current position states $k^{(i)}$, and $|b' = 0\rangle \to |b' = 0\rangle$ (i.e., the identity) for the other position states.

In Fig. 5, we illustrate the above-mentioned generalized $(n-m)$-Toffoli gates of each $Q_{1,i}^{(n,m)}$ for $n = 3$ and $m = 1$. These generalized $(n-m)$-Toffoli gates replace the NOT gate at the beginning of Q_{11} in Q_1 of $U_{\text{lin.}}^{(n)}$ in Ref. [1].

In Appendix A, we give the explicit definition of $Q_{1,i}^{(n,m)}$. In Appendix B, we, for the sake of completeness, write explicitly $Q_2^{(n,m)}$, which initializes the ancillary coins, but as mentioned the only change with respect to $Q_2^{(n)}$ of Ref. [1] is the number of ancillary qubits.

In appendices A and B, we have omitted certain identity tensor factors in certain equations, in order to lighten the writing.

In total, since at each stage the ancillary positions are encoded only if we have a current position state at stage i, then it means that the coin operator at a given position is applied only if that position is a current position at stage i: in other words, $U_i^{(n,m)}$ coincides on $\mathcal{H}$ with (i) the coin operator at a given position if that given position is a current position at stage i, and with (ii) the identity otherwise, which achieves our goal.

D. Depth

The purpose of this adjustable-depth circuit is that, depending on the parameter m, its width and depth will be modified, one for the benefit of the other. Let $w(\cdot)$ and $d(\cdot)$ be respectively the functions that return the width and the depth before compilation of an operator. One also needs to define functions $\varepsilon_w(x)$ and $\varepsilon_d(x)$ which return respectively the number of ancillary qubits that may be needed to implement an x-Toffoli gate, and the depth after compilation of the latter. Finally, $\delta_{i,j}$ is the Kronecker symbol.

Since one uses n position qubits, 2^m coin qubits and 2^m ancillary position qubits, the width of $U^{(n,m)}$ reads

$$w(U^{(n,m)}) = n + 2^{m+1} + \varepsilon_w(n-m) . \tag{17}$$

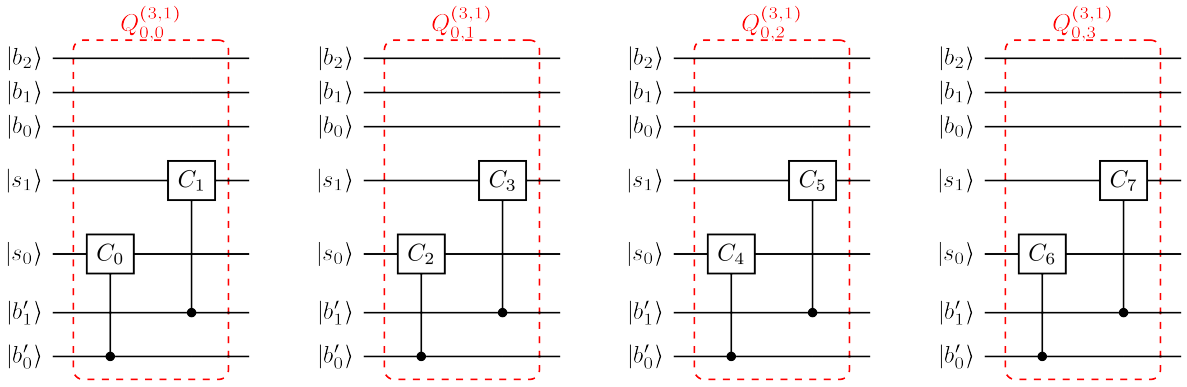

Figure 4: Quantum circuits implementing $Q_{0,i}^{(n=3,m=1)}$ for $i = 0, 1, 2, 3$, that is, we implement, at each stage, $2^m = 2$ coin operators (out of $2^n = 8$) in parallel.

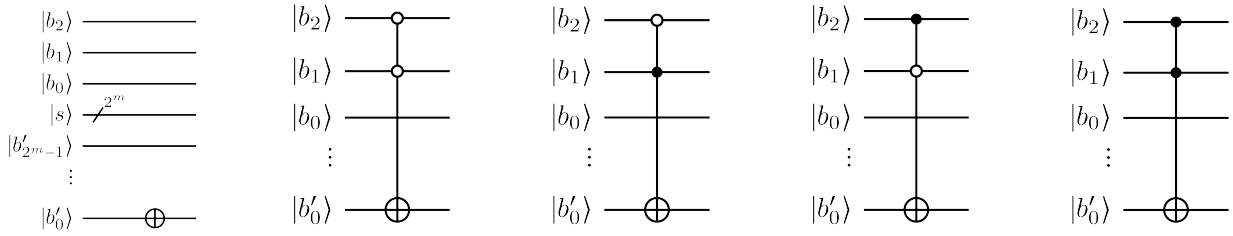

Figure 5: We choose $n = 3$ and $m = 1$. On the far left, there is a circuit with only a NOT gate on $|b'_0\rangle$, which is what would be used at the beginning of the controlled-SWAP operations of each $Q_{1,i}^{(n,m)}$, if there was no difference with $Q_1^{(n)}$ of $U_{\text{lin.}}^{(n)}$ in Ref. [1]. But, here is precisely the main new ingredient of the present adjustable-depth circuit to implement the position-dependent coin operators: instead of this NOT gate, we have to apply, at the beginning of the controlled-SWAP operations of $Q_{1,i}^{(n,m)}$, a generalized $(n-m)$-Toffoli gate which activates the NOT gate if and only if $b_{n-1}...b_m = i_2$, where i_2 is the binary writing of i. From left to right starting from the second circuit, we have depicted this generalized $(n-m)$-Toffoli gate for $i = 0, 1, 2, 3$. The term "generalized" simply refers to the fact that the controls can be positive (black dot) or negative (white dot).

As for the total depth of the circuit, we show in Appendix C that it is given by

$$d(U^{(n,m)}) = 2^{n-m}(20m + 2\varepsilon_d(n-m) + 8\delta_{m,0} - 5) - 2\,. \tag{18}$$

As shown in Ref. [45], one can implement an n-Toffoli gate linearly in n without using any ancilla. Therefore, one can consider the term $\varepsilon_d(n-m)$ to be linear in $n-m$, and $\varepsilon_w(n-m) = 0$. We finally remark the exponential dependence in the number of packs $n-m$, namely, 2^{n-m}, and the linear dependence in the number m of position qubits involved per pack, namely, $20m$ (plus the linear dependence of $\varepsilon_d(n-m)$ in $n-m$).

In Fig. 6, we present the different width and depth complexities of $U^{(n,m)}$ for some remarkable values of m, with $N = 2^n$.

m	Depth complexity	Width complexity
$m = 0$	$O(\log_2(N)N)$	$O(\log_2(N))$
$m = \frac{n}{2}$	$O(\log_2(N)\sqrt{N})$	$O(\sqrt{N})$
$m = n$	$O(\log_2(N))$	$O(N)$

Figure 6: Depth and width complexity of our adjustable-depth quantum circuit $U^{(n,m)}$, implementing the position-dependent coin operator, for remarkable values of m, where we recall that $N = 2^n$.

IV. IMPLEMENTATION

We have implemented our adjustable-depth quantum circuit on IBM's QASM, the classical simulator of IBM's quantum processors, thanks to the software Qiskit.

In Fig. 7, we show how this quantum circuit looks like for $n = 2$ position qubits and an adjustable parameter $m = 0, 1, 2$. In Appendix E, we give the pseudo-code

used in order to generate these circuits.

In Fig. 8, we show the probability distribution obtained after 100 time steps of running different circuits, with coin operators parametrized by

$$K(\alpha,\theta,\phi,\lambda) := e^{i\alpha}\begin{bmatrix} \cos\frac{\theta}{2} & -e^{i\lambda}\sin\frac{\theta}{2} \\ e^{i\phi}\sin\frac{\theta}{2} & e^{i(\phi+\lambda)}\cos\frac{\theta}{2} \end{bmatrix}, \tag{19}$$

with angles taken at random for each coin operators C_i, in the intervals

$$\alpha,\theta \in [0,\pi[\text{ and } \phi,\lambda \in [-\pi,\pi[\,. \tag{20}$$

The values obtained are given in Table 9.

In Fig. 10, we show, as a function of m, the depth, width and size of our adjustable-depth quantum circuit, after compilation by QASM. The size is the number of one- and two-qubit gates involved in the circuit. The compilation has been done with the following set of universal gates:

$$R_X(\theta) := \begin{bmatrix} \cos\frac{\theta}{2} & -i\sin\frac{\theta}{2} \\ -i\sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{21a}$$

$$R_Y(\theta) := \begin{bmatrix} \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2} \end{bmatrix} \tag{21b}$$

$$R_Z(\lambda) := \begin{bmatrix} e^{-i\frac{\lambda}{2}} & 0 \\ 0 & e^{i\frac{\lambda}{2}} \end{bmatrix} \tag{21c}$$

$$P(\lambda) := \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix} \tag{21d}$$

$$CNOT := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \tag{21e}$$

As m increases, the depth decreases but the width increases, which is expected (the decrease of the depth being the purpose of increasing m, and the increase of the width being the consequence of needing to add ancillary wires in order to decrease the depth). Moreover, it is interesting to note the following non-trivial behavior: the number of gates decreases as m increases, which may come from the fact that, as m increases, the number of multi-Toffoli gates decreases (indeed, these multi-Toffoli gates have a high cost in terms of number of one- and two-qubit gates needed to implement them).

V. CONCLUSIONS AND DISCUSSION

In this paper, we have provided a family of quantum circuits that implement a DQW on the line, with the following characteristics. This family is parametrized by $m \in \mathbb{N}$. For $m = 0$, the circuit coincides with the naive

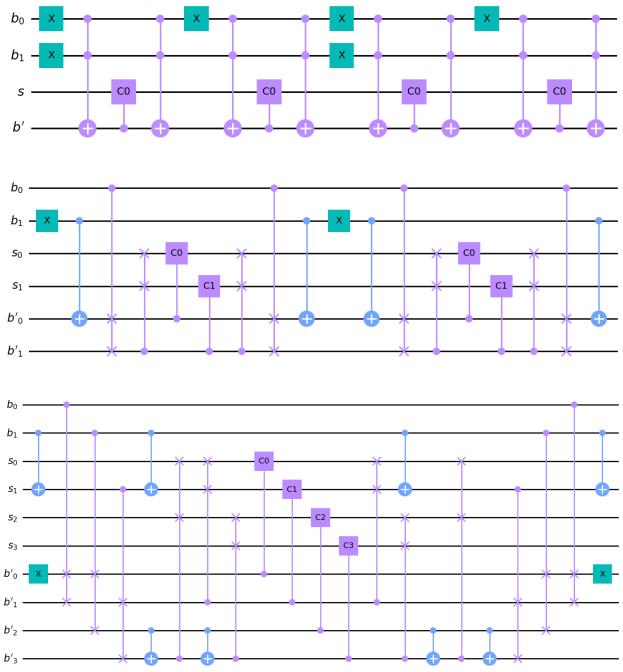

Figure 7: Our adjustable-depth quantum circuit for $n = 2$ position qubits, and packs of size 2^m, $m = 0$ (top circuit), 1 (middle circuit), and 2 (bottom circuit), which means that 2^m coin operators are executed in parallel at each of the 2^{n-m} stages in the quantum circuit.

circuit of Ref. [1], which means that all the coin operators at each site of the line are implemented sequentially. For $m = n$, the number of qubits used to encode the position of the walker, the circuit coincides with the linear-depth circuit of Ref. [1], which means that all the coin operators are implemented in parallel. A circuit with a given arbitrary m means that the circuit contains 2^{n-m} packs, each of them containing 2^m coin operators implemented in parallel. A higher (lower) m means that more (less) coin operators are implemented in parallel, so that one can choose m as best suited for the experimental platform, knowing that a higher m, and hence a smaller depth, requires more ancillary qubits.

It would be interesting to characterize the specific properties of DQWs, such as strict locality, at the level of their quantum-circuit translation [46]. It surely would be very interesting to extend the results of Ref. [1] and of this paper to quantum cellular automata (QCAs), which are multiparticle generalizations of DQWs [47, 48]. Given that there are already some QCAs which simulate quantum electrodynamics in $1+1$, $1+2$ and $1+3$ dimensions [49–51], this would mean having a quantum circuit which simulates some quantum field theory while implementing the strict locality of the transport. If more properties or symmetries of the continuum model are preserved by the QCA that simulates it, such as Lorentz symmetry [23, 26–28] or gauge invariance [49–51], such a quantum-

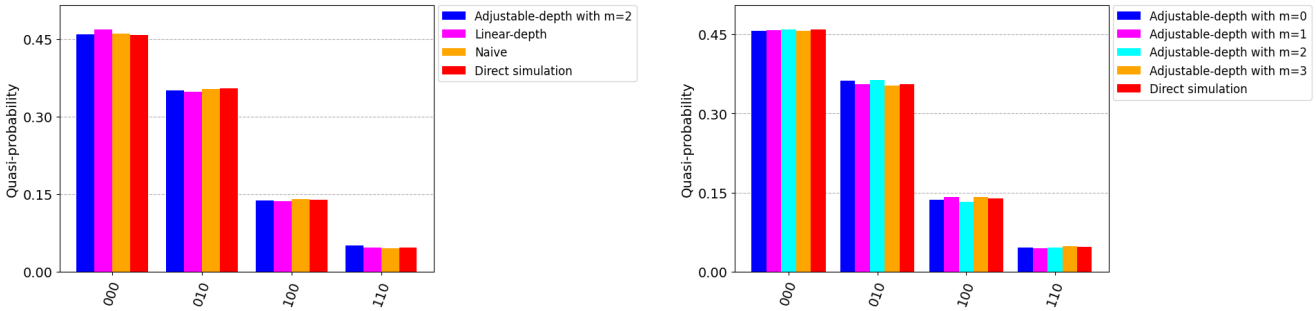

Figure 8: Final probability distribution obtained by running 10000 times a given circuit with $n = 3$ position qubits up to 100 time steps. In the left figure, we show the histograms for the naive circuit and the linear-depth circuit of Ref. [1], for our adjustable-depth circuit with $m = 2$, and for a direct simulation of the walk (i.e., without mapping it onto a quantum circuit), which we have called "Matrix simulation" in Ref. [1] instead of "Direct simulation"; the angles used for the different coin operators are given in Table 9. In the right figure, we show the histograms for different choices of the adjustable parameter $m = 0, 1, 2, 3$.

	α	θ	ϕ	λ
C_0	2.59157236	0.07621657	2.38136754	-2.79936369
C_1	0.99031147	0.27887516	-2.67278043	1.84368898
C_2	0.46354727	3.01620087	1.5039485	2.87444318
C_3	1.79814059	2.5016676	-0.50529796	-1.65451052
C_4	2.24708944	0.90254105	-1.48779474	-0.43149381
C_5	1.99400865	0.7635951	-2.15664402	-0.07448489
C_6	2.69049595	1.05540144	-3.12609191	0.22005922
C_7	2.77026061	1.2182012	-0.29889881	-0.72954279

Figure 9: Values of the angles of the coin operators, chosen for our implementation of the different circuits in Fig. 8. The parametrization is given in Eq. (19).

circuit translation program would provide quantum circuits for quantum field theories, which respect many of the symmetries of the continuum model, which is not only a guarantee of numerical accurateness, but also provides an alternate, lattice definition of these theories, endowed with all the symmetries required by physical principles, thus creating a new, lattice paradigm for such physical theories, phrased in terms of quantum circuits and thus directly implementable on most-used quantum hardware. Finally, a question which is interesting is the following. Imagine that a certain algorithm is conceived with quantum walks, say, DQWs, and we want to run it with a quantum circuit. If we use the known algorithms that translate a DQW into a quantum circuit, do we obtain an algorithm that is as efficient as the original one made with a DQW, or do we have to modify it to reach the original efficiency?

STATEMENT OF ABSENCE OF CONFLICT OF INTEREST

On behalf of all authors, the corresponding authors state that there is no conflict of interest.

DATA AVAILABILITY

Data will be made available upon reasonable request.

[1] U. Nzongani, J. Zylberman, C.-E. Doncecchi, A. Pérez, F. Debbasch, and P. Arnault, "Quantum circuits for discrete-time quantum walks with position-dependent coin operator," arXiv:2211.05271 (2022).

[2] J. Kempe, "Quantum random walks: An introductory overview," Contemp. Phys. **44**, 307–327 (2003).

[3] S. E. Venegas-Andraca, "Quantum walks: a comprehensive review," Quantum Inf. Process. **11**, 1015–1106 (2012).

[4] E. Farhi and S. Gutmann, "Quantum computation and decision trees," Phys. Rev. A **58**, 915–928 (1998).

[5] R. P. Feynman and A. R. Hibbs, *Quantum Mechanics and Path Integrals* (McGraw-Hill, 1965).

[6] Y. Aharonov, L. Davidovich, and N. Zagury, "Quantum random walks," Phys. Rev. A **48**, 1687–1690 (1993).

[7] I. Bialynicki-Birula, "Weyl, Dirac, and Maxwell equations on a lattice as unitary cellular automata," Phys. Rev. D **49**, 6920–6927 (1994).

[8] A. M. Childs, "Universal computation by quantum walk," Phys. Rev. Lett. **102**, 180501 (2009).

[9] A. M. Childs, D. Gosset, and Z. Webb, "Universal computation by multiparticle quantum walk," Science **339**, 791–794 (2013).

[10] N. B. Lovett, S. Cooper, M. Everitt, M. Trevers, and V. Kendon, "Universal quantum computation using the discrete-time quantum walk," Phys. Rev. A **81**, 042330 (2010).

[11] A. Rivosh A. Ambainis, J. Kempe, "Coins make quantum walks faster," in *Proc. 16th Ann. ACM-SIAM Symp. Discrete Algorithms* (ACM-SIAM, 2005).

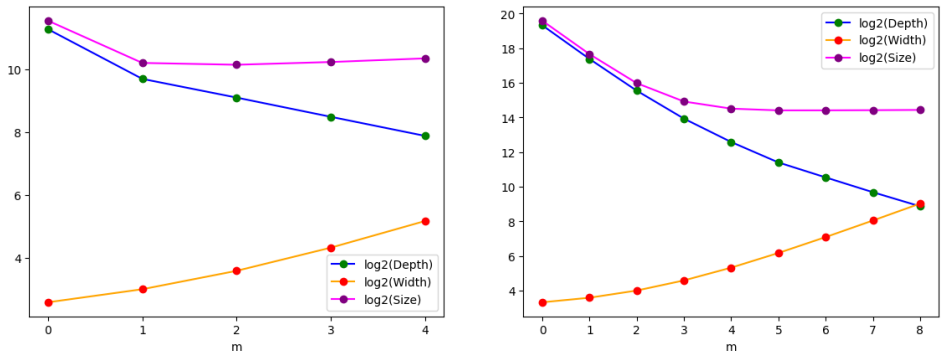

Figure 10: Depth, width, and size of the adjustable-depth quantum circuit, after compilation, for $n = 4$ (left plots) and $n = 8$ (right plots), as a function of the adjustable parameter m.

[12] A. Ambainis, "Quantum walk algorithm for element distinctness," SIAM J. Comput. **37**, 210–239 (2007).

[13] A. Tulsi, "Faster quantum-walk algorithm for the two-dimensional spatial search," Phys. Rev. A **78**, 012310 (2008).

[14] M. Roget, S. Guillet, P. Arrighi, and G. Di Molfetta, "Grover search as a naturally occurring phenomenon," Phys. Rev. Lett. **124** (2020), 10.1103/physrevlett.124.180501.

[15] T. Fredon, J. Zylberman, P. Arnault, and F. Debbasch, "Quantum spatial search with electric potential: Long-time dynamics and robustness to noise," Entropy **24**, 1778 (2022).

[16] F. W. Strauch, "Relativistic quantum walks," Phys. Rev. A **73**, 054302 (2006).

[17] Y. Shikano, "From discrete time quantum walk to continuous time quantum walk in limit distribution," J. Comput. Theor. Nanos. **10**, 1558–1570 (2013).

[18] P. Arrighi, V. Nesme, and M. Forets, "The Dirac equation as a quantum walk: higher dimensions, observational convergence," J. Phys. A **47**, 465302 (2014).

[19] G. M. D'Ariano and P. Perinotti, "Quantum cellular automata and free quantum field theory," Front. Phys. **12**, 120301 (2016).

[20] P. Arnault, A. Pérez, P. Arrighi, and T. Farrelly, "Discrete-time quantum walks as fermions of lattice gauge theory," Phys. Rev. A **99**, 032110 (2019).

[21] G. Di Molfetta and P. Arrighi, "A quantum walk with both a continuous-time limit and a continuous-spacetime limit," Quantum Inf. Process. **19** (2019).

[22] P. Arnault, "Clifford algebra from quantum automata and unitary Wilson fermions," Phys. Rev. A **106**, 012201 (2022), arXiv:2105.12314.

[23] F. Debbasch, "Action principles for quantum automata and Lorentz invariance of discrete time quantum walks," Ann. Phys. **405**, 340–364 (2019).

[24] P. Arnault and C. Cedzich, "A single-particle framework for unitary lattice gauge theory in discrete time," New J. Phys. **24**, 123031 (2022).

[25] P. Arnault, B. Pepper, and A. Pérez, "Quantum walks in weak electric fields and Bloch oscillations," Phys. Rev. A **101**, 062324 (2020).

[26] P. Arrighi, S. Facchini, and M. Forets, "Discrete Lorentz covariance for quantum walks and quantum cellular automata," New. J. Phys. **16**, 093007 (2014).

[27] A. Bisio, G. M. D'Ariano, and P. Perinotti, "Quantum walks, deformed relativity and Hopf algebra symmetries," Philos. Trans. R. Soc. A **374**, 20150232 (2016).

[28] F. Debbasch, "Discrete geometry from quantum walks," Condensed Matter **4**, 40 (2019).

[29] G. Di Molfetta, M. Brachet, and F. Debbasch, "Quantum walks as massless Dirac fermions in curved space," Phys. Rev. A **88**, 042301 (2013).

[30] G. Di Molfetta, F. Debbasch, and M. Brachet, "Quantum walks in artificial electric and gravitational fields," Physica A **397**, 157–168 (2014).

[31] P. Arnault and F. Debbasch, "Landau levels for discrete-time quantum walks in artificial magnetic fields," Physica A **443**, 179–191 (2016).

[32] P. Arnault and F. Debbasch, "Quantum walks and discrete gauge theories," Phys. Rev. A **93**, 052301 (2016).

[33] P. Arnault, G. Di Molfetta, M. Brachet, and F. Debbasch, "Quantum walks and non-Abelian discrete gauge theory," Phys. Rev. A **94**, 012335 (2016).

[34] P. Arnault and F. Debbasch, "Quantum walks and gravitational waves," Ann. Phys. (N. Y.) **383**, 645–661 (2017).

[35] P. Arrighi and S. Facchini, "Quantum walking in curved spacetime: (3+1) dimensions, and beyond," Quantum Inf. Comput. **17**, 810–824 (2017).

[36] P. Arrighi, G. Di Molfetta, I. Márquez-Martín, and A. Pérez, "Dirac equation as a quantum walk over the honeycomb and triangular lattices," Phys. Rev. A **97**, 062111 (2018).

[37] I. Márquez-Martín, P. Arnault, G. Di Molfetta, and A. Pérez, "Electromagnetic lattice gauge invariance in two-dimensional discrete-time quantum walks," Phys. Rev. A **98**, 032333 (2018).

[38] G. Jay, P. Arnault, and F. Debbasch, "Dirac quantum walks with conserved angular momentum," Quantum Stud.: Math. and Found. **8**, 419–430 (2021).

[39] V. Kendon and B. Tregenna, "Decoherence can be useful in quantum walks," Phys. Rev. A **67**, 042315 (2003).

[40] V. Kendon, "Decoherence in quantum walks - a review," Math. Struct. Comput. Sc. **17**, 1169–1220 (2007).

[41] A. Schreiber, K. N. Cassemiro, V. Potoček, A. Gábris, I. Jex, and Ch. Silberhorn, "Decoherence and disorder in quantum walks: From ballistic spread to localization," Phys. Rev. Lett. **106**, 180403 (2011).

[42] A. Ahlbrecht, C. Cedzich, R. Matjeschk, V. B. Scholz, A. H. Werner, and R. F. Werner, "Asymptotic behav-

ior of quantum walks with spatio-temporal coin fluctuations," Quantum Inf. Process. **11**, 12191249 (2012).
[43] G. Di Molfetta and F. Debbasch, "Discrete-time quantum walks in random artificial gauge fields," Quantum Studies: Mathematics and Foundations **3**, 293–311 (2016).
[44] P. Arnault, A. Macquet, A. Anglés-Castillo, I. Márquez-Martín, V. Pina-Canelles, A. Pérez, G. Di Molfetta, P. Arrighi, and F. Debbasch, "Quantum simulation of quantum relativistic diffusion via quantum walks," J. Phys. A: Math. Theor. **53**, 205303 (2020).
[45] M. Saeedi and M. Pedram, "Linear-depth quantum circuits for n-qubit toffoli gates with no ancilla," Phys. Rev. A **87** (2013).
[46] L. Piroli and J. I. Cirac, "Quantum cellular automata, tensor networks, and area laws," Phys. Rev. Lett. **125** (2020), 10.1103/physrevlett.125.190402.
[47] P. Arrighi, "An overview of quantum cellular automata," Natural Computing **18**, 885–899 (2019).
[48] T. Farrelly, "A review of quantum cellular automata," Quantum **4**, 368 (2020).
[49] P. Arrighi, C. Bény, and T. Farrelly, "A quantum cellular automaton for one-dimensional QED," Quantum Inf. Process. **19** (2020).
[50] K. Sellapillay, P. Arrighi, and G. Di Molfetta, "A discrete relativistic spacetime formalism for 1 + 1-QED with continuum limits," Sci. Rep. **12**, 2198 (2022).
[51] N. Eon, G. Di Molfetta, G. Magnifico, and P. Arrighi, "A relativistic discrete spacetime formulation of 3+1 QED," arXiv:2205.03148 (2022).

Appendix A: Explicit definition of $Q_{1,i}^{(n,m)}$

As mentioned in Sec. III C, the only difference between $Q_{1,i}^{(n,m)}$ and $Q_1^{(n)}$ of Ref. [1] is, apart from the number of ancillary qubits, the fact that the first NOT gate of $Q_{11}^{(n)}$ is replaced by a generalized $(n-m)$-Toffoli gate in $Q_{11,i}^{(n,m)}$ (that we are going to define). To describe $Q_{1,i}^{(n,m)}$ explicitly, we thus follow exactly the construction of $Q_1^{(n)}$ in Ref. [1, Appendix C3]. Thus, the explicit definition of operator $Q_{1,i}^{(n,m)}$ can be written

$$Q_{1,i}^{(n,m)} := Q_{10}^{(n,m)\dagger} Q_{11,i}^{(n,m)} Q_{10}^{(n,m)}\,, \tag{A1}$$

where $Q_{10}^{(n,m)}$ makes the copies of the values of the position qubits on the ancillary coins, in order to be able to perform the controlled SWAPs in parallel via $Q_{11,i}^{(n,m)}$, and then one undoes the copies via $Q_{10}^{(n,m)\dagger}$. The amount of copies and SWAPs is no longer quantified by n as in Ref. [1], but by the parameter m. On Fig. 11, we have depicted the quantum circuits implementing $Q_{1,i}^{(n,m)}$ for $n=3$ and $m=2$, so that $i=0,1$. Let us now write explicitly the copies operation, $Q_{10}^{(n,m)}$, and the controlled-SWAPs operation, $Q_{11,i}^{(n,m)}$.

1. The copies: $Q_{10}^{(n,m)}$

We have

$$Q_{10}^{(n,m)} := \prod_{j=0}^{m-2}{}^{L}\, q_{10}^{(n,m)}(j)\,, \tag{A2}$$

where $\forall j \geq 0$,

$$q_{10}^{(n,m)}(j) := \prod_{k=j+1}^{m-1}{}^{L}\, J_k^{(j)}\,, \tag{A3}$$

with

$$J_k^{(0)} := K_{b_k,s_{l_k}}(X) \tag{A4a}$$

$$J_k^{(j\geq 1)} := \left(\prod_{l'=0}^{2^j-2}{}^{L} K_{s_{l_k}+l',s_{l_k}+l'+2^j}(X)\right) K_{b_k,s_{l_k}+\sum_{u=1}^{j}2^{u-1}}(X)\,, \tag{A4b}$$

where we have defined

$$\begin{aligned} l_0 &:= 1 \\ l_k &:= 2^{k-1} - 1 + l_{k-1}, \forall k \geq 1\,, \end{aligned} \tag{A5}$$

and where we remind the reader that $K_{a,b}(C)$ corresponds to applying the one-qubit gate C on qubit $|b\rangle$ while controlling it on qubit $|a\rangle$ (we apply C only if $a=1$).

2. The controlled SWAPs: $Q_{11,i}^{(n,m)}$

The operator $Q_{11,i}^{(n,m)}$ is composed of the generalized $(n-m)$-Toffoli gate, that we denote by $T_i^{(n,m)}$, and of the controlled-SWAP operations. We can write it as

$$Q_{11,i}^{(n,m)} := \Bigg(\underbrace{\prod_{j=0}^{m-1}{}^{L} B_j}_{\text{controlled SWAPs}}\Bigg) T_i^{(n,m)}\,, \tag{A6}$$

where we are going to define $T_i^{(n,m)}$ and the B_j's below.

a. *The generalized multi-Toffoli gate*

What we call generalized multi-Toffoli gate is a multi-Toffoli gate for which the controls can be positive (i.e., on 1) or negative (i.e., on 0). The question here is how to express a negative control in terms of a positive control.

A first, naive idea is to express a negative control by the sequence of a NOT gate, then, a positive control, and finally another NOT gate. But, have in mind that we do not apply $Q_{11,i}^{(n,m)}$ alone, we apply the Hermitian conjugate $Q_{11,i}^{(n,m)\dagger}$ later on. Moreover, the $n-m$ last position

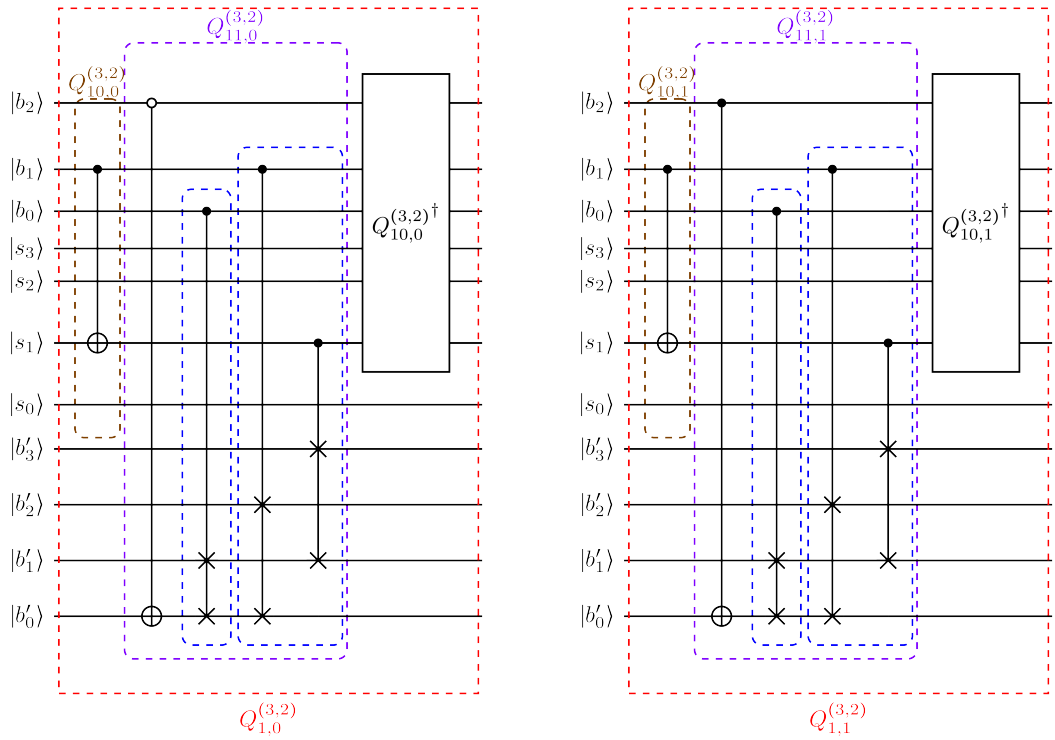

Figure 11: Quantum circuits implementing $Q_{1,i}^{(n=3,m=2)}$ for $i = 0, 1$.

qubits, on which the multi-Toffoli gate is controlled, are used by no other operation within $U_i^{(n,m)}$, neither the copies $Q_{10,i}^{(n,m)}$, nor the controlled SWAPs of $Q_{11,i}^{(n,m)}$, nor $Q_2^{(n,m)}$. So, what we can do is simply applying a NOT gate before applying a positive control in order to obtain a negative control, and then we just wait until the application of $Q_{11,i}^{(n,m)\dagger}$ to undo these operations on the $n-m$ last position qubits. One thus has to flip the qubit, i.e., to (i) apply a NOT gate on $|b_j\rangle$, with $j = m, \ldots, n-1$, whenever the bit h_{j-m} of i_2 is equal to 0, and then to (ii) apply a positive control, which delivers in total a negative control on $|b_j\rangle$.

Thus, we replace $T_i^{(n,m)}$ by

$$\tilde{T}_i^{(n,m)} := \underbrace{K_{\alpha,b'_0}^{\text{multi}}(X)}_{(n-m)\text{-Toffoli}} \left(\underbrace{\left(\bigotimes_{k=m}^{n-1} g_{i,k-m}^{b_k} \right)}_{\text{bit flips}} \otimes I_{2^m} \otimes I_{2^{(2^m)}} \otimes I_{2^{(2^m)}} \right), \tag{A7}$$

which we call almost generalized multi-Toffoli gate. In this equation, the function $g_{i,j}^{b_k}$ indicates when to place the NOT gates on the control position qubit $|b_k\rangle$:

$$g_{i,j}^{b_k} := \begin{cases} X \text{ if } \lfloor \frac{i}{2^j} \mod 2 \rfloor = 0 \\ I_2 \text{ otherwise} \end{cases} . \tag{A8}$$

Moreover, $K_{\alpha,b'_0}^{\text{multi}}(C)$ is the multiply controlled operation that applies gate C on qubit $|b'_0\rangle$ whenever all qubits are 1 in the set of $n-m$ control qubits

$$\alpha := \bigcup_{k=m}^{n-1} \{|b_k\rangle\} . \tag{A9}$$

In Appendix D, we present an alternative method for reducing the amount of NOT gates used in the generalized multi-Toffoli gate.

b. *The controlled SWAPs*

We can write

$$B_j := \left(\bigotimes_{k=1}^{2^j-1} E_{b'_k, b'_k+2^j}^{s_{j+k-1}} \right) E_{b'_0, b'_{2^j}}^{b_j} , \tag{A10}$$

where $E_{b,c}^a$ denotes the controlled-SWAP operation that swaps $|b\rangle$ and $|c\rangle$, controlling this by $|a\rangle$.

Appendix B: Explicit definition of $Q_2^{(n,m)}$

In this appendix, we give an explicit definition of $Q_2^{(n,m)}$, which is the same as that of $Q_2^{(n)}$ of Ref. [1] except for the number of ancillary wires. In Fig. 12, we show the quantum circuits implementing $Q_2^{(n,m)}$ for $n = 3$ and

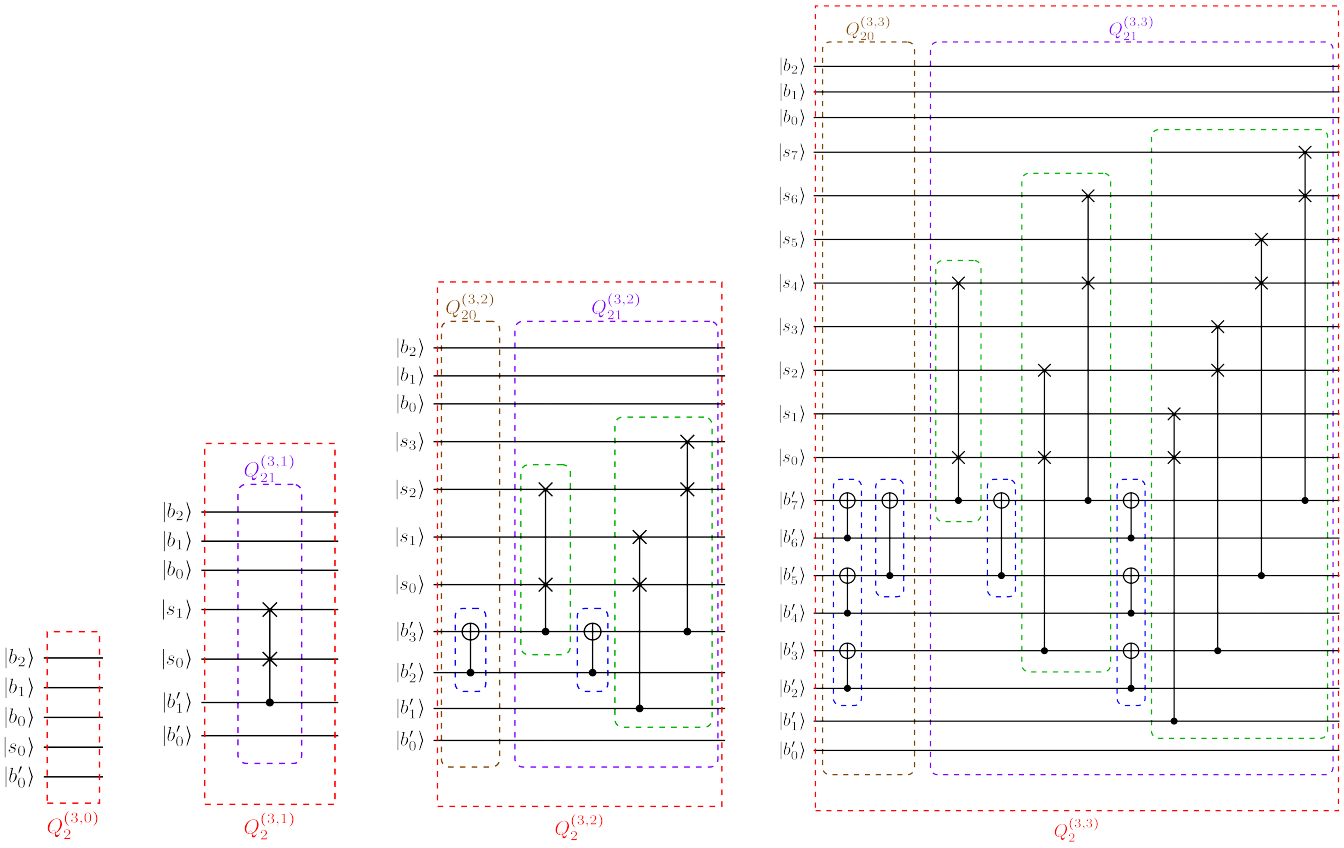

Figure 12: Quantum circuits implementing $Q_2^{(n=3,m)}$, for $m = 0, 1, 2, 3$.

$m = 0, 1, 2, 3$. The operation $Q_2^{(n,m)}$ can be written

$$Q_2^{(n,m)} := Q_{21}^{(n,m)} Q_{20}^{(n,m)}, \tag{B1}$$

where $Q_{20}^{(n,m)}$ corresponds to the the starting series of CNOT operations, and $Q_{21}^{(n,m)}$ to the controlled-SWAP series followed by CNOTs. The explicit definition of $Q_{20}^{(n,m)}$ reads

$$Q_{20}^{(n,m)} := I_{2^n} \otimes I_{2^{(2^m)}} \otimes \left[\prod_{j=0}^{m-2}{}^{L} \left(\bigotimes_{k=0}^{2^{m-j-1}-2} K_{b'_{2^{m-1}-2^{j+1}(\frac{1}{2}+k)}, b'_{2^{m-1}-k2^{j+1}}}(X) \right) \right], \tag{B2}$$

while the explicit definition of $Q_{21}^{(n,m)}$ reads

$$Q_{21}^{(n,m)} := I_{2^n} \otimes \left[\prod_{j=0}^{m-1}{}^{L} C_{m,j} \left(\bigotimes_{k=0}^{2^j-1} E^{b'_{(k+1)2^{m-j}-1}}_{s_{k2^{m-j}}, s_{2^{m-j}(\frac{1}{2}+k)}} \right) \right], \tag{B3}$$

where the CNOTs following the controlled SWAPs are given by

$$C_{m,j} := \begin{cases} \bigotimes_{l=0}^{2^{j+1}-2} K_{b'_{2^{m-1}-2^{m-j-1}(\frac{1}{2}+l)}, b'_{2^{m-1}-l2^{m-j-1}}}(X) & \text{if } j < m-1 \\ I_{2^{(2^m)}} \otimes I_{2^{(2^m)}} & \text{if } j = m-1 \end{cases} \tag{B4}$$

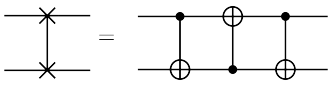

Figure 13: A way of expressing the SWAP operation, with 3 CNOT gates.

Appendix C: Depth-calculation details

In this appendix, we prove the result for the depth of the circuit, given in Eq. (18).
First, we recall that the depth of a SWAP operation counts for 3, as shown in Fig. 13.
Now, the depth of $U^{(n,m)}$, defined in Eqs. (10) and (13), is

$$d(U^{(n,m)}) = \sum_{i=0}^{2^{n-m}-1} d(U_i^{(n,m)}) \tag{C1a}$$

$$= \sum_{i=0}^{2^{n-m}-1} d(Q_{1,i}^{(n,m)\dagger}) + d(Q_2^{(n,m)\dagger}) + d(Q_{0,i}^{(n,m)}) + d(Q_2^{(n,m)}) + d(Q_{1,i}^{(n,m)}) \,. \tag{C1b}$$

Since we apply 2^m coin operators in parallel with $Q_{0,i}^{(n,m)}$, defined in Eq. (14) and illustrated in Fig. 4, we have that

$$d(Q_{0,i}^{(n,m)}) = 1 \,. \tag{C2}$$

The depth of the operator $Q_2^{(n,m)}$, defined in Eqs. (B1), (B2), (B3) and (B4), and illustrated in Fig. 12, is

$$\begin{aligned} d(Q_2^{(n,m)}) &= 3 \underbrace{m}_{\text{SWAPs}} + \underbrace{2(m-1)}_{\text{CNOTs}} \\ &= 5m - 2 \,. \end{aligned} \tag{C3}$$

If $m = 0$, we get a negative depth for $d(Q_2^{(n,m)})$, and the minimum value of the depth must be 0. To solve this problem we perform the following modification,

$$d(Q_2^{(n,m)}) = 5m - 2(1 - \delta_{m,0}) \,. \tag{C4}$$

Moreover, we have that

$$d(Q_2^{(n,m)}) = d(Q_2^{(n,m)\dagger}) \,. \tag{C5}$$

Lastly, the depth of the operator $Q_{1,i}^{(n,m)}$, defined in Eqs. (A1), (A2) and (A6), and illustrated in Fig. 11, is,

$$d(Q_{1,i}^{(n,m)}) = d(Q_{10,i}^{(n,m)\dagger}) + d(Q_{11,i}^{(n,m)}) + d(Q_{10,i}^{(n,m)}) \,, \tag{C6}$$

where

$$d(Q_{10,i}^{(n,m)}) = d(Q_{10,i}^{(n,m)\dagger}) \,, \tag{C7}$$

and

$$d(Q_{10,i}^{(n,m)}) = m - 1 \,. \tag{C8}$$

If $m = 0$ we obtain a negative depth for $d(Q_{10,i}^{(n,m)})$; we therefore perform the following modification,

$$d(Q_{10,i}^{(n,m)}) = m - 1 + \delta_{m,0} \,. \tag{C9}$$

As shown in Eq. (A6), the operator $Q_{11,i}^{(n,m)}$ can be separated into two operations: the $(n-m)$-Toffoli gate on $|b_0'\rangle$, and the series of controlled SWAPs. Let us first treat the $(n-m)$-Toffoli gate. As mentionned in Sec. III C, one has to

flip some of the position qubits before applying the $(n-m)$-Toffoli gate; the only pack i for which no flip is needed is when $i = 2^{n-m}-1$, i.e., the last pack; therefore, we get a contribution $1-\delta_{i,2^{n-m}-1}+\varepsilon_d(n-m)$ to $d(Q_{11,i}^{(n,m)})$, where we recall that $\varepsilon_d(n-m)$ denotes the depth of the $(n-m)$-Toffoli gate. Let us now treat the controlled SWAPs. The depth of the non-parallelized controlled SWAPs is $3\sum_{k=0}^{m-1} 2^k = 3(2^m-1)$; however, as one parallelizes this step in the circuit, the depth becomes only $3m$. Therefore, the depth of $Q_{11,i}^{(n,m)}$ finally reads

$$d(Q_{11,i}^{(n,m)}) = 1-\delta_{i,2^{n-m}-1}+\varepsilon_d(n-m)+3m\,. \tag{C10}$$

Inserting Eqs. (C7), (C9) and (C10) into Eq. (C6), one gets,

$$\begin{aligned} d(Q_{1,i}^{(n,m)}) &= d(Q_{10,i}^{(n,m)\dagger}) + d(Q_{11,i}^{(n,m)}) + d(Q_{10,i}^{(n,m)}) && \text{(C11a)}\\ &= 2d(Q_{10,i}^{(n,m)}) + d(Q_{11,i}^{(n,m)}) && \text{(C11b)}\\ &= 2(m-1+\delta_{m,0})+1-\delta_{i,2^{n-m}-1}+\varepsilon_d(n-m)+3m && \text{(C11c)}\\ &= 5m+\varepsilon_d(n-m)+2\delta_{m,0}-\delta_{i,2^{n-m}-1}-1\,. && \text{(C11d)} \end{aligned}$$

Moreover, we have that

$$d(Q_{1,i}^{(n,m)}) = d(Q_{1,i}^{(n,m)^\dagger})\,. \tag{C12}$$

Inserting now Eqs. (C5) and (C12), and then (C2), (C4) and (C11d), into the expression of $d(U_i^{(n,m)})$ given by Eq. (C1b), we obtain

$$\begin{aligned} d(U_i^{(n,m)}) &= d(Q_{1,i}^{(n,m)\dagger}) + d(Q_2^{(n,m)\dagger}) + d(Q_{0,i}^{(n,m)}) + d(Q_2^{(n,m)}) + d(Q_{1,i}^{(n,m)}) && \text{(C13a)}\\ &= 2(d(Q_2^{(n,m)}) + d(Q_{1,i}^{(n,m)})) + d(Q_{0,i}^{(n,m)}) && \text{(C13b)}\\ &= 2(5m-2(1-\delta_{m,0})+5m+\varepsilon_d(n-m)+2\delta_{m,0}-\delta_{i,2^{n-m}-1}-1)+1 && \text{(C13c)}\\ &= 20m+2\varepsilon_d(n-m)-2\delta_{i,2^{n-m}-1}+8\delta_{m,0}-5\,. && \text{(C13d)} \end{aligned}$$

The only term of $d(U_i^{(n,m)})$ which depends on i is $-2\delta_{i,2^{n-m}-1}$, which is equal to -2 when $i = 2^{n-m}-1$ and 0 for the rest; thus, inserting Eq. (C13d) into Eq. (C1a), we get

$$\begin{aligned} d(U^{(n,m)}) &= \sum_{i=0}^{2^{n-m}-1} d(U_i^{(n,m)}) && \text{(C14a)}\\ &= \sum_{i=0}^{2^{n-m}-1} 20m+2\varepsilon_d(n-m)-2\delta_{i,2^{n-m}-1}+8\delta_{m,0}-5 && \text{(C14b)}\\ &= (2^{n-m}-1)(20m+2\varepsilon_d(n-m)+8\delta_{m,0}-5)+20m+2\varepsilon_d(n-m)+8\delta_{m,0}-7 && \text{(C14c)}\\ &= 2^{n-m}(20m+2\varepsilon_d(n-m)+8\delta_{m,0}-5)-2\,, && \text{(C14d)} \end{aligned}$$

which is the result announced in Eq. (18).

Appendix D: Optimizing the number of NOT gates used in $Q_{1,i}^{(n,m)}$

The NOT gates applied in order to realize the almost generalized multi-Toffoli gate $\tilde{T}_i^{(n,m)}$ (see Eq. (A7)), i.e., applied with the function $g_{i,j}^{b_k}$ before the standard multi-Toffoli gate, are applied again when applying the conjugate transposed $Q_{1,i}^{(n,m)^\dagger}$, and part of these NOT gates of $(\tilde{T}_i^{(n,m)})^\dagger$ cancel out with the NOT gates applied in order to realize the next almost generalized multi-Toffoli gate $\tilde{T}_{i+1}^{(n,m)}$. Therefore, it makes sense to devise a function that only applies the NOT gates remaining after the cancelling out. More precisely, this function replaces $g_{i,j}^{b_k}$, and is applied only to implement $\tilde{T}_i^{(n,m)}$, i.e., only before the standard multi-Toffoli gate of $Q_{11,i}^{(n,m)}$, and not after applying the same standard multi-Toffoli gate of $Q_{11,i}^{(n,m)^\dagger}$. This is indeed possible because the operations which are in between $(\tilde{T}_i^{(n,m)})^\dagger$ and $\tilde{T}_{i+1}^{(n,m)}$, namely, the conjugate transposed of the copies $Q_{10}^{(n,m)^\dagger}$ and the copies $Q_{10}^{(n,m)}$ (which by the way simplify each other apart from the

last stage), do not involve the wires on which $(\tilde{T}_i^{(n,m)})^\dagger$ controls.

So, in each $U_i^{(n,m)}$, we do the following modifications: (i) instead of applying $\tilde{T}_i^{(n,m)}$ in $Q_{1,i}^{(n,m)}$ (see Eq. (A7)), we apply the same operation but replacing $g_{i,j}^{b_k}$ given in Eq. (A8) by

$$h_{i,j}^{b_k} := \begin{cases} X & \text{if } i \mod 2^j = 0 \\ I_2 & \text{otherwise} \end{cases} , \tag{D1}$$

which amounts to replacing $Q_{1,i}^{(n,m)}$ by an operation that we call $P_{1,i}^{(n,m)}$; (ii) moreover, instead of applying $(\tilde{T}_i^{(n,m)})^\dagger$, we simply apply the standard $(n-m)$-Toffoli gate $K_{\alpha,b_0'}^{\text{multi}}(X)$, which amounts to replacing ${Q_{1,i}^{(n,m)}}^\dagger$ by an operation that we call $\bar{P}_{1,i}^{(n,m)}$. In total, we have replaced $U_i^{(n,m)}$ by

$$(U_i^{(n,m)})' := \bar{P}_{1,i}^{(n,m)} {Q_2^{(n,m)}}^\dagger Q_{0,i}^{(n,m)} Q_2^{(n,m)} P_{1,i}^{(n,m)} . \tag{D2}$$

Let us notice that $h_{i,j}^{b_k}$ implements the NOT gates as in the naive circuit of Ref. [1]. In Fig. 14, we show how part of the NOT gates of the almost generalized multi-Toffoli gates $(\tilde{T}_i^{(n,m)})^\dagger$ and $\tilde{T}_{i+1}^{(n,m)}$ cancel between each other, and which NOT gates remain, that we encode via $h_{i,j}^{b_k}$.

Appendix E: Pseudo-code

The pseudo-code used to code, with Qiskit, the adjustable-depth quantum circuit, is given below:

Algorithm 1 $Q_{1,i}^{(n,m)}$ and $P_{1,i}^{(n,m)}$

```
Q_10^(n,m):
for j ∈ [0 ... m − 2] do
    for k ∈ [j + 1 ... m − 1] do
        if j = 0 then
            not(s_{l_k}).control(b_k)
        else
            not(s_{l_k + Σ_{u=1}^{j} 2^{u−1}}).control(b_k)
            for p ∈ [0 ... 2^j − 2] do
                not(s_{l_k+p+2^j}).control(s_{l_k+p})
            end for
        end if
    end for
end for

Q_{11,i}^(n,m) and P_{11,i}^(n,m):
α ← {b_k | k ∈ [m ... n − 1]}
for k ∈ [m ... n − 1] do
    if P_{11,i}^(n,m) then
        if i mod 2^{k−m} = 0 then
            not(b_k)
        end if
    else
        if ⌊i/2^{k−m} mod 2⌋ = 0 then
            not(b_k)
        end if
    end if
end for
if m ≠ n then
    not(b'_0).control(α)
else
    not(b'_0)
end if
for j ∈ [0 ... m − 1] do
    swap(b'_0, b'_{2^j}).control(b_j)
    for k ∈ [1 ... 2^j − 1] do
        swap(b'_k, b'_{2^k+2^j}).control(s_{j+k−1})
    end for
end for

P̄_{11,i}^(n,m):
α ← {b_k | k ∈ [m ... n − 1]}
if m ≠ n then
    not(b'_0).control(α)
else
    not(b'_0)
end if
for j ∈ [0 ... m − 1] do
    swap(b'_0, b'_{2^j}).control(b_j)
    for k ∈ [1 ... 2^j − 1] do
        swap(b'_k, b'_{k+2^j}).control(s_{j+k−1})
    end for
end for
```

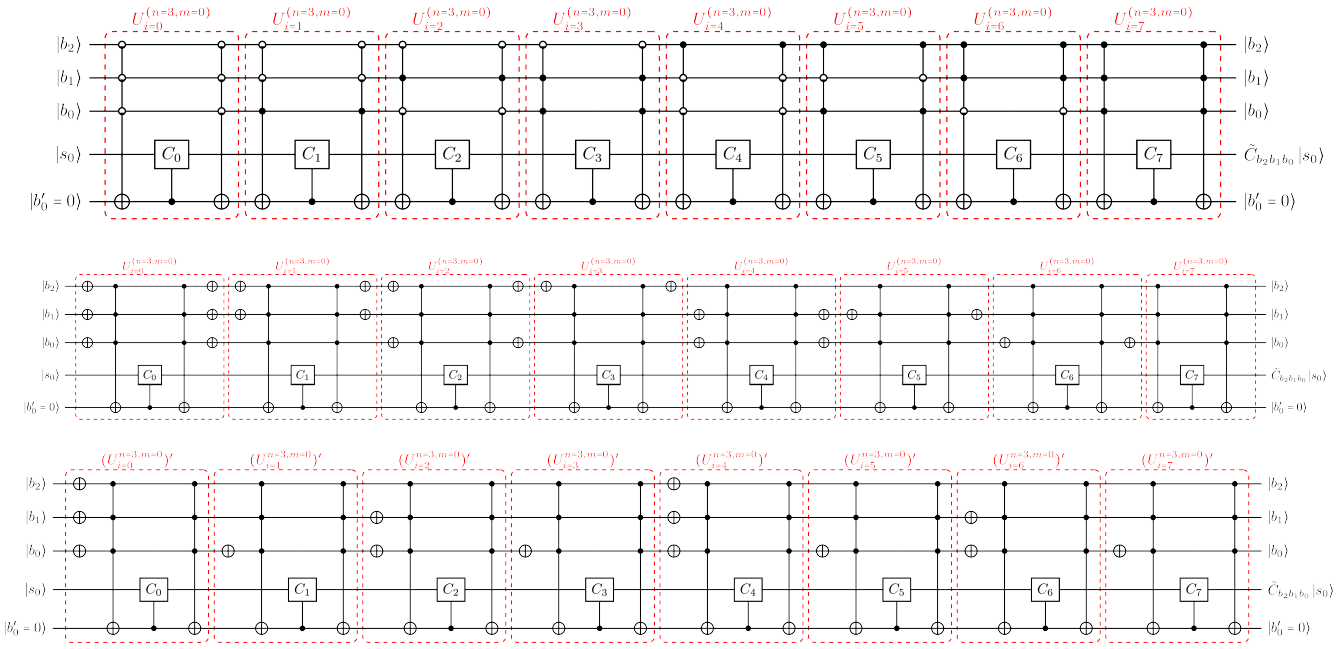

Figure 14: In the top figure, we show $U^{(n=3,m=0)}$. In the middle figure, we replace the generalized multi-Toffoli gates by the $\tilde{T}_i^{(n,m)}$'s, which enables to see that part of the NOT gates of the $(\tilde{T}_i^{(n,m)})^\dagger$'s cancel out with those of the $\tilde{T}_{i+1}^{(n,m)}$'s. In the bottom figure, we show the optimized circuit $(U^{(n,m)})'$, obtained by taking the product of the $(U_i^{(n,m)})'$'s, defined in Eq. (D2).

Algorithm 2 $Q_2^{(n,m)}$

for $j \in [0 \dots m-2]$ **do**
  for $k \in [0 \dots 2^{m-j-1}-2]$ **do**
    $not(b'_{2^m-1-k2^{j+1}}).control(b'_{2^m-1-2^{j+1}(1/2+k)})$
  end for
end for
for $j \in [0 \dots m-1]$ **do**
  for $k \in [0 \dots 2^j-1]$ **do**
    $swap(s_{k2^{m-j}}, s_{(1/2+k)2^{m-j}}).control(b'_{(k+1)2^{m-j}-1})$
  end for
  if $j \neq m-1$ **then**
    for $l \in [0 \dots 2^{j+1}-2]$ **do**
      $not(b'_{2^m-1-l2^{m-j-1}}).control(b'_{2^m-1-2^{m-j-1}(1/2+l)})$
    end for
  end if
end for

Algorithm 3 $Q_{0,i}^{(n,m)}$

for $k \in [0 \dots 2^m-1]$ **do**
  $C_{i2^m+k}(s_k, b'_k)$
end for