Forward and Backward Constrained Bisimulations for Quantum Circuits

A. Jiménez-Pastor[1(✉)], K. G. Larsen[1], M. Tribastone[2], and M. Tschaikowski[1]

[1] Aalborg University, Aalborg, Denmark
{ajpa,kgl,tschaikowski}@cs.aau.dk
[2] IMT Lucca, Lucca, Italy
mirco.tribastone@imtlucca.it

Abstract. Efficient methods for the simulation of quantum circuits on classic computers are crucial for their analysis due to the exponential growth of the problem size with the number of qubits. Here we study lumping methods based on bisimulation, an established class of techniques that has been proven successful for (classic) stochastic and deterministic systems such as Markov chains and ordinary differential equations. Forward constrained bisimulation yields a lower-dimensional model which exactly preserves quantum measurements projected on a linear subspace of interest. Backward constrained bisimulation gives a reduction that is valid on a subspace containing the circuit input, from which the circuit result can be fully recovered. We provide an algorithm to compute the constraint bisimulations yielding coarsest reductions in both cases, using a duality result relating the two notions. As applications, we provide theoretical bounds on the size of the reduced state space for well-known quantum algorithms for search, optimization, and factorization. Using a prototype implementation, we report significant reductions on a set of benchmarks. Furthermore, we show that constraint bisimulation complements state-of-the-art methods for the simulation of quantum circuits based on decision diagrams.

Keywords: bisimulation · quantum circuits · lumpability

1 Introduction

Quantum computers can solve certain problems more efficiently than classic computers. Earlier instances are Grover's quantum search [28] and Shor's factorization [47]; more recent works address the efficient solution of linear equations [30] and the simulation of differential equations [36]. Despite its potential and increasing interest from a commercial viewpoint [41], quantum computing is still in its infancy. The number of qubits of current quantum computers is prohibitively small; furthermore, low coherence times and quantum noise lead to high error rates. Further research and improvement of quantum circuits thus hinges on the availability of efficient simulation algorithms on classic computers.

B. Finkbeiner and L. Kovács (Eds.): TACAS 2024, LNCS 14571, pp. 343–362, 2024.
https://doi.org/10.1007/978-3-031-57249-4_17

Being described by a unitary complex matrix, any quantum circuit can be simulated by means of array structures and the respective matrix operations [32,50,38]. Unfortunately, direct array approaches are subject to the curse of dimensionality [41] because the size of the matrix is exponential in the number of qubits. This motivated the introduction of techniques that try to overcome the exponential growth, resting for instance upon the stabilizer formalism [1], tensor networks [55,56], path sum reductions [3] and decision diagrams [57,41,29].

Here we study bisimulation relations for quantum circuits. Bisimulation has a long tradition in computer science [46]. For the purpose of this paper, the most relevant strand of research on this topic regards bisimulations for quantitative models such as probabilistic bisimulation [34,9]. This is closely related to ordinary lumpability for Markov chains [13], also known as *forward* bisimulation [25], which yields an aggregated Markov chain by means of partitioning the original state space, such that the probability of being in each macro-state/block is equal to the sum of the probabilities of each state in that block. Exact lumpability [13], also known as *backward* bisimulation [49], exploits a specific linear invariant induced by a partition of the state space such that states in the same partition block have the same probability at all time points [13]. In an analogous fashion, forward and backward bisimulations have been developed for chemical reaction networks [16,51,15], rule-based systems [25,24], and ordinary differential equations [14,19].

In all these cases, lumping can be mathematically expressed as a specific linear transformation of the original state space into a reduced one that is induced by a partition. In general, however, lumping allows for arbitrary linear transformation [49,12]. Since this may introduce loss of information, *constrained lumping* allows one to specify a subspace of interest that ought to be preserved in the reduction [53,12,42,31]. In partition-based bisimulations, constraints can be specified as suitable user-defined initial partitions of states for which lumping is computed as their (coarsest) refinement [17,20]. Bisimulation relations for dynamical systems [43,11] and the notions of constrained linear lumping in [53,42,31], instead, can be understood as linear projections (also known as "lumping schemes") into a lower-dimensional system that preserves an arbitrary linear constraint subspace.

The aim of this paper is to boost simulation of quantum circuits via forward- and backward-type bisimulations that can be constrained to subspaces.[3] Analogously to the cited literature, with *forward constrained bisimulation* (FCB) the aim is to obtain a lower-dimensional circuit which (exactly) preserves the behavior of the original circuit on the subset of interest. In *backward constrained bisimulation* (BCB), the reduction is valid on the constraint subspace; in this manner, the original quantum state can be fully recovered from the reduced circuit. Overall, this setting has complementary interpretation with respect to the analysis of a quantum circuit. It is known that a quantum state can only be accessed by means of a quantum measurement, mathematically expressed as

[3] In the following, the simulation of quantum circuits refers to their execution on a classic computer and not to the notion of one-sided bisimulation.

a projection onto a given subspace. FCB, in general, preserves any projection onto the constraint subspace. That is, if the constraint subspace contains the measurement subspace, FCB will exactly preserve the quantum measurement, but the full quantum state cannot be recovered in general. Instead, constraining the invariant set of BCB to contain the input of the circuit ensures that the full circuit result can be recovered from the reduced circuit.

We show that FCB and BCB are related by a duality property stating that a lumping scheme is an FCB if and only if its complex conjugate transpose is a BCB. Interestingly, this is analogous to the duality established between ordinary (forward) and exact (backward) lumpability for Markov chains [20,18], although it does not carry over to other models in general [7,52]. A relevant implication of this result is that one needs only one algorithm to compute both FCB and BCB. As a further contribution of this paper, we present such an algorithm, developed an as adaptation of the CLUE method for the constrained lumping of systems of ordinary differential equations with polynomial right-hand sides [42] of which it inherits the polynomial-time complexity in matrix size.

To show the applicability of our constrained bisimulations, we analyze several case studies for which we report both theoretical and experimental results. Specifically, we first study three classic quantum circuits for search (Grover's algorithm [40, Section 6.2]), optimization [21], and factorization [40, Section 5.3.2], respectively. In Grover's algorithm, the cardinality of the search domain is exponential in the number of qubits; we prove that BCB can always reduce the circuit matrix to a 2×2 matrix while exactly preserving the output of interest. Next, we consider quantum approximate optimization algorithm for solving SAT and MaxCut instances [21]; in this setting, our main theoretical result is an upper bound on the size of the reduced (circuit) matrix by the number of clauses (SAT) or edges (MaxCut). Finally, for quantum factorization we prove that the size of the reduced matrix gives the multiplicative order, that is, solves the order finding problem to which the factorization problem can be reduced [40].

From an experimental viewpoint, using a prototype based on a publicly available implementation of CLUE, we compare the aforementioned theoretical bounds against the actual reductions on a set of randomly generated instances. Moreover, we conduct a large-scale evaluation on common quantum algorithms collected in the repository [44], showing considerable reductions in all cases. Finally, we demonstrate that constrained bisimulation complements state-of-the-art methods for quantum circuit simulation based on decision diagrams [41], as implemented in the tool DDSIM [57].

Further related work. Probabilistic bisimulations [9,5,6] have been considered for quantum extensions of process calculi, see [26,23] and references therein. Similar to their classic counterparts, these seek to identify concurrent (quantum) processes with similar behavior. The current work, instead, is about boosting the simulation of quantum circuits and is in line with [8,18,54]. Specifically, it operates directly over quantum circuits rather than processes and exploits general linear invariants in the (complex) state space. In engineering, invariant-based reductions of linear systems are known under the names of proper orthogonal decom-

position [4,39], Krylov methods [4], and dynamic mode decomposition [45,33,27]. Linear invariants describe also safety properties [10] in quantum model checking [59,58], without being used for reduction though. $\mathcal{L}$-bisimulation [12] and [33,27] yield the same reductions, with the difference being that the former obtains the smallest reduction up to a given initial constraint, while the latter computes the smallest reduction up to an initial condition. While relying similarly to us on reduction techniques, [33,27] focus on the reduction of quantum Hamiltonian dynamics, with applications mostly in quantum physics and chemistry. Instead, we study the reduction of quantum circuits which are the prime citizens of quantum computing. Moreover, we provide a prototype implementation of our approach and perform a large-scale numerical evaluation.

Paper outline. The paper is structured as follows. After a review of core concepts, Section 2 introduces forward and backward constrained bisimulation of (quantum) circuits. There, we also provide an algorithm for the computation of constrained bisimulations by extending [42,35] to circuits. Section 3 then derives bounds on the reduction sizes of quantum search [28], quantum optimization [21] and quantum order finding [40]. Section 4, instead, conducts a large-scale evaluation on published quantum benchmarks [44] and compares constrained bisimulations against DDSIM with respect to the possibility of speeding up circuit simulations. The paper concludes in Section 5.

2 Constrained Bisimulations for Quantum Circuits

Notation. We shall denote by n the number of qubits and set $N = 2^n$ for convenience. Column vectors are denoted by the *ket* notation $|z\rangle$, while the complex conjugate transpose of $|z\rangle$ is denoted by $|z\rangle^\dagger = \langle z|$, i.e., $\langle z| = |\bar{z}\rangle^T$ with $\bar{\cdot}$ and $\cdot^T$ denoting complex conjugation and transpose, respectively. In a similar vein, $\langle z| \, |z\rangle = \langle z|z\rangle$, where $\langle \cdot \mid \cdot \rangle$ is the standard scalar product over $\mathbb{C}^N$. Following standard notation, the canonical basis vectors of $\mathbb{C}^N$ are expressed using tensor products and bit strings $x \in \{0,1\}^n$; specifically, writing $\otimes$ for the Kronecker product, we have $|x_n\rangle \otimes |x_{n-1}\rangle \otimes \ldots \otimes |x_1\rangle = |x_n\rangle \, |x_{n-1}\rangle \ldots |x_1\rangle = |x_n x_{n-1} \ldots x_1\rangle = |d\rangle$, where $0 \leq d \leq 2^n - 1$ is a decimal representation of x, see [40] for details. We usually denote by $|x\rangle$ canonical basis vectors with $x \in \{0,1\}^n$, whereas $|u\rangle, |v\rangle, |w\rangle, |z\rangle \in \mathbb{C}^N$ refer to linear combinations in the form $|z\rangle = \sum_{x\in\{0,1\}^n} c_x \, |x\rangle$ with $c_x \in \mathbb{C}$. For any canonical basis vector, we have $|x\rangle = |\bar{x}\rangle$. To avoid confusion, forward constrained bisimulations (FCB) are denoted by row matrices $L \in \mathbb{C}^{d\times N}$ with $d \leq N$, while backward constrained bisimulations (BCB) are denoted by column matrices $L^\dagger \in \mathbb{C}^{N\times d}$.

Preliminaries. We begin by introducing core concepts from linear algebra and quantum computing [37,40].

Definition 1 (Core Concepts).

- *The column space of a matrix M are all linear combinations of its columns and is denoted by $\langle M\rangle_c$. One says, the columns of M span $\langle M\rangle_c$.*

- *The row space of a matrix is the set of all linear combinations of its rows and is denoted by* $\langle M \rangle_r$. *One says, the rows of* M *span* $\langle M \rangle_r$.
- *A (quantum) circuit over* n *qubits is described by a unitary map* $U \in \mathbb{C}^{N \times N}$, *that is,* $U^{-1} = U^\dagger$.
- *A (quantum) state* $|z\rangle \in \mathbb{C}^N$ *is a vector with (Euclidian) norm one.*
- *A matrix* $P \in \mathbb{C}^{N \times N}$ *is an orthogonal projection if* $P \circ P = P = P^\dagger$.
- *Any vector* $|z\rangle \in \mathbb{C}^N$ *generates the linear subspace* $S_{|z\rangle} = \langle |z\rangle \rangle_c$.

Throughout the paper, we do not work at the higher level where quantum circuits are defined by means of a quantum gate compositions [40]. Instead, we work directly at the level of the unitary maps that are induced by such compositions. With this in mind, we use the terms "unitary map" and "quantum circuit" interchangeably.

We distinguish between one- and multi-step applications of a quantum circuit [40]. For an input state $|w_0\rangle \in \mathbb{C}^N$, the full quantum state after *one-step* application is $U|w_0\rangle$. Instead, the full quantum state after a *multi-step* application is given by $U^k|w_0\rangle$, where $k > 1$ is the number of steps.

These definitions justify interpreting a quantum circuit as a discrete-time dynamical system as follows.

Definition 2 (Dynamical System). *A circuit* $U \in \mathbb{C}^{N \times N}$ *with input state* $|w_0\rangle$ *induces the discrete time dynamical system (DS)* $|w_{k+1}\rangle = U|w_k\rangle$, *with* $k \geq 0$. *We call* $|w_k\rangle$ *the* full quantum state *at step* k.

Example 1 *The one-qubit circuit* $U = \left(\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right)$ *is known as the Pauli X-gate [40]. In the case of* $k \geq 1$ *steps and input* $|w_0\rangle = |\phi\rangle$, *where* $|\phi\rangle = (1, -1)/\sqrt{2}$, *the induced DS can be shown to be* $|w_k\rangle = (-1)^k |\phi\rangle$.

The result of a quantum computation is not directly accessible and is usually queried using quantum measurements [40]. These can be described by projective measurements [40], formally given by a family of orthogonal projections $\{P_1, \ldots, P_m\}$ satisfying $P_1 + \ldots + P_m = I$. When a quantum state $|z\rangle \in \mathbb{C}^N$ is measured, the probability of outcome $1 \leq i \leq m$ is $\pi_i = \langle z| P_i |z\rangle$. In case of outcome i, the quantum state after the measurement is $P_i|z\rangle / \sqrt{\pi_i}$. We will be mostly concerned with the case $\{P, I - P\}$ for a given orthogonal projection P.

Often, one is interested in querying states from a specific subspace S. For instance, the result of the HHL algorithm [30], considered in Section 4, is stored in a subset of all qubits, i.e., in a subspace. To this end, it suffices to use a projective measurement identifying S.

Definition 3. *Given an orthogonal projection* P, *we call* $P|z\rangle$ *the* P*-measurement of* $|z\rangle$. *A subspace* $S \subseteq \mathbb{C}^N$ *is identifiable by* P *if* $P|z\rangle = |z\rangle$ *for all* $|z\rangle \in S$.

A particularly simple yet useful class of projective measurements are those that measure a single state $|w\rangle$, i.e., identify the space $S_{|w\rangle}$ spanned by $|w\rangle$. This is given by the orthogonal projection $P_{|w\rangle} := |w\rangle\langle w|$.

Example 2 *Assume that we are interested in measuring the result of Example 1 using measurement* $P_{|\phi\rangle}$ *that identifies* $S_{|\phi\rangle}$. *Then, for* $|w_0\rangle = \binom{1}{0}$, *it holds that* $P_{|\phi\rangle}|w_k\rangle = (-1)^k |\phi\rangle / \sqrt{2}$.

Forward Constrained Bisimulation. We next introduce FCB.

Definition 4 (Forward Constrained Bisimulation, FCB). *Fix a circuit defined by $U \in \mathbb{C}^{N\times N}$ with initial state $|w_0\rangle$ and a matrix $L \in \mathbb{C}^{d\times N}$ with orthonormal rows.*

a) The DS reduced by L is given by $|\hat{w}_{k+1}\rangle = \hat{U}\,|\hat{w}_k\rangle$, where $\hat{U} = LUL^\dagger$, and initial state $|\hat{w}_0\rangle = L\,|w_0\rangle$.
b) L is called forward constrained bisimulation of DS $|w_{k+1}\rangle = U\,|w_k\rangle$ wrt constraint subspace $S \subseteq \mathbb{C}^N$ when $S \subseteq \langle L^\dagger\rangle_c$ and $L\,|w_k\rangle = |\hat{w}_k\rangle$ for all $k \geq 1$.

Before commenting on the definition, we establish the following.

Lemma 1. *The reduced map $\hat{U}$ in Definition 4 is unitary.*

Proof. See proof of Theorem 2.

We remark that the reduction holds for any choice of initial state $|w_0\rangle$, analogously to the aforementioned forward-type bisimulations [14,18] for (real-valued) dynamical systems. The assumption of orthonormality of rows of L implies that $d \leq N$, i.e., L is a transformation onto a possibly smaller-dimensional state space. Although it can be dropped without loss of generality [42], it allows for a more immediate relation to projective measurements. Indeed, matrix L induces the orthogonal projection P_L defined as $P_L = L^\dagger L$. This projective measurement identifies S because $S \subseteq \langle L^\dagger\rangle_c$. Moreover, $P_L\,|w_k\rangle$ is preserved in the reduced system for any k. To see this, it suffices to multiply $L\,|w_k\rangle = |\hat{w}_k\rangle$ by $L^\dagger$ from the left and to note that this yields $P_L\,|w_k\rangle = L^\dagger\,|\hat{w}_k\rangle$.

Example 3 *Continuing Example 1, it can be shown that the 2×1 matrix $L = |\phi\rangle^\dagger = (1,-1)/\sqrt{2}$ is an FCB wrt $S_{|\phi\rangle}$. Indeed, since $\hat{U} = LUL^\dagger = -1$, we obtain $|\hat{w}_{k+1}\rangle = -\,|\hat{w}_k\rangle$, while a direct calculation confirms that $L\,|w_0\rangle = |\hat{w}_0\rangle$ implies $L\,|w_k\rangle = |\hat{w}_k\rangle$ for all $k > 0$. Multiplying both sides by $L^\dagger$ from the left yields $L^\dagger LU^k\,|w_0\rangle = (-1)^k L^\dagger L\,|w_0\rangle$. Consequently, the $P_{|\phi\rangle}$-measurement of the original map can be obtained from the $P_{|\phi\rangle}$-measurement of the reduced map.*

Algorithm 1 adapts the algorithm for (real-valued) systems of ordinary differential equations with polynomial derivatives developed in [42,35] to the complex domain and yields the minimal FCB wrt subspace S, i.e., it returns an orthonormal $L \in \mathbb{C}^{d\times N}$ whose dimension d is minimal.

Theorem 1 (Minimal FCB). *Algorithm 1 computes a minimal FCB $L \in \mathbb{C}^{d\times N}$ wrt subspace S, i.e., the rowspace of any FCB L' wrt S contains that of L. The complexity of Algorithm 1 is polynomial in N.*

Proof. See proof of Theorem 2.

We briefly comment on Algorithm 1. The idea behind it exploits that L can be shown to be an FCB whenever $L^\dagger$ is an invariant set of the map U, that is, if the column space of $L^\dagger U$ is contained in the column space of $L^\dagger$. The

Algorithm 1 Computation of an FCB L wrt subspace S

Require: Unitary map $U \in \mathbb{C}^{N\times N}$ and subspace $S \subseteq \mathbb{C}^N$.
1: **compute** orthonormal basis of S, store it in column matrix $L^\dagger \in \mathbb{C}^{N\times d_0}$
2: **repeat**
3: **for all** columns $|z\rangle$ of $L^\dagger$ **do**
4: **compute** $|\pi\rangle = P_L U |z\rangle$
5: **if** $|\pi\rangle \neq U|z\rangle$ **then**
6: $|w\rangle = U|z\rangle - |\pi\rangle$
7: **append** column $|w\rangle / \langle w|w\rangle$ to $L^\dagger$
8: **end if**
9: **end for**
10: **until** no columns have been appended to $L^\dagger$
11: **return** matrix $L^{\dagger\dagger}$.

algorithm begins by initializing $L^\dagger$ with a basis of S in line 1. This ensures that S is contained in the column space of the final result. For every column $|z\rangle$ of $L^\dagger$, the main loop in line 2 checks whether $U|z\rangle$ is in the column space of $L^\dagger$ (line 5) by computing its projection $|\pi\rangle$ onto the column space of $L^\dagger$ (line 4). If it is not in the column space, the projection will differ from $U|z\rangle$ and the residual $|w\rangle$ must be added to $L^\dagger$. This shows the correctness, while the minimality of FCB L follows from the fact that only the necessary residuals are being added to $L^\dagger$. The complexity of the algorithm, instead, follows by noting that at most N columns can be added to $L^\dagger$ and that all computations of the main loop require, similarly the computation in line 1, at most $\mathcal{O}(N^3)$ operations.

Remark 1. As can be noticed in Algorithm 1, e.g., line 4, the computation of an FCB subsumes the computation of a single step of the circuit. For practical applications to single-step circuits where the modeler is interested in only a single input, FCB may be as expensive as simulating the original circuit directly. Hence, it is obvious that that the effectiveness of constrained bisimulations is particularly relevant when simulating the circuit with respect to several inputs, or when considering multi-step applications. Examples of this are provided in Section 3 and a numerical evaluation is carried out in Section 4.

Example 4 *Consider the FCB $L = |\phi\rangle^\dagger$ wrt subspace $S_{|\phi\rangle}$ from Example 3. Then, noting that $(I - P_L)|\phi\rangle = 0$, we infer that Algorithm 1 terminates in Line 5. Hence, L is a minimal FCB wrt $S_{|\phi\rangle}$.*

Backward Constrained Bisimulation. BCB yields a reduced system through the identification of an invariant set, i.e., a subspace S such that $U^k|z\rangle \in S$ for any $|z\rangle \in S$ and $k \geq 1$. Whereas in FCB the reduced model can recover projective measurements onto the constraint set *for any* initial set, here one can recover the full quantum state, so long as the initial states belong to the invariant set.

Definition 5 (Backward Constrained Bisimulation, BCB). *Let U, L and $\hat{U}$ be as in Definition 4. Then, $L^\dagger$ is a BCB of the dynamical system $|w_{k+1}\rangle =$*

$U|w_k\rangle$ wrt a subspace of inputs $S \subseteq \mathbb{C}^N$ when $S \subseteq \langle L^\dagger \rangle_c$ and whenever $|w_0\rangle = L^\dagger|\hat{w}_0\rangle$ implies $|w_k\rangle = L^\dagger|\hat{w}_k\rangle$ for all $k \geq 1$.

Similarly to FCB, we assume without loss of generality that $L \in \mathbb{C}^{d\times N}$ has orthonormal rows. As anticipated above, FCB and BCB are not comparable in general. Indeed, an FCB L makes no assumption on the initial condition $|w_0\rangle$, while a BCB $L^\dagger$ does so by requiring $L^\dagger L|w_0\rangle = |w_0\rangle$. Conversely, a BCB $L^\dagger$ allows one to obtain $|w_k\rangle$, while an FCB L allows to obtain $L|w_k\rangle$ instead of $|w_k\rangle$ itself.

Example 5 *Fix $|\phi\rangle = (1,-1)^T/\sqrt{2}$ from Example 4 and recall that $L = |\phi\rangle^\dagger$, $U|\phi\rangle = -|\phi\rangle$ and $\hat{U} = -1$. Then, $L^\dagger$ is a BCB of U wrt $S_{|\phi\rangle}$. Indeed, $L^\dagger L|w_0\rangle = |w_0\rangle$ implies $|w_0\rangle = |\phi\rangle$, while*

$$L^\dagger|\hat{w}_k\rangle = (-1)^k L^\dagger|\hat{w}_0\rangle = (-1)^k L^\dagger L|w_0\rangle = (-1)^k|w_0\rangle = U^k|\phi\rangle = |w_k\rangle\,.$$

Example 5 anticipates the next result that states FCB and BCB are dual notions. This generalizes the known duality of ordinary and exact lumpability of Markov chains [20,18].

Theorem 2 (Duality). *Fix a unitary map $U \in \mathbb{C}^{N\times N}$ and a subspace $S \subseteq \mathbb{C}^N$. L is an FCB wrt S if and only if $L^\dagger$ is a BCB wrt S.*

Proof. Let $S_0 \subseteq S$ be a basis of some fixed $S \subseteq \mathbb{C}^N$. We first note that the discussion of [42,31,35] and [45,4] can be extended to the complex field in a direct manner. With this, we obtain:

1. $L \in \mathbb{C}^{d\times N}$ is an FCB wrt S if and only if $\langle LU\rangle_r \subseteq \langle L\rangle_r$ with $\langle S_0^\dagger\rangle_r \subseteq \langle L\rangle_r$.
2. $D \in \mathbb{C}^{N\times d}$ is a BCB wrt S if and only if $\langle UD\rangle_c \subseteq \langle D\rangle_c$ with $\langle S_0\rangle_c \subseteq \langle D\rangle_c$.

Moreover, we observe the following:

$$\begin{aligned}
\langle LU\rangle_r \subseteq \langle L\rangle_r &\Leftrightarrow [U \text{ bijection}]\\
\langle LU\rangle_r = \langle L\rangle_r &\Leftrightarrow [\text{daggering}]\\
\langle U^\dagger L^\dagger\rangle_c = \langle L^\dagger\rangle_c &\Leftrightarrow [U \text{ unitary}]\\
\langle U^{-1} L^\dagger\rangle_c = \langle L^\dagger\rangle_c &\Leftrightarrow [U \text{ bijection}]\\
\langle L^\dagger\rangle_c = \langle UL^\dagger\rangle_c &\Leftrightarrow [U \text{ bijection}]\\
\langle UL^\dagger\rangle_c \subseteq \langle L^\dagger\rangle_c &
\end{aligned}$$

This yields Theorem 2, i.e., $L \in \mathbb{C}^{d\times N}$ is an FCB of U wrt constraint S if and only if $L^\dagger \in \mathbb{C}^{N\times d}$ is a BCB of U wrt S (because $S_0^{\dagger\dagger} = S_0$). Moreover, if $L^\dagger$ is computed by Algorithm 1, then $L^\dagger$ is a BCB wrt S, while $L^{\dagger\dagger}$ is an FCB wrt S. This follows by noting that in such a case $L^\dagger \in \mathbb{C}^{N\times d}$ satisfies

$$\begin{aligned}
\langle L^\dagger\rangle_c &= \langle U^k|z\rangle \mid 0 \leq k \leq N-1, |z\rangle \in S\rangle_c\\
&= \langle U^k|z\rangle \mid 0 \leq k \leq N-1, |z\rangle \in S_0\rangle_c
\end{aligned}$$

The complexity follows from the discussion after Theorem 1. A detailed complexity discussion can be obtained in [42]. Exploiting that an FCB L satisfies $LUL^\dagger L = LU$ by [53], we obtain

$$(LUL^\dagger)^\dagger(LUL^\dagger) = (LU^\dagger L^\dagger)(LUL^\dagger) = LU^\dagger UL^\dagger = LL^\dagger = I_{d\times d},$$

showing that $\hat{U}$ is unitary. □

In light of the above result, we often speak of a (constrained bisimulation) reduction. Moreover, we note that Theorem 2 ensures that a BCB reduction up to input yields an FCB reduction up result, a discussed next.

Remark 2. Let $L^\dagger$ be the BCB of U wrt $S_{|w_0\rangle}$, where $|w_0\rangle$ is the input. Then, $L = L^{\dagger\dagger}$ is an FCB wrt $S_{|w_0\rangle}$, implying that $P_L = L^\dagger L$ identifies the column space of $L^\dagger$, see discussion after Definition 4. At the same time, result $U^k |w_0\rangle$ is in the column span of $L^\dagger$ because $U^k |w_0\rangle = |w_k\rangle = L^\dagger |\hat{w}_k\rangle = L^\dagger \hat{U} L |w_0\rangle$.

We end the section by pointing out that, thanks to Theorem 2, Algorithm 1 can be used to compute a minimal BCB $L^\dagger$ wrt subspace S. Indeed, the only difference is that one should return $L^\dagger$ rather than $L^{\dagger\dagger}$ in the last line of the algorithm. With this change, we notice that Algorithm 1 coincides, in the case of a one dimensional subspace $S \subseteq \mathbb{C}^N$, with the Krylov subspace [4] that can be obtained by the Arnoldi iteration [45].

3 Applications

In this section we demonstrate that established quantum algorithms enjoy substantial bisimulation reductions. For each application, we provide a brief description of the quantum algorithm and a theoretical bound on its reduction.

3.1 Quantum Search

Let us assume we are given a non-zero function $f : \{0,1\}^n \to \{0,1\}$ and that we are asked to find some $x \in \{0,1\}^n$ such that $f(x) = 1$. Grover's seminal algorithm describes how this can be achieved in $\mathcal{O}(\sqrt{N})$ steps on a quantum computer [40, Section 6.2], thus yielding a quadratic speed-up over a classic computer. For any $x \in \{0,1\}^n$, the Grover map is given by

$$G |x\rangle = (-1)^{f(x)}(I - 2 |\psi\rangle \langle\psi|) |x\rangle , \quad \text{with} \quad |\psi\rangle = \frac{1}{\sqrt{N}} \sum_{x=0}^{N-1} |x\rangle \tag{1}$$

The Grover map yields the following celebrated result.

Theorem 3 (Quantum Search [40]). *Map G is unitary. Moreover, if the number of sought solutions $M = |\{x \mid f(x) = 1\}|$ satisfies $M \leq N/2$, then measuring $G^\kappa |\psi\rangle$ for $\kappa = \lceil \frac{\pi}{4}\sqrt{N/M} \rceil$ yields a state $|x\rangle$ satisfying $f(x) = 1$ with probability at least $\frac{1}{2}$.*

The next result allows one to compute result $G^\kappa |\psi\rangle$ from Theorem 3 using a map over a single qubit.

Theorem 4 (Reduced Grover). *The BCB $L^\dagger \in \mathbb{C}^{N\times d}$ of G wrt $S_{|\psi\rangle}$ has dimension $d = 2$ and a column space spanned by $|\psi\rangle$ and $G|\psi\rangle$.*

Proof. The claim follows by noting that the column space of an BCB wrt $S_{|\psi\rangle}$ is spanned by $|\psi\rangle, G|\psi\rangle, G^2|\psi\rangle, \ldots, G^{N-1}|\psi\rangle$ and so on. This, in turn, is known to have as basis [40, Section 6.2]

$$|\alpha\rangle = \frac{1}{\sqrt{M}} \sum_{x:f(x)=1} |x\rangle \qquad \text{and} \qquad |\beta\rangle = \frac{1}{\sqrt{N-M}} \sum_{x:f(x)=0} |x\rangle,$$

where M is as above, while $|\alpha\rangle$ is the superposition (i.e., sum) of all solution states and $|\beta\rangle$ is the superposition of all non-solution states. □

Remark 3. While the BCB $L^\dagger$ wrt $S_{|\psi\rangle}$ has always dimension 2, its column space depends on the oracle function f. This is because f appears in G, see (1).

3.2 Quantum Optimization

Quantum approximate optimization algorithm (QAOA) [21] is a computational model that has the same expressive power as the common quantum circuit model [22,21,2]. It is described by two matrices. The first one is the *problem Hamiltonian* H_P for which we are interested to compute a maximal eigenstate, i.e., an eigenvector for a maximal eigenvalue of H_P. The second is the *begin Hamiltonian* H_B for which a maximal eigenstate $|\psi\rangle$ is known already. With this, a maximal eigenstate of H_P can be obtained by conducting the QAOA introduced next.

Definition 6 (QAOA [21]). *For a problem Hamiltonian H_P and a begin Hamiltonian H_B, fix the unitary matrices*

$$U_B(\delta) = \exp(-i\delta H_B) \qquad \text{and} \qquad U_P(\delta) = \exp(-i\delta H_P)$$

where $\delta > 0$ is a sufficiently small time step and $\exp(A)$ is the matrix exponential. For a sequence of natural numbers $(k_i, l_i)_{i=1}^{\kappa}$ of length $\kappa \geq 1$, we define

$$|w_\kappa\rangle = U_B(\delta)^{k_\kappa} U_P(\delta)^{l_\kappa} \cdot \ldots \cdot U_B(\delta)^{k_1} U_P(\delta)^{l_1} |\psi\rangle \tag{2}$$

The QAOA with $\kappa \geq 1$ stages is then given by $\max\{\langle w_\kappa| H_P |w_\kappa\rangle \mid (k_i, l_i)_{i=1}^{\kappa}\}$.

While the problem Hamiltonian H_P depends on the task or problem we are solving, the choice of the begin Hamiltonian H_B is informed by the so-called adiabatic theorem, a result that identifies conditions QAOA returns a global optimum. A common heuristic is to pick H_B such that H_B and H_P do not diagonalize over a common basis [22,21] and to assume without loss of generality that $|\psi\rangle = \sum_x |x\rangle / \sqrt{N}$ is the unique maximal eigenvector of H_B.

We next demonstrate bisimulation can be reduce QAOA when it is applied to SAT and MaxCut, two NP-complete problems [48]. We start by introducing the problem Hamiltonians H_P for both cases.

Definition 7 (SAT and MaxCut Problem Hamiltonians).

- *For a boolean formula* $\phi = \bigwedge_{i=1}^{M} C_i$*, where* C_i *is a clause over* n *boolean variables, the problem Hamiltonian is given by* $H_P = \sum_i H_i$*, where*

$$H_i \left|x\right\rangle = \begin{cases} \left|x\right\rangle & , \ C_i(x) \text{ is true} \\ 0 & , \ C_i(x) \text{ is false} \end{cases}$$

  for any $x \in \{0,1\}^n$ *representing a boolean assignment.*
- *For an undirected unweighted graph* $G = (V, E)$ *with vertices* $V = \{1, \ldots, n\}$ *and edges* $E \subseteq V \times V$*, we define the problem Hamiltonian* $H_P = \sum_{(i,j)\in E} H_{i,j}$*, where*

$$H_{i,j} \left|x\right\rangle = \begin{cases} \left|x\right\rangle & , \ x_i \neq x_j \\ 0 & , \ x_i = x_j \end{cases}$$

  for any $x \in \{0,1\}^n$ *that represents a cut* $C \subseteq \{1, \ldots, n\}$ *by setting* $i \in C$ *if and only if* $x_{i-1} = 1$*.*

Following this definition, it can be shown that the QAOA $\left\langle w_\kappa\right| H_P \left|w_\kappa\right\rangle$ from Definition 6 corresponds to a quantum measurement reporting either the expected number of satisfied clauses or the expected size of the cut. It is possible to guarantee that QAOA finds a global optimum for a sufficiently high κ [22,21].

The next result ensures that H_P has BCB $L^\dagger$ wrt $S_{\left|\psi\right\rangle}$ whose reduced map is provably small. Moreover, for any such L, it ensures that there exists a begin Hamiltonian H_B for which $L^\dagger$ is a BCB too, thus ensuring substantial reductions of the entire QAOA calculation (2).

Theorem 5 (Reduced QAOA). *Fix* H_P *as in Definition 7, any* $\delta > 0$ *and let* $L^\dagger \in \mathbb{C}^{N\times d}$ *be a BCB of* $U_P(\delta)$ *wrt* $S_{\left|\psi\right\rangle}$*. Then*

1. *The column space of* $L^\dagger$ *is spanned by*

$$\left(\left|\psi\right\rangle, U_P(\delta)\left|\psi\right\rangle, U_P^2(\delta)\left|\psi\right\rangle, \ldots, U_P^{M-1}(\delta)\left|\psi\right\rangle \right)^\dagger, \tag{3}$$

   where M *is the number of clauses (SAT) or edges (MaxCUT). Specifically, the dimension of the BCB* d *is bounded by* M*.*
2. *Then, for any Hamiltonian* $\hat{H}_B \in \mathbb{C}^{d\times d}$ *(i.e., Hermitian matrix), there is a Hamiltonian* $H_B \in \mathbb{C}^{N\times N}$ *such that*
   - $L^\dagger$ *is a BCB of* $U_B(\delta) = \exp(-i\delta H_B)$ *wrt* $S_{\left|\psi\right\rangle}$*, while its reduced map is* $\hat{U}_B(\delta) = \exp\left(-i\delta \hat{H}_B\right)$
   - *The computation (2) satisfies*

$$\begin{aligned} \left|w_\kappa\right\rangle &= U_B(\delta)^{k_\kappa} U_P(\delta)^{l_\kappa} \cdot \ldots \cdot U_B(\delta)^{k_1} U_P(\delta)^{l_1} \left|\psi\right\rangle \\ &= L^\dagger \hat{U}_B(\delta)^{k_\kappa} \hat{U}_P(\delta)^{l_\kappa} \cdot \ldots \cdot \hat{U}_B(\delta)^{k_1} \hat{U}_P(\delta)^{l_1} L \left|\psi\right\rangle \end{aligned} \tag{4}$$

   The QAOA in $\mathbb{C}^N$ *thus corresponds to a QAOA in the reduced space* $\mathbb{C}^d$*.*

Proof. We begin by proving 1. For SAT, it can be noticed that $H_P |x\rangle = \nu |x\rangle$, where $0 \leq \nu \leq M$ is the number of clauses that are satisfied by assignment x. A similar formula holds for MaxCut, with the difference being that ν is the size of the cut x. It is worth noting that H_P is in diagonal form for both SAT and MaxCut. If m denotes the number of distinct eigenvalues of H_P, then $m \leq M$, where M is in the case of SAT or MaxCUT, respectively, the number of clauses or edges. The same can be said concerning its matrix exponential $U_P(\delta)$ which, being unitary, enjoys an eigendecomposition, allowing us to write $|\psi\rangle = \sum_{i=1}^{m} c_i |z_i\rangle$, where $|z_i\rangle$ is an eigenvector for eigenvalue λ_i of $U_P(\delta)$. This yields

$$U^k |\psi\rangle = \sum_{i=1}^{m} c_i \lambda_i^k |z_i\rangle$$

for all $k \geq 0$. Without lost of generality, consider $d \leq m$ such that $c_k = 0$ for all $k > d$ and $c_k \neq 0$ otherwise. Writing vectors $\{U^k |\psi\rangle \mid 0 \leq k \leq m-1\}$ in basis $|z_1\rangle, \ldots, |z_d\rangle$ gives rise to a regular Vandermonde matrix [45] in $\mathbb{C}^{d\times d}$. This shows that $\{U^k |\psi\rangle \mid d \leq k \leq M-1\}$ are linear combinations of $\{U^k |\psi\rangle \mid 0 \leq k \leq d-1\}$, completing the proof of 1. Instead, 2. follows from the definition of BCB and Lemma 2 from below. □

The auxiliary result below is needed in the proof of Theorem 5.

Lemma 2. *Pick any $L \in \mathbb{C}^{d\times N}$ and $Q \in \mathbb{C}^{(N-d)\times N}$ so that the rows of L and Q comprise an orthonormal basis of $\mathbb{C}^N$, and define*

$$U_B = L^\dagger \hat{U}_B L + Q^\dagger \tilde{U}_B Q, \qquad \hat{U}_B = \exp\Big(-i\delta \hat{H}_B\Big), \qquad \tilde{U}_B = \exp\Big(-i\delta \tilde{H}_B\Big)$$

for any Hamiltonian $\hat{H}_B \in \mathbb{C}^{d\times d}$ and $\tilde{H}_B \in \mathbb{C}^{(N-d)\times(N-d)}$. Then, U_B is unitary, L is an FCB of it wrt $S_{|\psi\rangle}$, and $\hat{U}_B$ is its reduced map. Further, there exists a Hamiltonian $H_B \in \mathbb{C}^{N\times N}$ satisfying $U_B = \exp(-i\delta H_B)$.

Proof. We first show that $LU_B = LU_B L^\dagger L$ as this implies that L is an FCB of U by [53]. To see this, we note that

$$LU_B L^\dagger L = L\big(L^\dagger \hat{U}_B L + Q^\dagger \tilde{U}_B Q\big) L^\dagger L = LL^\dagger \hat{U}_B LL^\dagger L + LQ^\dagger \tilde{U}_B QL^\dagger L = \hat{U}_B L$$
$$LU_B = L\big(L^\dagger \hat{U}_B L + Q^\dagger \tilde{U}_B Q\big) = LL^\dagger \hat{U}_B L + LQ^\dagger \tilde{U}_B Q = \hat{U}_B L$$

where we have used that $LL^\dagger = 0$ and $LQ^\dagger = 0$, which follows from the choice of Q. From the calculation, we can also infer that $\hat{U}_B = LU_B L^\dagger$, i.e., $\hat{U}_B$ in indeed the reduced map. In a similar fashion, one can note that Q is also an FCB of U_B and that $\tilde{U}_B$ is the respective reduced map. Since both $\hat{U}_B$ and $\tilde{U}_B$ are unitary, we infer that also U_B is unitary (alternatively, a direct calculation yields $I = U_B U_B^\dagger$). Since any unitary matrix can be written as a matrix exponential of a Hamiltonian, there exists a Hamiltonian H_B satisfying $U_B = \exp(-i\delta H_B)$. □

3.3 Quantum Factorization and Order Finding

Let us assume that we are given a composite number N which we seek to factorize. As argued in [40, Section 5.3.2], this problem can be solved in randomized polynomial time, provided the same holds true for the order finding problem. Given some randomly picked $x \in \{2, 3 \ldots, N-1\}$, the latter asks to compute the multiplicative order of x modulo N, i.e., the smallest $r \geq 1$ satisfying $x^r \bmod N = 1$. Following [40, Section 5.3.1], we consider the quantum algorithm defined by the unitary map

$$U\,|y\rangle = \begin{cases} |xy \mod N\rangle & ,\ 0 \leq y < N \\ |y\rangle & ,\ N \leq y < 2^l \end{cases}$$

Here, $l \geq 1$ is the smallest number satisfying $N \leq 2^l$.

The next result allows us to relate the order of x to the dimension of the BCB wrt $S_{|1\rangle}$. This fact is exploited in Shor's factorization algorithm [40].

Theorem 6 (Reduced Order Finding). *The dimension of the BCB of U wrt $S_{|1\rangle}$ coincides with the order of x modulo N.*

Proof. If can be shown [40] that the U from above is unitary and that $U\,|u_s\rangle = e^{2\pi i s/r}\,|u_s\rangle$ for all $0 \leq s \leq r-1$, where

$$|u_s\rangle = \frac{1}{\sqrt{r}} \sum_{k=0}^{r-1} \exp\left[\frac{-2\pi i s k}{r}\right] |x^k \mod N\rangle \quad \text{and} \quad \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} |u_s\rangle = |1\rangle\,.$$

With this, $U^k\,|1\rangle = \frac{1}{\sqrt{r}} \sum_{s=0}^{r-1} (e^{2\pi i s/r})^k u_s$ for any $p \geq 0$. Hence, the minimal BCB with respect to $S_{|1\rangle}$ is contained in the span of $u_0, \ldots, u_{r-1}$. To see that the dimension is exactly r, we note that vectors $\{U^k\,|\psi\rangle \mid 0 \leq k \leq r-1\}$, written in basis $u_0, \ldots, u_{r-1}$, constitute a regular Vandermonde matrix [45] in $\mathbb{C}^{r \times r}$. □

4 Numerical Experiments

We evaluate our approach on the applications from Section 3 and the quantum benchmark repository [44]. The approach has been implemented in Python by extending the publicly available implementation of the CLUE algorithm from [42,35]. The prototype is accessible at https://www.doi.org/10.5281/zenodo.8431443. All results reported were executed on a machine with i7-8665U CPU, 32GB RAM and 1024GB SSD. The reduced circuits of CLUE were simulated using Python's `numpy` libraries. All simulations using quantum circuits were done with `qiskit` 0.44.1 and DDSIM simulations were performed with `mqt.ddsim` version 1.19.0. All libraries are available using the default `pip` command in Python.

In our prototype, we have implemented Algorithm 1 by changing in CLUE the domain of definition from the real numbers to complex numbers and, instead of using Gaussian elimination [42,31] for deciding membership properties, we used orthogonal projections.

4.1 Applications from Section 3

Here we report the results of numerical experiments on the applications discussed in Section 3 and a comparison against DDSIM. For this, we considered circuits ranging from 5 to 15 qubits and fixed a timeout of 500 seconds. To allow for a representative evaluation, we averaged runtimes over 5 independent runs of Grover's circuit; instead, in case of quantum optimization, we averaged over 50 independent runs because SAT formulas and graphs were picked randomly in each run. Specifically, the instances were generated as follows:

- *Grover algorithm* (Sec. 3.1): following the convention of [44], we set up a search function f where $f(x) = 0$ for all $x \in \{0,1\}^n$ except for $f(11...1) = 1$. This can be realized via an oracle using the Toffoli gate.
- *Quantum Optimization for SAT* (Sec. 3.2): for each number of qubits n, we generate a random formula with m clauses (m is randomly picked between n and $3n$), where each clause has 3 variables at most. We guarantee that every formula contains all n variables.
- *Quantum Optimization for MaxCut* (Sec. 3.2): for each number of qubits n, we generate an Erdős-Rényi graph with n nodes and edge probability $\frac{1}{3}$.

For Grover's algorithm the experiments confirmed the two-dimensional bisimulation reduction theoretically demonstrated in Theorem 4. Instead, for quantum optimization we measured the (average) achieved reduction against the theoretical bounds developed in Theorem 5. For the comparison against DDSIM, we measured DDSIM's wallclock execution time for each circuit instance against CLUE's corresponding end-to-end runtime consisting of both computing the constrained bisimulation and simulating the reduced circuit. For quantum optimization, the number of steps κ was set to the smallest integer greater or equal to $\sqrt{N}$. The choice of κ is motivated by Theorem 3 and the discussion around the so-called adiabatic theorem in [22,21].

Discussion. For quantum optimization, Table 1 reports logarithmic CLUE reductions, reducing in particular $2^{15} \times 2^{15}$ matrices to 15×15 matrices in less than 4 s. Overall, DDSIM was faster than CLUE in case of Grover, while CLUE outperformed DDSIM on quantum optimization. We explain this by the diagonal form of the quantum optimization circuit. The results for quantum optimization and Grover confirm the observation made in Remark 1 that CLUE reductions may be practically useful in multi-step applications.

4.2 Benchmark Circuits

In this section, we report a numerical evaluation of the quantum benchmarks from [44], available at https://www.cda.cit.tum.de/mqtbench/. For each number $0 \leq x \leq N-1$, we computed $U|x\rangle$ by computing the bisimulation wrt subspace $S_{|x\rangle}$ and the respective reduced circuit; we report only circuit families which could be reduced, which were 9 out of 17. As before, we used 5 instances for each model and a timeout of 500 s; in the computation of the average reduction dimension d across all subspaces $S_{|x\rangle}$, a timeout was reached when the computation across all N subspaces $S_{|x\rangle}$ took more than 500 s.

	Grover		SAT			MaxCut		
qubits	DDSIM	CLUE	DDSIM	CLUE	d	DDSIM	CLUE	d
5	0.292	0.482	0.313	0.001	4.93/15	0.162	0.001	4.28/20
6	0.109	2.271	0.529	0.002	5.51/18	0.368	0.001	5.33/30
7	0.184	7.254	2.267	0.006	6.83/21	0.645	0.002	7.15/42
8	0.272	22.787	5.417	0.014	7.11/24	1.128	0.003	9.05/56
9	0.431	111.920	20.319	0.031	8.77/27	3.873	0.006	10.61/72
10	0.896	369.531	232.948	0.072	9.15/30	6.069	0.013	13.13/90
11	1.262	>500	>500	0.147	9.74/33	105.713	0.027	15.26/110
12	1.574	>500	>500	0.361	10.92/36	287.671	0.059	18.24/132
13	2.431	>500	>500	0.738	11.63/39	442.855	0.114	20.62/156
14	3.583	>500	>500	1.496	11.74/42	>500	0.232	24.24/182
15	5.452	>500	>500	3.232	13.08/45	>500	0.528	26.21/210

Table 1: Comparison of simulation times between DDSIM and the reduced model by CLUE. The latter includes the runtimes for computing the bisimulations by Algorithm 1. For SAT and MaxCut, column d reports the average size of the reduced circuit and its theoretical bounds from Theorem 5, separated by backslash.

Table 2 differentiates between reduction dimension wrt subspace $S_{|0\rangle}$ and the average reduction dimension across all subspaces $S_{|x\rangle}$. This is because the former can be interpreted as a BCB since $|0\rangle$ is the default input for most quantum circuits. Instead, the latter is meant to study the average reduction power of FCB, since FCB preserves quantum measurements. We remark that some circuits were only available for specific number of qubits (e.g., HHL and price calls). The reduction ratio d/N is given by the quotient between the dimension of the reduction dimension d and $N = 2^n$ (unlike Table 1 no bounds on d were available).

Overall, it can be noticed that substantial reductions could be obtained for a number of benchmark families. However, given that the benchmarks from Table 2 are all single-step applications, DDSIM was consistently faster than CLUE, once again confirming the observation from Remark 1.

5 Conclusion

We introduced forward and backward constrained bisimulations for quantum circuits which allow by means of reduction to preserve an invariant subspace of interest. The applicability of the approach was demonstrated by obtaining substantial reductions of common quantum algorithms, including, in particular, quantum search, quantum approximate optimization algorithms for SAT and MaxCut, as well as a number of benchmark circuits. Overall, the results suggest that constrained bisimulation can be used as a tool for speeding up the simulation of quantum circuits on classic computers, complementing state-of-

Circuit name	#-qubits	$\frac{d}{N}$ wrt $S_{\|0\rangle}$	Avg. $\frac{d}{N}$ across $S_{\|x\rangle}$	Avg. time (s)	DDSIM time
Deutsch-Jozsa	3	50.00%	47.22%	0.046	0.019
	4	25.00%	24.26%	0.226	0.021
	5	12.50%	12.31%	1.127	0.023
	6	6.25%	6.20%	5.294	0.024
	7	3.12%	**TO**	**TO**	0.026
GHZ	3	75.00%	69.44%	0.025	0.088
	4	87.50%	83.08%	0.339	0.070
	5	50.00%	48.67%	1.802	0.073
	6	25.00%	24.66%	5.766	0.078
	7	12.50%	**TO**	**TO**	0.082
Graph State	3	50.00%	66.67%	0.073	0.102
	4	25.00%	23.23%	0.242	0.086
	5	25.00%	28.98%	2.636	0.093
	6	9.38%	10.96%	8.981	0.104
	7	6.25%	**TO**	**TO**	0.116
HHL algorithm	5	12.50%	78.79%	1.874	0.032
Pricing Call Option	5	25.00%	27.27%	0.256	0.564
	7	12.50%	**TO**	**TO**	0.736
	9	6.25%	**TO**	**TO**	0.996
Pricing Put Option	5	25.00%	27.27%	0.256	0.564
	7	12.50%	**TO**	**TO**	0.801
	9	6.25%	**TO**	**TO**	1.207
QFT	3	25.00%	41.67%	0.041	0.115
	4	12.50%	22.79%	0.159	0.108
	5	6.25%	11.93%	1.312	0.130
	6	3.12%	6.11%	5.971	0.157
	7	1.56%	**TO**	**TO**	0.184
Quantum Walk	3	75.00%	65.00%	0.026	0.123
	4	50.00%	45.83%	0.150	0.331
	5	50.00%	47.43%	0.968	0.739
	6	50.00%	48.58%	6.885	0.762
	7	50.00%	**TO**	**TO**	0.784
Travelling Salesman	4	87.50%	87.50%	1.014	0.178
	9	**TO**	**TO**	**TO**	0.316

Table 2: Evaluation of (single-step) quantum benchmarks from repository [44]. The simulation times of DDSIM refer to the computation with respect to input $|0\rangle$, while the third and fourth columns report dimensions of bisimulation reductions. Instead, the fifth column reports the average computation time of $U|x\rangle$ via a reduction wrt $S_{|x\rangle}$, including the computation time of the bisimulation. A cumulative timeout of 500 s is denoted by **TO**.

the-art methods based on decision diagrams especially when the circuit is to be simulated under several initial conditions or for *multi-step* applications.

In line with the relevant literature on bisimulations for dynamical systems, constrained bisimulations introduce loss of information due to their underlying projection onto a smaller dimensional state space; the information that is preserved, however, is exact. A relevant issue for future work is to consider approximate variants of bisimulation for quantum circuits, in order to find more aggressive reductions or to capture quantum-specific phenomena such as quantum noise. Another line of research considers the combination with complementary circuit simulation approaches, in particular those based on decision diagrams.

Acknowledgments This work was partially supported by the Poul Due Jensen Foundation grant 883901, the Villum Investigator Grant S4OS and the project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union - NextGenerationEU.

References

1. Aaronson, S., Gottesman, D.: Improved simulation of stabilizer circuits. Physical Review A **70**(5), 052328 (2004)
2. Aharonov, D., van Dam, W., Kempe, J., Landau, Z., Lloyd, S., Regev, O.: Adiabatic quantum computation is equivalent to standard quantum computation. In: 45th IEEE Symposium on Foundations of Computer Science. pp. 42–51 (2004)
3. Amy, M.: Towards large-scale functional verification of universal quantum circuits. In: Selinger, P., Chiribella, G. (eds.) QPL. vol. 287, pp. 1–21 (2018)
4. Antoulas, A.: Approximation of Large-Scale Dynamical Systems. Advances in Design and Control, SIAM (2005)
5. Bacci, G., Bacci, G., Larsen, K.G., Mardare, R.: Complete axiomatization for the bisimilarity distance on markov chains. In: Desharnais, J., Jagadeesan, R. (eds.) CONCUR. LIPIcs, vol. 59, pp. 21:1–21:14 (2016)
6. Bacci, G., Bacci, G., Larsen, K.G., Mardare, R.: A complete quantitative deduction system for the bisimilarity distance on markov chains. Log. Methods Comput. Sci. **14**(4) (2018)
7. Bacci, G., Bacci, G., Larsen, K.G., Tribastone, M., Tschaikowski, M., Vandin, A.: Efficient local computation of differential bisimulations via coupling and up-to methods. In: Symposium on Logic in Computer Science, LICS. pp. 1–14 (2021)
8. Bacci, G., Bacci, G., Larsen, K.G., Mardare, R.: The bisimdist library: Efficient computation of bisimilarity distances for markovian models. In: Joshi, K.R., Siegle, M., Stoelinga, M., D'Argenio, P.R. (eds.) QEST. pp. 278–281 (2013)
9. Baier, C., Hermanns, H.: Weak bisimulation for fully probabilistic processes. In: CAV. pp. 119–130 (1997)
10. Baier, C., Katoen, J.: Principles of model checking. MIT Press (2008)
11. Boreale, M.: Algebra, coalgebra, and minimization in polynomial differential equations. Log. Methods Comput. Sci. **15**(1) (2019)
12. Boreale, M.: Complete algorithms for algebraic strongest postconditions and weakest preconditions in polynomial odes. Sci. Comput. Program. **193**, 102441 (2020)
13. Buchholz, P.: Exact and ordinary lumpability in finite Markov chains. Journal of Applied Probability **31**(1), 59–75 (1994)

14. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Maximal aggregation of polynomial dynamical systems. Proceedings of the National Academy of Sciences **114**(38), 10029 – 10034 (2017)
15. Cardelli, L., Pérez-Verona, I.C., Tribastone, M., Tschaikowski, M., Vandin, A., Waizmann, T.: Exact maximal reduction of stochastic reaction networks by species lumping. Bioinform. **37**(15), 2175–2182 (2021)
16. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Forward and backward bisimulations for chemical reaction networks. In: CONCUR. pp. 226–239 (2015)
17. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Comparing chemical reaction networks: A categorical and algorithmic perspective. In: Symposium on Logic in Computer Science, LICS. pp. 485–494 (2016)
18. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Symbolic computation of differential equivalences. In: POPL. pp. 137–150 (2016)
19. Cardelli, L., Tribastone, M., Tschaikowski, M., Vandin, A.: Guaranteed error bounds on approximate model abstractions through reachability analysis. In: QEST. pp. 104–121 (2018)
20. Derisavi, S., Hermanns, H., Sanders, W.H.: Optimal state-space lumping in Markov chains. Information Processing Letters **87**(6), 309 – 315 (2003)
21. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)
22. Farhi, E., Goldstone, J., Gutmann, S., Sipser, M.: Quantum computation by adiabatic evolution. arXiv preprint quant-ph/0001106 (2000)
23. Feng, Y., Duan, R., Ji, Z., Ying, M.: Probabilistic bisimulations for quantum processes. Information and Computation **205**(11), 1608–1639 (2007)
24. Feret, J., Danos, V., Krivine, J., Harmer, R., Fontana, W.: Internal coarse-graining of molecular systems. Proceedings of the National Academy of Sciences **106**(16), 6453–6458 (2009)
25. Feret, J., Henzinger, T., Koeppl, H., Petrov, T.: Lumpability abstractions of rule-based systems. Theoretical Computer Science **431**(0), 137 – 164 (2012)
26. Gay, S.J., Nagarajan, R.: Communicating quantum processes. In: POPL. p. 145–157 (2005)
27. Goldschmidt, A., Kaiser, E., DuBois, J.L., Brunton, S.L., Kutz, J.N.: Bilinear dynamic mode decomposition for quantum control. New Journal of Physics **23**(3), 033035 (2021)
28. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Righth Annual ACM Symposium on Theory of computing. pp. 212–219 (1996)
29. Grurl, T., Fuß, J., Hillmich, S., Burgholzer, L., Wille, R.: Arrays vs. decision diagrams: A case study on quantum circuit simulators. In: IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL). pp. 176–181 (2020)
30. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Physical Review Letters **103**(15), 150502 (2009)
31. Jiménez-Pastor, A., Jacob, J.P., Pogudin, G.: Exact Linear Reduction for Rational Dynamical Systems, pp. 198–216. Springer International Publishing (2022)
32. Khammassi, N., Ashraf, I., Fu, X., Almudever, C.G., Bertels, K.: Qx: A high-performance quantum computer simulation platform. In: Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017. pp. 464–469. IEEE (2017)
33. Kumar, A., Sarovar, M.: On model reduction for quantum dynamics: symmetries and invariant subspaces. Journal of Physics A: Mathematical and Theoretical **48**(1), 015301 (2014)

34. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. Inf. Comput. **94**(1), 1–28 (1991)
35. Leguizamon-Robayo, A., Jiménez-Pastor, A., Tribastone, M., Tschaikowski, M., Vandin, A.: Approximate Constrained Lumping of Polynomial Differential Equations, pp. 106–123. Springer Nature Switzerland (2023)
36. Liu, J.P., Kolden, H.Ø., Krovi, H.K., Loureiro, N.F., Trivisa, K., Childs, A.M.: Efficient quantum algorithm for dissipative nonlinear differential equations. Proceedings of the National Academy of Sciences **118**(35) (2021)
37. Meyer, C.D.: Matrix Analysis and Applied Linear Algebra. SIAM (2001)
38. Murali, P., McKay, D.C., Martonosi, M., Javadi-Abhari, A.: Software mitigation of crosstalk on noisy intermediate-scale quantum computers. In: Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems. pp. 1001–1016 (2020)
39. Nielsen, A.E., Hopkins, A.S., Mabuchi, H.: Quantum filter reduction for measurement-feedback control via unsupervised manifold learning. New Journal of Physics **11**(10), 105043 (2009)
40. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2000)
41. Niemann, P., Wille, R., Miller, D.M., Thornton, M.A., Drechsler, R.: Qmdds: Efficient quantum function representation and manipulation. IEEE Trans. Comput. Aided Des. Integr. Circuits Syst. **35**(1), 86–99 (2016)
42. Ovchinnikov, A., Pérez Verona, I., Pogudin, G., Tribastone, M.: CLUE: exact maximal reduction of kinetic models by constrained lumping of differential equations. Bioinformatics **37**(19), 3385–3385 (08 2021)
43. Pappas, G.J., Lafferriere, G., Sastry, S.: Hierarchically consistent control systems. IEEE Trans. Automat. Contr. **45**(6), 1144–1160 (2000)
44. Quetschlich, N., Burgholzer, L., Wille, R.: MQT Bench: Benchmarking software and design automation tools for quantum computing (2022), MQT Bench is available at https://www.cda.cit.tum.de/mqtbench/
45. Rowley, C.W., Mezič, I., Bagheri, S., Schlatter, P., Henningson, D.S.: Spectral analysis of nonlinear flows. Journal of Fluid Mechanics **641**, 115–127 (2009)
46. Sangiorgi, D.: Introduction to Bisimulation and Coinduction. Cambridge University Press (2011)
47. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Review **41**(2), 303–332 (1999)
48. Sipser, M.: Introduction to the theory of computation. ACM SIGACT News **27**(1), 27–29 (1996)
49. Sproston, J., Donatelli, S.: Backward bisimulation in Markov chain model checking. Software Engineering, IEEE Transactions on **32**(8), 531–546 (Aug 2006)
50. Steiger, D.S., Häner, T., Troyer, M.: Projectq: an open source software framework for quantum computing. Quantum **2**, 49 (2018)
51. Tognazzi, S., Tribastone, M., Tschaikowski, M., Vandin, A.: EGAC: a genetic algorithm to compare chemical reaction networks. In: Bosman, P.A.N. (ed.) GECCO. pp. 833–840. ACM (2017)
52. Tognazzi, S., Tribastone, M., Tschaikowski, M., Vandin, A.: Backward Invariance for Linear Differential Algebraic Equations. In: CDC. pp. 3771–3776 (2018)
53. Tomlin, A.S., Li, G., Rabitz, H., Tóth, J.: The effect of lumping and expanding on kinetic differential equations. SIAM Journal on Applied Mathematics **57**(6), 1531–1556 (1997)
54. Tschaikowski, M., Tribastone, M.: Spatial fluid limits for stochastic mobile networks. Perform. Evaluation **109**, 52–76 (2017)

55. Vidal, G.: Efficient classical simulation of slightly entangled quantum computations. Phys. Rev. Lett. **91**, 147902 (Oct 2003)
56. Villalonga, B., Boixo, S., Nelson, B., Henze, C., Rieffel, E., Biswas, R., Mandrà, S.: A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. npj Quantum Information **5**(1), 86 (2019)
57. Wille, R., Hillmich, S., Burgholzer, L.: Tools for quantum computing based on decision diagrams. ACM Transactions on Quantum Computing **3**(3) (jun 2022)
58. Ying, M., Feng, Y.: Model checking quantum systems - A survey (2018), arxiv
59. Ying, M., Li, Y., Yu, N., Feng, Y.: Model-checking linear-time properties of quantum systems. ACM Trans. Comput. Logic **15**(3) (2014)