Optimized Density Matrix Representations

Improving the Basis for Noise-Aware Quantum Circuit Design Tools

Thomas Grurl*† Jürgen Fuß* Robert Wille‡§
*Secure Information Systems, University of Applied Sciences Upper Austria, Austria
†Institute for Integrated Circuits, Johannes Kepler University Linz, Austria
‡Chair for Design Automation, Technical University of Munich, Germany
§Software Competence Center Hagenberg GmbH (SCCH), Austria
thomas.grurl@fh-hagenberg.at juergen.fuss@fh-hagenberg.at robert.wille@tum.de
www.cda.cit.tum.de/research/quantum/

***Abstract*—By exploiting quantum mechanical effects, quantum computers can tackle problems that are infeasible for classical computers. At the same time, these quantum mechanical properties make handling quantum states exponentially hard—imposing major challenges on design tools. In the past, methods such as tensor networks or decision diagrams have shown that they can often keep those resource requirements in check by exploiting redundancies within the description of quantum states. But developments thus far focused on pure quantum states which do not provide a physically complete picture and, e.g., ignore frequently occurring noise effects. Density matrix representations provide such a complete picture, but are substantially larger. At the same time, they come with characteristics that allow for a more compact representation. In this work, we unveil this untapped potential and use it to provide a decision diagram representation that is optimized for density matrix representations. By this, we are providing a basis for more efficient design tools such as quantum circuit simulation which explicitly takes noise/error effects into account.**

I. Introduction

Quantum computing [1] promises to solve computational problems that are infeasible for classical computers. They achieve this by utilizing quantum mechanical effects such as superposition (a quantum bit can be in a combination of states) and entanglement (applying an operation to one qubit can affect other qubits as well). Famous early examples for quantum algorithms are Shor's algorithm for factoring integers in polynomial runtime [2] and Grover's database search algorithm [3]. Since then, plenty of further quantum algorithms have emerged and are finding several applications in a mid-term or even near-term perspective—including but not limited to quantum machine learning [4], quantum chemistry [5], and quantum optimization [6].

Together with the emergence of accessible quantum computers (provided, e.g., by Google, IBM, Amazon, and many more), this also leads to an increasing interest in automatic methods that support engineers in the design and validation of quantum algorithms. Recent developments led to a variety of corresponding design tools, e.g., for simulation [7]–[9], compilation [10]–[12], verification [13]–[16], and evaluation of quantum error-correcting codes [17], [18]. However, the quantum mechanical effects that need to be captured by these tools yield exponential descriptions of quantum states and quantum operations—leading to a huge challenge for corresponding tool developers.

In the past, several solutions, e.g., based on tensor networks [19]–[21] or decision diagrams [22]–[29] have been proposed which often can keep those resource requirements in check, e.g., by exploiting redundancies within the description of quantum states. But all those developments mostly focused on working with *pure* quantum states. Thereby, ignoring frequently occurring noise effects caused by the fragile nature of quantum mechanical effects [30].

Fortunately, noise effects are well understood and, hence, according descriptions in terms of so-called *mixed quantum states* described by means of *density matrices* exist [1]. But, although they allow one to properly represent a quantum state including possible noise effects, they are substantially larger than the (already exponential) pure quantum state representations. As a result, most existing design tools either do not yet support a noise-aware consideration of quantum states or are heavily limited in their efficiency and scalability (e.g., [7], [31]–[35]). This urgently motivates further research towards more optimized density matrix representations.

In this work, we propose an approach towards optimized density matrix representations based on decision diagrams. While first works have already demonstrated the potential of decision diagrams for considering noise using density matrices (see [36], [37]), the hermitian property of density matrices has not been exploited yet. However, doing this allows for the utilization of redundancies which have not been caught before (leading to a more compact representation in many cases). We illustrate this untapped potential and use it to provide an optimized decision diagram representation for density matrices. Afterwards, we discuss the magnitude of the resulting potential and empirically confirm these findings (using noise-aware quantum circuit simulation as a representative design tool). By this, we are providing a basis for more efficient noise-aware quantum circuit design tools.

The remainder of this paper is structured as follows: In Section II, the preliminaries including quantum computing and mixed states are reviewed. Section III discusses the shortcomings of current representations of density matrices in terms of decision diagrams and illustrates the potential that is yet untapped. Section IV then presents how this leads to an optimized density matrix representation. Afterwards, we discuss the magnitude of the unveiled potential in Section V and present empirical results confirming that in Section VI. Finally, Section VII concludes the paper.

II. Preliminaries

In this section, we review the basics of quantum computing and mixed states—both, providing the preliminaries of this work.

A. Quantum Computing

Similar to the classical realm, the smallest unit of information in the quantum realm is a bit, which can assume the states 0 and 1. However, in quantum computing they are called *quantum bits* or *qubits* and, in contrast to classical bits, these qubits can not only be in a state 0 or 1, but also in an (almost) arbitrary combination of those states. The states 0 and 1 are called *basis states* and—using Dirac notation—are written as $|0\rangle$ and $|1\rangle$. States which are a combination of those basis states are said to be in *superposition*. In general, such a state can be described as $|\psi\rangle = \alpha_0 \cdot |0\rangle + \alpha_1 \cdot |1\rangle$ with *amplitudes* $\alpha_0, \alpha_1 \in \mathbb{C}$. The amplitudes describe how strongly the qubit relates to each of the basis states and

must satisfy the normalization constraint that the sum of their squared magnitudes is equal to 1, i.e., $|\alpha_0|^2 + |\alpha_1|^2 = 1$ for a single qubit. Measuring a qubit collapses it to $|0\rangle$ ($|1\rangle$) with probability $|\alpha_0|^2$ ($|\alpha_1|^2$). The measurement is destructive and destroys possible superposition. Thus, repeated measurements always return the same result.

This state description can be extended for multiple qubits. For example, a two-qubit state is fully characterized by $2^2 = 4$ amplitudes described as $\alpha_{00} \cdot |00\rangle + \alpha_{01} \cdot |01\rangle + \alpha_{10} \cdot |10\rangle + \alpha_{11} \cdot |11\rangle$. Often, this state description is shortened to a state vector containing only the amplitudes, e.g., $[\alpha_{00}\ \alpha_{01}\ \alpha_{10}\ \alpha_{11}]^\top$ for $n = 2$ qubits.

Example 1. *Consider a quantum register which is in a state* $|\psi\rangle = 1/\sqrt{2} \cdot |00\rangle + 0 \cdot |01\rangle + 1/\sqrt{2} \cdot |10\rangle + 0 \cdot |11\rangle$ *. The corresponding vector description* $|\psi\rangle$ *is given as* $[1/\sqrt{2}\ 0\ 1/\sqrt{2}\ 0]^\top$*. This represents a valid state, since* $|1/\sqrt{2}|^2 + 0^2 + |1/\sqrt{2}|^2 + 0^2 = 1$ *satisfies the normalization constraint. Since* $|\psi\rangle$ *is in a superposition, measuring it would yield either* $|00\rangle$ *or* $|10\rangle$*—both with probability* $|1/\sqrt{2}|^2 = 1/2$*.*

Quantum states are altered by quantum operations, which are characterized by unitary matrices, i.e., square matrices whose inverse is their complex conjugate transpose. Important operations include $\mathrm{H} = 1/\sqrt{2}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$, $\mathrm{X} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, $\mathrm{Z} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$. The H operation transforms a basis state into a superposition, the X operation being the quantum counterpart of flipping a bit and the Z gate flipping the phase of a qubit. Operations can also affect multiple qubits at the same time. An important example of a two-qubit operation is the controlled-X (also known as CNOT) operation, which negates the state of a qubit, iff the chosen control qubit is $|1\rangle$. Operations are applied to the quantum state by matrix-vector multiplication.

Example 2. *Consider again the state* $|\psi\rangle$ *from Example 1. Applying a CNOT operation on it flips the basis of the second qubit if the first qubit is set to* $|1\rangle$*. This is described as*

$$\underbrace{\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}}_{\text{CNOT}} \cdot \underbrace{\begin{bmatrix}1/\sqrt{2}\\ 0\\ 1/\sqrt{2}\\ 0\end{bmatrix}}_{|\psi\rangle} = \underbrace{\begin{bmatrix}1/\sqrt{2}\\ 0\\ 0\\ 1/\sqrt{2}\end{bmatrix}}_{|\psi'\rangle}.$$

Possible outcomes of measuring $|\psi'\rangle$ *are* $|00\rangle$ *and* $|11\rangle$*, each with probability* $1/2$*. Note that the measurement of one qubit actually affects the other qubit as well—an essential concept of quantum computing called* entanglement*.*

B. Mixed states

The formalism presented above can be used to simulate the execution of *perfect* quantum computers. Unfortunately, real quantum computers are not perfect but prone to noise caused by the fragility of quantum mechanical effects [30]. This leads to gate errors (introduced by imperfect physical realizations of quantum operations) or coherence errors (introduced by the inability of qubits to hold information for an extended amount of time) [38].

These errors can be viewed as (unwanted) operations on the state. In contrast to ideal perfect quantum operations, however, these error operations come with an additional degree of randomness. Thus, the result of such an erroneous quantum computation cannot be described by a single vector anymore. Instead, the state is now characterized by a *mixture* (or ensemble) of possible states $\{(p_i, |\psi_i\rangle)\}$, with $|\psi_i\rangle$ representing quantum states and p_i representing the respective probability that the system is in this state. This is usually called a *mixed state*.

Example 3. *Consider the two-qubit state from Example 2* $|\psi'\rangle = 1/\sqrt{2}(|00\rangle + |11\rangle)$ *and assume that this state might be affected by a gate error in the first qubit—depolarizing it and setting it to a completely random state. Suppose the error only occurs with a probability of 1 %. Then, 99 % of the time nothing happens and the state remains unchanged. Otherwise (with probability 1 %), the first qubit is depolarized, which is captured by either applying X, Y, Z or leaving the state unchanged (each with probability 0.25 %)—yielding the state mixture* $\{(99\ \%, \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), (0.25\ \%, (\frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), (0.25\ \%, (\frac{i}{\sqrt{2}}(-|01\rangle + |10\rangle), (0.25\ \%, \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)(0.25\ \%, \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \}$*, which cannot be represented by a single pure two-qubit state anymore.*

As illustrated by the example, state vectors are unsuited to represent mixed states. Hence, so-called density matrices are used instead, which offer a structure to incorporate all possible states into a single description. Density matrices are generated from a state ensemble $\{(p_0, |\psi_0\rangle), (p_1, |\psi_1\rangle), \ldots, (p_i, |\psi_i\rangle)\}$ by

$$\rho = \sum_{i=0}^{n} p_i \left|\psi_i\right\rangle \left\langle\psi_i\right|, with \left\langle\psi_i\right| := \left|\psi_i\right\rangle^\dagger.$$

Example 4. *Considering again the state* $|\psi'\rangle$ *from above, the corresponding* density matrix ρ *is given by*

$$\begin{bmatrix}1/\sqrt{2}\\ 0\\ 0\\ 1/\sqrt{2}\end{bmatrix} \cdot \begin{bmatrix}1/\sqrt{2} & 0 & 0 & 1/\sqrt{2}\end{bmatrix} = \begin{bmatrix}1/2 & 0 & 0 & 1/2\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 1/2 & 0 & 0 & 1/2\end{bmatrix}.$$

Here, the diagonal entries from the first element in the upper-left to the last element in the lower-right represent the probabilities for measuring $|00\rangle, |01\rangle, |10\rangle$*, and* $|11\rangle$*, respectively. Overall, this allows one to represent a mixture of possible states.*

Having such an extended representation means the way in which quantum operations are applied has to be adjusted. As with pure states, quantum operations are still represented in the form of unitary matrices U. But to accommodate the density matrix representation of mixed states, now two matrix-matrix multiplications are necessary to apply them (instead of a matrix-vector multiplication as reviewed in Section II-A). That is, applying U onto ρ is given by

$$\rho' = U \rho\, U^\dagger,$$

where $U^\dagger$ represents the complex conjugate transpose of U.

III. Motivation and General Idea

In this section, we propose the general idea behind an improved representation of density matrices. To this end, we introduce the concept of recursively decomposing matrices into smaller structures—the basis for current decision diagram-based quantum state representations. Afterwards, we illustrate the shortcomings when working with density matrices and present the proposed decomposition scheme that additionally takes their characteristics into account.

A. Current Representation of Density Matrices

A major challenge when working with quantum states is that they grow exponentially in size with each tracked qubit. Thus, handling states consisting of more than a few qubits becomes very difficult. To address this problem, alternative data structures have been proposed for working with quantum states.

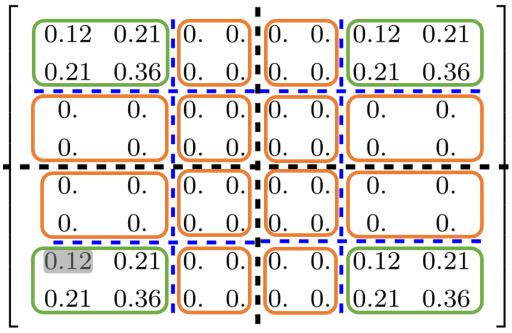


Fig. 1. Established matrix decomposition applied to density matrices

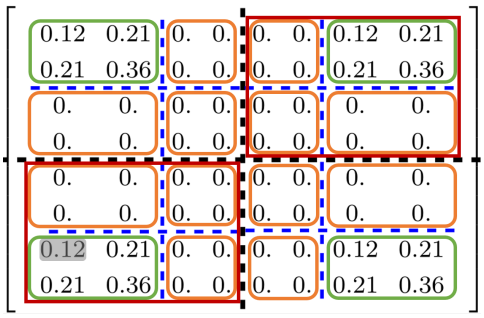


Fig. 2. Extended decomposition scheme optimized for density matrices

While the underlying problem still remains, those alternative descriptions sometimes allow for a more efficient handling of quantum states. One of those alternative approaches is based on the observation that matrices of (mixed) quantum states often contain redundant sub-structures (see [36], [37]). By recursively decomposing objects, those redundancies can be identified and removed—allowing for a more compact representation.

Example 5. *To illustrate the idea, the quantum register ρ from Example 4 is extended by $|\phi\rangle = 1/\sqrt{4}\,|0\rangle + \sqrt{3/4}\,|1\rangle$, using the Kronecker Product, which results in the state $\rho^\star = \rho \otimes |\phi\rangle\langle\phi|$. In Fig. 1, the corresponding density matrix of $\rho^\star$ is provided, with the values rounded to two decimal places for readability. The decomposition process revolves around recursively quartering the matrix and checking for redundancies within the sub-matrices. That is, $\rho^\star$ is quartered (as indicated by the dashed black lines) and the sub-matrices are checked for redundancies. Since all sub-matrices are different, no redundancies are determined and the decomposition process is repeated on all sub-matrices (as indicated by the dashed blue lines). In this step, redundant sub-matrices are identified, which are framed in green and orange. The sub-matrix containing only zeros is discarded and the decomposition process is only repeated for the sub-matrix framed green. Quartering this sub-matrix results in complex numbers only—terminating the decomposition process.*

After determining those redundancies, the matrix is represented using a graph-based data structure, namely decision diagrams [22]–[27], where the redundant structures are only stored once. In many cases, this yields descriptions that are much more compact than straightforward representations—often even allowing for linear size representations compared to the exponential size of those matrices in general [39]. Accordingly, this makes this approach interesting for many design tools such as quantum circuit simulation.

However, this decomposition scheme was originally developed for representing *pure* quantum state vectors and quantum operations, i.e., representations that do not consider the noise effects as reviewed above in Section II. As shown next, this leaves potential which would allow for a more optimized representation of density matrices and, hence, mixed quantum states.

B. Proposed Idea

While density matrices are substantially larger than state vectors, they also come with the nice characteristic that they are hermitian. Because of that, all density matrices are equal to the transpose of their complex conjugate, i.e., each density matrix M satisfies $M = M^\dagger$. Therefore, each off-diagonal element of a density matrix is mirrored (in a transposed and complex conjugated fashion) to the element on the other side of the diagonal. Thus, up to half the elements within the density matrix are redundant. Current (established) decomposition schemes for quantum objects (such as reviewed above and illustrated in Fig. 1) do not exploit this potential.

In order to exploit this characteristic, we propose to set the "mirrored" parts of the matrix equal. More precisely, we propose to extend the decomposition scheme described above in such a way that the matrix elements below the diagonal are discarded and only elements on and above the matrix diagonal are stored. To illustrate the idea, consider the following example:

Example 6. *Consider again the density matrix representation of $\rho^\star$ from Example 5 and the current (established) decomposition scheme sketched in Fig. 1. The newly proposed scheme is illustrated in Fig. 2. First, the matrix is again split into four sub-matrices (indicated by the black dashed lines). Although all sub-matrices appear different, we know that the upper right and lower left sub-matrices (framed in red) are just the complex conjugate transpose of each other (due to the hermitian characteristic). We therefore can discard one of the two sub-matrices without losing any information (without loss of generality, we always discard the lower left sub-matrix in the following). Then, the decomposition step is repeated on the remaining three sub-matrices (as indicated by the blue dashed lines). Here, two identical sub-matrices can be identified, which are framed in green and orange (the same that could be identified redundant with the established decomposition scheme reviewed in the previous section).*

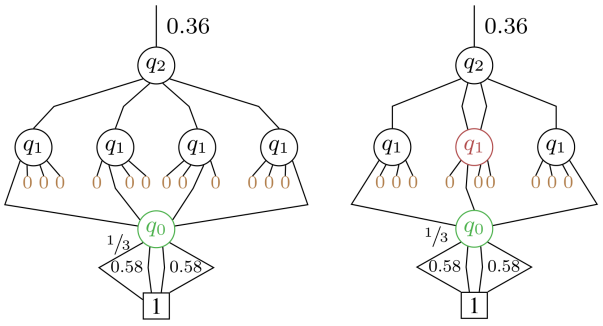
Obviously, exploiting the hermitian characteristic allows for much more potential in exploiting redundancies and, by this, determining much more compact representations. However, to properly use such an optimized representation and to apply quantum circuit operations on corresponding state representations, some care has to be taken when reconstructing the original matrix elements. This leads to an extended type of decision diagram which is introduced next.

IV. Optimized Decision Diagrams for the Representation of Density Matrices

In this section, we describe how, using the ideas sketched above, optimized decision diagrams representing density matrices can be constructed. Afterwards, we also present how quantum operations are applied to the resulting decision diagrams and how this affects their structure.

A. Representation

As described above, the key strength of decision diagrams is in determining and exploiting redundancies within structures. They are constructed from matrices by following the decomposition process presented above and reflecting this process in a graph-based representation. More precisely, consider a quantum register composed of n qubits $(q_0, q_1, \ldots, q_{n-1})$,



(a) Using established decomposition (b) Using proposed decomposition

Fig. 3. Decision diagrams representing density matrices

with q_{n-1} representing the most significant qubit. When the density matrix representing this quantum register is quartered, the upper left sub-matrix contains the probabilities for q_{n-1} to be $|0\rangle$ and the lower right sub-matrix contains the probabilities for q_{n-1} to be $|1\rangle$. The upper right and the lower left sub-matrices contain information about the coherence of the state. This is represented as a decision diagram node labeled q_{n-1} with four successor nodes, where the first one representing the upper left sub-matrix, the second one and the third one representing the upper right and the lower left sub-matrix, and the fourth one representing the lower right sub-matrix. This process is recursively repeated for $q_{n-2}, q_{n-3}, \ldots, q_0$, i.e., until matrices of size one (i.e., complex numbers) remain—yielding a so-called terminal node. During this decomposition process, redundant sub-matrices are represented by the same node (called *shared node*) which eventually leads to a more compact representation. Having that, values of the respective matrix entries are obtained by multiplying the edge weights along the corresponding path.

Example 7. *Fig. 3a shows the decision diagram representation of $\rho^\star$ which results when following the established decomposition scheme. To aid the readability of the decision diagram and following established conventions, edge weights of 1 are omitted and nodes with an incoming edge weight of 0 are represented as 0-stubs—indicating that matrix values of all possible states represented by this part of the decision diagram are 0. As can be seen, the redundancies discussed before in Example 5 and framed in green and orange in Fig. 1 are accordingly reflected in the decision diagram. To obtain a value of a matrix entry from this decision diagram, the edge weights of the corresponding path must be multiplied. For example, to reconstruct the value 0.12 from the lower left part of the matrix (highlighted gray), the edge weight of the root edge (0.36) must be multiplied with the third edge of q_2 (1), the first edge of q_1 (1), and the first edge of q_0 ($1/3$)—resulting in $0.36 \cdot 1 \cdot 1 \cdot 1/3 = 0.12$.*

Considering the hermitian property of density matrices, this representation can be optimized. Depending on the current position in the decomposition process, the upper right and the lower left sub-matrices must be the complex conjugate transpose of each other. In those cases, the first successor still represents the upper left sub-matrix, while the third successor represents the lower right sub-matrix. Eventually, the second successor represents both, the upper right and the lower left sub-matrix in a shared fashion. More precisely, without loss of generality, the upper right sub-matrix is directly represented by the second successor, while the lower left sub-matrix is discarded and replaced by a "pointer" to the second node (a pointer additionally storing that the corresponding successor node does represent the complex conjugate transpose). Of course, the scheme is recursively repeated for all qubits again. Values of the respective matrix entries are then obtained again by multiplying the edge weights along the corresponding path—but dynamically modifying values when a pointer is taken which indicates a complex conjugate transpose.

Example 8. *Fig. 3b shows the decision diagram representation of $\rho^\star$ which results when following the proposed decomposition scheme. As can be seen, the additional redundancies discussed before in Example 6 and framed in red in Fig. 2 can be exploited (in addition to the redundancies framed in green and orange that are detected with the currently established scheme). To obtain a value of a matrix entry from this decision diagram, again, the edge weights of the corresponding path must be multiplied; but now taking a complex conjugate transpose into account when indicated by the respective pointer, i.e., by flipping the second and third edges and conjugating the edge weights. For example, to reconstruct the values 0.12 from the lower left part of the matrix (highlighted gray), the edge weight of the root edge (0.36) must be multiplied with the right edge of q_2 (1), the second edge of q_1 (1), and the first edge of q_0 ($1/3$), resulting in $0.36 \cdot 1 \cdot 1 \cdot 1/3 = 0.12$. Notice, that the path at q_1 is flipped—corresponding to a transpose of q_1 (so that the correct matrix element could be restored). Overall, this yields a decision diagram which requires one node less than the one discussed in Example 7 and shown in Fig. 3a—a reduction by approx. 17 %*[1].

B. Applying Operations

Having an (optimized) quantum state representation of a density matrix obviously is good, but only really helps in design tools if the structure is also applicable for conducting the required operations on it. In the application scenario considered here, multiplication of a (density) matrix M with a matrix representing quantum operation U is a core requirement. This can be efficiently applied using the proposed decision diagram scheme. This is because $U \cdot M$ can be written as

$$U \cdot M = \begin{bmatrix} U_{00} & U_{01} \\ U_{10} & U_{11} \end{bmatrix} \cdot \begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} = \begin{bmatrix} U_{00} \cdot M_{00} & U_{00} \cdot M_{01} \\ U_{10} \cdot M_{00} & U_{10} \cdot M_{01} \end{bmatrix} + \begin{bmatrix} U_{01} \cdot M_{10} & U_{01} \cdot M_{11} \\ U_{11} \cdot M_{10} & U_{11} \cdot M_{11} \end{bmatrix},$$

with the subscripted elements representing sub-matrices corresponding to the decomposition scheme presented in Example 8, i.e., the upper left sub-matrix corresponds to U_{00}, the upper right sub-matrix corresponds to U_{01}, the lower left sub-matrix corresponds to U_{10}, and the lower right sub-matrix corresponds to U_{11} for U (the same applies to M). The necessary sub-products are determined by recursively continuing this process, until sub-matrices of size one, i.e., complex numbers, remain onto which those operations can be applied directly. Hence, multiplication can be decomposed in a similar fashion as the matrices yielding the (proposed) decision diagrams.

Since the off-diagonal elements of the density matrix are encoded into the same decision diagram sub-structure (although they are only the complex conjugate transpose of each other) some care has to be taken during the traversal of the decision diagram. Depending on the path taken in the decision diagram, pointers have to be modified corresponding to a conjugate complex transpose, i.e., the second and third edge has to be flipped and the edge weight must be conjugated. Besides that, however, operations can be applied in a similar fashion as before.

[1]Note that 17 % might not seem much. But considering that the example is rather small, it is significant. Discussions and evaluations respectively summarized and confirmed below show that improvements of up to 50 % are possible when exploiting the identified potential.

V. Resulting Potential

As shown above, exploiting the hermitian characteristics as proposed in this work may lead to the detection of redundancies that were not caught before—yielding optimized, i.e. more compact, representations for density matrices. However, the magnitude of the resulting potential strongly depends on the considered quantum states. This section briefly discusses the resulting potential in a conceptual fashion (before the derived conclusions are also confirmed empirically in the next section). To this end, we can roughly distinguish two cases:

1) Cases where few redundancies are exploited by earlier methods: If density matrices are considered where current (established) decomposition schemes could not identify any (or not many) redundancies, substantial improvements are possible with the scheme proposed in this work. In fact, even if nothing above the matrix diagonal is redundant (and, hence, offers potential for a compact representation), at least everything below the diagonal is definitely redundant and can be discarded. For a density matrix representing n qubits, this means that the representation of up to $(2^n-1)2^n/2$ entries of the matrix are redundant and can be avoided—yielding a reduction of up to 50 % (but never 50 % itself or more as the diagonal entries themselves still need to be represented). That is, the proposed scheme offers a particular improvement for instances where not much redundancy could be exploited by earlier methods.

2) Cases which already exploit many redundancies: In contrast, if current (established) decomposition schemes already were able to detect lots of redundancies, the improvements achieved with the scheme proposed in this work are less or even non-existent. In fact, if, e.g., the entries below the matrix diagonal are already available in a very compact fashion, exploiting the hermitian characteristic hardly provides any further room for compaction. After all, there is a limit on how compact states can be represented. That is, the proposed scheme offers substantially less or even no improvement for instances where already a substantial amount of redundancy could be exploited.

Example 9. *The effects described above can even be seen in the example considered above for the state $\rho^\star$ and its density matrix representation in Fig. 2 as well as decision diagram representations in Fig. 3. In the first decomposition step, more redundancy can be exploited with the scheme proposed in this work, since the sub-matrices framed red only need to be represented once (leading to one node less in the decision diagram). At the same time, in the remaining decomposition steps, all redundancies identified by the proposed scheme can also be identified by the currently established scheme. That is, since they already allow for a compact representation, no further optimizations are obtained here. Overall, this does not lead to the best possible improvement of up to 50 %, but still a significant one of approx. 17 % which could not be exploited before.*

VI. Empirical Results

In order to also empirically evaluate the potential discussed above, we implemented the proposed scheme in C++[2] using the open-source decision diagram package from [40]. Afterwards, we compared the performances of the current (established) decision diagrams with the ones proposed in this work when applied to noise-aware quantum circuit simulation (as a representative of a design task for which efficient representations of density matrices are key). In particular, we measured the runtime of simulating different quantum circuits, as well as the size of the resulting decision diagrams (i.e., the number of nodes).

As benchmarks we used the *Quantum Fourier Transform* (QFT, [1]) with an increasing number of qubits. The QFT is a common use case and an essential part of several important quantum algorithms (e.g., Shor's factorization algorithm [2] or quantum phase estimation [1]). Besides that, we also considered a selection of further quantum circuits taken from the benchmark set of [41]. All circuits have been considered with and without the application of noise effects. In the noisy case, we assumed gate errors (mimicked by depolarization of 1 %) as well as two types of coherence errors, namely, T1 errors with 2 % and T2 errors with 1 % [38].

In Table I, the results of the evaluation are summarized. For each considered quantum circuit, we list the node count of the final state and the runtime of the quantum circuit simulation—for the established scheme (*Est.*) as well as the proposed scheme (*Prop.*). Finally, we also provide the obtained improvements (*Improv.*).

The results clearly confirm the discussions from Section V. If the current (established) decomposition scheme already identifies a large number of redundancies, the scheme proposed in this work does not provide any further optimization. This is particularity the case when no noise is considered—here the involved decision diagrams mostly stay compact, so that no improvement can be reported. In contrast, if current (established) schemes were not able to detect many redundancies substantial improvements of up to 50 % can be achieved. This is particularly the case when noise effects are considered (and for which density matrices are required for).

The benefits yielded by the proposed scheme for quantum circuit simulation increase with the absolute size of the involved decision diagrams. So, while no runtime improvement can be reported when the simulation is already very fast, considerable improvements of up to 66 % are achieved for larger simulations.

VII. Conclusions

In this work, we proposed an optimized representation of density matrices which provide the main basis to describe quantum states including noise effects. To this end, we exploited the hermitian characteristic that allows for the utilization of redundancies that have not been caught before. We illustrated the proposed idea by reviewing the shortcomings of the current (established) decomposition approach and outlining the missed potential. This potential was eventually discussed and confirmed by empirical evaluations (using noise-aware quantum circuit simulation as a representative design tool). Overall, the results showed that substantial further improvements can be gained with the proposed scheme—providing promising prospects also for other noise-aware quantum circuit design tools for which efficient representations of density matrices are key.

Acknowledgments

This work received funding from the University of Applied Sciences, from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No. 101001318), was part of the Munich Quantum Valley, which is supported by the Bavarian state government with funds from the Hightech Agenda Bayern Plus, and has been supported by the BMWK on the basis of a decision by the German Bundestag through project QuaST.

[2]The implementation is available as open source at: www.github.com/cda-tum/ddsim

TABLE I
EMPIRICAL RESULTS

			Without Noise						With Noise					
			Nodes			Runtime			Nodes			Runtime		
Circuit	#Q	#G	Est.	Prop.	Improv.	Est.	Prop.	Imp.	Est.	Prop.	Improv.	Est.	Prop.	Improv.
qft	6	86	6	6	0 %	<1 s	<1 s	0 %	683	528	**23 %**	<1 s	<1 s	0 %
qft	7	116	7	7	0 %	<1 s	<1 s	0 %	2731	2080	**24 %**	<1 s	<1 s	0 %
qft	8	153	8	8	0 %	<1 s	<1 s	0 %	10923	8256	**24 %**	1 s	1 s	**16 %**
qft	9	192	9	9	0 %	<1 s	<1 s	0 %	43691	32896	**25 %**	8 s	6 s	**24 %**
qft	10	241	10	10	0 %	<1 s	<1 s	0 %	174763	131328	**25 %**	186 s	123 s	**34 %**
qaoa	6	270	741	420	**43 %**	<1 s	<1 s	0 %	1365	714	**48 %**	2 s	1 s	**13 %**
vqe	6	2282	345	190	**45 %**	2 s	2 s	0 %	1365	714	**48 %**	12 s	9 s	**23 %**
qpe	9	150	1368	717	**48 %**	<1 s	<1 s	0 %	43701	22106	**49 %**	28 s	15 s	**45 %**
adder	10	142	10	10	0 %	<1 s	<1 s	0 %	211901	106469	**49 %**	2637 s	934 s	**65 %**
multiply	14	124	13	13	0 %	<1 s	<1 s	0 %	161361	90980	**44 %**	871 s	300 s	**66 %**

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Jour. of Comp.*, vol. 26, no. 5, pp. 1484–1509, 1997.

[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Symp. on Theory of Computing*, 1996, pp. 212–219.

[4] D. Riste *et al.*, "Demonstration of quantum advantage in machine learning," *npj Quantum Information*, vol. 3, no. 1, pp. 1–5, 2017.

[5] Y. Cao *et al.*, "Quantum chemistry in the age of quantum computing," *Chemical reviews*, vol. 119, no. 19, pp. 10 856–10 915, 2019.

[6] E. Farhi, J. Goldstone, S. Gutmann, and L. Zhou, "The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size," *Quantum*, vol. 6, p. 759, 2022.

[7] H. Abraham *et al.*, *Qiskit: An open-source framework for quantum computing*, 2019.

[8] T. Jones, A. Brown, I. Bush, and S. C. Benjamin, "QuEST and high performance simulation of quantum computers," *Scientific reports*, vol. 9, no. 1, pp. 1–11, 2019.

[9] B. Villalonga *et al.*, "A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware," *npj Quantum Information*, vol. 5, no. 1, pp. 1–16, 2019.

[10] M. Amy, D. Maslov, M. Mosca, and M. Roetteler, "A meet-in-the-middle algorithm for fast synthesis of depth-optimal quantum circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 32, no. 6, pp. 818–830, 2013.

[11] E. A. Sete, W. J. Zeng, and C. T. Rigetti, "A functional architecture for scalable quantum computing," in *Int'l. Conf. on Rebooting Computing*, 2016.

[12] A. Zulehner, A. Paler, and R. Wille, "An efficient methodology for mapping quantum circuits to the IBM QX architectures," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 7, pp. 1226–1236, 2019.

[13] Z. Brakerski, P. Christiano, U. Mahadev, U. Vazirani, and T. Vidick, "A cryptographic test of quantumness and certifiable randomness from a single quantum device," in *Foundations of Computer Science*, 2018, pp. 320–331.

[14] A. Gheorghiu, T. Kapourniotis, and E. Kashefi, "Verification of quantum computation: An overview of existing approaches," *Theory of Computing Systems*, vol. 63, no. 4, pp. 715–808, 2019.

[15] L. Burgholzer and R. Wille, "Advanced equivalence checking for quantum circuits," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2021.

[16] L. Burgholzer, R. Raymond, and R. Wille, "Verifying results of the IBM Qiskit quantum circuit compilation flow," in *Int'l Conf. on Quantum Computing and Engineering*, 2020, pp. 356–365.

[17] C. Gidney, "Stim: A fast stabilizer circuit simulator," *Quantum*, vol. 5, p. 497, 2021.

[18] T. Grurl, C. Pichler, J. Fuß, and R. Wille, "Automatic implementation and evaluation of error-correcting codes for quantum computing: An open-source framework for quantum error-correction," in *VLSI Design*, 2023.

[19] R. Orús, "A practical introduction to tensor networks: Matrix product states and projected entangled pair states," *Annals of Physics*, vol. 349, pp. 117–158, 2014.

[20] I. L. Markov and Y. Shi, "Simulating quantum computation by contracting tensor networks," *SIAM J. Comput.*, vol. 38, no. 3, pp. 963–981, 2008.

[21] J. D. Biamonte and V. Bergholm, "Tensor networks in a nutshell," 2017. arXiv: 1708.00006.

[22] G. F. Viamontes, I. L. Markov, and J. P. Hayes, *Quantum Circuit Simulation*. Springer, 2009.

[23] A. Zulehner and R. Wille, "Advanced simulation of quantum computations," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 848–859, 2019.

[24] D. Miller and M. Thornton, "QMDD: A decision diagram structure for reversible and quantum circuits," in *Int'l Symp. on Multi-Valued Logic*, 2006.

[25] A. Abdollahi and M. Pedram, "Analysis and synthesis of quantum circuits by using quantum decision diagrams," in *Design, Automation and Test in Europe*, 2006, pp. 317–322.

[26] S.-A. Wang, C.-Y. Lu, I.-M. Tsai, and S.-Y. Kuo, "An XQDD-based verification method for quantum circuits," *IEICE Trans. Fundamentals*, vol. 91-A, no. 2, pp. 584–594, 2008.

[27] P. Niemann, R. Wille, D. M. Miller, M. A. Thornton, and R. Drechsler, "QMDDs: Efficient quantum function representation and manipulation," *IEEE Trans. on CAD of Integrated Circuits and Systems*, vol. 35, no. 1, pp. 86–99, 2016.

[28] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Graph-based simulation of quantum computation in the density matrix representation," in *Quantum Information and Computation II*, vol. 5436, 2004, pp. 285–296.

[29] S. Cheng *et al.*, "Simulating noisy quantum circuits with matrix product density operators," *Phys. Rev. Research*, vol. 3, p. 023 005, 2 2021.

[30] J. Preskill, "Quantum Computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[31] Atos SE, *Quantum learning machine*, Accessed: 2023-01-20, 2016. [Online]. Available: https://atos.net/en/solutions/quantum-learning-machine.

[32] D. Wecker and K. Svore, "LIQUi|>: A software design architecture and domain-specific language for quantum computing," *arXiv:1402.4467*, 2014.

[33] Cirq Developers, "Cirq," version v0.14.1, *Zenodo*, 2022.

[34] T. Jones, A. Brown, I. Bush, and S. Benjamin, "QuEST and high performance simulation of quantum computers," *arXiv:1802.08032*, 2018.

[35] G. G. Guerreschi, J. Hogaboam, F. Baruffa, and N. P. D. Sawaya, "Intel Quantum Simulator: A cloud-ready high-performance simulator of quantum circuits," *Quantum Sci. Technol.*, vol. 5, p. 034 007, 2020.

[36] G. F. Viamontes, I. L. Markov, and J. P. Hayes, "Graph-based simulation of quantum computation in the density matrix representation," *Quantum Info. Comput.*, vol. 5, no. 2, pp. 113–130, 2005.

[37] T. Grurl, J. Fuß, and R. Wille, "Noise-aware Quantum Circuit Simulation With Decision Diagrams," *IEEE Trans. on CAD of Integrated Circuits and Systems*, 2022.

[38] S. S. Tannu and M. K. Qureshi, "Not All Qubits Are Created Equal: A Case for Variability-Aware Policies for NISQ-Era Quantum Computers," in *ASPLOS*, 2019, pp. 987–999.

[39] T. Grurl, J. Fuß, S. Hillmich, L. Burgholzer, and R. Wille, "Arrays vs. decision diagrams: A case study on quantum circuit simulators," in *Int'l Symp. on Multi-Valued Logic*, vol. 50, 2020, pp. 176–181.

[40] A. Zulehner, S. Hillmich, and R. Wille, "How to efficiently handle complex values? Implementing decision diagrams for quantum computing," in *Int'l Conf. on CAD*, Implementation taken from github.com/iic-jku/ddsim, 2019.

[41] A. Li, S. Stein, S. Krishnamoorthy, and J. Ang, "Qasmbench: A low-level quantum benchmark suite for nisq evaluation and simulation," *ACM Transactions on Quantum Computing*, vol. 4, no. 2, 2023.