

Automatic Uniform Quantum State Preparation Using Decision Diagrams

Fereshte Mozafari Mathias Soeken Heinz Riener Giovanni De Micheli
Integrated Systems Laboratory, EPFL, Lausanne, Switzerland

Abstract—Most quantum algorithms assume some specific initial state in superposition of basis states before performing the desired application-specific computations. The preparation of such states itself requires a computation performed by a quantum circuit. In this paper, we investigate the automatic state preparation of a specific subset of quantum states that are uniform superpositions over a subset of basis states, called uniform quantum states. We exploit that such states can be represented using Boolean functions and present a recursive algorithm based on functional decomposition. When using binary decision diagrams as function representation, we can enable fast and scalable quantum state preparation with respect to the size of the decision diagram. We show that the algorithm can find quantum circuits for functions, where state-of-the-art algorithms cannot be applied anymore.

Index Terms—Quantum Computing, Quantum State Preparation, Boolean Functions, Decision Diagrams

I. INTRODUCTION

Quantum computing is concerned with developing computing technology based on the principles of quantum mechanics. In classical computing, a bit is a single piece of information that exists in one of the two classical states 0 and 1. In quantum computing the fundamental unit of information is a quantum bit, or for short qubit. The state of a qubit includes the two basis states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, but unlike a classical bit, the state can also be any superposition of these states $\alpha|0\rangle + \beta|1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, where α and β are two complex amplitudes. The square of the amplitude's absolute value is the probability of the measurement outcome being 0 or 1, therefore, $|\alpha|^2 + |\beta|^2 = 1$. Analogously, the combined state for n qubits is described in terms of 2^n complex-valued amplitudes. Each amplitude corresponds to one of the 2^n possible classical states, again, with the square of the amplitude's absolute value being the probability of resulting in the corresponding classical state after measuring all qubits.

A quantum gate on n qubits describes the time evolution of one quantum state into another one, and is modeled as $2^n \times 2^n$ unitary matrices. A combination of these gates represents a quantum circuit, which describes the interaction of quantum gates with qubits in the quantum computer. A quantum algorithm to solve a specific problem is often described in terms of a quantum circuit. Two well-known quantum algorithms are Grover's search algorithm [1], and Shor's algorithm [2] for period finding that can be used for prime factorization.

In general the initial quantum state is the classical basis state in which all bits are 0. Some quantum algorithms require a specific quantum state at the beginning of the computation. Hence, in addition to the quantum circuit that performs the

quantum algorithm, a specific quantum circuit is required that prepares the desired initial quantum state. Consequently, an efficient quantum state preparation is an important task in quantum compilation.

Some approaches [3]–[8] have been considered in the past to prepare arbitrary quantum states. Since these approaches can generate arbitrary quantum states, the input to such algorithms are 2^n complex-valued amplitudes, which limits their scalability drastically. Further, some of the algorithms require a rather abstract set of gates, and an additional compilation step in order to run on physical quantum computers.

In this paper, we target uniform quantum states. A uniform quantum state is a superposition of a nonempty subset of basis states. In other words, all nonzero amplitudes in such a state have the same value. Uniform quantum states are important because they are considered as initial quantum state for algorithms such as Grover walk [9]. Moreover, many important quantum states are uniform, such as the Bell state, the W state [10], the GHZ state [11], and the uniform superposition of *all* basis states. The W and GHZ states are used as fundamental resources in distributed quantum information processing [12].

Such states can be characterized by a Boolean function where each minterm corresponds to a nonzero amplitude. As a result, the quantum state can be represented in a compact form, if the Boolean function permits a compact representation. In particular, in this paper we propose a quantum state preparation algorithm that works directly on decision diagrams. This enables a scalable quantum state preparation, since many Boolean functions of practical interest have small representations, e.g., in terms of binary decision diagrams (BDDs) [13].

We propose an automatic quantum state preparation algorithm, which takes as input a Boolean function and produces a sequence of multiple-controlled gates. Afterwards, to run on a physical quantum computer, we use decomposition methods to generate a quantum circuit over CNOTs and single-qubit quantum gates. The detailed contributions are summarized as follows:

- Utilizing Boolean functions in order to provide a recursive algorithm
- Proposing an algorithm that works on BDDs to enable a fast execution when the function representation is small (algorithm runs in polynomial time with respect to the number of BDD nodes).
- Reducing the number of elementary quantum gates by removing redundancies in the BDDs as well as applying

a post-optimization technique for the GHZ state.

Experimental results show that the proposed approach can achieve a significant reduction in run-time compared to a state-of-the-art approach which relies on an explicit quantum state representation implemented in IBM's Qiskit quantum programming framework. Moreover, the results show that we can reduce the number of elementary quantum gates over the state of the art.

II. BACKGROUND AND RELATED WORK

In this section, we introduce necessary background on Boolean functions and quantum computation.

A. Boolean Functions

A Boolean function is a function $f : \mathbb{B}^n \rightarrow \mathbb{B}$, where $\mathbb{B} = \{0, 1\}$, which can be represented in terms of its truth table. A truth table is a column vector

$$f = (f(0, \dots, 0, 0), f(0, \dots, 0, 1), \dots, f(1, \dots, 1, 0), f(1, \dots, 1, 1))^T, \quad (1)$$

where each entry corresponds to the function value of one of the input assignments. In this paper, we use f to both refer to the function and its truth table.

The *on-set* and *off-set* of the function are the sets of all input assignments that map to 1 and 0, respectively. We define

$$\begin{aligned} \text{on}(f) &= \{x \in \mathbb{B}^n \mid f(x) = 1\}, \\ \text{off}(f) &= \{x \in \mathbb{B}^n \mid f(x) = 0\}. \end{aligned} \quad (2)$$

We define $|f| = |\text{on}(f)|$ as the number of minterms in f . A *minterm* is an input assignment $x \in \mathbb{B}^n$ for which $f(x) = 1$.

The *positive* and *negative co-factors* of a Boolean function $f(x_{n-1}, \dots, x_0)$ with respect to a variable x_i are obtained by assigning x_i to 1 and 0, respectively, which are defined by

$$\begin{aligned} f_{x_i} &= f(x_{n-1}, \dots, x_{i+1}, 1, x_{i-1}, \dots, x_0), \\ f_{\bar{x}_i} &= f(x_{n-1}, \dots, x_{i+1}, 0, x_{i-1}, \dots, x_0), \end{aligned} \quad (3)$$

respectively. We use co-factors in this paper to compute the influence of each variable on the function's output. To compute the probability of being 1 and 0 for x_i in f , we define

$$p_f(x_i) = \frac{|f_{x_i}|}{|f|} \quad \text{and} \quad p_f(\bar{x}_i) = \frac{|f_{\bar{x}_i}|}{|f|}, \quad (4)$$

respectively. The intuition is that the co-factors partition the function into two halves. Note that $p_f(x_i) + p_f(\bar{x}_i) = 1$. When f is clear from the context, we simply write $p(x_i)$ and $p(\bar{x}_i)$.

B. Quantum Gates and Quantum Circuits

A quantum circuit is a diagram to represent a quantum program. A combinational quantum circuit consists of a sequence quantum gates from left to right that correspond to the passage of time.

Qubits: A qubit models the fundamental unit of information in quantum computing that has two basis states, represented

using $|0\rangle$ and $|1\rangle$. In fact, a qubit can be any superposition of the two basis states, which can be denoted as

$$|\psi\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix}. \quad (5)$$

Here, $\alpha_0, \alpha_1 \in \mathbb{C}$ with $|\alpha_0|^2 + |\alpha_1|^2 = 1$. The squared complex numbers $|\alpha_0|^2$ and $|\alpha_1|^2$ indicate the probability that the quantum state will collapse to the classical state $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ or $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$ after the qubit is measured. Moreover, quantum states over n qubits are represented by

$$|\varphi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad (6)$$

a column vector of 2^n complex values α_i such that $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$. Each squared amplitude $|\alpha_i|^2$ indicates the probability that after measurement the n qubits are in the classical state i .

Quantum states can be combined by applying the Kronecker product to produce larger ones, e.g., $\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \frac{1}{\sqrt{2}} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{2}} (1, 1, 0, 0)^T$, which represents a 2-qubit state that is in the perfect superposition between the classical states 00 and 01 [14].

Quantum gates: Quantum gates are modelled as unitary operations which are applied on the qubits to alter their states. A single-qubit quantum gate acts on a single-qubit, and transforms its state into another state. The single-qubit gates are represented by 2×2 unitary matrices [15], [16].

Since single-qubit states correspond to points on the Bloch sphere [16], quantum gates on a single-qubit correspond to rotations. There are three types of rotation gates R_x , R_y , and R_z regarding the three axis x , y , and z . Each rotation gate is parameterized with a continuous angle $\theta \in \mathbb{R}$:

$$\begin{aligned} R_x(\theta) &= \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, & R_y(\theta) &= \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, \\ R_z(\theta) &= \begin{pmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{pmatrix}. \end{aligned} \quad (7)$$

Quantum gates that act on n qubits are represented in terms of $2^n \times 2^n$ unitary matrices. Some 2-qubit gates that we consider in this work are

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{and} \quad \text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (8)$$

In fact, a CNOT gate consists of one control and one target, the target will be flipped when the control is 1. A SWAP gate interchanges the state of 2 qubits with each other.

Uniformly-controlled rotation gates: In this work, we make use of a family of unitary matrices called *uniformly-controlled single-qubit gates* [3]. These unitary matrices are $2^{n+1} \times 2^{n+1}$ block diagonal matrices of the form

$$U = u_0 \oplus \dots \oplus u_{2^n-1} = \begin{pmatrix} u_0 & & & \\ & \ddots & & \\ & & & \\ & & & u_{2^n-1} \end{pmatrix}, \quad (9)$$

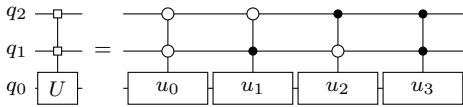


Fig. 1. A uniformly-controlled single-qubit gate with two controls.

where u_0, \dots, u_{2^n-1} are 2×2 unitary matrices.

An example of a uniformly-controlled single-qubit gate with two controls is shown in Fig. 1. The figure also shows the visual representation of the uniformly-controlled single-qubit gate on the left-hand side.

Decomposing an n -qubit uniformly-controlled rotation gate requires an exponential number of elementary gates ($2^n - 1$ rotation gates and $2^n - 2$ CNOTs) [3].

C. Related Works

Although approaches for the preparation of arbitrary quantum states have been proposed in the past [3]–[6], they lead to a large number of gates and additionally in some approaches the detailed decomposition is not presented. Thus, one has to re-design the quantum generation circuit. The quantum state preparation method in [7] is proposed based on unifying phases, probabilities, and finally removing superposition to get basis states. In this manner, it is required to do some swaps in between to reach the state vector which all have same amplitudes and phases. This approach can synthesize arbitrary states, but relies on an explicit representation of the state in terms of a column vector with 2^n entries. It is therefore not scalable for a large n .

State preparation described in [17] derives a lower bound on the number of CNOT gates using three different decomposition methods. But similarly to other related work, this approach works on arbitrary quantum states and requires an explicit representation. The algorithm is implemented for quantum state preparation in IBM’s QisKit framework.

Previous methods are not efficient when the number of qubits is large. We circumvent this problem by using BDDs as a symbolic function representation to describe the input quantum states. The use of BDDs for a compact function representation in quantum circuit synthesis was considered before, e.g., in [18]. However, in that work, the authors targeted synthesis of reversible functions, while we are targeting the preparation of quantum states.

III. PROPOSED APPROACH

In this section, we present a two-step approach to prepare uniform quantum states. In the first step, we use Boolean functions to generate a circuit of uniformly-controlled single-qubit gates. We introduce the problem (Section III.A), present the general idea (Section III.B), and propose a scalable algorithm using decision diagrams (Section III.C). In the second step, we discuss how to obtain a quantum realization executable on a quantum computer (Section III.D).

A. Problem Definition

In this work, we consider n -qubit quantum states that are uniform superpositions over a nonempty subset of the

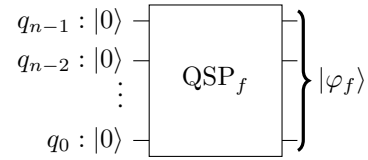


Fig. 2. The problem of quantum state preparation over Boolean function f .

basis states $|0\rangle, |1\rangle, \dots, |2^n - 1\rangle$. In such quantum states all amplitudes of the state vector are either 0 or have the same value $\alpha = 1/\sqrt{s}$, where s is the size of the subset of basis states. We exploit that such states can be characterized by a Boolean function $f : \mathbb{B}^n \rightarrow \mathbb{B}$ such that $f(x) = 1$, if and only if $|x\rangle$ is in the subset of the considered basis states, and therefore its corresponding amplitude is nonzero.

Example 1: The majority-of-three function $f = \langle x_0x_1x_2 \rangle$, where $\langle x_0x_1x_2 \rangle = x_0x_1 \vee x_0x_2 \vee x_1x_2$, and has the truth table $f = (0, 0, 0, 1, 0, 1, 1, 1)^T$. It encodes the uniform quantum state $|\varphi\rangle = \frac{1}{\sqrt{4}}(0, 0, 0, 1, 0, 1, 1, 1)^T$.

We are interested in generating a quantum state $|\varphi_f\rangle$ that corresponds to the normalized truth table of f

$$|\varphi_f\rangle = \frac{f}{\sqrt{|f|}} = \frac{1}{\sqrt{|f|}} \sum_{x \in \text{on}(f)} |x\rangle. \quad (10)$$

Next, we propose an automatic algorithm to find a quantum circuit for generating such a state given as input a Boolean function f in some representations. In the beginning, all states are assumed to be zero, then we are looking for a construction that transforms a given unitary matrix QSP_f into a circuit, where $\text{QSP}_f|0\rangle^{\otimes n} = |\varphi_f\rangle$. Fig. 2 summarizes our problem formulation.

Uniform quantum states can be found in various quantum algorithms, e.g., the uniform superposition of all basis states, for which $f = 1$ (tautology), the Bell state, for which $f = \bar{x}_1 \oplus x_2$, the generalized GHZ state, for which $f = \bar{x}_1 \bar{x}_2 \dots \bar{x}_n \oplus x_1 x_2 \dots x_n$, and the generalized W state, for which $f = [x_1 + x_2 + \dots + x_n = 1]$ that means only one of the variables is one and the others are equal to zero.

B. General Idea

To prepare an n -qubit uniform quantum state $|\varphi_f\rangle$ that encodes the Boolean function f , we define a correspondence between the qubits in the quantum circuit and the variables in f . Without loss of generality, we assume that qubit q_i and the uniform quantum state $|\varphi_f\rangle$ correspond to variable x_i and the Boolean function f , respectively. In the remainder, we will use the symbols interchangeably.

The general idea of our state preparation algorithm relies on the identity

$$\text{QSP}_f|0\rangle^{\otimes n} = (\text{QSP}_{f_{\bar{x}_i}} \oplus \text{QSP}_{f_{x_i}})(G(p_f(\bar{x}_i)) \otimes I_{2^{n-1}})|0\rangle,$$

where $G(p)$ is a unitary transformation gate that satisfies

$$G(p)|0\rangle = \sqrt{p}|0\rangle + \sqrt{1-p}|1\rangle. \quad (11)$$

To prepare the i^{th} qubit, a quantum gate $G(p_f(\bar{x}_i))$, is applied. This gate results in the probability of being zero

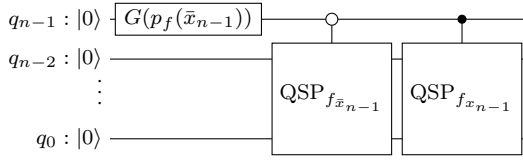


Fig. 3. The general idea.

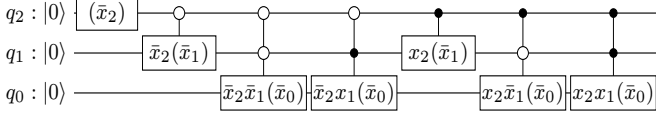


Fig. 4. The abstract quantum gates of $\text{QSP}_{\langle x_0 x_1 x_2 \rangle}$.

after measurement for x_i , which corresponds to q_i , in the Boolean function f . Next, the circuit is divided into two parts corresponding to the negative co-factor and positive co-factor, a negative control is added to the negative co-factor, and a positive control is added to the positive co-factor. Fig. 3 shows the application of this identity in the quantum circuit model for $i = n - 1$. By applying this identity recursively for all variables, we obtain the desired quantum circuit.

Example 2: Fig. 4 shows the result of the proposed recursive algorithm for the majority function $f = \langle x_0 x_1 x_2 \rangle$ from Example 1. In the figure, for the sake of readability, we use a shorthand notation to label boxes. For instance, the label $x_2 \bar{x}_1 (\bar{x}_0)$ denotes the unitary transformation gate $G(p_{f_{x_2 \bar{x}_1}}(\bar{x}_0))$.

Notice that in Example 2, the two gates with target q_1 can be moved next to each other, since the right-most gate commutes with the two gates left of it, with target q_0 . This movement allows us to utilize uniformly-controlled single-qubit gates. We generalize this idea which allows us to retrieve a generic quantum circuit with n uniformly-controlled single-qubit gates as shown in Fig. 5. The box labeled with $G(p(\bar{x}_i))$ denotes the probability of being zero for the variable x_i in the cofactors over previously prepared qubits. For example $G(p(\bar{x}_{n-2}))$ corresponds to $G(p_{f_{\bar{x}_{n-1}}}(\bar{x}_{n-2}))$ and $G(p_{f_{x_{n-1}}}(\bar{x}_{n-2}))$.

C. Approach using Decision Diagrams

In practice it is infeasible to store the Boolean function as a truth table for a large number of variables (typically more than 15 variables). As an alternative, the proposed approach can extract the quantum circuit for QSP_f , when f is represented as reduced ordered binary decision diagram (ROBDD, or BDD for short). This is due to the fact that counting all minterms and computing co-factors can be efficiently performed using BDDs. The compact representation not only enables a scalable quantum state preparation, if the BDD representation for f is small, but also can reduce the number of multiple-controlled gates.

We illustrate the BDD-based synthesis algorithm using the majority-of-three function as an example in Fig. 6. The BDD consists of 4 nodes. We traverse the BDD from bottom to top to first count the number of ones for each node, then from top to bottom to compute probabilities by dividing the number of ones from the low-child over the number of ones of the current

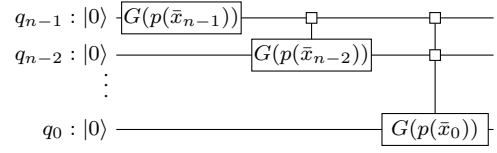


Fig. 5. The general structure of the proposed algorithm.

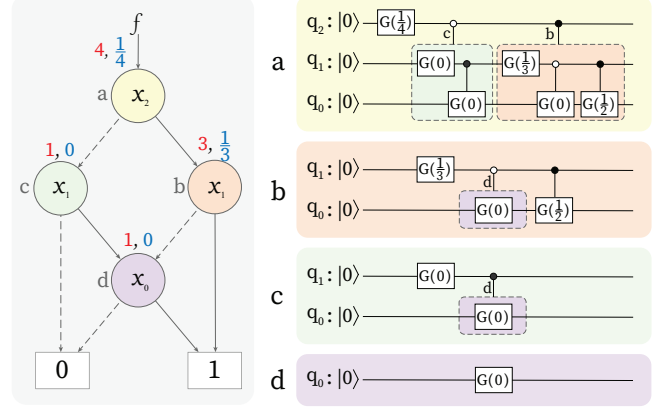


Fig. 6. BDD for $\langle x_0 x_1 x_2 \rangle$ and the procedure of extracting gates for each node from bottom to top.

node. The number of ones and probabilities for each node are shown within the figure in red and blue colors, respectively.

To construct multiple-controlled gates, we again traverse the BDD from bottom to top. In a recursive manner, we construct for each node a circuit that consists of one $G(p)$ gate, and a negative-controlled application of the circuit constructed by the low-child, as well as a positive-controlled application of the circuit constructed by the high-child. These gates are located in the figure using boxes in the right-hand side for each node with the same color. Special consideration is required for children that contain missing nodes on their path. In that case, we need to insert a $G(\frac{1}{2})$ gate for each of the missing nodes on that path which leads to the both 0 and 1 values with half probabilities for those nodes. Note that we do not consider this case for the 0 terminal. To provide more details, we explain the construction of the gates for node ‘b’. First, we have to apply $G(\frac{1}{3})$ to the qubit corresponding to ‘b’, then we connect the negative-control to the gates of node ‘d’ and the positive-control to a newly inserted $G(\frac{1}{2})$ for the missing node x_0 on this path.

Consequently, the BDD-based synthesis algorithm is linear in the number of nodes in order to construct the partial circuits for each node. Depending on the number of paths in the BDD, the monolithic circuit composed of all partial circuits may become exponential, however, the use of symbolic quantum circuit representations can prevent this.

D. Quantum Gate Realization

From the definition of $R_y(\theta)$ one can readily derive that

$$G(p) = R_y(2 \cos^{-1}(\sqrt{p})). \quad (12)$$

Consequently, by replacing all gates on the target line by R_y gates in Fig. 5, we obtain a circuit consisting only of multiple-

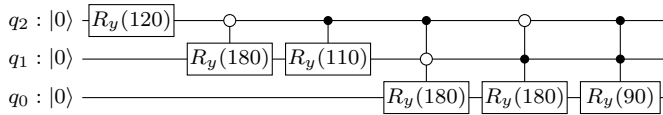


Fig. 7. Final majority-of-three quantum circuit consisting of multiple-controlled R_y rotation gates.

controlled R_y (MC- R_y) rotation gates. The resulting circuit for majority-of-three is shown in Fig. 7.

To obtain a circuit that can be run on a physical quantum computer, we need to decompose MC- R_y gates into elementary quantum gates. We utilize the decomposition method presented in [19] to synthesize uniformly-controlled rotation gates directly into a sequence of CNOT gates and R_y rotations. The decomposition of the i^{th} qubit yields at most 2^{i-1} R_y rotation gates and 2^{i-1} CNOTs. As a result, the quantum state preparation affords at most $2^n - 1$ R_y gates and $2^n - 2$ CNOTs. While, as we reduce the number of MC- R_y s using decision diagrams, it is better to decompose each MC- R_y , separately to reduce the number of elementary quantum gates. In this manner, we employ the decomposition method presented in [20]. Note that this method makes use of some helper qubits in order to reduce the number of elementary quantum gates. To avoid using helper qubits, we can use [21].

IV. EXPERIMENTAL EVALUATION

In this section, we discuss the experimental setup and results.

A. Experimental Setup

We implemented the proposed approach into the C++ library for quantum state preparation *angel*.¹ We used the *CUDD* library for BDD representation and traversal. To compare our results with the state of the art, we make use of *IBM's QisKit* [22] that implemented the algorithm of [17]. We performed experiments for the two most-known quantum states GHZ and W. As there is no standard benchmark set for quantum state preparation and our method utilizes Boolean functions, we further used the ISCAS benchmarks as practical benchmarks to extract large functions with different numbers of variables which correspond to the number of qubits (n). Since ISCAS benchmarks have multiple outputs, we extracted the logic cone for a given primary output. All experiments have been conducted on an Intel Core i7, 2.7 GHz with 16 GB memory.

B. Results and Discussion

Experimental results are shown in Table I. The first column names the benchmark. The suffixes for the ISCAS benchmarks correspond to the indices of the extracted logic cones. We evaluate the proposed method for run-time and circuit size. We future discuss a way to optimize the number of gates.

Run-time: We track the actual run-time, and also report the number of nodes in the BDD, since the algorithm's complexity

depends on the number of nodes. There are two columns in Table I that show run-time for the proposed method and the state of the art. Experimental results show that our proposed method reduced the run-time significantly for all cases. Moreover, as the proposed method uses decision diagrams in the implementation, the number of nodes is extracted only for the proposed method. A timeout of 9000 seconds is considered to extract the results. The results show that generating circuits for quantum state preparation using the proposed method is fast. When the number of qubits grows too large, approaches based on explicit state representation require too much time (see the cells marked TO).

Circuit size: Experimental results regarding circuit size are evaluated in terms of the number of MC- R_y s, the number of elementary quantum gates (CNOT and single-qubit quantum gates), and helper qubits. These results are summarized in Table I for both the proposed method and the state-of-the-art approach. Note that we only extracted MC- R_y s for the proposed method. The number of MC- R_y s is reduced using decision diagrams by removing redundancies. As shown in the table, we reduced the number of MC- R_y s from $2^n - 2$ into $n - 1$ for GHZ and W. Moreover, MC- R_y s are reduced for ISCAS benchmarks instead of growing exponentially. As we generate MC- R_y s, our algorithm has the potential to represent these gates in a compact way in terms of Boolean functions. It is an advantage of the proposed approach to provide the high-level representations for MC- R_y gates such that different decomposition methods may be applied later.

We transform the MC- R_y s into elementary quantum gates using decomposition methods that were explained in Section III.D. The method presented in [20] requires helper qubits that are shown in the table. The results show that reducing CNOTs and single-qubit gates (R_y s for the proposed method) is comparable over the state of the art. As we discussed in Section III.D, the upper bound on CNOTs and single-qubit gates are $2^n - 2$ and $2^n - 1$, respectively. The results show that we reduce the number of elementary quantum gates as much as possible while for the state of the art are close to the upper bounds.

Optimization: To prepare the GHZ quantum state, our method generates one MC- R_y on each line with positive controls. Fig. 8 on the left-hand side shows this sequence of gates for GHZ state on 4 qubits. As all qubits in the beginning are assumed to be 0, each line can alter to 1 only in one case, when the corresponding controls are 1. Hence, last control is 1 when previous controls are 1. Hence, we can reduce redundant controls and only keep the last control, which results in a sequence of $n - 1$ single-controlled R_y gates (decomposing these gates results $n - 1$ CNOTs and one R_y rotation gate). The optimized circuit for GHZ on 4 qubits is shown on the right-hand side of Fig. 8.

V. CONCLUSIONS

We have presented an algorithm to generate circuits for the preparation of uniform quantum states. This subclass is motivated by several quantum states in the literature which are

¹github.com/fmozafari/angel

TABLE I
EXPERIMENTAL RESULT REGARDING NUMBER OF MC- R_y ROTATION GATES, ELEMENTARY QUANTUM GATES AND TIME.

QSS	Qubits	Nodes	Proposed approach				State-of-the-art approach			
			MC- R_y s	CNOTs	Single-qubit gates	Helper qubits	Time	CNOTs	Single-qubit gates	Time
GHZ	15	29	14	539	720	6	< 0.01	32752	97857	60.57
GHZ	18	35	17	809	1083	8	< 0.01	262125	786071	929.03
GHZ	20	39	19	1027	1389	9	< 0.01	1048555	3145488	7809.02
GHZ	27	53	26	1951	2628	12	< 0.01	TO	TO	TO
GHZ	30	59	29	2437	3279	14	< 0.01	TO	TO	TO
W	15	29	14	539	720	6	< 0.01	32752	98175	104.8
W	18	35	17	809	1083	8	< 0.01	262125	786255	5087.05
W	27	53	26	1951	2628	12	< 0.01	TO	TO	TO
W	30	59	29	2437	3279	14	< 0.01	TO	TO	TO
c17-0	4	6	11	14	15	1	< 0.01	11	39	0.03
c17-1	4	6	8	14	15	1	< 0.01	11	37	0.03
c432-0	18	18	2024	103074	126514	8	< 0.01	262125	3399580	1801.26
c432-1	27	3795	1256482	98417906	108854256	12	16.35	TO	TO	TO
c7552-65	29	7389	2847926	241536510	277309441	13	45.34	TO	TO	TO
c7552-66	26	3501	681374	45046398	48877056	12	9.21	TO	TO	TO
c7552-67	23	1653	163474	6994910	7928448	10	1.73	TO	TO	TO
c7552-68	20	793	41188	1048574	1048575	9	0.37	1048576	3015488	5948.47

TO: time-out of 9000 seconds.

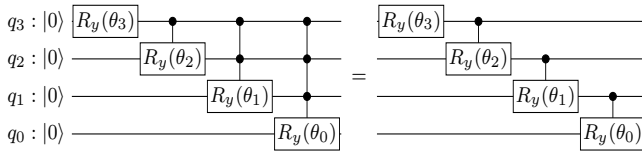


Fig. 8. Reducing the number of controls for GHZ state.

contained in this family. The advantage of such states is that they can be characterized by a Boolean function, and therefore permit a compact representation, if the Boolean function has a compact representation—e.g., in terms of a BDD. We have shown a recursive algorithm to generate a quantum circuit based on uniformly-controlled R_y rotation gates. Our algorithm provides a scalable quantum state preparation as well as reduces the number of multiple-controlled gates. **Acknowledgments:** This research was supported by the Swiss National Science Foundation (200021-169084 MAJesty).

REFERENCES

- [1] L. K. Grover, "A fast quantum mechanical algorithm for database search," *arXiv preprint quant-ph/9605043*, 1996.
- [2] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. IEEE, 1994, pp. 124–134.
- [3] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, "Quantum circuits with uniformly controlled one-qubit gates," *Physical Review A*, vol. 71, no. 5, p. 052330, 2005.
- [4] P. Kaye and M. Mosca, "Quantum networks for generating arbitrary quantum states," *arXiv preprint quant-ph/0407102*, pp. 1–3, 2004.
- [5] M. Mottonen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, "Transformation of quantum states using uniformly controlled rotations," *arXiv preprint quant-ph/0407010*, 2004.
- [6] V. V. Shende, S. S. Bullock, and I. L. Markov, "Synthesis of quantum-logic circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 25, no. 6, pp. 1000–1010, 2006.
- [7] P. Niemann, R. Datta, and R. Wille, "Logic synthesis for quantum state generation," in *2016 IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL)*. IEEE, 2016, pp. 247–252.
- [8] M.-X. Luo, S.-Y. Ma, Y. Deng, and X. Wang, "Deterministic generations of quantum state with no more than six qubits," *Quantum Information Processing*, vol. 14, no. 3, pp. 901–920, 2015.
- [9] N. Shenvi, J. Kempe, and K. B. Whaley, "Quantum random-walk search algorithm," *Physical Review A*, vol. 67, no. 5, p. 052307, 2003.
- [10] W. Dür, G. Vidal, and J. I. Cirac, "Three qubits can be entangled in two inequivalent ways," *Physical Review A*, vol. 62, no. 6, p. 062314, 2000.
- [11] D. M. Greenberger, M. A. Horne, and A. Zeilinger, "Going beyond Bell's theorem," in *Bell's theorem, quantum theory and conceptions of the universe*. Springer, 1989, pp. 69–72.
- [12] E. D'Hondt and P. Panangaden, "The computational power of the W and GHZ states," *arXiv preprint quant-ph/0412177*, 2004.
- [13] R. Bryant, "Graph-based algorithms for boolean function manipulation," *IEEE Transactions on Computers*, vol. 100, no. 8, pp. 677–691, 1986.
- [14] B. Schumacher, "Quantum coding," *Physical Review A*, vol. 51, no. 4, p. 2738, 1995.
- [15] R. P. Feynman, "Quantum mechanical computers," *Foundations of physics*, vol. 16, no. 6, pp. 507–531, 1986.
- [16] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
- [17] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, "Quantum circuits for isometries," *Physical Review A*, vol. 93, no. 3, p. 032318, 2016.
- [18] R. Wille and R. Drechsler, "BDD-based synthesis of reversible logic for large functions," in *Proceedings of the 46th Annual Design Automation Conference*. ACM, 2009, pp. 270–275.
- [19] M. Möttönen, J. J. Vartiainen, V. Bergholm, and M. M. Salomaa, "Quantum circuits for general multiqubit gates," *Physical Review Letters*, vol. 93, no. 13, p. 130502, 2004.
- [20] D. Maslov, "Advantages of using relative-phase Toffoli gates with an application to multiple control Toffoli optimization," *Physical Review A*, vol. 93, no. 2, p. 022311, 2016.
- [21] M. Soeken, F. Mozafari, B. Schmitt, and G. De Micheli, "Compiling permutations for superconducting qpus," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 1349–1354.
- [22] G. A. Héctor Abraham, Ismail Yunus Akhalwaya and et al., "QisKit: An open-source framework for quantum computing," 2019.