Circuits for Measurement Based Quantum State Preparation

Niels Gleinig
Department of Computer Science
ETH Zurich
Zurich, Switzerland
niels.gleinig@inf.ethz.ch

Torsten Hoefler
Department of Computer Science
ETH Zurich
Zurich, Switzerland
torsten.hoefler@inf.ethz.ch

***Abstract*—In quantum computing, state preparation is the problem of synthesizing circuits that initialize quantum systems to specific states. It has been shown that there are states that require circuits of exponential size to be prepared (when not using measurements), and consequently, despite extensive research on this problem, the existing computer-aided design (CAD) methods produce circuits of exponential size. In this paper, we show how CAD based state preparation can be made scalable by using techniques that are unique to quantum computing: measurements, and the resulting state collapses. With this approach, we are able to produce wide classes of states in polynomial time, resulting in an exponential improvement over existing CAD methods.**

***Index Terms*—quantum computing, quantum compilation, quantum state preparation**

I. Introduction

Quantum computing is an emerging technology that promises to revolutionize many fields by solving important computational problems asymptotically faster than classical computers. With Shor's algorithm [1] we can compute prime factorizations in polynomial time and with quantum linear system solvers [2] we can solve certain linear systems in time which is logarithmic in the number of variables. Opposing these theoretical breakthroughs, existing quantum hardware is very noisy and limited in size [3], making it imperative to sharply optimize all of the circuits deployed for the different parts of quantum computations: state preparation, evaluation of oracle functions, measurements. In this paper we focus on state preparation, the problem of synthesizing circuits that initialize quantum systems to specfic states. State preparation is needed to prepare input states which encode problem instances on which then a quantum algorithm is applied.

Many (and arguably most) of the existing quantum algorithms that achieve exponential speedups over classical algorithms assume that the input state is already given. However, this assumption is problematic since many quantum states take exponential time to be synthesized [4], [5]. Hence, state preparation (along with "reading outputs") becomes a bottleneck and a major obstacle towards outperforming classical computers. This "bottlenecked on input and output operations" is the case for Quantum Fourier Transform [6] and Quantum Linear System Solvers [2], just to mention two among many examples.

The computer-aided design (CAD) community has developed many interesting synthesis and optimization techniques for quantum circuits [7]–[12], including quantum circuits for state preparation [5], [13]–[19], but none of these can guarantee to produce in general circuits that are smaller than exponential. Quantum phenomena like superposition, entanglement, no-cloning, and state collapses make optimizations of quantum circuits challenging and counterintuitive, but as we will show in this paper, they also create powerful optimization opportunities that do not exist in classical computing.

In this paper we present a novel CAD method for state preparation that is based on measurements. First, we explicitly consider uniform quantum states, which are an important subclass of quantum states that arise for example in *quantum simulated annealing* [20] (where they are the instance-specific initial states that need to be prepared) and as *q-sampling states* [21] (where they encode probability distributions). Using measurements we are able to prepare many uniform states in polynomial time and we formulate a theorem that specifies which uniform states can be prepared efficiently with the proposed method. We then extend this approach and show how measurements can be used to prepare arbitrary quantum states. Again we give examples of general states that can be prepared in polynomial time and we formulate another theorem to specify which states can be prepared efficiently with our method. Finally, we experimentally verify our method in Qiskit. Using a quantum computing simulator we show that even under the influence of noise, the smaller size of our circuits allow us to prepare states which will be on expectation closer to the states that we wish to prepare than those prepared by general state preparation methods.

A. Background

1) Quantum States: For any $x \in \{0,1\}^n$, an n-qubit quantum system can be in the state $|x\rangle$. These *basis states* can be considered an orthonormal basis of a 2^n-dimensional vector space. A general quantum state can be a *superposition* of basis states. That is, it can be any normalized vector

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} c_x |x\rangle , \tag{1}$$

with $\sum_{x\in\{0,1\}^n} |c_x|^2 = 1$, in the vector space that is spanned by the basis states, and the *amplitudes* c_x are complex numbers. A quantum state is *uniform* if all its non-zero amplitudes are the same real, positive number. That is, a uniform quantum state

can be written like $|f\rangle := \frac{1}{\sqrt{|f|}} \sum_{x\in\{0,1\}^n, f(x)=1} |x\rangle$ for some boolean function $f : \{0,1\}^n \longrightarrow \{0,1\}$ and $|f| := |\{x \in \{0,1\}^n, f(x) = 1\}|$, or equivalently $|F\rangle := \frac{1}{\sqrt{|F|}} \sum_{x\in F} |x\rangle$ for some subset $F \subset \{0,1\}^n$.

2) Measurements and State Collapses: When we measure all qubits, the probability of observing a particular basis state $|x\rangle$ is given by $|c_x|^2$. Further, when we measure a single qubit q we observe either $|0\rangle$ or $|1\rangle$. The probability of each observation is equal to the sum of the probabilities $|c_x|^2$ over all basis states $|x\rangle$ with $x[q] = 0$ or $x[q] = 1$ respectively. That is, the probability of observing $|0\rangle$ is $\sum_{x\in\{0,1\}^n, x[q]=0} |c_x|^2$ and the probability of observing $|1\rangle$ is $\sum_{x\in\{0,1\}^n, x[q]=1} |c_x|^2$. Once we observe a value, the state *collapses* to a new state in which only those basis states "remain" that match the observation. The basis states that remain have now amplitudes that are proportional to the amplitudes they had before the measurement, whereas the non-matching basis states have now amplitude 0. So for example, if we have a 2-qubit quantum system in the state $\sqrt{\frac{1}{9}}|00\rangle - \sqrt{\frac{2}{9}}|01\rangle + \sqrt{\frac{3}{9}}|10\rangle + \sqrt{\frac{3}{9}}|11\rangle$ and we measure the first qubit, we observe $|0\rangle$ with probability $\left|\sqrt{\frac{1}{9}}\right|^2 + \left|-\sqrt{\frac{2}{9}}\right|^2 = \frac{3}{9}$. If we observe $|0\rangle$, the state collapses to $\frac{1}{\sqrt{3/9}}\left(\sqrt{\frac{1}{9}}|00\rangle - \sqrt{\frac{2}{9}}|01\rangle\right)$, where the normalization factor $\frac{1}{\sqrt{3/9}}$ makes it again a valid quantum state. If we now measure again the first qubit, we observe $|0\rangle$ with probability 1.

B. Embedding (boolean) functions into reversible circuits

Notice that for any given boolean function $f : \{0,1\}^n \longrightarrow \{0,1\}$ the transformation $(x, y) \mapsto (x, f(x) + y)$, $x \in \{0,1\}^n$, $y \in \{0,1\}$, is reversible and hence we can always synthesize a quantum circuit U_f that is defined by acting on the basis states as $U_f|x\rangle|y\rangle = |x\rangle|f(x) + y\rangle$. Since U_f only changes the state $|y\rangle$ of the $(n+1)$-st qubit, we say that U_f *targets* the $(n+1)$-st qubit. Synthesizing and optimizing these circuits is a task that has been widely investigated by the quantum computing community [9], [11].

C. Problem formulation

In this paper we consider the following general problem.

Quantum State Preparation: Given a quantum state

$$|\psi\rangle = \sum_{x\in\{0,1\}^n} c_x |x\rangle, \tag{2}$$

find a quantum circuit $C = g_1 g_2 \dots g_s$, where the g_i are either primitive gates (that is, unitary transformations on 1 or 2 qubits) or measurements, such that $C|0\rangle = |\psi\rangle$.

Before investigating this most general form of the state preparation problem, we will consider the special case when the quantum state is uniform.

D. Previous work

Many CAD methods have been developed for synthesizing and optimizing quantum circuits that are built out of classical gates (that is, gates that map basis states to basis states, like NOT, CNOT, Toffoli etc.) [7]–[12]. There also exist several methods for the problem of quantum state preparation (which cannot be solved using only classical gates). Shende et al. [13], Möttönen et al. [5], Kaye and Mosca [14], Araujo et al. [17], and Niemann et al. [15] propose different methods for general state preparation, however, none of these methods can guarantee to produce in general circuits of size less than $O(2^n)$. Mozafari et al. [16], [18] present methods for preparing uniform states, which also produce circuits of size $O(2^n)$.

State preparation has also been of much theoretical interest [22] as it has been recognized that polynomial time preparation methods for certain states would lead to solutions of some of the central open problems of quantum computing [21]. Unfortunately, it has also been found that [5] there are states that require at least $2^{n+1} - 2$ one-qubit-gates, and $\lceil \frac{1}{4}(2^{n+1} - 3n - 8) \rceil$ CNOT-gates when not using measurements. Hence, there cannot exist general state preparation methods without measurements that produce in general circuits with a total gate count of less than $\left\lceil \frac{1}{4}(5 \cdot 2^{n+1} - 3n - 10) \right\rceil$. Given these theoretical lower bounds, many existing preparation methods [5], [13]–[15] have indeed a worst-case complexity that is optimal for the general problem. However, whenever possible, state preparation should not be tackled with general "off-the-shelf" methods that work for all states in the same way, but instead the solution should adapt to specific properties of the given state as this will typically result in asymptotic improvements. For example, Gleinig and Hoefler [23] prepare states in time which is polynomial in the number of non-zero amplitudes and qubits, resulting in efficient preparation of sparse states. Following this line of research, the goal of this paper is to investigate efficient preparation for a complementary class of states.

It has been conjectured that there are states that can be prepared in polynomial time when allowing measurements, but that cannot be prepared in polynomial time without measurements (see for example chapter 3.3 of Aaronson [24]). Yet, to the best of our knowledge, Grover's work [19] is the only one that investigates measurement based state preparation (MBSP) of general states.

In this paper we present a novel approach to MBSP. Unlike Grover, we do not use amplitude amplification. This can increase the worst-case expected runtime quadratically, but it can decrease the size of the circuits exponentially. This makes our approach more resillient to noise, which under the current limitations of quantum hardware is more important than the actual runtime. Also compared to the previously mentioned state preparation methods we obtain exponential improvements on wide classes of states.

II. Uniform Quantum States

Uniform quantum states provide the most intuitive understanding of how MBSP works and yet they are an important class of states for which preparation has been investigated in the past [16], [18]. The general MBSP that we develop later is a generalization that can be obtained by extending to $\mathbb{C}$-valued

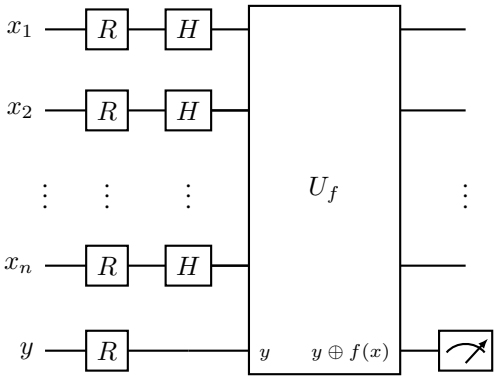

Fig. 1: The circuit R_f that is used by Algorithm 1. We let R denotes the "reset-operation", that is, performing a measurement and if we observe 1 apply a NOT gate.

functions what we present in this chapter for boolean functions. We start describing the actual method, then we analyse its complexity, and finally we discuss two examples.

A. Algorithmic description of MBSP

The idea is to prepare a state on which we can perform a measurement which will make the state collapse to the desired state. If we are in an $(n+1)$-qubit system and start with all qubits in the $|0\rangle$-state (which we denote $|0^n\rangle$), then applying Hadamard gates on each of the first n qubits and applying U_f targeting the $(n+1)$-st qubit, we obtain the state

$$U_f(H^{\otimes n} \otimes I)\,|0^{n+1}\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x\rangle\,|f(x)\rangle\,. \qquad (3)$$

If we now measure the $(n+1)$-st qubit we observe either $|0\rangle$ or $|1\rangle$. If we observe $|1\rangle$ the state collapses to

$$\frac{1}{\sqrt{|f|}} \sum_{x\in\{0,1\}^n, f(x)=1} |x\rangle\,|1\rangle\,, \qquad (4)$$

which is on the first n qubits identical to $|f\rangle$ as a pure state. However, if we measure $|0\rangle$ we collapse to a state from which it is more difficult to get to $|f\rangle$ than from the initial $|0\rangle$. Hence, we go back to $|0^{n+1}\rangle$ by measuring all qubits and applying NOT to those in state $|1\rangle$. Now we can start from the beginning. Repeating these steps we eventually observe $|1\rangle$, and hence obtain the desired state. This method to prepare the state is summarized as Algorithm 1. Figure 1 illustrates the circuit that is used by this algorithm. From now on we will let R_f denote this circuit, $s_f(n)$ the size of (optimal) classical circuits that compute f (as a function of the input size n) and $p_f(n) = \frac{|f|}{2^n}$ the proportion of inputs that are mapped to 1. The following theorem discusses the properties of Algorithm 1.

Theorem 1. *If $f : \{0,1\}^n \longrightarrow \{0,1\}$ is a function that can be computed with a (classical) circuit of size $s_f(n)$ and for which a proportion $p_f(n)$ of inputs are mapped to 1, then Algorithm 1 prepares the state $|f\rangle$ with a circuit of size $O(n + s_f(n))$ and an expected time of $O\left(\frac{n+s_f(n)}{p_f(n)}\right)$.*

Proof. We first analyze how large the circuit R_f is. The initial reseting and the application of Hadamard gates is done with a circuit of size $O(n)$. Since f can be computed with a classical circuit of size $s_f(n)$, the operator U_f can also be implemented with a quantum circuit of size $O(s_f(n))$. Hence, the circuit R_f has size $O(n+s_f(n))$. To analyze how often we need to apply this circuit, notice that the probability of observing $|1\rangle$ is given by $\sum_{x\in\{0,1\}^n, f(x)=1} \left|\frac{1}{\sqrt{2^n}}\right|^2 = \frac{|f|}{2^n} = p_f(n)$. Hence, according to Lemma 1 it takes on expectation $O\left(\frac{1}{p_f(n)}\right)$ repetitions until we observe $|1\rangle$. □

It follows from this theorem that with Algorithm 1 we can prepare $|f\rangle$ in polynomial time whenever f is a function that can be computed in $\mathrm{poly}(n)$ time and has a proportion of $\Omega(1/\mathrm{poly}(n))$ non-zeros.

ALGORITHM 1: Given an $(n+1)$-qubit quantum system and a circuit U_f that computes a boolean function f, this algorithm prepares the state $\sum_{x\in\{0,1\}^n, f(x)=1} |x\rangle$

```
do{
    Reset all qubits to |0^{n+1}⟩;
    Apply Hadamards on the first n qubits;
    Apply U_f targeting last qubit;
    Measure last qubit;
}while{observed |0⟩}
```

Example 1: Let

$$\mathrm{par}(x) := \begin{cases} 1 & \text{if an odd number of bits are set in } x \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$

denote the parity function and consider the state $|par\rangle = \frac{1}{\sqrt{|par|}} \sum_{x\in\{0,1\}^n, \mathrm{par}(x)=1} |x\rangle$. If we want to generate this state with our method we need to synthesize a circuit that computes $U_{par}\,|x\rangle\,|y\rangle = |x\rangle\,|y + \mathrm{par}(x)\rangle$ to implement R_f. This circuit can be constructed with n CNOT gates: one controlled on each of the first n lines and targeting the last line. Since half of the boolean strings of any given length have parity equal to 1, we have $\frac{|par|}{2^n} = \frac{1}{2}$ and hence the probability of measuring 1 is $\frac{1}{2}$. Consequently, the expected number of times we need to apply this circuit until obtaining $|par\rangle$ is $\frac{1}{\frac{1}{2}} = 2$.

Example 2: Now consider the uniform superposition over multiples of some number $q \in \{2, 3, \ldots, 2^n - 1\}$. That is, consider the uniform superposition over all basis states $|x\rangle$ for which number(x) is divisible by q. Letting

$$f_q(x) := \begin{cases} 1 & \text{if } q \text{ divides number}(x) \\ 0 & \text{otherwise} \end{cases} \qquad (6)$$

we can write this state as $|f_q\rangle = \frac{1}{\sqrt{|f_q|}} \sum_{x\in\{0,1\}^n, f_q(x)=1} |x\rangle$. Using [25] we can implement a circuit U_{f_q} that computes f_q. We can then use this circuit to construct the circuit R_{f_q} to prepare $|f_q\rangle$ with our method. Notice that since every q-th integer is divisible by q, we have for this state $\frac{|f_q|}{2^n} \approx \frac{1}{q}$ and hence the expected number of times that we need to apply the circuit is $\approx q$.

III. Non uniform states

Now we consider a general quantum state $|\phi\rangle$ which may be non-uniform. That is,

$$|\phi\rangle = \sum_{x\in\{0,1\}^n} c_x |x\rangle, \tag{7}$$

where $x \in \{0,1\}^n \mapsto c_x \in \mathbb{C}$ is a $\mathbb{C}$-valued function. Now let U_c' be a circuit on $n+1$ qubits that maps $U_c' : |x\rangle|0\rangle \mapsto |x\rangle(\sqrt{1-|c_x|^2}|0\rangle + c_x|1\rangle)$ for any $x \in \{0,1\}^n$. Using this circuit we can prepare

$$U_c'(H^{\otimes n}\otimes I)|0^{n+1}\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle(\sqrt{1-|c_x|^2}|0\rangle + c_x|1\rangle). \tag{8}$$

If we now measure the last qubit and observe $|1\rangle$, the state of the first n qubits collapses to $\sum_{x\in\{0,1\}^n} c_x|x\rangle$. The problem with this approach is that the probability of observing $|1\rangle$ is given by $\sum_{x\in\{0,1\}^n}\left|\frac{c_x}{\sqrt{2^n}}\right|^2 = \frac{1}{2^n}\sum_{x\in\{0,1\}^n}|c_x|^2 = \frac{1}{2^n}$, and hence the expected number of times we need to prepare state (8) until we measure $|1\rangle$ is 2^n. Now we will see how we can improve this by "scaling up" the coefficients c_x. First, we define $\hat{c} := \max_{x\in\{0,1\}^n}|c_x|$ and consider an operator U_c which maps $U_c : |x\rangle|0\rangle \mapsto |x\rangle\left(\sqrt{1-\left|\frac{c_x}{\hat{c}}\right|^2}|0\rangle + \frac{c_x}{\hat{c}}|1\rangle\right)$. Using U_c we can prepare the state $U_c(H^{\otimes n}\otimes I)|0^{n+1}\rangle = \frac{1}{\sqrt{2^n}}\sum_{x\in\{0,1\}^n}|x\rangle\left(\sqrt{1-\left|\frac{c_x}{\hat{c}}\right|^2}|0\rangle + \frac{c_x}{\hat{c}}|1\rangle\right)$.

If we now measure the last qubit and observe $|1\rangle$, the state of the first n qubits again collapses to the desired state

$$\frac{1}{\sqrt{\sum_{x\in\{0,1\}^n}|\frac{c_x}{\sqrt{2^n}\hat{c}}|^2}}\sum_{x\in\{0,1\}^n}\frac{c_x}{\sqrt{2^n}\hat{c}}|x\rangle = \sum_{x\in\{0,1\}^n} c_x|x\rangle. \tag{9}$$

However, using the operator U_c instead of U_c' has the advantage that the probability of observing $|1\rangle$ increases to $\sum_{x\in\{0,1\}^n}\left|\frac{c_x}{\sqrt{2^n}\hat{c}}\right|^2 = \frac{1}{2^n|\hat{c}|^2}\sum_{x\in\{0,1\}^n}|c_x|^2 = \frac{1}{2^n\hat{c}^2}$, and the expected number of times we need to apply U_c has decreased to $2^n\hat{c}^2$, which will be typically much smaller than 2^n because $\hat{c}$ is the absolute value of an amplitude. We summarize these insights in the following theorem.

Theorem 2. *Suppose we can implement U_c with a quantum circuit of size S. Then we can prepare the state (7) in time which is on expectation $2^n\cdot\hat{c}^2\cdot S$.*

Consequently, we can prepare a general state in polynomial time whenever U_c can be implemented by circuits of poly(n) size and $\hat{c} \in O\left(\frac{poly(n)}{\sqrt{2^n}}\right)$.

A. Example: Sampling Assignments for MaxSat

In the **MaxSat** problem we are given a boolean formula $\rho(x_1,\ldots,x_n) = \rho_1(x_1,\ldots,x_n)\wedge\ldots\wedge\rho_d(x_1,\ldots,x_n)$ in conjunctive normal form and the goal is to find a variable assignment $x_1,\ldots,x_n$ for which a maximal number of clauses is satisfied. It is well known that sampling a variable assignment uniformly at random we satisfy on expectation a number of clauses which is at least $\frac{1}{2}$ the optimum. We will now show how we can use the method presented in this section to prepare a state which is a superposition over variable assignments where the assignments that satisfy more clauses have larger amplitudes (i.e., higher probability of being observed). In order to construct an appropriate operator U_c we go one by one over the clauses of the expression and for each one of them we construct a circuit that applies the gate $R_{\frac{\pi}{2d}} = \begin{bmatrix}\cos\left(\frac{\pi}{2d}\right) & -\sin\left(\frac{\pi}{2d}\right)\\ \sin\left(\frac{\pi}{2d}\right) & \cos\left(\frac{\pi}{2d}\right)\end{bmatrix}$ on the ancilla qubit whenever the clause is satisfied. By applying these circuits, to any initial basis state $|x\rangle|0\rangle$ for which x satisfies k clauses, we will apply k times the gate $R_{\frac{\pi}{2d}}$ on the ancilla qubit. Hence the state of the ancilla qubit becomes $\begin{bmatrix}\cos(\frac{\pi}{2d}) & -\sin(\frac{\pi}{2d})\\ \sin(\frac{\pi}{2d}) & \cos(\frac{\pi}{2d})\end{bmatrix}^k|0\rangle = \begin{bmatrix}\cos(\frac{k\pi}{2d}) & -\sin(\frac{k\pi}{2d})\\ \sin(\frac{k\pi}{2d}) & \cos(\frac{k\pi}{2d})\end{bmatrix}|0\rangle = \cos\left(\frac{k\pi}{2d}\right)|0\rangle + \sin\left(\frac{k\pi}{2d}\right)|1\rangle$. So by applying Hadamard gates on the first n qubits and then these controlled rotations on the last qubit, we obtain the state

$$c\sum_{x\in\{0,1\}^n}|x\rangle\left(\cos\left(\frac{k_x\pi}{2d}\right)|0\rangle + \sin\left(\frac{k_x\pi}{2d}\right)|1\rangle\right), \tag{10}$$

where k_x is the number of clauses satisfied by x. Hence, if we measure the last qubit and observe $|1\rangle$, the state collapses to

$$c\sum_{x\in\{0,1\}^n}\sin\left(\frac{k_x\pi}{2d}\right)|x\rangle, \tag{11}$$

where c is a normalizing constant. Since k_x is a value between 0 and d, the amplitude $\sin\left(\frac{k_x\pi}{2d}\right)$ of $|x\rangle$ gets larger as the number k_x gets larger.

IV. Experiments

A. Noise tolerance

We used our method to prepare the state (11) from the MaxSat example for the boolean formula

$$\begin{aligned}\rho(x_1,\ldots,x_8) :=& (\neg x_7\vee\neg x_8)\wedge(x_4\vee\neg x_3)\wedge(\neg x_3\vee\neg x_6)\\ &\wedge(\neg x_6\vee\neg x_5)\wedge(x_7\vee\neg x_1)\wedge(\neg x_3\vee\neg x_6)\\ &\wedge(\neg x_4\vee\neg x_6)\wedge(\neg x_2\vee\neg x_6).\end{aligned} \tag{12}$$

As we know from the previous discussion, measuring this state we should observe each basis state $|x\rangle$ a proportion of approximately $p_x := c\cdot\sin\left(\frac{k_x\pi}{2d}\right)^2$ of the time, where k_x is the number of clauses satisfied by x and c is a normalizing constant. To verify this, we first prepared this state with our method on a "perfect quantum computer" (Qiskit's "Qasm_simulator" of a noiseless quantum computer [26]). We prepared the state (10) overall 50.000 times, and measuring the last qubit we observed 39.573 times $|1\rangle$, resulting 39.573 times in the preparation of state (11). In Figure 2 we show for each variable assignment x on the Y-axis the proportion of times $\hat{p}_x$ that x was observed and on the X-axis the number of clauses k_x satisfied by x. We can clearly see that the values $\hat{p}_x$ are centered at p_x.

To compare the noise tolerance of our method to the noise tolerance of the state preparation circuits synthesized by the

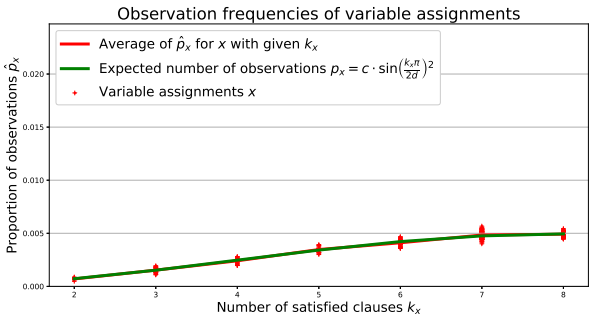

Fig. 2: Our method on a QC simulator without noise

Qiskit initializer [27], which is an optimized implementation of the method of Shende et al. [13], we repeated this experiment on a simulator of a noisy quantum computer (the "FakeJohannesburg()" backend [28]). Now we prepared the state (10) overall 500.000 times, and measuring the last qubit we obtained now 253.834 times the state (11) with our method. We then prepared this state 253.834 times with the Qiskit initializer. The results are shown in Figure 3. We can see that the values $\hat{p}_x$ are now in general less close to the expected values p_x, but with our method they are on expectation closer together, indiciating that our method is more noise-tolerant. This is confirmed by Figure 4, which show the difference $|p_x - \hat{p}_x|$ for all 256 variable assignments x using our method and the Qiskit initializer. The circuit produced by our method had 400 gates, whereas the circuit from the Qiskit initializer had 1231 gates (in both cases, this refers to the gate counts after transpilation to the backend).

B. Growth of circuits

To investigate how our circuits grow as the number of variables grows, we generated random boolean formulas with n ranging from 3 to 11 and $d = n$. For each n we sampled 5 boolean formulas for which we then synthesized circuits for state preparation with our method and the Qiskit initializer. In Figure 4 we show the average size (after transpilation to the backend) of the resulting circuits (the figure contains bars representing the range from smallest to largest observed size, but these bars are not visible as the variance is very small). We can clearly see the linear growth of our circuits and the exponential growth of the circuits from the Qiskit initializer.

C. Uniform states

We used our method to prepare several uniform states. We prepared the states $|par\rangle$ from Example 1 for n ranging from 5 to 10. We also used our method to prepare $|f_5\rangle$[1] from Example 2 for $n = 4$. We also prepared the states $|\rho_i\rangle$ for i ranging from 1 to 4, where ρ_i is the boolean function given by the conjunction of the first i clauses of the boolean formula (12).

We also implemented Grover's method [19]. For the multicontrolled operations in Grover's method, we used Gidney's recursive construction [30] (instead of the **mcx** gates of Qiskit since those grow exponentially in the number of controlls) and for U_f we used the same circuits as for our method.

In Table 5 we show the size of the circuits produced by our method, the Qiskit initializer, and Grover's method (after transpilation to the Qiskit gateset {'u', 'cx'}). We also show the expected number of times that we need to apply R_f until collapsing to the desired state $|f\rangle$.

[1] to compute f_5 we used the circuit from [29]

V. Conclusions

For states that can be "described" by some efficiently computable function, measurements allow state preparation with small quantum circuits. When a boolean function f can be computed in polynomial time and it has at most polynomially times more zeros than ones, then $|f\rangle$ can be prepared in polynomial time with our MBSP approach. By considering general $\mathbb{C}$-valued (instead of boolean) functions, we can use measurements to prepare general states. In general, regardless of the expected number of times we need to apply the circuit, as long as f can be computed with circuits of size $o(2^n)$ we eventually obtain $|f\rangle$ from a circuit that is smaller than those produced by general non-MBSP methods and this makes our approach more noise-tolerant, which we verified experimentally on a simulator of a quantum computer. This makes our MBSP method an excellent choice to scale state preparation on noisy quantum hardware.

Appendix

The following is a standard result from probability theory.

Lemma 1. *Given a coin that shows 'Heads' with probability $\theta \in (0, 1]$, the expected number of times that we need to toss the coin until we see 'Heads' is given by $\frac{1}{\theta}$*

Proof. Letting $T(\theta)$ denote the expected number of times for a given θ, we have $T(\theta) = \theta + (1 - \theta)(T(\theta) + 1)$. Hence, it follows that $T(\theta) = \frac{1}{\theta}$. □

References

[1] P. Shor, "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer", SIAM J. Comput, Society for Industrial and Applied Mathematics, pp. 1484–1509, 1997

[2] A.W. Harrow, A. Hassidim, S. Lloyd, "Quantum Algorithm for Linear Systems of Equations", Physical Review Letters, American Physical Society (APS), 2009

[3] J. Preskill, "Quantum Computing in the NISQ era and beyond", Quantum, Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften, 2018

[4] E. Knill, "Approximation by quantum circuits", arXiv, 1995

[5] M. Möttönen and J. Vartiainen and V. Bergholm and M. M. Salomaa, "Transformation of quantum states using uniformly controlled rotations", Quantum Inf. Comput., pp. 467-473, 2005

[6] D. Coppersmith, "An approximate Fourier transform useful in quantum factoring", Technical Report RC19642, IBM, 1994

[7] R. Wille, M. Soeken, R. Drechsler, "Reducing the Number of Lines in Reversible Circuits", 47th Design Automation Conference, ACM, pp. 647–652, 2010

[8] R. Wille, O. Keszocze, R. Drechsler, "Determining the minimal number of lines for large reversible circuits", Design, Automation and Test in Europe, pp. 1–4, 2011

[9] M. Soeken, R. Wille, O. Keszocze, D. M. Miller, R. Drechsler, "Embedding of Large Boolean Functions for Reversible Logic", J. Emerg. Technol. Comput. Syst., ACM, pp. 41:1–41:26, 2015

[10] D. Maslov, G. W. Dueck, "Reversible cascades with minimal garbage", IEEE Trans. on CAD of Integrated Circuits and Systems, pp. 1497–1509, 2004

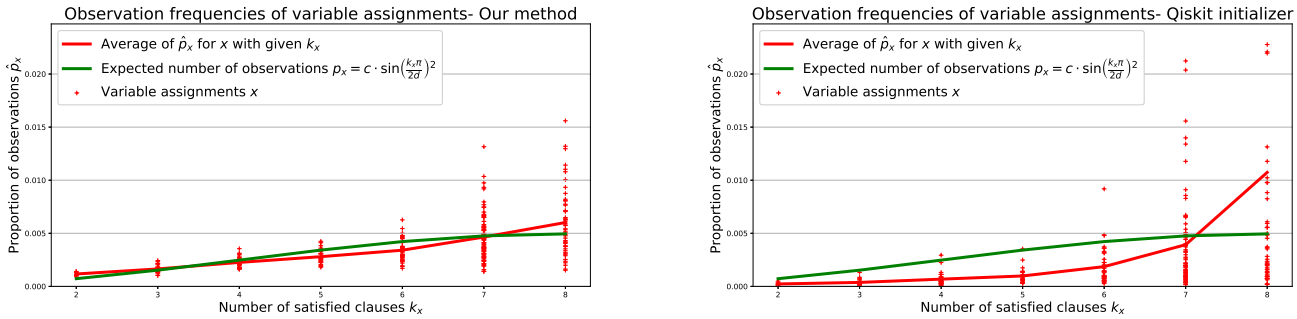

Fig. 3: Our method on a noisy simulator (left) and Qiskit initializer on a noisy simulator [27] [13] (right).

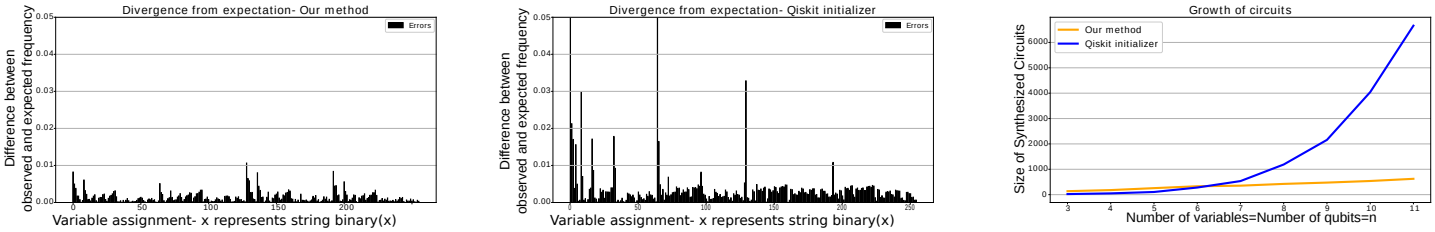

Fig. 4: $|p_x - \hat{p}_x|$ for all 256 variable assignments x with our method (left) and Qiskit initializer [27] [13] (middle). Growth of circuits for preparation of "MaxSat-States" with our method and Qiskit initializer (right).

State	Size	Expected repetitions	Size Qiskit	Size Grover
$\|f_5\rangle, n = 4$	24	4.000	29	208
$\|par\rangle, n = 5$	11	2.000	61	1028
$\|par\rangle, n = 6$	13	2.000	125	924
$\|par\rangle, n = 7$	15	2.000	253	1368
$\|par\rangle, n = 8$	17	2.000	509	2250
$\|par\rangle, n = 9$	19	2.000	1021	2692
$\|par\rangle, n = 10$	21	2.000	2045	3134
$\|\rho_1\rangle, n = 8$	37	1.333	509	1188
$\|\rho_2\rangle, n = 8$	80	1.777	509	2544
$\|\rho_3\rangle, n = 8$	91	2.133	509	2602
$\|\rho_4\rangle, n = 8$	231	2.666	509	2876

Fig. 5: Sizes of circuits produced for special states.

[11] A. Zulehner, R. Wille, "Make it reversible: Efficient embedding of non-reversible functions", Design, Automation & Test in Europe Conference & Exhibition, pp. 458–463, 2017

[12] N. Gleinig, F. A. Hubis, T. Hoefler, "Embedding Functions Into Reversible Circuits: A Probabilistic Approach to the Number of Lines", 56th Design Automation Conference, pp. 1-6, 2019

[13] V.V. Shende, S.S. Bullock, I.L. Markov, "Synthesis of quantum-logic circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, IEEE, pp. 1000–1010, 2006

[14] P. Kaye, M. Mosca, "Quantum Networks for Generating Arbitrary Quantum States", Optical Fiber Communication Conference and International Conference on Quantum Information, Optical Society of America, 2001

[15] P. Niemann, R. Datta, R. Wille, "Logic Synthesis for Quantum State Generation", IEEE 46th International Symposium on Multiple-Valued Logic (ISMVL), pp. 247–252 2016

[16] F. Mozafari, M. Soeken, G. De Micheli, "Preparation of Uniform Quantum States Utilizing Boolean Functions", 28th International Workshop on Logic Synthesis (IWLS), 2019

[17] I.F. Araujo, D.K. Park, F. Petruccione, A.J. da Silva, "A divide-and-conquer algorithm for quantum state preparation", Scientific Reports, 2021

[18] F. Mozafari, M. Soeken, H. Riener, G. De Micheli, "Automatic Uniform Quantum State Preparation Using Decision Diagrams", IEEE 50th International Symposium on Multiple-Valued Logic (ISMVL), pp. 170–175, 2020

[19] L.K. Grover, "Rapid sampling through quantum computing", Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC '00), pp. 618–626, 2000

[20] R. D. Somma, S. Boixo, H. Barnum, E. Knill, "Quantum Simulations of Classical Annealing Processes", Phys. Rev. Lett., American Physical Society, 2008

[21] D. Aharonov, A. Ta-Shma, "Adiabatic Quantum State Generation and Statistical Zero Knowledge", Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing, pp. 20–29, 2003

[22] S. Aaronson, "Multilinear Formulas and Skepticism of Quantum Computing", Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, pp. 118–127, 2004

[23] N. Gleinig, T. Hoefler, "An Efficient Algorithm for Sparse Quantum State Preparation", 58th Design Automation Conference, 2021

[24] S. Aaronson, "The Complexity of Quantum States and Transformations: From Quantum Money to Black Holes", arXiv, 2016

[25] H. Thapliyal, T. S. S. Varun, E. Munoz-Coreas, K. A. Britt, T. S. Humble, "Quantum Circuit Designs of Integer Division Optimizing T-count and T-depth", IEEE International Symposium on Nanoelectronic and Information Systems, 2017

[26] "qiskit.providers.aer.QasmSimulator", https://qiskit.org/documentation/stubs/qiskit.providers.aer.QasmSimulator.html, 2021 (accessed May 19, 2021)

[27] "Source code for qiskit.extensions.quantum_initializer.initializer", https://qiskit.org/documentation/_modules/qiskit/extensions/quantum_initializer/initializer.html#Initialize, 2021 (accessed May 19, 2021)

[28] "Device backend noise simulations", https://qiskit.org/documentation/tutorials/simulators/2_device_noise_simulation.html, 2021 (accessed May 19, 2021)

[29] D. Maslov, G.W. Dueck, and N. Scott, "Reversible Logic Synthesis Benchmarks Page", http://webhome.cs.uvic.ca/ dmaslov., 2005 (ac-cessed May 19, 2020)

[30] C. Gidney, "Constructing large controlled nots", https://algassert.com/circuits/2015/06/05/Constructing-Large-Controlled-Nots.html, 2015 (accessed May 5, 2021)