GAPs for Shallow Implementation of Quantum Finite Automata

Mansur Ziiatdinov
Kazan Federal University, Kazan 420008, Russia
gltronred@gmail.com

Aliya Khadieva
Faculty of Computing, University of Latvia, Rīga, Latvia
Kazan Federal University, Kazan 420008, Russia
aliya.khadi@gmail.com

Abuzer Yakaryılmaz
Faculty of Computing, University of Latvia, Rīga, Latvia
abuzer.yakaryilmaz@lu.lv

Quantum fingerprinting is a technique that maps classical input word to a quantum state. The obtained quantum state is much shorter than the original word, and its processing uses less resources, making it useful in quantum algorithms, communication, and cryptography. One of the examples of quantum fingerprinting is quantum automata algorithms for $MOD_p = \{a^{i \cdot p} \mid i \geq 0\}$ languages, where p is a prime number.

However, implementing such an automaton on the current quantum hardware is not efficient. Quantum fingerprinting maps a word $x \in \{0,1\}^n$ of length n to a state $|\psi(x)\rangle$ of $O(\log n)$ qubits, and uses $O(n)$ unitary operations. Computing quantum fingerprint using all available qubits of the current quantum computers is infeasible due to a large number of quantum operations.

To make quantum fingerprinting practical, we should optimize the circuit for depth instead of width in contrast to the previous works. We propose explicit methods of quantum fingerprinting based on tools from additive combinatorics, such as generalized arithmetic progressions (GAPs), and prove that these methods provide circuit depth comparable to a probabilistic method. We also compare our method to prior work on explicit quantum fingerprinting methods.

1 Introduction

A quantum finite state automaton (QFA) is a generalization of classical finite automaton [16, 4]. Here we use the known simplest QFA model [12]. Formally, a QFA is 5-tuple $M = (Q, A \cup \{¢, \$\}, |\psi_0\rangle, \mathscr{U}, \mathscr{H}_{acc})$, where $Q = \{q_1, \ldots, q_D\}$ is a finite set of states, A is the finite input alphabet, $¢, \$$ are the left and right end-markers, respectively. The state of M is represented as a vector $|\psi\rangle \in \mathscr{H}$, where $\mathscr{H}$ is the D-dimensional Hilbert space spanned by $\{|q_1\rangle, \ldots, |q_D\rangle\}$ (here $|q_j\rangle$ is a zero column vector except its j-th entry that is 1). The automaton M starts in the initial state $|\psi_0\rangle \in \mathscr{H}$, and makes transitions according to the operators $\mathscr{U} = \{U_a \mid a \in A\}$ of unitary matrices. After reading the whole input word, the final state is observed with respect to the accepting subspace $\mathscr{H}_{acc} \subseteq \mathscr{H}$.

Quantum fingerprinting provides a method of constructing automata for certain problems. It maps an input word $w \in \{0,1\}^n$ to much shorter quantum state, its fingerprint $|\psi(w)\rangle = U_w |0^m\rangle$, where U_w is the single transition matrix representing the multiplication of all transition matrices while reading w and $|0^m\rangle = \underbrace{|0\rangle \otimes \cdots \otimes |0\rangle}_{m \text{ times}}$. Quantum fingerprint captures essential properties of the input word that can be useful for computation.

Zs. Gazdag, Sz. Iván, G. Kovásznai (Eds.): 16th International Conference on Automata and Formal Languages (AFL 2023)
EPTCS 386, 2023, pp. 269–280, doi:10.4204/EPTCS.386.21

One example of quantum fingerprinting applications is the QFA algorithms for MOD_p language [3]. For a given prime number p, the language MOD_p is defined as $MOD_p = \{a^i \mid i \text{ is divisible by } p\}$. Let us briefly describe the construction of the QFA algorithms for MOD_p.

We start with a 2-state QFA M_k, where $k \in \{1, \ldots, p-1\}$. The automaton M_k has two base states $Q = \{q_0, q_1\}$, it starts in the state $|\psi_0\rangle = |q_0\rangle$, and it has the accepting subspace spanned by $|q_0\rangle$. At each step (for each letter) we perform the rotation

$$U_a = \begin{pmatrix} \cos\frac{2\pi k}{p} & \sin\frac{2\pi k}{p} \\ -\sin\frac{2\pi k}{p} & \cos\frac{2\pi k}{p} \end{pmatrix}.$$

It is easy to see that this automaton gives the correct answer with probability 1 if $w \in MOD_p$. However, if $w \notin MOD_p$, the probability of correct answer can be close to 0 rather than 1 (i.e., bounded below by $1 - \cos^2(\pi/p)$). To boost the success probability we use d copies of this automaton, namely M_{k_1}, …, M_{k_d}, as described below.

The QFA M for MOD_p has $2d$ states: $Q = \{q_{1,0}, q_{1,1}, \ldots, q_{d,0}, d_{d,1}\}$, and it starts in the state $|\psi_0\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |q_{i,0}\rangle$. In each step, it applies the transformation defined as:

$$|q_{i,0}\rangle \mapsto \cos\frac{2\pi k_i}{p} |q_{i,0}\rangle + \sin\frac{2\pi k_i}{p} |q_{i,1}\rangle \tag{1}$$

$$|q_{i,1}\rangle \mapsto -\sin\frac{2\pi k_i}{p} |q_{i,0}\rangle + \cos\frac{2\pi k_i}{p} |q_{i,1}\rangle \tag{2}$$

Indeed, M enters into equal superposition of d sub-QFAs, and each sub-QFA applies its rotation. Thus, quantum fingerprinting technique associates the input word $w = a^j$ with its fingerprint

$$|\psi\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} \cos\frac{2\pi k_i j}{p} |q_{i,0}\rangle + \sin\frac{2\pi k_i j}{p} |q_{i,1}\rangle .$$

Ambainis and Nahimovs [3] proved that this QFA accepts the language MOD_p with error probability that depends on the choice of the coefficients k_i's. They also showed that for $d = 2\log(2p)/\varepsilon$ there is at least one choice of coefficients k_i's such that error probability is less than ε. The proof uses a probabilistic method, so these coefficients are not explicit. They also suggest two explicit sequences of coefficients: cyclic sequence $k_i = g^i \pmod p$ for primitive root g modulo p and more complex AIKPS sequences based on the results of Ajtai et al. [2].

While quantum fingerprinting is versatile and has different applications [6, 1], it is not practical for the currently available real quantum computers. The main obstacle is that quantum fingerprinting uses an exponential (in the number m of qubits) circuit depth (e.g., see [11, 5, 15] for some implementations of the aforementioned automaton M). Therefore, the required quantum volume[1] V_Q is roughly $2^{|w|\cdot 2^m}$. For example, IBM reports [8] that its Falcon r5 quantum computer has 27 qubits with a quantum volume of 128. It means that we can use only 7 of 27 qubits for the fingerprint technique.

In this paper, we investigate how to obtain better circuit depth by optimizing the coefficients used by M: $k_1, \ldots, k_d$. We use generalized arithmetic progressions for generating a set of coefficients and show that such sets have a circuit depth comparable to the set obtained by the probabilistic method.

[1] Quantum volume is an exponent of the maximal square circuit size that can be implemented on the quantum computer [7, 18].

Table 1: Comparison of different methods.

Method	Width	Depth	Source	Note
Cyclic	$p^{c/\log\log p}$	$p^{c/\log\log p}$	[3]	for some constant $c > 0$
AIKPS	$\log^{2+3\varepsilon} p$	$(1+2\varepsilon)\log^{1+\varepsilon} p \log\log p$	[14]	
Probabilistic	$4\log(2p)/\varepsilon$	$2\log(2p)/\varepsilon$	[3]	
GAPs	p/ε^2	$\lceil \log p - 2\log\varepsilon \rceil + 2$	this paper	

We summarize the previous and our results in Table 1. Note that p is exponential in the number of qubits m. The depth of the circuits is discussed in Section 3.

The rest of the paper is organized as follows. In Section 2 we give the necessary definitions and results on quantum computation and additive combinatorics to follow the rest of the paper. Section 3 contains the construction of the shallow fingerprinting function and the proof of its correctness. Then, we present certain numerical simulations in Section 4. We conclude the paper with Section 5 by presenting some open questions and discussions for further research.

2 Preliminaries

Let us denote by $\mathscr{H}^2$ two-dimensional Hilbert space, and by $(\mathscr{H}^2)^{\otimes m}$ 2^m-dimensional Hilbert space (i.e., the space of m qubits). We use bra- and ket-notations for vectors in Hilbert space. For any natural number N, we use $\mathbb{Z}_N$ to denote the cyclic group of order N.

Let us describe in detail how the automaton M works. As we outlined in the introduction, the automaton M has $2d$ states: $Q = \{q_{1,0}, q_{1,1}, \ldots, q_{d,0}, d_{d,1}\}$, and it starts in the state $|\psi_0\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |q_{i,0}\rangle$. After reading a symbol a, it applies the transformation U_a defined by (1), (2):

$$|q_{i,0}\rangle \mapsto \cos\frac{2\pi k_i}{p} |q_{i,0}\rangle + \sin\frac{2\pi k_i}{p} |q_{i,1}\rangle$$
$$|q_{i,1}\rangle \mapsto -\sin\frac{2\pi k_i}{p} |q_{i,0}\rangle + \cos\frac{2\pi k_i}{p} |q_{i,1}\rangle$$

After reading the right endmarker \$, it applies the transformation $U_\$$ defined in such way that $U_\$ |\psi_0\rangle = |q_{1,0}\rangle$. The automaton measures the final state and accepts the word if the result is $q_{1,0}$.

So, the quantum state after reading the input word $w = a^j$ is

$$|\psi\rangle = \frac{1}{\sqrt{d}} \sum_{i=1}^{d} \cos\frac{2\pi k_i j}{p} |q_{i,0}\rangle + \sin\frac{2\pi k_i j}{p} |q_{i,1}\rangle .$$

If $j \equiv 0 \pmod{p}$, then $|\psi\rangle = |\psi_0\rangle$, and $U_\$$ transforms it into accepting state $|q_{1,0}\rangle$, therefore, in this case, the automaton always accepts. If the input word $w \notin MOD_p$, then the quantum state after reading the right endmarker \$ is

$$|\psi'\rangle = \frac{1}{d}\Big(\sum_{i=1}^{d} \cos\frac{2\pi k_i j}{p}\Big) |q_{1,0}\rangle + \ldots,$$

and the error probability is

$$P_e = \frac{1}{d^2}\Big(\sum_{i=1}^{d} \cos\frac{2\pi k_i x}{p}\Big)^2 .$$

In the rest of the paper, we denote by m the number of qubits in the quantum fingerprint, by $d = 2^m$ the number of parameters in the set K, by p the size of domain of the quantum fingerprinting function, and by $U_a(K)$ the transformation defined above, which depends on the set K.

Let us also define a function $\varepsilon : \mathbb{Z}_p^d \to \mathbb{R}$ as follows:

$$\varepsilon(K) = \max_{x \in \mathbb{Z}_p} \left(\frac{1}{d^2} \Big| \sum_{j=1}^{d} \exp \frac{2\pi i k_j x}{p} \Big|^2 \right).$$

Note that $P_e \leq \varepsilon(K)$.

We also use some tools from additive combinatorics. We refer the reader to the textbook by Tao and Vu [17] for a deeper introduction to additive combinatorics.

An additive set $A \subseteq Z$ is a finite non-empty subset of Z, an abelian group with group operation $+$. We refer Z as the ambient group.

If A, B are additive sets in Z, we define the sum set $A+B = \{a+b \mid a \in A, b \in B\}$. We define additive energy $E(A,B)$ between A, B to be

$$E(A,B) = \Big| \{(a,b,a',b') \in A \times B \times A \times B \mid a+b = a'+b'\} \Big|.$$

Let us denote by $e(\theta) = e^{2\pi i \theta}$, and by $\xi \cdot x = \xi x/p$ bilinear form from $\mathbb{Z}_p \times \mathbb{Z}_p$ into $\mathbb{R}/\mathbb{Z}$. Fourier transform of $f : \mathbb{Z}_p \to \mathbb{Z}_p$ is $\hat{f}(\xi) = \mathbf{E}_{x \in Z} f(x) \overline{e(\xi \cdot x)}$.

We also denote the characteristic function of the set A as 1_A, and we define $\mathbf{P}_Z(A) = \widehat{1_A}(0) = |A|/|Z|$.

Definition 1 ([17]). *Let Z be a finite additive group. If $A \subseteq Z$, we define Fourier bias $\|A\|_{\mathscr{U}}$ of the set A to be*

$$\|A\|_{\mathscr{U}} = \sup_{\xi \in Z \setminus \{0\}} |\widehat{1_A}(\xi)|$$

There is a connection between the Fourier bias and the additive energy.

Theorem 1 ([17]). *Let A be an additive set in a finite additive group Z. Then*

$$\|A\|_{\mathscr{U}}^4 \leq \frac{1}{|Z|^3} E(A,A) - \mathbf{P}_Z(A)^4 \leq \|A\|_{\mathscr{U}}^2 \mathbf{P}_Z(A)$$

Definition 2 ([17]). *Generalized arithmetic progression (GAP) of dimension d is a set*

$$A = \{x_0 + n_1 x_1 + \ldots + n_d x_d \mid 0 \leq n_1 \leq N_1, \cdots, 0 \leq n_d \leq N_d\},$$

where $x_0, x_1, \ldots, x_d, N_1, \ldots, N_d \in Z$. The size of GAP is a product $N_1 \cdots N_d$. If the size of set A, $|A|$, equals to $N_1 \cdots N_d$, we say that GAP is proper.

3 Shallow Fingerprinting

Quantum fingerprint can be computed by the quantum circuit given in Figure 1. The last qubit is rotated by a different angle $2\pi k_j x/q$ in different subspaces enumerated by $|j\rangle$. Therefore, the circuit depth is $|K| = t = 2^m$. As the set K is random, it is unlikely that the depth can be less than $|K|$.

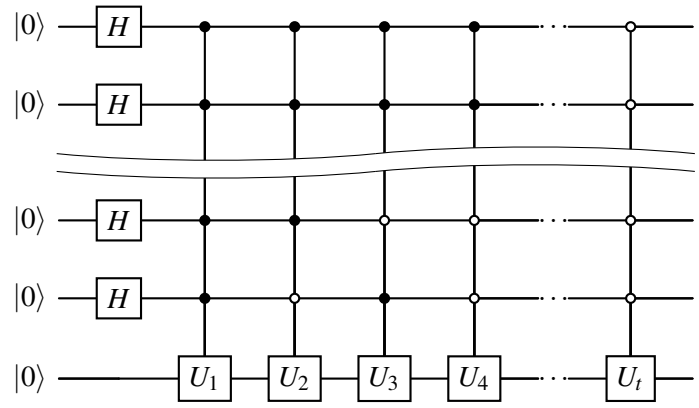


Figure 1: Deep fingerprinting circuit example. Gate U_j is a rotation $R_y(4\pi k_j x/p)$. Controls in controlled gates run over all binary strings of length s

Let us note that fingerprinting is similar to quantum Fourier transform. Quantum Fourier transform computes the following transformation:

$$|x\rangle \mapsto \frac{1}{N}\sum_{k=0}^{N-1} \omega_N^{xk} |k\rangle, \tag{3}$$

where $\omega_N = e(1/N)$. Here is the quantum fingerprinting transform:

$$|x\rangle \mapsto \frac{1}{t}\sum_{j=1}^{t} \omega_N^{k_j x} |k\rangle.$$

The depth of the circuit that computes quantum Fourier transform is $O((\log N)^2)$, and it heavily relies on the fact that in Eq. (3) the sum runs over all $k = 0,\ldots,N-1$. Therefore, to construct a shallow fingerprinting circuit we desire to find a set K with special structure.

Suppose that we construct a coefficient set $K \subset \mathbb{Z}_p$ in the following way. We start with a set $T = \{t_1,\ldots,t_m\}$ and construct the set of coefficients as a set of sums of all possible subsets:

$$K = \Big\{ \sum_{t\in S} t \mid S \subseteq T \Big\},$$

where we sum modulo p.

The quantum fingerprinting function with these coefficients can be computed by a circuit of depth $O(m)$ [9] (see Figure 2).

Finally, let us prove why the construction of the set $K \subset \mathbb{Z}_p$ works.

Theorem 2. *Let $\varepsilon > 0$, let $m = \lceil \log p - 2\log\varepsilon \rceil$ and $d = 2^m$.*

Suppose that the number $t_0 \in \mathbb{Z}_p$ and the set $T = \{t_1,\ldots,t_m\} \subset \mathbb{Z}_p$ are such that

$$B = \{2t_0 + n_1 t_1 + \cdots + n_m t_m \mid 0 \le n_1 < 3, \ldots, 0 \le n_m < 3\}$$

is a proper GAP.

Then the set A defined as

$$A = \left\{ t_0 + \sum_{t\in S} t \mid S \subseteq T \right\}$$

has $\varepsilon(A) \le \varepsilon$.

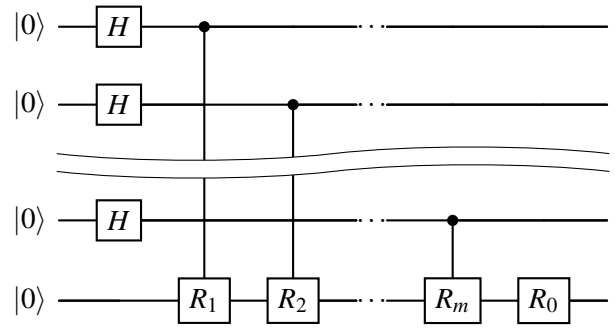


Figure 2: Shallow fingerprinting circuit example. Gate R_j is a rotation $R_y(4\pi t_j x/p)$

Let us outline the proof of this theorem. Firstly, we estimate the number of solutions to $a+b=n$. Secondly, we use it to bound the additive energy $E(A,A)$ of the set A. Thirdly, we bound the Fourier bias $\|A\|_{\mathscr{U}}$. Finally, we get a bound on $\varepsilon(A)$ in terms of p and m.

Proof. Let us denote a set $R_n(A)$ of solutions to $a+b=n$, where $a,b\in A$ and $n\in\mathbb{Z}_p$:

$$R_n(A)=\{(a,b)\mid a+b=n;\ a,b\in A\}.$$

Note that we have $E(A,A)=\sum_{n\in Z}R_n(A)^2$.

Suppose that n is represented as $n=2t_0+\sum_{i=1}^{m}\gamma_i t_i$, $\gamma_i\in\{0,1,2\}$. If such representation exists, it is unique, because B is a proper GAP. Let us denote $c_0:=\{i\mid\gamma_i=0\}$, $c_1:=\{i\mid\gamma_i=1\}$, $c_2:=\{i\mid\gamma_i=2\}$. It is clear that $c_0\uplus c_1\uplus c_2=[m]$.

Now suppose that $n=a+b$ for some $a,b\in A$. But $a=t_0+\sum_i\alpha_i t_i$ and $b=t_0+\sum_i\beta_i t_i$, $\alpha_i,\beta_i\in\{0,1\}$. We get that if $i\in c_0$ or $i\in c_2$ then the corresponding coefficients α_i and β_i are uniquely determined. Consider $i\in c_1$. Then we have two choices: either $\alpha_i=1;\beta_i=0$, or $\alpha_i=0;\beta_i=1$. Therefore, we have $R_n(A)=2^{|c_1(n,A)|}$.

We have that

$$E(A,A)=\sum_{n\in Z}R_n(A)^2=\sum_{n\in Z}2^{2|c_1(n,A)|}.$$

Using the fact that $|c_0(n,A)|+|c_1(n,A)|+|c_2(n,A)|=m$, we see that

$$E(A,A)=\sum_{n\in Z}2^{2|c_1(n,A)|}=\sum_{j=0}^{m}\binom{m}{j}2^{m-j}2^{2j}=\sum_{j=0}^{m}\binom{m}{j}2^{m+j}\le 2^{3m}$$

We can bound the Fourier bias by Theorem 1:

$$\|A\|_{\mathscr{U}}^4\le\frac{1}{|Z|^3}E(A,A)-\mathbf{P}_Z(A)^4\le\|A\|_{\mathscr{U}}^2\,\mathbf{P}_Z(A)$$

$$\|A\|_{\mathscr{U}}^4\le\frac{2^{3m}}{2^{3\cdot 2^m}}-\frac{2^{4m}}{2^{4\cdot 2^m}}=\frac{d^3}{2^{3d}}-\frac{d^4}{2^{4d}}$$

$$\|A\|_{\mathscr{U}}\le\frac{d^{3/4}}{p^{3/4}}$$

Finally, we have

$$\varepsilon(A)=\left(\frac{p}{d}\|A\|_{\mathscr{U}}\right)^2\le\frac{p^{1/2}}{d^{1/2}}.$$

By substituting the definitions of d and m, we prove the theorem. □

Corollary 1. *The depth of the circuit that computes $U_a(A)$ is $\lceil \log p - 2\log \varepsilon \rceil$.*

Theorem 3 (Circuit depth for AIKPS sequences)**.** *For given $\varepsilon > 0$, let*

$$R = \{r \mid r \text{ is prime}, (\log p)^{1+\varepsilon}/2 < r < (\log p)^{1+\varepsilon}\},$$
$$S = \{1, 2, \ldots, (\log p)^{1+2\varepsilon}\},$$
$$T = \{s \cdot r^{-1} \mid r \in R, s \in S\},$$

where r^{-1} is the inverse of r modulo p.

Then the depth of the circuit that computes $U_a(T)$ is less than $(1+2\varepsilon)\log^{1+\varepsilon} p \log\log p$.

Proof. Let us denote the elements of R by $r_1, r_2, \ldots$. Let $S \cdot \{r^{-1}\}$ be a set $\{s \cdot r^{-1} \mid s \in S\}$.

Consider the following circuit $\mathscr{C}_j$ (see Figure 3) with $w = \lceil (1+2\varepsilon)\log\log p \rceil + 1$ wires.

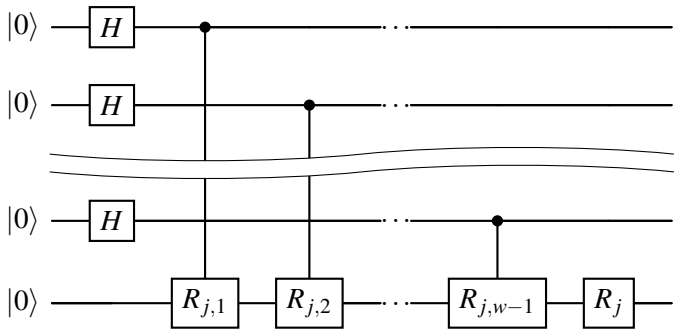


Figure 3: Circuit $\mathscr{C}_j$ for AIKPS subsequence. Gate R_j is a rotation $R_y(4\pi(r_j^{-1})/p)$. Gate $R_{j,k}$ is a rotation $R_y(2^{k-1} \cdot 4\pi(r_j^{-1})/p)$

The circuit $\mathscr{C}_j$ has depth $\lceil (1+2\varepsilon)\log\log p \rceil + 1$ and computes the transformation $U_a(S \cdot \{r_j^{-1}\})$. By repeating the same circuit for all $r_j \in R$ we get the required circuit for $U_a(T)$ (see Figure 4).

Since $|R| < (\log p)^{1+\varepsilon}$, we obtain that the depth of the circuit $U_a(T)$ is less than

$$(1+2\varepsilon)\log^{1+\varepsilon} p \log\log p. \qquad \square$$

4 Numerical Experiments

We conduct the following numerical experiments. We compute sets of coefficients K for the automaton for the language MOD_p with minimal computational error.

Finding an optimal set of coefficients is an optimization problem with many parameters, and the running time of a brute force algorithm is large, especially with an increasing number m of control qubits and large values of parameter p. Then, the original automaton has $2d$ states, where $d = 2^m$. We observe circuits for several m values and use a heuristic method for finding the optimal sets K with respect to an error minimization. For this purpose, the coordinate descend method [19] is used.

We find an optimal sets of coefficients for different values of p and m and compare computational errors of original and shallow fingerprinting algorithms for the automaton (see Figure 5). Namely, we set $m = 3, 4, 5$ and find sets using the coordinate descend method for each case. Even heuristic computing, for $s > 5$, takes exponentially more computational time and it is hard to implement on our devices.

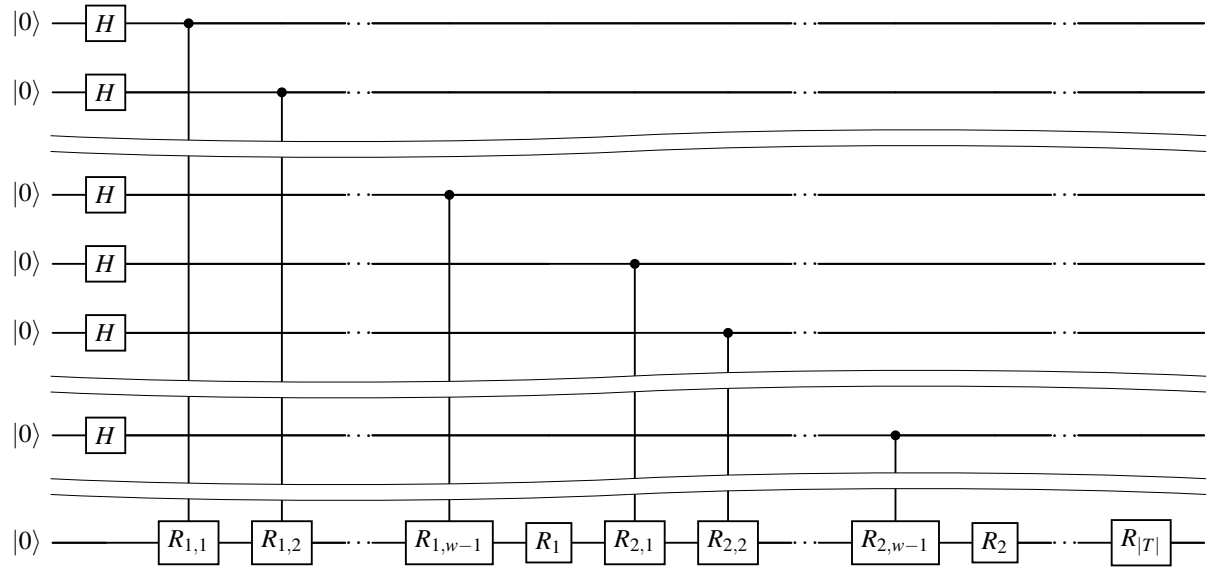


Figure 4: Circuit for $U_a(T)$. Gate R_j is a rotation $R_y(4\pi(r_j^{-1})/p)$. Gate $R_{j,k}$ is a rotation $R_y(2^{k-1} \cdot 4\pi(r_j^{-1})/p)$

One can note that difference between errors becomes bigger with increasing m, especially for big values p. The program code and numerical data are presented in a git repo [10].

The graphics in Figure 6 show a proportion of the errors of the original automaton over the errors of the shallow automaton for $m = 3,4,5$ and the prime numbers until 1013.

As we see, for a number of control qubits $m = 3$, the difference between the original and shallow automata errors is approximately constant. The ratio of values fluctuates between 1 and 1.2. In the case $m = 4$, this ratio is approximately 1.5 for almost all observed values p. The ratio of errors is nearly between 1.5 and 3, for $m = 5$.

According to the results of our experiments, the circuit depth $m+1$ is enough for valid computations, while the original circuit uses $O(2^m)$ gates. Since the shallow circuit is much simpler than the original one, its implementation on real quantum machines is much easier. For instance, in such machines as IBMQ Manila or Baidu quantum computer, a "quantum computer" is represented by a linearly related sequence of qubits. CX-gates can be applied only to the neighbor qubits. For such a linear structure of qubits, the shallow circuit can be implemented using $3m+3$ CX-gates. Whereas a nearest-neighbor decomposition [13] of the original circuit requires $O(d \log d) = O(m2^m)$ CX-gates.

5 Conclusions

We show that generalized arithmetic progressions generate some sets of coefficients k_i for the quantum fingerprinting technique with provable characteristics. These sets have large sizes, however, their depth is small and comparable to the depth of sets obtained by the probabilistic method. These sets can be used in the implementations of quantum finite automata suitable for running on the current quantum hardware.

We run numerical simulations. They show that the actual performance of the coefficients found by our method for quantum finite automata is not much worse than the performance of the other methods.

Optimizing quantum finite automata implementation for depth also poses an open question. The

Figure 5: Computational errors for $m = 3, 4, 5$ of original and shallow automata

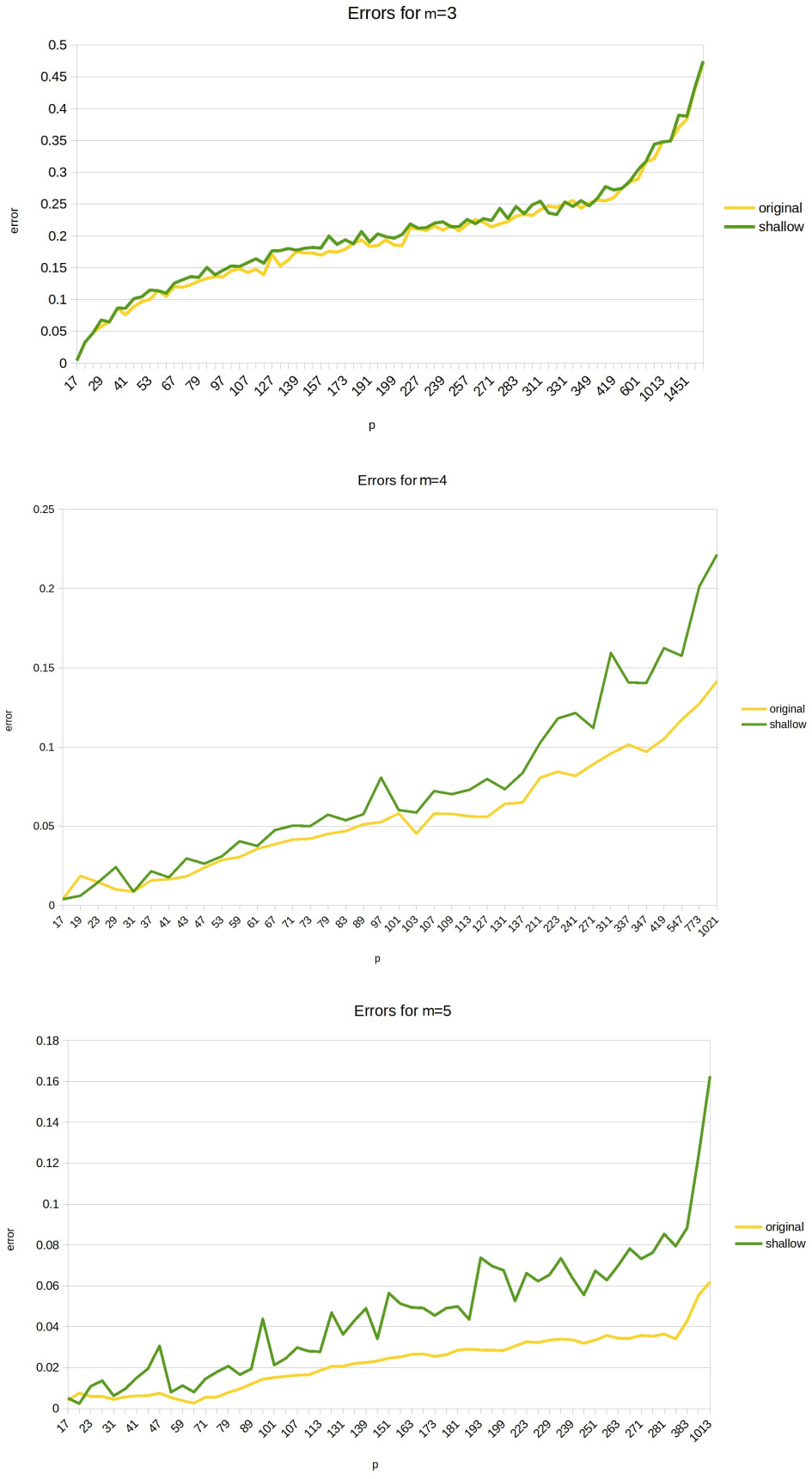
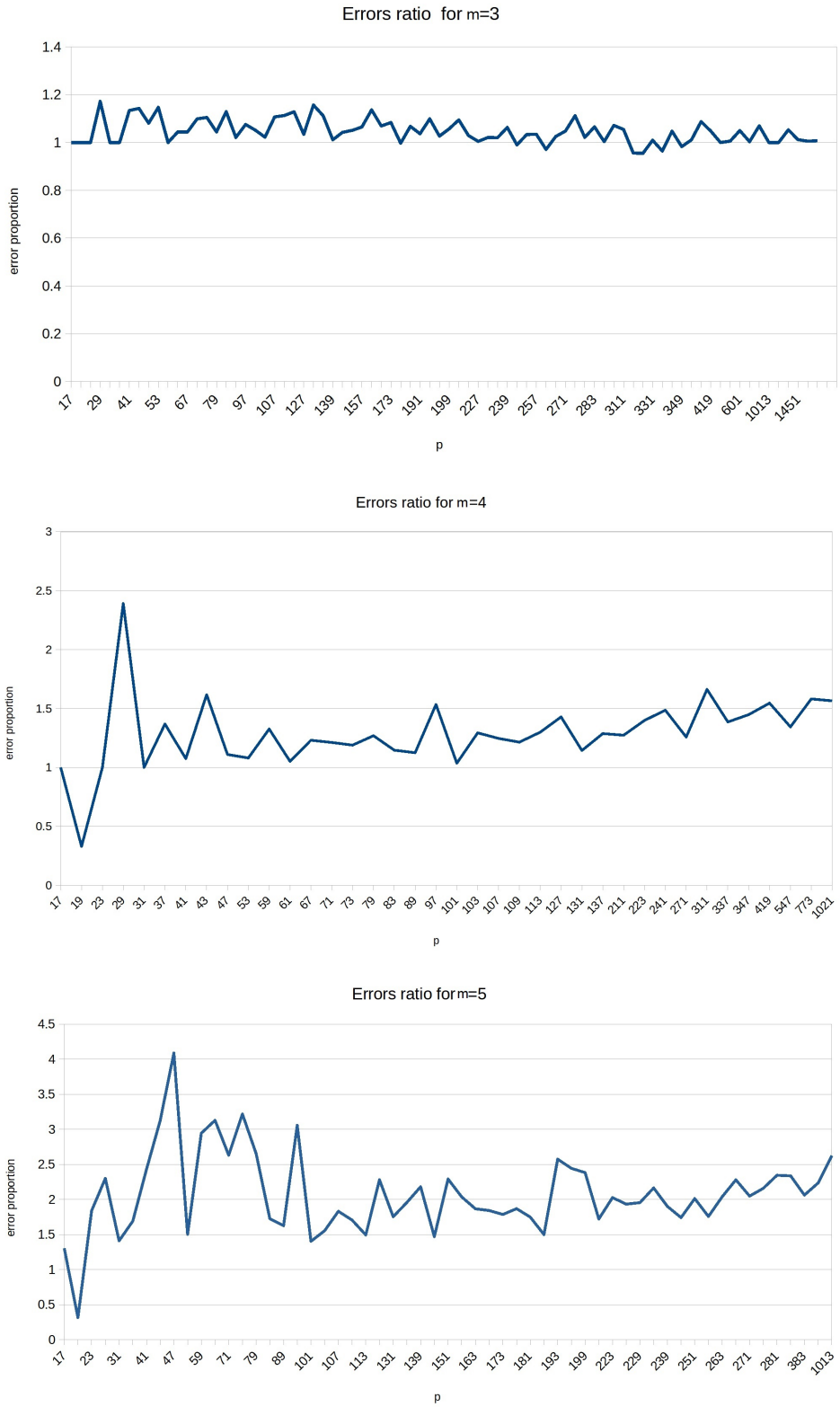


Figure 6: Proportions of the shallow automaton errors over the original automaton errors for $m = 3,4,5$ and different values of p



lower bound for the size of K in terms of p and ε is known [1]. Therefore, for given p and ε, quantum finite automata cannot have less than $O(\log p/\varepsilon)$ states. But, to our knowledge, a lower bound for the circuit depth of the transition function implementation is not known. So, we pose an open question: is it possible to implement a transition function with depth less than $O(\log p)$? What is the lower bound for it?

6 Acknowledgments

Yakaryılmaz was partially supported by the ERDF project Nr. 1.1.1.5/19/A/005 "Quantum computers with constant memory" and the project "Quantum algorithms: from complexity theory to experiment" funded under ERDF programme 1.1.1.5.

This paper has been supported by the Kazan Federal University Strategic Academic Leadership Program ("PRIORITY-2030"). Research in Section 4 were supported by the subsidy allocated to Kazan Federal University for the state assignment in the sphere of scientific activities, project No. 0671-2020-0065.

References

[1] Farid Ablayev, Marat Ablayev, Alexander Vasiliev & Mansur Ziatdinov (2016): *Quantum Fingerprinting and Quantum Hashing. Computational and Cryptographical Aspects*. Baltic Journal of Modern Computing 4(4), pp. 860–875, doi:10.22364/bjmc.2016.4.4.17.

[2] Miklós Ajtai, Henryk Iwaniec, János Komlós, János Pintz & Endre Szemerédi (1990): *Construction of a thin set with small Fourier coefficients*. Bulletin of the London Mathematical Society 22(6), pp. 583–590, doi:10.1112/blms/22.6.583.

[3] Andris Ambainis & Nikolajs Nahimovs (2009): *Improved constructions of quantum automata*. Theoretical Computer Science 410(20), pp. 1916–1922, doi:10.1016/j.tcs.2009.01.027.

[4] Andris Ambainis & Abuzer Yakaryılmaz (2021): *Automata and quantum computing*. In Jean Éric Pin, editor: Handbook of Automata Theory, chapter 39, 2, European Mathematical Society Publishing House, pp. 1457–1493, doi:10.4171/Automata-2/17.

[5] Utku Birkan, Özlem Salehi, Viktor Olejar, Cem Nurlu & Abuzer Yakaryılmaz (2021): *Implementing Quantum Finite Automata Algorithms on Noisy Devices*. In: International Conference on Computational Science, Springer, pp. 3–16, doi:10.1007/978-3-030-77980-1_1.

[6] Harry Buhrman, Richard Cleve, John Watrous & Ronald de Wolf (2001): *Quantum Fingerprinting*. Physical Review Letters 87(16), p. 167902, doi:10.1103/PhysRevLett.87.167902. arXiv:0102001.

[7] Andrew W. Cross, Lev S. Bishop, Sarah Sheldon, Paul D. Nation & Jay M. Gambetta (2019): *Validating quantum computers using randomized model circuits*. Physical Review A 100, p. 032328, doi:10.1103/PhysRevA.100.032328.

[8] IBM (2022): *Eagle's quantum performance progress*. Available at `https://research.ibm.com/blog/eagle-quantum-processor-performance`.

[9] Martin Kālis (2018): *Kvantu Algoritmu Realizācija Fiziskā Kvantu Datorā (Quantum Algorithm Implementation on a Physical Quantum Computer)*. Master's thesis, University of Latvia.

[10] Aliya Khadieva: *Optimal Parameters Computing Code*. Available at `https://github.com/aliyakhadi/Parameters_counting`.

[11] Aliya Khadieva & Mansur Ziatdinov (2023): *Deterministic Construction of QFAs Based on the Quantum Fingerprinting Technique*. Lobachevskii Journal of Mathematics 44(2), pp. 713–723, doi:10.1134/S199508022302021X.

[12] Cristopher Moore & James P Crutchfield (2000): *Quantum automata and quantum grammars*. *Theoretical Computer Science* 237(1-2), pp. 275–306, doi:10.1016/S0304-3975(98)00191-1.

[13] Mikka Möttönen & Juha J Vartiainen (2006): *Decompositions of general quantum gates*. *Trends in Quantum Computing Research*, doi:10.48550/ARXIV.QUANT-PH/0504100.

[14] Alexander Razborov, Endre Szemerédi & Avi Wigderson (1993): *Constructing small sets that are uniform in arithmetic progressions*. *Combinatorics, Probability and Computing* 2(4), pp. 513–518, doi:10.1017/S0963548300000870.

[15] Özlem Salehi & Abuzer Yakaryılmaz (2021): *Cost-efficient QFA Algorithm for Quantum Computers*, doi:10.48550/arXiv.2107.02262.

[16] A. C. Cem Say & Abuzer Yakaryılmaz (2014): *Quantum finite automata: A modern introduction*. In: *Computing with New Resources*, Springer, pp. 208–222, doi:10.1007/978-3-319-13350-8_16.

[17] Terence Tao & Van Vu (2006): *Additive combinatorics*. *Cambridge Studies in Advanced Mathematics* 105, Cambridge University Press, doi:10.1017/CBO9780511755149.

[18] Andrew Wack, Hanhee Paik, Ali Javadi-Abhari, Petar Jurcevic, Ismael Faro, Jay M. Gambetta & Blake R. Johnson (2021): *Quality, Speed, and Scale: three key attributes to measure the performance of near-term quantum computers*, doi:10.48550/ARXIV.2110.14108.

[19] Stephen J Wright (2015): *Coordinate descent algorithms*. *Mathematical programming* 151(1), pp. 3–34, doi:10.1007/s10107-015-0892-3.