

Space Complexity of Streaming Algorithms on Universal Quantum Computers

Yanglin Hu, Darya Melnyk, Yuyi Wang, and Roger Wattenhofer

ETH Zurich, Switzerland
{yahu,dmelnyk,yuwang,wattenhofer}@ethz.ch

Abstract. Universal quantum computers are the only general purpose quantum computers known that can be implemented as of today. These computers consist of a classical memory component which controls the quantum memory. In this paper, the space complexity of some data stream problems, such as PartialMOD and Equality, is investigated on universal quantum computers. The quantum algorithms for these problems are believed to outperform their classical counterparts. Universal quantum computers, however, need classical bits for controlling quantum gates in addition to qubits. Our analysis shows that the number of classical bits used in quantum algorithms is equal to or even larger than that of classical bits used in corresponding classical algorithms. These results suggest that there is no advantage of implementing certain data stream problems on universal quantum computers instead of classical computers when space complexity is considered.

Keywords: Streaming algorithm · Universal quantum computer · Space complexity · Solovay-Kitaev algorithm.

1 Introduction

In the past two decades, scientists have made significant progress in the field of quantum computation. Quantum computer protocols based on different physical principles have been constructed and manufactured. Despite this progress, large-scale quantum computers are still not available.

According to the no-programming Theorem [19], a quantum-controlled quantum computer is not better than a classically controlled quantum computer. Therefore, a modern quantum computer consists of a large classical memory controlling a small quantum memory. The limited quantum memory poses great challenges to physicists and computer scientists. In particular, one must decide how to use this limited quantum memory efficiently. One possible way is to build larger-scale quantum computers. Another way is to introduce algorithms that require a small quantum memory, but a large classical memory. In this work, we address the latter case for a special class of problems – the data stream problems.

Data stream problems process data streams where the input data comes at a high rate. The massive input data challenges communication, computation, and storage. In particular, one may not be able to transmit, compute and store

the whole input. For such problems, classical and quantum algorithms have been proposed with the aim to reduce space complexity. On quantum computers, such algorithms usually use polynomially or even exponentially less quantum memory than their classical counterparts using classical memory.

However, quantum algorithms are generally performed on a universal quantum computer. Note that for some structures quantum gates can change continuously by slowly varying some physical parameters, and it seems that one should use a continuous set of quantum gates to describe them. However, by considering the uncertainty principle, physical parameters can only be measured with errors. Due to these errors, quantum gates with slightly different parameters can therefore often not be distinguished and should be regarded as the same quantum gate. This brings us back to a discrete set of quantum gates and a universal quantum computer. On such a universal quantum computer, only a finite set of quantum gates – the universal quantum gates – can be used directly. Other quantum gates are approximated by quantum gate array to a certain accuracy.

According to the no-programming theorem, universal quantum computers need extra memory, in particular, they need classical memory in order to store the program for the desired quantum gate array. Therefore, the length of the desired quantum gate array would determine the length of the program, which requires extra memory. In this work, we include the extra memory for programs when considering the space complexity, and show that if the extra memory is taken into account, the space complexity of the proposed quantum algorithm for the PartialMOD problem is approximately equal to the space complexity of the respective classical algorithms and that for the Equality problem is even worse. This way, the considered streaming algorithm on universal quantum computers have no advantage over their classical counterparts. Note that our result does not imply that these problems cannot be solved efficiently in a different model. Instead, it suggests that different problems may be solved more efficiently in some particular model, but not in others. We therefore see our result as an inspiration to consider quantum algorithms with respect to the framework in which they can be implemented.

2 Related Work

Classical data stream problems have been first formalized and popularized by Alon et al. [4] in order to estimate the frequency moment of a sequence using as little memory as possible. The PartialMOD [5] and Equality-like problems [23] are well-known examples of problems in this class. For the PartialMOD problem, Ambainis et al. [5] proved a tight bound of $\log p$ bits in the deterministic setting. Ablayev et al. [1, 3] proved a tight bound of n bits for the deterministic classical streaming algorithms computing Equality problems.

For the quantum version of data stream problems, Watrous [22] proved the well-known result that the complexity class $\text{PrSPACE}(s)$ is equal to the complexity class $\text{PrQSPACE}(s)$, which implies that to some extent, quantum algorithms are not better than classical algorithms with respect to their space complexity.

For the PartialMOD problem, Ablayev et al. [2] proposed a quantum algorithm that requires only 1 qubit while classical algorithms need $\log p$ bits. Ablayev et al. [3] later proposed quantum streaming algorithms for Equality Boolean functions. Their results show that some problems have both logarithmic or better quantum algorithms, whereas at least a logarithmic number of bits is needed for classical algorithms. Based on both previous results, Khadiev et al. [14, 13, 15] proposed quantum stream algorithms with constant space complexity, which is better than classical streaming algorithms that require polylogarithmically many bits. Le Gall [17] also investigated a certain variation of the Equality problem and proposed a quantum algorithm with exponentially lower space complexity (both quantum and classical) than the corresponding classical algorithm.

The field of communication complexity also investigated Equality problems. Buhrman et al. [7] introduced quantum fingerprinting and proposed to use it in communication theory. They chose the Equality problem as an example in their paper. Recently, Guan et al. [10] managed to realize the above progress experimentally.

In our paper, we focus on the space complexity of data stream problems on universal quantum computers. For such computers, the Solovay-Kitaev algorithm [8, 16, 11] states that any operator can be approximated to an accuracy of ϵ by $\log^c \frac{1}{\epsilon}$ quantum gates from a finite set of gates. Different versions of the Solovay-Kitaev algorithm consider different values of c . In [8], Dawson and Nielsen introduced a version with $c \approx 4$. Kitaev et al. [16] proposed a version with $c \approx 2$, and Harrow et al. [11] finally proved a lower bound of $c = 1$. Moreover, they showed that the corresponding algorithm exists but cannot be given explicitly.

3 Background

We will start by describing the notation for quantum computation we use in this paper. We will introduce fundamental concepts of quantum physics using the Dirac notation, and also present the Bloch sphere model, which is a geometric way to comprehend quantum algorithms. In Section 3.2, we will then clarify the Solovay-Kitaev algorithm [8] which gives a way to efficiently approximate any desired operation on a universal quantum computer with a finite set of operations. Finally, in Section 3.3, we will explain the quantum no-programming theorem [19]. This theorem points out that we must use orthonormal quantum states to perform different operators with deterministic quantum gate arrays and as such it forms the basis of a classically controlled quantum computer.

3.1 Notation

Let $|i\rangle$ denote the i -th classical state of the (complete orthonormal) computational basis of a Hilbert space. We can write a pure state of a quantum memory as a column vector $|\psi\rangle = (\alpha_1, \dots, \alpha_n)^T = \sum_{i=1}^n \alpha_i |i\rangle$. Its norm satisfies $\langle \psi | \psi \rangle = \sum_{i=1}^n |\alpha_i|^2 = 1$, where $\langle \psi |$ is the conjugate transpose of $|\psi\rangle$.

The evolution of a state can be represented by a unitary operator U . That is, given an initial state $|\psi\rangle$, the final state after applying U is $|\psi'\rangle = U|\psi\rangle$. It is easy to verify that the norm of a state does not change after an evolution.

A single-qubit memory can be represented as a point on the so-called Bloch sphere. Explicitly, a unitary operator

$$U = \begin{pmatrix} \cos(\frac{\theta}{2}) & -e^{-i\phi} \sin(\frac{\theta}{2}) \\ e^{i\phi} \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{pmatrix}.$$

corresponds to the vector $U|0\rangle$, pointing to (θ, ϕ) on the sphere, where θ is the angle between the vector and the z -axis, and ϕ is the angle between the projection of the vector onto xOy plane and the x -axis. Note that the north pole corresponds to all $(0, \phi)$ and the south pole corresponds to all (π, ϕ) .

A projective measurement can be represented by a set of orthogonal projectors followed by a normalization. That is, if we apply a measurement $\{P_i\}$ to the initial state $|\psi\rangle$, the final state becomes $|\psi'_i\rangle = P_i|\psi\rangle/\|P_i|\psi\rangle\|$ with probability $|\langle\psi|P|\psi\rangle|$. In particular, if the measurement is $\{|i\rangle\langle i|\}$ and the initial state is $\sum \alpha_i|i\rangle$, the final state is $|i\rangle$ with probability $|\alpha_i|^2$. Note that the total probability of all possible final states is 1.

3.2 Solovay-Kitaev Algorithm

In order to present the Solovay-Kitaev algorithm, we first need to introduce the concept of universality.

Definition 1 (Universal quantum gates [20]). *A set of quantum gates is universal for quantum computation if any unitary operator can be approximated to arbitrary accuracy by a quantum circuit involving only these gates.*

Note that an example of such a set can be found in Chapter 4 of [20].

Based on the well-defined universal set, we can now state the Solovay-Kitaev theorem which talks about how efficient a universal set is:

Theorem 1 (Solovay-Kitaev [8]). *There exist algorithms that can approximate any unitary operator U to an accuracy of $\|U - U_{approx}\|_2 \leq \epsilon$ with $O(\log^c \frac{1}{\epsilon})$ universal quantum gates.*

The proof of the theorem can be found in [8].

According to Harrow [11], $\Omega(\log \frac{1}{\epsilon})$ quantum gates are needed in order to approximate any unitary operator in two dimensions to an accuracy of ϵ . The Solovay-Kitaev algorithm is optimal if we disregard poly-logarithmic differences in the number of quantum gates.

The Solovay-Kitaev theorem does not exclude the possibility that we can approximate some unitary operator with a quantum gate array much shorter than $O(\log \frac{1}{\epsilon})$ to an accuracy of ϵ .

3.3 No-Programming Theorem

The no-programming theorem [19] shows that we cannot use fewer qubits than classical bits for programming if we want to implement a quantum gate array deterministically. We view our quantum computer as a unitary operator G acting on both the quantum program $|P\rangle$ and the memory $|d\rangle$. G acting on a quantum program $|P\rangle$ for unitary U results in $U|d\rangle \otimes |P'\rangle$. After measurement we get $U|d\rangle$ *deterministically*. However, G acting on a superposition of orthogonal quantum programs $\frac{1}{\sqrt{2}}(|P_1\rangle + |P_2\rangle)$ results in a superposition of orthogonal states $\frac{1}{\sqrt{2}}(U_1|d\rangle \otimes |P'_1\rangle + U_2|d\rangle \otimes |P'_2\rangle)$. Therefore, after our measurement, we obtain either $U_1|d\rangle$ or $U_2|d\rangle$ *stochastically*.

Theorem 2. *On a fixed, general purposed quantum computer, if we want to deterministically implement a quantum gate array, quantum programs $|P_1\rangle, \dots, |P_n\rangle$ performing distinct unitary operator U_1, \dots, U_n are orthogonal. The program memory is at least N -dimensional, that is, it contains at least $\log(N)$ qubits.*

The theorem shows that, when used for programming a deterministic quantum gate array, a quantum program has no advantage over a classical program, i.e. in this aspect a quantum controlled quantum computer is no better than a classically controlled quantum computer.

When used for programming a probabilistic quantum gate array, there are quantum programs that use exponentially less space but succeed with exponentially smaller probability, which is not practical. In our paper, we thus only consider classical bits for programming.

4 Data Stream Problems

In this section, we present selected examples of data stream problems and study their space complexity. Each section is organized as follows: we first introduce the problem statement and the corresponding proposed algorithm for quantum computers with a continuous set of gates. In practice, quantum computers with a continuous set of gates cannot be realized, which makes such algorithms only of theoretic interest. In the following section, we assume that our universal quantum computer first selects a certain universal set of gates, then it is asked data stream problems with any possible scale and parameter. The quantum computer should answer any possible question using the same universal set of gates. We therefore analyze the space complexity of the respective algorithm on such a universal quantum computer and show that it has no advantage over the space complexity of the best known classical algorithm.

4.1 PartialMOD Problem

In this section, we study the PartialMOD problem as presented in [15, 5, 1]. In this problem, we receive some unknown bitstring bit by bit of which we know that the number of bits with value 1 is a multiple of a given number. The task

is to determine the parity of the multiplier of this number while storing as few bits as possible in the memory.

Definition 2 (PartialMOD problem). *Let (x_1, \dots, x_n) be an input sequence of classical bits. Assume that we know in advance that $\#_1$ is a multiple of p , i.e., $\#_1 = v \cdot p$, where $\#_1$ denotes the number of ones in the string. The bits are received one by one by the algorithm. The problem is to determine the parity of v , i.e., to output $v \bmod 2$.*

Algorithm with a Continuous Set of Gates Ambainis and Yakaryilmaz [5] showed that there exists no deterministic or probabilistic algorithm to compute PartialMOD problem with $o(\log p)$ classical bits. In their paper, they also propose a quantum algorithm solving PartialMOD using only one qubit. This algorithm works as follows: There is only one qubit in the quantum memory. Let the initial state of the qubit be $|0\rangle$, which is the north pole of the Bloch sphere, and set $\theta_p = \frac{\pi}{2p}$. Each time we receive a 1 as the next bit, we apply a unitary operator

$$R(\theta_p) = \begin{pmatrix} \cos \theta_p & \sin \theta_p \\ -\sin \theta_p & \cos \theta_p \end{pmatrix}.$$

on the qubit, which is a rotation by $2\theta_p$ around y -axis on the Bloch sphere. After $v \cdot p$ steps, we receive all the input bit and get the state

$$|\psi_f\rangle = \left(\cos(v \frac{\pi}{2}) \quad -\sin(v \frac{\pi}{2}) \right)^T.$$

If $v \bmod 2 = 0$, we return to the north pole of the Bloch sphere and the final state of the qubit is $|0\rangle$. If $v \bmod 2 = 1$, we reach the south pole of the Bloch sphere and the final state is $|1\rangle$. Finally, we can measure the qubit and obtain its state.

With this procedure, we only need one qubit to solve the PartialMOD problem on quantum computers. In contrast, a classical computer requires to use $\log p$ bits, as is shown in [1].

Analysis on Universal Quantum Computers In the following, we show that the proposed quantum algorithm is not space efficient on universal quantum computers. We suppose that our universal quantum computer is able to solve any specific PartialMOD problem, which requires that we should be able to apply any $R(\theta_p)$ to the demanded accuracy. Observe that there are infinitely many choices of p , and thus infinitely many different $R(\theta_p)$. Since only finitely many gates can be selected in the universal set of a quantum computer, $R(\theta_p)$ have to be approximated by a quantum gate array, where each gate of the array is from the universal set. This leads to possibly wrong outputs. Assume therefore that we approximate $R(\theta_p)$ by $R(\theta_p + \epsilon_p)$, which satisfies

$$\|R(\theta_p) - R(\theta_p + \epsilon_p)\|_2 = 4 \sin \frac{\epsilon_p}{4}.$$

Starting with the initial state $|0\rangle$, we reach the state

$$|\psi_f\rangle = \begin{pmatrix} \cos(v\frac{\pi}{2} + vp \cdot \epsilon_p) \\ -\sin(v\frac{\pi}{2} + vp \cdot \epsilon_p) \end{pmatrix}.$$

after vp steps. With probability $\sin^2(vp \cdot \epsilon_p)$ we may get an incorrect output from the measurement. If $vp \cdot \epsilon_p$ is small enough, we can bound the probability of an incorrect output by a positive constant δ as follows

$$\frac{1}{2}vp \cdot 4 \sin\left(\frac{\epsilon_p}{4}\right) \leq \sin(vp \cdot \epsilon_p) \leq \sqrt{\delta}.$$

Therefore, an accuracy of $\frac{2\sqrt{\delta}}{vp}$ must be achieved. Such an accuracy comes at the cost of additional quantum gates.

Intuitively, applying the Solovay-Kitaev algorithm, we need a quantum gate array of $\log \frac{vp}{2\sqrt{\delta}}$. We will show next that a quantum gate array of at least $\Omega(\log(\frac{v}{\sqrt{\delta}} \log p))$ gates must be used in order to approximate $R(\theta_p)$ in the proposed algorithm to an accuracy of $\frac{2\sqrt{\delta}}{vp}$. Note that in this theorem we do not assume the optimality of the Solovay-Kitaev algorithm. Because the optimality of Solovay-Kitaev algorithm is in the sense of polylogarithmic equivalence, and the truly optimal algorithm has not been given, simply assuming this algorithm to be optimal may cause difficulties. However, even without such an assumption, Theorem 3 still shows that the quantum algorithm performs worse in some situations.

Theorem 3. *No algorithm can approximate all $R(\theta_p)$, where $\theta_p = \frac{\epsilon}{2p}$ and $p \leq p_0$, to an accuracy of $\epsilon_p = \frac{\epsilon}{2p}$ using $o(\log(\frac{1}{\epsilon} \log p_0))$ quantum gates on a universal computer, where p_0 is sufficiently large and ϵ sufficiently small. We do not assume the optimality of the Solovay-Kitaev algorithm here.*

We will not present the proof here, but the general idea of the proof is inspired by [11].

Theorem 3 implies that at least $O(\log(\frac{v}{\sqrt{\delta}} \log p_0))$ quantum gates are needed in order to approximate all $R(\theta_p)$, where $p \leq p_0$, to the demanded accuracy of $\epsilon_p = \frac{2\sqrt{\delta}}{vp}$ in order to ensure a success probability of at least $1 - \delta$. Since we have to store the arrangement of the quantum gate array for each $R(\theta_p)$, the number of classical bits required is equal to the number of gates in the quantum gate array, that is, at least $O(\log(\frac{v}{\sqrt{\delta}} \log p))$ classical bits. It is obvious that when v approaches infinity while p remains finite, the quantum algorithm for PartialMOD is not more space-efficient than the corresponding classical algorithm.

Assuming the optimality of the Solovay-Kitaev algorithm, which is discussed in [11], we can also show that in order to obtain such an accuracy, at least $O(\log(\frac{vp}{\sqrt{\delta}}))$ quantum gates must be used by any algorithm.

Theorem 4. *Let p be sufficiently large. No algorithm can approximate all $R(\theta_p)$, where $\theta_p = \frac{\pi}{2p}$ to any accuracy ϵ_p with $o(\log(\frac{1}{\epsilon_p}))$ quantum gates on a universal computer, if the optimality of the Solovay-Kitaev algorithm is assumed.*

This theorem can be proved by contradiction: if one can approximate these operators with $o(\log(\frac{1}{\epsilon_p}))$ quantum gates, then it is possible to construct a better algorithm than the Solovay-Kitaev algorithm.

Theorem 4 shows that there exists some $R(\theta_p)$ for which we need at least $\Omega(\log \frac{1}{\epsilon_p}) = \Omega(\log v + \log p)$ gates in order to approximate it to the demanded accuracy of $\epsilon_p = \frac{2\sqrt{\delta}}{vp}$, assuming the optimality of the Solovay-Kitaev theorem. Since we have to store the arrangement of the quantum gate array, the number of classical bits needed is $\Omega(\log v + \log p)$. When v or p approach infinity, the quantum algorithm is not more space-efficient than the classical algorithm.

Theorem 3 and Theorem 4 are proved under different assumptions. Together they show that the previously proposed algorithm is not more space-efficient than its classical counterpart under certain conditions.

4.2 Equality Problem

In this section, we investigate the so-called Equality problem [3, 18, 7]. In this problem, two bitstrings are received once one after another bit by bit. The task is to find out whether these two given sequences of bits are equal while storing a minimal amount of information.

Definition 3 (Equality problem). *We are given an input sequence $(x, y) = (x_1, \dots, x_n, y_1, \dots, y_n)$ of classical bits. The bits are received one by one by the algorithm. We do not receive any bit of y before we have received all bits of x . The output is whether x and y are equal, i.e., $O = \delta(\|x - y\|) = 1, x = y; 0, x \neq y$.*

Algorithm with a Continuous Set of Gates According to [21, 18], there is no classical deterministic algorithm that can compute the equality problem with $o(n)$ classical bits, while there is a randomized algorithm, i.e. Karp-Rabin algorithm, with a space complexity of $O(\log n)$ [12]. There also exists a quantum algorithm that has the same performance. Ablayev et al. [3] applied quantum fingerprinting in a quantum streaming algorithm to solve this problem with $O(\log n)$ qubits on a quantum computer with a continuous set of gates. Their algorithm seems to have the same performance as the Karp-Rabin algorithm.

The quantum memory is divided into two parts. The first part is the first qubit, whose state is in a 2-dimensional space. The second part contains the remaining $\log t$ qubits in a t -dimensional space. The initial state is $|0\rangle \otimes |0\rangle$. The strategy is to first apply Hadamard gates on all qubits of the second part and receive $\frac{1}{\sqrt{t}}|0\rangle \otimes \sum_{j=1}^t |j\rangle$. If we receive a 1 for x_i , we apply a unitary operator $U_i = \sum_{j=1}^t \{R(\theta_{ij}) \otimes |j\rangle\langle j|\}$, where $R(\theta_{ij})$ is a rotation on the first qubit by $\theta_{ij} = \frac{2\pi m_j}{2^{i+1}}$ and m_j some positive integer. If we receive a 1 for y_i , we replace $R(\theta_{ij})$ with $R(-\theta_{ij})$ in U_i . After receiving all the input bits, the state is

$$\frac{1}{\sqrt{t}} \sum_j R\left(\frac{2\pi m_j(x-y)}{2^{n+1}}\right) |0\rangle \otimes |j\rangle.$$

Then we apply Hadamard gates on all qubits in the second part. The final state becomes

$$\frac{1}{t} \sum_j \left(\cos \frac{2\pi m_j (x - y)}{2^{n+1}} \right) |0\rangle \otimes |0\rangle + rest.$$

If $x = y$, we return to the initial state. If $x \neq y$, we reach a non-initial state. We require that the coefficient of $|0\rangle \otimes |0\rangle$ in the final state is approximately a delta function, that is,

$$\left\| \frac{1}{t} \sum_j \cos \frac{2\pi m_j (x - y)}{2^{n+1}} - \delta(x - y) \right\| \leq \sqrt{\epsilon}.$$

Then we can easily verify whether $x = y$ by checking whether we get $|0\rangle \otimes |0\rangle$ after measurement. If $x = y$, we obtain $|0\rangle$ with probability 1. If $x \neq y$, we obtain $|0\rangle \otimes |0\rangle$ with probability less than ϵ .

If we apply discrete Fourier transform to $\delta(g)$, that is, m_j s take $t = 2^n$ integers from 0 to 2^n , ϵ is exactly 0. But in that case we need $\log(t) = n$ qubits. It is however possible that if we do not apply discrete Fourier transform, that is, m_j s only take $t = O(n \log \frac{1}{\epsilon}) \ll 2^n$ integers from 0 to 2^n , ϵ is also bounded. The next theorem states this fact, its proof can be found in [3].

Theorem 5. *There exists a set of $t > \frac{2}{\epsilon} \ln(2m)$ elements, $\{m_j, j = 1, \dots, t\}$ such that*

$$\frac{1}{t} \left\| \sum_j \cos \left(\frac{2\pi m_j g}{m} \right) \right\| \leq \sqrt{\epsilon}, \forall g \neq 0.$$

Theorem 5 implies that there exists a set of $t = \frac{2}{\epsilon} \ln(2m) + 1$ elements, $\{m_j, j = 1, \dots, t\}$, which ensures $\cos \left(\frac{2\pi k_i g}{m} \right)$'s to almost cancel each other. Indeed, if we select integers uniformly at random from 0 to $m - 1$, we are likely to get such m_j . By applying Theorem 5 to the Equality problem, we only need $\log(n) + 1$ qubits on quantum computers with a continuous set of gates, which is exponentially better than n bits deterministic algorithms on computers, as was shown by Babai et al. [6].

Analysis on Universal Quantum Computers The proposed algorithm to solve the Equality problem is not space-efficient on universal quantum computers. Similar to the PartialMOD problem, we will first bound the accuracy of each operator. Let us denote the probability for the algorithm to accept the input, i.e., in the case where the final state is $|0\rangle \otimes |0\rangle$, as $\Pr(x, y)$. Further, assume that it is possible to approximate the operator $R(\theta_{ij})$ to an accuracy of δ_{ij} . After applying Theorem 5, the partial derivative of $\Pr(x, y)$ becomes

$$\delta \Pr(x, y) \leq \frac{\sqrt{\epsilon}}{t} \left\| \sum_{j=1}^t \sum_i \sin \left(\frac{m_j \pi 2^i (x - y)_i}{2^n} \right) \delta_{ij} \right\|.$$

Differently than in the PartialMOD problem, it is challenging to bound the accuracy for the Equality problem precisely. Instead, we simply assume we need $\Omega(1)$ gates for each $R(\theta_{ij})$. The following theorem defines an upper bound on the accuracy needed, and shows our simple assumption is reasonable.

Theorem 6. *Let $|\delta_{ij}| \leq \frac{1}{n}$. Then, there exists a set of $t = \frac{2}{\epsilon}(n+3)$ elements $m_j, j = 1, \dots, t$, such that the following two inequalities are satisfied*

$$\frac{1}{t} \left\| \sum_j \cos\left(\frac{\pi m_j g}{2^n}\right) \right\| \leq \sqrt{\epsilon}, \quad \forall g \neq 0,$$

and

$$\frac{1}{t} \left\| \sum_{j=1}^t \sum_i \sin\left(\frac{m_j \pi 2^i g_i}{2^n}\right) \delta_{ij} \right\| \leq \sqrt{\epsilon}, \quad \forall g \neq 0.$$

Here, we prove it via a method similar to that of Theorem 5, shown in [3].

The algorithm for the Equality problem will succeed as long as we reach an accuracy of $\frac{1}{n}$ for a suitably chosen set of $O(n)$ elements. In order to achieve such accuracy, we need at most $O(\log^4 n)$ quantum gates according to the Solovay-Kitaev theorem. Since we need to apply at least one quantum gate in order to be able to implement an operator, it is reasonable to assume that we need at least $\Omega(1)$ quantum gates for each operator to achieve such accuracy.

Now we can analyze the space complexity, for which we also take into account classical bits. When we perform the above algorithm we need to store the set $\{m_j\}$, since the set $\{m_j\}$ is not chosen arbitrarily. There are two natural ways to do so. One way is to store $\{m_j\}$ directly: consider m_j that range from 0 to 2^n , and thus need n classical bits. We have n such integers, and thus at least $\Omega(n^2)$ bits are needed. This strategy requires even more bits than a classical brute force method which saves all $O(n)$ bits of the input. The second way is to store $\{R(\theta_{ij})\}$: note that $R(\theta_{ij})$ need at least $\Omega(1)$ quantum gates for each operator, and thus each need $\Omega(1)$ classical bits. Since we have n^2 such operators in our algorithm, at least $\Omega(n^2)$ bits of storage are needed, which is more than that in the classical deterministic algorithm. In the following theorem, we provide a more rigorous proof.

Theorem 7. *At least $\Omega(n^2)$ bits are needed in order to store a set $\{m_j, j = 1..t\}$ where $m_j \in [0, 2^n - 1]$ and $t = \frac{2}{\epsilon}(n+3)$ without pre-knowledge of the set.*

Proof. We first consider the classical case. The number of possible choices in the classical case is

$$C_t^{2^n} = \frac{2^n!}{t! \cdot (2^n - t)!}.$$

The information entropy of knowing a certain choice from all possible choices with equal possibility is $S = \ln(C_t^{2^n})$. Consider when n is sufficiently large, $2^n \gg t = \frac{2}{\epsilon}(n+3)$, use $\ln(1+x) \approx x$ and $\ln(x!) \approx x \ln(x) - x$, we have

$$S = \ln(2^n) + \dots + \ln(2^n - t + 1) - \ln(t!) \approx nt - \frac{t^2}{2^n} - t \ln(t) + t = O(n^2).$$

Since the number of bits required is linearly dependent on the information entropy, $O(n^2)$ bits are needed in order to store this set.

We next consider the quantum case. The set $\{m_j, j = 1 \dots t, m_j \in [0, 2^n - 1]\}$ is used to program our quantum computer. Due to the quantum no-programming theorem in 3.3, quantum programs have no advantage over the classical program with respect to space complexity. Therefore, $\Omega(n^2)$ bits or qubits are needed to store this set.

Therefore, the considered algorithm for the Equality problem has no advantage over the classical deterministic algorithm.

5 Conclusion

Based on the Solovay-Kitaev algorithm, we investigated the space complexity of streaming algorithms on a universal computer when only a finite number of quantum gates are available. We used the PartialMOD problem and the Equality problem to analyze the quantum streaming algorithms in systems where classical bits are used in order to control quantum gates. By applying the Solovay-Kitaev algorithm we concluded that the considered quantum streaming algorithms do not beat their classical counterparts in this system.

Our work shows that not all quantum streaming algorithms can perform well on a universal quantum computer. There are also data stream problems for which quantum algorithms may perform well on a universal quantum computer. One example is the variation of the Equality problem proposed in [17]. In this problem, the input is repeated many times, which is different from the Equality problem discussed in this paper, where we receive the input only once. Another possible candidate is the problem based on the universal (ϵ, l, m) -code of matrices proposed by Sauerhoff et al. in [21] and Gavinsky et al. [9], where the input directly corresponds to a quantum gate array, and one can therefore save space when storing quantum gates for application. By comparing these algorithms, we conclude that a framework can be extremely efficient for a certain set of problems and corresponding algorithms, but not necessarily for all problems. We therefore think that the space complexity of algorithms should be analyzed with respect to the framework of the quantum computer in which they can be implemented.

References

1. Ablayev, F., Gainutdinova, A., Karpinski, M., Moore, C., Pollett, C.: On the computational power of probabilistic and quantum branching program. *Information and Computation* **203**(2) (2005)
2. Ablayev, F., Gainutdinova, A., Khadiev, K., Yakaryılmaz, A.: Very Narrow Quantum OBDDs and Width Hierarchies for Classical OBDDs. In: *Descriptional Complexity of Formal Systems*. Springer International Publishing (2014)
3. Ablayev, F., Khasianov, A., Vasiliev, A.: On complexity of quantum branching programs computing equality-like boolean functions. *Electronic Colloquium on Computational Complexity* (2010)

4. Alon, N., Matias, Y., Szegedy, M.: The Space Complexity of Approximating the Frequency Moments. In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. STOC (1996)
5. Ambainis, A., Yakaryılmaz, A.: Superiority of exact quantum automata for promise problems. *Information Processing Letters* **112**(7) (2012)
6. Babai, L., Kimmel, P.G.: Randomized simultaneous messages: Solution of a problem of Yao in communication complexity. In: Proceedings of the 12th Annual IEEE Conference on Computational Complexity. CCC (1997)
7. Buhrman, H., Cleve, R., Watrous, J., de Wolf, R.: Quantum Fingerprinting. *Physical Review Letters* **87** (2001)
8. Dawson, C.M., Nielsen, M.A.: The Solovay-Kitaev Algorithm. *Quantum Information and Computation* **6**(1) (2006)
9. Gavinsky, D., Kempe, J., Kerenidis, I., Raz, R., de Wolf, R.: Exponential Separations for One-way Quantum Communication Complexity, with Applications to Cryptography. In: Proceedings of the Thirty-ninth Annual ACM Symposium on Theory of Computing. STOC (2007)
10. Guan, J.Y., Xu, F., Yin, H.L., Li, Y., Zhang, W.J., Chen, S.J., Yang, X.Y., Li, L., You, L.X., Chen, T.Y., Wang, Z., Zhang, Q., Pan, J.W.: Observation of Quantum Fingerprinting Beating the Classical Limit. *Physical Review Letters* **116** (2016)
11. Harrow, A.W., Recht, B., Chuang, I.L.: Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics* **43**(9) (2002)
12. Karp, R.M., Rabin, M.O.: Efficient randomized pattern-matching algorithms. *IBM journal of research and development* **31**(2), 249–260 (1987)
13. Khadiev, K., Khadieva, A., Kravchenko, D., Rivosh, A.: Quantum versus Classical Online Algorithms with Advice and Logarithmic Space (2017)
14. Khadiev, K., Khadieva, A., Mannapov, I.: Quantum online algorithms with respect to space complexity. *Lobachevskii Journal of Mathematics* **39** (2017)
15. Khadiev, K., Ziatdinov, M., Mannapov, I., Khadieva, A., Yamilov, R.: Quantum Online Streaming Algorithms with Constant Number of Advice Bits (2018)
16. Kitaev, A.Y., Shen, A., Vyalyi, M.N.: Classical and Quantum Computation. American Mathematical Society, Boston, MA, USA (2002)
17. Le Gall, F.: Exponential separation of quantum and classical online space complexity. *Theory of Computing Systems* **45** (2009)
18. Newman, I., Szegedy, M.: Public vs. Private Coin Flips in One Round Communication Games (Extended Abstract). In: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing. STOC (1996)
19. Nielsen, M.A., Chuang, I.L.: Programmable Quantum Gate Arrays. *Physical Review Letters* **79** (1997)
20. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press (2010)
21. Sauerhoff, M., Sieling, D.: Quantum branching programs and space-bounded nonuniform quantum complexity. *Theoretical Computer Science* **334**(1) (2005)
22. Watrous, J.H.: Space-bounded Quantum Computation. Ph.D. thesis, The University of Wisconsin - Madison (1998)
23. Yao, A.C.C.: Some Complexity Questions Related to Distributive Computing (Preliminary Report). In: Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing. STOC (1979)