ASIAN JOURNAL OF MATHEMATICS AND PHYSICS
VOLUME 3, ISSUE 1, 2019, 1-12
HTTP://AJMP.SCIENCEASIA.ASIA

Various New Forms of the Bernstein-Vazirani Algorithm Beyond Qubit Systems

Koji Nagata,[1] Santanu Kumar Patro,[2] Han Geurdes,[3] Shahrokh Heidari,[4] Do Ngoc Diep,[5,6] and Tadao Nakamura[7]

[1]*Department of Physics, Korea Advanced Institute of Science and Technology, Daejeon 34141, Korea*
E-mail: ko_mi_na@yahoo.co.jp
[2]*Incessive Education & Research (P) Ltd, Odisha-761055, India*
[3]*Geurdes Datascience, KvK 64522202, C vd Lijnstraat 164, 2593 NN, Den Haag Netherlands*
[4]*Young Researchers and Elite Club, Kermanshah Branch, Islamic Azad University, Kermanshah, Iran*
[5]*TIMAS, Thang Long University, Nghiem Xuan Yem road, Hoang Mai district, Hanoi, Vietnam*
[6]*Institute of Mathematics, VAST, 18 Hoang Quoc Viet road, Cau Giay district, Hanoi, Vietnam*
[7]*Department of Information and Computer Science, Keio University,*
3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223-8522, Japan

(Received 29 May, 2019)

Here, we present various new forms of the Bernstein-Vazirani algorithm beyond qubit systems. First, we review the Bernstein-Vazirani algorithm for determining a bit string. Second, we discuss the generalized Bernstein-Vazirani algorithm for determining a natural number string. The result is the most notable generalization. Thirdly, we discuss the generalized Bernstein-Vazirani algorithm for determining an integer string. Finally, we discuss the generalized Bernstein-Vazirani algorithm for determining a complex number string. The speed of determining the strings is shown to outperform the best classical case by a factor of the number of the systems in every cases. Additionally, we propose a method for calculating many different matrices simultaneously. The speed of solving the problem is shown to outperform the classical case by a factor of the number of the elements of them. We hope our discussions will give a first step to the quantum simulation problem.

PACS numbers: 03.67.Ac, 03.67.Lx, 03.65.Ca
Keywords: Quantum algorithms, Quantum computation, Formalism

I. INTRODUCTION

The Bernstein-Vazirani algorithm aims at determining a bit string [1, 2]. Regardless of entanglement properties, an experimental implementation of a quantum algorithm with the aim of solving the Bernstein-Vazirani parity problem is investigated [3]. Using three qubits, Brainis et al. suggests fiber-optics implementations of the Deutsch-Jozsa and Bernstein-Vazirani quantum algorithms [4]. Moreover, a variant of the algorithm for quantum learning algorithm, a robust one against noise, is studied by Cross et al. [5]. In 2015, based on the Bernstein-Vazirani algorithm, Li and Yang investigated the influences of variables on Boolean functions and its applications [6]. The Bernstein-Vazirani algorithm is also a versatile in quantum key distribution [7, 8] and in a transport implementation with ion qubits [9]. In 2016, Krishna et al. introduced a generalization of the Bernstein-Vazirani algorithm to qudit systems [10]. A simple algorithm for complete factorization of an N-partite pure quantum state is discussed [11]. Fujikawa et al. discusses a classical limit of Grover's algorithm induced by dephasing: coherence versus entanglement [12]. Quantum dialogue protocol based on Grover's search algorithms is presented [13].

Using a Boolean-valued function (the outputs are either 0 or 1), the Bernstein-Vazirani algorithm was extended for determining the values of the function by Nagata et al. in 2017 [14, 15]. In the method, the values of the function are restricted to $\{0,1\}$. With respect to the provided extension, not only a calculation of multiplications, but also the root finding problem can be carried out based on quantum computation [15, 16].

To determine the values of a function, which are extended to the natural numbers, Nagata et al. uses the Bernstein-Vazirani algorithm efficiently [17]. The extended algorithm is capable of determining a natural number string instead of a bit string. It is worth mentioning that a homogeneous linear function can be determined by using the provided extension as a quantum algorithm [18].

Likewise, an extension of the Bernstein-Vazirani algorithm can be utilized to determine the values of a function, which are extended to integers according to Ref. [19], meaning that the extended algorithm aims at figuring out an integer string instead of a natural number string. Moreover, quantum communication can be performed by determining a matrix using the provided extension [20].

Integers are more general than natural numbers. Complex numbers are more general than integers. In fact, we solve many problems by using complex numbers and we cannot answer many problems without them. Hence, we definitely need complex numbers. Thus, we want to generalize the Bernstein-Vazirani algorithm for determining the values of a special function that the values are complex numbers because the utility of it becomes very wider than previous ones. For example, we would find all the roots in the root finding problem simultaneously. Also, we can plot a graph of a given function immediately.

Quantum mechanics is formulated by the matrix theory. So it is useful to treat matrices by quantum computing if we want to simulate quantum phenomena by such a computer. For example, it is desirable if quantum computers calculate highly complicated quantum chemistry phenomena. Now, we generalize the Bernstein-Vazirani algorithm from a bit string into a matrix string. We hope our discussions will give a first step to the quantum simulation problem.

In this article, we present various new forms of the Bernstein-Vazirani algorithm beyond qubit systems. First, we review the Bernstein-Vazirani algorithm for determining a bit string. Second, we discuss the generalized Bernstein-Vazirani algorithm for determining a natural number string. Thirdly, we discuss the generalized Bernstein-Vazirani algorithm for determining an integer string. Finally, we discuss the generalized Bernstein-Vazirani algorithm for determining a complex number string. The speed of determining the strings is shown to outperform the best classical case by a factor of the number of the systems in every cases. We hope the generalization would improve quantum algorithm science.

Additionally, we propose a method for calculating many different matrices $A, B, C, \ldots$ into $g(A), g(B), g(C), \ldots$ simultaneously. The speed of solving the problem is shown to outperform the classical case by a factor of the number of the elements of them.

The article is organized as follows.

In Sec. II, we review the Bernstein-Vazirani algorithm for determining a bit string. The algorithm has the feature of the Hadamard transform of a quantum state in a two-dimensional space to solve the problem of finding a bit string.

In Sec. III, we discuss the generalized Bernstein-Vazirani algorithm for determining a natural number string. The generalized algorithm presented here has the feature of the normal Fourier transform of a quantum state in a finite-dimensional space to solve the problem of finding a string of natural numbers.

In Sec. IV, we discuss the generalized Bernstein-Vazirani algorithm for determining an integer string. The generalized algorithm presented here has the feature of a general discrete non-unitary transform of a quantum state in a finite-dimensional space to solve the problem of finding a string of integers.

In Sec. V, we discuss the generalized Bernstein-Vazirani algorithm for determining a complex number string. The generalized algorithm presented here has the feature of a general continuous non-unitary transform of a quantum state in an infinite-dimensional space to solve the problem of finding a string of complex numbers.

In Sec. VI, we propose a method for calculating many different matrices $A, B, C, \ldots$ into $g(A), g(B), g(C), \ldots$ simultaneously.

Section VII concludes the article.

II. ALGORITHM FOR DETERMINING A BIT STRING

In this section, we review the Bernstein-Vazirani algorithm.

We suppose that the following sequence of complex numbers is given

$$a_1, a_2, a_3, \ldots, a_N. \tag{1}$$

We introduce a special function

$$g : \mathbf{C} \rightarrow \{0,1\}. \tag{2}$$

Our goal is for determining the following values (as a bit string)

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N). \tag{3}$$

Recall that in the best classical case, we need N queries, that is, N separate evaluations of the function (2). In case of the Bernstein-Vazirani algorithm, we shall require a single query.

We define

$$g(a) = (g(a_1), g(a_2), g(a_3), \ldots, g(a_N)), \tag{4}$$

where each entry of $g(a)$ is a bit. Here $g(a) \in \{0,1\}^N$. We define $f(x)$ as follows:

$$\begin{aligned} f(x) &= (g(a) \cdot x) \bmod 2 \equiv g(a) \odot x \\ &= \{g(a_1)x_1 + g(a_2)x_2 + \cdots + g(a_N)x_N\} \bmod 2, \end{aligned} \tag{5}$$

where $x = (x_1, \ldots, x_N)$ and $x_j \in \{0,1\}$. The entries of x and $g(a_j)$ are in $\{0,1\}$. Let us follow the quantum states through our algorithm.

The input state is

$$|\psi_0\rangle = |0\rangle^{\otimes N}|1\rangle \tag{6}$$

where $|0\rangle^{\otimes N} = \overbrace{|0\rangle \otimes |0\rangle \otimes \ldots \otimes |0\rangle}^{N}$. After the componentwise Hadamard transforms on the state (6)

$$\overbrace{H|0\rangle \otimes H|0\rangle \otimes \ldots \otimes H|0\rangle}^{N} \otimes H|1\rangle \tag{7}$$

we have, for quantum gates and their theory,

$$|\psi_1\rangle = \sum_{x \in \{0,1\}^N} \frac{|x\rangle}{\sqrt{2^N}} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. \tag{8}$$

Next, the function f is evaluated using

$$U_f|x, y\rangle = |x, y \oplus f(x)\rangle \tag{9}$$

in giving

$$|\psi_2\rangle = (-1)^{f(x)}|\psi_1\rangle. \tag{10}$$

Here $y \oplus f(x)$ is the bitwise XOR (exclusive OR) of y and $f(x)$. By checking the cases $x_1 = 0$ and $x_1 = 1$ separately, we see that for a single qubit

$$H|x_1\rangle = \sum_{z_1}(-1)^{x_1 z_1}|z_1\rangle/\sqrt{2}. \tag{11}$$

Thus we have

$$\begin{aligned} &H^{\otimes N}|x_1, \ldots, x_N\rangle \\ &= \frac{\sum_{z_1,\ldots,z_N}(-1)^{\{x_1 z_1 + \cdots + x_N z_N\}(mod 2)}|z_1, \ldots, z_N\rangle}{\sqrt{2^N}}. \end{aligned} \tag{12}$$

Here, xz is in modulo 2. This can be summarized more succinctly in the very useful equation

$$H^{\otimes N}|x\rangle = \frac{\sum_z (-1)^{x \odot z}|z\rangle}{\sqrt{2^N}} \tag{13}$$

where $x \cdot z$ is the bitwise inner product of x and z, modulo 2. Using the equation (10) and (13), we can now evaluate $H^{\otimes N}|\psi_2\rangle = |\psi_3\rangle$

$$|\psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x \odot z + f(x)}|z\rangle}{2^N} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. \tag{14}$$

Thus we have

$$|\psi_3\rangle = \sum_z \sum_x \frac{(-1)^{x \odot z + g(a) \odot x}|z\rangle}{2^N} \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]. \tag{15}$$

We can see that

$$\sum_x (-1)^{x \odot z + g(a) \odot x} = 2^N \delta_{g(a), z}. \tag{16}$$

Therefore, the sum is zero if $z \neq g(a)$ and is 2^N if $z = g(a)$. Thus we have

$$\begin{aligned}|\psi_3\rangle &= \sum_z \sum_x \frac{(-1)^{x\odot z + g(a)\odot x}|z\rangle}{2^N}\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\ &= \sum_z \frac{2^N \delta_{g(a),z}|z\rangle}{2^N}\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \\ &= |(g(a_1), g(a_2), \cdots, g(a_N))\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right] \quad (17)\end{aligned}$$

from which

$$|(g(a_1), g(a_2), \cdots, g(a_N))\rangle \quad (18)$$

can be obtained. That is to say, if we measure $|(g(a_1), g(a_2), \cdots, g(a_N))\rangle$ then we can retrieve the following values (as a bit string)

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N) \quad (19)$$

using a single query.

III. EXTENSION TO A NATURAL NUMBER STRING

Let us discuss the Bernstein-Vazirani algorithm of determining a natural number string. The detail calculations of the section can be seen in Ref. [19].

We suppose that the following sequence of complex numbers is given

$$a_1, a_2, a_3, \ldots, a_N. \quad (20)$$

We introduce a special function

$$g : \mathbf{C} \to \{0, 1, 2, 3, ..., d-1\}. \quad (21)$$

Our aim is to determine the values below as a string of natural numbers,

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N). \quad (22)$$

Recall that in the best classical case, we need N queries, that is, N separate evaluations of the function (21). In case of the generalized Bernstein-Vazirani algorithm, we shall require a single query.

We introduce a positive integer d. Throughout the discussion, we consider the problem in modulo d. Assume the following

$$0 \leq \overbrace{g(a_1), g(a_2), g(a_3), \ldots, g(a_N)}^{N} \leq d-1, \quad (23)$$

where $g(a_j) \in \{0, 1, ..., d-1\}$, and we define

$$g(a) = (g(a_1), g(a_2), g(a_3), \ldots, g(a_N)), \quad (24)$$

where each entry of $g(a)$ is a natural number. Here $g(a) \in \{0, 1, ..., d-1\}^N$. We define $f(x)$ as follows:

$$\begin{aligned}f(x) &= (g(a)\cdot x) \bmod d \equiv g(a) \odot x \\ &= \{g(a_1)x_1 + g(a_2)x_2 + \cdots + g(a_N)x_N\} \bmod d, \quad (25)\end{aligned}$$

where $x = (x_1, ..., x_N) \in \{0, 1, ..., d-1\}^N$. Let us follow the quantum states through the algorithm.

The input state is

$$|\psi_0\rangle = |0\rangle^{\otimes N}|d-1\rangle, \quad (26)$$

where $|0\rangle^{\otimes N}$ means $\overbrace{|0, 0, ..., 0\rangle}^{N}$. Here $|0\rangle$ and $|d-1\rangle$ are quantum states in a d-dimensional space. We discuss the Fourier transform of $|0\rangle$

$$|0\rangle \to \sum_{y=0}^{d-1} \frac{|y\rangle}{\sqrt{d}}. \quad (27)$$

We define a quantum state in a d-dimensional space $|\phi\rangle$ as follows:

$$|\phi\rangle = \frac{1}{\sqrt{d}}(\omega^d|0\rangle + \omega^{d-1}|1\rangle + \cdots + \omega|d-1\rangle), \quad (28)$$

where $\omega = e^{2\pi i/d}$. In the following, we discuss the Fourier transform of $|d-1\rangle$

$$|d-1\rangle \to |\phi\rangle. \quad (29)$$

The Fourier transform of $|x_1 ... x_N\rangle$ is as follows:

$$|x_1 ... x_N\rangle \to \sum_{z\in K} \frac{\omega^{z\odot x}|z\rangle}{\sqrt{d^N}}, \quad (30)$$

where $K = \{0, 1, ..., d-1\}^N$ and z is $(z_1, z_2, ..., z_N)$. Here, for completeness, $\sum_{z\in K}$ is a shorthand to the compound sum

$$\sum_{z_1\in\{0,1,...,d-1\}} \cdots \sum_{z_N\in\{0,1,...,d-1\}}. \quad (31)$$

After the componentwise Fourier transforms of the first N quantum states in a d-dimensional space and after the Fourier transform of the quantum state in a d-dimensional space $|d-1\rangle$ in (26)

$$\overbrace{F|0\rangle \otimes F|0\rangle \otimes ... \otimes F|0\rangle}^{N} \otimes F|d-1\rangle, \tag{32}$$

we have, for quantum gates and their theory,

$$|\psi_1\rangle = \sum_{x\in K} \frac{|x\rangle}{\sqrt{d^N}} |\phi\rangle. \tag{33}$$

Here, the notation $F|0\rangle$ means the Fourier transform of $|0\rangle$ and the notation $F|d-1\rangle$ means the Fourier transform of $|d-1\rangle$.

We introduce $O_{f(x)}$ gate

$$O_{f(x)}|x\rangle|j\rangle = |x\rangle|(f(x)+j) \bmod d\rangle, \tag{34}$$

where

$$f(x) = g(a)\cdot x \bmod d = g(a)\odot x. \tag{35}$$

We have the following formula by phase kick-back

$$O_{f(x)}|x\rangle|\phi\rangle = \omega^{f(x)}|x\rangle|\phi\rangle. \tag{36}$$

We have $|\psi_2\rangle$, by operating $O_{f(x)}$ to $|\psi_1\rangle$,

$$O_{f(x)}|\psi_1\rangle = |\psi_2\rangle = \sum_{x\in K} \frac{\omega^{f(x)}|x\rangle}{\sqrt{d^N}} |\phi\rangle. \tag{37}$$

After the Fourier transform of $|x\rangle$, using the previous equations (30) and (37), we can now evaluate $|\psi_3\rangle$ as follows:

$$|\psi_3\rangle = \sum_{z\in K}\sum_{x\in K} \frac{(\omega)^{x\odot z + g(a)\odot x}|z\rangle}{d^N} |\phi\rangle. \tag{38}$$

Notice

$$\sum_{x\in K} (\omega)^{x\odot(z+g(a))} = d^N \delta_{z,\vec{d}-g(a)}, \tag{39}$$

where $\vec{d} = \overbrace{(d,d,...,d)}^{N}$. Therefore, the above summation is zero if $z \neq \vec{d} - g(a)$ and the above summation is d^N if $z = \vec{d} - g(a)$. Thus we have

$$\begin{aligned} |\psi_3\rangle &= \sum_{z\in K} \frac{d^N \delta_{z,\vec{d}-g(a)}|z\rangle}{d^N} |\phi\rangle \\ &= |\vec{d} - (g(a_1), g(a_2), g(a_3), \ldots, g(a_N))\rangle|\phi\rangle \end{aligned} \tag{40}$$

from which

$$|\vec{d} - (g(a_1), g(a_2), g(a_3), \ldots, g(a_N))\rangle \tag{41}$$

can be obtained. That is to say, if we measure the first N quantum states in a d-dimensional space of the state $|\psi_3\rangle$, that is, $|\vec{d} - (g(a_1), g(a_2), g(a_3), \ldots, g(a_N))\rangle$, then we can retrieve the following values (as a natural number string)

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N) \tag{42}$$

using a single query.

IV. EXTENSION TO AN INTEGER STRING

We present an algorithm of determining the values of a function that are extended to integers. That is, the extended algorithm determines an integer string instead of a natural number string. This is a different point. Here, we use the general non-unitary transform instead of the Fourier transform. The transform is a straightforward generalization of the Fourier transform. The detail calculations of the section can be seen in Ref. [19].

We suppose that the following sequence of complex numbers is given

$$a_1, a_2, a_3, \ldots, a_N. \tag{43}$$

We introduce a special function

$$g : \mathbf{C} \to \{-(d-1), ..., -2, -1, 0, 1, 2, ..., d-1\}. \tag{44}$$

Our aim is to determine the values below as a string of natural numbers,

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N). \tag{45}$$

Recall that in the best classical case, we need N queries, that is, N separate evaluations of the function (44). In case of the generalized Bernstein-Vazirani algorithm, we shall require a single query.

We introduce a positive integer d. Throughout the discussion, we consider the problem in modulo d. Assume the following

$$-(d-1) \le \overbrace{g(a_1), g(a_2), g(a_3), \ldots, g(a_N)}^{N} \le d-1, \tag{46}$$

where $g(a_j) \in \{-(d-1),...,-1,0,1,...,d-1\}$, and we define

$$g(a) = (g(a_1), g(a_2), g(a_3), \ldots, g(a_N)), \tag{47}$$

where each entry of $g(a)$ is an integer. Here $g(a) \in \{-(d-1),...,-1,0,1,...,d-1\}^N$. We define $f(x)$ as follows:

$$\begin{aligned} f(x) &= (g(a)\cdot x) \bmod d \equiv g(a) \odot x \\ &= \{g(a_1)x_1 + g(a_2)x_2 + \cdots + g(a_N)x_N\} \bmod d, \end{aligned} \tag{48}$$

where $x = (x_1,...,x_N) \in \{-(d-1),...,-1,0,1,...,d-1\}^N$. Let us follow the quantum states through the algorithm.

The input state is

$$|\psi_0\rangle = |0\rangle^{\otimes N}|d-1\rangle, \tag{49}$$

where $|0\rangle^{\otimes N}$ means $\overbrace{|0,0,...,0\rangle}^{N}$. Here $|0\rangle$ is a quantum state in a $(2d-1)$-dimensional space and $|d-1\rangle$ is a quantum state in a d-dimensional space. We discuss the general transform of $|0\rangle$

$$|0\rangle \rightarrow \sum_{y=-(d-1)}^{d-1} \frac{|y\rangle}{\sqrt{2d-1}}. \tag{50}$$

A calculation shows that this is not a unitary operation; (in fact, it does not have full rank).

We define a quantum state in a d-dimensional space $|\phi\rangle$ as follows:

$$|\phi\rangle = \frac{1}{\sqrt{d}}(\omega^d|0\rangle + \omega^{d-1}|1\rangle + \cdots + \omega|d-1\rangle), \tag{51}$$

where $\omega = e^{2\pi i/d}$. In the following, we discuss the Fourier transform of $|d-1\rangle$

$$|d-1\rangle \rightarrow |\phi\rangle. \tag{52}$$

The general transform of $|x_1...x_N\rangle$ is as follows:

$$|x_1...x_N\rangle \rightarrow \sum_{z\in K} \frac{\omega^{z\odot x}|z\rangle}{\sqrt{(2d-1)^N}}, \tag{53}$$

where $K = \{-(d-1),...,-1,0,1,...,d-1\}^N$ and z is $(z_1,z_2,...,z_N)$. $z_j \in \{-(d-1),...,-1,0,1,...,d-1\}$. This is not a unitary operation. Here, for completeness, $\sum_{z\in K}$ is a shorthand to the compound sum

$$\sum_{z_1\in\{-(d-1),...,-1,0,1,...,d-1\}} \cdots \sum_{z_N\in\{-(d-1),...,-1,0,1,...,d-1\}}. \tag{54}$$

After the componentwise general transforms of the first N quantum states in a $(2d-1)$-dimensional space and after the Fourier transform of the quantum state in a d-dimensional space $|d-1\rangle$ in (49)

$$\overbrace{G|0\rangle \otimes G|0\rangle \otimes ... \otimes G|0\rangle}^{N} \otimes F|d-1\rangle, \tag{55}$$

we have, for quantum gates and their theory,

$$|\psi_1\rangle = \sum_{x\in K} \frac{|x\rangle}{\sqrt{(2d-1)^N}}|\phi\rangle. \tag{56}$$

Here, the notation $G|0\rangle$ means the general transform of $|0\rangle$ and the notation $F|d-1\rangle$ means the Fourier transform of $|d-1\rangle$.

We introduce $O_{f(x)}$ gate

$$O_{f(x)}|x\rangle|j\rangle = |x\rangle|(f(x)+j) \bmod d\rangle, \tag{57}$$

where

$$f(x) = g(a)\cdot x \bmod d = g(a) \odot x. \tag{58}$$

We have the following formula by phase kick-back

$$O_{f(x)}|x\rangle|\phi\rangle = \omega^{f(x)}|x\rangle|\phi\rangle. \tag{59}$$

We have $|\psi_2\rangle$, by operating $O_{f(x)}$ to $|\psi_1\rangle$,

$$O_{f(x)}|\psi_1\rangle = |\psi_2\rangle = \sum_{x\in K} \frac{\omega^{f(x)}|x\rangle}{\sqrt{(2d-1)^N}}|\phi\rangle. \tag{60}$$

After the general transform of $|x\rangle$, using the previous equations (53) and (60), we can now evaluate $|\psi_3\rangle$ as follows:

$$|\psi_3\rangle = \sum_{z\in K}\sum_{x\in K} \frac{(\omega)^{x\odot z + g(a)\odot x}|z\rangle}{(2d-1)^N}|\phi\rangle. \tag{61}$$

Notice

$$\sum_{x\in K} (\omega)^{x\odot(z+g(a))} = (2d-1)^N \delta_{z,-g(a)}. \tag{62}$$

Therefore, the above summation is zero if $z \neq -g(a)$ and the above summation is $(2d-1)^N$ if $z = -g(a)$. Thus we have

$$\begin{aligned} |\psi_3\rangle &= \sum_{z\in K} \frac{(2d-1)^N \delta_{z,-g(a)}|z\rangle}{(2d-1)^N}|\phi\rangle \\ &= |-(g(a_1),g(a_2),g(a_3),\ldots,g(a_N))\rangle|\phi\rangle \end{aligned} \tag{63}$$

from which

$$|-(g(a_1),g(a_2),g(a_3),\ldots,g(a_N))\rangle \tag{64}$$

can be obtained. That is to say, if we measure the first N quantum states in a $(2d-1)$-dimensional space of the state $|\psi_3\rangle$, that is, $|-(g(a_1), g(a_2), g(a_3), \ldots, g(a_N))\rangle$, then we can retrieve the following values (as an integer string)

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N) \tag{65}$$

using a single query.

V. EXTENSION TO A COMPLEX NUMBER STRING

In this section, we propose a generalization of the Bernstein-Vazirani algorithm to find out a complex number string, in which the general continuous non-unitary transform is utilized instead of the discrete non-unitary transform.

Suppose that the following sequence of complex numbers is given

$$a_1, a_2, a_3, \ldots, a_N. \tag{66}$$

A special function is provided as follows:

$$g : \mathbf{C} \to \mathbf{C}. \tag{67}$$

Our final goal is for determining the following values as a complex number string

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_N). \tag{68}$$

It is worth pointing out that in the best classical case, to determine $g(a_1), \ldots, g(a_N)$, N queries are expected to be answered, which are N separate evaluations of the function (67). However, in case of the generalized Bernstein-Vazirani algorithm, we shall require a single query instead of N queries.

We determine the real part. Assume the following sequence of complex numbers as follows:

$$a_1, a_2, a_3, \ldots, a_N. \tag{69}$$

Take a special function l as follows:

$$l : \mathbf{C} \to (-d, +d). \tag{70}$$

Our aim is for determining the following values as the real part of the complex number string

$$l(a_1), l(a_2), l(a_3), \ldots, l(a_N). \tag{71}$$

Recall that in the best classical case, to determine $l(a_1), \ldots, l(a_N)$ as the coefficients of the linear function $f(x) = l(a_1)x_1 + \cdots + l(a_N)x_N$, we need N queries for values of $f(x)$ in which N is separate evaluations of the function (70). Using the generalized Bernstein-Vazirani algorithm, we require just a single query.

Considering a positive integer as d, the problem is discussed in modulo d:

$$-d < \overbrace{l(a_1), l(a_2), l(a_3), \ldots, l(a_N)}^{N} < d, \tag{72}$$

where $l(a_j) \in (-d, +d)$, and $l(a)$ is as follows:

$$l(a) = (l(a_1), l(a_2), l(a_3), \ldots, l(a_N)), \tag{73}$$

where each entry of $l(a)$ is a real number, and $l(a) \in (-d, +d)^N$. We define $f(x)$ as follows:

$$\begin{aligned} f(x) &= (l(a) \cdot x) \bmod d \equiv l(a) \odot x \\ &= \{l(a_1)x_1 + l(a_2)x_2 + \cdots + l(a_N)x_N\} \bmod d, \end{aligned} \tag{74}$$

where $x = (x_1, ..., x_N) \in (-d, +d)^N$. Let us follow the quantum states through the algorithm.

The input state is

$$|\psi_0\rangle = |0\rangle^{\otimes N}|\phi\rangle, \tag{75}$$

where $|0\rangle^{\otimes N}$ means $\overbrace{|0, 0, ..., 0\rangle}^{N}$. Here $N < \infty$ that is N is a finite natural number. This is defined as follows:

$$|\phi\rangle = \sum_{j \in [0, +d)} \frac{\omega^{d-j}|j\rangle}{\sqrt{d}} \tag{76}$$

where $\omega = e^{2\pi i/d}$.

The general transform of $|0\rangle$ is as follows:

$$|0\rangle \to \sum_{y \in (-d, +d)} \frac{|y\rangle}{\sqrt{2d}}. \tag{77}$$

A calculation shows that this is not a unitary operation. Indeed, it does not have a full rank.

The general transform of $|x_1 ... x_N\rangle$ is as follows:

$$|x_1 ... x_N\rangle \to \sum_{z \in K} \frac{\omega^{z \odot x}|z\rangle}{\sqrt{(2d)^N}}, \tag{78}$$

where $K = (-d, d)^N$ and z is $(z_1, z_2, ..., z_N)$. $z_j \in (-d, +d)$. This is not a unitary operation. Here, for completeness, $\sum_{z \in K}$

is written in shorthand to a compound sum as:

$$\sum_{z_1\in(-d,+d)} \cdots \sum_{z_N\in(-d,+d)} . \tag{79}$$

After the componentwise general transforms of the first N quantum states in an infinite-dimensional space in (75)

$$\overbrace{G|0\rangle \otimes G|0\rangle \otimes \ldots \otimes G|0\rangle}^{N} \otimes |\phi\rangle, \tag{80}$$

we have, for quantum gates and their theory,

$$|\psi_1\rangle = \sum_{x\in K} \frac{|x\rangle}{\sqrt{(2d)^N}} |\phi\rangle. \tag{81}$$

Here, the notation $G|0\rangle$ means the general transform of $|0\rangle$. It is a straightforward generalization of the Fourier transform.

We introduce $O_{f(x)}$ gate

$$O_{f(x)}|x\rangle|j\rangle = |x\rangle|(f(x)+j) \bmod d\rangle, \tag{82}$$

where

$$f(x) = (l(a)\cdot x) \bmod d = l(a)\odot x. \tag{83}$$

We have the following formula by phase kick-back

$$O_{f(x)}|x\rangle|\phi\rangle = \omega^{f(x)}|x\rangle|\phi\rangle. \tag{84}$$

In what follows, we discuss the rationale behind of the above relation (84). Now consider the action of the $O_{f(x)}$ gate to the state $|x\rangle|\phi\rangle$. Each term in $|\phi\rangle$ is of the form $\omega^{d-j}|j\rangle$. We see

$$\begin{aligned} &O_{f(x)}\omega^{d-j}|x\rangle|j\rangle \\ &= \omega^{d-j}|x\rangle|(j+f(x)) \bmod d\rangle. \end{aligned} \tag{85}$$

k is introduced such as $f(x)+j=k$ then $d-j=d+f(x)-k$. Consequently, (85) becomes

$$\begin{aligned} &O_{f(x)}\omega^{d-j}|x\rangle|j\rangle \\ &= \omega^{f(x)}\omega^{d-k}|x\rangle|k \bmod d\rangle. \end{aligned} \tag{86}$$

Now, when $k<d$, we have $|k \bmod d\rangle = |k\rangle$ and thus, the terms in $|\phi\rangle$ such that $k<d$ are transformed as follows:

$$O_{f(x)}\omega^{d-j}|x\rangle|j\rangle = \omega^{f(x)}\omega^{d-k}|x\rangle|k\rangle. \tag{87}$$

Furthermore, as $f(x)$ and j are both bounded above by d, k is strictly less than $2d$. Thus, when $d\le k<2d$, we have $|k \bmod d\rangle = |k-d\rangle$. Now, introducing m as $k-d=m$, we have

$$\begin{aligned} &\omega^{f(x)}\omega^{d-k}|x\rangle|k \bmod d\rangle = \omega^{f(x)}\omega^{-m}|x\rangle|m\rangle \\ &= \omega^{f(x)}\omega^{d-m}|x\rangle|m\rangle. \end{aligned} \tag{88}$$

Hence, the terms in $|\phi\rangle$ such that $k\ge d$ are transformed as follows:

$$O_{f(x)}\omega^{d-j}|x\rangle|j\rangle = \omega^{f(x)}\omega^{d-m}|x\rangle|m\rangle. \tag{89}$$

Subsequently, regarding (87) and (89), we have

$$O_{f(x)}|x\rangle|\phi\rangle = \omega^{f(x)}|x\rangle|\phi\rangle. \tag{90}$$

Therefore, the relation (84) holds. Operating $O_{f(x)}$ to $|\psi_1\rangle$, $|\psi_2\rangle$ is transformed as:

$$O_{f(x)}|\psi_1\rangle = |\psi_2\rangle = \sum_{x\in K} \frac{\omega^{f(x)}|x\rangle}{\sqrt{(2d)^N}}|\phi\rangle. \tag{91}$$

After the general transform of $|x\rangle$, using the previous equations (78) and (91), $|\psi_3\rangle$ can be evaluated as follows:

$$|\psi_3\rangle = \sum_{z\in K}\sum_{x\in K} \frac{(\omega)^{x\odot z + l(a)\odot x}|z\rangle}{(2d)^N}|\phi\rangle. \tag{92}$$

Notice

$$\sum_{x\in K}(\omega)^{x\odot(z+l(a))} = (2d)^N \delta_{z,-l(a)}. \tag{93}$$

Therefore, the above sum is zero if $z\ne -l(a)$ and the above sum is $(2d)^N$ if $z=-l(a)$. So, we have

$$\begin{aligned} |\psi_3\rangle &= \sum_{z\in K}\frac{(2d)^N\delta_{z,-l(a)}|z\rangle}{(2d)^N}|\phi\rangle \\ &= |-(l(a_1),l(a_2),l(a_3),\ldots,l(a_N))\rangle|\phi\rangle \end{aligned} \tag{94}$$

from which

$$|-(l(a_1),l(a_2),l(a_3),\ldots,l(a_N))\rangle \tag{95}$$

can be obtained. That is to say, if we measure the first N quantum states in an infinite-dimensional space of the state $|\psi_3\rangle$, that is, $|-(l(a_1),l(a_2),l(a_3),\ldots,l(a_N))\rangle$, then we can retrieve the following values as the real part of a complex number strings

$$l(a_1),l(a_2),l(a_3),\ldots,l(a_N) \tag{96}$$

using a single query.

Next, we determine the imaginary part. We suppose that the following sequence of complex numbers is given

$$a_1, a_2, a_3, \ldots, a_N. \tag{97}$$

We introduce a special function

$$h : \mathbf{C} \to (-d, +d). \tag{98}$$

Our aim is to determine the following values as the imaginary part of a complex number string,

$$h(a_1), h(a_2), h(a_3), \ldots, h(a_N). \tag{99}$$

Recall that in the best classical case, to determine $h(a_1), \ldots, h(a_N)$, we need N queries, that is, N separate evaluations of the function (98). In case of the generalized Bernstein-Vazirani algorithm, we shall require a single query.

Finally, the two evaluations (the real part and the imaginary part) can be performed, in parallel computing method, simultaneously. In the final step, we measure the following quantum state:

$$\begin{aligned} &|-(l(a_1), l(a_2), l(a_3), \ldots, l(a_N))\rangle \otimes \\ &|-(h(a_1), h(a_2), h(a_3), \ldots, h(a_N))\rangle. \end{aligned} \tag{100}$$

That is, we determine the N complex values $g(a_j) = l(a_j) + ih(a_j)$ simultaneously. The speed of determining the string of complex numbers is shown to outperform the best classical case by a factor of N.

VI. EXTENSION TO A MATRIX STRING

We propose a method for calculating many different matrices $A, B, C, \ldots$ into $g(A), g(B), g(C), \ldots$ simultaneously. That is, the generalized Bernstein-Vazirani algorithm determines a matrix string instead of a number string. The speed of solving the problem is shown to outperform the classical case by a factor of the number of the elements of them.

Let the jth matrix be a $N_j \times M_j$ matrix (N_j rows, M_j columns, and $j = 1, 2, 3, \ldots$). To simplify, let us consider the case of two matrices ($j = 2$). Given the elements of the matrices $a_1, a_2, a_3, \ldots, a_{N_1 \times M_1}, b_1, b_2, b_3, \ldots, b_{N_2 \times M_2}$, and a special function g newly, we calculate $O(N_1 \times M_1 + N_2 \times M_2)$ values of the function $g(a_1), g(a_2), g(a_3), \ldots, g(a_{N_1 \times M_1})$, $g(b_1), g(b_2), g(b_3), \ldots, g(b_{N_2 \times M_2})$ simultaneously. In the classical case, we have to evaluate each $O(N_1 \times M_1 + N_2 \times M_2)$ values.

However, in the quantum case, we need just a query. Thus, the speed of calculating the two matrices is shown to outperform the classical case by a factor of $O(N_1 \times M_1 + N_2 \times M_2)$. The generalization of it, to many matrices, can be done in a straightforward manner. The most significant result of the section is that it can be better than its classical counterpart regarding time complexity. The speed of calculating the different matrices is shown to outperform the classical case by a factor of $O(N_1 \times M_1 + N_2 \times M_2 + N_3 \times M_3 + \ldots)$.

Let us consider the case of a single $N \times M$ matrix A (N rows and M columns). Given the elements of the matrix $a_1, a_2, a_3, \ldots, a_{N \times M}$, and a special function g newly, we can obtain the following values (as a complex number string)

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_{N \times M}) \tag{101}$$

using a single query. This is the most significant point of our method and it is possible by the generalized Bernstein-Vazirani algorithm. We consider them as the elements of the matrix $g(A)$;

$$g(A) = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}, \tag{102}$$

where $(g(a_1), g(a_2), g(a_3), \ldots, g(a_{N \times M})) = (a_{11}, a_{21}, \ldots, a_{nm})$. Therefore, we obtain $g(A)$ by a single query. In the classical case, we obtain $g(A)$ by $O(N \times M)$ queries. Thus, the speed of calculating the matrix $g(A)$ is shown to outperform the classical case by a factor of $O(N \times M)$.

Let us consider the case of two matrices A and B (N_j rows, M_j columns, and $j = 1, 2$). Given the elements of the matrices $a_1, a_2, a_3, \ldots, a_{N_1 \times M_1}, b_1, b_2, b_3, \ldots, b_{N_2 \times M_2}$, and a special function g newly, we can obtain the following values (as a complex number string)

$$g(a_1), g(a_2), g(a_3), \ldots, g(a_{N_1 \times M_1}),$$
$$g(b_1), g(b_2), g(b_3), \ldots, g(b_{N_2 \times M_2}), \tag{103}$$

using a single query. Again it is possible by the generalized Bernstein-Vazirani algorithm. We consider them as the elements of the matrices $g(A)$ and $g(B)$;

$$g(A) = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1m} \\ a_{21} & a_{22} & \ldots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \ldots & a_{nm} \end{pmatrix}, \tag{104}$$

and

$$g(B) = \begin{pmatrix} b_{11} & b_{12} & \ldots & b_{1m} \\ b_{21} & b_{22} & \ldots & b_{2m} \\ \vdots & \vdots & & \vdots \\ b_{n1} & b_{n2} & \ldots & b_{nm} \end{pmatrix}, \tag{105}$$

where $(g(a_1), g(a_2), g(a_3), \ldots, g(a_{N_1 \times M_1})) = (a_{11}, a_{21}, ..., a_{nm})$. $(g(b_1), g(b_2), g(b_3), \ldots, g(b_{N_2 \times M_2})) = (b_{11}, b_{21}, ..., b_{nm})$. Therefore, we obtain $g(A)$ and $g(B)$ by a single query. In the classical case, we obtain $g(A)$ and $g(B)$ by $O(N_1 \times M_1 + N_2 \times M_2)$ queries. Thus, the speed of calculating the matrix $g(A)$ and $g(B)$ is shown to outperform the classical case by a factor of $O(N_1 \times M_1 + N_2 \times M_2)$.

The generalization of it, to many matrices, can be done in a straightforward manner. The speed of calculating the matrices is shown to outperform the classical case by a factor of $O(N_1 \times M_1 + N_2 \times M_2 + N_3 \times M_3 + ...)$.

In what follows, we give an example in the case of three matrices. Let $g(x)$ be x^2. Let A, B, C be

$$A = \begin{pmatrix} 0 & 2 \\ 1 & 3 \end{pmatrix}, B = \begin{pmatrix} 4 & 6 \\ 5 & 7 \end{pmatrix}, C = \begin{pmatrix} 8 & 10 \\ 9 & 11 \end{pmatrix}. \tag{106}$$

The elements are

$$\overbrace{0, 1, 2, 3}^{A}, \overbrace{4, 5, 6, 7}^{B}, \overbrace{8, 9, 10, 11}^{C}. \tag{107}$$

In the classical case, we have to evaluate the following values separately

$$g(0), g(1), g(2), g(3), g(4), g(5),$$
$$g(6), g(7), g(8), g(9), g(10), g(11). \tag{108}$$

In the quantum case, from the generalized Bernstein-Vazirani algorithm, using g, we have

$$\overbrace{0, 1, 4, 9}^{g(A)}, \overbrace{16, 25, 36, 49}^{g(B)}, \overbrace{64, 81, 100, 121}^{g(C)} \tag{109}$$

using a single query. Hence we have

$$g(A) = \begin{pmatrix} 0 & 4 \\ 1 & 9 \end{pmatrix}, g(B) = \begin{pmatrix} 16 & 36 \\ 25 & 49 \end{pmatrix},$$
$$g(C) = \begin{pmatrix} 64 & 100 \\ 81 & 121 \end{pmatrix}, \tag{110}$$

using a single query. The speed of calculating the matrices is shown to outperform the classical case by a factor of 12; $(2 \times 2 + 2 \times 2 + 2 \times 2 = 12)$.

Now, we generalize the Bernstein-Vazirani algorithm from a bit string into a matrix string. Let us suppose that the following sequence of matrices is given

$$A_1, A_2, A_3, \ldots, A_N. \tag{111}$$

Let us now introduce a special function newly

$$g : \mathbf{C} \to \mathbf{C}. \tag{112}$$

Our goal is for determining the following matrices (as a matrix string)

$$g(A_1), g(A_2), g(A_3), \ldots, g(A_N). \tag{113}$$

In the generalized Bernstein-Vazirani algorithm, we shall require a single query. We hope our discussions will give a way to the quantum simulation problem.

VII. CONCLUSIONS

In conclusion, we have presented various new forms of the Bernstein-Vazirani algorithm beyond qubit systems. First, we have reviewed the Bernstein-Vazirani algorithm for determining a bit string. The algorithm has had the feature of the Hadamard transform of a quantum state in a two-dimensional space to solve the problem of finding a bit string.

Second, we have discussed the generalized Bernstein-Vazirani algorithm for determining a natural number string.

The generalized algorithm presented here has had the feature of the normal Fourier transform of a quantum state in a finite-dimensional space to solve the problem of finding a string of natural numbers.

Thirdly, we have discussed the generalized Bernstein-Vazirani algorithm for determining an integer string. The generalized algorithm presented here has had the feature of a general discrete non-unitary transform of a quantum state in a finite-dimensional space to solve the problem of finding a string of integers.

Finally, we have discussed the generalized Bernstein-Vazirani algorithm for determining a complex number string. The generalized algorithm presented here has had the feature of a general continuous non-unitary transform of a quantum state in an infinite-dimensional space to solve the problem of finding a string of complex numbers.

All of the generalized algorithms presented here have had the following structure. Given the set of complex numbers $\{a_1, a_2, a_3, \ldots, a_N\}$ and a special function g, we have determined N values of the function $g(a_1), g(a_2), g(a_3), \ldots, g(a_N)$ simultaneously. The speed of determining the strings has been shown to outperform the best classical case by a factor of N in every cases.

Additionally, we have proposed a method for calculating many different matrices $A, B, C, \ldots$ into $g(A), g(B), g(C), \ldots$ simultaneously. The speed of solving the problem has been shown to outperform the classical case by a factor of the number of the elements of them.

We are working a lot for the Bernstein-Vazirani algorithm. We also suggest we should discuss about implementation. An optical implementation of Bernstein-Vazirani string finding algorithm is the interested [21].

NOTE

On behalf of all authors, the corresponding author states that there is no conflict of interest.

ACKNOWLEDGMENTS

We thank Prof. Germano Resconi for valuable comments.

[1] E. Bernstein and U. Vazirani, Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing (STOC '93), pp. 11-20 (1993).

[2] E. Bernstein and U. Vazirani, SIAM J. Comput. 26-5, pp. 1411-1473 (1997).

[3] J. Du, M. Shi, X. Zhou, Y. Fan, B. J. Ye, R. Han, and J. Wu, Phys. Rev. A **64**, 042306 (2001).

[4] E. Brainis, L.-P. Lamoureux, N. J. Cerf, Ph. Emplit, M. Haelterman, and S. Massar, Phys. Rev. Lett. **90**, 157902 (2003).

[5] A. W. Cross, G. Smith, and J. A. Smolin, Phys. Rev. A **92**, 012327 (2015).

[6] H. Li and L. Yang, Quantum Inf. Process. **14**, 1787 (2015).

[7] K. Nagata and T. Nakamura, Open Access Library Journal, 2: e1798 (2015).

[8] K. Nagata, T. Nakamura, and A. Farouk, Int. J. Theor. Phys. **56**, 2887 (2017).

[9] S. D. Fallek, C. D. Herold, B. J. McMahon, K. M. Maller, K. R. Brown, and J. M. Amini. New J. Phys. **18**, 083030 (2016).

[10] R. Krishna, V. Makwana, and A. P. Suresh, arXiv:1609.03185 [quant-ph] (2016).

[11] D. P. Mehendale and P. S. Joag, Quantum Physics Letters, Vol. 6, No. 1, 73 (2017).

[12] K. Fujikawa, C. H. Oh, and K. Umetsu, Modern Physics Letters A, Vol. 34, No. 07n08, 1950146 (2019).

[13] A. Yin, K. He, and P. Fan, Modern Physics Letters A, (2019). https://doi.org/10.1142/S0217732319501694

[14] K. Nagata, G. Resconi, T. Nakamura, J. Batle, S. Abdalla, and A. Farouk, MOJ Ecology and Environmental Science, Volume 2, Issue 1, 00010, (2017).

[15] K. Nagata, T. Nakamura, H. Geurdes, J. Batle, S. Abdalla, and A. Farouk, Int. J. Theor. Phys. **57**, 1605 (2018).

[16] K. Nagata, T. Nakamura, H. Geurdes, J. Batle, A. Farouk, D. N. Diep, and S. K. Patro, Int. J. Theor. Phys. **57**, 2546 (2018).

[17] K. Nagata, H. Geurdes, S. K. Patro, S. Heidari, A. Farouk, and T. Nakamura, Quantum Stud.: Math. Found. (2019). https://doi.org/10.1007/s40509-019-00196-4

[18] K. Nagata, T. Nakamura, H. Geurdes, J. Batle, S. Abdalla, A. Farouk, and D. N. Diep, Int. J. Theor. Phys. **57**, 973 (2018).

[19] K. Nagata, T. Nakamura, S. Heidari, A. Farouk, and D. N. Diep, viXra:1809.0553 (2018). http://vixra.org/abs/1809.0553

[20] K. Nagata and T. Nakamura, Int. J. Theor. Phys. **58**, 247 (2019).

[21] D. N. Diep, K. Nagata, T. Nakamura, and L. T. Thanh, An Optical Implementation of Bernstein-Vazirani String Finding Algorithm, (private communications) (2019).