



Circuit Design for k -Coloring Problem and Its Implementation in Any Dimensional Quantum System

Amit Saha¹ · Debasri Saha¹ · Amlan Chakrabarti¹

Received: 29 May 2021 / Accepted: 6 August 2021 / Published online: 25 August 2021
© The Author(s), under exclusive licence to Springer Nature Singapore Pte Ltd 2021

Abstract

With the evolution of quantum computing, researchers nowadays tend to incline to find solutions to NP-complete problems using quantum algorithms to gain asymptotic advantage. In this paper, we solve k -coloring problem (NP-complete problem) using Grover's algorithm in any dimensional quantum system or any d -ary quantum system for the first time to the best of our knowledge, where $d \geq 2$. Till date, k -coloring problem has been implemented only in binary and ternary quantum systems, hence, we abide to $d = 2$ or $d = 3$, that is for binary and ternary quantum systems for comparing our proposed work with the state-of-the-art techniques. Our comparator-based approach has reduced the qubit cost, compared to the state-of-the-art binary quantum systems (Saha et al. in IEEE Int Symp Smart Electron Syst 1:17–22, 2020). Further in this paper, with the help of newly proposed ternary comparator, a substantial reduction in quantum gate count for the ternary oracle circuit of the k -coloring problem than the previous approaches has been obtained. Later, this proposed comparator-based approach helps to generalize the implementation of the k -coloring problem in any dimensional quantum system. An end-to-end automated framework has been put forward for implementing the k -coloring problem for any undirected and unweighted graph on any available near-term quantum devices or Noisy Intermediate-Scale Quantum (NISQ) devices or multi-valued quantum simulator, which helps in generalizing our approach.

Keywords k -Coloring problem · Grover's algorithm · NISQ · Multi-valued quantum circuit synthesis

Introduction

Modern day researchers have shown a startling interest for implementing quantum algorithms [2–4], which give a potential speedup over many of their classical counterparts as the advancement of quantum computer has achieved a phenomenal success in recent years [5]. There is an immense urge for the implementation of NP-complete problems on quantum computers with the thriving quantum wave [6]. According to the seminal work on computational complexity by Karp [7], if a solution to any of the NP-complete problems can be obtained then any other NP-complete

problem can be polynomially reducible to that problem. In this paper, we have focused on one of the well-known NP-complete problems i.e., k -coloring problem. Our main focus is to provide an end-to-end framework that automatically implements an NP-complete problem i.e., k -coloring problem in any dimensional quantum system, so that if anyone can map their computational problem to the k -coloring problem in polynomial time, will be able to implement further, without prior knowledge of gate-based quantum circuit implementation.

An automatic circuit synthesis for k -coloring problem with the help of Grover's algorithm [8] is presented in the context of multi-valued quantum system in this paper. Although physical systems in classical hardware are typically binary, but, common quantum hardware, such as in superconducting [9] and trapped ion computers [10], has an infinite spectrum of discrete energy levels. Quantum hardware may be configured to manipulate the d energy levels by operating on d -ary quantum system. Qudit gates have already been successfully implemented [11–17] indicating it is possible to consider higher-level systems apart from

This article is part of the topical collection “Hardware for AI, Machine Learning and Emerging Electronic Systems” guest edited by Himanshu Thapliyal, Saraju Mohanty and VS Kanchana Bhaaskaran.

✉ Amit Saha
abamitsaha@gmail.com

¹ A. K. Choudhury School of Information Technology,
University of Calcutta, Calcutta 700 106, India

qubit only systems. Thus, the question of higher states, beyond the standard two being implemented and performed, no longer stands strong [18]. Therefore, we graduate to multi-valued quantum systems or qudits [19], which in the course reduce the circuit complexity and commend the efficiency of quantum algorithms [20] to provide larger state space with simultaneous multiple control operations [21–23]. For instance, N qubits can be formulated as $\frac{N}{\log_2 d}$ qudits, which immediately gives $\log_2 d$ -factor in run-time [24–26] for solving any computational problem using quantum algorithm.

k -coloring problem is an NP-complete problem that assigns colors to every nodes or vertices of a given graph with the available k colors in such a way that every adjacent vertices connected by an edge have distinct colors. Suppose n is the number of nodes of a given graph, k is the number of colors, then to find an appropriate solution using a classical algorithm requires $O(d^{n \cdot \log_d k})$ number of steps in d -dimensional system [27, 28]. On the other hand, by using oracle and the diffusion operator of multi-valued Grover's algorithm [29, 30], finding the exact solution needs $O\sqrt{N}$ number of iterations where N is $d^{n \cdot \log_d k}$. Many researchers have already addressed the graph coloring problem for binary and ternary quantum systems. Earlier in [27, 31], graph coloring problem with the help of Grover's algorithm is talked about with respect to the binary quantum systems. Again, in [32], SAT reduction technique, the state-of-the-art approach, is used for solving 3-coloring problem and it gives an end-to-end framework for the implementation of it in the IBMQ quantum processor [33]. But, SAT reduction technique generates an immense qubit cost, resulting in inefficient circuit cost. Previously in [34] and [28], circuit synthesis for graph coloring problem using Grover's algorithm has been presented with respect to ternary quantum systems with the help of ternary comparators [35]. Albeit, in these works, the gate cost remains colossal. A space-efficient solution of graph coloring problem [36, 37] using QAOA (Quantum Approximate Optimization Algorithm) is also proposed in the literature, but we stick to Grover's algorithm in this work.

In [1], we proposed a comparator-based approach for implementing the k -coloring problem in the binary quantum structure using Grover's search, which has less qubit cost as compared to the state-of-the-art approach [32] i.e., based on Grover's algorithm. In this paper, we have generalized the comparator for d -ary quantum systems, which helps to overcome the engineering challenge of the implementation of k -coloring problem in d -ary quantum systems. We have proposed an automated end-to-end framework for any dimensional quantum system to implement k -coloring problem using a newly proposed generalized comparator to map the high-level description of the proposed circuit to any

hardware-level quantum operations with an abstraction with better quantum cost in terms of quantum gate cost relative to the state-of-the-art works. In addition to this, we have claimed the following, for further establishing the novelty of the propose research work:

- We propose an automated end-to-end framework for k -coloring problem using quantum search algorithm in any dimensional quantum system for the first time, to the best of our knowledge.
- The design of the proposed framework is such that, the quantum solution of k -coloring problem can be mapped into any available near-term quantum devices or multi-valued quantum technology, which makes our approach generalized in nature.
- We show that our newly proposed comparator helps to implement k -coloring problem with reduced quantum cost with respect to quantum gates as compared to the state-of-the-art approaches in ternary quantum domain.
- A generalized comparator for d -ary system is elaborated in this paper, which is a first of its kind approach.

The paper is structured as follows. In “[Background](#)”, the brief description of quantum circuits and graph coloring using Grover's algorithm are described. The proposed methodology is vividly explained in “[General Flow of Proposed Automated Framework for Mapping k-coloring Problem to Quantum Computers](#)”. In “[Proposed Methodology of Circuit Design for k-coloring Problem in d-dimensional Quantum Systems](#)”, implementation of k -coloring problem has been shown. Experimental results are captured in “[Experimental Results](#)” followed by concluding remarks.

Background

In this section, a few preliminaries of the quantum circuit and Grover's algorithm are presented. We have also explicitly described the existing comparator-based k -coloring problem using Grover's algorithm in binary quantum systems.

Quantum Circuit

Any quantum algorithm can be expressed or visualized in the form of a quantum circuit. Commonly for binary quantum systems, logical qubits and quantum gates comprise these quantum circuits [38]. The number of gates present in a circuit is called gate count and the number of qubits present in a circuit is known as qubit cost. In this work, we mainly deal with qudits and generalized quantum gates.

Qudits

Logical qudit that encodes input/output of a quantum algorithm in multi-valued quantum systems is often referred to as data qudit. Ancilla qudit are another type of qudit used to store temporary results. In d -dimensional quantum systems *qudit* is the unit of quantum information. Qudit states can be manifested as a vector in the d dimensional Hilbert space \mathcal{H}_d . The span of orthonormal basis vectors $\{|0\rangle, |1\rangle, |2\rangle, \dots, |d-1\rangle\}$ is the vector space. In qudit system, the general form of quantum state can be expressed as

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle + \alpha_2|2\rangle + \dots + \alpha_{d-1}|d-1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{d-1} \end{pmatrix} \tag{1}$$

where $|\alpha_0|^2 + |\alpha_1|^2 + |\alpha_2|^2 + \dots + |\alpha_{d-1}|^2 = 1$ and $\alpha_0, \alpha_1, \dots, \alpha_{d-1} \in \mathbb{C}^d$.

Generalized Quantum Gates

In this subsection, an outline of generalized qudit gates [39, 40] is conferred. The generalisation can be delineated as discrete quantum states of any arity. In a quantum algorithm, for modification of the quantum state, unitary qudit gates are applied. For logic synthesis of Grover’s algorithm in d -dimensional quantum systems, it is necessary to take into account one-qudit generalized gates namely NOT gate (X_d), phase-shift gate (Z_d), Hadamard gate (F_d), two-qudit generalized CNOT gate ($C_{X,d}$) and Generalized multi-controlled Toffoli gate ($C_{X,d}^n$). These gates are expressed in detail for a better understanding in the Appendix 1.

Generalized Grover’s Algorithm in d -dimensional quantum systems

The generalized Grover’s algorithm in d -dimensional quantum systems is presented here. There exists two sub-parts of the algorithm namely oracle and diffusion [30]. As per convention, Grover’s algorithm for searching in an unstructured database can be defined as follows: given a collection

of unstructured database elements $x = 1, 2, \dots, N$, and an *oracle function* $f(x)$ that acts on a marked element s as follows [29],

$$f(x) = \begin{cases} 1, & x = s, \\ 0, & x \neq s, \end{cases} \tag{2}$$

perceive the marked element with as few calls to $f(x)$ as possible [29, 30]. The database is encoded into a superposition of quantum states where each element is assigned to a corresponding basis state. Grover’s algorithm searches over every possible outcome, which is put forward as a basis vector $|x\rangle$ in an n -dimensional Hilbert space in d -dimensional quantum systems. Likewise, the marked element is encoded as $|s\rangle$. Thus, the search can be done in parallel, after application of unitary operations as an oracle function to the *superposition* of the different possible outcomes. The generalized diffusion operator, also known as inversion about the average operator, amplifies the amplitude of the marked state to increase its measurement probability using constructive interference, with simultaneous enfeeblement of all other amplitudes, and searches the marked element in $O(\sqrt{N})$ steps, where $N = d^n$ [29].

The circuit diagram for the generalized Grover’s algorithm in a d -dimensional quantum systems is presented in Fig. 1, where at least $n + 1$ qudits are required. More elaborately, the steps of the Grover’s algorithm are as follows:

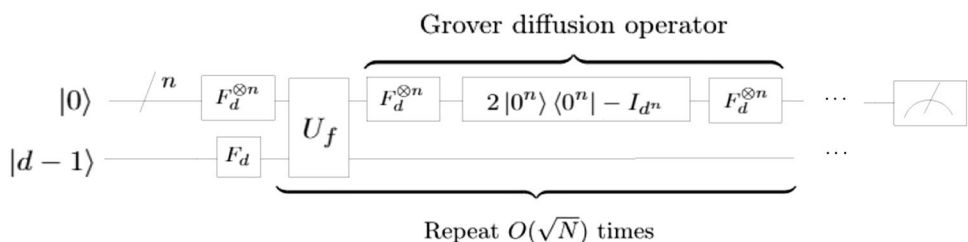
Initialization: The algorithm starts with the uniform superposition of all the basis states on the n input qudits in $|0\rangle$ by assimilating generalized Hadamard or quantum DFT gate. The last ancilla qudit is used as an output qudit, which is initialized to $F_d|d-1\rangle$. Thus, we obtain the d -dimensional quantum state $|a\rangle$:

$$|a\rangle = F_d^{\otimes n} |0_d\rangle = \frac{1}{\sqrt{d^n}} \sum_{x=1}^{d^n} |x\rangle$$

Oracle query: The oracle (U_f) of Grover’s search marks the marked state $|s\rangle$ while keeping all the other states unchanged, and can be expressed as:

$$|x\rangle \xrightarrow{U_f} (-1)^{f(x)} |x\rangle$$

Fig. 1 Generalized circuit for Grover’s algorithm in d -dimensional quantum systems



The oracle block U_f as shown in Fig. 1 is dependant on the problem instance. The oracle using Unitary transformation is needed to be designed as per requirement.

Diffusion: The diffusion operator of Grover’s search is generic, it doesn’t depend on specific problem . As shown in Fig. 1, the diffusion operator is initially assigned with generalized Hadamard ($F_d^{\otimes n}$), then by $2|0^n\rangle\langle 0^n| - I_{d^n}$ and generalized Hadamard ($F_d^{\otimes n}$) again. The diffusion operator (D) can be expressed as:

$$D = F_d^{\otimes n} [2|0^n\rangle\langle 0^n| - I_{d^n}] F_d^{\otimes n}$$

The matrix representation of generalized diffusion operator [29] for d -dimensional quantum systems is shown below:

$$D = \begin{pmatrix} \frac{2}{d^n} - 1 & \frac{2}{d^n} & \frac{2}{d^n} & \dots & \frac{2}{d^n} \\ \frac{2}{d^n} & \frac{2}{d^n} - 1 & \frac{2}{d^n} & \dots & \frac{2}{d^n} \\ \frac{2}{d^n} & \frac{2}{d^n} & \frac{2}{d^n} - 1 & \dots & \frac{2}{d^n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{2}{d^n} & \frac{2}{d^n} & \frac{2}{d^n} & \dots & \frac{2}{d^n} - 1 \end{pmatrix}$$

Figure 2 shows the generalized circuit for Grover’s diffusion operator in d -dimensional quantum systems. It can be constructed using generalized Hadamard gate, generalized NOT gate and generalized multi-controlled Toffoli gate.

We have already discussed that the diffusion operator is predefined for all problem instances, the oracle is specific to the given search problem. The combination of the oracle and the diffusion gives generalized Grover operator G ,

$$G = DU_f$$

We need to din the Grover’s operator $O(\sqrt{N})$ times to get the coefficient of the marked state $|s\rangle$ large enough so that it can be obtained from measurement with probability close to 1. As we have considered every gate to be ideal in our proposed circuit design, and therefore, the success rate of the Grover’s algorithm remains same as for any other circuit design of a specific problem. In this paper, we start the algorithm with the equal superposition of all the basis states, and hence the success probability is ~ 1 after \sqrt{N} steps and thus round off the Grover’s algorithm.

Circuit Design and Implementation of k -coloring Problem using Grover’s Algorithm in Qubit System

Graph coloring problem, as discussed in the introduction, Given a graph, we need to color it with the constraint that no two adjacent nodes will have the same color. In [32], a SAT reduction-based technique was introduced for solving 3-coloring problem as an example of k -coloring problem in binary quantum systems. A SAT problem is defined as a function of OR and AND logical operations on input variables so as to reach to a solution. In [32], if n is the number of vertices in a given graph and k is the given number of color to solve the k -coloring problem, then it is shown that k -coloring problem can be solved as a Boolean satisfiability problem using Grover’s algorithm. For that, all possible input variables of $n \cdot k$ are needed to be encoded as input data to implement Grover’s algorithm. Hence $n \cdot k$ data qubits are required to implement k -coloring problem by SAT reduction-based technique. In [1], we proposed a comparator-based approach for implementing the k -coloring problem in binary quantum structure using Grover’s search, which has less qubit cost i.e., $n \cdot \lceil \log_2 k \rceil$ as compared to [32]. All possible combination of k colors to color the nodes of the given graph can be implemented with $n \cdot \lceil \log_2 k \rceil$ input qudits. Instead of using OR and AND logical operation, our proposed comaparator helped us in achieving logarithmic reduction with respect to the input qubit cost. As we know in Grover’s algorithm, diffusion operator uses an MCT gate of the size of the number data qubits, thus as since we reduce the number of data qubits by our propose approach, hence the circuit depth will also be reduced [41].

Now, we have discussed about the proposed comparator in [1]. In Fig. 3, circuit synthesis for 2-qubit and 4-qubit comparator is shown, the functional description of which is given in Equation 3. The complete circuit synthesis for the binary comparator is designed using CNOT, NOT, Toffoli/ MCT gates.

$$\text{Binary_Comparator}(a, b, f) = \begin{cases} f = 0, & \text{if } a = b; \\ f = 1, & a \neq b. \end{cases} \quad (3)$$

Fig. 2 Generalized circuit for Grover’s diffusion operator in d -dimensional quantum systems [29]

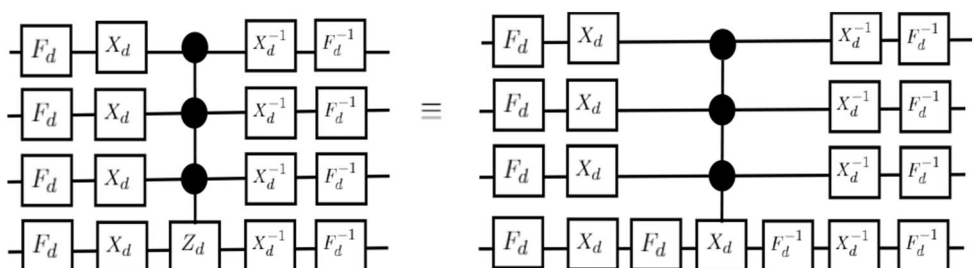


Fig. 3 Example binary comparator: **a** 2-qubit; **b** 4-qubit

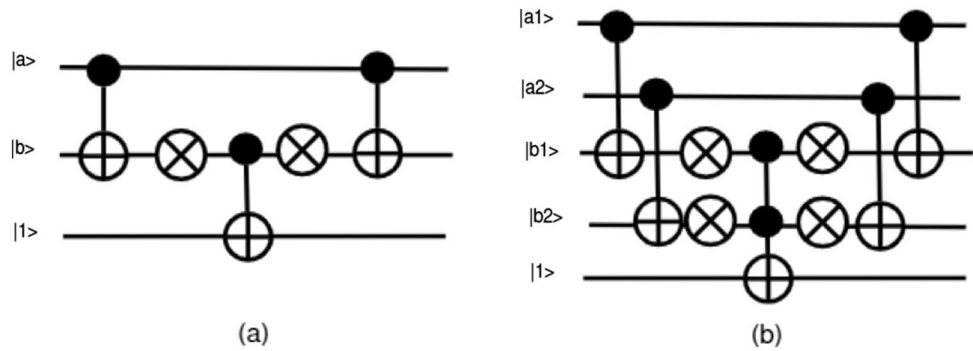


Table 1 Comparative analysis of binary oracle

Parameters	Hu et al. [32]	Proposed work
Data qubit cost	$n \cdot k$	$n \cdot \lceil \log_2 k \rceil$
Ancilla qubit cost	$O((n \cdot k)^2)$	$O(n)$
Processor	IBMQ	Any NISQ Device

Our proposed comparator-based oracle gives better result with respect to the data qubit and ancilla qubit as $n \cdot \lceil \log_2 k \rceil$ and $O(n)$ respectively, as compared to [32]. The comparative analysis is shown in Table 1.

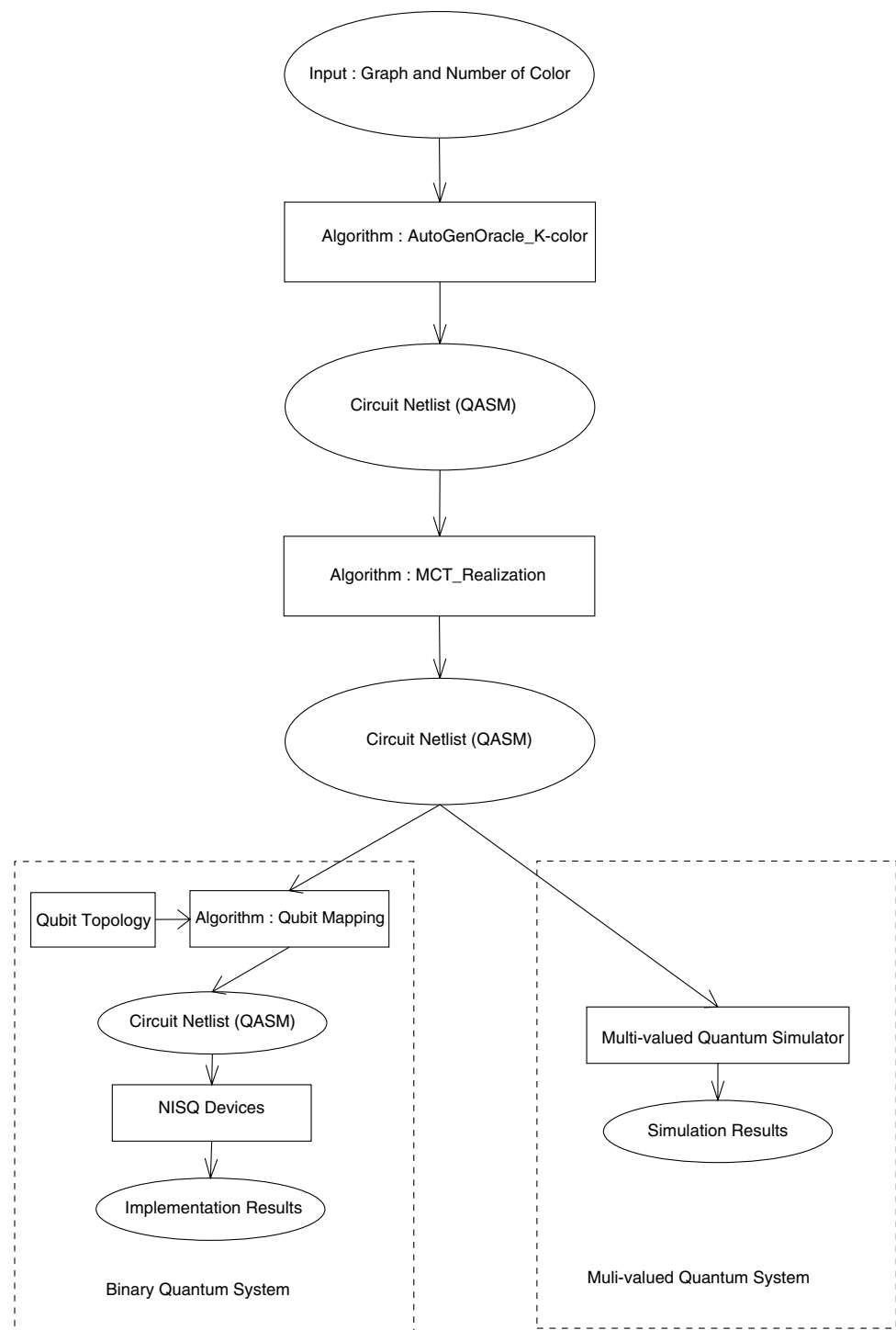
In [1], we further showed that the mapping of generated comparator-based oracular circuit using Grover’s search to NISQ devices through MCT realization and qubit mapping algorithm. For mapping the logical synthesized circuit to quantum hardware or NISQ devices [5], the logical quantum gates are needed to be realized to hardware-specific one and two-qubit gates for making it hardware compatible for implementation. The process of decomposition of MCT gate to NISQ compatible one-qubit and two-qubit gates is presented in [42]. At first, the MCT gate is to be decomposed to MCZ gate. After that, MCZ gate is realized to $MCR_x(\pi)$. Lastly, without using any ancilla qubit, $MCR_x(\pi)$ is reduced to one-qubit and two-qubit gates. As the quantum circuit is logical, there is no constraint of qubit connectivity. There is a specific qubit topology or coupling graph for NISQ devices. The interaction between two physical qubits is defined by coupling graph. This is different for different NISQ devices. Hence, obviously mapping the logical circuit to the physical one is challenging. The solution to this problem is to insert SWAP gates between the two qubits to comply to the hardware constraint without compromising on the logic of the quantum circuit. The concept behind a good qubit mapping problem is to minimize the number of SWAP insertion gates as well as the depth of the circuit. Li et. al. proposed SWAP-based Bidirectional heuristic search algorithm (SABRE) in [43], which deals with any arbitrary qubit topology for any NISQ devices. Three main features make SABRE noticeable. Firstly, an exhaustive search is not performed on the entire circuit, but a SWAP-based

heuristic search is performed keeping in mind the qubit dependency. Then, initial mapping is optimized with the use of a novel reverse traversal technique. Lastly, the decay effect being introduced to enable the trade-off between the depth and the number of gates of the entire algorithm. Tan et. al. formulated layout synthesis for quantum computing, which is a benchmark, as optimization problems [44]. They handed over two synthesizers: an exact layout synthesizer (OLSQ) and an approximate, transition-based synthesizer (TB-OLSQ). OLSQ is the first, that guarantees optimality and efficiency both in time and space for general quantum processors, as compared to previous exact approaches. This approach shows some promises of being beneficial for realistic applications for near-term quantum computers. The proposed circuit can easily be mapped to any arbitrary qubit topology by using one of these protocols. Apart from these protocols, we performed all our experiments on IBM quantum computer [33], specifically `ibmq_16_melbourne` (quantum volume is 8), which has a transpiler that automatically maps the decomposed logical circuit into physical IBM hardware. The above-discussed methodology of mapping k -coloring problem to binary quantum computer is further extended to any dimensional quantum system in this paper.

General Flow of Proposed Automated Framework for Mapping k -Coloring Problem to Quantum Computers

The flowchart in Fig. 4 depicts the complete flow of our proposed automated end-to-end framework. Mainly three algorithms: `AutoGenOracle_K-color`, `MCT_Realization` and `Qubit Mapping (binary)` are the basis of our framework. At first, `AutoGenOracle_K-color` algorithm, which is based on the proposed comparator, is implemented with inputs: graph information i.e, adjacency matrix of the given graph and the number of colors (k) to get the output, quantum circuit netlist in the form QASM. Now, `MCT_Realization` algorithm takes as input, generated circuit netlist and MCT gates to NISQ hardware compatible 1-qubit and 2-qubit gates [42] and multi-valued quantum

Fig. 4 General flow of proposed automated framework for mapping k -coloring problem to any dimensional quantum system



technology compatible 1-qudit and 2-qudit gates are realised. These `MCT_Realization` and qubit mapping algorithm are well-defined in the literature. We have just adopted them as a support in our framework.

Due to multi-valued quantum technology constraint, implementation of multi-valued circuit netlist on multi-valued quantum hardware is technologically infeasible till date, rather we have simulated and verified the multi-valued

circuit netlist on multi-valued simulator. As the proposed framework works for any dimensional quantum system, hence it must work for binary quantum systems as well. We have already discussed about that in the previous subsection, therefore, we stick to multi-valued quantum systems from here on.

Proposed Methodology of Circuit Design for k -Coloring Problem in d -dimensional Quantum Systems

This section outlines the proposed methodology for the circuit synthesis of the k -coloring problem using Grover’s algorithm. k -coloring problem takes an adjacency matrix of a graph and k number of colors as input and determines that the graph is properly colored or not. As mentioned in the previous section, there are two parts of the Grover’s search algorithm, where only the oracle is specific to the given search problem. Hence, this paper proposes the design of an oracle circuit that takes a graph and k number of colors as input. The oracle then marks those states that determine the properly colored nodes of the given graph before the diffusion operator is applied to them. Let’s start with the proposed oracle for k -coloring problem.

Proposed Oracle for k -Coloring Problem in d -Dimensional Quantum Systems

The proposed oracle of the k -coloring problem in any dimensional quantum system as an application of the Grover’s search algorithm is sketched in this subsection. Figure 5 shows the quantum circuit block of oracle for the k -coloring problem in d -dimensional quantum systems. The construction of oracle for k -coloring problem is divided into four parts. It starts with initialization, which is essentially required in Grover’s Algorithm.

Initialization

Let there be a graph which has n vertices and e edges, k is the given number of colors, then the total number of data qudits that are required for representing all the colored vertices are $n \cdot \lceil \log_d k \rceil$. The oracle performs a check to find

all possible right combination of properly colored vertices with k /fewer colors from a combination of all possible colored vertices. A superposition of $m = n \cdot \lceil \log_d k \rceil$ qudits thus generates all possible combination of colored vertices. The initial data qudits in Fig. 5 include m qudits prepared in the ground state $|\psi\rangle = |0\rangle^{\otimes m}$, due to the re-usability property of ancilla qudits, $r = n$ ancilla qudits in the excited state $|\theta\rangle = |d - 1\rangle^{\otimes r}$ (These r ancilla qudits are required to prepare Invalid_Color detector block and comparator block that are thoroughly explained in the next subsection), one ancilla qudit in the ground state $|\zeta\rangle = |0\rangle$ (1 ancilla is required if invalid color exists), and one output qudit in the excited state $|\phi\rangle = |d - 1\rangle$ is required to perform generalized CNOT/Toffoli/MCT operation of the oracle. This entire initialization can be mathematically depicted as:

$$|\psi\rangle \otimes |\theta\rangle \otimes |\zeta\rangle \otimes |\phi\rangle = |0\rangle^{\otimes m} \otimes |d - 1\rangle^{\otimes r} \otimes |0\rangle \otimes |d - 1\rangle$$

Generalized Hadamard Transformation

The next step after the initialization is Hadamard transformation. The generalized Hadamard transform $F_d^{\otimes m}$ on data qudits and F_d on output qudit is performed, hence all possible states are superposed as $|\psi_0\rangle \otimes |\theta_0\rangle \otimes |\zeta_0\rangle \otimes |\phi_0\rangle$, where

$$|\psi_0\rangle = \frac{1}{\sqrt{d^m}} \sum_{i=0}^{d^m-1} |i\rangle$$

$$|\theta_0\rangle = |(d - 1)(d - 1)(d - 1)(d - 1) \dots r(\text{times})\rangle$$

$$|\zeta_0\rangle = |0\rangle$$

$$|\phi_0\rangle = \frac{1}{\sqrt{d}} (|0\rangle + \omega^{d-1}|1\rangle + \omega^{2(d-1)}|2\rangle + \dots + \omega^{(d-1)(d-1)}|d - 1\rangle)$$

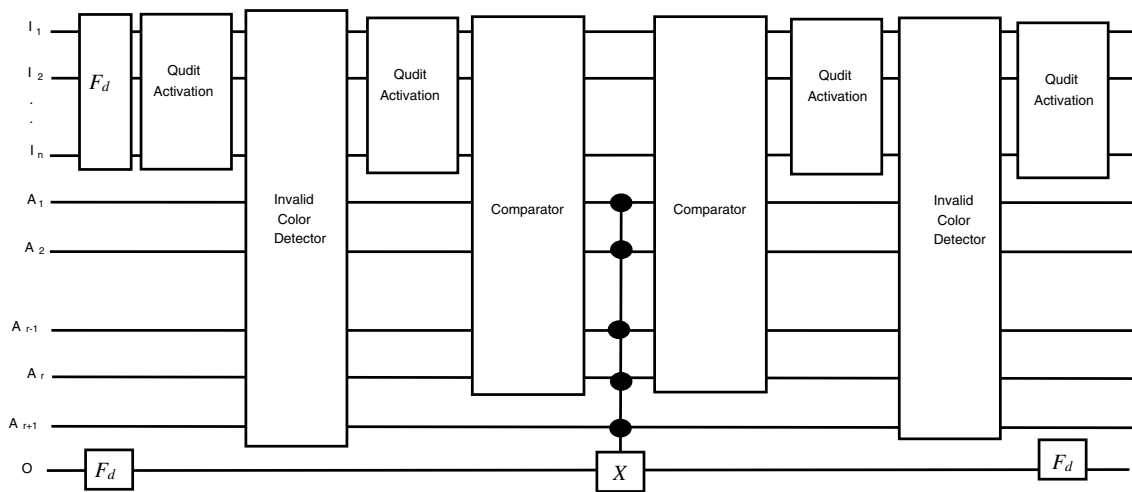


Fig. 5 Block diagram of generalized oracular circuit for k -coloring problem in any dimensional quantum system

Proposed U_f Transformation for d -dimensional quantum systems:

The proposed unitary U_f transformation is distinctly divided into two parts.

(1) Reduction of Invalid Colors: As color, $c = \lceil \log_d k \rceil$, so maximum d^c colors are considered. If $d^c = k$, then all colors are valid colors, else there exists a set of $d^c - k$ invalid colors. The valid colors are used to optimize the search space. Using the following steps this can be executed:

Qudit Activation: Colors should be numbered sequentially as $\{0, 1, 2, \dots, d^c - 1\}$. After the generalized Hadamard transformation, the input data qudit lines act as the d -dimensional representation of combination of all possible colored vertices. The oracle performs a check for only k , which is the combination of valid colors. All the input qudit lines should be in the excited state $|d - 1\rangle$ for those particular combinations of invalid colors by making input qudit lines suitable as control lines for Generalized CNOT/Toffoli/MCT operation to be assured that the oracle is checking only the k -colored combination of vertices. A number of generalized NOT gates must be imposed on the input qudit lines, that are in the ground state $|0\rangle$ accompanied by the application of 'Invalid Color Detector'. Again after 'Invalid Color Detector', this 'Qudit Activation' is to be applied to roll back to the initial superposed quantum state.

Invalid Color Detector: In any combination of colored vertices if any invalid color is noticed then that combination is rejected with the use of the function ICD (Invalid Color Detector) as:

$$\text{Generalized_ICD}(I_1, I_2, \dots, I_n, f) = \begin{cases} f = 0, & \text{if } I_1 \text{ or } I_2 \text{ or } \dots \text{ or } I_n = \text{Invalid color;} \\ f = d - 1, & \text{No invalid color.} \end{cases} \quad (4)$$

As an instance, the circuit synthesis of 'Invalid Color Detector' for ternary quantum systems is described in Fig. 6. This is functionally shown in Equation 5 for n vertices, where I_1, I_2, \dots, I_n are the data qutrits.

$$\text{Ternary_ICD}(I_1, I_2, \dots, I_n, f) = \begin{cases} f = 0, & \text{if } I_1 \text{ or } I_2 \text{ or } \dots \text{ or } I_n = \text{Invalid color;} \\ f = 2, & \text{No invalid color.} \end{cases} \quad (5)$$

(2) Comparator: A newly proposed generalized comparator circuit for d -dimensional quantum systems can be defined as:

$$\text{Generalized_Comparator}(a, b, f) = \begin{cases} f = 0, & \text{if } a = b; \\ f = d - 1, & a \neq b. \end{cases} \quad (6)$$

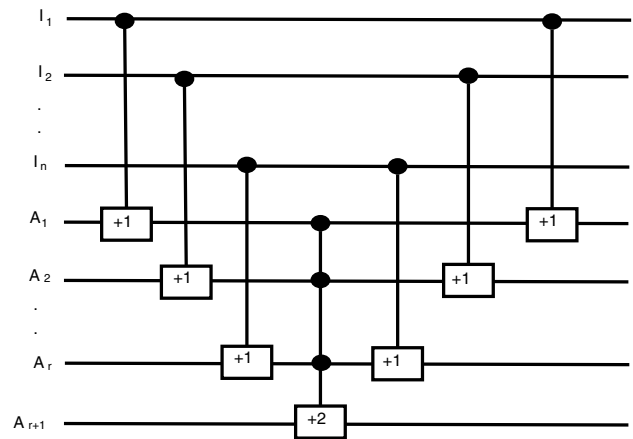


Fig. 6 Ternary invalid color detector

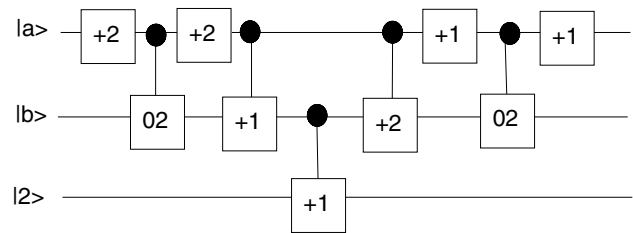


Fig. 7 Ternary comparator

where a and b are the comparing inputs representing the colored vertices of the given graph and the ancilla qudit is f .

Induced from the above, a newly proposed ternary comparator can be defined as:

$$\text{Ternary_Comparator}(a, b, f) = \begin{cases} f = 0, & \text{if } a = b; \\ f = 2, & a \neq b. \end{cases} \quad (7)$$

Circuit synthesis for 2-qutrit ternary comparator is shown in Fig. 7. 1-qutrit and 2-qutrit permutative gate are used to design the complete circuit synthesis for a ternary comparator. Initially, the total number of gate count of our newly proposed comparator is reduced by one as compared to the comparator proposed in [35]. But, gate count can be reduced further at the time of oracular circuit synthesis as one of the outputs of our comparator is '2' instead of '1' [35].

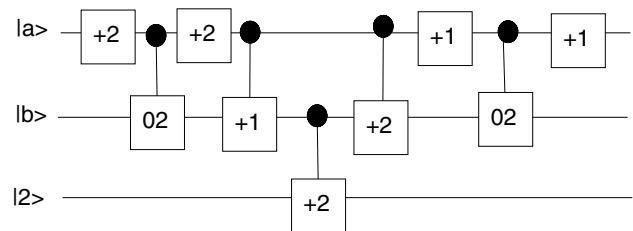


Fig. 8 Ternary inverse comparator

As everything has to be mirrored at the time of circuit synthesis in order to eliminate the cost of wires, there is a need to design an inverse comparator. Our newly proposed 2-qudit ternary inverse comparator is shown in Fig. 8. This newly proposed ternary inverse comparator has a reduced gate count compared to the work in [35].

Generalized MCT Operation

The $|\phi_0\rangle$, output qudit state, is initialized as:

$$\frac{1}{\sqrt{d}}(|0\rangle + \omega^{d-1}|1\rangle + \omega^{2(d-1)}|2\rangle + \dots + \omega^{(d-1)(d-1)}|d-1\rangle)$$

Applying a Generalized MCT gate on the output line with ancilla qudits as control, an eigenvalue kickback $\omega^{(d-1)(d-1)}$ occurs as a result. This causes a phase shift for the respective

input state(s), which in turn helps in finding out all the combinations of a properly colored set of vertices. The algorithm that causes the gate level synthesis of the proposed method is discussed in the next subsection.

Proposed Generalized Algorithm for Oracle Circuit Synthesis of k -Coloring Problem in Any Dimensional Quantum System

The proposed algorithm Algorithm 1 (AutoGenOracle_K-Coloring) of automated oracular circuit synthesis for the k -coloring problem in any dimensional quantum system is illustrated in this subsection. The algorithm accepts the adjacency matrix of the given graph and the number of colors k as input parameters. A circuit netlist in the form of QASM is the output of the algorithm.

ALGORITHM 1: AutoGenOracle_K-Coloring($G(V, E)$)

- 1: **INPUT** : Adjacency matrix $adj(n, n)$ of graph(G) $G(V, E)$, $V = n$ and $E = e$ where, V is the set of nodes and E is the set of edges, Number of input data qudit lines required $I_r = n \cdot \lceil \log_d k \rceil$ (input lines for n nodes and k colors, $d = d$ -dimensional quantum systems)+ ancilla lines required = $n + 1$ (ancilla line for reduction of invalid colors(if required)) + 1(output line(O)), A_r represents ancilla line where, $1 \leq r \leq n$, A_{r+1} represents ancilla line for invalid color (if required).
 - 2: **OUTPUT** : Circuit netlist (QASM)
 - 3: Initialize I_r input lines with $|0\rangle$ followed by generalized Hadamard gate (F_d), ancilla lines A_r with $|d-1\rangle$, A_{r+1} with $|0\rangle$, and output line O with $|d-1\rangle$ followed by a generalized Hadamard gate (F_d).
 - 4: Apply generalized 'Invalid Color Detector' (if required) for all possible invalid colors with suitable Qudit Activation with I_r, A_r as control and A_{r+1} as target.
 - 5: $l \leftarrow n, f \leftarrow 1$
 - 6: for $i \leftarrow 0$ to $n-2$ do
 - 7: $r \leftarrow f, m \leftarrow f$
 - 8: for $j \leftarrow i+1$ to $n-1$ do
 - 9: if $adj(i, j) \leftarrow 1$ (i and j are connected by an edge(e)) then
 - 10: Use a generalized comparator circuit with the input lines (I_i, I_j) corresponding (i, j) as control and the ancilla line A_r as target.
 - 11: $r \leftarrow r + 1$
 - 12: end if
 - 13: end for
 - 14: if $r > f + 1$ then
 - 15: Use a generalized Toffoli/MCT gate with all ancilla lines A_r as control and A_l as target
 - 16: $l \leftarrow l - 1$
 - 17: for $m \leftarrow n-1$ to $i+1$ do
 - 18: if $adj(i, m) \leftarrow 1$ (i and m are connected by an edge(e)) then
 - 19: Use a generalized inverse comparator circuit with the input lines (I_i, I_m) corresponding (i, m) as control and the ancilla line A_m as target.
 - 20: $m \leftarrow m + 1$
 - 21: end if
 - 22: end for
 - 23: else if $r = f + 1$ then
 - 24: $f \leftarrow f + 1$
 - 25: end if
 - 26: end for
 - 27: Use a generalized MCT gate with all the ancilla lines A_1, A_2, \dots, A_{r+1} as control and O as output.
 - 28: Repeat step 5-26.
 - 29: Repeat step 4.
-

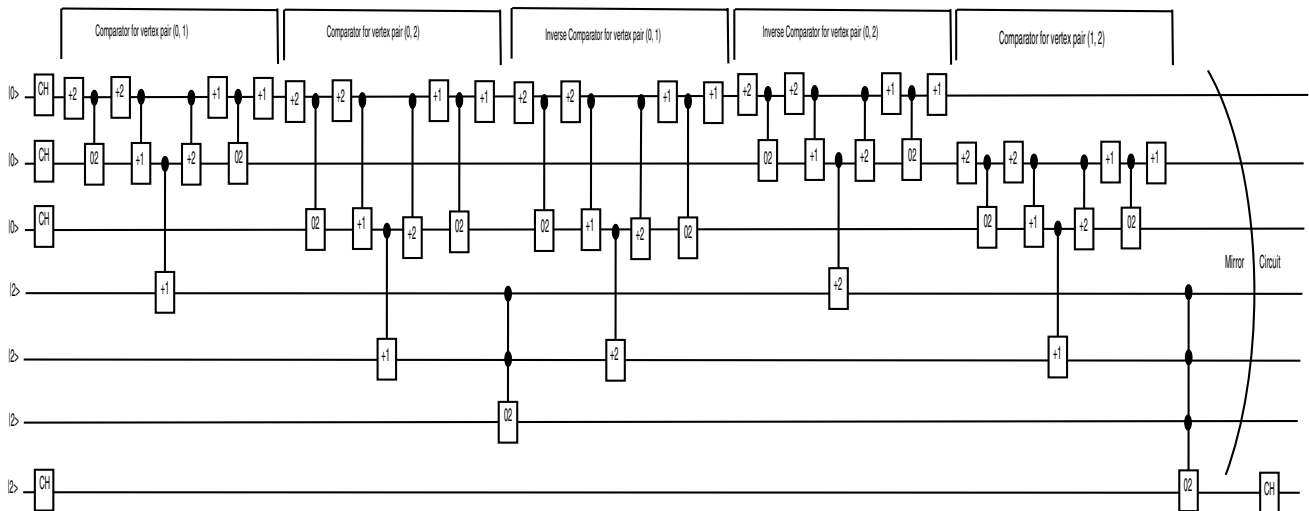


Fig. 9 Oracle circuit for k -coloring problem of a graph with three vertices in ternary quantum systems

The total number of qudit lines that are required for the generation of the oracle circuit can be easily evaluated from the details of the adjacency matrix and the number of given colors. All the input data qudits are initialized with $|0\rangle$ followed by generalized Hadamard, ancilla lines (A_r) are initialized with $|d - 1\rangle$, ancilla line A_{r+1} is initialized with $|0\rangle$ and the output line is initialized with $|d - 1\rangle$ followed by generalized Hadamard. First of all, 'Invalid Color Detector' is applied with suitable Qudit Activation (if invalid color exists) with I_r, A_r as control and A_{r+1} as the target. After that, between two adjacent vertices (i, j), a generalized comparator circuit is used with two input lines (i, j) as control and the ancilla line (A_r) as output. They perform the same task for all the adjacent vertices. Following this, a generalized MCT gate is used with all the ancilla lines A_r and A_{r+1} as control and the output line as output for the flip operation of the Grover's oracle. To achieve the mirror of the oracle circuit, we have repeated the previous steps as shown in Algorithm 1.

Generation of Proposed Oracle Circuit for k -Coloring Problem for an Exemplified Graph

As an example, the circuit synthesis of the 3-coloring problem for a graph of three vertices with three connected edges (K_3) in ternary quantum systems is illustrated in Fig. 9. For a 3-vertex graph, adjacency matrix is represented by $adj(3, 3)$ and vertices are indexed as 0, 1, 2. As we are solving 3-coloring problem, the value of k is 3. Hence, the total $3 \cdot \lceil \log_3 3 \rceil = 3$ qudit lines are required for inputs. As per Algorithm 1, input qudit lines are represented as I_r , where $1 \leq r \leq 3$. These input qudit lines are initialized with $|0\rangle$ followed by Chrestenson gate (Generalized Hadamard gate in ternary quantum systems) as we have considered ternary

quantum systems. The above three input qudit lines represent three vertices of the given graph. In order to store comparator information, three ancilla lines A_r are initialized with $|2\rangle$, where $1 \leq r \leq 3$. One ancilla line A_{r+1} is not required as there is no invalid color in this case. One output line O initialized with $|2\rangle$ followed by Chrestenson gate is required to store the output of the circuit. Hence, a total of 7 qudit lines are required for the simulation.

- At first the algorithm considers to apply 'Invalid Color Detector', if it is required. As the number of color is 3 in this example, So the color c becomes $c = \lceil \log_d k \rceil = \lceil \log_3 3 \rceil = 1$ and $d^c = 3^1 = 3$. Therefore, we can conclude that $d^c = k$, hence, all the given colors are valid colors. So, there is no need of 'Invalid Color Detector'. Let's assume the number of color be given 2, then we had to apply 'Invalid Color Detector' as $d^c \neq k$.
- The algorithm now initializes variable l as 3 and variable f as 1.
- Thereafter, for every possible vertex pair (0, 1), (0, 2), and (1, 2), the algorithm checks that they are connected by an edge or not by accessing the information of adjacency matrix.
- At first for vertex pair (0, 1), $adj(0, 1) = 1$, insert a ternary comparator circuit with I_1, I_2 and A_1 qudit lines.
- In case of vertex pair (0, 2), $adj(0, 2) = 1$, insert an ternary comparator circuit with I_1, I_3 and A_2 qudit lines.
- Again to make two A_1 and A_2 ancilla lines reusable, insert a ternary Toffoli gate with A_1 and A_2 as control and A_3 as target by accessing the value of the variables r and f followed by inserting two inverse comparator circuit for vertex pairs (0, 1) and (0, 2) by accessing the value of the variables l and m .

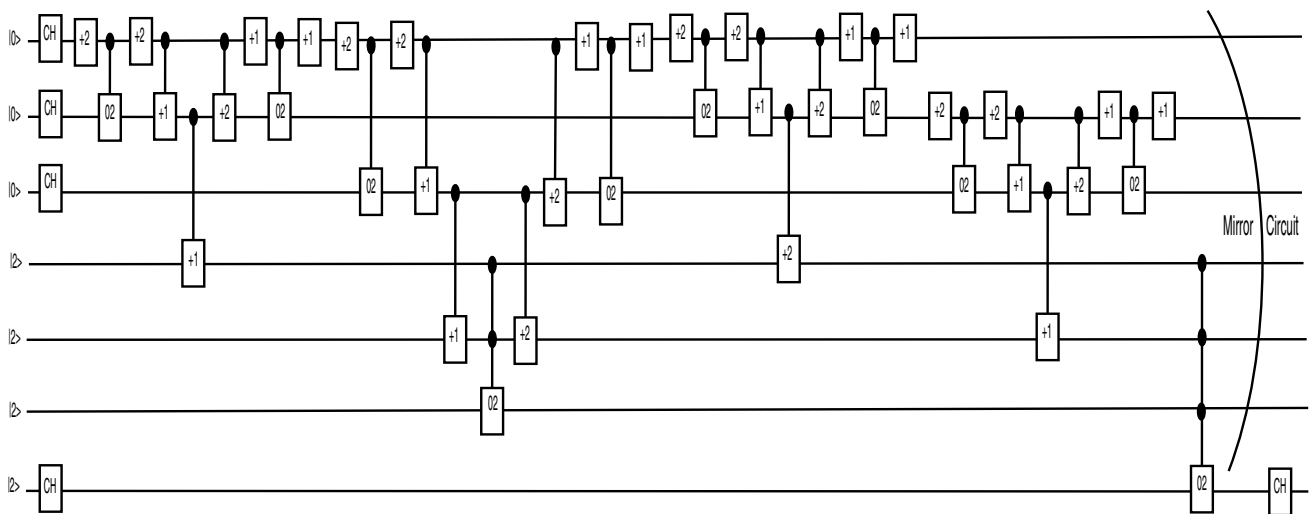


Fig. 10 Gate-optimized oracle circuit for k -coloring problem of a graph with three vertices in ternary quantum systems

- Similarly, for vertex pair (1, 2), the entry for the adjacency matrix $adj(1, 2)$ is 1, so insert a ternary comparator circuit with I_2, I_3 and A_2 qutrit lines after re-initializing the value of the variables r and m to f .
- Thereafter, Insert a ternary three-controlled Toffoli gate with each of the three $A_1, A_2,$ and A_3 ancilla lines as control and O ancilla as target.
- The remaining steps of the algorithm generate the mirror for the original circuit in order to keep the overall circuit netlist reversible.

After optimizing the gate-cost, the circuit synthesis of the 3-coloring problem for above-mentioned example graph is presented in Fig. 10.

Circuit Cost Estimation

The design of generalized oracle for our proposed algorithm has already been described. Now, in this subsection, we furnish the circuit cost analysis of the oracular circuit in Table 2.

$n \cdot \lceil \log_d k \rceil$ data qudits are required for n -vertices graph and k given color. At most $n + 1$ number of ancilla and at most $O(n^2 \cdot \log_d k)$ gates are required to design the oracular circuit for n -vertices graph.

Comparative Analysis

The new ternary oracle circuit for 3-vertex graph has achieved 41% reduction in the quantum gate cost as compared to the most recent related work [28] and 85% reduction in the quantum gate cost as compared to the work [34]. Table 3 shows that the comparative analysis for higher

vertices graphs as well. We show that for 4-vertex and 5-vertex graph, it can reach to 43% reduction of gate cost.

Experimental Results

In this section, we have vividly discuss about the implementation results of k -coloring problem in multi-valued quantum systems.

Simulation Results of k -Coloring Problem in Ternary Quantum Domain

Our ternary circuit instance is verified through simulation with the help of MATLAB simulator [45]. Unfortunately, MATLAB has some limitations of memory constraints. For that, the verification of our circuit needs to be restricted with at most seven qutrits. Therefore, a complete graph of three vertices i.e., a triangle is taken as an example case for simulating the k -coloring problem. The number of color k is given as 3. As per Algorithm 1, three qutrits ($n \cdot \lceil \log_3 k \rceil$) are required to represent the vertices of the given graph. As the example graph have three vertices, three ancilla qutrits are required to perform the comparator circuit synthesis and one final qutrit as output. Three colors are given, which are encoded as $|0\rangle, |1\rangle$ and $|2\rangle$. Generally, a graph with three vertices can be colored with the help of three colors in $3^3 = 27$ different ways, that can be represented in the ternary quantum system as $|000\rangle \dots |222\rangle$. Thus, our database contains 27 elements. In Fig. 11, the gate level representation of the Grover’s circuit is shown.

The simulation steps of Fig. 11 are:

Step 1: At first, all the qutrits of the 3-qutrit data register are initialized with $|0\rangle$.

Step 2: Then, the *CH* gate is applied to create the all possible 27 states ($|000\rangle \dots |222\rangle$), which is portrayed in Fig. 12a.

Step 3: The ternary oracle compares the colors of every adjacent vertices and inverts the amplitude of the solution elements, which are $|012\rangle, |021\rangle, |102\rangle, |120\rangle, |201\rangle, |210\rangle$, illustrated in Fig. 12b.

Step 4: The output of ternary oracle is acted upon by the ternary diffusion operator. This diffusion operator amplifies the amplitudes of the marked states of step 3.

Step 5: Steps 3 and 4 are repeated for $\sqrt{N/M}$ times (For the multiple solution of Grover’s operator having $N =$ number of elements in the database and $M =$ number of marked states), where the value of N is $d^{n \cdot \log_d k}$ for this k -coloring problem.

The probability distribution of quantum states after two iterations of Grover’s operator is shown in Fig. 12c, where the amplitude amplification is performed using diffusion operator to amplify the amplitudes of marked states. It can be verified that the location of the searched states (output of step 5) by converting its index values to its equivalent ternary value, having higher probability from all the index values of Fig. 12c. For example, if index value is taken as five from Fig. 12c, after converting it to ternary, we get the string $|012\rangle$, which is one of the searched states. Analysis of the simulation result confirms that our oracle successfully verifies that any set of vertices of a graph, is properly colored by the given set of colors.

Mapping of k -Coloring Problem to Multi-valued Quantum Systems

Multi-valued quantum hardware is not presently publicly available for common users. But, in this section, we discuss about a few works which guide us to map a multi-valued quantum algorithm to quantum hardware in future. As in binary quantum systems, for this mapping, generalized

multi-controlled Toffoli gates are needed to be decomposed into one and two-qudit gates followed by a qudit mapping algorithm. In [41], Saha et. al. have shown the decomposition of multi-controlled Toffoli gate in d -dimensional quantum systems. A generalized Toffoli decomposition in d -dimensional system using $|d\rangle$ state is shown in Fig. 13. An akin construction for the Toffoli gate in binary using qutrit is evident from a previous work [46]; Saha et. al. have extended it for d -dimensional quantum systems. The idea is to execute an X_d operation on the target qudit (third qudit) if and only if the two control qudits, are both $|d - 1\rangle$. Firstly, a $|d - 1\rangle$ -controlled X_{d+1}^+ , where $+1$ and $d + 1$ are used to denote that the target qudit is incremented by 1 (mod $d + 1$), is implemented on the first and the second qudits. This eventually upgrades the second qudit to $|d\rangle$ as long as the first and the second qudits were both $|d - 1\rangle$. Then, a $|d\rangle$ -controlled X_d gate is applied to the target qudit. Therefore, X_d is executed only when both the first and the second qudits were $|d - 1\rangle$, as expected. The controls are rolled back to their original states by a $|d - 1\rangle$ -controlled X_{d+1}^- gate, which reverses the effect of the first gate. The most important aspect in this decomposition is that the $|d\rangle$ state from $d + 1$ -dimensional quantum systems can be used instead of ancilla to store temporary information.

For further decomposition of the Toffoli for simulation purpose, the $d + 1$ -dimensional Toffoli gate has been decomposed into $d + 2$ -dimensional CNOT gates. A generalized CNOT gate for $d + 2$ -dimensional quantum system as $C_{X,d+2}^+$, where $+1$ and $d + 2$ denote that the target qudit is incremented by 1 (mod $d + 2$) only when the control qudit value is $d + 1$. The $((d + 2)^2 \times (d + 2)^2)$ matrix representation of the $C_{X,d+2}^+$ gate is as follows:

$$C_{X,d+2}^+ = \begin{pmatrix} I_{d+2} & 0_{d+2} & 0_{d+2} & \dots & 0_{d+2} \\ 0_{d+2} & I_{d+2} & 0_{d+2} & \dots & 0_{d+2} \\ 0_{d+2} & 0_{d+2} & I_{d+2} & \dots & 0_{d+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_{d+2} & 0_{d+2} & 0_{d+2} & \dots & X_{d+2}^+ \end{pmatrix}$$

where X_{d+2}^+ and 0_{d+2} are both $(d + 2) \times (d + 2)$ matrices as shown below:

Table 2 Circuit cost analysis of oracle

No. of vertex	No. of color	Data qudits required	Maximum ancilla required	Maximum gate count
n	k	$n \cdot \lceil \log_d k \rceil$	$O(n)$	$O(n^2 \cdot \log_d k)$

Table 3 Comparative analysis of ternary oracle

No. of vertex	Proposed gate cost	Gate cost [28]	Reduction(%)	Gate cost [34]	Reduction(%)
3	62	106	41%	343	85%
4	170	298	43%	< 1000	86%
5	282	494	43%	< 2700	86%

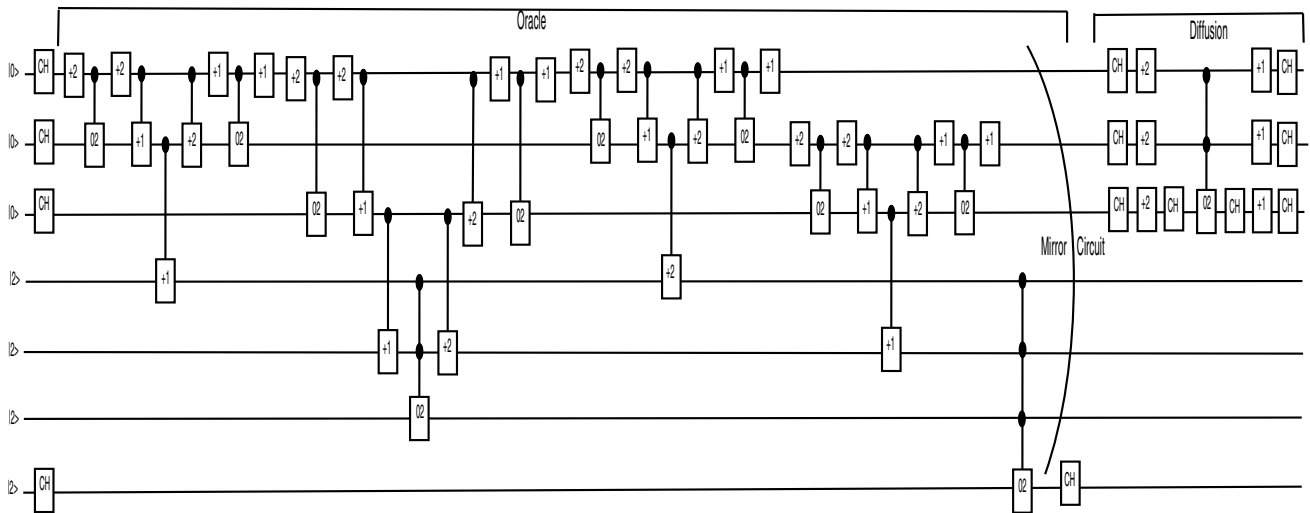


Fig. 11 Complete circuit for k -coloring problem of a graph with three vertices in ternary quantum systems

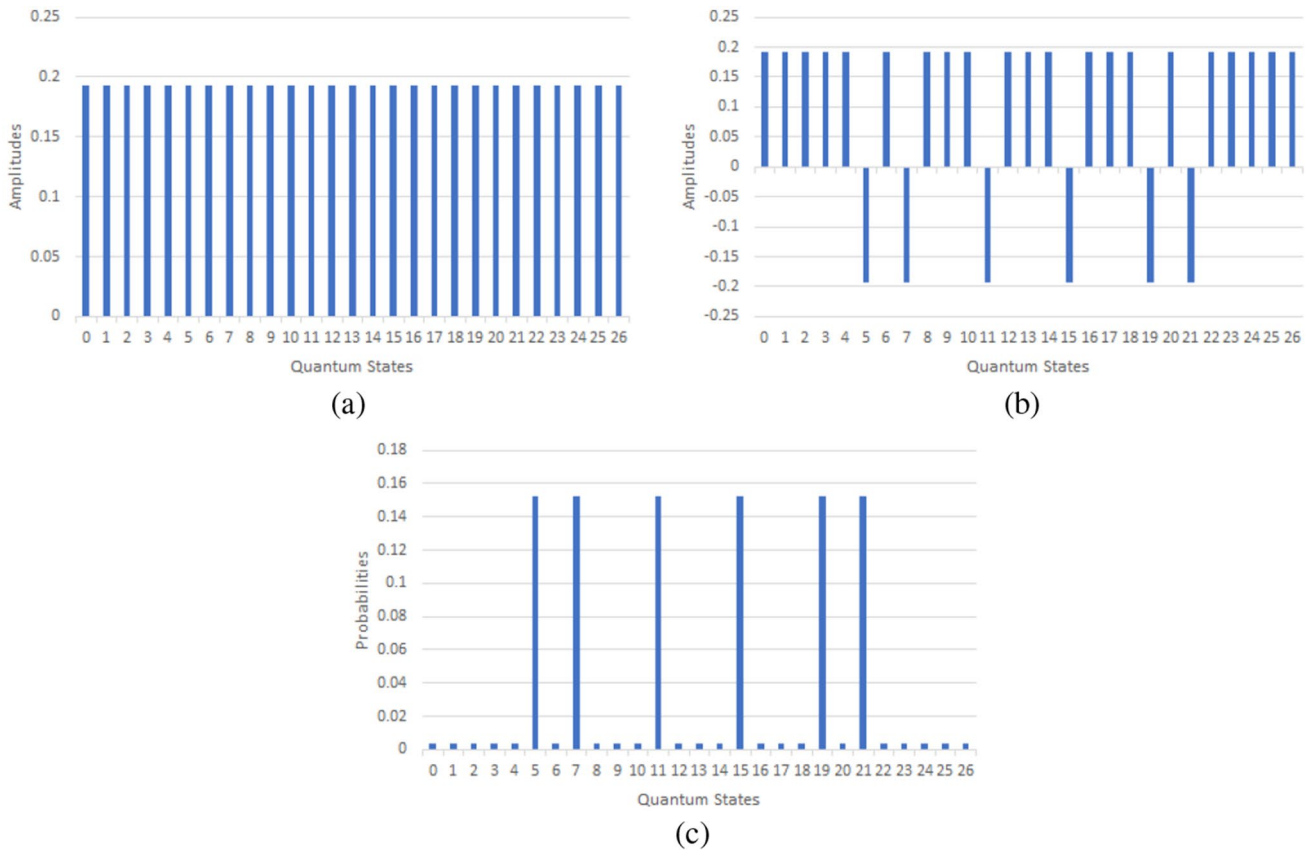


Fig. 12 Simulation results: (a) the amplitude of the quantum states after applying CH gate to input qutrits (b) The amplitude of the quantum states after applying oracle (c) The probability distribution of the quantum states after two iterations of Grover's operator

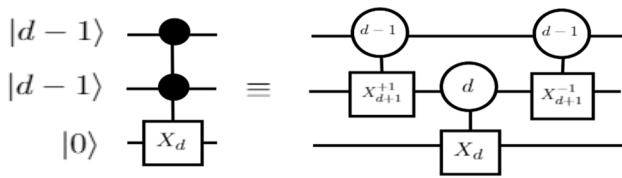


Fig. 13 Generalized Toffoli in d -dimensional quantum systems [41]

$$X_{d+2}^{+1} = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix} \quad \text{and,} \quad 0_{d+2} = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{pmatrix}$$

As an example, a 8-qudit Toffoli gate is shown in Fig. 14a. First, we need to decompose it as in [46] as shown in Fig. 14b. Further, we need to decompose all the $d + 1$ -dimensional Toffoli gates into $(d + 2)$ -dimensional CNOT gates as shown in Fig. 14c with the help of the Saha et. al. decomposition of the generalized Toffoli in any dimensional quantum system [41]. All the $d - 1$ -controlled Toffoli gates are decomposed into $d - 1$ -controlled and d -controlled CNOT gates as shown in Fig. 14c. Likewise, all the d -controlled Toffoli gates are decomposed into d -controlled and $d + 1$ -controlled CNOT gates. Consequently, with the help of $|d\rangle$ and $|d + 1\rangle$ quantum state of $(d + 2)$ -dimensional system, X_d is carried out effectively if all the controlled qudits are in

$|d - 1\rangle$ state. In this way, any dimensional multi-controlled Toffoli gate can be decomposed. In [18], the authors have exhibited that how quantum states could be moved without using SWAP gate in any dimensional quantum system. Thus, it paves the path for future qudit mapping problem to be implemented, in a cost-effective manner.

Conclusion

Here, in this paper, we have proposed an end-to-end framework, which includes mapping of k -coloring problem to any NISQ devices/multi-valued quantum technology through automatic generation of oracle circuit using Grover’s search followed by MCT realization for any dimensional quantum system. This proposed approach for any dimensional quantum system is applicable for any undirected and unweighted given graph, which makes our approach generalized. We exhibited that our comparator-based approach requires $n \cdot \lceil \log_d k \rceil$ data qudits, which leads to a query complexity $O(\sqrt{n \cdot \log_d k})$ to solve k -coloring problem using Grover’s algorithm. Further, we have shown that our newly proposed ternary comparator is a key component in designing the circuit for the k -coloring problem using quantum search algorithm. Our new ternary oracle circuit achieved an at least 41% reduction in the quantum gate cost as compared to the most recent related work [28] and an at least 85% reduction in the quantum gate cost as compared to the work [34]. In the future, with the evolution of qudit-supported quantum hardware, we would like to validate our designs.

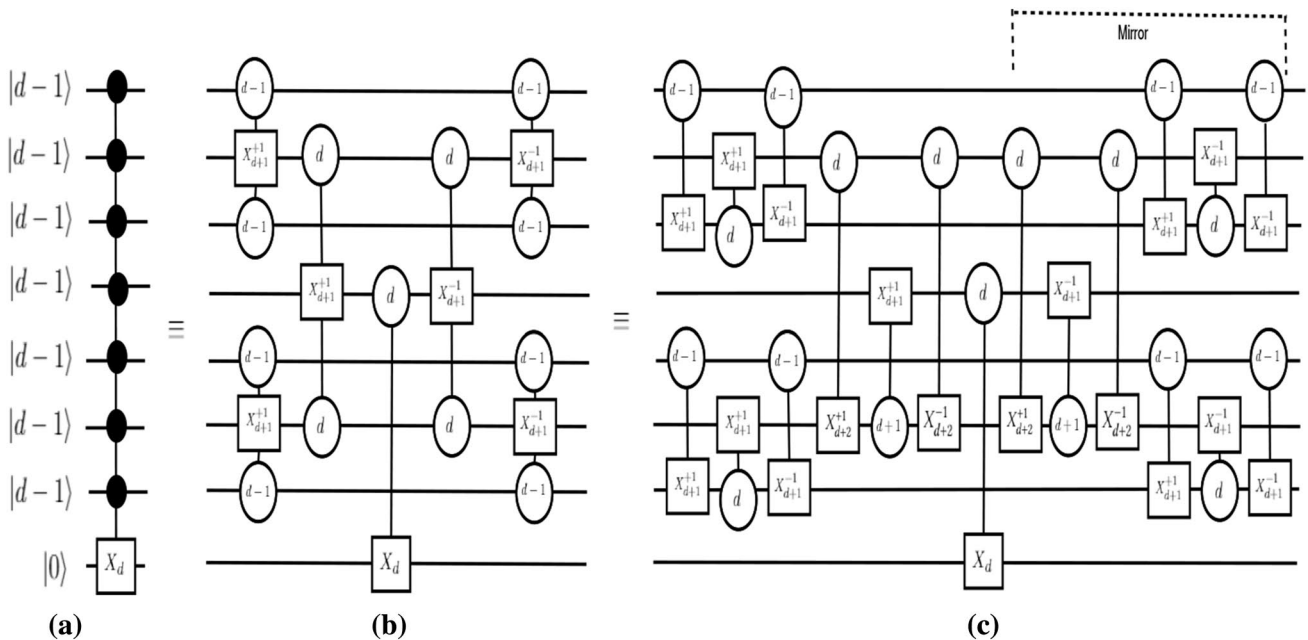


Fig. 14 Decomposition of 8-qudit Toffoli gate : (a) 8-qudit Toffoli gate (b) decomposed into generalized Toffoli gates (c) decomposed into one and two-qudit Generalized CNOT gates [41]

Appendix 1

Here, we thoroughly discuss about generalized quantum gate along with binary and ternary gates that are used in this paper for designing the proposed circuit.

Generalized NOT Gate: X_d , the generalized NOT or increment gate, for a $(d \times d)$ matrix is as follows:

$$X_d = \begin{pmatrix} 0 & 0 & \dots & 0 & 1 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{pmatrix}$$

Generalized Phase-Shift Gate: Z_d , the generalized phase-shift gate of a $(d \times d)$ matrix is as follows, with $\omega = e^{\frac{2\pi i}{d}}$;

$$Z_d = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & \omega & 0 & \dots & 0 \\ 0 & 0 & \omega^2 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \omega^{d-1} \end{pmatrix}$$

Generalized Hadamard Gate: F_d , the generalized quantum Fourier transform or generalized Hadamard gate, produces the superposition of the input basis states. The $(d \times d)$ matrix representation of it is as shown below :

$$F_d = \frac{1}{\sqrt{d}} \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^2 & \dots & \omega^{d-1} \\ 1 & \omega^2 & \omega^4 & \dots & \omega^{2(d-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \omega^{d-1} & \omega^{2(d-1)} & \dots & \omega^{(d-1)(d-1)} \end{pmatrix}$$

Generalized CNOT Gate: Quantum entanglement is a unparalleled property of quantum mechanics, and can be attained by a controlled NOT (CNOT) gate in a binary quantum system. For d -dimensional quantum systems, the binary 2-qubit CNOT gate is generalised to the *INCREMENT* gate:

$INCREMENT|x\rangle|y\rangle = |x\rangle|(x + y) \pmod{d}$, if $x = d - 1$, and $= |x\rangle|y\rangle$, otherwise.

The $(d^2 \times d^2)$ matrix representation of the generalized CNOT $C_{X,d}$ gate is as follows:

$$C_{X,d} = \begin{pmatrix} I_d & 0_d & 0_d & \dots & 0_d \\ 0_d & I_d & 0_d & \dots & 0_d \\ 0_d & 0_d & I_d & \dots & 0_d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_d & 0_d & 0_d & \dots & X_d \end{pmatrix}$$

where I_d and 0_d are both $d \times d$ matrices as shown below:

$$I_d = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \quad \text{and,} \quad 0_d = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}$$

Generalized Multi-controlled Toffoli Gate: We extend the generalized CNOT or *INCREMENT* further to operate over n qudits as a generalized Multi-controlled Toffoli Gate or n -qudit Toffoli gate $C_{X,d}^n$. For $C_{X,d}^n$, the target qudit is increased by 1 (mod d) only when all $n - 1$ control qudits have the value $d - 1$. The $(d^n \times d^n)$ matrix representation of generalized Multi-controlled Toffoli (MCT) gate is as follows:

$$C_{X,d}^n = \begin{pmatrix} I_d & 0_d & 0_d & \dots & 0_d \\ 0_d & I_d & 0_d & \dots & 0_d \\ 0_d & 0_d & I_d & \dots & 0_d \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0_d & 0_d & 0_d & \dots & X_d \end{pmatrix}$$

Owing to technology constraints, a multi-controlled Toffoli gate can be substituted by an equivalent circuit comprising one-qudit and/ two-qudit gates, although at first the multi-controlled Toffoli must be decomposed into a set of Toffoli gates for any dimensional quantum system. The binary [38] and ternary gate [19, 40] representation of all the above mentioned generalized quantum gates are portrayed as follows (Fig. 15):

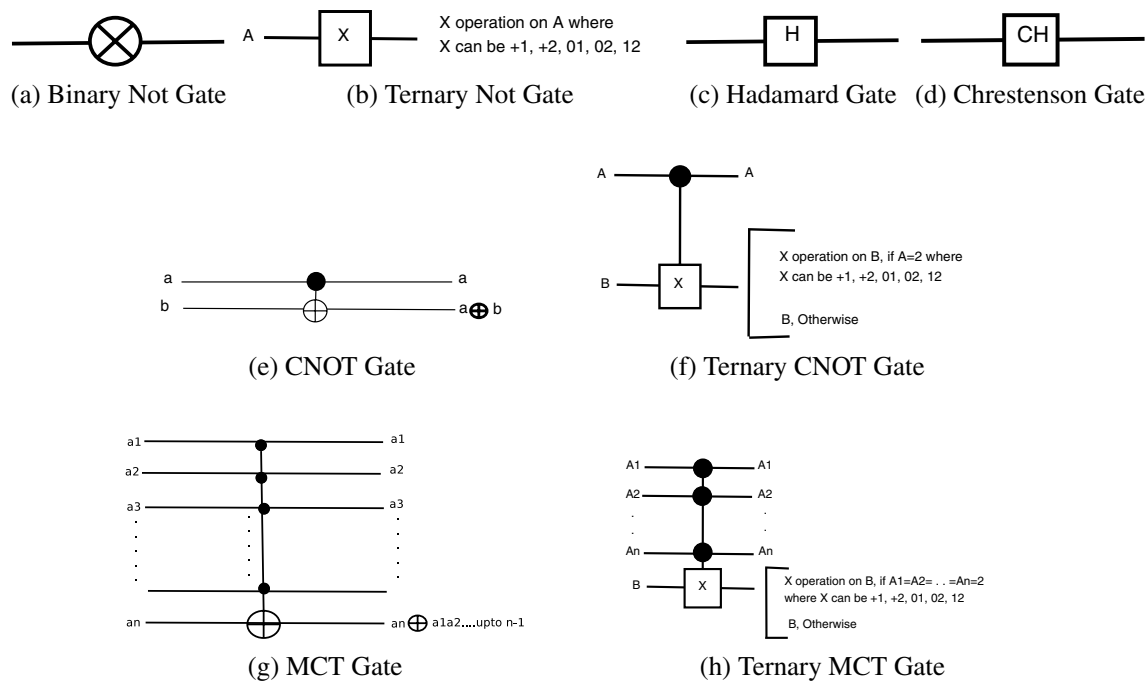


Fig. 15 Binary and ternary quantum gates

Declarations

Conflict of interest The authors declare that they have no conflict of interest

References

- Saha A, Saha D, Chakrabarti A. Circuit design for K-coloring problem and its implementation on near-term quantum devices. *IEEE Int Symp Smart Electron Syst.* 2020. <https://doi.org/10.1109/iSES50453.2020.00015>.
- Nielsen MA, Chuang IL. *Quantum computation and quantum information.* Cambridge: Cambridge University Press; 2002.
- Farhi E, Gutmann S. Quantum computation and decision trees. *Phys Rev A.* 1998;58(2):915–28. <https://doi.org/10.1103/PhysRevA.58.915>.
- Shor PW. Algorithms for quantum computation: discrete logarithms and factoring. *Found Comput Sci.* 1994 pp. 124–134. <https://doi.org/10.1109/SFCS.1994.365700>.
- Preskill J. Quantum computing in the NISQ era and beyond. *Quantum.* 2018. <https://doi.org/10.22331/q-2018-08-06-79>.
- Sanyal A, Saha A, Saha B, Chakrabarti A. Circuit design of clique problem and its implementation on NISQ using combinatorial approach of classical-quantum hybrid model. 2020. [arXiv:2004.10596](https://arxiv.org/abs/2004.10596).
- Karp RM. Reducibility among combinatorial problems. In: Miller RE, Thatcher JW, Bohlinger JD, editors. *Complexity of computer computations, (The IBM Research Symposia Series).* Boston: Springer; 1972. p. 85–103.
- Grover L. A fast quantum mechanical algorithm for database search. *Proc ACM Symp Theory Comput.* 1996;96:212–9.
- Koch J, Yu TM, Gambetta J, Houck AA, Schuster DI, Majer J, Blais A, Devoret MH, Girvin SM, Schoelkopf RJ. Charge-insensitive qubit design derived from the cooper pair box. *Phys Rev A.* 2007;A76(4):042319. <https://doi.org/10.1103/PhysRevA.76.042319>.
- Klimov AB, Guzmán R, Retamal JC, Saavedra C. Qutrit quantum computer with trapped ions. *Phys Rev.* 2003;A67(6):062313. <https://doi.org/10.1103/PhysRevA.67.062313>.
- Adcock MRA, Høyer P, Sanders BC. Quantum computation with coherent spin states and the closeHadamard problem. *Quant Inform Process.* 2016;15(4):1361–86. <https://doi.org/10.1007/s11128-015-1229-0>.
- Bartlett SD, de Guise H, Sanders BC. Quantum encodings in spin systems and harmonic oscillators. *Phys Rev.* 2002;A65:5. <https://doi.org/10.1103/physreva.65.052316>.
- Bocharov A, Roetteler M, Svore KM. Factoring with qutrits: shor's algorithm on ternary and metaplectic quantum architectures. *Phys Rev A.* 2017. <https://doi.org/10.1103/PhysRevA.96.012306>.
- Cui SX, Hong SM, Wang Z. Universal quantum computation with weakly integral anyons. *Quant Inform Process.* 2015;14(8):2687–727. <https://doi.org/10.1007/s11128-015-1016-y>.
- Dogra S, Arvind Dorai K. Determining the parity of a permutation using an experimental NMR qutrit. *Phys Lett.* 2014;A378(46):3452–6. <https://doi.org/10.1016/j.physleta.2014.10.003>.
- Gao X, Erhard M, Zeilinger A, Krenn M. Computer-inspired concept for high-dimensional multipartite quantum gates. *Phys Rev Lett.* 2020. <https://doi.org/10.1103/physrevlett.125.050501>.
- Leuenberger M. N, Loss D. Quantum computing in molecular magnets. *Nature.* 2001;410(6830):789–93. <https://doi.org/10.1038/35071024>.
- Saha A, Saha D, Chakrabarti A. Moving quantum states without SWAP via intermediate higher dimensional qudits. 2021. [arXiv:2106.09089](https://arxiv.org/abs/2106.09089).
- Muthukrishnan A, Stroud CR Jr. Multi-valued logic gates for quantum computation. *Phys Rev A.* 2000;62:0523091–8.

20. Saha A, Mandal SB, Saha D, Chakrabarti A. One-dimensional lazy quantum walk in ternary system. *IEEE Trans Quant Eng.* 2021. <https://doi.org/10.1109/TQE.2021.3074707>.
21. Fan Y. Application of multi-valued quantum algorithm. 2008. [arXiv:0809.0932](https://arxiv.org/abs/0809.0932).
22. Li X, Yang G, Zheng D. Logic synthesis of ternary quantum circuits with minimal qutrits. *Proc J Comput.* 2013;8(8):1941–46.
23. Bocharov A, Cui SX, Roetteler M, Svore KM. Improved quantum ternary arithmetics. 2015. [arXiv:1512.03824](https://arxiv.org/abs/1512.03824).
24. Fan F, Yang G, Yang G, Hung WNN. A synthesis method of quantum reversible logic circuit based on elementary qutrit quantum logic gates. *J Circuits Syst Comput.* 2015;24(08):1550121. <https://doi.org/10.1142/S0218126615501212>.
25. Khan FS, Perkowski M. Synthesis of multi-qudit hybrid and d-valued quantum logic circuits by decomposition. *Theoret Comput Sci.* 2006;367(3):336–46. <https://doi.org/10.1016/j.tcs.2006.09.006>.
26. Kiktenko EO, Nikolaeva AS, Peng X, Shlyapnikov GV, Fedorov AK. Scalable quantum computing with qudits on a graph. *Phys Rev.* 2020. <https://doi.org/10.1103/physreva.101.022304>.
27. Saha A, Chongder A, Mandal SB, Chakrabarti A. Synthesis of vertex coloring problem using Grover's algorithm. *IEEE Int Symp Nanoelectron Inform Syst.* 2015;2015:101–6. <https://doi.org/10.1109/iNIS.2015.55>.
28. Mandal SB, Chakrabarti A, Sur-Kolay S. Synthesis of ternary Grover's algorithm. In: *IEEE 44th International Symposium on Multiple-Valued Logic (ISMVL)*, 2014.
29. Ivanov SS, Tonchev HS, Vitanov NV. Time-efficient implementation of quantum search with qudits. *Phys Rev A.* 2012. <https://doi.org/10.1103/PhysRevA.85.062321>.
30. Hunt S, Gadouleau M. Grover's Algorithm and many-valued quantum logic. 2020. [arXiv:2001.06316](https://arxiv.org/abs/2001.06316).
31. Shimizu K, Mori R. Exponential-time quantum algorithms for graph coloring problems. [arXiv:1907.00529](https://arxiv.org/abs/1907.00529).
32. Hu S, Liu P, Chen CR, Pistoia M, Gambetta J. Reduction-based problem mapping for quantum computing. *Computer.* 2019;52(6):47–57. <https://doi.org/10.1109/MC.2019.2909709>.
33. IBM Research. Quantum Experience. <http://www.research.ibm.com/quantum/>.
34. Wang Y, Perkowski M. Improved complexity of quantum oracle for ternary Grover algorithm for graph coloring. In: *41st International Symposium on Multiple-Valued Logic*, Tusulu: Finland, 2009, pp. 294–301.
35. Khan MHA. Design of reversible/quantum ternary comparator circuits. *Eng Lett.* 2008.
36. Tabi Z, El-Safty KH, Kallus Z, Haga P, Kozsik T, Glos A, Zimboras Z. Quantum optimization for the graph coloring problem with space-efficient embedding. *IEEE Int Conf Quant Comput Eng.* 2020. <https://doi.org/10.1109/qce49297.2020.00018>.
37. Glos A, Krawiec A, Zimboras Z. Space-efficient binary optimization for variational computing. 2020. [arXiv:2009.07309](https://arxiv.org/abs/2009.07309).
38. Barenco A, Bennett CH, Cleve R, DiVincenzo DP, Margolus N, Shor P, Sleator T, Smolin JA, Weinfurter H. Elementary gates for quantum computation. *Phys Rev A.* 2019;52(5):3457–67.
39. Di YM, Wei HR. Synthesis of multi-valued quantum logic circuits by elementary gates. *Phys Rev A.* 2013;87:1. <https://doi.org/10.1103/PhysRevA.87.012325>.
40. Di YM, Wei HR. Elementary gates for ternary quantum logic circuit. 2020. [arXiv:1105.5485](https://arxiv.org/abs/1105.5485).
41. Saha A, Majumdar R, Saha D, Chakrabarti A, Sur-Kolay S. Asymptotically improved Grover's algorithm in any dimensional quantum system with novel decomposed n-qudit Toffoli gate. 2020. [arXiv:2012.04447](https://arxiv.org/abs/2012.04447).
42. Acasiete F, Agostini FP, Khatibi Moqadam J, Portugal R. Experimental implementation of quantum walks on IBM quantum computers. 2020. <https://doi.org/10.1007/s11128-020-02938-5>.
43. Li G, Ding Y, Xie Y. Tackling the qubit mapping problem for NISQ-era quantum devices. *Proc Int Conf Archit Support Program Lang Oper Syst.* 2019. <https://doi.org/10.1145/3297858.3304023>.
44. Tan B, Cong J. Optimal layout synthesis for quantum computing. *Proc Int Conf Comput-Aided Des.* 2020;137:1–9. <https://doi.org/10.1145/3400302.3415620>.
45. MATLAB, R2012b, The MathWorks, Natick, MA, 2012.
46. Gokhale P, Baker JM, Duckering C, Brown NC, Brown KR, Chong FT. Asymptotic improvements to quantum circuits via qutrits. *Proc Int Symp Comput Archit.* 2019. <https://doi.org/10.1145/3307650.3322253>.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.