Network coding for distributed quantum computation over cluster and butterfly networks

Seiseki Akibue and Mio Murao

Abstract—To apply network coding for quantum computation, we study the distributed implementation of unitary operations over all separated input and output nodes of quantum networks. We consider a setting of networks where quantum communication between nodes is restricted to sending just a qubit, but classical communication is unrestricted. We analyze which N-qubit unitary operations are implementable over cluster networks by investigating transformations of a given cluster network into quantum circuits. We show that any two-qubit unitary operation is implementable over the butterfly network and the grail network, which are fundamental primitive networks for classical network coding. We also analyze probabilistic implementations of unitary operations over cluster networks.

Index Terms—Quantum computing, Network Coding, Quantum entanglement

I. INTRODUCTION

DISTRIBUTED quantum computation over multiple spatially separated quantum systems represented by nodes connected by mediating quantum systems represented by edges, is one of the most promising candidates for scalable quantum computation. A serious problem for any kind of distributed computation is the bottleneck problem, which is caused by the collision of communication pathways between the nodes. The bottleneck problem worsens as the scale and complexity of a communication network grow. Thus it is important to consider how to optimize transmission protocols so that the amount of quantum communications is reduced. In classical network information theory, network coding, which incorporates processing at each network node in addition to routing, provides efficient transmission protocols that can resolve the bottleneck problem [1]. Consider a communication task over the butterfly network and the grail network presented in Fig. 1 that aims to transmit single bits x and y from i1 to o2 and i2 to o1 simultaneously via nodes n1, n2, n3 and n4. The directed edges denote transmission channels with 1-bit capacity. One of the channels in each network exhibits the bottleneck without network coding shown in Fig. 1.

Quantum communication with quantum network coding has been studied by analogy to classical network coding [2], [3], [4], [5], [6], [7]. k-pair quantum communication over a network is a unicast communication task to faithfully transmit a k-qubit state given at distinct input nodes \{i_1, i_2, \ldots, i_k\} to distinct output nodes \{o_1, o_2, \ldots, o_k\} through a given network. Two examples of 2-pair quantum communication over a butterfly network and a grail network are shown in Fig. 2. In quantum network coding, perfect multicast communications are not allowed by the no-cloning theorem. This also makes it impossible to perform k-pair communication over the networks by using a simple extension of classical network coding. Indeed, in the setting where each edge can be used for either 1-bit classical communication or 1-qubit quantum communication, perfect quantum 2-pair communication over the butterfly network has been shown to be impossible [2], [3]. However, it has been shown that in the setting where each edge can be used for either 2-bit classical communication or 1-qubit quantum communication, perfect quantum 2-pair communication over the butterfly network is possible, if and only if input nodes share two Bell pairs [4]. In the setting where each edge has 1-qubit channel capacity and classical communication is freely allowed between any nodes, however, it has been shown that there exists a quantum network coding protocol to achieve the 2-pair quantum communication over the butterfly and grail networks perfectly [5], [6], [7].

In k-pair quantum communication, the output state \(|\text{output}\rangle_{o_1\ldots o_k}\) at the output nodes can be regarded as a state obtained by performing a k-qubit unitary operation U on the input state \(|\text{input}\rangle_{i_1\ldots i_k}\) given at the input nodes

\[
|\text{output}\rangle_{o_1\ldots o_k} = U |\text{input}\rangle_{i_1\ldots i_k},
\]

where U is a permutation operation. We do not need to restrict the k-qubit unitary operation U in Eq. (1) to be a permutation operation, it can be a general quantum operation. This leads to the idea of network coding for quantum computation, which aims to perform a quantum operation on a state given at distinct input nodes and to faithfully transmit the resulting

Fig. 1. Network coding for a classical communication task over i) the butterfly network and ii) the grail network. Two bits of information x, y ∈ {0, 1} are given at the input nodes i1 and i2, respectively. x ⊕ y denotes addition of x and y modulo 2.
state to the distinct output nodes efficiently over the network at the same time [8], [9]. By computing and communicating simultaneously, quantum computation over the network may reduce communication resources in the distributed quantum computation scenario.

In this paper, we investigate a cluster network, which is a special class of networks with \( k \) input nodes and \( k \) output nodes, as a first step to apply network coding for more general quantum computation. The cluster network contains the grail network as its special case. We focus on the setting where classical communication is freely allowed between any two nodes. This setting is justified in practical situations, where classical communication is much easier to implement experimentally than quantum communication. We present which class of unitary operations is implementable over cluster networks in this setting by investigating transformations of cluster networks into quantum circuits. The transformation method of cluster networks provides constructions of quantum network coding to implement any two-qubit unitary operations over the grail and butterfly networks, which are fundamental primitive networks for classical network coding. We also analyze probabilistic implementation of \( N \)-qubit unitary operations over the cluster network to understand the properties of quantum network coding for quantum computation when the requirement of deterministic implementations are relaxed but that of exact implementations are kept.

The rest of this paper is organized as follows. In Section II, we define the cluster network and the implementability of a unitary operation over a quantum network. In Section III, we present a method to convert a given cluster network into quantum circuits that implement unitary operations that are implementable over the network. In Section IV, we analyze the implementability of unitary operations over the butterfly and grail networks by construction of protocols. In Section V, we investigate a condition for unitary operations implementable over a given cluster network and show that our conversion method presented in Section III gives all implementable unitary operations over the cluster networks with 2 and 3 input nodes. In Section VI, we investigate probabilistic implementation of unitary operations over the cluster network.

A conclusion is given in Section VII.

II. Cluster Network

We denote the Hilbert space of a set of qubits specified by a set \( Q \) by \( \mathcal{H}_Q \) and the Hilbert space corresponding to a qubit \( Q_k \) specified by an index \( k \) by \( \mathcal{H}_{Q_k} = \mathbb{C}^2 \). In our setting where quantum communications are restricted but classical communications are unrestricted, quantum communication of a qubit state between two nodes is replaced by teleportation between two nodes. Since any direction of classical communications is allowed, quantum communication of a single qubit state can be achieved by sharing a maximally entangled two-qubit state between the nodes and the direction of quantum communication is not limited. Thus quantum network coding is equivalent to perform local operations (at each nodes) and classical communication (LOCC) assisted by the resource state that consists of a set of maximally entangled two-qubit states (the Bell pairs) \( |\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \) shared between the nodes connected by edges.

We investigate which unitary operations are implementable by LOCC assisted by the resource state for a given network where nodes are represented by a two-dimensional lattice. We consider that a node represented by \( v_{i,j} \) is on the coordinate of the two-dimensional lattice \((i,j)\) and edges connect nearest neighbor nodes. We call these networks cluster networks. We first give a formal definition of a cluster network.

**Definition 1.** A network \( G = \{V, E, \mathcal{I}, \mathcal{O}\} \) is a \((k, N)\)-cluster network if and only if,

\[
V = \{v_{i,j}; 1 \leq i \leq k, 1 \leq j \leq N\}
\]

\[
\mathcal{I} = \{v_{i,1}; 1 \leq i \leq k\}
\]

\[
\mathcal{O} = \{v_{k,j}; 1 \leq i \leq k\}
\]

\[
E = \mathcal{S} \cup \mathcal{K}
\]

where

\[
\mathcal{S} = \{(v_{i,j}, v_{i+1,j}); 1 \leq i \leq k-1, 1 \leq j \leq N\},
\]

\[
\mathcal{K} = \{(v_{i,j}, v_{i,j+1}); 1 \leq i \leq k, 1 \leq j \leq N-1\},
\]

\(k \geq 1\) and \(N \geq 1\). \(V\) represents the set of all nodes, \(\mathcal{I}\) and \(\mathcal{O}\) represent \(k\) input nodes and \(k\) output nodes, respectively. \(E\) represents the set of all edges and \(\mathcal{S}\) and \(\mathcal{K}\) represent the sets of vertical and horizontal edges, respectively.

Next we define the resource state corresponding to the \((k, N)\)-cluster network. We introduce qubits \(S_{i,j}^1\) at node \(v_{i,j}\) and \(S_{i+1,j}^2\) at node \(v_{i+1,j}\) to represent a Bell pair corresponding to an edge \((v_{i,j}, v_{i+1,j}) \in \mathcal{S}\). Similarly, we introduce qubits specified by \(K_{i,j}^1\) at node \(v_{i,j}\) and \(K_{i,j+1}^2\) at node \(v_{i,j+1}\) to represent a Bell pair corresponding to an edge \((v_{i,j}, v_{i,j+1}) \in \mathcal{K}\). We denote the set of all qubits in the resource state by \(R = \{S_{i,j}^1, S_{i+1,j}^2|1 \leq i \leq k-1, 1 \leq j \leq N\} \cup \{K_{i,j}^1, K_{i,j+1}^2|1 \leq i \leq k, 1 \leq j \leq N-1\}\). The resource state \(|\Phi\rangle_R\) corresponding to a cluster network is defined by the following.
Definition 2. For a given \((k, N)\)-cluster network, the resource state \(|\Phi\rangle_{\mathcal{R}} \in \mathcal{H}_{\mathcal{R}}\) is defined by

\[
|\Phi\rangle_{\mathcal{R}} = \bigotimes_{i=1}^{k-1} \bigotimes_{j=1}^{N} |\Phi^+\rangle_{S_{i,j},S_{i+1,j}} \otimes |\Phi^+\rangle_{K_{i,j},K_{i+1,j}},
\]

For example, the \((3,3)\)-cluster network and the corresponding resource state are shown in Fig. 3. Note that the resource state for a cluster network represented by Eq. (4) is different from the cluster states used in measurement based quantum computation [10]. While we can convert the resource state for a cluster network into a cluster state by applying a projective measurement on all qubits at each node, a cluster state cannot be converted to the resource state for the corresponding cluster network by LOCC. The resource state for a cluster network is equivalent to a valence bond solid state [11] introduced in condensed matter physics.

Finally we define the implementability of a unitary operation over a \(k\)-pair network. In addition to resource qubits \(\mathcal{R}\), we introduce input qubits \(I_1\) at the input node \(v_{i,1}\) and output qubits \(O_1\) at the output node \(v_{i,N}\), a set of input qubits \(I_Q = \{I\}_{1 \leq i \leq k}\) and a set of output qubits \(O_Q = \{O\}_{1 \leq i \leq k}\) for a \((k, N)\)-cluster network. Note that each input and output node has only one input or output qubit since we concentrate on the implementability of a unitary operation, and the state of output qubits is initially set to be in \(|0\rangle \in \mathcal{H}_{\mathcal{O}_Q}\).

**Definition 3.** For a \((k, N)\)-cluster network specified by \(G = (\mathcal{V}, \mathcal{E}, I, O)\), a unitary operation \(U \in \mathcal{U}(\mathcal{H}_{\mathcal{Q}} : \mathcal{H}_{\mathcal{O}_Q})\) is deterministically implementable over the network if and only if there exists a LOCC map \(\Gamma\) such that for any pure state \(|\psi\rangle \in \mathcal{H}_{\mathcal{Q}}\),

\[
\Gamma(|\psi\rangle \otimes |\Phi\rangle_{\mathcal{R}}) = U|\psi\rangle|\psi\rangle^{\dagger},
\]

where LOCC map \(\Gamma\) consists of local operations on each node and classical communications and \(\mathcal{U}(\mathcal{H}_{\mathcal{Q}} : \mathcal{H}_{\mathcal{O}_Q})\) is the set of unitary operations from \(\mathcal{H}_{\mathcal{Q}}\) to \(\mathcal{H}_{\mathcal{O}_Q}\).

Note that the main difference between this network computation model for implementing a unitary operation over a cluster network and standard measurement based quantum computation is that any operations inside each node are allowed including adding arbitrary ancilla states in this model whereas only projective measurements on the cluster state in each node are allowed in measurement based quantum computation. Thus the full set of implementable unitary operations over a \((k, N)\)-cluster network is larger than a set of operations implementable by measurement based quantum computation using the corresponding cluster states converted from the resource state for the \((k, N)\)-cluster network by LOCC.

III. CONVERSION OF A CLUSTER NETWORK INTO QUANTUM CIRCUITS

We propose a method to convert a \((k, N)\)-cluster network into quantum circuits representing a class of unitary operations implementable by LOCC assisted by the resource state corresponding to a given cluster network. By using the converted circuit, it is easier to construct a network coding protocol since a set of implementable unitary operations are represented by a set of parameters of the converted circuit instead of a complicated LOCC protocol. The class of implementable unitary operations represented by the converted circuit is a subset of that over the cluster network in general since the conversion method does not guarantee to give all possible constructions. However, in some cases, the constructions given by the conversion methods cover all possible implementable unitary operations as will be shown in Section V.

We define a set of vertically aligned nodes \(V_{ij}^v := \{v_{i,j}\}_{i=1}^{k}\) and a set of vertically aligned edges \(S_j := \{(v_{i,j}, v_{i,j+1})\}_{i=1}^{k}\) where \(1 \leq j \leq N\). We also define a set of horizontally aligned nodes \(V_{ij}^h := \{v_{i,j}\}_{j=1}^{N}\) and a set of horizontally aligned edges \(K_i := \{(v_{i,j}, v_{i,j+1})\}_{j=1}^{N}\) where \(1 \leq i \leq k\). We consider that the Bell pairs given for a set of vertically aligned edges \(S_j\) are used for implementing global unitary operations between nodes whereas each Bell pair given for a set of horizontal aligned edges \(K_i\) is used for teleporting a qubit state from node \(v_{i,j}\) to node \(v_{i,j+1}\).

We investigate which kinds of global operations are implementable between the nodes in \(V_{ij}^v\) if only one Bell pair is given for each edge and LOCC between the nodes is allowed. In this case, a two-qubit controlled unitary operation

\[
C_{t,\alpha}(|\alpha\rangle_{a=0,1} := \sum_{\alpha=0}^{1} |\alpha\rangle \langle \alpha| \otimes u_{\alpha}^{(a)}\)
where $l$ represents the vertical coordinate of the node $v_{l,j}$ of a control qubit and $n$ represents the vertical coordinate of the node $v_{n,j}$ of a target qubit, and $u_n^{(a)} (a = 0, 1)$ are single qubit unitary operations on the target qubit, can be performed by using the method to implement a controlled unitary operation using a Bell pair proposed by [12]. If $n \neq l \pm 1$, all Bell pairs represented by edges between $l$ and $n$ are consumed to implement the two-qubit control unitary. When we do not specify the single qubit unitary operations $\{u_n^{(a)}\}$ on the target qubit we denote a two-qubit controlled unitary operation simply by $C_{l,m,n}$.

In addition to the two-qubit control unitary operations, we can perform three-qubit fully controlled unitary operations defined by

$$C_{l,m,n}(\{u_n^{(ab)}\}_{a,b=0,1}) := \sum_{a=0}^{1} \sum_{b=0}^{1} \langle ab | l \rangle \otimes u_n^{(ab)} \ .$$

where $l$ and $m$ represent the vertical coordinates of the nodes $v_{l,j}$ and $v_{m,j}$ of two control qubits, respectively, and $n$ represents the vertical coordinate of the node $v_{n,j}$ of a target qubit, and $u_n^{(ab)} (a,b = 0,1)$ represents single qubit operations on the target qubit. (See Appendix A for details of the LOCC protocol implementing three-qubit fully controlled unitary operations.) Note that the indices $l$, $m$ and $n$ should be taken such that $l < n < m$ or $m < n < l$. Similarly to the case of a two-qubit controlled unitary operation, we denote a three-qubit fully controlled unitary operation by $C_{l,m,n}$ when we do not specify the target single qubit operations. On the other hand, any four-qubit fully controlled unitary, where three of the four qubits are control qubits and the rest of the qubit is a target qubit, is not implementable on qubits that are all in different nodes of $V_j^n$ in a $(k,N)$-cluster network, if a single Bell pair is given for each edge in $S_j$. Obviously any single qubit unitary operations can be implemented on any qubit.

We present a protocol to convert a given $(k,N)$-cluster network into quantum circuits. First (step 1 to step 3), we construct quantum circuits of unitary operations that are implementable on qubits in a set of vertically aligned nodes $V_j^n$ by LOCC assisted by the Bell pairs given for a set of vertically aligned edges $S_j$ for a certain $j$. Then (step 4), we repeat the procedure given by the first part (step 1 to step 3) for different $j$ of $1 \leq j \leq N$.

The conversion protocol:

1) Draw $k$ horizontal wire segments where each of the wire segments corresponds to a set of qubits at vertically aligned nodes $V_j^n$.

2) Symbols representing two-qubit controlled unitary operations $C_{l,n}$ or three-qubit fully controlled unitary operations $C_{l,m,n}$ are added on the horizontal wire segments according to the following rules.

- To represent $C_{l,n}$, draw a black dot representing a control qubit on the $l$-th wire, draw a vertical segment from the dot to the $n$-th wire segment and draw a box representing a target unitary operation on the $n$-th wire segment at the end of the vertical segment. Write index $l$ at the side of the vertical segment between the horizontal wire segments. An example is shown in Fig. 4 i).

- To represent $C_{l,m,n}$, draw two black dots representing control qubits on the $l$-th and $m$-th wire segments, draw vertical segments from each dot to the $n$-th wire and draw a box representing an arbitrary target unitary operation on the $n$-th wire segment at the end of the vertical segment. Write indices $l$ and $m$ at the sides of the vertical segment between the horizontal wire segments. An example is shown in Fig. 4 ii).

- Multiple gates of $C_{l,n}$ or $C_{l,m,n}$ can be added as long as there are only one type of index appearing between the horizontal wire segments and no target unitary operation represented by a box is inserted between two black dots on a horizontal wire segment. An example of possible circuits generated in this protocol is shown in Fig. 4 iii). We also give an example of circuits that do not follow the rule in Fig. 4 iv).

3) Arbitrary single qubit unitary operations represented by boxes are inserted between before and after the sequence of $C_{l,n}$ and $C_{l,m,n}$ (but not during the sequence) obtained by step 2.

4) Repeat step 1 to step 3 for each $1 \leq j \leq N$ and connect all the $i$-th horizontal wire segments.

In Appendix B, we show that a unitary operation represented by the quantum circuit obtained by step 1 to step 3 of the conversion protocol is implementable in a set of vertically aligned nodes $V_j^n$, namely, it is implementable by LOCC assisted by $(k-1)$ Bell pairs corresponding to a set of vertically aligned edges $S_j$. As examples, quantum circuits converted from the $(2,3)$-cluster and $(3,2)$-cluster networks are shown in Fig. 5.

Our conversion method generates infinitely many quantum circuits in general. However for special cluster networks, standard forms of quantum circuits can be obtained. In Appendix C, we show that any converted circuit obtained from a $(2,3)$-cluster network can be simulated by the circuit presented in Fig. 5 i), and any converted circuit obtained from a $(3,2)$-cluster network can be simulated by the circuit presented in Fig. 5 ii).
IV. IMPLEMENTABILITY OF UNITARY OPERATIONS OVER THE BUTTERFLY AND GRAIL NETWORKS

For classical network coding, it has been shown that there exists a network coding protocol over a 2-pair network, which has two input nodes and two output nodes, if and only if the network has at least one of the butterfly, train and identity substructures [13], [14]. Thus any classical network coding protocol over a 2-pair network can be reduced into a combination of protocols over the butterfly, train or identity networks, and these networks are fundamental primitive networks for classical network coding. As a first step to investigate the implementability of quantum computation over general 2-pair quantum networks, we investigate implementability of two-qubit unitary operations over the butterfly and train networks in this section by using the method converting a \((k, N)\)-cluster network into quantum networks introduced in the previous section.

A general two-qubit unitary operation \(U \in \mathcal{U}(H_{\mathcal{I}_2} : \mathcal{H}_{\mathcal{O}_2})\) where \(H_{\mathcal{I}_2} = \mathcal{H}_{\mathcal{O}_2} = \mathbb{C}^2 \otimes \mathbb{C}^2\) can be decomposed into a canonical form (the Kraus-Cirac decomposition) introduced in [15], [16], [17] given by

\[
U = (u \otimes u')e^{i(xX \otimes Y + yY \otimes Y + zZ \otimes Z)}(w \otimes w'). \tag{8}
\]

where \(u, u', w, w'\) are single-qubit unitary operations and \(X, Y, Z\) are the Pauli operators on \(\mathbb{C}^2\) and \(x, y, z \in \mathbb{R}\). Since it is trivial that the single-qubit unitary operations \(u'\) and \(u''\) are implementable at the input nodes and \(w\) and \(w'\) are implementable at the output nodes, we just need to analyze the implementability of the two-qubit global unitary part

\[
U_{\text{global}}(x, y, z) := e^{i(xX \otimes X + yY \otimes Y + zZ \otimes Z)} \tag{9}
\]

over the butterfly and train networks. In Eq. (9), the parameters \(x, y, z\) in \(0 \leq x < \pi/2\) (or \(0 \leq x < \pi/4\) if \(z = 0\)), \(0 \leq y \leq \min\{x, \pi/2 - x\}\) and \(0 \leq z \leq y\) cover all two-qubit global unitary operations up to the local unitarily equivalence (the Weyl chamber [16]). We define the Kraus-Cirac number of a two-qubit unitary operation \(U\) as the number of non-zero parameters \(x, y, z\) in \(U_{\text{global}}(x, y, z)\) and denote by \(\text{KC}\#(U)\). \(\text{KC}\#(U)\) characterizes nonlocal properties (globalness) of \(U\) [13]. \(U\) with \(\text{KC}\#(U) = 0\) is a product of local unitary operations, \(U\) with \(\text{KC}\#(U) = 1\) is locally unitarily equivalent to a controlled unitary operation, \(U\) with \(\text{KC}\#(U) = 2\) is locally unitarily equivalent to a special class of two-qubit unitary operations called a matchgate [19], [20], [21]. The rest of two-qubit unitary operations including \(\text{SWAP}\) operations have \(\text{KC}\#(U) = 3\).

By constructing a protocol for implementing \(U_{\text{global}}(x, y, z)\) for arbitrary \(x, y, z\), we obtain the following theorem.

**Theorem 1.** Any two-qubit unitary operation is deterministically implementable over the butterfly network.

**Proof.** For the implementability of \(U_{\text{global}}(x, y, z)\) over the butterfly network represented by the left hand side of Fig. 6 we consider a \((3, 2)\)-cluster network represented by the right hand side of Fig. 6 by assigning the nodes \(\{i_1, i_2, o_1, o_2\}\) of the butterfly network to the nodes \(\{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}\) of the \((3, 2)\)-cluster network, respectively. In this assignment, the correspondence of the edges of the butterfly network and the horizontal and vertical sets of edges \(K_1, S_1, S_2\) of the \((3, 2)\)-cluster network is given...
by

\[
\{ E_1, E_5, E_3 \} \leftrightarrow K_1, \\
\{ E_2, E_4 \} \leftrightarrow S_1, \\
\{ E_6, E_7 \} \leftrightarrow S_2. \quad (10)
\]

Thus any two-qubit unitary operation is deterministically implementable over the butterfly network if any \(U_{global}(x, y, z)\) in the form of Eq. (9) is deterministically implementable over the \((3, 2)\)-cluster network where input states are given at nodes \(v_{1.1}\) and \(v_{3.1}\) and output states are obtained at nodes \(v_{1.2}\) and \(v_{3.2}\), since the topology of the butterfly network is the same as that of the \((3, 2)\)-cluster network.

We construct a protocol implementing two-qubit unitary \(U_{global}(x, y, z)\) by setting a fixed input state at node \(v_{2.1}\) and arbitrary two-qubit input state at nodes \(v_{1.1}\) and \(v_{3.1}\) as a three-qubit input state at input nodes \(I = \{v_{1.1}, v_{2.1}, v_{3.1}\}\) and implementing a three-qubit unitary operation denoted by \(U_3\) over the \((3, 2)\)-cluster network followed by an LOCC map denoted by \(\Gamma\) performed at output nodes \(O = \{v_{1.2}, v_{2.2}, v_{3.2}\}\). Recall that a unitary operation of represented by the quantum circuit shown in Fig. 5 ii) is implementable over the \((3, 2)\)-cluster network. That is, two three-qubit fully controlled unitary operations \(C_{1,3,2}\) are implementable, one at nodes \(I\) and another at nodes \(O\). The following protocol shows that by choosing appropriate parameters for one of the three-qubit fully controlled unitary operations and one of the single-qubit local unitary operations in \(U_3\), we can implement \(U_{global}(x, y, z)\) with arbitrary \(x, y, z\).

The protocol for implementing \(U_{global}(x, y, z)\):

1) An arbitrary two-qubit input state \(\rho\) is given for qubits at input nodes \(v_{1.1}\) and \(v_{3.1}\) and a fixed input state \(|0\rangle\) is prepared for the qubit at node \(v_{2.1}\).

2) Implement \(U_3\) of which the quantum circuit representation is given by the left shaded part of Fig. 7 over the \((3, 2)\)-cluster network.

a) All single-qubit unitary operations appearing in the circuit representation of \(U_3\) are trivially performed at each node.

b) The first fully controlled unitary operation implemented at input nodes \(I\) using the Bell pairs represented by vertical edges \(S_1\) is given by \(C_{1,3,2}(u_{ab})_{a,b=0,1}\) where \(u_{00} = u_{11} = I\) and \(u_{01} = u_{10} = Z\).

c) To transmit a qubit state from input nodes \(v_{1.i}\) to output node \(v_{2.i}\) for \(i = 1, 2, 3\), quantum teleportation is performed for each \(i\) by using the Bell pair represented by a horizontal edge in \(K_1\).

d) The second fully controlled unitary operation implemented at output nodes \(O\) contains parameters \(y\) and \(z\) and is given by \(C'_{1,3,2}(u_{ab})_{a,b=0,1}\) where

\[
\begin{align*}
\begin{cases}
\begin{align*}
u_{00} & = u_{11} = e^{i(z+y)}|0\rangle\langle 0| - ie^{-i(z+y)}|1\rangle\langle 1|, \\
\nu_{01} & = u_{10} = e^{-i(z+y)}|0\rangle\langle 0| - ie^{i(z+y)}|1\rangle\langle 1|.
\end{align*}
\end{cases}
\end{align*}
\]

e) After implementing \(C'_{1,3,2}(u_{ab})_{a,b=0,1}\), a single-qubit unitary operation parameterized by \(x\) given by

\[
u(x) = \frac{1}{\sqrt{2}} \begin{pmatrix}
e^{ix} & -ie^{-ix} \\
e^{ix} & ie^{-ix}
\end{pmatrix} \quad (11)
\]
is performed at node \(v_{2.2}\) in \(O\).

3) Perform an LOCC map \(\Gamma\) at output nodes \(O\) of which the quantum circuit representation is given by the right shaded part of Fig. 7. The map \(\Gamma\) consists of the following three steps.

a) Perform a projective measurement on the qubit at node \(v_{2.2}\) in the computational basis \(|0\rangle\langle 0|, |1\rangle\langle 1|\).

b) Classically communicate the measurement outcome \(k \in \{0, 1\}\) from node \(v_{2.2}\) to \(v_{1.2}\) and also to \(v_{3.2}\).

c) If \(k = 1\), perform a conditional operation \(X\) on output qubits at nodes \(v_{1.2}\) and \(v_{3.2}\) otherwise do nothing.

This protocol maps any input state \(\rho\) given at input nodes \(v_{1.1}\) and \(v_{3.1}\) to

\[
U_{global}(x, y, z)\rho U_{global}^\dagger(x, y, z) = \Gamma(U_3(\rho \otimes |0\rangle\langle 0|)U_3^\dagger) \quad (12)
\]
at output nodes \(v_{1.2}\) and \(v_{3.2}\) where \(|0\rangle\) represents the fixed input state at node \(v_{2.1}\). See Appendix F for details of calculations. It is straightforward to translate the protocol over the \((3, 2)\)-cluster network to a protocol to implement \(U_{global}(x, y, z)\) over the butterfly network by using the correspondence of vertices and edges. Thus, \(U_{global}(x, y, z)\) is deterministically implementable over the butterfly network.

For the implementability of \(U_{global}(x, y, z)\) over the grail network, we consider a \((2, 3)\)-cluster network by assigning the nodes \(\{n_1, n_2, o_1, v_2, n_3, n_4\}\) of the grail network to the
nodes \{v_{1,1}, v_{1,2}, v_{1,3}, v_{2,1}, v_{2,2}, v_{2,3}\} of the (2, 3)-cluster network, respectively (Fig. 8). The (2, 3)-cluster network can be converted to a quantum circuit containing three controlled-NOT gates and arbitrary single unitary operations that are inserted between the controlled-NOT gates. It is shown that any two-qubit unitary operations \(U_{\text{global}}(x, y, z)\) can be decomposed by three controlled-NOT gates and single unitary operations inserted between the controlled-NOT gates [22]. Thus any two-qubit controlled unitary operation is deterministically implementable over the grail network.

V. THE SET OF ALL IMPLEMENTABLE UNITARY OPERATIONS FOR \(k = 2, 3\)

In this section, we derive the condition for \(k\)-qubit unitary operations to be implementable over a given cluster network. We show that our conversion method presented in Section III gives all implementable unitary operations over the \((k, N)\)-cluster network for \(k = 2, 3\).

Theorem 2. If i) a \(k\)-qubit unitary operation \(U\) is deterministically implementable over the \((k, N)\)-cluster network \((k \geq 2, N \geq 1)\), then ii) the matrix representation of \(U\) in terms of the computational basis \(U^M\) can be decomposed into

\[
U^M = V_1^M V_2^M \cdots V_N^M ,
\]

where each \(V_i^M\) is a \(2^k\) by \(2^k\) unitary matrix such that

\[
V_i^M = \sum_{a_1=0}^{1} \sum_{a_2=0}^{1} \cdots \sum_{a_{k-1}=0}^{1} E_{1,1}^{(a_1)} \otimes E_{2,1}^{(a_1,a_2)} \otimes E_{3,1}^{(a_2,a_3)} \otimes \cdots \otimes E_{k-1,1}^{(a_{k-2},a_{k-1})} \otimes E_{k,k}^{(a_{k-1})} ,
\]

where \(E_{i,j}^{(m,n)}\) and \(E_{i,j}^{(m)}\) are 2 by 2 complex matrices.

To prove Theorem 2, we use Lemma 3 represented in Appendix D about a class of bipartite separable maps that preserves entanglement. A bipartite separable map \(\Gamma_{\text{sep}}\) is a completely positive and trace preserving (CPTP) map whose Kraus operators are product as follows:

\[
\Gamma_{\text{sep}}(\rho_{AB}) = \sum_k (A_k \otimes B_k)\rho_{AB}(A_k \otimes B_k)^\dagger ,
\]

where \(\sum_k (A_k \otimes B_k)^\dagger (A_k \otimes B_k) = I_A \otimes I_B\). Since quantum network coding is equivalent to perform LOCC assisted by the resource state in our setting, we have to analyze multiparticle LOCC. However, the analysis of multiparticle LOCC is extremely difficult. Thus, we analyze multiparticle separable maps, which are much easier to analyze than LOCC due to their simple structure. Note that a set of separable maps is exactly larger than that of LOCC [23].

Proof of Theorem 2. Denote by \(\mathcal{H}_{I_Q} = \otimes_{i=1}^k \mathcal{H}_i\) and \(\mathcal{H}_{Q_1} = \otimes_{i=1}^k \mathcal{H}_{O_i}\) the Hilbert spaces of \(k\) input qubits and \(k\) output qubits, respectively. By introducing another ancillary Hilbert space \(\mathcal{H}_I;\) at the input nodes \(v_{i,1}\), denote the Hilbert space of \(k\) qubits by \(\mathcal{H}_{I_Q} = \otimes_{i=1}^k \mathcal{H}_I\). A joint state of \(k\) copies of the Bell pairs in \(\mathcal{H}_{I_Q} \otimes \mathcal{H}_{I_Q}\) is denoted by

\[
\left| \right\rangle := \frac{1}{\sqrt{D}} \sum_i \left| i \right\rangle_{I_Q} \left| i \right\rangle_{I_Q}' = \otimes_{i=1}^k |\Phi^+\rangle_{I_i,I_i}' ,
\]

where \(D = \dim(\mathcal{H}_{I_Q}) = 2^k\). If \(U \in \mathcal{U}(\mathcal{H}_{I_Q} : \mathcal{H}_{O_Q})\) is deterministically implementable over a \((k, N)\)-cluster network for \(k \geq 2\) and \(N \geq 1\), and we consider applying \(U\) on \(\left| \right\rangle\).

That is, there exists a LOCC map \(\Gamma\) such that

\[
\frac{1}{D} \sum_{i,j} \Gamma(i) \langle j |_{I_Q} \otimes |\Phi\rangle_{\mathcal{H}_R} \otimes |i \rangle \langle j |_{I_Q}' = |U\rangle\langle U| ,
\]

where \(|\Phi\rangle_{\mathcal{H}_R}\) is the resource state of the \((k, N)\)-cluster network and \(|U\rangle\) is defined by

\[
|U\rangle := (U \otimes I) \left| \right\rangle \in \mathcal{H}_{O_Q} \otimes \mathcal{H}_{I_Q} .
\]

By defining a map represented by the left hand side of Eq. (16) as \( \Gamma'(|\Phi\rangle_{\mathcal{H}_R}) := N \sum_{i,j} \Gamma(i) (j |_{I_Q} \otimes |\Phi\rangle_{\mathcal{H}_R} \otimes |i \rangle \langle j |_{I_Q}'\rangle\), where \(\Gamma'\) is also a LOCC map if we assume two qubits belonging to \(\mathcal{H}_I\) and \(\mathcal{H}_I'\) are in the same input node for all \(i\). Since any LOCC maps are separable maps, there exists a separable map \(\Gamma_{\text{sep}}'\) satisfying

\[
\Gamma_{\text{sep}}'(|\Phi\rangle_{\mathcal{H}_R}) = |U\rangle\langle U| ,
\]

if \(U\) is deterministically implementable over a \((k, N)\)-cluster network. Since \(\Gamma_{\text{sep}}'\) is a map from a pure state to a pure state, the action of \(\Gamma_{\text{sep}}'\) represented by Eq. (18) can be equivalently given by the existence of a set of linear operators (the Kraus operators) \(\{A_{i,j}^{m}\}_{m}\) for each node \(v_{i,j}\) and a probability distribution \(\{p_m\}\) such that

\[
\forall m; \otimes_{i=1}^k \otimes_{j=1}^N A_{i,j}^{m} |\Phi\rangle_{\mathcal{H}_R} = \sqrt{p_m} |U\rangle ,
\]

\[
\sum_{m} \otimes_{i=1}^k \otimes_{j=1}^N (A_{i,j}^{m})^\dagger A_{i,j}^{m} = I ,
\]
where

\[
\begin{align*}
A_{1,1}^m & \in L(\mathcal{H}S_{1,1}^m \otimes \mathcal{H}K_{1,1}^i : \mathcal{H}I_i), \\
A_{1}^m & \in L(\mathcal{H}S_{1,1}^m \otimes \mathcal{H}S_{2,1}^m \otimes \mathcal{H}K_{1,1}^i : \mathcal{H}I_i), \\
A_{2,1}^m & \in L(\mathcal{H}S_{2,1}^m \otimes \mathcal{H}K_{1,1}^i : \mathcal{H}I_i), \\
A_{ij}^m & \in L(\mathcal{H}S_{i,j}^m \otimes \mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{2,1}^j : \mathcal{C}) \quad (2 \leq j \leq N - 1), \\
A_{i,j}^m & \in L(\mathcal{H}S_{i,j}^m \otimes \mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{2,1}^j : \mathcal{C}) \quad (2 \leq i \leq k - 1), \\
A_{k,1}^m & \in L(\mathcal{H}S_{k,1}^m \otimes \mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{2,1}^j : \mathcal{C}) \\
A_{k,N}^m & \in L(\mathcal{H}S_{k,N}^m \otimes \mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{2,1}^j : \mathcal{C}),
\end{align*}
\]

and \(L(\mathcal{H} : \mathcal{H}')\) is the set of linear operators from \(\mathcal{H}\) to \(\mathcal{H}'\).

First, letting \(E_m = \otimes_{i=1}^k A_{ii}^m, F_m = \otimes_{i=1}^k \otimes_{j=2}^N A_{ij}^m\) and applying Lemma 3 presented in Appendix D, we obtain for all \(\{m|p_m \neq 0\}\),

\[\exists a_{1,m}, \exists V_{1,m}^M \in U(\mathbb{C}^D); \quad E_m^M = a_{1,m}V_{1,m}^M, \quad (22)\]

where \(U(\mathbb{C}^D)\) is the set of \(D\) by \(D\) unitary matrices and \(E_m^M\) is a \(D\) by \(D\) matrix satisfying

\[
(E_m^M)_{a,b} = \langle a|z_i^a \otimes z_i^b A_{ii}^m|b\rangle = \langle a|z_i^a \otimes z_i^b A_{ii}^m|b\rangle, \quad (a,b \in \mathbb{C})
\]

and

\[
|A_b\rangle S_{i,1}^m K_{1,1}^i = \otimes_{i=1}^{k-1} |\Phi^+\rangle S_{i,1}^m S_{i+1,1}^m \otimes |b\rangle K_{1,1}^i \cdots K_{1,1}^j.
\]

Let

\[
\begin{align*}
A_{1,1}^m &= \sum_{a_1=0}^1 \langle a_1|s_{1,1}^m \otimes E_{1,1}^{(a_1)},m \rangle, \\
A_{i,1}^m &= \sum_{a_2=0}^1 \sum_{a_1=0}^1 \langle a_1|s_{i,1}^m \otimes a_2|s_{i,1}^m \otimes E_{i,1}^{(a_1,a_2)},m \rangle \quad (2 \leq i \leq k - 1), \\
A_{k,1}^m &= \sum_{a_1=0}^1 \langle a_1|s_{k,1}^m \otimes E_{k,1}^{(a_1)},m \rangle.
\end{align*}
\]

where \(E_{1,1}^{(a_1)},m \in L(\mathcal{H}K_{1,1}^i : \mathcal{H}I_i), E_{i,1}^{(a_1,a_2)},m \in L(\mathcal{H}K_{1,1}^i : \mathcal{H}I_i)\) and \(E_{k,1}^{(a_1)},m \in L(\mathcal{H}K_{1,1}^i : \mathcal{H}I_i)\). Thus, \(V_{1,m}\) can be decomposed into

\[
V_{1,m} = \sum_{a_1, \ldots, a_{k-1}=0}^1 E_{1,1}^{(a_1)},m \otimes E_{2,1}^{(a_1,a_2)},m \otimes \cdots \otimes E_{k-1,1}^{(a_2, \ldots, a_{k-1})},m \otimes E_{k,1}^{(a_{k-1})},m.
\]

Note that we identify a linear operation and its matrix representation in the computational basis, e.g., \(E_{1,1}^{(a_1)},m\) is a 2 by 2 complex matrix.

Next, letting \(E_m = \otimes_{i=1}^{k-1} \otimes_{j=2}^N A_{ij}^m, F_m = \otimes_{i=1}^k \otimes_{j=3}^N A_{ij}^m\) and using Lemma 4 represented in Appendix D, we obtain for all \(\{m|p_m \neq 0\}\),

\[\exists a_{2,m}, \exists V_{2,m}^M \in U(\mathbb{C}^D); \quad E_m^M = a_{2,m}V_{2,m}^M, \quad (27)\]

where \(E_m^M\) is a \(D \times D\) matrix such that

\[
(E_m^M)_{a,b} = \langle a|z_i^a \otimes z_i^b A_{ij}^m|b\rangle = \langle a|z_i^a \otimes z_i^b A_{ij}^m|b\rangle, \quad (28)
\]

and

\[
|A_b\rangle S_{i,1}^m S_{i+1,1}^m \Phi^+ = \otimes_{i=1}^{k-1} \otimes_{j=2}^N A_{ij}^m, \quad (29)
\]

Let

\[
\begin{align*}
A_{1,1}^m &= \sum_{a_1=0}^1 \langle a_1|s_{1,1}^m \otimes E_{1,1}^{(a_1)},m \rangle, \\
A_{i,1}^m &= \sum_{a_2=0}^1 \sum_{a_1=0}^1 \langle a_1|s_{i,1}^m \otimes a_2|s_{i,1}^m \otimes E_{i,1}^{(a_1,a_2)},m \rangle \quad (2 \leq i \leq k - 1), \\
A_{k,1}^m &= \sum_{a_1=0}^1 \langle a_1|s_{k,1}^m \otimes E_{k,1}^{(a_1)},m \rangle,
\end{align*}
\]

where \(E_{1,1}^{(a_1)},m \in L(\mathcal{H}K_{1,1}^i : \mathcal{H}I_i), E_{i,1}^{(a_1,a_2)},m \in L(\mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{1,1}^j : \mathcal{C})\) and \(E_{k,1}^{(a_1)},m \in L(\mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{1,1}^j : \mathcal{C})\).

By straightforward calculation, \(V_{2,m}\) are shown to be decomposed into

\[
V_{2,m} = \sum_{a_1, \ldots, a_{k-1}=0}^1 E_{i,1}^{(a_1,a_2),m} \otimes \cdots \otimes E_{k,1}^{(a_{k-2},a_{k-1}),m} \otimes E_{k,1}^{(a_{k-1}),m}, \quad (30)
\]

where \(E_{i,1}^{(a_1,a_2),m} \in L(\mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{1,1}^j : \mathcal{C}), E_{k,1}^{(a_{k-2},a_{k-1}),m} \in L(\mathcal{H}K_{1,1}^i \otimes \mathcal{H}K_{1,1}^j : \mathcal{C})\).

Iterating this procedure, we obtain for all \(\{m|p_m \neq 0\}\),

\[\exists a > 0, \exists W^M \in U(\mathbb{C}^D); \quad F_m = aW^M, F_m = \frac{\sqrt{p_m}}{a} W^M, \quad (32)\]

where \(W^M\) and \(\overline{W^M}\) can be decomposed into

\[
W^M = V_{1,1}^MW_{1,1} \cdots V_{N-1,1}^MW_{N-1,1}
\]

and

\[
\overline{W^M} = U^MW_{N,1}^MV_{N,1}^M \cdots U^MW_{1,1}^MV_{1,1}^M
\]

In the case of the \((2,N)\)-cluster networks, which we call \(N\)-bridge ladder networks, \(V_i\) is locally unitarily equivalent to the two-qubit controlled unitary operation since its operator Schmidt rank is 2 [23]. Thus, statements i) and ii) in Theorem 2 are equivalent since a sequence of \(N\) two-qubit controlled
unitary operations is implementable by the converted circuit presented in Fig. 5. Then we obtain the following theorem for the ladder networks.

**Theorem 3.** A unitary operation $U$ is deterministically implementable over the $N$-bridge ladder network if and only if $\text{KC#}(U) \leq N$, where $\text{KC#}(U)$ is the Kraus-Cirac number of a unitary operation $U \in SU(4)$, which is the number of non-zero parameters $x, y, z$ in Eq. (9) characterizing the global part of $SU(4)$.

This theorem is proven by using the following lemma relating the Kraus-Cirac number of a two-qubit unitary operation and the decomposition of the unitary operation into controlled unitary operations shown in [18].

**Lemma 1.** Consider a set of two-qubit unitary operations $U_c$ that is locally unitarily equivalent to a controlled unitary operation. The decomposition of a unitary operation $U \in SU(4)$ into a shortest sequence of two-qubit unitary operations in $U_c$ depends on the Kraus-Cirac number $\text{KC#}(U)$ of $U$ as

\[
\{U \in SU(4) | \text{KC#}(U) \leq 1\} = \{U | U \in U_c\},
\]

\[
\{U \in SU(4) | \text{KC#}(U) \leq 2\} = \{UV | U, V \in U_c\},
\]

\[
\{U \in SU(4) | \text{KC#}(U) \leq 3\} = \{UVW | U, V, W \in U_c\}.
\]

**Proof of Theorem 3.** Since $\text{KC#}(U)$ is less than or equal to $N$ if and only if $U$ can be decomposed into $N$ two-qubit controlled unitary operations as shown in Lemma 1 and $N$ two-qubit controlled unitary operations are deterministically implementable over $N$-bridge ladder network, Theorem 3 is straightforwardly shown.

We also show that statements i) and ii) are equivalent in the case of the $(3, N)$-cluster networks in Appendix E.

**VI. Probabilistic Implementation of Unitary Operations**

In this section, we investigate the probabilistic implementation of unitary operations. There is no classical network coding protocol to send single bits from $v_{1,1}$ to $v_{2,2}$ and from $v_{2,1}$ to $v_{1,2}$ over a $(2, 2)$-cluster network since there is no butterfly, grail or identity substructure. This task corresponds to implementing a swap operator in quantum network coding. It is interesting to know whether there exists a task that is not achievable by classical network coding but a corresponding task is achievable in a quantum setting or not. We give a negative result in this section. Using Theorem 3, we see that a swap operator is not deterministically implementable over a $(2, 2)$-cluster network, which is a 2-bridge ladder network, since the Kraus-Cirac number of the swap operator is 3. Furthermore, we show that a swap operator is not implementable even probabilistically in this section.

**Theorem 4.** A $k$-qubit unitary operation $U$ is probabilistically implementable over the $(k, N)$-cluster network ($k \geq 2, N \geq 1$) if and only if the matrix representation of $U$ in terms of the computational basis $U_M$ can be decomposed into

\[
U_M = F_1^M F_2^M \cdots F_N^M,
\]

where each $F_i^M$ is a $2^k$ by $2^k$ complex matrix that can be decomposed in the same way as Eq. (14).

**Proof.** Similar to the case of deterministic implementation, we consider applying $U \in U(\mathcal{H}_{2^k} \otimes \mathcal{H}_{2^k})$ on a part of $k$ maximally entangled states $|i\rangle \in \mathcal{H}_{2^k} \otimes \mathcal{H}_{2^k}$. Then $U$ is probabilistically implementable over the $(k, N)$-cluster network ($k \geq 2, N \geq 1$) if and only if there exists a stochastic LOCC (SLOCC) map $\Gamma^M$ and non-zero probability $p > 0$ such that

\[
\Gamma^M(|\Phi\rangle\langle\Phi|_R) = p|U\rangle\langle U|,
\]

where $|\Phi\rangle$ is the resource state of the $(k, N)$-cluster network and $|U\rangle \in \mathcal{H}_{2^k} \otimes \mathcal{H}_{2^k}$ is defined by Eq. (17). Eq. (36) is equivalent to the statement that there exists a set of linear operators $\{A_{i,j}\}$ and non-zero probability $p > 0$ such that

\[
\otimes_{i=1}^k \otimes_{j=1}^N A_{i,j} |\Phi\rangle_R = \sqrt{p} |U\rangle.
\]

The conditions of $\{A_{i,j}\}$ given by Eq. (37) is similar to the conditions of Kraus operators $\{A_{i,j}\}_m$ given by Eq. (19) presented in the proof of Theorem 2. The index $m$ is dropped in Eq. (37) since the map we consider is SLOCC instead of LOCC considered in Theorem 2. By taking the correspondence between $A_{i,j}$ and $A_{i,j}^m$, we obtain a decomposition of the form presented in Eq. (35).

**Lemma 2.** A swap operation $U_{\text{swap}} = |00\rangle\langle 00| + |01\rangle\langle 10| + |10\rangle\langle 01| + |11\rangle\langle 11|$ is not implementable over the 2-bridge ladder network with non-zero probability.

**Proof.** By using Theorem 4, the swap operation is probabilistically implementable over the $(2, 2)$-cluster network (2-bridge ladder network) if and only if there exists a linear operation $X, Y \in L(\mathbb{C}^4, \mathbb{C}^4)$ and $E_{i,j} \in L(\mathbb{C}^2, \mathbb{C}^2)$ such that

\[
U_{\text{swap}} = XY,
\]

\[
X = E_{1,1}^{(0)} \otimes E_{2,1}^{(0)} + E_{1,1}^{(1)} \otimes E_{2,1}^{(1)}
\]

\[
Y = E_{1,2}^{(0)} \otimes E_{2,2}^{(0)} + E_{1,2}^{(1)} \otimes E_{2,2}^{(1)}.
\]

For any linear operations $M$, there exists the operator Schmidt decomposition, and we can define the operator Schmidt rank $\text{op#}(M)$, which is the number of non-zero coefficients of the operator Schmidt decomposition. Since $X$ and $Y$ can be decomposed into Eq. (38) and Eq. (40), we can derive

\[
\text{op#}(X) \leq 2,
\]

\[
\text{op#}(Y) \leq 2.
\]

Since $\text{op#}(U_{\text{swap}}) = 4$, $\text{op#}(X) = \text{op#}(Y) = 2$. In [25], it is shown that if $\text{op#}(X^2) = 2$ and $X$ is invertible, $\text{op#}(X^{-1}) = 2$. Thus, the swap operation is probabilistically implementable if and only if there exists linear operations $X, Y \in L(\mathbb{C}^4, \mathbb{C}^4)$ such that

\[
Y = U_{\text{swap}}X,
\]

\[
\text{op#}(X) = 2, \text{ rank}(X) = 4
\]

\[
\text{op#}(Y) = 2, \text{ rank}(Y) = 4.
\]
In general, we can regard $X$ as a matrix representation of a four qubit pure state $|\Phi\rangle_{1,2,3,4}$:

$$X = \sum_{i=1}^{4} |i\rangle_{1,2} |\Phi\rangle_{1,2,3,4} |i\rangle_{1,2}.$$  \hspace{1cm} (46)

Then, the following correspondences are obtained,

$$\text{rank}(X) = 4 \iff \text{Sch}^{3/4}_{1,2}(|\Phi\rangle) = 4, \hspace{1cm} (47)$$

$$\text{op}(X) = 2 \iff \text{Sch}^{3/2}_{1,3}(|\Phi\rangle) = 2, \hspace{1cm} (48)$$

$$\text{op}(U_{\text{swap}}X) = 2 \iff \text{Sch}^{3/2}_{1,4}(|\Phi\rangle) = 2, \hspace{1cm} (49)$$

where $\text{Sch}^{3/2}_{1,4}(|\Phi\rangle)$ is a Schmidt number in terms of a partition between qubit 1, 2 and qubit 3, 4. We show that there is no four qubit state simultaneously satisfying Eqs. (47), (48), and (49) in Appendix G.

**VII. CONCLUSION**

We have investigated the implementability of $k$-qubit unitary operations over the $(k, N)$-cluster networks to apply the idea of network coding for distributed quantum computation where the inputs and outputs of quantum computation are given in all separated nodes and quantum communication between nodes is restricted. We have presented a method to obtain quantum circuit representations of unitary operations implementable over a given cluster network. For the $(k, N)$-cluster networks of $k = 2, 3$, we have shown that our method provides all implementable unitary operations over the cluster network. As a first step to find the fundamental primitive networks of network coding for quantum settings, we have shown that both of the butterfly and grail networks are sufficient resources for implementing arbitrary two-qubit unitary operations, meanwhile the $(2, 2)$-cluster network is not sufficient to implement arbitrary two-qubit unitary operations even probabilistically.

**ACKNOWLEDGMENT**

This work is supported by the Project for Developing Innovation Systems of MEXT, Japan and JSPS by KAKENHI (Grant No. 23540463, No. 23240001, No. 26330006). We also gratefully acknowledge the ELC project (Grant-in-Aid for Scientific Research on Innovative Areas MEXT KAKENHI (Grant No. 24106009)) for encouraging this research.

**APPENDIX A**

**A LOCC PROTOCOL FOR IMPLEMENTING THREE-QUBIT FULLY CONTROLLED UNITARY OPERATIONS**

We show a construction of a protocol to implement a three-qubit fully controlled unitary operation $C_{l,m;n}$ on qubits located at a set of vertically aligned nodes $V^y_l$ over the $(k, N)$-cluster networks, where $l$ and $m$ represent two control qubits at nodes $v_{l,j}$ and $v_{m,j}$ respectively, and $n$ represents a target qubit at node $v_{n,j}$. We present a LOCC protocol to implement $C_{l,m;n}$ assisted by the resource states consisting of the Bell pairs corresponding to the vertical edges $S_j$ of the $(k, N)$-cluster networks.

We consider to implement $C_{l,m;n}$ on a state of three qubits indexed by $Q_l$, $Q_m$, and $Q_n$ at node $v_{l,j}$, $v_{m,j}$ and $v_{n,j}$, respectively, and its explicit form is given by

$$C_{l,m;n} = \sum_{a,b} \lambda_{ab} |ab\rangle_{lm} \otimes u_n^{ab} |\phi\rangle_n.$$  \hspace{1cm} (50)

where $\{|ab\rangle\}_{a,b=\{0,1\}}$ is the two-qubit computational basis of $H_{Q_l} \otimes H_{Q_m}$ of the two controlled qubits and $u_n^{ab}$ acts on $H_{Q_n}$ of the target qubit.

To show how our LOCC protocol works, we consider an arbitrary state of the control qubits by $\sum_{a,b} \lambda_{ab} |ab\rangle_{lm} \in H_{Q_l} \otimes H_{Q_m}$ where $\{|ab\rangle\}$ is a set of arbitrary complex coefficients satisfying the normalization condition $\sum_{a,b} |\lambda_{ab}|^2 = 1$ and we represent an arbitrary state of the target qubit by $|\phi\rangle \in H_{Q_n}$.

In the following, we show that our protocol transforms the joint state of controlled qubits and a target qubit to

$$C_{l,m;n} \sum_{a,b} \lambda_{ab} |ab\rangle_{lm} |\phi\rangle_n = \sum_{a,b} \lambda_{ab} |ab\rangle_{lm} u_n^{ab} |\phi\rangle_n.$$  \hspace{1cm} (51)

1) Ancillary qubits indexed by $Q'_l$, $Q'_m$, are introduced at nodes $v_{l,j}$ and $v_{m,j}$ respectively. Set both of the ancillary qubits to be in $|0\rangle$. Each of the two states of control qubits $Q_l$ and $Q_m$ is transformed to a two-qubit state by applying a CNOT gate on the control qubit and the ancillary qubit at the same node, namely applying CNOT on $Q_l'$ and $Q_m'$ and also $Q_m$ and $Q_m'$. Then the joint state of five qubits $Q_l$, $Q'_l$, $Q_m$, $Q_m'$ and $Q_n$ is given by

$$\sum_{a,b} \lambda_{ab} |ab\rangle_{lm}^{Q_l'} |ab\rangle_{lm}^{Q_m'} |\phi\rangle_n.$$  \hspace{1cm} (52)

2) By consuming the Bell pairs corresponding to the vertical edges $S_j$ between $v_{l,j}$ and $v_{n,j}$ and also between $v_{m,j}$ and $v_{n,j}$, perform quantum teleportation to transmit the states of qubits $Q'_l$ and $Q'_m$ from nodes $v_{l,j}$ and $v_{m,j}$ to $v_{n,j}$. A circuit representation of the protocol of quantum teleportation represented by $T$ in Fig. [9] is given by Fig. [10]. We denote indices of two qubits at node $v_{n,j}$ representing the teleported states from $Q'_l$ and $Q'_m$ by $Q'_l''$ and $Q'_m''$, respectively.

3) At node $v_{n,j}$, perform $C_{l,m;n}$ on $H_{Q'_l''} \otimes H_{Q'_m''} \otimes H_{Q_n}$. Then we obtain the state given by

$$\sum_{a,b} \lambda_{ab} |ab\rangle_{lm}^{Q'_l''} |ab\rangle_{lm}^{Q'_m''} u_n^{ab} |\phi\rangle_n.$$  \hspace{1cm} (53)
In Appendix A, we have shown a protocol to implement a three-qubit fully controlled unitary operation \( C_{l,m,n} \), where qubits in the first shaded region are at the node \( v_{l,j} \), those in the second shaded region are at the node \( v_{m,j} \), and those in the third shaded region are at the node \( v_{n,j} \). The protocol consists of introducing ancillary qubits \( Q_{l'} \) and \( Q_{m'} \) at the nodes \( v_{l,j} \) and \( v_{m,j} \), respectively, teleporting ancillary qubit states from the nodes \( v_{l,j} \) and \( v_{m,j} \) to the node \( v_{n,j} \) represented by qubits \( Q_{l''} \) and \( Q_{m''} \), applying \( C_{l,m,n} \) on controlled qubits \( Q_{l''} \) and \( Q_{m''} \) and a target qubit \( Q_n \) at the node \( v_{n,j} \), performing Hadamard operations and measurements in the computational basis on \( Q_{l''} \) and \( Q_{m''} \) at node \( v_{n,j} \) and finally applying conditional Z operations depending on the measurement outcome on two control qubits \( Q_l \), \( Q_m \) at nodes \( v_{l,j} \) and \( v_{m,j} \), respectively.

4) At node \( v_{n,j} \), we apply the Hadamard operations and perform projective measurements in the computational basis on both \( H_{Q_{l''}} \) and \( H_{Q_{m''}} \). The measurement outcomes of qubits \( Q_{l''} \) and \( Q_{m''} \) are sent to nodes \( v_{l,j} \) and \( v_{m,j} \), respectively, by classical communication. At each of nodes \( v_{l,j} \) and \( v_{m,j} \), if the measurement outcome is 0, do nothing, and if the outcome is 1, perform Z for a correction on qubit \( Q_l \) or \( Q_m \). By straightforward calculation, we obtain the state of three qubits \( Q_l \), \( Q_m \) and \( Q_n \) at nodes \( v_{l,j} \), \( v_{m,j} \) and \( v_{n,j} \), respectively, given by

\[
\sum_{a,b} \lambda_{ab} |ab\rangle_{lm} v_n^{(ab)} |\psi\rangle_n. \quad (53)
\]

Therefore, \( C_{l,m,n} \) is successfully applied on the control qubits at nodes \( v_{l,j} \) and \( v_{m,j} \) and the target qubit at node \( v_{n,j} \) by LOCC assisted by the Bell pairs corresponding to the vertical edges \( S_j \) between nodes \( v_{l,j} \) and \( v_{m,j} \).

**APPENDIX B**

**LOCC IMPLEMENTATION OF CONVERTED QUANTUM CIRCUITS**

In Appendix A, we have shown a protocol to implement a three-qubit fully controlled unitary operation in a set of vertically aligned nodes \( V_j \). In some cases, we can implement more than one three-qubit or two-qubit controlled unitary operations in parallel using the same resource. We show how a sequence of controlled unitary operations represented by converted circuits can be implemented by LOCC assisted by the resource state given by a collection of \((k-1)\) Bell pairs corresponding to a set of vertically aligned edges \( S_j \) in this appendix.

We introduce a new notation for controlled unitary operations for simplifying and unifying descriptions of two-qubit and three-qubit controlled unitary operations. We represent a two-qubit controlled unitary operations that controls the \( i \)-th qubit and targets the \( j \)-th qubit as

\[
(i, i; j), \quad (54)
\]

and a three-qubit controlled unitary gates that controls the \( i \)-th and \( j \)-th qubit and targets the \( k \)-th qubit as

\[
(i, j; k). \quad (55)
\]

Note that we represented \((i, i; j)\) as \( C_{i;j} \) and \((i, j; k)\) as \( C_{i;j;k} \) in the previous sections. Let \( G = \{g_n\} \) be a sequence of controlled unitary operations that is added in step 2 of the conversion protocol. For example, the converted circuit represented by Fig. 11 is described by a sequence

\[
g_1 = (1, 1; 2) \quad (56)
g_2 = (4, 4; 2) \quad (57)
g_3 = (1, 4; 2) \quad (58)
g_4 = (4, 4; 5) \quad (59)
g_5 = (4, 4; 3) \quad (60)
g_6 = (5, 5; 6) \quad (61)
g_7 = (4, 4; 5). \quad (62)
\]

Let \( C_i \) be a set of controlled unitary operations that controls the \( i \)-th qubit:

\[
C_i = \{(a, b; c) \in G; a = i \lor b = i\}. \quad (63)
\]
For example, for $G$ defined by Eqs. (56)–(62),
\[
C_1 = \{g_1, g_3\} \quad (64)
\]
\[
C_4 = \{g_2, g_3, g_4, g_5, g_7\} \quad (65)
\]
\[
C_5 = \{g_6\} \quad (66)
\]
\[
C_2 = C_3 = C_6 = \emptyset. \quad (67)
\]

Define the range of $C_i \neq \emptyset$ as
\[
\text{range}(C_i) = \left( \min_i \{i, \min_c \{(a, b; c) \in C_i\}\}, \max_i \{i, \max_c \{(a, b; c) \in C_i\}\} \right). \quad (68)
\]

For example, for $C_i$ defined by Eqs. (64)–(67),
\[
\text{range}(C_1) = (1, 2) \quad (69)
\]
\[
\text{range}(C_4) = (2, 5) \quad (70)
\]
\[
\text{range}(C_5) = (5, 6). \quad (71)
\]

All the controlled unitary operations in $G$ are implementable by using the following protocol extending the one presented in Appendix A.

The protocol for implementing a sequence of controlled unitary operation in $G$:

1) For applying gates in $C_i$, we create an ancillary qubit state entangled to the $i$-th qubit state by preparing an ancillary qubit in $|0\rangle$ and applying CNOT where the ancillary qubit is the target qubit of CNOT. Then the ancillary qubit state is sent from the $i$-th node $v_{i,j} \in V^j_i$ to the target node by using teleportation. If several different target qubits are included in $C_i$, create another ancillary qubit by the same method at a target node, keep one of the ancillary qubits at the target node and send the other to the next target node. We consume $n_i$ Bell pairs to teleport ancillary qubit states to the target nodes, where $n_i = b - a$ and range($C_i$) = $(a, b)$. Since there is no overlap between ranges of $C_i$ and there is no target unitary operation inserted between control qubits, we can teleport all the ancillary qubit states entangled to the control states to all the target nodes by just consuming $(k - 1)$ Bell pairs.

2) We apply all the controlled unitary operations in $G$ in the target nodes by using the teleported ancillary qubit states entangled to the control qubit states as the control qubits.

3) We decouple the ancillary qubit states by performing the projective measurements on the ancillary qubits similarly to the case of Appendix A in the target nodes and apply correction unitary operations in the control nodes depending on the measurement outcomes.

**APPENDIX C**

**CONVERTED CIRCUIT OF (2, N) AND (3, N)-CLUSTER NETWORK**

First, we prove that any converted circuits of a $(2, N)$-cluster network can be simulated by a circuit consisting of a sequence of $N$ two-qubit unitary operations and local unitary operations. In this case, only two-qubit unitary operations $(1, 1; 2)$ or $(2, 2; 1)$ can be added in step 2 of the conversion protocol. Since applying the gate $(1, 1; 2)$ sequentially for $k \in \mathbb{N}$ times can be simulated by just one use of gate $(1, 1; 2)$ and gate $(2, 2; 1)$ can be simulated by one use of gate $(1, 1; 2)$ and additional local unitary operations, any circuits generated in step 2 of the conversion protocol can be simulated by one use of $(1, 1; 2)$ and local unitary operations.

Next, we prove that any converted circuits of a $(3, N)$-cluster network can be simulated by the circuit of a sequence of $N$ three-qubit fully controlled unitary operations given in the form of
\[
\begin{align*}
|00\rangle|01\rangle|1, 3, 0\rangle \otimes u_{2}^{(00)}(01) |0\rangle|01\rangle|1, 3, 0\rangle \otimes u_{2}^{(01)} |10\rangle|11\rangle|1, 3, 0\rangle \otimes u_{2}^{(10)} |11\rangle|11\rangle|1, 3, 0\rangle \otimes u_{2}^{(11)}
\end{align*}
\]

and local unitary operations. In step 2 of the conversion protocol, every converted circuits can be simulated by six classes of circuits shown in Fig. 10.

In the following, we show that all of these six classes (from class i) to class vi) represented in Fig. 10) can be simulated by a three-qubit fully controlled unitary operation and local unitary operations by investigating each class.

i) A unitary operation obtained by circuit i) is given by
\[
|0\rangle|0\rangle|1\rangle \otimes u_{2}^{(00)}(0) \otimes u_{2}^{(01)}(1) |1\rangle |1\rangle \otimes u_{2}^{(10)}(0) \otimes u_{3}^{(00)}(1) \otimes u_{3}^{(01)}(1) \otimes u_{3}^{(10)}(0) \otimes u_{3}^{(11)}(0)\quad (73)
\]

where $u_{2}^{(i)}$ is a one-qubit unitary operation and $u_{2}^{(i)} \otimes u_{3}^{(i)}$ represents local unitary equivalence. Diagonalize $u_{2}^{(1)}(0)$ and $u_{3}^{(1)}(0)$ as
\[
\begin{align*}
\begin{bmatrix} e^{i\theta_1} & 0 \\
0 & e^{i\theta_2}
\end{bmatrix}
\end{align*}
\]
\[
\begin{align*}
\begin{bmatrix} e^{i\theta_3} & 0 \\
0 & e^{i\theta_4}
\end{bmatrix}
\end{align*}
\]

Since the right handside of Eq. (73) is locally unitarily equivalent to a diagonal unitary operation in the computational basis, this circuit can be simulated by a three-qubit fully controlled unitary operation and local unitary operations.

![Fig. 12. The six classes of converted quantum circuits obtained by step 2 of the conversion protocol of a $(3, N)$-cluster network.](image-url)
ii) In circuit ii), the two-qubit controlled unitary operation (2, 2; 3) can be decomposed into
\[
|0\rangle \langle 0|_2 \otimes u_3^{(0)} + |1\rangle \langle 1|_2 \otimes u_3^{(1)} = v_3 \left( \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta_3} \end{pmatrix} \otimes |0\rangle \langle 0|_3 + \begin{pmatrix} 1 & 0 \\ 0 & e^{i\theta_3} \end{pmatrix} \otimes |1\rangle \langle 1|_3 \right) v_3^\dagger u_3^{(0)},
\]
where \(v_3\) is a unitary operation that diagonalizes \(u_3^{(1)} u_3^{(0)\dagger}\). Thus, this circuit is locally unitarily equivalent to a three-qubit fully controlled unitary operation.

iii) Circuit iii) consists of just a three-qubit fully controlled unitary operation.

iv) Circuit iv) can be simulated by a three-qubit fully controlled unitary operation and local unitary operations since we can diagonalize a unitary operation obtained by the circuit in the same way as circuit i).

v) In the same way as circuit ii), circuit v) is locally unitarily equivalent to a three-qubit fully controlled unitary operation.

vi) In the same way as circuit i), circuit vi) is locally unitarily equivalent to a three-qubit fully controlled unitary operation.

APPENDIX D
MAXIMALLY ENTANGLED STATE CONVERSION BY A SEPARABLE MAP

We prove a lemma about convertibility of maximally entangled states in this appendix. We analyze a property of bipartite separable maps which preserves entanglement of maximally entangled states.

Let \(\Psi_{in} = \sum_i |A_i\rangle |B_i\rangle\) and \(\Psi_{out} = \sum_i |a_i\rangle |b_i\rangle\), where \(\{|A_i\rangle \in \mathcal{H}_A\}\) and \(\{|B_i\rangle \in \mathcal{H}_B\}\) are orthonormal sets and \(\{|a_i\rangle \in \mathcal{H}_a\}\) and \(\{|b_i\rangle \in \mathcal{H}_b\}\) are orthonormal bases.

Lemma 3. Let \(\{E_k \in \mathcal{L}(\mathcal{H}_A : \mathcal{H}_a)\}, \{F_k \in \mathcal{L}(\mathcal{H}_B : \mathcal{H}_b)\}\) be sets of linear operators. If \(\{E_k \otimes F_k\}\) satisfies
\[
\sum_k E_k^\dagger E_k \otimes F_k^\dagger F_k = I_{AB}
\]
and for all \(k\),
\[
E_k \otimes F_k |\Psi_{in}\rangle = \sqrt{p_k} |\Psi_{out}\rangle
\]
is satisfied, then for all \(\{k|p_k \neq 0\}\),
\[
\exists \alpha_k > 0, \exists U_k^M \in \mathcal{U}(\mathbb{C}^d), \ E_k^M = \alpha_k U_k^M, \ F_k^M = \frac{\sqrt{p_k}}{\alpha_k} U_k^M,
\]
where \(E_k^M\) and \(F_k^M\) are \(d \times d\) matrices such that \(E_k^M_{i,j} = |a_i\rangle |E_k A_j\rangle\), \(F_k^M_{i,j} = |b_i\rangle |F_k B_j\rangle\). \(\mathcal{U}(\mathbb{C}^d)\) is the set of \(d \times d\) unitary matrices and \(U^M\) is the complex conjugate of \(U^M\).

Proof. By straightforward calculation, we obtain
\[
\forall k, \ E_k^M (F_k^M)^\dagger = \sqrt{p_k} I_d \Rightarrow \forall k \in \{k|p_k \neq 0\}, \ F_k^M = \sqrt{p_k} ((E_k^M)^\dagger)^T, \quad (80)
\]
and
\[
\sum_k (E_k^M)^\dagger E_k^M \otimes (F_k^M)^\dagger F_k^M = I_{d^2}. \quad (81)
\]
By using Eq. (80) and Eq. (81), we obtain
\[
\text{tr} \left( \sum_k E_k^M F_k^M \otimes (E_k^M)^\dagger (F_k^M)^\dagger \right) = \text{tr} \left( \sum_k E_k^M F_k^M \otimes (E_k^M)^\dagger (F_k^M)^\dagger \right) + \epsilon = \sum_k p_k \text{tr} \left( E_k^M F_k^M \otimes (E_k^M F_k^M)^\dagger \right) = \epsilon = d^2 - \epsilon,
\]
where \(\epsilon = \text{tr} \left( \sum_k (E_k^M F_k^M \otimes (E_k^M F_k^M)^\dagger) \right) \geq 0\). We let \(P_k = E_k^M F_k^M\) be a \(d \times d\) positive matrix and \(\{\lambda_k > 0\}_{i=0}^{d-1}\) be the set of eigenvalues of \(P_k\). Then the eigenvalues of \(E_k^M F_k^M \otimes (E_k^M F_k^M)^\dagger \) are \(\{1/\lambda_k\}_{i=0}^{d-1}\) and the condition Eq. (82) is given by
\[
\sum_k p_k \frac{d}{i} \lambda_k^d = d^2 - \epsilon. \quad (83)
\]
Using the Cauchy-Schwarz inequality, we obtain
\[
\left( \sum_i \lambda_k^d \right) \left( \sum_j \frac{1}{\lambda_k^d} \right) \geq \left( \sum_i 1 \right)^2 = d^2. \quad (84)
\]
The equality holds if and only if \(\lambda_k = \alpha^2 > 0\) for all \(i\). By using Eqs. (83) (84) and the fact that \(\{pk|p_k \neq 0\}\) is a probability distribution, we obtain for all \(\{k|p_k \neq 0\}\),
\[
\exists \alpha > 0; \ P_k = E_k^M F_k^M = \alpha^2 I_d, \quad (85)
\]
\[
\epsilon = 0. \quad (86)
\]
\[\square\]

APPENDIX E
TWO CONDITIONS IN THEOREM 2 ARE EQUIVALENT IN THE CASE OF THE \((3, N)\)-CLUSTER NETWORKS

For \(k = 3\), the 2\({}^k\) by 2\({}^k\) unitary matrix \(V_i^M\) in Theorem 2 is written by
\[
V_i^M = E_{1,i}^{(0)} \otimes E_{2,i}^{(0)} \otimes E_{3,i}^{(0)} + E_{1,i}^{(0)} \otimes E_{2,i}^{(1)} \otimes E_{3,i}^{(1)} + E_{1,i}^{(1)} \otimes E_{2,i}^{(0)} \otimes E_{3,i}^{(1)} + E_{1,i}^{(1)} \otimes E_{2,i}^{(1)} \otimes E_{3,i}^{(1)}. \quad (87)
\]
By using the result on local unitary equivalence of unitary operations with operator Schmidt rank 2 obtained by Cohen and Yu [26] (Theorem 1 of [26]), we have
\[ V_M^{i} \overset{LU}{=} |00\rangle_{AC} \otimes W_B^{0(0)} + |01\rangle_{AC} \otimes W_B^{0(1)} + |10\rangle_{AC} \otimes W_B^{1(0)} + |11\rangle_{AC} \otimes W_B^{1(1)}, \]
where \( W_B^{0(0)} \) and \( W_B^{0(1)} \) are unitary matrices, \( \overset{LU}{=} \) represents a locally unitarily equivalence and we identify a three-qubit unitary operation on \( H_A \otimes H_B \otimes H_C \) as its matrix representation \( V_M^{i} \). Thus, it is shown that
\[ V_M^{i} \overset{LU}{=} |\phi\rangle_{1,3}|0\rangle_2, \]
in step (i), where we denote the index of the qubit corresponding to the first horizontal wire as 1 and that of the others likewise. After applying Hadamard gates in step (ii), we obtain
\[ H_1 \otimes H_3|\phi\rangle_{1,3}|+\rangle_2, \]
where \( |\pm\⟩ = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle) \). After applying \( C_{1,3,2} \) in step (iii), we obtain
\[ \frac{1}{\sqrt{2}} (H_1 \otimes H_3|\phi\rangle_{1,3}|0\rangle_2 + Z_1 H_1 \otimes Z_3 H_3|\phi\rangle_{1,3}|1\rangle_2). \]
After applying Hadamard gates and Pauli X operations in step (iv), we obtain
\[ \frac{1}{\sqrt{2}} (X_1 \otimes X_3|\phi\rangle_{1,3}|+\rangle_2 + |\phi\rangle_{1,3}|−\rangle_2). \]
After applying \( C_{1,3,2} \) in step (v), we obtain
\[ \frac{1}{2} (|00\rangle_{11} + |11\rangle_{00})|\phi\rangle_{1,3}(e^{ix}|0\rangle - ie^{-ix}|1\rangle) + \frac{1}{2} (|01\rangle_{10} + |10\rangle_{01})|\phi\rangle_{1,3}(e^{-ix}|0\rangle - ie^{ix}|1\rangle) + \frac{1}{2} (|00\rangle_{00} + |11\rangle_{11})|\phi\rangle_{1,3}(e^{ix}|0\rangle + ie^{-ix}|1\rangle) + \frac{1}{2} (|01\rangle_{01} + |10\rangle_{10})|\phi\rangle_{1,3}(e^{-ix}|0\rangle + ie^{ix}|1\rangle). \]
After applying a single qubit unitary operation \( u(x) \) given by
\[ u(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} e^{ix} & -ie^{-ix} \\ e^{ix} & ie^{-ix} \end{pmatrix} \]
in step (vi), we obtain
\[ \frac{1}{\sqrt{2}} e^{ix} \cos(x - y)(|00\rangle_{00} + |11\rangle_{11})|\phi\rangle_{1,3}|0\rangle_2 + \frac{1}{\sqrt{2}} e^{-ix} \cos(x + y)(|01\rangle_{01} + |10\rangle_{10})|\phi\rangle_{1,3}|0\rangle_2 + \frac{1}{\sqrt{2}} ie^{ix} \sin(x - y)(|00\rangle_{00} + |11\rangle_{11})|\phi\rangle_{1,3}|0\rangle_2 + \frac{1}{\sqrt{2}} ie^{-ix} \sin(x + y)(|01\rangle_{01} + |10\rangle_{10})|\phi\rangle_{1,3}|0\rangle_2 + \frac{1}{\sqrt{2}} e^{ix} \cos(x - y)(|00\rangle_{00} + |11\rangle_{11})|\phi\rangle_{1,3}|1\rangle_2 + \frac{1}{\sqrt{2}} e^{-ix} \cos(x + y)(|01\rangle_{01} + |10\rangle_{10})|\phi\rangle_{1,3}|1\rangle_2 + \frac{1}{\sqrt{2}} ie^{ix} \sin(x - y)(|00\rangle_{00} + |11\rangle_{11})|\phi\rangle_{1,3}|1\rangle_2 + \frac{1}{\sqrt{2}} ie^{-ix} \sin(x + y)(|01\rangle_{01} + |10\rangle_{10})|\phi\rangle_{1,3}|1\rangle_2 \]
for any measurement outcome. By straightforward calculation, we can verify Eq. (97) is equivalent to \( U_{global}(x, y, z)|\phi\rangle_{1,3} \).

**APPENDIX G**

**ANALYSIS OF A BIPARTITE PROPERTY OF FOUR QUBIT STATES**

We prove that there is no pure state of four qubits \( |\Phi\rangle_{1,2,3,4} \) satisfying
\[
\begin{align*}
\text{Sch#}_{1,2}^{3,4}(|\Phi\rangle) & = 4, \\
\text{Sch#}_{1,3}^{2,4}(|\Phi\rangle) & = 2, \\
\text{Sch#}_{1,4}^{2,3}(|\Phi\rangle) & = 2.
\end{align*}
\]
In [27], it is shown that any pure states of four qubits can, up to permutations of the qubits, be transformed into one of the following nine families of states by determinant 1 SLOCC:

\[
\Phi_1 = \frac{a + d}{2} (|0000\rangle + |1111\rangle) + \frac{c}{2} (|0101\rangle + |1010\rangle) + \frac{b}{2} (|0110\rangle + |1001\rangle)
\]
\[
\Phi_2 = \frac{a + d}{2} (|0000\rangle + |1111\rangle) + \frac{c}{2} (|0101\rangle + |1010\rangle) + \frac{b}{2} (|0110\rangle + |1001\rangle)
\]
\[
\Phi_3 = a (|0000\rangle + |1111\rangle) + b (|0101\rangle + |1010\rangle) + c (|0110\rangle + |1001\rangle)
\]
\[
\Phi_4 = a (|0000\rangle + |1111\rangle) + \frac{a + b}{2} (|0101\rangle + |1010\rangle) + \frac{a - b}{2} (|0110\rangle + |1001\rangle)
\]
\[
\Phi_5 = a (|0000\rangle + |0101\rangle + |1010\rangle + |1111\rangle) + i (|0001\rangle + |0110\rangle - |1001\rangle)
\]
\[
\Phi_6 = a (|0000\rangle + |1111\rangle) + b (|0101\rangle + |1010\rangle) + c (|0110\rangle + |1001\rangle)
\]
\[
\Phi_7 = (|0000\rangle + |0101\rangle + |1000\rangle + |1100\rangle)
\]
\[
\Phi_8 = (|0000\rangle + |1111\rangle + |1010\rangle + |1101\rangle + |1110\rangle)
\]
\[
\Phi_9 = (|0000\rangle + |0111\rangle)
\]
where \(a, b, c, d\) are complex parameters.

Since the Schmidt number of a state cannot be increased under SLOCC and determinant 1 SLOCC is invertible, the Schmidt number of a state is invariant under determinant 1 SLOCC. Thus, we show that no state of the nine families simultaneously satisfies Eqs. (98)-(100). There are three ways to divide four qubits into a pair of two qubits. We denote the set of Schmidt numbers of a four qubit state \(\Phi\) for all separations as \(\text{Sch}#(\Phi) = \{\text{Sch}#_{1,2}(\Phi), \text{Sch}#_{1,3}(\Phi), \text{Sch}#_{1,4}(\Phi)\}\).}

**Theorem 5.** There is no four qubit state \(\Phi \in H_1 \otimes H_2 \otimes H_3 \otimes H_4\) such that

\[
\text{Sch}#(\Phi) = \{4, 2, 2\}.
\]

**Proof.** By straightforward calculation, we can easily check that

\[
\text{Sch}#(\Phi_6) = \{n_6, n_6, n_6\}
\]
\[
\text{Sch}#(\Phi_7) = \{3, 3, 3\}
\]
\[
\text{Sch}#(\Phi_8) = \{3, 3, 3\}
\]
\[
\text{Sch}#(\Phi_9) = \{2, 2, 2\}
\]

where \(n_6 = \# \{\sqrt{2}, \frac{1}{\sqrt{2}} \sqrt{1 + 4 |a|^2 + 1}, \frac{1}{\sqrt{2}} \sqrt{1 + 4 |a|^2 - 1}\}\) and \(\#S\) is the number of non-zero elements of set \(S\). Since \(n_6 = 2\) or \(n_6 = 3\), these four states do not satisfy Eq. (101).

An element of \(\text{Sch}#(\Phi_6)\) is \# \{1, \sqrt{2}, 2|a|\}. To satisfy Eq. (101), \(a = 0\) is required. Then

\[
\text{Sch}#(\Phi_6) = \{2, 3, 3\},
\]

which does not satisfy Eq. (101).

An element of \(\text{Sch}#(\Phi_4)\) is \# \{|b| \} + \# \{|x| x^2 - (3|a|^2 + 2)x^2 + (3|a|^4 + 2|a|^2 + 1)x - |a|^6 = 0\}. To satisfy Eq. (101), the element must be 2 or 4. If the element is 2, since \# \{|x| x^2 - (3|a|^2 + 2)x^2 + (3|a|^4 + 2|a|^2 + 1)x - |a|^6 = 0\} is larger than 1 and is 2 if and only if \(a = 0\), we have

\[
a = b = 0.
\]

Then \(\text{Sch}#(\Phi_4) = \{2, 2, 2\}\). Thus, the element must be 4. Since \# \{|x| x^2 - (3|a|^2 + 2)x^2 + (3|a|^4 + 2|a|^2 + 1)x - |a|^6 = 0\} is 3 and if only if \(a \neq 0\), we have

\[
a \neq 0, b \neq 0.
\]

Another element of \(\text{Sch}#(\Phi_4)\) is \# \{|a - b| \} + \# \{|x| 64x^2 + \cdots x^2 + \cdots x - |a|^2|3a + b|^2 = 0\}, where we abbreviate coefficients of \(x^2\) and \(x\). Since this element must be 2, it is necessary that

\[
a - b = 0 \text{ or } 3a + b = 0.
\]

The other element of \(\text{Sch}#(\Phi_4)\) is \# \{|a + b| \} + \# \{|x| 64x^2 + \cdots x^2 + \cdots x - |a|^2|3a - b|^2 = 0\}, where we abbreviate coefficients of \(x^2\) and \(x\). Since this element must be 2, it is necessary that

\[
a + b = 0 \text{ or } 3a - b = 0.
\]

We can easily check that it is impossible to simultaneously satisfy Eqs. (108)-(110).

\[
\text{Sch}#(\Phi_3) = \{n_3, n_3, n_3\},
\]

where

\[
n_3 = \# \{|a + b, |a - b|\},
\]
\[
n_3' = \# \{|x| 1 + 4|a|^2 \pm 1\}.
\]

To satisfy Eq. (101), \(n_3'\) must be 2, that is \(a = b = 0\). Then \(n_3 = 1\), which does not satisfy Eq. (101).

\[
\text{Sch}#(\Phi_2) = \{n_2, n_2, n_2\},
\]

where

\[
n_2 = \# \{|a, |b|, \sqrt{1 + 4|a|^2} \pm 1\},
\]
\[
n_2' = \# \{|x| a + b \pm 2|c|, \sqrt{1 + |a - b|^2} \pm 1\},
\]
\[
n_2'' = \# \{|x - b - 2|c|, \sqrt{1 + |a + b|^2} \pm 1\}.
\]

In the following, we verify that \(\{n_2, n_2', n_2''\}\) cannot be \{4, 2, 2\} \{2, 4, 2\} or \{2, 2, 4\}.

1) \(\{n_2, n_2', n_2''\} \neq \{4, 2, 2\}:

   If \(n_2 = 4\), it is necessary that

   \[
a \neq 0, \ b \neq 0, \ c \neq 0.
\]

   If \(n_2' = 2\), it is necessary that

   \[
a - b = a + b + 2c = 0,
\]
\[
a - b = a - b - 2c = 0,
\]
   \[\text{or } a + b - 2c = a + b + 2c = 0\]

   If \(n_2'' = 2\), it is necessary that

   \[
a + b = a - b + 2c = 0,
\]
\[
a + b = a - b - 2c = 0,
\]
   \[\text{or } a - b - 2c = a - b + 2c = 0\]

We can easily check that it is impossible to simultaneously satisfy Eqs. (116)-(122).
2) \( \{n_2, n_2', n_2''\} \neq \{2, 4, 2\} \):
If \( n_2 = 2 \), it is necessary that
\[
\begin{align*}
a &= b = 0, \\
a &= c = 0, \\
or \ b &= c = 0.
\end{align*}
\]
(123)
(124)
(125)

With the necessary condition for \( n_2'' = 2 \), we obtain that
\[
a = b = c = 0.
\]
(126)

Then, it is impossible to satisfy \( n_2'' = 4 \).

3) \( \{n_2, n_2', n_2''\} \neq \{2, 2, 4\} \):
If \( n_2 = 2 \), it is necessary that
\[
\begin{align*}
a &= b = 0, \\
a &= c = 0, \\
or \ b &= c = 0.
\end{align*}
\]
(127)
(128)
(129)

With the necessary condition for \( n_2' = 2 \), we obtain that
\[
a = b = c = 0.
\]
(130)

Then, it is impossible to satisfy \( n_2'' = 4 \).

Finally, we analyze \( Sch\#(\Phi_1) \). \( Sch\#(\Phi_1) \) is \( \{n_1, n_1', n_1''\} \), where
\[
\begin{align*}
n_1 &= \#\{a, |b|, |c|, |d|\}, \\
n_1' &= \#\{|a + b - c - d|, |a + b + c - d|, \\
&\quad |a - b + c - d|, |a + b + c + d|\}, \\
n_1'' &= \#\{|a + b + c + d|, |a - b + c + d|, \\
&\quad |a + b - c + d|, |a + b - c - d|\}.
\end{align*}
\]
(131)
(132)
(133)

Note that \( n_1, n_1' \) and \( n_1'' \) are invariant under permutation of \( a, b, c \) and \( d \). We verify that \( \{n_1, n_1', n_1''\} \) cannot be \( \{4, 2, 2\}, \{2, 4, 2\} \) or \( \{2, 2, 4\} \) in the following.

1) \( \{n_1, n_1', n_1''\} \neq \{4, 2, 2\} \):
If \( n_1 = 4 \), it is necessary that
\[
a \neq 0, \ b \neq 0, \ c \neq 0, \ d \neq 0.
\]
(134)

If \( n_1' = 2 \), it is necessary that in general
\[
a + b - c - d = 0, \ a - b + c - d = 0
\]
\[
\Leftrightarrow a = d, \ b = c.
\]
(135)
(136)

Then
\[
n_1'' = \#\{|2b|, |2a|, |2a|, |2b|\} = 4.
\]
(137)

2) \( \{n_1, n_1', n_1''\} \neq \{2, 4, 2\} \) and \( \{n_1, n_1', n_1''\} \neq \{2, 2, 4\} \):
If \( n_1 = 2 \), it is necessary that in general
\[
a = 0, \ b = 0, \ c \neq 0, \ d \neq 0.
\]
(138)

Then
\[
n_1' = n_1'' = \#\{|c + d|, |c + d|, |c - d|, |c - d|\}.
\]
(139)