International Journal of Quantum Information
Vol. 10, No. 3 (2012) 1250034 (18 pages)

DOI: 10.1142/S0219749912500347

World Scientific
www.worldscientific.com

ON A GENERALIZED QUANTUM SWAP GATE

COLIN M. WILMOTT

Faculty of Informatics, Masaryk University, Botanická 68a,
602 00 Brno, Czech Republic
wilmott@fi.muni.cz

PETER R. WILD

Department of Mathematics, Royal Holloway,
University of London, Egham,
Surrey TW20 0EX, United Kingdom

Received 25 November 2011
Published 4 May 2012

The SWAP gate plays a central role in network designs for qubit quantum computation. However, there has been a view to generalize qubit quantum computing to higher dimensional quantum systems. In this paper we construct a generalized SWAP gate using only instances of the generalized controlled-NOT gate to cyclically permute the states of d qudits for d prime.

Keywords: Quantum computation; elementary quantum gates; quantum circuits; modular binomial coefficients.

1. Introduction

Of central importance to the theory of quantum computation is the role assumed by multiple qubit gates in establishing a basis for quantum network design. Moreover, the multiple qubit component that best establishes itself as the hallmark of quantum network design is the controlled-NOT (CNOT) gate. The CNOT gate possesses a fundamental importance in the theory of quantum computation assuming key roles in quantum measurement and quantum error correction. Barenco *et al.* have shown that the CNOT gate is a principal component in universal quantum gate constructions.[1] Furthermore, when we note that our ability to preserve quantum coherence rests with our ability to successfully implement quantum computations, the CNOT gate further distinguishes itself as the hallmark multiple qubit gate as it is one of the few quantum gates to have been experimentally realized within its coherence time.[2]

A standard feature of quantum circuitry design is to express any multiple qubit gate in terms of single qubit gates and the CNOT gate.[1] An example of this is

provided by the well-known SWAP gate which describes the quantum operation that permutes the states of two qubits. The SWAP gate is seen as an important component in the network design of Shor's algorithm[3] and Liang and Li maintain that successfully implementing the SWAP gate is a necessary condition for the networkability of quantum computation.[4] Recently, however, it has been asserted that there exist advantages in generalizing quantum computation to higher dimension basis systems.[5] Considering new quantum network designs may therefore help to reveal the promise of qudit quantum computing. Indeed, such new designs design may have merit in itself.

In this paper, we concern ourselves with the design of a quantum circuit to realize a generalized SWAP gate that cyclically permutes d input qudit subsystems for d prime. We restrict ourselves to using only instances of the generalized CNOT gate. Section 2 introduces preliminary material as motivation for the design of quantum circuits which exploit the generalized CNOT gate. Section 3 introduces the design method for a quantum circuit to realize a generalized SWAP of d qudits for d prime. The analysis makes great use of modular binomial relationships to achieve the desired result. Finally, Sec. 4 revises the design method of the previous section to achieve certain permutations of d qudits for d other than prime.

2. Preliminaries

Let $\mathcal{H}$ denote the d-dimensional complex Hilbert space $\mathbb{C}^d$. We fix each orthonormal basis state of the d-dimensional space to correspond to an element of ring $\mathbb{Z}_d$ of integers modulo d. The basis $\{|0\rangle, |1\rangle, \ldots, |d-1\rangle\} \subset \mathbb{C}^d$ whose elements correspond to the column vectors of the identity matrix $\mathbb{I}_d$ is called the computational basis. A *qudit* is a d-dimensional quantum state $|\psi\rangle \in \mathcal{H}$ written as $|\psi\rangle = \sum_{i=0}^{d-1} \alpha_i |i\rangle$ where $\alpha_i \in \mathbb{C}$ and $\sum_{i=0}^{d-1} |\alpha_i|^2 = 1$. For a pair of qudits $|\psi\rangle, |\phi\rangle \in \mathcal{H}$, the generalized CNOT gate is a two-qudit quantum gate that acts on the state $|\psi\rangle \otimes |\phi\rangle \in \mathcal{H} \otimes \mathcal{H}$. The generalized CNOT gate has control qudit $|\psi\rangle$ and target qudit $|\phi\rangle$, and its action on the basis states $|m\rangle \otimes |n\rangle \in \mathcal{H} \otimes \mathcal{H}$ is given by

$$\text{CNOT}|m\rangle \otimes |n\rangle = |m\rangle \otimes |n \oplus m\rangle, \quad m, n \in \mathbb{Z}_d, \tag{1}$$

with $\oplus$ denoting addition modulo d. Figure 1 illustrates the role played by the CNOT gate in describing the well-known SWAP gate for qubits.

We now introduce a generalized SWAP gate that cyclically permutes the states of d qudit subsystems for d prime. The construction process is restricted to using only instances of the generalized CNOT gate.

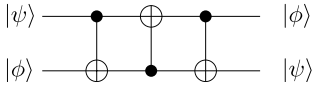

Fig. 1. The SWAP gate illustrating the permutation of two qubits through the use of three CNOT gates. The system begins in the state $|\psi\rangle \otimes |\phi\rangle$ and ends in the state $|\phi\rangle \otimes |\psi\rangle$.

3. A Generalized SWAP Gate

We construct a generalized quantum SWAP gate composed entirely in terms of the generalized CNOT gate to cyclically permute the states of d qudit subsystems. We note that Wilmott has shown that pairwise swaps of two qudits using only instances of the generalized CNOT gate cannot be implemented in dimensions $d \equiv 3(\text{mod}) 4$.[6] Consequently, arguing a generalized SWAP of d qudit subsystems entirely in terms of the generalized CNOT gate via a staggered set of pairwise swaps of qudits is not readily achievable.

Figure 2 outlines a generalized quantum SWAP gate of d qudit subsystems whereby the design method is restricted to using only generalized CNOT gates. We suppose that the first quantum system $\mathcal{A}_0$ prepared in the state $|e_0\rangle_0$, the second system $\mathcal{A}_1$ prepared in the state $|e_1\rangle_1$ and so forth, with the final system $\mathcal{A}_{d-1}$ prepared in the state $|e_{d-1}\rangle_{d-1}$. The process describes a quantum network that uses only generalized CNOT gates to realize a generalized SWAP of d qudits for d prime with the result that the system $\mathcal{A}_0$ is in the state $|e_1\rangle_0$, the system $\mathcal{A}_1$ is in the state $|e_2\rangle_1$ and so forth, until the system $\mathcal{A}_{d-1}$ is in the state $|e_0\rangle_{d-1}$. We make use of the following result.

Lemma 1.[7] $\sum_{n=0}^{k} \binom{l+n}{n} = \binom{l+k+1}{k}$.

Theorem 1. *Let $d = p$ be a prime. We provide an algorithm for the construction of a generalized SWAP gate using only instances of the generalized CNOT gate. The generalized SWAP gate has*

Input: $|e_k\rangle_k; \quad k = 0, \ldots, d-1$
Output: $|e_{k+1}\rangle_k; \quad k = 0, \ldots, d-2, \ |e_0\rangle_{d-1}$.

The generalized SWAP gate algorithm is described as follows:

Input: $e_k := i_k^0; \quad k = 0, \ldots, d-1$
Output: $i_k^{d+2} = e_{k+1}; \quad k = 0, \ldots, d-1, \ i_{d-1}^{d+2} = e_0$.

Stage 1 Initialization $j = 0$.

$$e_k := i_k^0$$

for $k = 0, \ldots, d-1$. The algorithm initiates at stage 1, step $j = 0$ by making the correspondence between a representative input element i_k^0 of the algorithm and each standard basis state e_k.

Stage 2 $j = 1, \ldots, d-1$.

$$\begin{aligned} i_0^j &= i_0^{j-1}, \\ i_k^j &= i_{k-1}^j + i_k^{j-1}; \quad k = 1, \ldots, d-1. \end{aligned}$$

Stage 2 consists of $d-1$ steps which repeat the sequence of gates of step $j = 1$. The sequence of gates at step $j = 1$, see Fig. 3, is targeted on systems $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}$. Each step of Fig. 3 is a composition of generalized CNOT gates acting on consecutive pairs of systems and is written as a shorthand form to represent a sequence of generalized CNOT gates as illustrated in

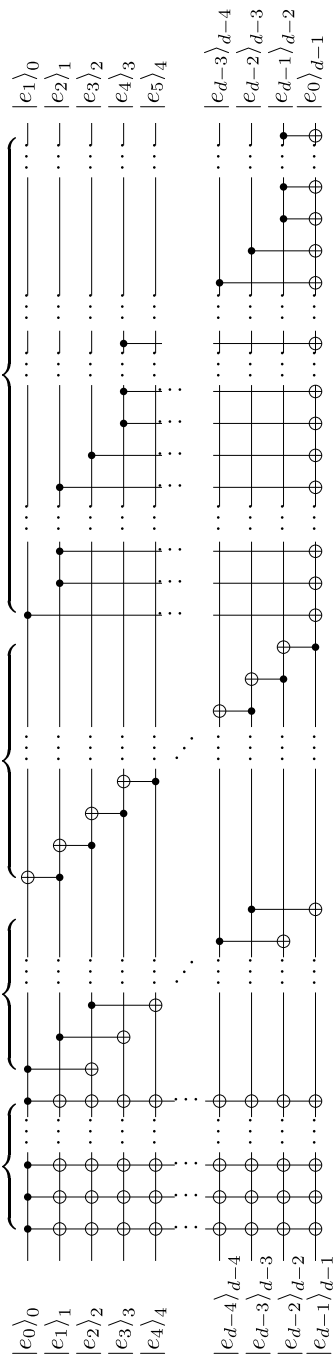

Fig. 2. A generalized SWAP gate composed entirely in terms of the generalized CNOT gate that cyclically permutes the states of d qudit subsystems.

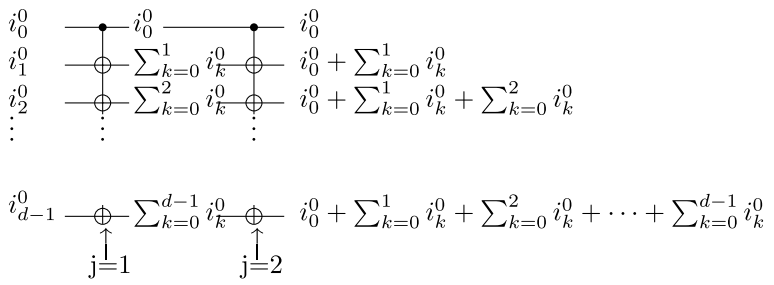

Fig. 3. Generalized SWAP gate; stage 2, steps $j = 1, 2$.

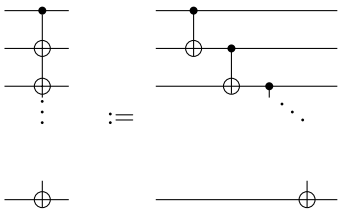

Fig. 4. Generalized SWAP gate; stage 2, step $j \in \{1, \ldots, d-1\}$. We present a shorthand description of algorithm steps occurring at each step of stage 2. The shorthand description denotes a composition of generalized CNOT gates acting on consecutive pairs of systems.

Fig. 4. The algorithm process of step $j = 1$ transforms the input sequence $i_0^0, i_1^0, i_2^0, \ldots, i_{d-1}^0$ to the state given by $i_0^0, \sum_{k=0}^{1} i_k^0, \sum_{k=0}^{2} i_k^0, \ldots, \sum_{k=0}^{d-1} i_k^0$. Similarly, the algorithm at step $j = 2$ takes the output from step $j = 1$ as input and repeats the sequence of gates. The resulting state of the circuit at step $j = 2$ is given by i_0^0, $i_0^0 + \sum_{k=0}^{1} i_k^0$, $i_0^0 + \sum_{k=0}^{1} i_k^0 + \sum_{k=0}^{2} i_k^0, \ldots, i_0^0 + \sum_{k=0}^{1} i_k^0 + \sum_{k=0}^{2} i_k^0 + \cdots + \sum_{k=0}^{d-1} i_k^0$. This process continues to step $j = d-1$. Figure 5 illustrates initialization on the circuit and the subsequent $d-1$ steps of stage 2.

Stage 3 $j = d$.

$$
\begin{aligned}
i_0^d &= i_0^{d-1}, \\
i_1^d &= i_1^{d-1}, \\
i_k^d &= i_{k-2}^d + i_k^{d-1}; \quad k = 2, \ldots, d-1.
\end{aligned}
$$

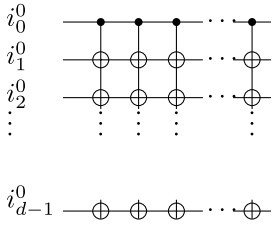

Fig. 5. Generalized SWAP gate; stage 2, steps $j = 1, \ldots, d-1$.

The sequence of values $i_0^{d-1}, i_1^{d-1}, \ldots, i_{d-1}^{d-1}$ corresponding to the final step of stage 2 are carried forward as an input sequence for stage 3, step $j = d$. The algorithm step keeps the values i_0^{d-1}, i_1^{d-1} and returns them as outcomes i_0^d, i_1^d for step $j = d$. The remaining systems are then targeted in an iterative process. For instance, the outcome i_2^d for step $j = d$ is given as $i_0^d + i_2^{d-1}$. This value is then stored as the result i_2^d for e_2 at stage 3. The outcome state for $\mathcal{A}_0, \mathcal{A}_1, \mathcal{A}_2$ at stage 3 have thus been determined. To evaluate the result value for $\mathcal{A}_3$, the algorithm computes $i_1^d + i_3^{d-1}$ and stores this value as the outcome i_3^d for stage 3. Figure 6 illustrates the process that determines the current state of the algorithm following stage 3, step $j = d$ in diagrammatic shorthand form for the sequence of generalized CNOT gates.

Stage 4 $j = d + 1$.

$$\begin{aligned} i_k^{d+1} &= i_k^d + i_{k+1}^d, \\ i_{d-1}^{d+1} &= i_{d-1}^d; \quad k = 0, \ldots, d-2. \end{aligned}$$

Stage 4 is depicted in Fig. 7 and consists of a single step, $j = d + 1$, whose primary algorithm operation acts as a generalized CNOT gate on the $d - 1$ consecutive pairs of systems $(\mathcal{A}_k, \mathcal{A}_{k+1})$ for $k = 0, \ldots, d-2$, computing $(i_k^d + i_{k+1}^d)$ and storing these values as the outcome i_k^{d+1}. The value i_{d-1}^d is returned as the outcome i_{d-1}^{d+1}.

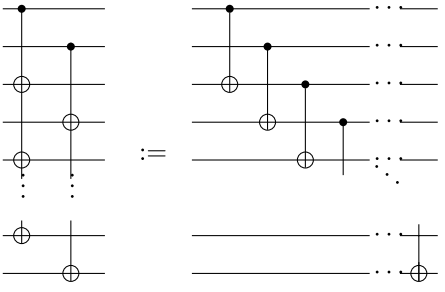

Fig. 6. Generalized SWAP gate; stage 3, step $j = d$. A shorthand description for the sequences of generalized CNOT gates that alternate between system pairs $(\mathcal{A}_{2k}, \mathcal{A}_{2(k+1)})$ and $(\mathcal{A}_{2k+1}, \mathcal{A}_{2k+3})$.

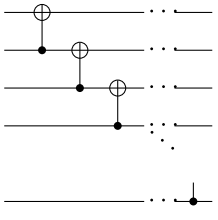

Fig. 7. Generalized SWAP gate; stage 4, step $j = d + 1$.

Stage 5 $j = d + 2$.

$$\begin{aligned} i_k^{d+2} &= i_k^{d+1}, \\ i_{d-1}^{d+2} &= i_{d-1}^{d+1} + \sum_{k=0}^{d-2} \eta_k i_k^{d+2}; \quad k = 0, \ldots, d-2 \end{aligned}$$

with

$$\sum_{k=0}^{d-2} \eta_k i_k^{d+2} := \sum_{t=0}^{\lfloor \frac{d-2}{2} \rfloor} (d-1) i_{2t+1}^{d+2} + \sum_{t=0}^{\frac{d-3}{2}} i_{2t}^{d+2}. \tag{2}$$

Stage 5 concludes the algorithm with a set of gates targeted on system $\mathcal{A}_{d-1}$ whose current state is represented by i_{d-1}^{d+1}. The values $i_0^{d+2}, i_1^{d+2}, \ldots, i_{d-2}^{d+2}$ for the respective systems $\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_{d-2}$ are unchanged from their representative values $i_0^{d+1}, i_1^{d+1}, \ldots, i_{d-2}^{d+1}$ at step $j = d + 1$, and are returned as outcomes in the final state for step $j = d + 2$. The final state of $\mathcal{A}_{d-1}$ is given by $i_{d-1}^{d+2} = i_{d-1}^{d+1} + \sum_{k=0}^{d-2} \eta_k i_k^{d+2} = i_{d-1}^{d+1} + i_0^{d+1} + (d-1) i_1^{d+1} + i_2^{d+1} + (d-1) i_3^{d+1} + \cdots + i_{d-3}^{d+1} + (d-1) i_{d-2}^{d+1}$. Thus, for odd valued k there is a gate with i_k^{d+2} as control, and, for even valued k there are $d-1$ gates with i_k^{d+2} as control. Stage 5 is represented in Fig. 8.

Proof. We show that the generalized SWAP gate algorithm outputs $i_k^{d+2} = e_{k+1}$ for $k = 0, \ldots, d+2$ and $i_{d-1}^{d+2} = e_0$. At step $j = 0$, we have that,

$$e_k := i_k^0; \quad k = 0, \ldots, d-1. \tag{3}$$

At stage 2, step $j = 1$ the algorithm sets $i_0^1 = i_0^0$ and computes i_1^1 as $i_1^1 = i_0^1 + i_1^0$. Similarly, $i_2^1 = i_1^1 + i_2^0 = i_0^0 + i_1^0 + i_2^0 = \sum_{m=0}^{2} i_m^0$. Therefore, for $k = 1, \ldots, d-1$, we have

$$\begin{aligned} i_k^1 &= i_{k-1}^1 + i_k^0 \\ &= i_{k-2}^1 + i_{k-1}^0 + i_k^0 \end{aligned}$$

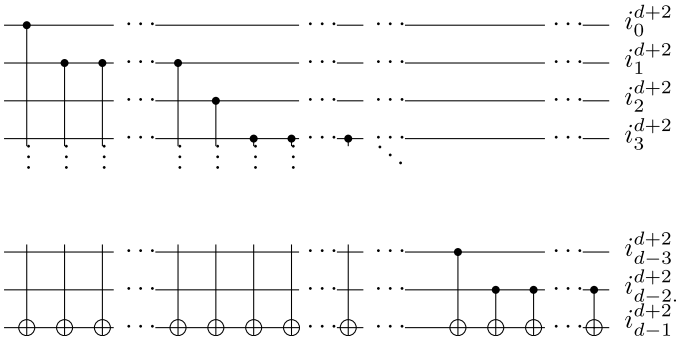

Fig. 8. Generalized SWAP gate; stage 5, step $j = d + 2$.

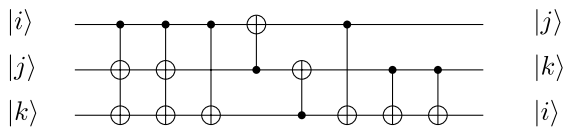

Fig. 9. A SWAP gate describing the cyclical permutation of three qutrit states. This circuit demonstrates the complete construction method for the generalized SWAP gate presented in Theorem 1 for the special case when $d = 3$. The network also provides a succinct overview of all the ideas of the general case.

$$\begin{aligned}
&= i^1_{k-3} + i^0_{k-2} + i^0_{k-1} + i^0_k \\
&= \cdots \\
&= i^1_0 + i^0_1 + i^0_2 + \cdots + i^0_{k-3} + i^0_{k-2} + i^0_{k-1} + i^0_k \\
&= \sum_{m=0}^{k} i^0_m. \qquad (4)
\end{aligned}$$

Next, stage 2 step $j = 2$, repeats the set of gates of step 1. By definition $i^2_0 = i^1_0 = i^0_0$. The case for $k = 1, \ldots, d-1$ follows from the algorithm step,

$$\begin{aligned}
i^2_k &= i^2_{k-1} + i^1_k \\
&= i^2_{k-2} + i^1_{k-1} + i^1_k \\
&= \cdots \\
&= i^2_0 + i^1_1 + i^1_2 + \cdots + i^1_{k-2} + i^1_{k-1} + i^1_k \\
&= i^0_0 + \sum_{m=0}^{1} i^0_m + \sum_{m=0}^{2} i^0_m + \cdots + \sum_{m=0}^{k} i^0_m \\
&= \sum_{l=0}^{k} \sum_{m=0}^{l} i^0_m \\
&= \sum_{m=0}^{k} \sum_{l=m}^{k} i^0_m \\
&= \sum_{m=0}^{k} \sum_{l=0}^{k-m} i^0_m \\
&= \sum_{m=0}^{k} \binom{k-m+1}{1} i^0_m. \qquad (5)
\end{aligned}$$

For steps $j = 1, \ldots, d-1$, we show by induction that

$$i^j_k = \sum_{m=0}^{k} \binom{k-m+j-1}{j-1} i^0_m, \quad k = 0, \ldots, d-1.$$

We have shown that this is true for $j = 1$. Let $1 \leq j < d-1$ and suppose that

$$i^j_k = \sum_{m=0}^{k} \binom{k-m+j-1}{j-1} i^0_m, \qquad (6)$$

$k = 0, \ldots, d-1$. Now, $i_0^{j+1} = i_0^j = i_0^0$. For $1 \leq k \leq d-1$, we have

$$\begin{aligned} i_k^{j+1} &= i_k^j + i_{k-1}^{j+1} \\ &= i_k^j + i_{k-1}^j + i_{k-2}^{j+1} \\ &= \cdots \\ &= i_k^j + i_{k-1}^j + \cdots + i_2^j + i_1^j + i_0^{j+1} \\ &= i_k^j + i_{k-1}^j + \cdots + i_2^j + i_1^j + i_0^j. \end{aligned} \tag{7}$$

Since $i_0^{j+1} = i_0^j$ follows from the algorithm step, we have that $i_k^{j+1} = \sum_{m=0}^{k} i_m^j$. Hence, by the induction process,

$$\begin{aligned} i_k^{j+1} = \sum_{m=0}^{k} i_m^j &= i_0^0 + \sum_{l=0}^{1} \binom{1-l+j-1}{j-1} i_l^0 + \sum_{l=0}^{2} \binom{2-l+j-1}{j-1} i_l^0 + \cdots \\ &\quad + \sum_{l=0}^{k} \binom{k-l+j-1}{j-1} i_l^0 \\ &= \sum_{m=0}^{k} \sum_{l=0}^{m} \binom{m-l+j-1}{j-1} i_l^0 \\ &= \sum_{m=0}^{k} \sum_{l=0}^{k} \binom{m-l+j-1}{j-1} i_l^0 \\ &= \sum_{l=0}^{k} \sum_{m=l}^{k} \binom{m-l+j-1}{j-1} i_l^0 \\ &= \sum_{l=0}^{k} \binom{k-l+j}{j} i_l^0. \end{aligned} \tag{8}$$

Therefore, the induction step is true for $j+1$,

$$i_k^{j+1} = \sum_{m=0}^{k} \binom{k-m+j}{j} i_m^0, \tag{9}$$

and the result for stage 2 follows.

The algorithm at stage 3, step $j = d$ yields that

$$\begin{aligned} i_0^d &= i_0^{d-1} = i_0^0, \\ i_1^d &= i_1^{d-1} = \sum_{m=0}^{1} \binom{1-m+d-2}{d-2} i_m^0 = (d-1) i_0^0 + i_1^0. \end{aligned} \tag{10}$$

Implementing the algorithm step $i_k^d = i_{k-2}^d + i_k^{d-1}$ for $k = 2, \ldots, d-1$, we have that

$$\begin{aligned} i_2^d &= i_0^d + i_2^{d-1} = i_0^0 + \sum_{m=0}^{2} \binom{2-m+d-2}{d-2} i_m^0, \\ i_3^d &= i_1^d + i_2^{d-1} = \sum_{m=0}^{1} \binom{1-m+d-2}{d-2} i_m^0 + \sum_{m=0}^{3} \binom{3-m+d-2}{d-2} i_m^0, \end{aligned}$$

$$i_4^d = i_2^d + i_4^{d-1} = i_0^0 + \sum_{m=0}^{2} \binom{2-m+d-2}{d-2} i_m^0 + \sum_{m=0}^{4} \binom{4-m+d-2}{d-2} i_m^0$$
$$\cdots$$

In particular, for odd valued k,

$$\begin{aligned} i_k^d &= \sum_{t=0}^{\frac{k-1}{2}} \sum_{m=0}^{2t+1} \binom{2t+1-m+d-2}{d-2} i_m^0 \\ &= \sum_{t=0}^{\frac{k-1}{2}} \sum_{m=2t}^{2t+1} \binom{2t+1-m+d-2}{d-2} i_m^0 \quad \text{(as } d \text{ is prime)} \\ &= \sum_{t=0}^{\frac{k-1}{2}} (d-1) i_{2t}^0 + i_{2t+1}^0, \end{aligned} \tag{11}$$

and, similarly, for even valued k,

$$\begin{aligned} i_k^d &= \sum_{t=0}^{\frac{k}{2}} \sum_{m=0}^{2t} \binom{2t-m+d-2}{d-2} i_m^0 \\ &= \sum_{t=0}^{\frac{k}{2}} \sum_{m=2t}^{2t} \binom{2t-m+d-2}{d-2} i_m^0 \quad \text{(as } d \text{ is prime)} \\ &= \sum_{t=0}^{\frac{k}{2}} (d-1) i_{2t-1}^0 + i_{2t}^0. \end{aligned} \tag{12}$$

The next stage, stage 4, step $j = d+1$, of the algorithm is given by $i_k^{d+1} = i_k^d + i_{k+1}^d$ for $k = 0, \ldots, d-2$. Let us consider the value i_k^{d+1}. There are two cases to note. For even valued k, we have that

$$\begin{aligned} i_k^d &= i_{k-2}^d + i_k^{d-1} \\ &= i_{k-4}^d + i_{k-2}^{d-1} + i_k^{d-1} \\ &= \cdots \\ &= \sum_{t=0}^{\frac{k}{2}} i_{2t}^{d-1} \end{aligned} \tag{13}$$

while

$$\begin{aligned} i_{k+1}^d &= i_{k-1}^d + i_{k+1}^{d-1} \\ &= i_{k-3}^d + i_{k-1}^{d-1} + i_{k+1}^{d-1} \\ &= \cdots \\ &= \sum_{t=0}^{\lfloor \frac{k+1}{2} \rfloor} i_{2t+1}^{d-1}. \end{aligned} \tag{14}$$

Therefore, $i_k^{d+1} = \sum_{t=0}^{k+1} i_t^{d-1}$ for even valued k. Correspondingly, for odd valued k, $i_k^d = \sum_{t=0}^{\frac{k-1}{2}} i_{2t+1}^{d-1}$ while $i_{k+1}^d = \sum_{t=0}^{\frac{k+1}{2}} i_{2t}^{d-1}$ and thus $i_k^{d+1} = \sum_{t=0}^{k+1} i_t^{d-1}$. Hence, we find that

$$\begin{aligned} i_k^{d+1} &= \sum_{t=0}^{k+1} i_l^{d-1} \\ &= \sum_{l=0}^{k+1} \sum_{m=0}^{t} \binom{t-m+d-2}{d-2} i_m^0 \\ &= \sum_{m=0}^{k+1} \sum_{l=m}^{k+1} \binom{l-m+d-2}{d-2} i_m^0 \\ &= \sum_{m=0}^{k+1} \binom{k-m+d}{d-1} i_m^0 \\ &= i_{k+1}^0 \pmod{d}. \end{aligned} \tag{15}$$

For prime dimensions, $d = p$, recall that under arithmetic modulo d, the coefficients $\binom{k-m+d}{d-1}$ vanish for $m \neq k+1$. Therefore, we deduce that $i_k^{d+1} = i_{k+1}^0$ for $k = 0, \ldots,$ $d-2$. When $k = d-1$, we have

$$i_{d-1}^{d+1} = i_{d-1}^d = \sum_{t=0}^{\frac{d-1}{2}} \sum_{m=0}^{2t} \binom{2t-m+d-2}{d-2} i_m^0. \tag{16}$$

The generalized SWAP gate algorithm concludes following stage 5, step $j = d+2$ with the implementation of a sequence of gates targeted on i_{d-1}^{d+1}. For $k = 0, \ldots, d-2$, we have the result

$$i_k^{d+2} = i_k^{d+1} = i_{k+1}^0 \pmod{d}. \tag{17}$$

For $k = d-1$, the value i_{d-1}^{d+2} is given as

$$\begin{aligned} i_{d-1}^{d+2} &= i_{d-1}^{d+1} + \sum_{k=0}^{d-2} \eta_k i_k^{d+2} \\ &= \sum_{t=0}^{\frac{d-1}{2}} \sum_{m=0}^{2t} \binom{2t-m+d-2}{d-2} i_m^0 + \sum_{k=0}^{d-2} \eta_k i_k^{d+2}. \end{aligned}$$

To see that this yields the desired result (i.e. $i_{d-1}^{d+2} \bmod d = i_0^0$), we consider the value $i_{d-1}^{d+1} (\bmod\ d)$. □

Lemma 2. $i_{d-1}^{d+1} = \sum_{t=0}^{\frac{d-1}{2}} i_{2t}^0 + \sum_{t=0}^{\frac{d-1}{2}-1} (d-1) i_{2t+1}^0 (\bmod\ d)$.

Proof.

$$i_{d-1}^{d+1} = \sum_{t=0}^{\frac{d-1}{2}} \sum_{m=0}^{2t} \binom{2t-m+d-2}{d-2} i_m^0$$

$$= \sum_{m=0}^{d-1} \sum_{t=\lceil \frac{m}{2} \rceil}^{\frac{d-1}{2}} \binom{2t-m+d-2}{d-2} i_m^0. \tag{18}$$

Since $\binom{2t-m+d-2}{d-2} = 0 \bmod d$ for $t > \lceil \frac{m}{2} \rceil$ then

$$\begin{aligned} i_{d-1}^{d+1} &= \sum_{m=0}^{d-1} \binom{2\lceil \frac{m}{2} \rceil - m + d - 2}{d-2} i_m^0 \\ &= \sum_{l=0}^{\frac{d-1}{2}} i_{2l}^0 + \sum_{l=0}^{\lfloor \frac{d-2}{2} \rfloor} (d-1) i_{2l+1}^0 \pmod d. \end{aligned} \tag{19}$$

Thus, $i_{d-1}^{d+1} = \sum_{t=0}^{\frac{d-1}{2}} i_{2t}^0 + \sum_{t=0}^{\lfloor \frac{d-2}{2} \rfloor} (d-1) i_{2t+1}^0 \pmod d$. □

Finally, by definition of stage 5, we have $\sum_{k=0}^{d-2} \eta_k i_k^{d+2} = \sum_{t=0}^{\lfloor \frac{d-2}{2} \rfloor} (d-1) i_{2t+1}^{d+2} + \sum_{t=0}^{\frac{d-3}{2}} i_{2t}^{d+2}$. The value of i_{d-1}^{d+2} is then given by

$$\begin{aligned} i_{d-1}^{d+2} &= i_{d-1}^{d+1} + \sum_{k=0}^{d-2} \eta_k i_k^{d+2} \\ &= \sum_{t=0}^{\lfloor \frac{d-1}{2} \rfloor} i_{2t}^0 + \sum_{t=0}^{\lfloor \frac{d-2}{2} \rfloor} (d-1) i_{2t+1}^0 + \sum_{t=0}^{\lfloor \frac{d-2}{2} \rfloor} i_{2t+1}^0 + \sum_{t=1}^{\lfloor \frac{d-1}{2} \rfloor} (d-1) i_{2t}^0. \end{aligned} \tag{20}$$

Consequently, we have the desired result $i_{d-1}^{d+2} \pmod d = i_0^{d+2} = i_0^0$. This completes the proof of Theorem 1 ensuring that the generalized SWAP gate algorithm cyclically permutes the input sequence $i_k^0 = e_k,\ k = 0, \ldots, d-1$ to the output sequence $i_k^{d+2} = e_{k+1},\ k = 0, \ldots, d-2$ with $i_{d-1}^{d+1} = e_0$.

We now show that if the generalized SWAP gate network swaps an input basis state then the generalized SWAP gate will swap all possible d^d sequences of input states.

Theorem 2. *Let $\mathcal{A}_0, \ldots, \mathcal{A}_{d-1}$ be d-dimensional systems with bases $|e_0\rangle_j, |e_1\rangle_j, \ldots, |e_{d-1}\rangle_j,\ j = 0, \ldots, d-1$, where $e_0, \ldots, e_{d-1} \in \mathbb{Z}_d$. Let $\mathcal{A} = \mathcal{A}_0 \otimes \cdots \otimes \mathcal{A}_{d-1}$. If a network implements a generalized SWAP on each basis state $|a_0 a_1 \cdots a_{d-1}\rangle = |a_0\rangle_0 \otimes |a_1\rangle_1 \otimes \cdots \otimes |a_{d-1}\rangle_{d-1}$ of $\mathcal{A}$ where $a_0, \ldots, a_{d-1} \in \mathbb{Z}_d$ then the network implements a generalized SWAP on any input state $|\psi\rangle = |\psi_0\rangle_0 \otimes |\psi_1\rangle_1 \otimes |\psi_{d-1}\rangle_{d-1}$.*

Proof. Let $|\psi_j\rangle_j = \sum_{k_j=0}^{d-1} \alpha_{jk_j} |e_{k_j}\rangle_j,\ j = 0, \ldots, d-1$. Then

$$|\psi\rangle = \sum_{k_0=0}^{d-1} \cdots \sum_{k_{d-1}=0}^{d-1} \alpha_{0k_0} \cdots \alpha_{(d-1)k_{d-1}} |k_0 \cdots k_{d-1}\rangle. \tag{21}$$

Now,

$$\begin{aligned}\mathrm{SWAP}|\psi\rangle &= \sum_{k_0=0}^{d-1}\cdots\sum_{k_{d-1}=0}^{d-1}\alpha_{0k_0}\cdots\alpha_{(d-1)k_{d-1}}\mathrm{SWAP}|k_0\cdots k_{d-1}\rangle\\ &= \sum_{k_0=0}^{d-1}\cdots\sum_{k_{d-1}=0}^{d-1}\alpha_{0k_0}\cdots\alpha_{(d-1)k_{d-1}}|k_1\cdots k_{d-1}k_0\rangle\\ &= \sum_{k_1=0}^{d-1}\cdots\sum_{k_{d-1}=0}^{d-1}\sum_{k_0=0}^{d-1}\alpha_{1k_1}\cdots\alpha_{(d-1)k_{d-1}}\alpha_{0k_0}|k_1\cdots k_{d-1}k_0\rangle\\ &= |\psi_1\rangle_0\otimes\cdots\otimes|\psi_{d-1}\rangle_{d-2}\otimes|\psi_0\rangle_{d-1} \qquad (22)\end{aligned}$$

as required. □

As an example Fig. 10 provides the circuit design of a generalized SWAP restricted to qutrits. The quantum circuit presented comprises of ten two-qutrit CNOT gates and represents a concise summary of the main design features of the generalized SWAP gate algorithm.

4. On a Generalized SWAP Gate for d Other than Prime

In this section we consider the question of revising the algorithm construction of Sec. 3 to induce a set of cyclic permutations of d qudits for d other than prime. We show however that such a revision is not possible as it induces an unavoidable sign change in one subsystem. This question was motivated by the case $d = 4$ wherein we considered if it was possible to cyclically permute the states of four four-dimensional subsystem using only instances of the CNOT gate. We now state this section's main result.

Theorem 3. *A revision of the generalized SWAP gate algorithm given in Sec. 3 to induce a set of cyclic permutations of d qudits for d other than prime is not possible.*

Proof. Let us consider a revised quantum circuit algorithm possessing a stage 1 and stage 2 identical to the generalized swap algorithm of Sec. 3. Following Eq. (6), i.e. stage 2 of generalized SWAP algorithm with step $j = d-1$, the state of the algorithm is $i_0^{d-1} = i_0^0$, and $i_k^{d-1} = \sum_{m=0}^{k}\binom{k-m+d-2}{d-2} i_m^0$ for $k = 1,\ldots,d-1$. Next, we

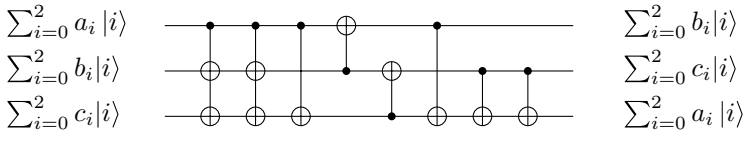

Fig. 10. The qutrit SWAP gate illustrating the cyclical permutation of three qutrit states. This network which was presented in Fig. 9 now shows that general qutrit states given in terms of computational basis can also be cyclically permuted.

seek the particular algorithm state output

$$\begin{pmatrix} i_0^0 \\ (d-1)i_0^0 + i_1^0 \\ i_0^0 + (d-1)i_1^0 + i_2^0 \\ \vdots \\ (d-1)i_0^0 + i_1^0 + (d-1)i_2^0 + \cdots + i_{d-1}^0 \end{pmatrix} \tag{23}$$

on systems $\mathcal{A}_0, \ldots, \mathcal{A}_{d-1}$, respectively. Outcome (23) is generated by the algorithm process of Sec. 3 for d prime. However, achieving outcome (23) for d other than prime requires that stage 3, step $j = d$ of Sec. 3 be revised. By revising stage 3, step $j = d$ and taking i_k^{d-1} with the following linear combination

$$\sum_{s=0}^{k-2} a_s i_{k-2-s}^{d-1} = \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+d-2}{d-2} i_m^0 \right), \tag{24}$$

where

$$a_s = d - \left[\binom{s+2+d-2}{d-2} + \sum_{t=0}^{s-1} a_t \binom{s-t+d-2}{d-2} \right] + (-1)^s, \tag{25}$$

we then obtain outcome (23).

Lemma 3. *For d other than prime, the algorithm process at stage 3, step $j = d$ given by*

$$\begin{aligned} i_k^d &= i_k^{d-1} + \sum_{s=0}^{k-2} a_s i_{k-2-s}^{d-1} \\ &= \sum_{m=0}^{k} \binom{k-m+d-2}{d-2} i_m^0 + \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+d-2}{d-2} i_m^0 \right), \end{aligned}$$

for $k = 0, \ldots, d-1$, returns outcome (23).

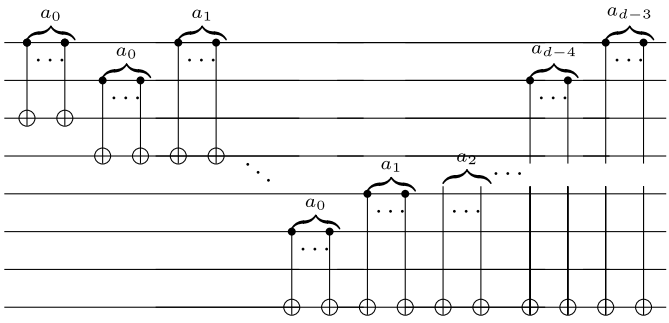

Fig. 11. Stage 3, step $j = d$. Circuit description representing the modified stage of the generalized SWAP gate for d other than prime.

Proof. For $k = 0, 1$, we have that $i_0^d = i_0^{d-1}$ and $i_1^d = i_1^{d-1}$. Thus, the states e_0 and e_1 are given as i_0^0 and $(d-1)i_0^0 + i_1^0$ respectively. The state e_2 is written as

$$\begin{aligned} i_2^d &= \sum_{m=0}^{2} \binom{2-m+d-2}{d-2} i_m^0 + a_0 i_0^0 \\ &= \left(\binom{d}{d-2} + \left(d - \binom{d}{d-2} + 1 \right) \right) i_0^0 + (d-1) i_1^0 + i_2^0 \\ &= i_0^0 + (d-1) i_1^0 + i_2^0 \pmod{d}. \end{aligned} \tag{26}$$

We show by induction that, for $k = 0 \pmod 2$,

$$\begin{aligned} i_k^d &= \sum_{m=0}^{k} \binom{k-m+d-2}{d-2} i_m^0 + \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+d-2}{d-2} i_m^0 \right) \\ &= i_0^0 + (d-1) i_1^0 + i_2^0 + \cdots + (d-1) i_{k-1}^0 + i_k^0 \pmod{d} \end{aligned} \tag{27}$$

and for $k \neq 0 \pmod 2$,

$$i_k^d = (d-1) i_0^0 + i_1^0 + (d-1) i_2^0 + \cdots + (d-1) i_{k-1}^0 + i_k^0 \pmod{d}. \tag{28}$$

We have shown that this is true for $k = 0, 1, 2$. Suppose $0 \leq k \leq d-2$ and further suppose that

$$\begin{aligned} i_k^d &= \sum_{m=0}^{k} \binom{k-m+d-2}{d-2} i_m^0 + \sum_{s=0}^{k-2} \left(a_s \sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+d-2}{d-2} i_m^0 \right) \\ &= \sum_{m=0}^{k} (-1)^{k-m} i_m^0 \\ &= i_0^0 + (d-1) i_1^0 + i_2^0 + \cdots + (d-1) i_{k-1}^0 + i_k^0 \pmod{d} \end{aligned} \tag{29}$$

for $k = 0 \pmod 2$, and

$$i_k^d = (d-1) i_0^0 + i_1^0 + (d-1) i_2^0 + \cdots + (d-1) i_{k-1}^0 + i_k^0 \pmod{d} \tag{30}$$

for $k \neq 0 \pmod 2$. Therefore, for $j = d$, we have,

$$\begin{aligned} i_{k+1}^d &= \sum_{m=0}^{k+1} \binom{k+1-m+d-2}{d-2} i_m^0 \\ &\quad + \sum_{s=0}^{k-1} \left(a_s \sum_{m=0}^{(k-1)-s} \binom{(k-1)-s-m+d-2}{d-2} i_m^0 \right) \\ &= \sum_{m=0}^{k} \left(\binom{k-m+d-2}{d-2} i_{m+1}^0 + \binom{k+1+d-2}{d-2} i_0^0 \right) \\ &\quad + \sum_{s=0}^{k-1} a_s \left(\left(\sum_{m=0}^{(k-2)-s} \binom{(k-2)-s-m+d-2}{d-2} i_{m+1}^0 \right) \right. \end{aligned}$$

$$
\begin{aligned}
&+\binom{(k-1)-s+d-2}{d-2} i_0^0\Bigg) \\
&= \sum_{m=0}^{k} (-1)^{k-m} i_{m+1}^0 \\
&\quad + \left(\binom{k+1+d-2}{d-2} + \sum_{s=0}^{k-1} a_s \binom{k-1-s+d-2}{d-2}\right) i_0^0. \qquad (31)
\end{aligned}
$$

Recall that the binomial coefficients of $i_k^{d-1} = \sum_{m=0}^{k} \binom{k-m+d-2}{d-2} i_m^0$ are precisely those coefficients of $i_{k+1}^{d-1} = \sum_{m=0}^{k+1} \binom{k+1-m+d-2}{d-2} i_m^0$ for $m = 1, \ldots, k+1$. Hence, the particular combination of systems $\mathcal{A}_{(k-2)-s}$ that return the state $i_k^d = i_0^0 + (d-1)i_1^0 + i_2^0 + \cdots + (d-1)i_{k-1}^0 + i_k^0 \pmod d$ is the combination that yields the similar sequence on i_{k+1}^d for $m = 1, \ldots, k+1$. With $k = 0 \pmod 2$, then for i_{k+1}^d we require that the scalar value for i_0^0 degenerates to $d-1 \pmod d$. Thus, for $m = 0$ and by definition of a_s, we have

$$
\begin{aligned}
&\left(\binom{(k+1)+d-2}{d-2} + \sum_{s=0}^{k-1} a_s \binom{(k-1)-s+d-2}{d-2}\right) i_0^0 \\
&\quad = \left(\binom{(k+1)+d-2}{d-2} + \sum_{s=0}^{k-2} \left(a_s \binom{(k-1)-s+d-2}{d-2}\right) + a_{k-1}\right) i_0^0 \\
&\quad = \Bigg(\binom{(k+1)+d-2}{d-2} + \sum_{s=0}^{k-2} a_s \binom{(k-1)-s+d-2}{d-2} \\
&\qquad + \left(d - \left[\binom{(k+1)+d-2}{d-2} + \sum_{s=0}^{k-2} a_s \binom{(k-1)-s+d-2}{d-2}\right] + (-1)^{k+1}\right)\Bigg) i_0^0 \\
&\quad = (-1)^{k+1} i_0^0 \pmod d. \qquad (32)
\end{aligned}
$$

Hence, $i_{k+1}^d = \sum_{m=0}^{k+1} (-1)^{k+1-m} i_m^0 \pmod d$, and the result follows. □

We continue with stage 4 of the generalized SWAP gate algorithm of Sec. 3;

$$
i_k^{d+1} = i_k^d + i_{k+1}^d \qquad (33)
$$

for $k = 0, \ldots, d-2$ and

$$
i_{d-1}^{d+1} = i_{d-1}^d + \sum_{m=0}^{d-2} (-1)^{d-1-s} i_m^0. \qquad (34)
$$

Finally, revising stage 5 of Sec. 3 so that $\sum_{k=1}^{d-1} \eta_k^* i_k^{d+2} = \sum_{t=0}^{\lfloor \frac{d-1}{2} \rfloor} (d-1) i_{2t+1}^0 + \sum_{t=0}^{\frac{d-2}{2}} i_{2t}^0$, we obtain the state

$$
\left(i_1^0, i_2^0, i_3^0, \ldots, i_{d-1}^0, (d-1)i_0^0\right). \qquad (35)
$$

Unfortunately, we have not achieved a generalized SWAP for d other than prime as the revised algorithm has unavoidably induced a sign change, i.e. $-1 \pmod d$, in the last subsystem $\mathcal{A}_{d-1}$.

Indeed, the following argument now shows that a different algorithm would be required. On obtaining the outcome (35), for d other than prime, no sequence of generalized CNOT gates will return the desired cyclic permutation. To show this claim, consider the more general case of outcome (35) given by the revised algorithm;

$$\big(\xi i_1^0, \xi i_2^0, \xi i_3^0, \ldots, \xi i_{d-1}^0, (d-\xi) i_0^0\big). \tag{36}$$

Consider the pairs $(\xi i_k^0, \xi i_{k+1}^0)$ for $k \in \{1, \ldots, d-3\}$ together with the final pair $(\xi i_{d-1}^0, (d-\xi)\xi i_0^0)$. Given $(\xi i_k^0, \xi i_{k+1}^0)$ for $k \in \{1, \ldots, d-3\}$, and a CNOT mapping that targets e_{k+1}, we have that $(\xi i_k^0, \xi i_{k+1}^0) \mapsto (\xi i_k^0, \xi i_k^0 + \xi i_{k+1}^0)$. Denote by P_ξ the inverse of $\xi \pmod d$, whence, $P_\xi \xi = 1 \pmod d$. Applying $P_\xi - 1$ gates, see Fig. 12, to target ξi_k^0 in each pair yields

$$\begin{aligned}(\xi i_k^0, \xi i_k^0 + \xi i_{k+1}^0) \mapsto\ & (P_\xi \xi i_k^0 + (P_\xi - 1)\xi i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0) \\ &= (i_k^0 + (1-\xi) i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0).\end{aligned} \tag{37}$$

In eliminating ξi_k^0 in Eq. (37), we apply $d - \xi$ gates that target $\xi i_k^0 + \xi i_{k+1}^0$;

$$\begin{aligned}(i_k^0 + (1-\xi) i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0) \mapsto\ & (i_k^0 + (1-\xi) i_{k+1}^0, \xi i_k^0 + \xi i_{k+1}^0 + (d-\xi)(i_k^0 \\ & + (1-\xi) i_{k+1}^0) \\ &= (i_k^0 + (1-\xi) i_{k+1}^0, \xi i_{k+1}^0 + (d-\xi)(1-\xi) i_{k+1}^0) \\ &= (i_k^0 + (1-\xi) i_{k+1}^0, \xi i_{k+1}^0 + (-\xi + \xi^2) i_{k+1}^0) \\ &= (i_k^0 + (1-\xi) i_{k+1}^0, \xi^2 i_{k+1}^0).\end{aligned} \tag{38}$$

Similarly, applying the set gates as outlined in Figs. 12 and 13 to the pair $(\xi i_{d-1}^0, (d-\xi) i_0^0)$, we obtain $(i_{d-1}^0 + (\xi - 1) i_0^0, -\xi^2 i_0^0)$. Thus, we have

$$\begin{aligned}&\big(\xi i_1^0, \xi i_2^0, \ldots, \xi i_{d-3}^0, \xi i_{d-2}^0, \xi i_{d-1}^0, (d-\xi) i_0^0\big) \mapsto \\ &\big(i_1^0 + (1-\xi) i_2^0, \xi^2 i_2^0, \ldots, i_{d-3}^0 + (1-\xi) i_{d-2}^0, \xi^2 i_{d-2}^0, i_{d-1}^0 + (\xi - 1) i_0^0, -\xi^2 i_0^0\big).\end{aligned} \tag{39}$$

Since the scalar values ξ^2 and $-\xi^2$ cannot be both 1 $(\text{mod } d)$, it seems that any mapping will fail to return a state with scalars all equal to unity. This suggests that a generalized SWAP gate composed entirely in terms of generalized CNOT gates may not be possible for d other than prime. This completes the proof of the theorem. □

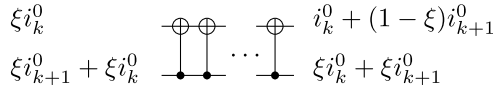

Fig. 12. $P_\xi - 1$ generalized CNOT gates on pairs (e_k, e_{k+1}), $k \in \{0, \ldots, d-3\}$.

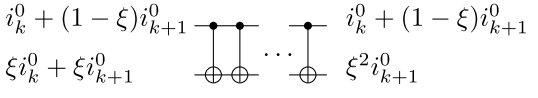

Fig. 13. $d - \xi$ generalized CNOT gates on on pairs (e_k, e_{k+1}), $k \in \{0, \ldots, d-3\}$.

5. Conclusion

We discussed the construction of a generalized SWAP gate that cyclically permutes the states of d qudit subsystems for d prime. The design restricted itself to only using instances of the generalized CNOT gate, and the analysis made great use of modular binomial relationships. Lastly, we illustrated how the generalized SWAP gate design may be revised to yield certain permutations of d qudits for d other than prime.

Acknowledgments

It is a pleasure to acknowledge the advice and help of Prof. Matthew G. Parker and Prof. Rüdiger Schack.

References

1. A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin and H. Weinfurter, *Phys. Rev. A* **52**(5) (1995) 3457.
2. F. Vatan and C. Williams, *Phys. Rev. A* **69**(3) (2004) 032315.
3. A. G. Fowler, S. J. Devitt and L. C. L. Hollenberg, *Quantum Inf. Comput.* **4** (2004).
4. L. Liang L and C. Li, *Phys. Rev. A* **72**(2) (2005) 024303.
5. M. Grassl, M. Rötteler and T. Beth, *Int. J. Found. Comput. Sci.* **14** (2003) 757.
6. C. M. Wilmott, *Int. J. Quant. Inform.* **9**(6) (2011) 1511.
7. K. H. Rosen and J. G. Michaels, *Handbook of Discrete and Combinatorial Mathematics* (CRC Press, 2000).