Necessary Conditions and Empirical Observations for Rearrangeable Banyan-Type Networks

SATORU OHTA
Department of Information Systems Engineering, Faculty of Engineering,
Toyama Prefectural University,
5180 Kurokawa, Imizu-shi, Toyama 939-0398,
JAPAN

Abstract: - A banyan-type network is constructed by aligning unit switches with two inlets and outlets in multiple stages. Rearrangeable banyan-type networks are crucial for applications such as communication systems because they can universally establish connections for any request without blocking. If the number of network inputs (or outputs) is 2^n ($n > 0$), the banyan-type network should have $2n - 1$ or more stages to be rearrangeable. A few rearrangeable $2n - 1$ stage networks have been reported. However, the class of rearrangeable $2n - 1$ stage banyan-type networks has not been completely clarified. This study examines the identification of rearrangeable $2n - 1$ stage banyan-type networks that are not isomorphic to one another. This is done by generating candidate networks and checking their rearrangeability via the satisfiability problem. The drawback of this approach is its poor scalability due to numerous candidates. To eliminate this drawback, it is shown that the candidates can be reduced to a smaller number of networks called pure banyan networks. This is achieved by analyzing network isomorphism. Next, necessary conditions are derived for rearrangeability. Utilizing the conditions, the number of candidate networks further decreases because blocking networks are identified and removed from the candidates. For the reduced number of candidate networks, rearrangeability is assessed through computer experiments for n = 4 and 5. For n = 4, the result shows that any rearrangeable configuration is isomorphic to previously reported rearrangeable networks. For n = 5, the blocking probability is extremely low and the rearrangeability is inconclusive for two groups of networks.

Key-Words: - switching network, nonblocking network, banyan network, rearrangeable network, graph isomorphism, satisfiability problem.

Received: February 6, 2023. Revised: November 17, 2023. Accepted: December 22, 2023. Published: December 31, 2023.

1 Introduction

Switching networks, [1], [2], [3] are indispensable as components of various information and communication systems, [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. Switching networks are configured by placing small switches in multiple stages. This configuration flexibly provides connectivity between numerous inputs and outputs.

A well-known class of switching networks comprises multistage unit switches, each of which has two inlets and outlets. Various networks of this type have been reported in the literature with different names, [16], [17], [18], [19], [20], In this study, we refer to them as banyan-type networks.

For some applications, a switching network is required to be nonblocking. Nonblocking banyan-type networks can be classified as rearrangeable networks, [1]. Let $N = 2^n$ be the number of inputs (or outputs) in a banyan-type network. Then, the number of stages should be greater than or equal to $2n - 1$ for the network to be rearrangeable. Actually, there are a few known rearrangeable $2n - 1$ stage networks. However, the class of rearrangeable $2n - 1$ stage banyan-type networks has not been clarified, despite numerous previous related studies, [21], [22], [23], [24], [25], [26], [27], [28], [29]. The clarification of rearrangeable banyan-type networks is essential because it enables us to choose the best configuration from all possible networks for an application. For example, if we find the best configuration for the photonic switching application, it will become easier to implement a large-scale photonic switch. This enables us to utilize higher transmission bandwidth in communication services through photonic networks. It is also a theoretically interesting challenge to search for previously unreported rearrangeable banyan-type networks.

In a banyan-type network, the set of links connecting adjacent stages is called an exchange. The link configuration rule of an exchange is represented by a cyclic bit-position permutation of indices assigned to switch inlets and outlets. A banyan-type network is specified by which permutation is applied to each exchange.

Isomorphism should be considered when assessing the rearrangeability of a banyan-type network. A given banyan-type network can be transformed into another by exchanging the positions of switches and redrawing links. Then, the original and transformed networks are isomorphic and thus equivalent in terms of their rearrangeability. Suppose that two or more networks are isomorphic. Then, it is unnecessary to assess the rearrangeability for all of them; it is sufficient to only check the rearrangeability for any one of them.

In this study, an efficient method is proposed for searching for rearrangeable banyan-type networks by improving the method presented in, [30] (hereinafter, the base method). In the base method, a set of bit-position permutations is first assumed, and candidate networks are exhaustively generated by assigning elements of the permutation set to exchanges. Then, it is checked whether connections can be established in each candidate network for a given connection request set. This is achieved by solving a satisfiability (SAT) problem that models the routing of connections, [31]. By repeating the above procedure, if connections are blocked for a certain request set, the network is a blocking network, whereas if connections can always be established for many request sets, the network is likely to be rearrangeable. The base method is disadvantageous for scalability because the number of generated candidate networks grows unacceptably large for a large n value. Therefore, this study effectively reduces the number of candidate networks.

The major contributions of this article are as follows.

(1) It is shown that any banyan-type network configured by bit-position permutations commonly used in the reported networks is isomorphic to a configuration referred to as a "pure banyan network" (Theorem 1). The number of pure banyan networks is significantly less than that of the possible candidate networks examined in, [30].

(2) Necessary conditions for the rearrangeability of pure banyan networks are derived (Theorems 2 and 3). The conditions suggest that a considerable number of pure banyan networks are blocking networks. This further reduces the number of candidate networks to be assessed for rearrangeability.

(3) By investigating the Beneš network structure, a class of blocking networks is revealed (Theorem 4). By discovering the isomorphism between this network class and candidate networks, it is possible to further filter out blocking networks.

(4) The rearrangeability of pure banyan networks is assessed by computer experiments for $n = 4$ and 5. For $n = 4$, it is shown that only 36 networks are sufficient to be checked although 117,649 candidates were tested in, [30]. For $n = 5$, candidate networks were categorized into 11 groups, each comprising isomorphic networks. The networks of these groups were tested, and the networks of nine groups were blocking networks. However, the rearrangeability is inconclusive for the two other groups.

The remainder of this article is organized as follows. In Section 2, previous related studies are reviewed. In Section 3, we define a banyan-type network and explore the isomorphism, rearrangeability, terminology, and definitions as preliminaries. The characteristics of isomorphism are investigated in Section 4, wherein a previously reported banyan-type network is shown to be isomorphic to a pure banyan network. In Section 5, the necessary conditions for the rearrangeability of a pure banyan network are derived. By observing the Beneš network structure, a condition for blocking networks is also derived. Section 6 presents the empirical results of computer experiments. Finally, Section 7 concludes this study.

2 Related Work

Many studies have been conducted on multistage switching networks comprising unit switches, each of which has two inlets and outlets. This category of switching networks includes shuffle-exchange network [16], omega network [17], n-cube network [18], banyan network [19], and baseline network [20]. It is also possible to consider that the Beneš network [1], [2] falls into this category. In this study, we refer to these networks as banyan-type networks.

Banyan-type networks have various applications. They were first proposed and studied mainly for parallel processing, [16], [17], [18], [19], [20]. For this application, the networks were employed to connect multiple processors and memories. Banyan-type networks are well suited for this purpose because of their control simplicity. Then, studies based on fast packet-switching technology were conducted to apply them to communication systems, [13], [14], [15]. In a node of communication networks, a banyan-type network can be employed to interconnect network interfaces. For communication system applications, the advantages of banyan-type networks include their self-routing nature and modularity. Recently, banyan-type networks have been often employed in photonic switching systems, [4], [5], [6], [7]. A banyan-type network is suitable for photonic switches because it is relatively easy to implement a unit switch using phase shifters, [7]. Another recent application of banyan-type networks is in decoding LDPC (low-density parity-check) code, [8], [9], [10], which is a high-performance error-correcting code. Banyan-type networks have also been applied to image encryption, [11] and neural networks [12].

For applications such as communication systems, switching networks should be nonblocking to always satisfy arbitrary connection requests between idle inputs and outputs. If a banyan-type network is nonblocking, it is necessarily rearrangeable, [1].

An interesting topic is the number of stages required for rearrangeability. The total number of connection request sets is $N!$. As reported in, [21],

$$\log_2 N \approx N \log_2 N - N + 1, \text{ for } N \gg 1.$$

Meanwhile, if $N/2$ unit switches are aligned in s stages and each switch takes one of the two connection states, the number of possible network states has an upper bound of $2^{sN/2}$. Thus, the following inequality is needed for rearrangeability:

$$sN / 2 > N \log_2 N - N + 1,$$
$$\therefore s > 2n - 2 + 2^{-n+1}.$$

From the above relation, a rearrangeable banyan-type network must have $2n - 1$ or more stages.

For the number of stages needed for rearrangeability, [22], considered a network where the same permutation is applied to every exchange between stages. This structure is observed in shuffle-exchange and omega networks. For this structure, let d denote the minimum number of links that one must go through to reach all unit switches of a stage from a switch of the first stage. In, [22], it is conjectured that $2d + 1$ stages are necessary and sufficient for the network to be rearrangeable. For the shuffle-exchange network, $d = n - 1$. This means that the $2n - 1$ stage shuffle-exchange network is rearrangeable.

As, [23], reported, numerous scholars have attempted to prove the conjecture of, [22]. In particular, [24], claimed to have proved the conjecture, but it has been reported that the proof is unreliable, [25]. Nevertheless, it appears that the conjecture is correct for any value of n. At least, the rearrangeability of the $2n - 1$ shuffle-exchange network has been confirmed for $n = 4$, [26], [27].

In the, [28], it is investigated the least cost rearrangeable network and showed that it should have as many stages as possible and as few unit switches as possible. This principle leads to a $2n - 1$ stage rearrangeable network, which is currently known as the Beneš network.

The, [29], derived a proof method for rearrangeability and proposed $2n - 1$ stage rearrangeable networks, referred to as the reduced $\Omega_N \Omega_N^{-1}$. However, these networks are isomorphic to the Beneš network. Thus, the work of, [29], does not essentially reveal previously unknown rearrangeable networks.

Despite considerable research effort, the class of rearrangeable banyan-type networks has not been completely clarified. Rearrangeability is often proved using an algorithm that determines the connection routes. However, it is not easy to derive such algorithms for all possible network configurations. In, [30], authors tackled this problem using a completely different approach and succeeded in showing a class of rearrangeable networks for n = 3 and 4. They modeled the routing in a network into an SAT problem according to the formulation method detailed in, [31]. The SAT problem approach eliminates the need to develop a routing algorithm for any individual switching network. It is possible to check the rearrangeability by solving SAT problems for many connection request sets. They also checked the isomorphism between each network and known rearrangeable networks. For $n = 4$, they found 2,600 rearrangeable networks from 117,649 candidate networks. These are isomorphic to known rearrangeable networks

An obvious shortcoming of the aforementioned approach is its scalability. The number of candidate networks grows unacceptably large if $n > 4$. However, among the networks generated in, [30], a considerable number of networks are trivially blocking networks. In addition, isomorphic networks have the same characteristics for rearrangeability. Therefore, it is unnecessary to assess multiple isomorphic networks. By addressing these points, it will be possible to efficiently reduce the number of candidate networks and find rearrangeable networks for $n > 4$.

3 Preliminaries

This section introduces banyan-type networks and presents some key concepts such as rearrangeability as well as explores some mathematical definitions needed to describe the network characteristics.

3.1 Banyan-Type Networks

A banyan-type network has a structure, which is constructed by aligning unit switches with two inlets and outlets in multiple stages. For this structure, the number of network inputs (or outputs) is 2^n because 2^{n-1} unit switches are placed in each stage. As examples of such networks, Figure 1 shows the omega and banyan networks. Also, Figure 1 (c) shows an example of the practical electronics design of a unit switch.

In a banyan-type network, the link configuration between two adjacent stages is called an exchange. The link configuration between the inputs and the first stage or that between the last stage and outputs is also an exchange. The link configuration rule of an exchange is represented by a cyclic bit-position permutation of inlet/outlet indices, as detailed in Subsection 3.4.

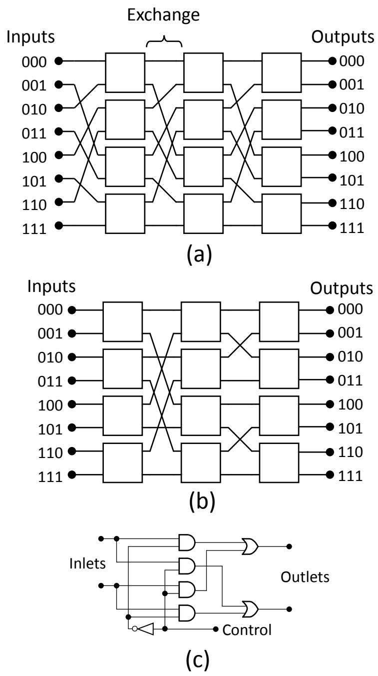

Fig. 1: Examples of banyan-type networks: (a) omega network (b) banyan network, and (c) the practical electronics design of a unit switch.

3.2 Rearrangeability

Switching networks are categorized as blocking and nonblocking networks. For some applications, nonblocking networks are indispensable. Nonblocking networks are further classified into several groups, [1], [2], [3]. A class of nonblocking networks is rearrangeable networks. In a rearrangeable network, a requested connection may be blocked. However, the system can always be unblocked by rearranging the routes of existing connections. A nonblocking banyan-type network is rearrangeable, [1].

As described in Section 2, $2n-1$ or more stages are necessary for a banyan-type network to be rearrangeable. Known $2n-1$ stage rearrangeable networks include the Beneš network (Figure 2) and the $2n-1$ stage shuffle-exchange network. The omega network is trivially isomorphic to the shuffle-exchange network because the former is obtained by swapping the inputs and outputs of the latter. Thus, the $2n-1$ stage omega network is also rearrangeable.

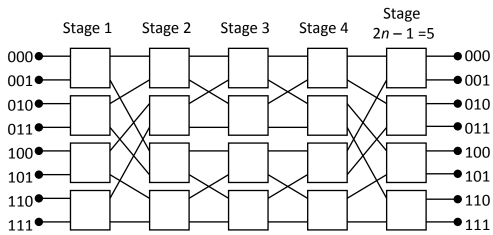

Fig. 2: Example of the Beneš network: rearrangeable $2n-1$ stage network.

As a tool to assess rearrangeability, [31], proposed the SAT problem approach. In this approach, the conditions that must be satisfied by the connection routes are modeled in the conjunctive normal form (CNF) of Boolean variables. The CNF-SAT problem determines the existence of variable values that satisfy the required conditions, [32]. In a rearrangeability assessment, a CNF-SAT problem is formulated for a network, and a set of connection requests is given between the inputs and outputs. The set of requested connections is represented as a permutation of output indices. If the solution to a problem is unsatisfiable for a given connection request set, the network is a blocking network. If the solution is always satisfiable for many connection request sets, the network is likely to be rearrangeable.

3.3 Isomorphism

To identify rearrangeable banyan-type networks, it is crucial to consider isomorphism between networks. Figure 3 shows an example of two isomorphic networks. The network in Figure 3 (a) is distinct from that in Figure 3 (b) because of the difference in the bit-position permutation assigned to exchange 1. However, suppose that the positions of switches 01 and 10 are interchanged at stage 1 in the network in Figure 3 (a). Then, it is easy to see that the two networks are equivalent.

Trivially, if a banyan-type network can satisfy an arbitrary connection request, its isomorphic network can also satisfy an arbitrary connection request. Meanwhile, if blocking occurs in a network for some connection request sets, blocking also occurs in its isomorphic network. Suppose that we are checking the rearrangeability of multiple isomorphic networks. Then, it is unnecessary to assess the rearrangeability of each network; it is sufficient to check for any one of them.

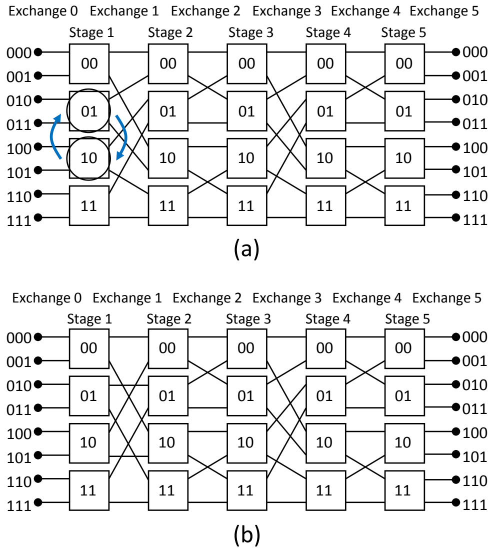

Fig. 3: Isomorphic banyan-type networks. The network of (a) becomes equivalent to the network of (b) by exchanging the unit switch positions.

Isomorphism can be assessed by using a graph isomorphism algorithm, [33], as described in, [30].

3.4 Definitions

Figure 4 shows an example of a banyan-type network where n = 3 and s = 3. The figure also depicts how each inlet/outlet of a unit switch is indexed by an n-bit binary number. Every input/output of the network is also indexed by a binary number. Each stage has 2^{n-1} unit switches indexed by ($n-1$)-bit binary numbers. The indices 00…0, …, 11…1 are given by the direction from the top to the bottom. Because of this indexing rule, the two inlets (or outlets) of a unit switch $b_1b_2\ldots b_{n-1}$ are $b_1b_2\ldots b_{n-1}0$ and $b_1b_2\ldots b_{n-1}1$, where b_j = 0 or 1 $(1 \le j \le n-1)$.

A set of links connecting two adjacent stages is an exchange. Exchanges are also defined between the inputs and stage 1 as well as between stage s and the outputs. The exchange between stages k and $k+1$ is referred to as exchange k $(1 \le k \le s-1)$. Exchange 0 is the link set between the inputs and stage 1, while exchange s is that between stage s and the outputs.

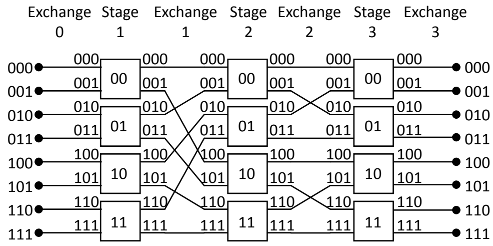

Fig. 4: Banyan-type network with indices for network inputs/outputs, unit switches, and inlets/outlets of unit switches.

The connecting rule of an exchange is represented by a cyclic permutation of bit positions of an inlet/outlet index. A cyclic permutation is represented by a format such as (2 3 4) or (1 5). In Figure 4, the permutation of exchange 1 is (1 2 3). This permutes the first bit to the second, the second bit to the third, and the third bit to the first. Thus, stage 2 outlet 101 is connected to stage 3 inlet 110, permuted from 101 according to (1 2 3).

Notably, the permutations of exchanges 0 and s do not affect rearrangeability [30]. Thus, these permutations can be ignored to assess rearrangeability.

A banyan-type network is specified by the permutations applied to exchanges 0, 1,..., s. Based on this, the notation format used in, [25], [30] is employed to specify a banyan-type network. The format is defined as follows:

$$[p_0 : p_1 : \ldots : p_s]_n,$$

where p_k is the permutation applied to exchange k $(0 \le k \le s)$. If no bits are permuted in an exchange, the symbol id is used. With this notation, the network in Figure 4 is described as follows:

$$[id : (1\ 2\ 3) : (2\ 3) : id]_3.$$

Let $\sigma(i)$ denote the binary number obtained by moving the bit positions of a binary number i according to a permutation σ. For example, if σ is (1 2 3) and i = 101, $\sigma(i)$ = 110.

By the cascade of two permutations σ and τ, a binary number i is converted to $\tau(\sigma(i))$. If $\tau(\sigma(i)) = i$ for an arbitrary i, τ is the inverse of σ and denoted by σ^{-1}. For example, if σ is (4 3 2), its inverse σ^{-1} is (2 3 4).

There exist various bit-position permutations, which may be applied to banyan-type networks. However, a limited number of permutations have been employed in the reported configurations. A banyan network employs permutations such as (1 4) and (2 4). Permutations such as (1 2 3 4) and (2 3 4) are used in a baseline network. In a shuffle-exchange network, the permutation is (4 3 2 1). The permutations used in the reported banyan-type networks can be categorized as follows.

(1) Banyan permutation: defined as $(m\ \ n)$ and used in banyan networks, where $1 \le m < n$.
(2) Baseline permutation: defined as $(m\ \ m+1\ \ldots\ n)$ and used in baseline networks.
(3) Reverse baseline permutation: defined as $(n\ \ n-1\ \ldots\ m+1\ \ m)$ and used in shuffle-exchange or Beneš networks.

Although it is also possible to employ other permutation types, no reported configurations have used other permutations, such as (1 3 4) or (1 3 2 4). Therefore, we focus on configurations constructed using the above three permutation types.

4 Isomorphism Characteristics

Section 3 shows that different but isomorphic banyan-type networks become identical by interchanging the positions of unit switches in a stage. This section specifies the rule for such position interchange by the insertion of bit permutations between a stage and its adjacent exchanges. The next lemma describes the method.

Lemma 1: Let σ denote a permutation that does not include n (e.g., (4 3 2) when n = 5). For a banyan-type network X, the network X' is constructed by

· inserting σ between exchange $k-1$ and the inlets of stage $k\,(1 \le k \le s)$ and

· inserting σ^{-1} between the outlets of stage k and exchange k.

Then, X and X' are isomorphic.

Proof: Let τ_{k-1} and τ_k denote the permutations assigned to exchanges $k-1$ and k in X, respectively. Then, two τ_{k-1} outlets, $b_1b_2\cdots b_{n-1}0$ and $b_1b_2\cdots b_{n-1}1$, are connected to two inlets of switch $b_1b_2\cdots b_{n-1}$ at stage k $(b_j \in \{0,1\}, j = 1,2,\ldots,n-1)$. By inserting σ, $b_1b_2\cdots b_{n-1}0$ is redirected to $b'_1b'_2\cdots b'_{n-1}0$ in X'. Because σ does not include n, the last bit 0 does not change. Similarly, $b_1b_2\cdots b_{n-1}1$ is redirected to $b'_1b'_2\cdots b'_{n-1}1$ in X' because the first $n-1$ bits of $b_1b_2\cdots b_{n-1}1$ are identical to that of $b_1b_2\cdots b_{n-1}0$, and the last bit is unchanged. Therefore, the two τ_{k-1} outlets are redirected and connected to the same switch $b'_1b'_2\cdots b'_{n-1}$ after the insertion of σ in X'. In X', the outlets of the switch $b'_1b'_2\cdots b'_{n-1}$ are $b'_1b'_2\cdots b'_{n-1}0$ and $b'_1b'_2\cdots b'_{n-1}1$. By the insertion of σ^{-1}, these outlets are connected to two τ_k inlets, $b_1b_2\cdots b_{n-1}0$ and $b_1b_2\cdots b_{n-1}1$. Thus, the τ_{k-1} outlets are connected to the τ_k inlets through a single unit switch. This means that the relation between τ_{k-1} and τ_k does not change before and after the insertion of σ and σ^{-1}. By reassigning the switch index from $b'_1b'_2\cdots b'_{n-1}$ to $b_1b_2\cdots b_{n-1}$, X' becomes identical to X. Therefore, X' is isomorphic to X. □

An illustration of the proof of Lemma 1 is shown in Figure 5. Clearly, the network X' in Figure 5 (b) coincides with network X in Figure 5 (a).

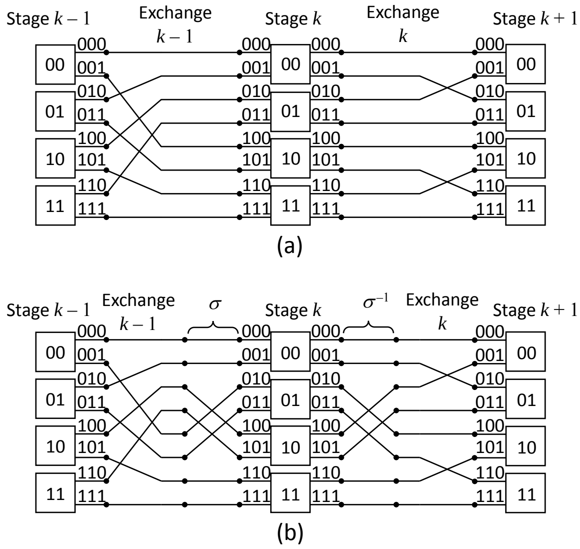

Fig. 5: Explanation of Lemma 1: (a) stage k of network X and (b) stage k of an isomorphic network X′, constructed by inserting σ and σ^{-1}.

In the proof of Lemma 1, if the unit switch index is not rewritten, the permutations of exchanges $k-1$ and k are modified to permutation cascades. The following lemmas state the characteristics of the cascaded permutations.

Lemma 2: Let τ denote a baseline permutation $(m\ \ m+1\ \ldots\ n)$, where $1 \le m < n$, and σ denote permutation $(n-1\ \ n-2\ \ldots\ m)$. Then, the cascade of σ and τ is a banyan permutation $(m\ \ n)$.

Proof: Define the binary number $i = b_1b_2\cdots b_n$. Then, $\sigma(i)$ is $b_1b_2\cdots b_{m-1}b_{m+1}\cdots b_{n-1}b_mb_n$. Applying τ to this, $\tau(\sigma(i))$ is $b_1b_2\cdots b_{m-1}b_nb_{m+1}b_{m+2}\cdots b_{n-1}b_m$. This is the number obtained by interchanging the m-th and n-th bits of i. Thus, the cascaded permutation is $(m\ \ n)$. □

Lemma 3: Let τ denote a reverse baseline permutation $(n\ \ n-1\ \ldots\ m)$, where $1 \le m < n$, and σ denote permutation $(m\ \ m+1\ \ldots\ n-1)$. Then, the cascade of σ and τ is a banyan permutation $(n-1\ \ n)$.

Proof: Define the binary number $i = b_1b_2\cdots b_n$. Then, $\sigma(i)$, is $b_1b_2\cdots b_{m-1}b_{n-1}b_mb_{m+1}\cdots b_{n-2}b_n$. Applying τ to this, $\tau(\sigma(i))$ is $b_1b_2\cdots b_{m-1}b_mb_{m+1}b_{m+2}\cdots b_{n-2}b_nb_{n-1}$. This is the number obtained by interchanging the $(n-1)$-th and n-th bits of i. Thus, the cascaded permutation is $(n-1\ \ n)$. □

Figure 6 illustrates an example to explain Lemma 2. Figure 6 (a) shows permutation $\tau = (1\ \ 2\ \ 3\ \ 4)$. By cascading $\sigma = (3\ \ 2\ \ 1)$ and τ, the exchange becomes the configuration in Figure 6 (b). By clearing the link drawing, the configuration is as in Figure 6 (c). This configuration is clearly a banyan permutation (1 4).

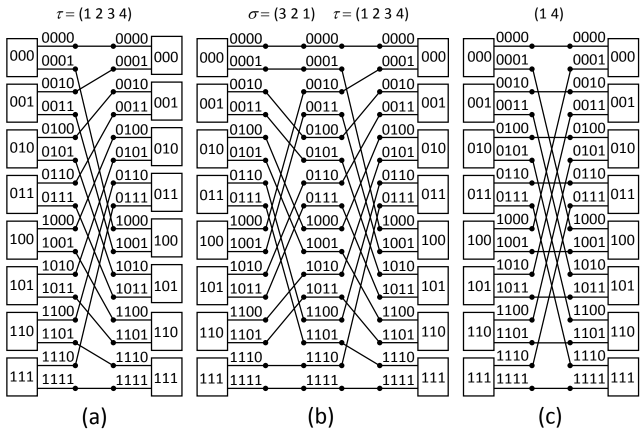

Fig. 6: Banyan permutation converted by the cascade of σ and a baseline permutation: (a) a baseline permutation τ = (1 2 3 4), (b) the cascade of σ = (3 2 1) and τ, and (c) obtained permutation (1 4).

Lemma 4: Let τ denote a banyan permutation (m n), where $1 \le m < n$, and σ denote permutation ($n - 1$ $n - 2$ … r), where $1 \le r < n-1$. Then, the cascade of σ, τ, and σ^{-1} is a banyan permutation, which is determined based on the relation between m and r as follows:

if $m < r$, (m n),
if $r \le m < n-1$, ($m + 1$ n), and
if $m = n - 1$, (r n).

Proof: Define the binary number $i = b_1 b_2 \cdots b_n$. Then,

$$\sigma(i) = b_1 b_2 \cdots b_{r-1} b_{r+1} b_{r+2} \cdots b_{n-1} b_r b_n .$$

If $m < r$,

$$\tau(\sigma(i)) = b_1 b_2 \cdots b_n \cdots b_{r-1} b_{r+1} b_{r+2} \cdots b_{n-1} b_r b_m .$$

By applying σ^{-1} to this,

$$\sigma^{-1}(\tau(\sigma(i))) = b_1 b_2 \cdots b_n \cdots b_{r-1} b_r b_{r+1} b_{r+2} \cdots b_{n-2} b_{n-1} b_m .$$

Thus, the cascaded permutation is (m n) for $m < r$.

If $r \le m < n-1$,

$$\tau(\sigma(i)) = b_1 b_2 \cdots b_{r-1} b_{r+1} b_{r+2} \cdots b_n \cdots b_{n-1} b_r b_{m+1} ,$$

where b_n is m-th bit. Notably, the last bit is b_{m+1} because the m-th bit of $\sigma(i)$ is b_{m+1}. By applying σ^{-1} to this,

$$\sigma^{-1}(\tau(\sigma(i))) = b_1 b_2 \cdots b_{r-1} b_r b_{r+1} b_{r+2} \cdots b_n \cdots b_{n-2} b_{n-1} b_{m+1} .$$

Because b_n is moved to the ($m + 1$)-th bit by σ^{-1}, the cascaded permutation is ($m + 1$ n) for $r \le m < n-1$.

If $m = n - 1$,

$$\tau(\sigma(i)) = b_1 b_2 \cdots b_{r-1} b_{r+1} b_{r+2} \cdots b_{n-1} b_n b_r .$$

By applying σ^{-1} to this,

$$\sigma^{-1}(\tau(\sigma(i))) = b_1 b_2 \cdots b_{r-1} b_n b_{r+1} b_{r+2} \cdots b_n \cdots b_{n-1} b_r .$$

This permutation is clearly (r n). □

Lemma 5: Let τ denote a banyan permutation (m n), where $1 \le m < n$, and σ denote permutation (r $r + 1$ … $n - 1$), where $1 \le r < n-1$. Then, the cascade of σ, τ, and σ^{-1} is a banyan permutation, which is determined depending on the relation between m and r as follows:

if $m < r$, (m n),
if $m = r$, ($n - 1$ n), and
if $r < m \le n-1$, ($m - 1$ n)

Proof: Define the binary number $i = b_1 b_2 \cdots b_n$. Then,

$$\sigma(i) = b_1 b_2 \cdots b_{r-1} b_{n-1} b_r \cdots b_{n-2} b_n .$$

If $m < r$,

$$\tau(\sigma(i)) = b_1 b_2 \cdots b_n \cdots b_{r-1} b_{n-1} b_r \cdots b_{n-2} b_m .$$

By applying σ^{-1} to this,

$$\sigma^{-1}(\tau(\sigma(i))) = b_1 b_2 \cdots b_n \cdots b_{r-1} b_r \cdots b_{n-2} b_{n-1} b_m$$

Thus, the cascaded permutation is (m n) for $m < r$.

If $m = r$,

$$\tau(\sigma(i)) = b_1 b_2 \cdots b_{r-1} b_n b_r \cdots b_{n-2} b_{n-1}$$

because the m-th bit of $\sigma(i)$ is b_{n-1}. By applying σ^{-1} to this,

$$\sigma^{-1}(\tau(\sigma(i))) = b_1 b_2 \cdots b_{r-1} b_r \cdots b_{n-2} b_n b_{n-1}$$

Thus, the cascaded permutation is ($n - 1$ n) for $m = r$.

If $r < m$,

$$\tau(\sigma(i)) = b_1 b_2 \cdots b_{r-1} b_{n-1} b_r \cdots b_n \cdots b_{n-2} b_{m-1}$$

because the m-th bit of $\sigma(i)$ is b_{m-1}. By applying σ^{-1} to this,

$$\sigma^{-1}(\tau(\sigma(i))) = b_1 b_2 \cdots b_{r-1} b_r \cdots b_n \cdots b_{n-2} b_{n-1} b_{m-1}$$

Because b_n moves to the ($m - 1$)-th bit by σ^{-1}, the cascaded permutation is ($m - 1$ n) for $r < m$. □

Figure 7 shows an example where Lemma 4 holds. The figure shows the case where τ is (1 4) and σ is (3 2 1). Thus, $n = 4$ and $m = r = 1$. Figure 7 (a) shows the cascade of σ, τ, and σ^{-1}. By redrawing links, the configuration shown in Fig. 7 (b) is obtainable. This is clearly represented by permutation (2 4).

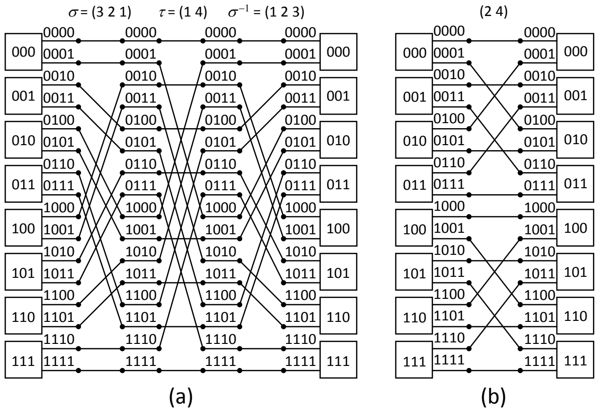

Fig. 7: Example showing where Lemma 4 holds: (a) a cascade of σ = (3 2 1), τ = (1 4), σ^{-1} = (1 2 3) and (b) banyan permutation (2 4) obtained by cascading σ, τ, and σ^{-1}.

Lemma 6: Suppose that we have a banyan-type network X constructed as follows:

· the permutations of exchanges 1, 2, $k-1$ ($k < s$) are banyan permutations

· the permutation of exchange k is a baseline permutation or reverse baseline permutation.

Then, there exists a network X' isomorphic to X and can be constructed as follows:

· the permutations of exchanges 1, 2, …, k are banyan permutations.

Proof: Assume that the permutation of exchange k is a baseline permutation (m $m+1$ … n) in X. For this case, let us define σ as (m $m+1$ … $n-1$). Thus, σ^{-1} is ($n-1$ $n-2$ … m). Then, assume that X is converted to a network by inserting σ before each of stages 1, 2, …, k and σ^{-1} after each of stages 1, 2, …, k. By Lemma 1, the obtained network is isomorphic to X. In the converted network, the permutation of exchange k is a cascade of ($n-1$ $n-2$ … m) and (m $m+1$ … n). Lemma 2 asserts that this cascade is (m n). In addition, the permutation for each of exchanges 1, 2, …, $k-1$ is a cascade of ($n-1$ $n-2$ … m), (r n), and (m $m+1$ … $n-1$), where $1 \le r < n$. By Lemma 4, this cascade is a banyan permutation. Thus, the permutations of exchanges 1, 2, …, k are banyan permutations.

If exchange k of X is a reverse baseline permutation (n $n-1$ … m), define σ as ($n-1$ $n-2$ … m) and create a network by inserting σ and σ^{-1} before and after each of stages 1, 2, …, $k-1$. By Lemma 1, the obtained network is isomorphic to X. Moreover, by Lemmas 3 and 5, the permutations of exchanges 1, 2, …, k are banyan permutations. □

Definition: A network constructed using only banyan permutations in exchanges 1, 2, …, $s-1$ is defined as a "pure banyan network."

The next theorem shows that most banyan-type networks are isomorphic to pure banyan networks.

Theorem 1: Consider a network X with permutations selected from the following:

· baseline permutations,

· reverse baseline permutations, and

· banyan permutations.

Then, there exists a pure banyan network isomorphic to X.

Proof: Assume that the baseline and reverse baseline permutations appear at exchanges $k_1, k_2, \ldots, k_t$ ($0 < k_1 < k_2 < \ldots < k_t < s$). Then, by Lemma 6, there is an isomorphic network X' such that the permutations of exchanges 1, 2, …, k_1 are banyan permutations. Moreover, the network X' is isomorphic to a network X'' such that exchanges 1, 2, …, k_2 are banyan permutations. By repeating this, we can find a network isomorphic to X and constructed by banyan permutations. □

According to Theorem 1, it is unnecessary to investigate the rearrangeability of a network that has baseline or reverse baseline permutations in their exchanges. The rearrangeability of such a network can be assessed by examining its isomorphic pure banyan network. Because the number of pure banyan networks is much smaller than that of networks including baseline or reverse baseline permutations, by Theorem 1, the rearrangeability assessment can be efficiently performed.

5 Necessary Conditions

5.1 Pure Banyan Networks

This section explores the necessary conditions for a $2n-1$ stage pure banyan network to be rearrangeable. Without loss of generality, we assume that no bits are permuted in exchanges 0 and $2n-1$ in the following.

For the derivation of the conditions, it is necessary to understand how a banyan permutation (m n) affects a connection path. Let us focus on an outlet of a switch in a certain stage. Let the index of the outlet be $b_1b_2\ldots b_m\ldots b_n$ ($b_j \in \{0,1\}$, $1 \le j \le n$). If this outlet is connected to the inlet of the next stage through (m n), its index will be $b_1b_2\ldots b_n\ldots b_m$. Because b_n can be set to 0 or 1 by a switch, (m n) determines the m-th bit of the index assigned to the next stage outlet, at which the path arrives. Meanwhile, (m n) does not affect any bits other than b_m and b_n. This leads to the following lemmas.

Lemma 7: If there is no permutation $(m\ n)$ for a certain m $(1 \le m \le n-1)$ in exchanges $1, 2, \ldots, 2n-2$ of a $2n-1$ stage pure banyan network, it is impossible to connect input 00…0 to output 11…1.

Proof: We focus on a path originating from input 00…0. A unit switch can invert the n-th bit of the outlet index the path goes through. In addition, if there is a permutation $(m\ n)$, it is possible to invert the m-th bit by interchanging the m-th and n-th bits. However, if $(m\ n)$ does not exist in exchanges, there is no means to alter the m-th bit to 1 in any switch inlet/outlet indices that the path goes through. Thus, the path cannot reach the output with the index whose m-th bit is 1. Therefore, input 00…0 cannot be connected to output 11…1. □

Figure 8 illustrates Lemma 7's proof. The figure shows the case for $n = 4$ and $(2\ 4)$ is missing at any exchanges. In the figure, the red thick lines indicate possible routes of a path originating from 0000. The path routes cannot reach any switch inlet, switch outlet, or output with a second bit of 1. The number of possible routes from 0000 is 128 because one of two routes can be selected at each of 7 stages. However, the second bit of the destinations is always 0 for these routes. Thus, the number of reachable outputs is limited to $2^{n-1} = 8$, a half of N.

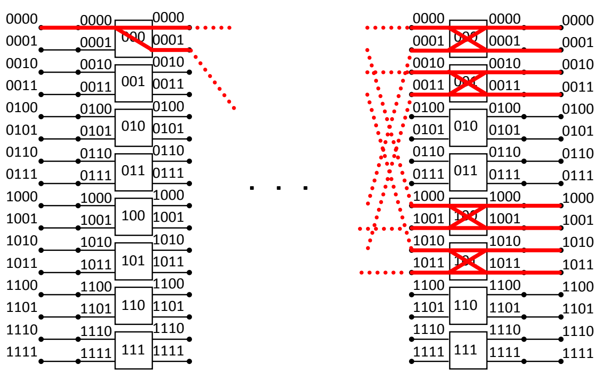

Fig. 8: Pure banyan network without permutation (2 4) in exchanges and possible route of a path that originated from 0000.

Lemma 8: If permutation $(m\ n)$ appears only at a single exchange among exchanges $1, 2, \ldots, 2n-2$ of a $2n-1$ stage pure banyan network, it is impossible to connect the following inputs and outputs:

- 2^{n-1} inputs such that the m-th bit of the indices is 0 and
- 2^{n-1} outputs such that the m-th bit of the indices is 0.

Proof: Consider that $(m\ n)$ appears at exchange k $(1 \le k < 2n-1)$ and does not appear at any other exchange. Then, we focus on a path originating from an input such that the m-th bit of the index is 0. At stage k, because $(m\ n)$ does not appear in exchanges $1, 2, .., k-1$, the path must go through the unit switch inlet such that the m-th bit of the index is 0. Because the index of a unit switch is the first $n-1$ bits of its inlet index, the m-th bit of the switch index is also 0. The number of such unit switches is 2^{n-2}. Meanwhile, there are 2^{n-1} paths originating from inputs such that the m-th bit of their indices is 0. Therefore, two of these paths must go through a stage k switch such that the m-th bit of the index is 0. For this switch, the n-th bit of the index is 0 for one outlet and 1 for the other. Thus, half of the 2^{n-1} paths pass the outlets with the n-th bit of 1. Then, these 2^{n-2} paths go through $(m\ n)$ of exchange k and reach the inlets of stage $k+1$ switches, where the m-th bit of the indices is 1. Exchanges $k+1, \ldots, 2n-2$ do not have $(m\ n)$ and thus do not invert the m-th bit. Therefore, the 2^{n-2} paths cannot reach the outputs such that the m-th bit of the indices is 0. □

Figure 9 illustrates how the connection requests shown in Lemma 8 are not satisfied. The figure shows the case where permutation (2 4) appears once at exchange k. The red thick lines indicate possible routes from inputs such that the second bit of the indices is 0. Each of these paths goes through one of the four switches 000, 001, 100, and 101 at stage k. Therefore, four of the eight paths must pass outlets 0001, 0011, 1001, and 1011, with the fourth bit of 1. The blue thick lines indicate possible routes stemming from outlets 0001, 0011, 1001, and 1011. Exchange k interchanges the second and fourth bits; hence, the paths from 0001, 0011, 1000, and 1011 reach stage $k + 1$ inlets 0100, 0110, 1100, and 1110. Afterward, there is no means to invert the second bit. Thus, the blue lines do not reach outputs with the second bit of 0, e.g., 0000 or 1011.

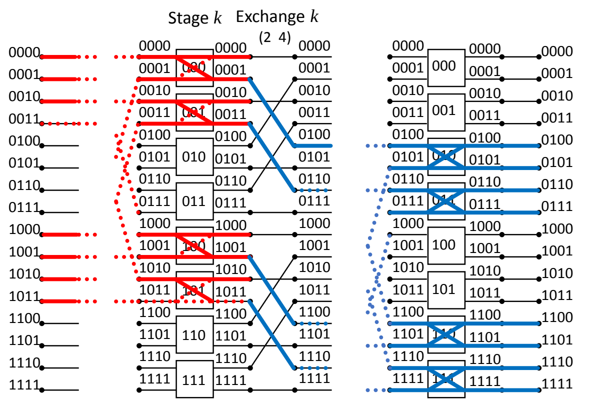

Fig. 9: Pure banyan network where permutation (2 4) appears only once in exchanges. The blue line paths cannot reach the outputs such that the second bit of their indices is 0.

From Lemmas 7 and 8, it becomes possible to derive the condition that must be satisfied for rearrangeability.

Theorem 2: If a $2n-1$ stage pure banyan network is rearrangeable, each of (1 n), (2 n), …, ($n-1$ n) appears twice among exchanges 1, 2, …, $2n-2$.

Proof: Let P_m denote how many times (m n) appears among exchanges 1, 2, …, $2n-2$, where $1 \le m < n$. Because the number of exchanges is $2n-2$,

$$\sum_{m=1}^{n-1} P_m = 2n-2\,. \qquad (1)$$

If the network is rearrangeable, Lemma 7 asserts $P_m \neq 0$ for all m. Moreover, Lemma 8 states that $P_m \neq 1$ for all m. Thus,

$$P_m \ge 2,\ \text{for } 1 \le m \le n-1 \qquad (2)$$

Both of (1) and (2) are satisfied only if $P_m = 2$. This proves the theorem. □

Theorem 2 provides a necessary condition. Thus, even if a network satisfies this condition, it may be a blocking network. A stronger necessary condition is derived as follows.

Lemma 9: Assume that Theorem 2 holds. Assume further that (m n) $(1 \le m < n)$ appears at exchanges k_1 and k_2 $(1 \le k_1 < k_2 < 2n-2)$ in a $2n-1$ stage pure banyan network. If a permutation (r n) $(1 \le r < n,\ r \neq m)$ does not appear at any of exchanges 1, 2, …, k_2-1, it is impossible to simultaneously connect the following inputs and outputs:

- 2^{n-1} inputs such that the r-th bit of their indices is 0 and
- 2^{n-1} outputs such that the m-th bit of their indices is 0.

Proof: We focus on a path originating from an input such that the r-th bit of its index is 0. At stage k_2, this path reaches a switch inlet such that the r-th bit of its index is 0 because it does not go through (r n) from the input to stage k_2 and the r-th bit is not inverted. The r-th bit of the switch index is also 0 because the switch index is the first $n-1$ bits of the inlet index. There are 2^{n-1} paths originating from the 2^{n-1} inputs such that the r-th bit of their indices is 0. At stage k_2, these paths reach 2^{n-2} switches such that the r-th bit of the indices is 0. Half of these 2^{n-2} switches have the indices whose m-th bit is 1. Thus, the 2^{-2} paths must go through stage k_2 switches such that the m-th bit of the index is 1. For these switches, the m-th bit of the outlet indices is also 1. Because Theorem 2 holds, the permutations of exchanges k_2+1, …, $2n-2$ are not (m n) and thus do not invert the m-th bit. Therefore, the 2^{n-2} paths cannot reach outputs where the m-th bit of their index is 0. □

Theorem 3: If a $2n-1$ stage pure banyan network is rearrangeable, each of (1 n), (2 n), …, ($n-1$ n) appears once at its exchanges 1, 2, …, $n-1$ and appears once at its exchanges n, $n+1$, …, $2n-2$.

Proof: Assume that (r n) appears twice at exchanges 1, 2, …, $n-1$ and are assigned to exchanges k_1 and k_2 $(1 \le r < n,\ 1 \le k_1 < k_2 \le n-1)$. Then, for some m $(1 \le m < n, m \neq r)$, (m n) does not appear at any of exchanges 1, 2, ..., $n-1$. By Lemma 9, this configuration is not rearrangeable. Therefore, if the network is rearrangeable, each of (1 n), (2 n), …, ($n-1$ n) appears once at exchanges 1, 2, …, $n-1$. Theorem 2 asserts that each of (1 n), (2 n), …, ($n-1$ n) must appear one more time at exchanges n, $n+1$, …, $2n-2$ to be rearrangeable. □

There are $(n-1)!$ ways of assigning $n-1$ permutations to $n-1$ exchanges. Therefore, $((n-1)!)^2$ pure banyan networks satisfy the necessary condition of Theorem 3. When $n = 4$, $((4-1)!)^2 = 36$. Thus, for the discovery of rearrangeable networks, the number of candidate networks is as small as 36. Meanwhile, the total number of possible pure banyan networks is $(n-1)^{2n-2}$, which is 729 for $n = 4$. As this numerical example shows, Theorem 3 suggests that it is necessary to test a very small number of possible networks as candidates.

5.2 Beneš-Type Structure

Let B_n denote a Beneš network that has 2^n inputs and outputs. Then, the Beneš network definition is shown in Figure 10. As shown in Figure 10 (a), B_1 is a unit switch. For $n > 1$, B_n is recursively constructed using two B_{n-1}'s (Figure 10 (b)).

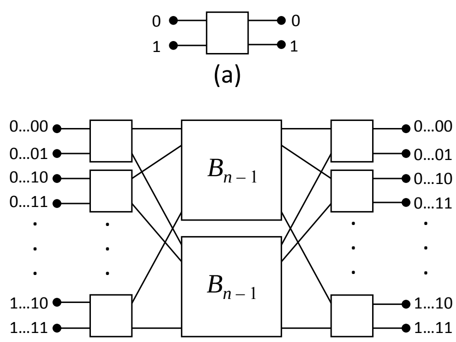

Fig. 10: Definition of the Beneš network: (a) configuration for n = 1 and (b) configuration for n > 1.

We consider a network configured by replacing B_{n-1} with a different network C_{n-1} ($C_{n-1} \neq B_{n-1}$) in Figure 10 (b). The network is shown in Figure 11.

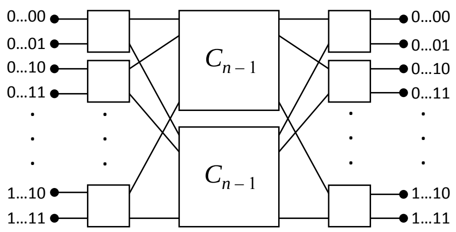

Fig. 11: Beneš-type configuration. This network is blocking if subnetwork Cn − 1 is blocking.

The network in Figure 11 is rearrangeable if C_{n-1} is rearrangeable. This is obvious because of the sufficient condition for the rearrangeability of three-stage switching networks. Therefore, for $n = 4$, the network is rearrangeable when C_3 is the 5-stage omega (or shuffle-exchange) network with eight inputs/outputs because the 5-stage omega network is rearrangeable. We call this configuration a 7-stage omega–Beneš network because it is a hybrid of Beneš and omega networks.

For $n = 5$, rearrangeable networks are obtainable by setting C_4 to the 7-stage omega network and the 7-stage omega-Beneš network. Hereinafter, the former is called a 9-stage omega–Beneš network, while the latter is called a 9-stage omega–Beneš–Beneš network.

The next theorem provides another characteristic of the network in Figure 11.

Theorem 4: If C_{n-1} is a blocking network, the network in Figure 11 is also a blocking network.

Proof: Suppose that a connection request set π cannot be satisfied by C_{n-1}. For π, assume that input x must be connected to output $p(x)$. Let x and $p(x)$ be, respectively, represented in binary form as follows:

$$x = b_1 b_2 ... b_{n-1}$$

$$p(x) = c_1 c_2 \ldots c_{n-1}.$$

Then, suppose that the following connection request is given to the network in Figure 11:

· connect input $b_1 b_2 \ldots b_{n-1}0$ to output $c_1 c_2 \ldots c_{n-1}0$

· connect input $b_1 b_2 \ldots b_{n-1}1$ to output $c_1 c_2 \ldots c_{n-1}1$.

To satisfy this request, C_{n-1} must connect π independently of how the unit switches of the first and final stages are set up. This is impossible. Thus, the connection request set is blocked. □

6 Empirical Observations

To identify the class of rearrangeable banyan-type networks, two computer experiments were performed for $n = 4$ and 5. As in the base method, candidate networks were first generated by assigning bit permutations to exchanges. Then, the rearrangeability of each network was assessed by solving CNF-SAT problems for many connection request sets. In this study, based on Theorems 1–4, the number of examined banyan-type networks was significantly reduced compared with that in [30]. First, the study concentrates on the assessment of pure banyan networks based on Theorem 1. Moreover, by Theorems 2 and 3, only $((n-1)!)^2$ of pure banyan networks are sufficient for assessment. For example, if $n = 4$, only 36 networks should be examined, while the study of [30] tested as many as 117,649 networks. In addition, Theorem 4 asserts that some of $((n-1)!)^2$ pure banyan networks are blocking networks because of the isomorphism to the network in Figure 11 with blocking subnetwork C_{n-1}.

In the experiments, graph isomorphism was checked by Nauty [34]. CaDiCal was employed as the SAT solver, [35]. Some custom programs were developed for banyan network generation, graph generation for isomorphism, random connection request generation, and SAT problem formulation from a network and a connection request set.

6.1 Case of $n = 4$

Rearrangeability was assessed for $n = 4$. First, $((4-1)!)^2 = 36$ pure banyan networks that satisfy the necessary conditions of Theorems 2 and 3 were generated. The networks are divided into five subsets, each comprising isomorphic networks. Among these, the networks of three subsets are isomorphic to known rearrangeable networks. One of the subsets comprises six networks, each of which is isomorphic to the Beneš network. The other two subsets also comprise six networks, respectively. For one of these subsets, each network is isomorphic to the 7-stage omega network. For the other subset, each network is isomorphic to the 7-stage omega–Beneš network. For the remaining two subsets, no network is isomorphic to a known rearrangeable network. We call these subsets groups 1 and 2. Group 1 comprises 12 isomorphic networks, one of which is

$$[id : (1\ 4) : (2\ 4) : (3\ 4) : (1\ 4) : (3\ 4) : (2\ 4) : id]_4.$$

Meanwhile, group 2 comprises six isomorphic networks, one of which is

$$[id : (1\ 4) : (2\ 4) : (3\ 4) : (3\ 4) : (1\ 4) : (2\ 4) : id]_4.$$

For each network of groups 1 and 2, SAT problems were generated and solved by randomly generating 10^5 connection request sets. As a result, it was found that the networks of groups 1 and 2 are blocking networks. From this result, it is concluded that a rearrangeable banyan network is inevitably isomorphic to one of the Beneš network, 7-stage omega network, and 7-stage omega–Beneš network for $n = 4$.

6.2 Case of *n* = 5

For $n = 5$, it is sufficient to consider $((5 - 1)!)^2 = 576$ pure banyan networks for assessing rearrangeability. Among these networks, the following are isomorphic to known rearrangeable networks and thus rearrangeable:

- 24 networks isomorphic to the 9-stage omega network
- 24 networks isomorphic to the Beneš network
- 24 networks isomorphic to the 9-stage omega–Beneš network, and
- 24 networks isomorphic to the 9-stage omega–Beneš–Beneš network.

According to Theorem 4 and the result for $n = 4$, the configuration in Figure 11 is blocking if C_{n-1} is $[id : (1\ 4) : (2\ 4) : (3\ 4) : (1\ 4) : (3\ 4) : (2\ 4) : id]_4$ or $[id : (1\ 4) : (2\ 4) : (3\ 4) : (3\ 4) : (1\ 4) : (2\ 4) : id]_4$. It was found that 48 networks are isomorphic to the former configuration, while 24 networks are isomorphic to the latter configuration. Thus, these 72 networks are blocking networks.

For the remaining 408 networks, the rearrangeability was assessed. These networks are divided into 11 subsets, each comprising isomorphic networks. Hereinafter, these subsets are called groups 1, 2, …, 11, comprising 48, 24, 48, 48, 48, 24, 48, 24, 24, 48, and 24 networks, respectively. For each group, a network was selected as a representative as follows:

- Group 1: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (1\ 5) : (2\ 5) : (4\ 5) : (3\ 5) : id]_5$
- Group 2: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (1\ 5) : (3\ 5) : (2\ 5) : (4\ 5) : id]_5$
- Group 3: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (1\ 5) : (3\ 5) : (4\ 5) : (2\ 5) : id]_5$
- Group 4: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (1\ 5) : (4\ 5) : (2\ 5) : (3\ 5) : id]_5$
- Group 5: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (1\ 5) : (4\ 5) : (3\ 5) : (2\ 5) : id]_5$
- Group 6: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (2\ 5) : (1\ 5) : (4\ 5) : (3\ 5) : id]_5$
- Group 7: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (2\ 5) : (4\ 5) : (1\ 5) : (3\ 5) : id]_5$
- Group 8: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (3\ 5) : (4\ 5) : (1\ 5) : (2\ 5) : id]_5$
- Group 9: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (4\ 5) : (1\ 5) : (2\ 5) : (3\ 5) : id]_5$
- Group 10: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (4\ 5) : (1\ 5) : (3\ 5) : (2\ 5) : id]_5$
- Group 11: $[id : (1\ 5) : (2\ 5) : (3\ 5) : (4\ 5) : (4\ 5) : (3\ 5) : (1\ 5) : (2\ 5) : id]_5$.

For the representative network of each group, rearrangeability was assessed using the CNF-SAT approach. For each network, the SAT problems were solved for more than 10^6 randomly generated connection request sets. After computations that lasted several weeks, unsatisfiable, i.e., blocking, cases were found for groups 3, 4, ..., 11. Thus, 336 networks included in these nine groups are blocking networks. For some groups, blocking rarely occurred. Therefore, many connection request sets had to be examined to find an unsatisfiable case. For example, the representative network of group 3 caused blocking for only a single connection request set among 2.6×10^8 connection request sets.

For the networks of groups 1 and 2, no unsatisfiable cases were found after 7×10^8 connection request sets were examined. Unsatisfiable cases may also appear for these networks if a larger number of connection request sets are examined. Unfortunately, it is not easy to increase the number of connection request sets because of the enormous computational time. Meanwhile, the networks of groups 1 and 2 may be nonblocking. Therefore, the rearrangeability is inconclusive for 72 networks belonging to groups 1 and 2.

7 Conclusion

This study classified rearrangeable $2n - 1$ stage banyan-type networks. This is done by first generating candidate networks and then checking their rearrangeability through the CNF-SAT modeling method. The problem with this approach is poor scalability due to the enormous number of candidate networks. This study effectively reduced the number of candidate networks by revealing the theoretical characteristics of banyan-type networks. First, it was shown that any banyan-type network constructed with a baseline, reverse baseline, or banyan permutation is isomorphic to a pure banyan network. A pure banyan network is constructed exclusively using banyan permutation (m n). Next, necessary conditions for the rearrangeability of a pure banyan network were presented. It is unnecessary to assess the rearrangeability of networks that do not satisfy the conditions. As a result, the number of candidate networks is reduced to $((n-1)!)^2$. A condition for a blocking network was also revealed by observing the Beneš network structure. The number of candidate networks further decreases by removing the networks that are isomorphic to a blocking network.

Based on the theoretical results, computer experiments were performed to assess the rearrangeability of pure banyan networks for $n = 4$ and 5. For $n = 4$, 36 candidate networks are categorized into five groups, each comprising isomorphic networks. Among the groups, the networks of three groups are isomorphic to known rearrangeable networks. It was found through the CNF-SAT method that 18 networks of the remaining two groups are blocking networks.

For $n = 5$, 96 of 576 candidate networks are isomorphic to known rearrangeable networks. Moreover, 72 networks are isomorphic to blocking networks. The remaining 408 networks are categorized into 11 groups, each comprising isomorphic networks. The CNF-SAT method was applied to these groups. The result shows that 336 networks of nine groups are blocking networks. For 72 networks of the other two groups, the blocking cases were not found. The rearrangeability of these networks is inconclusive.

In this study, the rearrangeability of banyan-type networks was assessed for $n = 4$ and 5 to some extent. However, the approach has a scalability problem, although the number of candidate networks significantly decreased compared with that reported in [30]. This problem is because blocking rarely occurs in some networks for a larger n value, and the occurrence of blocking must be checked for an extremely large number of connection request sets. Because of this problem, rearrangeability could not be determined for some networks when $n = 5$. To completely clarify the rearrangeability for larger n values, a different theoretical approach is required. The development of such a theory is an open problem.

Among the theorems derived in this paper, Theorems 2, 3, and 4 provide the necessary conditions for rearrangeability. A future work is to show the necessary and sufficient condition for the rearrangeability. Another future work is to focus on the reliability of the networks, as found in, [36], [37].

References:

[1] F.K. Hwang, *The Mathematical Theory of Nonblocking Switching Networks 2nd Edition*, World Scientific, Singapore, 2004.

[2] W. Kabaciński, *Nonblocking Electronic and Photonic Switching Fabrics*, Springer, New York, 2005.

[3] V.E. Benes *Mathematical Theory of Connecting Networks and Telephone Traffic*, Academic Press, New York, 1965.

[4] A. Das, G. Zhang, H.R. Rahbardar Mojaver, and O. Liboiron-Ladouceur, Low loss 8 × 8 silicon photonic banyan switch, *In Proc. 2020 IEEE Photonics Conference (IPC 2020)*, Vancouver, BC, Canada, pp. 1-2, Sept. 2020.

[5] Q. Cheng, Y. Huang, H. Yang, M. Bahadori, N. Abrams, X. Meng, M. Glick, Y. Liu, M. Hochberg, and K. Bergman, Silicon photonic switch topologies and routing strategies for disaggregated data centers, *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 26, No. 2, pp. 1-10, Mar.-Apr. 2020.

[6] A. Das, H. R. Mojavar, G. Zhang, O. Liboiron-Ladouceur, Acalable SiPh-InP hybrid switch based on low-loss building blocks for lossless operation, *IEEE Photonics Technology Letters*, Vol. 32, pp. 1401-1404, Nov. 2020.

[7] B.G. Lee and N. Dupuis, Silicon photonics switch fabrics: technology and architecture, *Journal of Lightwave Technology*, Vol. 37, No. 1, pp. 6-20, Jan. 2019.

[8] S. Song, H. Cui, and Z. Wang, A universal efficient circular-shift network for reconfigurable quasi-cyclic LDPC decoders, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, Vol. 30, No. 10, pp. 1553-1557, Oct. 2022.

[9] Y. Chen, J. Wang, S. Li, J. Xie, Q. Zhang, K.K. Parhi, and X. Zeng, A reconfigurable 74-140Mbps LDPC decoding system for CCSDS standard, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E104-a, No. 11, pp. 1509-1515, Nov. 2021.

[10] T.T. Nguyen, T.T.B. Nguyen, and H. Lee, Low complexity multi-size circular-shift network for 5G new radio LDPC decoders, *Sensors*, Vol. 22, No. 5, p. 1792, Feb. 2022.

[11] Q. Huang, L. Huang, S. Cai, X. Xiong, and H. Zhang, On a symmetric image cryptosystem based on a novel one-dimensional chaotic system and banyan network, *Mathematics*, Vol. 11, No. 21, p. 4411, Nov. 2023.

[12] K. Freivalds, E. Ozoliņš, and A. Šostaks, Neural shuffle-exchange networks – sequence

processing in $O(n \log n)$ time, *In Proc. 33rd Conference on Neural Information Processing System (NeurIPS 2019)*, poster 131, Vancouver, Canada, Dec. 2019.

[13] J.Y. Hui and E. Arthurs, A broadband packet switch for integrated transport, *IEEE Journal on Selected Areas in Communications*, Vol. SAC-5, No. 8, pp. 1264-1273, Oct. 1987.

[14] J.S. Turner, Design of a broadcast packet switching network, *IEEE Transactions on Communications*, Vol. 36, No. 6, pp. 734-743, June 1988.

[15] S.F. Oktuğ and M.U. Çağlayan, Design and performance evaluation of a banyan network based interconnection structure for ATM switches, *IEEE Journal on Selected Areas in Communications*, Vol. 15, No. 5, pp. 807-816, June 1997.

[16] H.S. Stone, Parallel processing with the perfect shuffle, *IEEE Transactions on Computers*, Vol. C-20, No. 2, pp. 153-161, Feb. 1971.

[17] D.H. Lawrie, Access and alignment of data in an array processor, *IEEE Transactions on Computers*, Vol. C-24, No. 12, pp. 1145-1155, Dec. 1975.

[18] F.P. Preparata and J. Vuillemin, The cube-connected cycles: a versatile network for parallel computation, *Communications of the ACM*, Vol. 24, No. 5, pp. 300-309, May 1981.

[19] L.R. Goke and G.J. Lipovsky, Banyan networks for partitioning multiprocessor systems, *In Proc. the 1st Annual Symposium on Computer Architecture (ISCA '73)*, Gainesville, FL, USA, pp. 21-28, Dec. 1973.

[20] C.-L. Wu and T.-Y. Feng, On a class of multistage interconnection networks, *IEEE Transactions on Computers*, Vol. C-29, Vol. 8, pp. 694-702, Aug. 1980.

[21] A. Waksman, A permutation network, *Journal of the ACM*, Vol. 15, No. 1, pp. 159-163, Jan. 1968.

[22] V.E. Beneš, Proving the rearrangeability of connecting networks by group calculations, *Bell System Technical Journal*, Vol. 54, No. 2, pp. 421-434, Feb. 1975.

[23] H.Q. Ngo and D.-Z. Du, Remarks on Beneš conjecture, *In Switching Networks: Recent Advances*, D.-Z. Du and H.Q. Ngo Eds, Kluwar Academic Publishers, Dordrecht, pp. 257-258, 2001.

[24] H. Çam, Rearrangeability of $(2n - 1)$-stage shuffle-exchange networks, *SIAM Journal on Computing*, Vol. 32, No. 3, pp. 557-585, Mar. 2003.

[25] S.-Y. Li and X.J. Tan, On rearrangeability of tandem connection of banyan-type networks, *IEEE Transactions on Communications*, Vol. 57, No. 1, pp. 164-170, Jan. 2009.

[26] H. Dai and X. Shen, Rearrangeability of 7-stage 16×16 shuffle exchange networks, *Frontiers of Electrical and Electronic Engineering in China*, Vol. 3, No. 4, pp. 440-458, Sept. 2008.

[27] H.Q. Ngo and D.-Z. Du, On the rearrangeability of shuffle-exchange networks, *Department of Computer Science and Engineering, University of Minnesota, Technical Report*, p. TR00-045, Sept. 2000.

[28] V.E. Beneš, Optimal rearrangeable multistage connecting networks, *Bell System Technical Journal*, Vol. 43, No. 4, pp. 1641-1656, July 1964.

[29] K.Y. Lee, On the rearrangeability of $2(\log_2 N) - 1$ stage permutation networks, *IEEE Transactions on Computers*, Vol. C-34, Vol. 5, pp. 412-425, May 1985.

[30] S. Ohta and N. Tsuji, On the existence of unknown rearrangeable banyan-type networks, *In Proc. the 6th International Conference on Electronics, Communications, and Control. Engineering (ICECC 2023)*, Fukuoka, Japan, pp. 153-158, Mar. 2023.

[31] S. Ohta, CNF-SAT modeling for banyan-type networks and its application for assessing the rearrangeability, *In Proc. 10th International Conference on Mathematical Modeling in Physical Sciences (IC-MSQUARE 2021), Journal of Physics: Conference Series*, Vol. 2090, p. 012133, Sept. 2021.

[32] K. Claessen, N. Een, M. Sheeran, N. Sörensson, A. Voronov, and K. Åkesson, SAT-solving in practice, with a tutorial example from supervisory control, *Discrete Event Dynamic Systems*, Vol. 19, No. 4, pp. 495-524, Sept. 2009.

[33] B.D. McKay and A. Piperno, Practical graph isomorphism, II, *Journal of Symbolic Computation*, Vol. 60, pp. 94-112, 2014.

[34] B.D. McKay and A. Piperno, Nauty and Traces User's Guide Version 2.7. https://pallini.di.uniroma1.it/Guide.html, Oct. 16, 2022.

[35] A. Biere, CaDiCaL at the SAT race 2019, *In Proc. SAT race, 2019 Solver and Benchmark Descriptions*, Lisbon, Portugal, pp. 8-9, July 2019.

[36] R. Abedini and R. Ravanmehr, Parallel SEN: a new approach to improve the reliability of shuffle-exchange network, *The Journal of*

Supercomputing, Vol. 76, pp. 10319-10353, Mar. 2020.

[37] F. Bistouni and M. Jahanshahi, Determining the reliability importance of switching elements in the shuffle-exchange networks, *International Journal of Parallel, Emergent and Distributed Systems*, Vol. 34, No. 4, pp. 448-476, July 2018.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

Satoru Ohta conducted the whole study.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The authors have no conflicts of interest to declare that are relevant to the content of this article.