CNF-SAT modelling for banyan-type networks and its application for assessing the rearrangeability

S Ohta
Toyama Prefectural University
5180 Kurokawa, Imizu-shi, Toyama, 939-0398, JAPAN
E-mail: ohta@pu-toyama.ac.jp

Abstract. A banyan-type network is a switching network, which is constructed by placing unit switches with two inputs and two outputs in \( s \) \((s > 1)\) stages. In each stage, \( 2^{n-1} \) \((n > 1)\) unit switches are aligned. Past studies conjecture that this network becomes rearrangeable when \( s \geq 2n - 1 \). Although a considerable number of theoretical analyses have been done, the rearrangeability of the banyan-type network with \( 2n - 1 \) or more stages is not completely proved. As a tool to assess the rearrangeability, this study presents a CNF-SAT (conjunctive normal form - satisfiability) modelling scheme for banyan-type networks. In the proposed scheme, the routing is formulated to a SAT problem represented in CNF. By feeding the problem to a SAT solver, it is found whether the problem is satisfiable or unsatisfiable. If the problem is unsatisfiable for a certain request, the network is not rearrangeable. By contrast, if the problem is satisfiable for any requests, the network is rearrangeable. This study applies the CNF-SAT modelling scheme to various configurations of \( 2n - 1 \) stage banyan-type networks. These networks are assessed for rearrangeability by solving the SAT problems. The proposed method will be helpful to conduct future theoretical studies on banyan-type networks.

1. Introduction
Communication and computer systems have utilized switching networks [1, 2], exchanging data between multiple inputs and outputs. A well-studied class of switching networks is built by aligning unit switches with \( d \) inputs and \( d \) outputs \((d \times d \) switches\) in multiple stages. Particularly for \( d = 2 \), past studies have presented various networks, which differ in their link configurations between stages. These include shuffle-exchange network [3], omega network [4], baseline network [5], banyan network [6], etc. This paper refers to this type of switching network as banyan-type network. A banyan-type network is configured by aligning \( 2^{n-1} \) \( 2 \times 2 \) switches in \( s \) stages. With this configuration, the network has \( 2^n \) inputs and \( 2^n \) outputs. Banyan-type networks have significant applications such as multiprocessor computer systems [3–6] and communication systems [7–10].

Let us assume that \( 2^n \) inputs and outputs of a banyan-type network are indexed \( 0, 1, \ldots, 2^n - 1 \). Then, the indices of outputs requested to connect inputs \( 0, 1, \ldots, 2^n - 1 \) are represented as indices’ permutations. For a given input/output permutation, the route must be appropriately determined from input to output by setting the state of unit switches in the network. A switching network is often required to establish connections for an arbitrary input/output permutation in practical systems. The switching network that has this capability is rearrangeable [1, 2, 11]. Therefore, it is significant to identify the rearrangeable
switching network with the smallest stages to obtain the least hardware cost and deal with arbitrary connection requests.

Beneč [12] presented a conjecture on the number of stages required for rearrangeable switching networks. According to the conjecture, a rearrangeable network constructed with \(2 \times 2\) switches has at least \(2n - 1\) stage. Various theoretical studies have been done on this conjecture and the number of required stages [13–19]. Among these, Li and Tan [19] revealed a considerably broad range of \(2n - 1\) stage rearrangeable networks. They investigated the rearrangeability of tandem connections of two \(n\)-stage banyan-type networks and showed a condition for rearrangeability. By removing a trivially unnecessary stage from a \(2n\) stage rearrangeable network, a \(2n - 1\) stage rearrangeable network is obtained. However, their result is a sufficient condition. Therefore, there may be other rearrangeable networks that do not satisfy Li and Tan’s condition.

Despite previous theoretical studies, the rearrangeability of \(2n - 1\) stage banyan-type networks remains a mystery. Therefore, it is beneficial to develop a tool to systematically judge the rearrangeability of a given network to comprehend rearrangeable banyan-type networks fully. It is also necessary to figure out how to route connections for a given permutation request if a network is found to be rearrangeable.

This research uses Boolean variables to model a connection route in a banyan-type network. The variables' restriction conditions then represent the feasibility of connection routes. The network will not establish connection routes for the given input/output permutation if the restriction conditions are not met. Determining whether the conditions are satisfiable is referred to as a SAT (satisfiability) problem [20]. The goal is to discover whether a variable value set exists that satisfies all of the constraints. The network does not achieve the requested permutation if there is no such value set (un satisfiable). Its unsatisfiability demonstrates the network's inability to establish connections for a specific permutation. As a result, this method can be used to evaluate the rearrangeability of banyan-type networks. Additionally, if the problem is satisfiable, the routing is also found by the value set that satisfies the restrictions. Thus, it is unnecessary to develop a routing algorithm for a given network.

This study formulates the SAT problem in CNF. A SAT problem represented in CNF is efficiently solved by a SAT solver [21]. Although SAT is NP complete, modern SAT solvers can handle considerably large problems. Thus, it is possible to test the rearrangeability of a network with a modest size.

The contribution of this paper is as follows.

- The paper shows how the routing in banyan-type networks is modeled into a CNF-SAT problem.
- Various banyan-type networks are tested for their rearrangeability by solving CNF-SAT problems constructed for all \((n = 3)\) or many \((n = 4)\) input/output permutations.

The paper is constructed as follows. Section 2 presents the basics of banyan-type networks and SAT and describes the investigated problem. Related past studies are reviewed in Section 3. Section 4 explores how the routing in a banyan-type network is formulated into a CNF-SAT problem. In Section 5, various \(2n - 1\) banyan-type networks are tested for rearrangeability by solving the CNF-SAT problem. Finally, Section 6 concludes the paper.

2. Preliminaries

2.1. Banyan-type networks and rearrangeability

Banyan-type networks are constructed by placing \(2^{n-1} \times 2 \times 2\) unit switches in multiple stages. This construction provides a switching network that has \(N = 2^n\) inputs and outputs. Figure 1 shows examples of such networks constructed by setting \(n\) to 3 and the number of stages to 3. In this figure, (a) is known as a banyan network, whereas (b) is known as a baseline network. These networks, somewhat alike, are classified by the difference in link configurations between stages. The link configuration between the two stages is referred to as “exchange” hereafter.
Let us give a binary index to every input/output of a unit switch and every input/output terminal, as shown in figure 1. Then, an exchange is defined by how the bit positions of the indices given to the output and input connected by a link are permuted. In literature [18, 19], the type of exchange is cyclically represented by the bit position permutation. For example, the rule between stages 1 and 2 is represented as (1 3) for the banyan network shown in figure 1 (a). The notation (1 3) means that the first bit is permuted to the third bit; meanwhile, the third bit is permuted to the first bit. Here, the position of the second bit is unchanged. Therefore, stage 1 output 011 is connected to stage 2 input 110 by swapping the first and third bits. Similarly, the rule between the second and the third stage is (2 3). Thus, the second and third bits are swapped while the first bit is unchanged. For example, stage 2 output 010 is connected to stage 3 input 001. The rule used in a banyan network is generally represented as (k n), where k < n. Exchange rule (k n) of a banyan network is also referred to as rank k.

In the baseline network shown in figure 1 (b), the exchange between stages 1 and 2 is characterized by bit permutation (1 2 3), which means bits are converted as the first bit to the second, the second bit to the third, and the third bit to the first. Thus, stage 1 output 110 is linked to stage 2 input 011. Similarly, the rule is (2 3) between stages 2 and 3. A baseline network features (k k + 1 … n) as an exchange.

In the networks shown in figure 1, an exchange between an input terminal and a first stage switch connects identical indices and does not permute any bits. The symbol id represents this type of exchange. With using this symbol, the banyan network of Fig. 1 (a) is characterized by exchanges id, (1 3), (2 3), and id. The sequence of exchanges and n uniquely determines the network configuration. Thus, the banyan network of Fig. 1 (a) is denoted as [id : (1 3) : (2 3) : id]. Similarly, the baseline network of figure 1 (b) is denoted as [id : (1 2 3) : (2 3) : id]. This notation, introduced in [18], is an accurate way of identifying the construction of banyan-type networks and thus employed hereafter.

This research assumes that a single connection uses a link exclusively. Furthermore, it is assumed that all connections are unicast. A connection from input to output is requested for data transmission in a switching network. The network is rearrangeable if it can establish connections for every possible input/output permutation [1, 11]. A requested connection between an input and an output may be
blocked in such a network, but it can be unblocked by rearranging existing connections. The networks in figure 1 are not rearrangeable. Rearrangeable networks are important because they can handle any connection requests.

For a rearrangeable switching network constructed with $2 \times 2$ switches, it is conjectured that the number of stages should be $2n - 1$ or larger for $N = 2^n$ inputs/outputs [13]. It is also known that rearrangeable networks exist with the minimum number ($= 2n - 1$) of stages. An example of such a network is the Beneš network shown in figure 2. With using the above-mentioned notation, this network is expressed as $[id : (1 2 3) : (2 3) : (3 2 1) : id]$. The rearrangeability of the Beneš network is easily confirmed because it is constructed by extending a rearrangeable three stage network.

![Figure 2. An example of Beneš network.](image)

Although the networks shown in figure 1 are not rearrangeable, they are routable. In a routable network, at least one route exists between each pair of an input and an output when no connections are set up.

2.2. SAT problems and solver

Boolean variables were used to model the routes of connections in banyan-type networks in this study. Then, based on the constraints on variables, a set of feasible routes for the connections is determined. However, the network cannot establish requested connections and thus is not rearrangeable if some restrictions are not met for a given permutation. Finding feasible restrictions values is thus a SAT (satisfiability) problem [20].

A SAT problem is represented in CNF (conjunctive normal form). The CNF format consists of Boolean variables, a set of literals, and a set of clauses. A literal is a variable or a negation of a variable. A clause consists of literals connected by logical or. Then, the problem is finding the existence of variable values such that the value of clauses connected by logical and is true. For example, consider Boolean variables $x_1$, $x_2$, and $x_3$. Then, $x_1 \lor \neg x_2 \lor \neg x_3$ are literals. Clause examples are: $x_1 \lor \neg x_2 \lor x_3$, $x_2 \lor x_3$, etc. A satisfiable example CNF is,

$$(x_1 \lor \neg x_2 \lor \neg x_3) \land (x_2 \lor x_3) \land (\neg x_1 \lor \neg x_2).$$

SAT is known to be NP complete. On the other hand, recent SAT solvers can handle large SAT problems because SAT solving techniques have advanced significantly [21]. It is possible to assess the rearrangeability from the satisfiability by modeling the routing problem of a banyan-type network into SAT. The solver returns the variable values that satisfy the constraints if the problem is satisfiable. This value represents the connection routes that achieve the given permutation. Using the SAT modeling approach, it is thus unnecessary to develop a routing algorithm for the considered network.

2.3. Problem description

Past studies have proved the rearrangeability of some $2n - 1$ stage networks constructed by $2 \times 2$ unit switches. However, the rearrangeability is not completely clarified for arbitrary $2n - 1$ stage
rearrangeable networks. For the efficient assessment of rearrangeability, it is mandatory to develop a method to easily judge whether connections can be established in the network for a given input/output permutation or not. As such a method, this study examines the CNF-SAT modeling. Through the CNF-SAT modeling and SAT solver, the study examines the connection capability of $2n - 1$ stage banyan-type networks for various permutations. It is expected that this approach accelerates theory development on the rearrangeability of banyan-type networks.

3. Related work

Various multistage switching networks constructed with $2 \times 2$ switches have been proposed. These networks include shuffle-exchange networks [3], omega networks [4], baseline networks [5], and banyan networks [6]. These networks are originally presented in the configuration of $n$ stages for $2^n$ network inputs/outputs. These $n$-stage networks are blocking. A greater number of stages are necessary to build a rearrangeable network with $2 \times 2$ unit switches.

The concept of rearrangeable switching networks has been known since 1950’s. An early study on a rearrangeable switching network is found in [22], which investigates a three stage rearrangeable network. Based on such a three stage rearrangeable network, Beneš [11] pointed out that the optimal rearrangeable network should have as many stages as possible, switches that are as small as possible, and the largest switches being in the middle stage. This principle leads to a $2n - 1$ stage rearrangeable network with $2 \times 2$ unit switches. Hence, the network is widely known as the “Beneš network,” an example of which is depicted in figure 2.

Beneš [12] also presented a conjecture that at least $2n - 1$ stage are necessary to construct a rearrangeable network that consists of $2 \times 2$ unit switches and has $2^n$ inputs/outputs. Many studies have been done on the proof of Beneš’s conjecture and rearrangeable networks other than the Beneš network [13]. Among these studies, Wu and Feng [14] presented a routing algorithm for $3n - 1$ stage shuffle-exchange network. Their algorithm can establish connections for any input/output permutations, and thus the rearrangeability of $3n - 1$ stage shuffle-exchange network was proved. Huang and Tripathi [15] showed that $2n - 1$ stages are necessary and $3n - 3$ stages are sufficient for a rearrangeable shuffle-exchange network. Some studies claimed that Beneš’s conjecture is proved [16, 17]. However, it was reported [13, 18, 19] that these proofs are not correct.

Rearrangeability is evaluated for the tandem connection of two $n$-stage banyan-type networks [18, 19]. A $2n$ stage network is created by connecting two $n$ stage networks in tandem. A $2n - 1$ stage network can be created by removing a trivially redundant stage from a $2n$ stage tandem cascade. The approach of [18, 19] focuses on how terminal indices’ bit positions are permuted through exchanges and introduces the concepts of guide and trace. For each of two $n$ stage networks, the sufficient condition for rearrangeability is shown as the condition for the guide and trace. The condition demonstrated by Li and Tan [19] appears to reveal rearrangeability for a wide range of $2n - 1$ and $2n$ stage banyan-type networks. However, other rearrangeable $2n - 1$ stage banyan-type networks that do not satisfy the condition may exist because their result is a sufficient condition.

4. CNF-SAT model

For the CNF-SAT modeling, we need to uniquely identify the unit switches, links, and connections by providing appropriate indices. This study investigates the network with $N = 2^n$ inputs/outputs and $2n - 1$ stage. As shown in figure 1, the inputs and outputs of the network, as well as unit switches, are indexed as $00...0(2^n), 00...1(2^n), ..., 11...1(2^n)$, namely, $0, 1, 2, ..., N - 1$ in decimal. The stages are indexed $1, 2, ..., 2n - 1$. Meanwhile, exchanges between stages $j$ and $j + 1$ is termed “exchange $j$.” The exchange between the inputs and stage 1 is exchange 0, whereas between stage $2n - 1$ and the network outputs is exchange $2n - 1$. Unit switches are indexed as $0, 1, ..., N/2 - 1$ from top to bottom in each stage. Then, the indices of two inputs and outputs of unit switch $i$ is $2i$ and $2i + 1$. Let us assume that a link in exchange $j$ is identified by the input index $k$ ($0 \leq k \leq N / 2 - 1$) of stage $j + 1$ ($0 \leq j < 2n - 1$) or the output ($j = 2n - 1$). With this scheme, it becomes possible to globally identify a link by the pair “$j$, $k$” of exchange index $j$ and link index $k$. 


The bit position permutation performed by exchange $j$ is denoted by function $\sigma(\bullet)$, which returns the right-side index connected to left-side index $\bullet$. For example, if exchange $j$ performs bit permutation $(1 \ 4)$, $\sigma(0001_{(2)}) = 1000_{(2)}$.

Let $d(i)$ denote the index of the output terminal connected to input terminal $i$. Additionally, let us identify a connection by the index of the network input. Namely, connection $i$ means a connection set up between input $i$ and output $d(i)$. The request for connections is given by $d(0), d(1), \ldots, d(N-1)$, which is a permutation of $0, 1, \ldots, N-1$.

The routes of connections are represented by the following Boolean variable $x_{i,j,k}$.

$$x_{i,j,k} = \begin{cases} 1, & \text{if connection } i \text{ goes through link } k \text{ of exchange } j \\ 0, & \text{otherwise} \end{cases}$$  \hfill (1)

The restriction that the routes must satisfy is represented as a SAT problem for variables $x_{i,j,k}$’s. The restrictions on connections include:

- A single connection occupies each link.
- If a connection goes through an input of a unit switch, it goes out to one of its two outputs.
- In exchange 0, connection $i$ goes through link $\sigma(i)$.
- In exchange $2n-1$, connection $i$ goes through link $d(i)$.

These restrictions are represented by a conjunction of clauses $C_1, C_2, C_3, \text{ and } C_4$. So then, the SAT problem is written in CNF as follows.

$$C_1 \wedge C_2 \wedge C_3 \wedge C_4$$  \hfill (2)

Conjunction $C_1$ states that any two connections must not share the same link. Thus, it is necessary to prevent both of $x_{i,j,k}$ and $x_{i',j,k}(i \neq i')$ from being 1. Since this is required for any combinations of $i, i', j, k$,

$$C_1 = \bigwedge_{0 \leq i < N \atop 0 \leq j < 2n} \left( \neg x_{i,j,k} \lor \neg x_{i',j,k} \right)$$  \hfill (3)

The second restriction is modeled by five clauses, denoted by $C_{2,1}(i, j, l), C_{2,2}(i, j, l), C_{2,3}(i, j, l), C_{2,4}(i, j, l), \text{ and } C_{2,5}(i, j, l)$. $C_2$ is the conjunction of these clauses for every combination of connections, stages, and unit switches,

$$C_2 = \bigwedge_{0 \leq i < N \atop 0 \leq j < 2n \atop 0 \leq l < N/2} \left( C_{2,1}(i, j, l) \wedge C_{2,2}(i, j, l) \wedge C_{2,3}(i, j, l) \wedge C_{2,4}(i, j, l) \wedge C_{2,5}(i, j, l) \right)$$  \hfill (4)

According to the link index definition, the indices of the links connected to two inputs of unit switch $l$ in stage $j$ are $2l$, and $2l+1$. Let $a$ and $b$ denote indices of the links connected the outputs of the unit switch. The output indices of the switch are $2l$ and $2l+1$ and the link indices are derived by permuting bit positions according to $\sigma(\ )$. Thus,

$$a = \sigma_j(2l)$$ \hfill (5)

$$b = \sigma_j(2l+1)$$ \hfill (6)

The relationship among a unit switch and indices is illustrated in figure 3.
Clauses $C_{2,1}(i, j, l)$ and $C_{2,2}(i, j, l)$ assert that connection $i$ does not pass the output-side links when $i$ does not pass any of two input-side links. Thus, if $x_{i, j - 1, 2l}$ and $x_{i, j - 1, 2l + 1}$ are 0, $x_{i, j, a}$ and $x_{i, j, b}$ must be also 0. Consequently, these clauses are written as follows.

$$C_{2,1}(i, j, l) = x_{i, j - 1, 2l} \lor x_{i, j - 1, 2l + 1} \lor \neg x_{i, j, a}$$  \hspace{1cm} (7)

$$C_{2,2}(i, j, l) = x_{i, j - 1, 2l} \lor x_{i, j - 1, 2l + 1} \lor \neg x_{i, j, b}$$  \hspace{1cm} (8)

If connection $i$ goes through either one of two inputs, $i$ must go out through one of two outputs. For example, if $x_{i, j - 1, 2l}$ is 1, $x_{i, j, a}$ and $x_{i, j, b}$ must be 1. This requirement is represented by $C_{2,3}(i, j, l)$ and $C_{2,4}(i, j, l)$, which are defined as follows.

$$C_{2,3}(i, j, l) = \neg x_{i, j - 1, 2l} \lor x_{i, j, a} \lor x_{i, j, b}$$  \hspace{1cm} (9)

$$C_{2,4}(i, j, l) = \neg x_{i, j - 1, 2l - 1} \lor x_{i, j, a} \lor x_{i, j, b}$$  \hspace{1cm} (10)

Connection $i$ is not allowed to pass both of two outputs simultaneously. Therefore, both of $x_{i, j, a}$ and $x_{i, j, b}$ must not be 1. This condition is represented by $C_{2,5}(i, j, l)$ as follows.

$$C_{2,5}(i, j, l) = \neg x_{i, j, a} \lor \neg x_{i, j, b}$$  \hspace{1cm} (11)

Since connection $i$ is set up between network input $i$ and output $d(i)$, $i$ goes through the links attached input $i$ and output $d(i)$. These conditions are stated by $C_3$ and $C_4$, which are defined as,

$$C_3 = \left( \bigwedge_{0 \leq s < N} X_{j, 0, \sigma_1(s)} \right) \land \left( \bigwedge_{1 \leq r < l \leq s', 0 \leq s < N} \neg X_{j, r, \sigma_2(s)} \right),$$  \hspace{1cm} (12)

$$C_4 = \left( \bigwedge_{0 \leq s < N} X_{j, 2n - 1, \sigma_1(s) + d(i)} \right) \land \left( \bigwedge_{1 \leq r < l \leq s', 0 \leq s < N} \neg X_{j, r, 2n - 1, d(i)} \right),$$  \hspace{1cm} (13)

Since all of the restrictions on $x_{i, j, k}$’s are described by equations (2)-(13), the feasibility of routes is found by solving equation (2) with a SAT solver.

5. Assessment of rearrangeability
This section assesses the rearrangeability of $2n - 1$ stage banyan-type networks by solving the SAT problems formulated for various input/output permutations. The assessment was executed as follows.
First, permutations of 0, 1, 2, …, \(N-1\) were generated and stored in files. Then, a program was executed to generate the SAT problem in the CNF formula defined by equations (2)-(13) for each permutation and the assessed network. The result is stored in a DIMACS format text file and fed to a SAT solver. The output of the SAT solver shows whether the permutation is achievable by the network or not. As the SAT solver, this study employed CaDiCaL [23] version 1.0.3.

The assessment was performed for the cases of \(n=3\) and \(n=4\). For \(n=3\), the number of possible permutations is \(2^3! = 40320\). It is easy to solve the problems for all of these possible permutations. If all of \(40320\) problem cases are satisfiable, it is proved that the network is rearrangeable for \(n=3\). Thus, satisfiability was checked for all of \(40320\) cases when \(n=3\).

When \(n=4\), the number of permutations is \(2^4! \approx 2.1 \times 10^{13}\). It is impossible to solve SAT problems for all of these cases exhaustively. Instead, SAT problems were generated and solved for randomly generated \(1 \times 10^5\) permutations. If the problems are satisfiable for all of \(1 \times 10^5\) permutations, it will be concluded that blocking rarely occurs in the network. In other words, it will be likely that the network is rearrangeable. The algorithm of [24] generated random permutations. Among the generated permutations, duplications were checked, and duplicated permutations were discarded. With this scheme, \(1 \times 10^5\) distinct permutations were obtained. For the generated permutations, SAT problems were generated and solved.

For \(n=3\), the following networks BN1, BN2, …, BN10 were tested.

BN1 \([id : (1 3) : (2 3) : (1 3) : (2 3) : id];\)
BN2 \([id : (1 3) : (2 3) : (2 3) : (1 3) : id];\)
BN3 \([id : (2 3) : (1 3) : (2 3) : (1 3) : id];\)
BN4 \([id : (1 3) : (1 3) : (2 3) : (2 3) : id];\)
BN5 \([id : (1 3) : (2 3) : (1 3) : (1 3) : id];\)
BN6 \([id : (1 2 3) : (2 3) : (1 2 3) : (2 3) : id];\)
BN7 \([id : (1 2 3) : (2 3) : (2 3) : (1 2 3) : id];\)
BN8 \([id : (1 2 3) : (1 2 3) : (1 2 3) : (2 3) : id];\)
BN9 \([id : (1 2 3) : (2 3) : (1 2 3) : (1 2 3) : id];\)
BN10 \([id : (2 3) : (1 2 3) : (1 2 3) : (1 2 3) : id];\)

The result is shown in table 1. The table presents the numbers of satisfiable and unsatisfiable cases for each network.

| Network | Satisfiable Cases | Unsatisfiable Cases |
|---------|------------------|---------------------|
| BN1     | 40320 (100 %)    | 0 (0 %)             |
| BN2     | 40320 (100 %)    | 0 (0 %)             |
| BN3     | 40320 (100 %)    | 0 (0 %)             |
| BN4     | 20736 (51.43 %)  | 19584 (48.57 %)     |
| BN5     | 20736 (51.43 %)  | 19584 (48.57 %)     |
| BN6     | 40320 (100 %)    | 0 (0 %)             |
| BN7     | 9216 (22.86 %)   | 31104 (77.14 %)     |
| BN8     | 40320 (100 %)    | 0 (0 %)             |
| BN9     | 40320 (100 %)    | 0 (0 %)             |
| BN10    | 20736 (51.43 %)  | 19584 (48.57 %)     |
The result shown in table 1 proves that six of the tested networks are rearrangeable. The rearrangeable networks are BN1, BN2, BN3, BN6, BN8, and BN9. Figure 4 shows BN7, which is not rearrangeable. This network and Beneš network (figure 2) look alike except for exchange 4. However, there is a great difference between these networks. Namely, the 3-stage subnetwork constructed by stages 3, 4, and 5 of BN7 is not routable, whereas the subnetwork constructed by stages 3-5 of the Beneš network is routable. By checking the route from switch 0 of stage 3 to switch 1 of stage 5, it is easy to see that the part of stages 3-5 (in other words, second half) of BN7 is not routable. Similarly, the second half of the network is not routable in BN4 and BN5. The table shows that these networks are also not rearrangeable. BN10 is not rearrangeable, though the subnetwork constructed by stages 3-5 is routable. However, the subnetwork of stages 1, 2, and 3, namely, the first half, is not routable for this network. Meanwhile, the common characteristic of rearrangeable cases is that both the network’s first and second halves are routable for \( n = 3 \). However, this is not a sufficient condition for rearrangeability.

For \( n = 4 \), the following networks BN11, BN12, ..., BN20 were tested.

- \( \text{BN}_{11} \) \([id : (1 \ 4) : (2 \ 4) : (3 \ 4) : (1 \ 4) : (2 \ 4) : (3 \ 4) : id]_4\)
- \( \text{BN}_{12} \) \([id : (1 \ 4) : (2 \ 4) : (3 \ 4) : (2 \ 4) : (1 \ 4) : id]_4\)
- \( \text{BN}_{13} \) \([id : (2 \ 4) : (1 \ 4) : (3 \ 4) : (2 \ 4) : (1 \ 4) : id]_4\)
- \( \text{BN}_{14} \) \([id : (1 \ 4) : (2 \ 4) : (3 \ 4) : (1 \ 4) : (2 \ 4) : id]_4\)
- \( \text{BN}_{15} \) \([id : (2 \ 4) : (1 \ 4) : (3 \ 4) : (1 \ 4) : (2 \ 4) : id]_4\)
- \( \text{BN}_{16} \) \([id : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (3 \ 4) : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (3 \ 4) : id]_4\)
- \( \text{BN}_{17} \) \([id : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (2 \ 3 \ 4) : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (3 \ 4) : id]_4\)
- \( \text{BN}_{18} \) \([id : (1 \ 2 \ 3 \ 4) : (1 \ 2 \ 3 \ 4) : (3 \ 4) : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (3 \ 4) : id]_4\)
- \( \text{BN}_{19} \) \([id : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (3 \ 4) : (1 \ 2 \ 3 \ 4) : (1 \ 2 \ 3 \ 4) : (3 \ 4) : id]_4\)
- \( \text{BN}_{20} \) \([id : (2 \ 3 \ 4) : (1 \ 2 \ 3 \ 4) : (3 \ 4) : (1 \ 2 \ 3 \ 4) : (2 \ 3 \ 4) : (3 \ 4) : id]_4\)

The result for the case of \( n = 4 \) is summarized in table 2. Let us define the first half part as the 4-stage subnetwork configured by stage 1 to stage 4 and the second half as the 4-stage subnetwork configured by stage 4 to stage 7. The result shown in table 2 implies that BN11, BN12, BN13, BN16, BN17, and BN19 are probably rearrangeable. For these networks, both of the first and second half parts are routable. Meanwhile, BN14 and BN20 do not have this characteristic. Namely, the second half part of BN14 and the first half part of BN20 are not routable. Table 2 indicates that these networks are not rearrangeable. Meanwhile, both of the first and second half parts are routable for BN15 and BN18, and they are not rearrangeable. Thus, a \( 2n - 1 \) stage network is not always rearrangeable even if both of the first and second half parts are routable. Notably, BN19 is configured by swapping the first and second half parts of BN19. Thus, it is expected that the characteristics of these networks will not greatly differ. However, BN19 can establish connections for every tested permutation, while BN18 cannot establish...
connections for some permutations. Further studies are necessary to understand why this difference occurs between BN_{18} and BN_{19}.

| Network | Satisfiable Cases | Unsatisfiable Cases |
|---------|------------------|---------------------|
| BN_{11} | 100000 (100 %)   | 0 (0 %)             |
| BN_{12} | 100000 (100 %)   | 0 (0 %)             |
| BN_{13} | 100000 (100 %)   | 0 (0 %)             |
| BN_{14} | 36311 (36.31 %)  | 63689 (63.69 %)     |
| BN_{15} | 99974 (99.97 %)  | 26 (0.03 %)         |
| BN_{16} | 100000 (100 %)   | 0 (0 %)             |
| BN_{17} | 100000 (100 %)   | 0 (0 %)             |
| BN_{18} | 98500 (98.50 %)  | 1500 (1.50 %)       |
| BN_{19} | 100000 (100 %)   | 0 (0 %)             |
| BN_{20} | 9760 (9.76 %)    | 90240 (90.24 %)     |

From the above result, the new rearrangeable network can be derived. A 16 × 16 7-stage rearrangeable network is obtained by two 8 × 8 5-stage and sixteen 2 × 2 unit switches as shown in figure 5. For 8 × 8 5-stage rearrangeable networks, any BN_{1}, BN_{2}, BN_{3}, BN_{6}, BN_{8}, and BN_{9} are usable. By repeating this process, 2^{n} × 2^{n} (n > 3) rearrangeable networks can be constructed. This process is the same as that of constructing a Beneš network. However, the obtained network is not a Beneš network because stages n − 2 to n + 2 are different. Thus, a new 2n − 1 rearrangeable stage network is obtainable via the presented SAT modeling.

![Figure 5](image_url)
6. Conclusion
The rearrangeability of banyan-type networks was investigated in this paper. CNF-SAT modeling was investigated as a method to check the rearrangeability of a banyan-type network. The proposed method uses Boolean variables to represent network connection routes. The variables were then restricted by the conditions satisfied by the routes at links and unit switches. The restrictions are written as a CNF-SAT problem, for which a SAT solver determines satisfiability. The network’s rearrangeability is denied if the SAT problem for a given network and permutation is unsatisfiable. The paper tested various $2n - 1$ stage banyan-type networks for $n = 3$ and 4. The SAT problem was solved for every possible permutation when $n = 3$. For $n = 4$, a large number of randomly generated permutations were examined. As a result, several rearrangeable networks were found for $n = 3$. Additionally, the result for $n = 4$ shows several networks that are likely to be rearrangeable. It was also pointed out that $2^n \times 2^n$ rearrangeable networks for $n > 3$ can be obtained from the networks found to be rearrangeable for $n = 3$ through the proposed method.

The proposed method is expected to be useful in fully establishing the theory of banyan-type network rearrangeability. If someone conjectures or proves the rearrangeability of a network, for example, the correctness of proof or conjecture will be checked by solving the SAT problem. It is also possible to find rearrangeable networks that have not been discovered before using the proposed method.

It is an open problem to clearly describe the common characteristics of the rearrangeable networks discovered by the proposed SAT modeling. Further studies are necessary to concretely describe the necessary and sufficient condition on the rearrangeability. It is also an open problem to determine the relationship between the rearrangeable networks found in this study and the rearrangeability conditions revealed by past studies.

References
[1] Hwang F H 2004 The Mathematical Theory of Nonblocking Switching Networks 2nd Edition (Singapore: World Scientific)
[2] Kabaciński W 2005 Nonblocking Electronic and Photonic Switching Fabrics (New York: Springer)
[3] Stone H S 1971 Parallel processing with the perfect shuffle IEEE Trans. Comput. C-20, 153–61
[4] Lawrie D H 1975 Access and alignment of data in an array processor IEEE Trans. Comput. C-24 1145–55
[5] Wu C -L and Feng T -Y 1980 On a class of multistage interconnection networks IEEE Trans. Comput. C-29 694–702
[6] Goke L R and Lipovski G J 1973 Banyan networks for partitioning multiprocessor systems Proc. 1st Annual Symp. on Computing Architecture (Gainsville, USA, Dec. 1973) pp 21–8.
[7] Hui J Y and Arthurs E 1987 A broadband packet switch for integrated transport IEEE J. Sel. Areas Commun. SAC-5 1264–73
[8] Haney M W and Christensen M P 1996 Sliding banyan network J. Light. Technol. 14 703–10
[9] Cheng Q, Huang Y, Yang H, Bahadori M, Abrams N, Meng X, Glick M, Liu Y, Hochberg M and Bergman K 2019 Silicon photonic switch topologies and routing strategies for disaggregated data centers IEEE J. Sel. Topics in Quantum Electron. 26 8302010
[10] Das A, Zhang G, Mojaver H R and Liboiron-Ladouceur O 2020 Low loss $8 \times 8$ silicon photonic banyan switch in Proc. IEEE Photonics Conf. (Vancouver, Canada, Sept. 2020) pp 1–2
[11] Beneš V E 1965 Mathematical Theory of Connecting Networks, (New York and London: Academic Press)
[12] Beneš V E 1975 Proving the rearrangeability of connecting networks by group calculations Bell Syst. Tech. J. 54 421–34
[13] Ngo H Q and Du D -Z 2001 Remarks on Beneš conjecture Switching Networks: Recent Advances ed D -Z Du and H Q Ngo (Dordrecht: Kluwer Academic Publishers) pp 257–8
[14] Wu C L and Feng T Y 1981 The universality of the shuffle-exchange network IEEE Trans. Comput. C-30 324–32
[15] Huang S T and Tripathi S K 1986 Finite state model and compatibility theory: new analysis tools for permutation networks *IEEE Trans. Comput.* **C-35** 591–601

[16] Feng T -y and Seo S W 1994 A new routing algorithm for a class of rearrangeable networks *IEEE Trans. Comput.* **43** 1270–80

[17] Çam H 2003 Rearrangeability of (2n – 1)-stage shuffle-exchange networks *SIAM J. Comput.* **32** 557–85

[18] Tan X and Li S -Y R 2007 Rearrangeability of tandem cascade of banyan-type networks *IEICE Trans. on Inf. and Syst.* **E90-D** 67–74

[19] Li S Y R and Tan X J 2009 On rearrangeability of tandem connection of banyan-type networks *IEEE Trans. Commun.* **57** 164–70

[20] Claessen K, Een N, Sheeran M, Sörensson N, Voronov A and Åkesson K 2009 SAT-solving in practice, with a tutorial example *Discrete Event Dyn. Syst.* **19** 495–524

[21] Gong W and Zhou X 2017 A survey of SAT solver *Proc. the 1st Int. Conf. on Applied Mathematics and Computer Science* (Rome, Italy, Jan. 2017) p 020059

[22] Paull M C 1962 Reswitching of connecton networks *Bell Syst. Tech. J.* **41** 833–55

[23] Biere A 2019 CaDiCaL at the SAT Race 2019 *Proc. SAT RACE 2019* (Lisbon, Portugal, July 2019) pp 8-9

[24] Durstenfeld R 1964 Algorithm 235: random permutation *Commun. ACM* **7** 420