Wide-Sense Nonblocking Clos Networks Under Packing Strategy

Yuanyuan Yang, Senior Member, IEEE, and Jianchao Wang, Member, IEEE Computer Society

Abstract—In this paper, we study wide-sense nonblocking conditions under packing strategy for the three-stage Clos network, or $v(m, n, r)$ network. Wide-sense nonblocking networks are generally believed to have lower network cost than strictly nonblocking networks. However, the analysis for the wide-sense nonblocking conditions is usually more difficult. Moore (cited in Benes' book [2]) proved that a $v(m, n, 2)$ network is nonblocking under packing strategy if the number of middle stage switches $m \geq \frac{2n}{r+1}$. This result has been widely cited in the literature, and is even considered as the wide-sense nonblocking condition under packing strategy for the general $v(m, n, r)$ networks in some papers, such as [7]. In fact, it is still not known that whether the condition $m \geq \frac{2n}{r+1}$ holds for $v(m, n, r)$ networks when $r \geq 3$. In this paper, we introduce a systematic approach to the analysis of wide-sense nonblocking conditions for general $v(m, n, r)$ networks with any $r$ value. We first translate the problem of finding the nonblocking condition under packing strategy for a $v(m, n, r)$ network to a set of linear programming problems. We then solve this special type of linear programming problems and obtain a closed form optimum solution. We prove that the necessary condition for a $v(m, n, r)$ network to be nonblocking under packing strategy is $m \geq \left(\frac{2 - \frac{1}{F_{r-1}}}{F_{r-1}}\right)n$, where $F_{r-1}$ is the Fibonacci number. In the case of $n \leq F_{r-1}$, this condition is also a sufficient nonblocking condition for packing strategy. We believe that the systematic approach developed in this paper can be used for analyzing other wide-sense nonblocking control strategies as well.

Index Terms—Interconnection networks, wide-sense nonblocking, routing control strategies, packing, linear programming, Fibonacci numbers.

1 INTRODUCTION

The well-known Clos networks [1], [2] have been widely used for multiprocessor interconnections and data communications. Some recent applications include the NEC ATOM switch designed for BISDN [4], the IBM GF11 multiprocessor [5], and ANSI Fibre Channel Standard for interconnection of processors to the I/O system. In general, a three-stage Clos network with $N$ input ports and $N$ output ports has $r$ switch modules of size $n \times m$ in the input stage, $m$ switch modules of size $r \times r$ in the middle stage, and $r$ switch modules of size $m \times n$ in the output stage. Such a three-stage network is denoted as a $v(m, n, r)$ network. A general schematic of a $v(m, n, r)$ network is shown in Fig. 1.

The connecting capability of the network in Fig. 1 is determined by the parameters $n$, $r$, and $m$. For a given $n$ and $r$, varying $m$ will vary its connecting capability. Therefore, the main focus of the study on Clos networks has been on finding the minimum value of the network parameter $m$ for a certain type of connecting capability to achieve the minimum network cost.

The $v(m, n, r)$ networks have been extensively studied in the literature. We summarize some fundamental results below. Clos showed [1] that a $v(m, n, r)$ network is strictly nonblocking if the number of middle stage switches $m \geq 2n - 1$. If the network satisfies this condition, any middle stage switch may be chosen arbitrarily to make a connection without any rearrangement of existing connections in the network. Fewer middle stage switches are sufficient if only rearrangeability is needed. Benes showed [3] that a $v(m, n, r)$ network is rearrangeable if the number of middle stage switches $m \geq n$. Under this condition, satisfying a connection request may require rearrangement of existing connections in the network. Since the rearrangement causes disruption of on-going communications and time delay in path routing, nonblocking capability is generally desirable.

Wide-sense nonblocking capability can be considered as a compromise between strictly nonblocking capability and rearrangeability. For a wide-sense nonblocking network, an “intelligent” routing control strategy must be employed to govern the process of path routing. Through carefully selecting the paths used to satisfy the current connection request, the nonblocking capability for future connection requests can be maintained and, at the same time, lower network cost can be achieved. However, in general, the analysis of wide-sense nonblocking conditions is much more difficult than that of strictly nonblocking conditions.

A commonly used routing control strategy for wide-sense nonblocking $v(m, n, r)$ networks is the so-called packing strategy [2]. Under packing strategy, a connection is realized on a path found by trying the most used part of the network first and the least used part last. For a $v(m, n, r)$ network, this means that, when choosing a middle stage switch for satisfying a connection request, an empty middle stage switch is not used unless there is not any partially
filled middle stage switch that can satisfy this connection request. If we specify how to choose one switch from more than one partially filled middle stage switches that can satisfy the connection request, we get more specific packing strategies.

Packing strategy can also be combined with repacking. Repacking means when a connection is released one or more existing connections are moved to the most used part of the network. Repacking essentially is a type of rearrangement, although the rearrangement is performed at the time a connection is released, not at the time a connection is established.

It is generally believed that packing/repacking can improve network performance and reduce network cost. There has been some work on packing/repacking routing in \( v(m, n, r) \) networks in the literature [8], [9], [10], [6], [2]. Ackroyd [8] and Girard and Hurtubise [9] showed by simulation that packing/repacking strategies have lower blocking probability over random routing. Mun et al. [10] proposed a probabilistic model for the analysis of blocking probability under packing strategy and their model confirmed the Ackroyd’s simulation results. Jajszczyk and Jekel [6] derived the necessary and sufficient condition for a repackable \( v(m, n, r) \) network and showed that a repackable network requires fewer middle stage switches than a strictly nonblocking network.

However, the theoretical analysis of the nonblocking conditions under packing strategy is difficult. The only known result concerning packing strategy is Moore’s work (cited in Benes’ book [2]). Moore [2] analyzed the wide-sense nonblocking conditions under packing strategy for \( v(m, n, 2) \) networks, and showed that a \( v(m, n, 2) \) network is nonblocking under packing strategy if \( m \geq \left\lceil \frac{3n}{2} \right\rceil \), which is a significant improvement over the strictly nonblocking condition \( m \geq 2n - 1 \). This result has been widely cited in the literature, e.g., [6], [7]. In some papers, such as [7], it is even considered as the wide-sense nonblocking condition under packing strategy for the general \( v(m, n, r) \) networks with any \( r \) value. In fact, it is still not known whether the condition \( m \geq \left\lceil \frac{3n}{2} \right\rceil \) holds for \( v(m, n, r) \) networks when \( r \geq 3 \) and, if the answer is negative, what the wide-sense nonblocking conditions under packing strategy are for the general \( v(m, n, r) \) networks.

Moore’s approach [2] is based on enumerating all possible states of a middle stage switch and considering all state transitions. As can be seen in the next section, the number of states grows very fast as \( r \) increases. Therefore, it is difficult to directly extend Moore’s approach to analyzing wide-sense nonblocking conditions for general \( v(m, n, r) \) networks for \( r \geq 3 \).

In this paper, we will introduce a systematic approach to the analysis of wide-sense nonblocking conditions for general \( v(m, n, r) \) networks with any \( r \) value. We first translate the problem of finding the nonblocking condition under packing strategy for \( v(m, n, r) \) networks to a set of linear programming problems. We then show that, for this special type of linear programming problems, we can find a closed form optimum solution and, thus, obtain a necessary nonblocking condition for the original problem. We also discuss that, for some values of \( n \) and \( r \), this condition is a sufficient nonblocking condition.

The rest of the paper is organized as follows: Section 2 provides definitions on connection requests and network states. Section 3 describes the general nonblocking condition suitable to any control strategy for \( v(m, n, r) \) networks. Section 4 derives the special constraints under packing strategy and describes the nonblocking condition under packing strategy in terms of a set of linear programming problems. Section 5 solves the linear programming problem formulated in Section 4. Section 6 finds the maximum objective values among a set of linear programming problems and obtains the closed form nonblocking condition under packing strategy. Section 7 constructs an example that actually reaches the necessary condition. Section 8 concludes the paper. Finally, Appendices A, B, C, and D give some detailed mathematical proofs.

2 Connection Requests and Network States

In this section, we provide some definitions on connection requests and network states of a \( v(m, n, r) \) network.

For any \( i, j \in \{1, 2, \ldots, r\} \), we denote a connection request from an input of input stage switch \( i \) to an output of output stage switch \( j \) in a \( v(m, n, r) \) network as \( (i, j) \). Based on the network structure in Fig. 1, if a connection request \( (i, j) \) is satisfied, it must go through a middle stage switch from input \( i \) to output \( j \) of the switch. In this case, we say connection \( (i, j) \) passes this middle stage switch. For an \( r \times r \) middle stage switch, there will be up to \( r \) mutually disjoint connections of the form \( (i_1,j_1),(i_2,j_2),\ldots,(i_r,j_r) \), where \( 1 \leq k \leq r, 1 \leq i_1 < i_2 < \cdots < i_r \leq n, 1 \leq j_1,j_2,\ldots,j_r \leq r \), and for any \( 1 \leq s, t \leq r, j_s \neq j_t \) when \( s \neq t \).

In Moore’s approach [2], the state of a middle stage switch was defined as all connections passing that middle stage switch. Take a middle stage switch as an example: there are a total of \( 2^C \) possible states of a middle stage switch. For an \( r \times r \) middle stage switch, there are \( 2^C \) possible states. We can see that the number of states increases dramatically when \( r \) gets larger. Thus, it is difficult to directly generalize Moore’s approach
to general \(v(m,n,r)\) networks with any \(r\) value. In the following, we will introduce a systematic approach to the analysis of wide-sense nonblocking condition for general \(v(m,n,r)\) networks. We then apply this approach to a special packing strategy and obtain a necessary nonblocking condition for general packing strategy.

Without loss of generality, in the rest of the paper, we will always assume that the current connection request to be satisfied is \((1,1)\), that is, we are about to connect an input of input stage switch 1 to an output of output stage switch 1. We define the middle stage switch states with respect to a connection request \((1,1)\) as follows.

**Definition 1.** For an \(r \times r\) middle stage switch, the switch is said to be in state \([1,1]\) if and only if there exists a connection \((1,1)\) or there exist two connections \((1,j)\) and \((i,1)\) in the switch where \(i,j \neq 1\).

**Definition 2.** For an \(r \times r\) middle stage switch, the switch is said to be in state \([i,j]\) where \(j \neq 1\), if and only if there exists a connection \((1,j)\) and does not exist any connection \((i,1)\) for any \(i\) in this switch.

**Definition 3.** For an \(r \times r\) middle stage switch, the switch is said to be in state \([i,1]\) where \(i \neq 1\), if and only if there exists a connection \((i,1)\) and does not exist any connection \((1,j)\) for any \(j\) in this switch.

**Definition 4.** Any middle stage switch without connections \((1,1)\), \((1,j)\), or \((i,1)\) in it is said to be in a nonforbidden state with respect to a connection request \((1,1)\). Each of the states defined in Definitions 1-3 is referred to as a forbidden state with respect to a connection request \((1,1)\).

Clearly, a new connection request \((1,1)\) cannot be satisfied by a middle stage switch in a forbidden state. Note that the above definition on the states of a middle stage switch is unique. In fact, if a middle stage switch is said to be in state \([i_1,j_1]\) (where \(i_1 = 1\) or \(j_1 = 1\)) and in state \([i_2,j_2]\) (where \(i_2 = 1\) or \(j_2 = 1\)), then it must be that \(i_1 = i_2\) and \(j_1 = j_2\). Also, all middle stage switches in a nonforbidden state are considered in the same state. Therefore, for an \(r \times r\) middle stage switch, there are a total of \(2r\) different states, among which \(2r-1\) are forbidden states.

### 3 The General Nonblocking Condition

In this section, we derive the general nonblocking condition that is suitable to any control strategy.

Since we are about to satisfy a connection request \((1,1)\), we are interested in those middle stage switches in states \([1,2], [1,3], \ldots, [1,r], [2,1], [3,1], \ldots, [r,1],\) and \([1,1]\). For any \(i,j \in [1,2,\ldots,r]\), we denote the number of middle stage switches in state \([i,j]\) (where \(i = 1\) or \(j = 1\)) as \(x_{i,j}\). Since a connection request \((1,1)\) cannot be satisfied by any middle stage switch in states \([i,1], [1,j],\) or \([1,1]\), it will be satisfied by a middle stage switch in a nonforbidden state. Thus, the number of middle stage switches required for satisfying a connection request \((1,1)\) is

\[
\sum_{i=2}^{r} x_{i,1} + \sum_{j=2}^{r} x_{1,j} + x_{1,1} + 1.
\]

We have the following theorem concerning the general nonblocking condition of a \(v(m,n,r)\) network.

**Theorem 1.** The necessary and sufficient condition for a \(v(m,n,r)\) network to be nonblocking is

\[
m \geq \max_{\Delta} \left\{ \sum_{i=2}^{r} x_{i,1} + \sum_{j=2}^{r} x_{1,j} + x_{1,1} \right\} + 1,
\]

where, \(\Delta\) is a set of constraints for variables \(x_{1,1}, x_{1,1}\) and \(x_{1,j}\) for \((i,j) \neq (1,1)\).

**Proof.** Since the number of middle stage switches that cannot be used for satisfying a connection request \((1,1)\) is no more than \(\sum_{i=2}^{r} x_{i,1} + \sum_{j=2}^{r} x_{1,j} + x_{1,1}\), one more middle stage switch that is definitely in a nonforbidden state can be used for realizing the connection request \((1,1)\). Therefore, the sufficiency holds. On the other hand, let \(x_{1,1}, x_{1,j}\) for \((i,j) \neq (1,1)\), and \(x_{1,1}\) equal to some values that make \(\sum_{i=2}^{r} x_{i,1} + \sum_{j=2}^{r} x_{1,j} + x_{1,1}\) achieve its maximum under constraints \(\Delta\). Then, we need at least

\[
\max_{\Delta} \left\{ \sum_{i=2}^{r} x_{i,1} + \sum_{j=2}^{r} x_{1,j} + x_{1,1} \right\} + 1
\]

middle stage switches to satisfy a connection request \((1,1)\). Thus, the necessity holds.

Clearly, constraints \(\Delta\) are control strategy dependent. In the following, we will first introduce some general constraints that are suitable to any control strategy for the \(v(m,n,r)\) nonblocking network. Then, as a simple application of Theorem 1, we will derive the nonblocking condition for the strictly nonblocking \(v(m,n,r)\) network by using the general constraints.

We have the following lemma that is useful for establishing the general constraints for nonblocking \(v(m,n,r)\) networks.

**Lemma 1.** In a \(v(m,n,r)\) network, at any time, among all existing connections passing middle stage switches, there are no more than \(n\) connections of the form \((i,*)\) (i.e., starting from some input of the \(i\)th input stage switch), and there are no more than \(n\) connections of the form \((*,j)\) (i.e., connected to some output of the \(j\)th output stage switch), for any \(i,j \in [1,2,\ldots,r]\).

**Proof.** As shown in Fig. 2, consider input stage switch \(i\) and output stage switch \(j\). Since there are \(n\) inputs on input switch \(i\), we can have at most \(n\) connections starting from input switch \(i\) at any time. Therefore, we can have at most \(n\) connections of the form \((i,*)\) passing middle stage switches. Similarly, since there are \(n\) outputs on output switch \(j\), we can have at most \(n\) connections connected to output switch \(j\) and, therefore, \(n\) connections of the form \((*,j)\) passing middle stage switches.

By Lemma 1, we know that, before we satisfy a connection request \((1,1)\), we can have at most \(n-1\) existing connections of the form \((1,\ast)\) and at most \(n-1\) existing connections of the form \((\ast,1)\). We can obtain two general constraints as follows:
We now see how to apply these constraints to derive the nonblocking condition for strictly nonblocking $v(m, n, r)$ networks. Note that the objective function to be maximized in Theorem 1 is

$$\sum_{i=2}^{r} x_{i,1} + \sum_{j=2}^{r} x_{1,j} + x_{1,1} + 1 = \left(\sum_{i=2}^{r} x_{i,1} + x_{1,1} + 1\right) + \left(\sum_{j=2}^{r} x_{1,j} + x_{1,1} + 1\right) - (x_{1,1} + 1) \leq n + n - (x_{1,1} + 1) \leq 2n - 1.
$$

The maximum value $2n - 1$ can be achieved by setting $x_{1,1} = 0$, $\sum_{i=2}^{r} x_{i,1} = n - 1$, and $\sum_{j=2}^{r} x_{1,j} = n - 1$. Thus, by Theorem 1, the condition for the strictly nonblocking $v(m, n, r)$ network is $m \geq 2n - 1$.

Note that, under any control strategy, a wide-sense nonblocking $v(m, n, r)$ network should always satisfy general constraints (1) and (2). In addition, under different control strategies, it should also satisfy some other constraints that are strategy-dependent.

4 THE NONBLOCKING CONDITION UNDER PACKING STRATEGY

As mentioned earlier, the only rule of the general packing strategy is to choose a nonempty middle stage switch first. There are many variants of the general packing strategy depending on how to choose one switch from more than one nonempty middle stage switches. This makes the analysis of the nonblocking condition under the general packing strategy difficult. For easy analysis, in this paper, we consider a special packing strategy which is state-dependent. Under this strategy, when satisfying a connection request, we first consider those middle stage switches related to a particular connection, say, a connection $(1, 1)$. We will choose middle stage switches for satisfying a connection request in the following order: switches in state $[1, 1]$, switches in state $[1, j]$, switches in state $[i, 1]$, other nonempty middle stage switches, and, finally, empty middle stage switches. By employing the above packing strategy, we can derive a set of linear constraints, use the systematic approach in the previous section, and obtain a nonblocking condition under this strategy. The maximized nonblocking condition derived this way means that, in the worst case, to satisfy a connection request $(1, 1)$, at least this amount of middle stage switches are required, regardless of which variant of the packing strategy is used. Hence, this leads to a necessary condition for the general packing strategy. However, since this packing strategy is state-dependent, we cannot guarantee the nonblocking condition obtained is a sufficient condition for general packing strategy. For example, since this strategy favors a connection $(1, 1)$ and always considers middle stage switches in states $[1, 1]$ or $[1, 1]$ first, for other connections, say, a connection $(2, 2)$, we cannot say it can always be satisfied within this amount of middle stage switches. In the following, we will simply refer to this special packing strategy as packing strategy wherever it is unspecified.

4.1 The Constraints Under Packing Strategy

In this section, we derive additional constraints for variables $x_{1,1}$, $x_{1,1}$, and $x_{1,j}$ $(j \neq 1)$ when packing strategy is employed; that is, we investigate the relationship among $x_{1,1}$ and $x_{1,8}$ under packing strategy.

To find the constraints for $x_{1,1}$ and $x_{1,8}$, we need to know the history of how these $x_{1,8}$ and $x_{1,8}$ were formed. For example, consider the $x_{1,1} middle stage switches in state $[3, 1]$. Among these $x_{1,1} middle stage switches, let’s focus on the middle stage switch whose state was changed to $[3, 1]$ most recently. Clearly, there is a connection $(3, 1)$ in this switch. Assume that, before we satisfied this connection request $(3, 1)$, there were $x_{1,2} middle stage switches in state $[1, 2]$ and $x_{1,3} middle stage switches in state $[1, 3]$, and the connection request $(3, 1)$ could not be satisfied by any of these middle stage switches and must be satisfied by a middle stage switch in a nonforbidden state. Under the packing strategy, which looks for the middle stage switches in states $[1, 1]$, $[1, j]$ and $[i, 1]$ first, it must be the case that there had already been a connection $(3, *)$ in each of these $x_{1,2}$ middle stage switches in state $[1, 2]$ and in each of these $x_{1,3}$ middle stage switches in state $[1, 3]$. By Lemma 1, we know that the number of middle stage switches containing connection $(3, *)$ is no more than $n$. Thus, we obtain the following constraint:

$$x_{1,1} + x_{1,2} + x_{1,3} \leq n.$$

From the above example, we notice that the constraints depend on the history of the network states or the order in which the connection requests are satisfied. We define a connection sequence as follows:

**Definition 5.** Let $S$ be the set of $2r - 2$ connections $(2, 1)$, $(3, 1)$, ..., $(r, 1), (1, 2), (1, 3),..., (1, r)$. For the states of middle stage switches at a time, we define a connection sequence.
sequence $p_1p_2\ldots p_{2r-2}$ corresponding to the order of states $[i, 1]$ and $[1, j]$ ($i, j \neq 1$) as follows.

- $p_1p_2\ldots p_{2r-2}$ is a permutation of the $2r - 2$ connections in $S$;
- The connections in $S$ are permuted to $p_1p_2\ldots p_{2r-2}$ in the order of the last state transition of the corresponding state. In other words, for any $k_1$ and $k_2$, $p_{k_1} = (i_1, j_1)$ precedes $p_{k_2} = (i_2, j_2)$ in the connection sequence $p_1p_2\ldots p_{2r-2}$ if and only if the last middle stage switch in state $[i_1, j_1]$ reached its current state before the last middle stage switch in state $[i_2, j_2]$ reached its current state.

We are now in the position to derive the constraints under packing strategy.

**Lemma 2.** Assume that there are middle stage switches currently in states $[i_1, 1], [i_2, 1], \ldots, [i_k, 1]$, $2 \leq i_1, i_2, \ldots, i_k \leq r$. If, at this time, satisfying a connection request $(1, j)$ ($j \neq 1$) through a middle stage switch will cause the state of the switch to change to $[1, j]$, then, after this connection is satisfied, we have the following constraint

$$\sum_{s=1}^{k} x_{i_s, 1} + x_{1, j} \leq n. \quad (3)$$

Similarly, assume that there are middle stage switches currently in states $[1, j_1], [1, j_2], \ldots, [1, j_k]$, $2 \leq j_1, j_2, \ldots, j_k \leq r$. If, at this time, satisfying a connection request $(i, 1)$ ($i \neq 1$) through a middle stage switch will cause the state of the switch to change to $[i, 1]$, then, after this connection is satisfied, we have the following constraint

$$\sum_{s=1}^{k} x_{1, j_s} + x_{i, 1} \leq n. \quad (4)$$

**Proof.** Assume that there are middle stage switches currently in states $[i_1, 1], [i_2, 1], \ldots, [i_k, 1]$ and, at this time, satisfying a connection request $(1, j)$ through a middle stage switch will cause the state of the switch to change to $[1, j]$. This implies that the connection request $(1, j)$ could not be satisfied by any middle stage switches in states $[i_1, 1], [i_2, 1], \ldots, [i_k, 1]$. By packing strategy, since the connection request $(1, j)$ could not be satisfied by a middle stage switch in state $[i_1, 1]$ there must be some connection of the form $(\ast, j)$ in that switch. Similarly, there must be some connection of the form $(i, \ast)$ in the switches in states $[i_2, 1], \ldots, [i_k, 1]$. By Lemma 1, the number of connections of the form $(\ast, j)$ is no more than $n$. Thus, (3) holds. The proof of the second part of the lemma is similar. □

**Lemma 3.** For a connection sequence $p_1p_2\ldots p_{2r-2}$, we have a total of $2r - 1$ constraints for $2r - 1$ variables $x_{1, 1}, x_{1, 1}$ and $x_{1, j}$ ($i, j \neq 1$).

**Proof.** Consider $p_t$ for any $t \geq 2$. Without loss of generality, let $p_t = (1, j)$. If $p_1p_2\ldots p_{t-1}$ contain $(i_1, 1), (i_2, 1), \ldots, (i_k, 1)$, where $1 \leq k \leq t - 1$. By Lemma 2, we have the constraint

$$\sum_{s=1}^{k} x_{i_s, 1} + x_{1, j} \leq n.$$

Otherwise, if $p_1p_2\ldots p_{t-1}$ do not contain any $(i, 1)$ for $i \neq 1$, we have the constraint

$$x_{1, j} \leq n.$$

Symmetrically, for $p_t = (i, 1)$, we can obtain the corresponding constraint. Thus, each $p_t$ ($t \geq 2$) corresponds to a constraint. By adding general constraints (1) and (2) from the previous section, we have a total of $2r - 1$ constraints for $2r - 1$ variables $x_{1, 1}, x_{1, 1}, \ldots, x_{1, j}$ ($i, j \neq 1$). □

The proof of Lemma 3, in fact, tells us how to write these constraints for a connection sequence as well.

**Example.** Write constraints for connection sequence $p_1p_2p_3p_4 = (2, 1)(1, 2)(3, 1)(1, 3)$.

By Lemmas 2 and 3, we can obtain the following constraints:

- For $p_2 = (1, 2)$: $x_{2, 1} + x_{1, 2} \leq n$
- For $p_3 = (3, 1)$: $x_{1, 2} + x_{3, 1} \leq n$
- For $p_4 = (1, 3)$: $x_{2, 1} + x_{3, 1} + x_{1, 3} \leq n$. □

4.2 The Necessary Condition

We have the following theorem concerning the nonblocking condition under packing strategy for a $v(m, n, r)$ network.

**Theorem 2.** The necessary condition for a $v(m, n, r)$ network to be nonblocking under packing strategy is

$$m \geq \max_{p \in P} \{\max_{x \in \mathbb{Z}} c^T x\},$$

where, $c$, $x$ and $b$ are length $(2r - 1)$ vectors with

- $c = [1 \ 1 \ \ldots \ 1]^T$
- $x = [x_1 \ x_2 \ \ldots \ x_{2r-1}]^T$
- $x = [x_1 \ x_2 \ \ldots \ x_{2r} \ x_{2r+1} \ x_{2r+2} \ \ldots \ x_{2r+1} + 1]^T$
- $b = [n \ n \ \ldots \ n]^T$.

$A_p$ is a $(2r - 1) \times (2r - 1)$ matrix, $A_p x \leq b$ represents the $2r - 1$ constraints for the connection sequence $p$ described in Lemma 3, $P$ is the set of all connection sequences, that is, all possible permutations of $2r - 1$ connections $[(1, 2), (1, 3), \ldots, (1, r), (2, 1), (3, 1), \ldots, (r, 1)]$.

**Proof.** Apply the constraints derived in Lemmas 2 and 3 to the necessity part of Theorem 1. □

Theorem 2 translates the problem of finding $m$ that is necessary for a $v(m, n, r)$ network to be nonblocking under packing strategy to the problem of finding the maximum objective value among a set of linear programming problems. In the next two sections, we will show how to solve the set of linear programming problems shown in (5) in two steps:

Step 1. For a given connection sequence $p$ in a $v(m, n, r)$ network, find the closed form optimum solution of the corresponding set of linear programs.
Step 2. Find the maximum objective value among all the optimum values of the above linear programming problems for all possible connection sequences in a $v(m, n, r)$ network.

5 Solving the Linear Programming Problem

In this section, we find the optimum solution of the set of linear programs for a given connection sequence. There are numerous linear programming books, e.g., [12], [13], discussing standard methods of solving linear programming problems. In general, for an LP problem, there may not be a closed form optimum solution. However, as will be seen later, for this type of LP problem, by employing some special techniques, we can obtain a closed form optimum solution.

5.1 Preliminaries

Notice the following lemma that can simplify our linear programming problem.

**Lemma 4.** The linear programming problem

$$\max_{A_p x \leq b} c^T x,$$

where $A_p$, $c$, and $b = [n \, n \, \ldots \, n]^T$ are described in Theorem 2, can be transformed to the linear programming problem

$$\max_{A_p x' \leq b'} c^T x',$$

where $A_p$ and $c$ are the same and $b' = [1 \, 1 \, \ldots \, 1]^T$.

**Proof.** In fact, if the LP (7) has a solution $x' = [a_1 a_2 \ldots a_{2r-1}]^T$ and $c^T x' = \sum_{i=1}^{2r-1} a_i$, then the LP (6) has a solution $x = [a_1 n \, a_2 n \ldots a_{2r-1} n]^T$ and $c^T x = n \sum_{i=1}^{2r-1} a_i$. \hfill \Box

Lemma 4 allows us to simplify the constraints (1)-(4) by replacing $n$ on their right hand sides by 1 and 1 on their left hand sides (if any) by $\frac{1}{n}$. In the rest of the paper, we will use this normalized form and simply denote the LP (7) as

$$\max_{A_p x \leq b} c^T x,$$

where $A_p$ and $c$ are the same as in the LP (6), but $b = [1 \, 1 \, \ldots \, 1]^T$.

The following lemma is useful in solving our special LP problem.

**Lemma 5.** Given any $n \times n$ matrix $A$ and length $n$ vectors $b$ and $c$, if $x^* \geq 0$ is a solution to the system of linear equations $A x = b$ and $y^* \geq 0$ is a solution to the system of linear equations $y^T A = c^T$, $x^*$ is an optimum solution of the LP

$$\max_{A_p x \leq b} c^T x,$$

$y^*$ is an optimum solution of the dual LP

$$\min_{y^T b \leq c} b^T y,$$

and

$$c^T x^* = y^T b.$$

**Proof.** Since $x^*$ satisfies $x^* \geq 0$ and $A x^* \leq b$, $x^*$ is a feasible solution of the LP (9). Similarly, $y^*$ is a feasible solution of the dual LP (10). On the other hand, we have

$$c^T x^* = (y^T A)x^* = y^T(Ax^*) = y^T b.$$

Therefore, given any feasible solution $\bar{x}$ of the LP (9), since $y^* \geq 0$, we have

$$c^T \bar{x} = y^T A \bar{x} \leq y^T b = c^T x^*.$$

That is, $x^*$ is an optimum solution of the LP (9). Similarly, $y^*$ is an optimum solution of the dual LP (10). \hfill \Box

5.2 An Example for $r = 4$

Before we solve the general form of the LP (8), let’s consider an example for $r = 4$.

First, consider the connection sequence $p = (2, 1)(1, 2)(3, 1)(1, 3)(4, 1)(1, 4)$ shown in Fig. 3a. By Lemmas 3 and 4, we can obtain the following constraints (using the normalized form stated in Lemma 4)

$$x_{1,2} + x_{2,1} \leq 1,$$

$$x_{1,3} + x_{2,1} + x_{3,1} \leq 1,$$

$$x_{2,1} + x_{3,1} + x_{4,1} \leq 1,$$

$$x_{1,2} + x_{1,3} + x_{2,1} \leq 1,$$

$$x_{1,2} + x_{1,3} + x_{1,4} + \left(\frac{x_{1,1} + \frac{1}{n}}{n}\right) \leq 1,$$

$$x_{2,1} + x_{3,1} + x_{4,1} + \left(\frac{x_{1,1} + \frac{1}{n}}{n}\right) \leq 1,$$

$$x_{1,2}, x_{1,3}, x_{1,4}, x_{2,1}, x_{3,1}, x_{4,1}, x_{1,1} \geq 0.$$

Let $x_1 = x_{1,2}$, $x_2 = x_{1,3}$, $x_3 = x_{1,4}$, $x_4 = x_{2,1}$, $x_5 = x_{3,1}$, $x_6 = x_{4,1}$, and $x_7 = x_{1,1} + \frac{1}{n}$. The constraint matrix in the LP (8) is

$$A_p = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix}$$

It can be verified that the $A_p x = b$ has a unique solution

$$x_1 = \frac{8}{13}, x_2 = \frac{3}{13}, x_3 = \frac{1}{13}, x_4 = \frac{5}{13},$$

and that the $y^T A_p = c^T$ has a unique solution
By Theorem 2, to find the nonblocking condition for a $v(m, n, 4)$ network, we need to examine all possible permutations of connections $(1, 2), (1, 3), (1, 4), (2, 1), (3, 1),$ and $(4, 1)$. However, for this special LP (8), we may make use of the symmetry among connection sequences. For example, it is clear that if we rotate $(2, 1), (3, 1),$ and $(4, 1)$ in $p'$ to $(3, 1), (4, 1),$ and $(2, 1)$ and obtain a new connection sequence $p'' = (3, 1), (1, 2), (1, 3), (1, 4), (4, 1), (2, 1)$, then $p''$ has the same optimum value as that for $p'$. After considering all connection sequences, it turns out that $\frac{m}{n}$ is the maximum value among all optimum values of the LP (8). Thus, the nonblocking condition in Theorem 2 is $m \geq \frac{2n}{3} n$ for a $v(m, n, 4)$ network.

5.3 The General Form of Constraint Matrix $A_p$

To find the optimum solution of the LP (8) for any $r$ value, we need to first determine the general form of constraint matrix $A_p$ in the LP (8) for a connection sequence $p = (i_1, j_1)(i_2, j_2)\ldots(i_{r-2}, j_{r-2})$.

Based on the symmetry of connection sequences and for presentational convenience, without loss of generality, we assume that connections $(1, 2), (1, 3), \ldots, (1, r)$ keep their relative order in $p$ and connections $(2, 1), (3, 1), \ldots, (r, 1)$ also keep their relative order in $p$, but $(1, 2), (1, 3), \ldots, (1, r)$ may be interlaced with $(2, 1), (3, 1), \ldots, (r, 1)$. Also assume that $(2, 1)$ is the first connection in the connection sequence $p$, i.e., $(i_1, j_1) = (2, 1)$. The connection sequences

$p = (2, 1)(1, 2)(3, 1)(1, 3)(4, 1)(4, 1)$

and

$p' = (2, 1)(1, 2)(1, 3)(1, 4)(3, 1)(4, 1)$

in Fig. 3 are examples of such connection sequences.

Under the above assumptions, we list all constraints for connection sequence $p$ as follows: For a connection $(1, j)$, $2 \leq j \leq r$, if $(2, 1), (3, 1), \ldots, (k_1-1, 1), (k_1, 1)$ are all the $(*, 1)$s preceding $(1, j)$ (we know at least $(2, 1)$ is ahead of $(1, j)$), from (3) and Lemma 4, we have

$$x_{1, j} + \sum_{s=2}^{k_1} x_{s, 1} \leq 1.$$  \hspace{1cm} (11)

For a connection $(i, 1), 3 \leq i \leq r$, if $(1, 2), (1, 3), \ldots, (1, k_2-1), (1, k_2)$ are all the $(1, *)$s preceding $(i, 1)$, from (4) and Lemma 4, we have

$$\sum_{t=2}^{k_2} x_{1, t} + x_{i, 1} \leq 1.$$  \hspace{1cm} (12)

Clearly, if no $(1, *)$ precedes $(i, 1)$ in $p$, the constraint will be reduced to the trivial case $x_{1, 1} \leq 1$. The last two constraints are directly from (1), (2), and Lemma 4

$$\sum_{j=2}^{r} x_{1, j} + (x_{1, 1} + \frac{1}{n}) \leq 1$$ \hspace{1cm} (13)

$$\sum_{i=2}^{r} x_{i, 1} + (x_{1, 1} + \frac{1}{n}) \leq 1$$ \hspace{1cm} (14)
Thus, the \((2r - 1) \times (2r - 1)\) constraint matrix \(A_p\) can be written as
\[
A_p = \begin{bmatrix}
I & H & 0_{r-1} \\
L & S & e_{r-1} \\
0_{r-1}^T & 1_{r-1}^T & 1
\end{bmatrix}.
\]

The details and properties of matrix \(A_p\) are discussed in Appendix A.

Without loss of generality, in the following we will consider nontrivial connection sequences as defined in Appendix A.

5.4 The Solution of the LP (8)

In this subsection, we find the optimum solution of the LP (8). According to Lemma 5, we need to first solve two systems of linear equations \(A_p x = b\) and \(y^T A_p = c^T\).

The following theorem gives the main result of this section.

THEOREM 3. For a nontrivial connection sequence \(p\), the system of linear equations \(A_p x = b\) has a unique solution \(x^*\), which satisfies \(x^* \geq 0\) and
\[
\sum_{i=1}^{2r-1} x_{i}^* = 2 - \frac{1}{|\det(A_p)|}.
\]

PROOF. See Appendix B. \(\square\)

We can apply the same technique to the system of linear equations \(y^T A_p = c^T\) and obtain the following theorem.

THEOREM 4. For a nontrivial connection sequence \(p\), the system of linear equations \(y^T A_p = c^T\) has a unique solution \(y^*\), which satisfies \(y^* \geq 0\).

PROOF. See Appendix C. \(\square\)

By using Theorems 3 and 4, we can obtain the closed form optimum values of LP (8) and LP (6).

THEOREM 5. The optimum values of the LP (8) and the LP (6) are \((2 - \frac{1}{|\det(A_p)|})n\) and \((2 - \frac{1}{\max_{p \in P} |\det(A_p)|})n\), respectively.

PROOF. Applying Theorems 3 and 4 to Lemma 5, we have the optimum value of the LP (8) is
\[
\max_{A_p x = b} \quad c^T x = c^T x^* = \sum_{i=1}^{2r-1} x_{i}^* = 2 - \frac{1}{|\det(A_p)|}.
\]

Also, by Lemma 4, we have the optimum value of the LP (6) is
\[
\max_{x \geq 0} \quad c^T x = c^T x^* = \left(2 - \frac{1}{|\det(A_p)|}\right)n.
\]

\(\square\)

Now, let’s apply Theorem 5 to the connection sequences in the example of Section 5.2. For connection sequence \(p = (2, 1)(1, 2)(3, 1)(1, 3)(4, 1)(1, 4)\), we can easily verify that \(|\det(A_p)| = 13\). Therefore, the LP (6) for \(p\) has the optimum value \((2 - \frac{1}{13})n = \frac{22}{13}n\). For connection sequence \(p' = (2, 1)(1, 2)(1, 3)(1, 4)(3, 1)(4, 1)\), we can obtain that \(|\det(A_{p'})| = 10\). Thus, LP (6) for \(p'\) has the optimum value \((2 - \frac{1}{10})n = \frac{19}{10}n\).

6 Maximum Objective Value Among a Set of LP Problems

The maximum objective value among the set of LPs that correspond to all possible connection sequences can be written as
\[
\max_{p \in P} \max_{x \geq 0} c^T x = \max_{p \in P} \left\{ \left(2 - \frac{1}{|\det(A_p)|}\right)n \right\} = \left(2 - \frac{1}{\max_{p \in P} |\det(A_p)|}\right)n. \tag{15}
\]

Thus, the problem now is transformed to the problem of finding the maximum value among a set of determinants. The following theorem gives the maximum value of \(|\det(A_p)|\).

THEOREM 6.
\[
\max_{p \in P} |\det(A_p)| = F_{2r-1},
\]

where \(F_{2r-1}\) is the Fibonacci number.

PROOF. See Appendix D. \(\square\)

Finally, we can obtain the following closed form nonblocking condition for \(v(m, n, r)\) networks.

THEOREM 7. The necessary condition for a \(v(m, n, r)\) network to be nonblocking under packing strategy is
\[
m \geq \left(2 - \frac{1}{F_{2r-1}}\right)n,
\]

where \(F_{2r-1}\) is the Fibonacci number.

PROOF. From (15) and Theorem 6, we have that the maximum objective value among the set of LPs that correspond to all possible connection sequences
\[
\max_{p \in P} \max_{x \geq 0} c^T x = \left(2 - \frac{1}{F_{2r-1}}\right)n, \tag{16}
\]

where \(F_{2r-1}\) is the Fibonacci number. Also, note that \(x_{i,j}\)s in \(x\) are the numbers of middle stage switches in state \([i, j]\) and, therefore, should be integers and they should also satisfy constraints \(Ax \leq b\). Thus, from (16) and Theorem 2, we obtain the necessary nonblocking condition is
\[
m \geq \left(2 - \frac{1}{F_{2r-1}}\right)n. \tag{16}\]

From Theorem 7, we can see that, when \(r = 2\), we have \(m \geq \lceil \frac{6n}{7} \rceil\), which agrees with the wide-sense nonblocking condition obtained by Moore [2], when \(r = 3\), \(m \geq \lceil \frac{17n}{19} \rceil\) and when \(r = 4\), \(m \geq \lceil \frac{57n}{101} \rceil\), etc. In Table 1, we list the nonblocking \(m\) values under packing strategy for \(2 \leq r \leq 10\).
Note that, in Theorem 7 when \( n \leq F_{2r-1} \), we obtain \( m \geq 2n - 1 \), which agrees with the strictly nonblocking condition obtained by Clos [1]. We summarize it into the following corollary:

**COROLLARY 1.** For any \( v(m, n, r) \) network with \( n \leq F_{2r-1} \), where \( F_{2r-1} \) is the Fibonacci number, the necessary and sufficient condition for the network to be nonblocking under packing strategy is \( 2n - 1 \).

**PROOF.** When \( n \leq F_{2r-1} \), the necessary condition in Theorem 7 becomes

\[
m \geq \left( 2 - \frac{1}{F_{2r-1}} \right) n = \left[ 2n - \frac{n}{F_{2r-1}} \right] = 2n - 1,
\]

which matches the sufficient condition for strictly nonblocking \( v(m, n, r) \) networks [1]. □

It should be pointed out that, in practice, the two parameters of a \( v(m, n, r) \) network, \( n \) and \( r \), are generally in the same order, for example, \( n = r = \sqrt{N} \). Since \( F_{2r-1} \) is an exponential function in \( r \), the condition of Corollary 1 generally holds for a practical network. Corollary 1 indicates that, in terms of the number of middle stage switches required for nonblocking, packing strategy may perform better only in the case of \( n > F_{2r-1} \) (i.e., \( n \gg r \)) and has no advantage over random routing (strictly nonblocking) when \( n \leq F_{2r-1} \).

### 7 An Example for the Necessary Condition

In this section, we construct a connection sequence that actually reaches the necessary condition in Theorem 7.

Consider \( r = 4 \). Given a connection sequence \( p = (2, 1)(1, 2)(3, 1)(1, 3)(4, 1)(1, 4) \), we can obtain all constraints and matrix \( A_p \) as shown in Section 5.2. Then, by using the techniques discussed in Section 5, we can solve the LP (6) and obtain the optimum solution

\[
x = \begin{bmatrix} x_{1,2} & x_{1,3} & x_{1,4} & x_{2,1} & x_{3,1} & x_{4,1} & x_{1,1} + 1 \end{bmatrix}^T
\]

\[
= \begin{bmatrix} 8 & 3 & n & 5 & 5 & 2 & n \end{bmatrix}^T.
\]

Recall that \( x_{i,j} \) denotes the number of middle stage switches in state \([i,j]\). In the following, we will show how to actually reach this network state under packing strategy, that is, how to reach a network state with appropriate numbers of middle stage switches in the corresponding states.

We start with an empty network, that is, no connections in the network. We then generate a sequence of connection/disconnection requests and realize them under packing strategy, as shown in Table 2. The corresponding middle stage switch states are shown in Fig. 4. In particular, Fig. 4i shows the middle stage switch states immediately after Step \( i \). For presentation convenience, we name the middle stage switches in the same state a group.

Note that all steps in Table 2 follow packing strategy and, after Step 21, we have a total of

\[
\sum_{i=2}^{4} x_{i,1} + \sum_{j=2}^{4} x_{1,j} + (x_{1,1} + 1) = \frac{25}{13} n
\]

nonempty middle stage switches. On the other hand, from Table 1, we can see that, for \( r = 4 \), the Fibonacci number \( F_{2r-1} = F_7 = 13 \) and the nonblocking condition is

\[
m \geq \left( 2 - \frac{1}{F_{2r-1}} \right) n = \left( 2 - \frac{1}{13} \right) n = \frac{25}{13} n.
\]

### 8 Conclusions

In this paper, we have studied wide-sense nonblocking Clos networks, or \( v(m, n, r) \) networks, under packing strategy. We introduced a systematic approach to the analysis of wide-sense nonblocking conditions for the general \( v(m, n, r) \) networks with any \( r \) value. We first translated the problem of finding the nonblocking condition under packing strategy for \( v(m, n, r) \) networks to a set of linear programming problems. We then solved this special type of linear programming problems and obtained a closed form optimum solution. We have proven that the necessary condition for a \( v(m, n, r) \) network to be nonblocking under packing strategy is the number of middle stage switches

\[
m \geq \left( 2 - \frac{1}{F_{2r-1}} \right) n \text{ where } F_{2r-1} \text{ is the Fibonacci number.}
\]

When \( n \leq F_{2r-1} \), which generally holds for a practical network, this condition is also a sufficient nonblocking condition for packing strategy. We believe that the systematic approach developed in this paper can be used for analyzing other wide-sense nonblocking control strategies as well.

### Appendix

#### A: The General Form of Matrix \( A_p \)

In general, the \((2r-1) \times (2r-1)\) constraint matrix \( A_p \) has the following form:

<table>
<thead>
<tr>
<th>( r )</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>( F_{2r-1} )</td>
<td>2</td>
<td>5</td>
<td>13</td>
<td>34</td>
<td>89</td>
<td>233</td>
<td>610</td>
<td>1597</td>
<td>4181</td>
</tr>
<tr>
<td>( m )</td>
<td>( \frac{3}{2} n )</td>
<td>( \frac{9}{5} n )</td>
<td>( \frac{25}{13} n )</td>
<td>( \frac{67}{34} n )</td>
<td>( \frac{177}{89} n )</td>
<td>( \frac{465}{233} n )</td>
<td>( \frac{1219}{610} n )</td>
<td>( \frac{3193}{1597} n )</td>
<td>( \frac{8361}{4181} n )</td>
</tr>
</tbody>
</table>
### Table 2
The Sequence of Connection/Disconnection Operations for the Example

<table>
<thead>
<tr>
<th>Connection/disconnection operations:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Realize $x_{2,1}$ connections $(2, 1)$ in middle stage switch Group (A).</td>
<td></td>
</tr>
<tr>
<td>2. Realize $x_{2,1}$ connections $(3, 2)$ in middle stage switch Group (A).</td>
<td></td>
</tr>
<tr>
<td>3. Realize $x_{1,2}$ connections $(1, 2)$ in middle stage switch Group (B).</td>
<td></td>
</tr>
<tr>
<td>4. Realize $x_{1,2}$ connections $(3, 3)$ in middle stage switch Group (B).</td>
<td></td>
</tr>
<tr>
<td>5. Release the connections in Step 2.</td>
<td></td>
</tr>
<tr>
<td>6. Realize $x_{3,1}$ connections $(3, 1)$ in middle stage switch Group (C).</td>
<td></td>
</tr>
<tr>
<td>8. Realize $x_{2,1}$ connections $(3, 3)$. Note that in this case, there are several choices and we put them in middle stage switch Group (A).</td>
<td></td>
</tr>
<tr>
<td>9. Realize $x_{3,1}$ connections $(2, 3)$. There are several choices and we put them in middle stage switch Group (C).</td>
<td></td>
</tr>
<tr>
<td>10. Realize $x_{1,3}$ connections $(1, 3)$ in middle stage switch Group (D).</td>
<td></td>
</tr>
<tr>
<td>12. Realize $x_{1,2}$ connections $(4, 3)$ in middle stage switch Group (B).</td>
<td></td>
</tr>
<tr>
<td>13. Realize $x_{1,3}$ connections $(4, 2)$ in middle stage switch Group (D).</td>
<td></td>
</tr>
<tr>
<td>14. Realize $x_{4,1}$ connections $(4, 1)$ in middle stage switch Group (E).</td>
<td></td>
</tr>
<tr>
<td>16. Realize $x_{2,1}$ connections $(3, 4)$ in middle stage switch Group (A).</td>
<td></td>
</tr>
<tr>
<td>17. Realize $x_{3,1}$ connections $(2, 4)$ in middle stage switch Group (C).</td>
<td></td>
</tr>
<tr>
<td>18. Realize $x_{4,1}$ connections $(3, 4)$ in middle stage switch Group (E).</td>
<td></td>
</tr>
<tr>
<td>19. Realize $x_{4,4}$ connections $(1, 4)$ in middle stage switch Group (F).</td>
<td></td>
</tr>
<tr>
<td>21. Realize $x_{1,1} + 1$ connections $(1, 1)$ in middle stage switch Group (G).</td>
<td></td>
</tr>
</tbody>
</table>

\[
A_p = \begin{bmatrix}
I & H & 0_{r-1} \\
L & S & e_{r-1} \\
0^T & 1^T & 1
\end{bmatrix}.
\]

In the above matrix, $0_{r-1}$, $1_{r-1}$, and $e_{r-1}$ are length $r - 1$ vectors with

- $0_{r-1} = [0 \ 0 \ \ldots \ 0]^T$,
- $1_{r-1} = [1 \ 1 \ \ldots \ 1]^T$,
- $e_{r-1} = [0 \ 0 \ \ldots \ 1]^T$.

$I$, $S$, $L$, and $H$ are $(r - 1) \times (r - 1)$ matrices, where $I$ is an identity matrix.

\[
S = \begin{bmatrix}
0 & 1 & 0 & \ldots & 0 \\
0 & 0 & 1 & \ldots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \ldots & 1 \\
0 & 0 & 0 & \ldots & 0
\end{bmatrix}.
\]

$L$ and $H$ will be described below. It is easy to see that for a different connection sequences $p$, $A_p$ differs only in $L$ and $H$. We now examine matrices $L$ and $H$ in more detail.

Consider $(1, j_1)$ and $(1, j_2)$ in the connection sequence $p$, where $j_1 \leq j_2$. If $(2, 1), (3, 1), \ldots, (k_1, 1)$ are all the $(*, 1)'s$ preceding $(1, j_1)$ and $(2, 1), (3, 1), \ldots, (k_2, 1)$ are all the $(*, 1)'s$...
Fig. 4. The states of middle stage switches for connection sequence $p = (2, 1)(1, 2)(3, 1)(1, 3)(4, 1)(1, 4)$. 
Fig. 5. A staircase matrix.

preceding (1, j2), then we must have k1 ≤ k2. That is, H (as well as L by a similar argument) is a staircase matrix with 1s in the bottom-left part and 0s in the top-right part, as shown in Fig. 5.

Precisely, we have $L = [L_1^T L_2^T \ldots L_{r-1}^T]^T$, where, $L_1, L_2, \ldots, L_{r-1}$ are all length $(r - 1)$ row vectors of the form $[1 1 1 0 0 \ldots 0]$ with $l_1, l_2, \ldots, l_{r-1}$ consecutive 1s on the left, respectively, and $0 ≤ l_1 ≤ l_2 ≤ \cdots ≤ l_{r-1} = r - 1$. Note that $l_{r-1} = r - 1$ is derived from the constraint (13).

Similarly, $H = [H_1 H_2 \ldots H_{r-1}]$, where $H_1, H_2, \ldots, H_{r-1}$ are all length $(r - 1)$ column vectors of the form $[0 0 \ldots 0 1 1 \ldots 1]^T$ with $h_1, h_2, \ldots, h_{r-1}$ consecutive 1’s at the bottom, respectively, and $r - 1 = h_1 ≥ h_2 ≥ \cdots ≥ h_{r-1} ≥ 0$. Note that $h_1 = r - 1$ is derived from the assumption that (2, 1) appears first in the connection sequence $p$ and by the constraint (13).

Moreover, $L$ and $H$ are closely related. Consider $(i, 1)$ and $(1, j)$ in the connection sequence $p$, where $3 ≤ i ≤ r$ and $2 ≤ j ≤ r$. Either $(i, 1)$ precedes $(1, j)$ or $(1, j)$ precedes $(i, 1)$. If $(i, 1)$ precedes $(1, j)$, then $x_{1,1}$ will appear in the constraint (11) corresponding to this $(1, j)$ (see Section 5.3), but $x_{1,i}$ will not appear in the constraint (12) corresponding to this $(i, 1)$. On the other hand, if $(1, j)$ precedes $(i, 1)$, then $x_{1,j}$ will appear in the constraint (12) corresponding to this $(i, 1)$, but $x_{1,1}$ will not appear in the constraint (11) corresponding to this $(1, j)$.

Therefore, matrices $L$ and $H$ have the following forms:

$$L = \begin{bmatrix}
  t_{1,1} & t_{1,2} & \cdots & t_{1,r-1} \\
  t_{2,1} & t_{2,2} & \cdots & t_{2,r-1} \\
  \vdots & \vdots & \ddots & \vdots \\
  t_{r-2,1} & t_{r-2,2} & \cdots & t_{r-2,r-1} \\
  1 & 1 & \cdots & 1 \\
\end{bmatrix}$$

$$H = \begin{bmatrix}
  1 & \bar{t}_{1,1} & \bar{t}_{1,2} & \cdots & \bar{t}_{1,r-2} \\
  1 & \bar{t}_{2,1} & \bar{t}_{2,2} & \cdots & \bar{t}_{2,r-2} \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & \bar{t}_{r-2,1} & \bar{t}_{r-2,2} & \cdots & \bar{t}_{r-2,r-1} \\
  \end{bmatrix}$$

where $t_{i,j}$ and $\bar{t}_{i,j} \in \{0, 1\}$, and $\bar{t}_{i,j}$ is the complement of $t_{i,j}$.

Also, by the assumptions above, we have

$$t_{i,j} ≥ t_{i-1,j}$$

and

That is, we can establish the following relations:

$$h_{i+1} + l_i = r - 1, \ i = 1, 2, \ldots, r - 2,$$

and have

$$L \cdot H = \begin{bmatrix}
  l_1 & 0 & 0 & \cdots & 0 \\
  l_2 & l_2 - l_1 & 0 & \cdots & 0 \\
  l_3 & l_3 - l_1 & l_3 - l_2 & \cdots & 0 \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  l_{r-2} & l_{r-2} - l_1 & l_{r-2} - l_2 & \cdots & l_{r-2} - l_{r-3} \\
  l_{r-1} & l_{r-1} - l_1 & l_{r-1} - l_2 & \cdots & l_{r-1} - l_{r-2} \\
\end{bmatrix}. \quad (17)$$

Note that $l_1, l_2, \ldots, l_{r-1}$ are integers and satisfy

$$0 ≤ l_1 ≤ l_2 ≤ \cdots ≤ l_{r-1} = r - 1.$$

In fact, we can show that the above condition can be reduced to

$$1 ≤ l_1 ≤ l_2 ≤ \cdots ≤ l_{r-1} = r - 1.$$

Consider a connection sequence $p$ for which $l_1 = l_2 = \cdots = l_{k-1} = 0$ and $l_k ≥ 1$, that is, the corresponding matrices $L$ and $H$ are

$$L = \begin{bmatrix}
  0 & 0 & \cdots & 0 \\
  0 & 0 & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & t_{r-2} & \cdots & t_{r-1} \\
  1 & 1 & \cdots & 1 \\
\end{bmatrix} \quad (18)$$

$$H = \begin{bmatrix}
  1 & 1 & \cdots & 0 \\
  1 & 1 & \cdots & \bar{t}_{r-2} \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & 1 & \cdots & \bar{t}_{r-2} \\
\end{bmatrix}$$

Now, we construct another connection sequence $p'$ for which $l_1 ≥ 1$, that is, the corresponding matrices $L$ and $H$ are

$$L = \begin{bmatrix}
  1 & 0 & \cdots & 0 \\
  0 & 0 & \cdots & 0 \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & t_{r-2} & \cdots & t_{r-1} \\
  1 & 1 & \cdots & 1 \\
\end{bmatrix} \quad (19)$$

$$H = \begin{bmatrix}
  1 & 0 & \cdots & 0 \\
  1 & 1 & \cdots & \bar{t}_{r-2} \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & 1 & \cdots & \bar{t}_{r-2} \\
\end{bmatrix}$$
Notice that only \( k \) constraints are different between \( p \) and \( p' \).
For \( p \) we have
\[
x_{1,2} + x_{2,1} + x_{3,1} + \cdots + x_{k,1} + x_{k+1,1} \leq 1
\]
which corresponds to the matrix \( H \) in (18), and
\[
x_{i,1} \leq 1, \quad i = 3, 4, \ldots, k + 1
\]
which correspond to the matrix \( L \) in (18). Note that constraints (21) can be derived from constraint (20) and, therefore, can be discarded. As for \( p' \) we have
\[
x_{1,2} + x_{2,1} \leq 1
\]
which corresponds to the matrix \( H \) in (19), and
\[
x_{1,2} + x_{i,1} \leq 1, \quad i = 3, 4, \ldots, k + 1
\]
which correspond to the matrix \( L \) in (19).
Since any \( x \) vector that satisfies constraint (20) also satisfies constraints (22) and (23), the feasible solution space corresponding to \( p \) is contained by the feasible solution space corresponding to \( p' \). Therefore, it is not necessary to consider connection sequence \( p' \) with \( l_1 = 1 \). In the rest of the paper, we will always assume \( l_1 \geq 1 \), and we refer to a connection sequence with all \( l_i \geq 1 \) as a nontrivial connection sequence.

**B: Proof of Theorem 3**

Before we formally prove Theorem 3, we need to simplify the system \( A_p x = b \).

We solve the system \( A_p x = b \) by Gaussian elimination [12] and eliminate \( x_{2r-1} \) first. After we eliminate \( x_{2r-1} \), the system \( A_p x = b \) is equivalent to

\[
\begin{bmatrix}
I & H \\
L & \Lambda
\end{bmatrix}
\begin{bmatrix}
X_1 \\
X_2
\end{bmatrix}
= \begin{bmatrix}
1 \\
1_0
\end{bmatrix}
\]

and
\[
1 - x_{2r-1} = \sum_{i=1}^{r-1} x_i = \sum_{i=r}^{2r-2} x_i,
\]

where
\[
\Lambda = \begin{bmatrix}
0 & 1 & 0 & \cdots & 0 \\
0 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \cdots & 1 \\
-1 & -1 & -1 & \cdots & -1
\end{bmatrix},

X_1 = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{r-1}
\end{bmatrix},

X_2 = \begin{bmatrix}
x_r \\
x_{r+1} \\
\vdots \\
x_{2r-2}
\end{bmatrix},

1 = \begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix},

1_0 = \begin{bmatrix}
1 \\
0 \\
\vdots \\
0
\end{bmatrix}.

The following lemma simplifies the system \( A_p x = b \).

**Lemma 6.** The system of linear equations (24) is equivalent to

\[
(L \cdot H - \Lambda) X_2 = L \cdot 1 - 1_0.
\]

and

\[
X_1 = 1 - H \cdot X_2
\]

**Proof.** It can be easily shown by expanding the system (24).

It is straightforward to verify that

\[
|\det(A_p)| = |\det\begin{bmatrix}
I & H \\
L & \Lambda
\end{bmatrix}| = |\det(L \cdot H - \Lambda)|
\]

and, by (17), the system (26) becomes

\[
\begin{bmatrix}
l_1 & -1 & 0 & \cdots & 0 \\
l_2 & l_2 - l_1 & -1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
l_{r-1} & l_{r-2} - l_1 & l_{r-2} - l_2 & \cdots & -1 \\
l_{r-1} & l_{r-2} - l_1 & l_{r-2} - l_2 & \cdots & l_{r-1} - l_{r-2}
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{r-1} \\
x_{r-1}
\end{bmatrix}
= \begin{bmatrix}
l_1 - 1 \\
l_2 - 1 \\
\vdots \\
l_{r-2} - 1 \\
l_{r-2} - 1
\end{bmatrix}
\]

where \( l'_{r-1} = l_{r-1} + 1 \) and

\[
1 \leq l_1 \leq l_2 \leq \cdots \leq l_{r-2} \leq l_{r-1} = r - 1.
\]

The following theorem gives a solution to a system of a more general form than the system in (29):

**Theorem 8.** Let

\[
A = \begin{bmatrix}
a_{11} & -1 & 0 & \cdots & 0 \\
a_{21} & a_{22} & -1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
a_{k-1,1} & a_{k-1,2} & \cdots & a_{k-1,k-1} & -1 \\
a_{k,1} & a_{k,2} & \cdots & a_{k,k-1} & a_{k,k}
\end{bmatrix},

X = \begin{bmatrix}
x_1 \\
x_2 \\
\vdots \\
x_{k-1} \\
x_k
\end{bmatrix},

\]

which satisfy

1. \( a_{ij} - a_{i,i} = a_{k,i+1} \) for \( 1 \leq j \leq i' \leq i \leq k \),
2. \( f(i) \), the determinant of the upper left \((i \times i)\) submatrix of \( A \) (in particular, \( f(k) = \det(A) \)) is nonzero for \( 1 \leq i \leq k \).

Then, the system \( AX = d \) has a unique solution \( X^* \) and

\[
1^T \cdot X^* = \sum_{i=1}^{k} x_i^* = 1 - \frac{1}{\det(A)},
\]

especially if \( a_{11} \geq 1 \) and \( a_{ij} \geq 0 \) for \( 1 \leq j \leq i \leq k \), we have \( f(i) > 0 \) for \( 1 \leq i \leq k \) and \( X^* \geq 0 \).

We now see how to solve the system \( AX = d \) in Theorem 8.

Consider the determinant of the upper left \((i \times i)\) submatrix of \( A \), \( f(i) \). We have
\begin{align*}
f(1) &= |a_{11}| = a_{11}, \quad f(2) = \begin{vmatrix}
a_{11} & -1 \\
a_{21} & a_{22}
\end{vmatrix}, \\
f(3) &= \begin{vmatrix}
a_{11} & -1 & 0 \\
a_{21} & a_{22} & -1 \\
a_{31} & a_{32} & a_{33}
\end{vmatrix}, \ldots, \quad f(k) = \det(A).
\end{align*}

In addition, we denote \( f(0) = 1 \).

The following lemma gives a recurrence for \( f(i) \):

**LEMMA 7.**

\( f(i) = \sum_{j=1}^{i} a_{i,j} f(j-1) \). \hfill (30)

**PROOF.** Without loss of generality, we consider \( f(k) = \det(A) \). Expanding on the last row of \( \det(A) \), we have

\[ f(k) = \det(A) = \sum_{j=1}^{k} (-1)^{k+j} a_{k,j} \det(A_{k,j}), \hfill (31) \]

where \( A_{k,j} \) arises from \( A \) by removing the \( k \)th row and the \( j \)th column. Note that

\[
\det(A_{k,1}) = \begin{vmatrix}
-1 & 0 & \cdots & 0 & 0 \\
a_{22} & -1 & \cdots & 0 & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
a_{k-2,2} & a_{k-2,3} & \cdots & -1 & 0 \\
a_{k-1,2} & a_{k-1,3} & \cdots & a_{k-1,k-1} & -1
\end{vmatrix} = (-1)^{k-1} = (-1)^{k-1} f(0)
\]

\[
\det(A_{k,k}) = \begin{vmatrix}
a_{11} & -1 & 0 & \cdots & 0 & 0 \\
a_{21} & a_{22} & -1 & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
a_{k-2,1} & a_{k-2,2} & a_{k-2,3} & \cdots & a_{k-2,k-2} & -1 \\
a_{k-1,1} & a_{k-1,2} & a_{k-1,3} & \cdots & a_{k-1,k-1} & a_{k-1,k-1}
\end{vmatrix} = f(k-1)
\]

and for \( 2 \leq j \leq k-1 \), \( A_{k,j} \) has the form \( \begin{bmatrix} D_1 & 0 \\ D_2 & D_3 \end{bmatrix} \),

where \( D_1 \), \( D_2 \) and \( D_3 \) are \( (j-1) \times (j-1) \), \( (k-j) \times (j-1) \) and \( (k-j) \times (k-j) \) matrices, respectively, with

\[ \det(D_1) = f(j-1) \] and \( \det(D_3) = (-1)^{k-j} \). That is,

\[ \det(A_{k,j}) = \det(D_1) \cdot \det(D_3) = (-1)^{k-j} \cdot f(j-1). \]

Plugging the value of \( \det(A_{k,j}) \) into \( (31) \) gives the recurrence \( (30) \). \hfill \Box

We will simplify the system \( AX = d \) by Gaussian elimination and transform \( A \) to an upper triangular matrix. For presentational convenience, we introduce a new function

\[ g(i, j) = \sum_{l=1}^{j} a_{i,l} f(l-1), \quad 1 \leq j \leq i \leq k. \]

We can obtain some useful relations between \( f(i) \) and \( g(i, j) \).

**LEMMA 8.**

\[ f(i) - f(i-1) = a_{i,j} \sum_{j=1}^{i} f(j-1) \hfill (32) \]

\[ g(i, j) = a_{i,j} f(j-1) + g(i, j-1), \hfill (33) \]

where \( 1 \leq j \leq i \leq k \),

\[ g(i, j) = f(j) + a_{i,j+1} \sum_{l=1}^{j} f(l-1), \hfill (34) \]

where \( 1 \leq j < i \leq k \).

**PROOF.** Since condition 1 in Theorem 8 yields \( a_{i,j} - a_{i-1,j} = a_{i,i} \), we have

\[ f(i) - f(i-1) = \sum_{j=1}^{i} (a_{i,j} - a_{i-1,j}) f(j-1) + a_{i,i} f(i-1) \]

\[ = a_{i,i} \sum_{j=1}^{i} f(j-1), \]

that is, \( (32) \) holds. Directly from the definition of \( g(i, j) \), we can obtain \( (33) \). To prove \( (34) \), we use condition 1 in Theorem 8 again as follows:

\[ g(i, j) - f(j) = g(i, j) - g(j, j) = \sum_{l=1}^{j} (a_{i,l} - a_{j,l}) f(l-1) \]

\[ = a_{i,j+1} \sum_{l=1}^{j} f(l-1). \]

\hfill \Box

For a clearer presentation, in the rest of this paper, we will always assume that the blank entries of the matrices are all zeros.

**LEMMA 9.** The system \( AX = d \) in Theorem 8 can be transformed by Gaussian elimination to \( A \vec{x} = \vec{d} \) with

\[
\vec{A} = \begin{bmatrix}
\tilde{a}_{11} & -1 \\
\tilde{a}_{22} & -1 \\
\vdots & \ddots \\
\tilde{a}_{k-1,k-1} & -1 \\
\tilde{a}_{kk}
\end{bmatrix},
\]

\[
\vec{d} = \begin{bmatrix}
\tilde{d}_1 \\
\tilde{d}_2 \\
\vdots \\
\tilde{d}_{k-1} \\
\tilde{d}_k
\end{bmatrix},
\]

where \( \tilde{a}_{i,i} = \frac{f(i)}{f(i-1)} \) for \( 1 \leq i \leq k \), \( \tilde{d}_1 = a_{11} - 1 \), and \( \tilde{d}_i = \frac{a_{i,i}}{f(i-1)} \) for \( 2 \leq i \leq k \).
PROOF. In matrix form, we transform \( AX = d \) by Gaussian elimination as follows. Let

\[
Q_j = \begin{bmatrix}
1 & & & & \\
& 1 & & & \\
& & \ddots & & \\
& & & 1 & \\
& & & & \\
-\frac{g(j+1,j)}{f(j)} & & & & \\
-\frac{g(j+2,j)}{f(j)} & & & & \\
& \ddots & & & \\
& & \frac{g(k,j)}{f(j)} & & \\
& & & 1 & \\
\end{bmatrix}
\]

where \( 1 \leq j \leq k-1 \). Notice that \( f(1) = a_{11} \) and \( g(i,1) = a_{i,1} \). Then,

\[
Q_1 = \begin{bmatrix}
1 & & & & \\
& 1 & & & \\
& & \ddots & & \\
& & & 1 & \\
& & & & \\
-\frac{g(2,1)}{f(1)} & & & & \\
-\frac{g(3,1)}{f(1)} & & & & \\
& \ddots & & & \\
& & \frac{g(k-1,1)}{f(1)} & & \\
& & & 1 & \\
\end{bmatrix}
\begin{bmatrix}
1 & & & & \\
& 1 & & & \\
& & \ddots & & \\
& & & 1 & \\
& & & & \\
-\frac{g(2,1)}{f(1)} & & & & \\
-\frac{g(3,1)}{f(1)} & & & & \\
& \ddots & & & \\
& & \frac{g(k,1)}{f(1)} & & \\
& & & 1 & \\
\end{bmatrix}
\]

Multiplying both sides of \( AX = d \) by \( Q_1 \) on the left yields \( A^{[1]}X = d^{[1]} \) with

\[
A^{[1]} = Q_1 A
\]

\[
= \begin{bmatrix}
f(1) & -1 & & & \\
\frac{f(2)}{f(1)} & -1 & & & \\
\frac{f(3)}{f(1)} & a_{33} & -1 & & \\
\frac{f(4)}{f(1)} & a_{43} & a_{44} & -1 & \\
& \ddots & \ddots & \ddots & \ddots \\
\frac{f(k-1)}{f(1)} & a_{k-1,3} & a_{k-1,4} & \ldots & a_{k-1,k-1} & -1 \\
\frac{f(k)}{f(1)} & a_{k,3} & a_{k,4} & \ldots & a_{k,k-1} & a_{k,k} \\
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
\vdots \\
d_k \\
\end{bmatrix}
\]

\[
d^{[1]} = Q_1 d = \begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
\vdots \\
d_k \\
\end{bmatrix}
\]

where \( d_1 = a_{11} - 1 \), and \( d_i = \frac{a_i}{f(i)} \) for \( 2 \leq i \leq k \).

In fact, \( a_{i,1}^{[1]} = a_{11} = f(1) \), and for \( 2 \leq i \leq k \),

\[
a_{i,1}^{[1]} = -\frac{a_{i,1}}{a_{11}} a_{11} + a_{i,1} = 0
\]

\[
a_{i,2}^{[1]} = a_{i,1} + a_{i,2} = a_{i,1} + a_{i,2} f(1) = a_{i,2} \frac{f(1)}{f(i)} = g(i,2) f(1).
\]

In particular, \( a_{2,2}^{[1]} = \frac{a_{2,2}}{f(1)} = \frac{a_{2,2} f(1)}{f(i)} \). All other \( a_{i,j} \)'s remain unchanged. Also, for \( 2 \leq i \leq k \),

\[
d_i^{[1]} = -d_i a_{i,1} + d_i = \frac{(a_{i,1} - 1)a_{i,1} - (a_{i,1} - 1)a_{1,1}}{f(1)}.
\]

The rest of the Gaussian elimination steps are described below. In step \( j \) \((2 \leq j < k)\), given the system \( A^{[j-1]}X = d^{[j-1]} \) from step \( j - 1 \), multiplying both sides of \( A^{[j-1]}X = d^{[j-1]} \) by \( Q_j \) on the left gives \( A^{[j]}X = d^{[j]} \) with \( A^{[j]} = Q_j A^{[j-1]} \) and \( d^{[j]} = Q_j d^{[j-1]} \). Now, we claim that \( A^{[j]} \) and \( d^{[j]} \) have the following form:

\[
A^{[j]} = \begin{bmatrix}
f(1) & -1 & & & \\
\frac{f(2)}{f(1)} & -1 & & & \\
\frac{g(j+1,j)}{f(j)} & a_{j+1,j} & -1 & & \\
\frac{g(j+2,j)}{f(j)} & a_{j+2,j} & a_{j+2,j+1} & -1 & \\
& \ddots & \ddots & \ddots & \ddots \\
\frac{g(j,k,j)}{f(j)} & a_{j,k,j} & a_{j,k,j+1} & \ldots & a_{j,k,k-1} & -1 \\
\frac{g(j,k+1,j)}{f(j)} & a_{j,k+1,j} & a_{j,k+1,j+1} & \ldots & a_{j,k+1,k-1} & a_{j,k+1,k} & -1 \\
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
\vdots \\
d_j \\
\end{bmatrix}
\]

\[
d^{[j]} = \begin{bmatrix}
d_1 \\
d_2 \\
d_3 \\
\vdots \\
d_j \\
\end{bmatrix}
\]

where \( d_1 = a_{11} - 1 \), \( d_i^{[i-1]} = \frac{a_i}{f(i-1)} \) for \( 2 \leq i \leq j \), and \( d_i^{[j]} = \frac{a_i}{f(i)} \) for \( j + 1 \leq i < k \).

We prove the above claim by induction on \( j \). The claim holds for \( j = 1 \). Suppose this claim is true for \( j - 1 \), we will prove that it is also true for \( j \). In fact, for \( j + 1 \leq i \leq k \), \( A^{[j]} = Q_j A^{[j-1]} \) implies

\[
a_{i,j}^{[j]} = -\frac{g(i,j)}{f(j)} a_{j,j}^{[j-1]} + a_{i,j}^{[j-1]} = -\frac{g(i,j)}{f(j)} f(j) + g(i,j) f(j - 1) + \frac{g(i,j)}{f(j)} = 0
\]

and

\[
a_{i,j+1}^{[j]} = \frac{g(i,j)}{f(j)} a_{j+1,j}^{[j-1]} + \frac{g(i,j)}{f(j)} a_{j,j}^{[j-1]} + a_{i,j+1}^{[j-1]} = \frac{g(i,j)}{f(j)} a_{j+1,j}^{[j-1]} + \frac{g(i,j)}{f(j)} a_{j,j}^{[j-1]} + g(i,j) a_{i,j+1}
\]

(by using (33)). In particular, \( a_{i,j+1}^{[j+1]} = \frac{g(j+1,j+1)}{f(j+1)} = \frac{f(j+1)}{f(j)} \). The rest of \( a_{i,j}^{[j]} \) is the same as \( a_{i,j}^{[j-1]} \). That is, \( A^{[j]} \) for \( 1 \leq j < k \) has the form in (35).

Now, we prove that \( d^{[j]} \) also has the form in (35). Clearly, \( d_i^{[j]} \) for \( 1 \leq i \leq j \) is the same as \( d_i^{[i-1]} \), that is,
\[ d_i^{[i]} = d_i = a_{11} - 1 \quad \text{and} \quad d_i^{[i-1]} = \frac{a_{ij}}{f(i-1)} \quad \text{for} \quad 2 \leq i \leq j. \]

Now, for \( j + 1 \leq i \leq k, \)
\[
d_i^{[i]} = d_i^{[i-1]} - d_i^{[i-1]} g(i, j) / f(j) = \frac{a_{ij}}{f(j)} - \frac{a_{ij}}{f(j)} g(i, j) / f(j) = \frac{a_{ij} f(j) - a_{ij} g(i, j)}{f(j) - 1}. \]

Then, by (34),
\[
d_i^{[i]} = \frac{a_{ij} f(j) - a_{ij} g(i, j) + a_{ij+1} \sum_{l=1}^{i+1} f(l-1)}{f(j) - 1}. \]

Applying condition 1 in Theorem 8 and (32) gives
\[
d_i^{[i]} = \frac{a_{ij+1} f(j) - a_{ij+1} f(j) - f(j-1)}{f(j) - 1} = \frac{a_{ij+1} f(j) - a_{ij+1} f(j)}{f(j) - 1}. \]

Thus, \( d_i^{[i]} \) has the form in (35).

After step \( k-1 \), we obtain \( A^{[k-1]} X = d^{[k-1]} \) with \( A^{[k-1]} = \tilde{A} \) and \( d^{[k-1]} = \tilde{d} \).

We are now in the position to prove Theorem 3.

**PROOF OF THEOREM 3.** Note that the matrix in (29) has the form of \( A \) in Theorem 8. For notational convenience, we let \( l_0 = 0 \) and denote \( l_{r-1} = l_r \). Thus, we have \( a_{ij} = l_i - l_{i-1} \geq 0 \) for \( 1 \leq j \leq i \leq r - 1 \) and \( a_{11} = l_i - l_0 = l_i \geq 1 \). We first show that (29) or its equivalent (26) satisfies condition 1 in Theorem 8.

Since \( a_{ij} = l_i - l_{i-1} \) for \( 1 \leq j \leq i \leq r - 1 \), we have \( a_{ij} - a_{ij} = (l_i - l_{i-1}) - (l_{i-1} - l_{i-2}) = l_i - l_{i-1} = a_{ij+1} f(i) \) for \( 1 \leq j \leq i \leq r - 1 \). Thus, condition 1 in Theorem 8 holds. We prove condition 2 by induction on \( i \). First, we have \( f(i) = a_{11} = l_i > 0 \). Assume \( f(k) > 0 \) for all \( k \leq i - 1 \). Note that \( a_{i,1}, a_{i,2}, \ldots, a_{i,i} \geq 0 \) and at least one of them is greater than 0 (e.g., \( a_{i,1} = l_i - l_0 = l_i > 0 \). By using the recurrence (30), \( f(i) = \sum_{j=1}^{i+1} a_{ij} f(j-1) \), we have \( f(i) > 0 \). Then, condition 2 in Theorem 8 holds.

Applying Theorem 8 and (28) to (29), we have that the system (29) or (26) has a unique solution \( X_2^* \geq 0 \) and satisfies \( \sum_{i=1}^{2r-2} x_i^* = 1 - \frac{1}{| \det(A_p) |} \). By (25) and (27), we have
\[
\sum_{i=1}^{r-1} x_i^* = \sum_{i=r}^{2r-2} x_i^* = 1 - \frac{1}{| \det(A_p) |} \quad \text{and} \quad x_{2r-1}^* = \frac{1}{| \det(A_p) |} > 0.
\]

Therefore,
\[
\sum_{i=1}^{2r-1} x_i^* = 2 \left( 1 - \frac{1}{| \det(A_p) |} \right) + \frac{1}{| \det(A_p) |} = 2 - \frac{1}{| \det(A_p) |}. \]

The remainder is checking \( X_1^* \geq 0 \). In fact, since all elements in \( H \) are 0s or 1s, from (27) we have, for any \( i, 1 \leq i \leq r - 1, \)
\[
x_i^* \geq 1 - \sum_{j=0}^{r-2} x_j^* = \frac{1}{| \det(A_p) |} > 0.
\]

**C: PROOF OF THEOREM 4**

**PROOF.** Suppose we eliminate \( y_{2r-1} \) in the system \( y^T A_p = c^T \) first. After that, the system \( y^T A_p = c^T \) is equivalent to
Clearly, the system (40) has the form of the linear equations in Theorem 8. Now, we will show that it also satisfies the conditions in Theorem 8. Again, we let \( b_0 = 0 \) and denote \( l'_{r-1} \) as \( l_{r-1} \). We have \( a_{i,j} = l_{r-j} - l_{r-i-1} \geq 0 \) for \( 1 \leq j \leq i \leq r - 1 \). Condition 1 in Theorem 8 holds because 
\[
  a_{i,j} - a_{i,j} = (l_{r-j} - l_{r-i-1}) - (l_{r-j} - l_{r-i-1}) = \]
\[
  l_{r-j} - l_{r-i-1} = a_{i,j} \text{ for } 1 \leq j \leq i \leq r - 1.
\]
Also, note that \( a_{1,1} = l'_{r-1} - l_{r-1} = 1 + l_{r-1} - l_{r-1} \geq 1 \). By a similar argument as in the proof of Theorem 3, we conclude that all conditions of Theorem 8 are satisfied. Therefore, we have that the system (40) or its equivalent (38) has a unique solution \( Y^*_i \geq 0 \). From (37) and (39), we also get a unique \( y_{2r-1} \geq 0 \) and \( Y^*_i \geq 0 \). That proves the theorem.

\[\Box\]

**D: PROOF OF THEOREM 6**

By observing the relations in (28) and (29), the determinant we need to maximize

\[
|\det(A_p)|_{(2r-1) \times (2r-1)} = \]

\[
\begin{bmatrix}
  l_1 & l_2 & \cdots & l_{r-3} & l_{r-2} & l_{r-1} \\
  -1 & l_2 - l_1 & l_{r-3} - l_1 & l_{r-2} - l_1 & l_{r-1} - l_1 \\
  0 & -1 & \cdots & l_{r-3} - l_2 & l_{r-2} - l_2 & l_{r-1} - l_2 \\
  \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\
  0 & 0 & \cdots & 1 & l_{r-2} - l_{r-3} & l_{r-1} - l_{r-3} \\
  0 & 0 & \cdots & 0 & -1 & l_{r-1} - l_{r-2} \\
\end{bmatrix}
\]

where \( l_{r-1} = l_{r-1} + 1 = r \) and

\[
1 \leq l_1 \leq l_2 \leq \cdots \leq l_{r-2} \leq l_{r-1} = r - 1.
\]

By rearranging the equations and their variables, the system (38) can be rewritten as

\[
\begin{bmatrix}
  l_{r-1} - l_{r-2} & -1 & 0 & \cdots & 0 & 0 \\
  l_{r-1} - l_{r-3} & l_{r-2} - l_{r-3} & -1 & \cdots & 0 & 0 \\
  \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\
  l_{r-1} - l_{r-2} & l_{r-2} - l_{r-3} & \cdots & 1 & -1 & 0 \\
  l_{r-1} - l_{r-3} & l_{r-2} - l_{r-3} & \cdots & l_{r-3} - l_2 & l_{r-2} - l_2 & l_{r-1} - l_2 \\
  l_{r-1} - l_{r-2} & l_{r-2} - l_{r-3} & \cdots & l_{r-3} - l_1 & l_{r-2} - l_1 & l_{r-1} - l_1 \\
\end{bmatrix}
\]

where, \( \tilde{Y}_2 = [y_{2r-2} \ y_{2r-3} \ \cdots \ y_r]^T \).
PROOF. For notational convenience, let

\[
\Delta = \begin{vmatrix}
\delta_1 & -1 \\
-1 & \delta_2 & -1 \\
\vdots & \ddots & \ddots \\
& \ddots & \ddots & -1 \\
& & \ddots & \delta_{k-1} & -1 \\
& & & -1 & \delta_k \\
\end{vmatrix}
\]  

(42)

where \( \delta_i \), \( 1 \leq i \leq k \), is an integer not less than 2, and \( \sum_{i=1}^{k} \delta_i = 3k - 1 \).

The following theorem gives the maximum value of determinant \( \Delta \).

**THEOREM 9.** The optimum value for

\[
\max_{\sum_{i=1}^{k} \delta_i = c} \Delta,
\]

where \( \Delta \) is the determinant in (42) and \( Z^+ \), the set of all nonnegative integers, is achieved at \( (\delta_1, \delta_2, \ldots, \delta_{k-1}, \delta_k) = (2, 3, \ldots, 3, 3) \) or, symmetrically, \( (3, 3, \ldots, 3, 2) \).

**PROOF.** For notational convenience, let \( c = 3k - 1 \). We will find the maximum value of

\[
\max_{\sum_{i=1}^{k} \delta_i = c} \Delta.
\]

(43)

Consider \( \delta_i \in R^+ \) first. By using the method of Lagrange multipliers [14], the optimum value for (43) can be achieved as follows:

We introduce a function

\[
\Phi = \Delta + \lambda(c - \sum_{i=1}^{k} \delta_i)
\]

where \( \lambda \) is a new variable. Equating its partial derivatives with respect to \( \delta_i \) to zero gives

\[
\frac{\partial \Phi}{\partial \delta_i} = \frac{\partial \Delta}{\partial \delta_i} - \lambda = 0, \quad 1 \leq i \leq k.
\]

Notice that \( \Delta \) in (42) is the determinant of a tridiagonal matrix. It is easy to check that

\[
\frac{\partial \Delta}{\partial \delta_i} = \Delta_{i,j},
\]

where \( \Delta_{i,j} \) arises from \( \Delta \) by removing the \( i \)th row and the \( i \)th column. Therefore, the optimum value of (42) satisfies the following equations:

\[
\Delta_{1,1} = \Delta_{2,2} = \cdots = \Delta_{k,k}
\]

(44)

and

\[
\delta_1 + \delta_2 + \cdots + \delta_k = c.
\]

(45)

We have the following lemma regarding the values of \( \delta_i \)s.

**LEMMA 10.**

\[
\delta_1 = \delta_2 = \delta_3 = \cdots = \delta_{k-1} = \delta_1 + \frac{1}{\delta_1}
\]

satisfy (44), and

\[
\Delta_{1,1} = \Delta_{2,2} = \cdots = \Delta_{k,k} = \delta_1^{k-1}.
\]

(46)

To prove Lemma 10, we need the following lemma.

**LEMMA 11.** For \( j \geq 1 \), if \( \delta_1 = \delta \) and \( \delta_i = \delta + \frac{1}{\delta} \) for \( 1 \leq i \leq j \), then

\[
\begin{vmatrix}
\delta_1 & -1 \\
-1 & \delta_2 & -1 \\
\vdots & \ddots & \ddots \\
& \ddots & \ddots & -1 \\
& & \ddots & \delta_{j-1} & -1 \\
& & & -1 & \delta_j \\
\end{vmatrix}
= \delta^j.
\]

**PROOF.** By induction on \( j \geq 1 \).

For \( j = 1 \), \( \det(\delta_1) = \delta_1 = \delta \). For \( j = 2 \),

\[
\begin{vmatrix}
\delta_1 & -1 \\
-1 & \delta_2 \\
\end{vmatrix}
= \delta_1 \delta_2 - 1 = \delta(\delta + \frac{1}{\delta}) - 1 = \delta^2.
\]

Now, suppose the lemma holds for any integer between 1 and \( j \). Then, for \( j + 1 \), expanding on the last row, we obtain

\[
\begin{vmatrix}
\delta_1 & -1 \\
-1 & \delta_2 & -1 \\
\vdots & \ddots & \ddots \\
& \ddots & \ddots & -1 \\
& & \ddots & \delta_{j} & -1 \\
& & & -1 & \delta_{j+1} \\
\end{vmatrix}
= \delta^j.
\]

\[
\begin{vmatrix}
\delta_1 & -1 \\
-1 & \delta_2 & -1 \\
\vdots & \ddots & \ddots \\
& \ddots & \ddots & -1 \\
& & \ddots & \delta_{j-1} & -1 \\
\end{vmatrix}
= \delta^{j-1}.
\]

Therefore, the lemma holds for all integer \( j \geq 1 \). \( \square \)

**PROOF OF LEMMA 10.** By Lemma 11, we have

\[
\Delta_{1,1} = \delta_{k,k} = \begin{vmatrix}
\delta_1 & -1 \\
-1 & \delta_2 & -1 \\
\vdots & \ddots & \ddots \\
& \ddots & \ddots & -1 \\
& & \ddots & \delta_{k-2} & -1 \\
& & & -1 & \delta_{k-1} \\
\end{vmatrix}
= \delta^{k-1},
\]

and, for \( 2 \leq i \leq k - 1 \), we have
Let’s look at some examples. For the determinant problem. Plugging (46) into (45) gives

\[ Therefore, we have \]

that is,

\[ \hat{c} \]

Letting \( \hat{c} = 3k - 1 \), we have the solutions to the above quadratic equation

\[ \delta_1 = \frac{3k - 1 + \sqrt{5k^2 + 2k + 1}}{2k}, \quad \delta_1 = \frac{3k - 1 - \sqrt{5k^2 + 2k + 1}}{2k}. \]

The second solution is discarded because it obviously cannot make determinant \( \Delta \) achieve its maximum value. Therefore, we have

\[ \delta^*_1 = \frac{3k - 1 + \sqrt{5k^2 + 2k + 1}}{2k}. \quad (47) \]

Thus, when \( \delta_1 \in R^+ \) and for \( k \geq 2 \), the maximum value for (43) is achieved at

\[ \delta^* = (\delta^*_1, \delta^*_1, \ldots, \delta^*_1, \delta^*_1). \]

Let’s look at some examples. For \( k = 2 \), \( \delta^* = (\frac{5}{2}, \frac{5}{2}) \); for \( k = 3 \), \( \delta^* = (2.54, 2.93, 2.54) \); for \( k = 4 \), \( \delta^* = (2.55, 2.95, 2.95, 2.55) \), and, as \( k \to \infty \),

\[ \delta^* = \left( \frac{3 + \sqrt{5}}{2}, \ldots, \frac{3 + \sqrt{5}}{2} \right) \approx (2.62, 3, \ldots, 2.62). \]

For this particular problem, we are interested in an integer solution. From the above examples, we can observe that all components of \( \delta^* \) are very close to 3 with the first and the last components slightly smaller and all other components equal. Also, note that \( \delta^* \) must satisfy \( \sum_{i=1}^{k} \delta^*_i = 3k - 1 \). Thus, we can take \( \delta^* = (2, 3, \ldots, 3, 2) \). \( \Box \)

Now, we evaluate the \( k \times k \) determinant that achieves the optimum value in Theorem 9:

\[ f(k) = \begin{vmatrix} 2 & -1 & \ldots & -1 \\ -1 & 3 & \ldots & -1 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \ldots & 3 \end{vmatrix}. \]

It is easy to see that

\[ f(1) = 2, \]

and

\[ f(2) = 5. \]

Expanding on the last row of this determinant, we can establish the following recurrence

\[ f(k) = 3f(k - 1) - f(k - 2), \text{ for } k \geq 3. \]

It is interesting to note the relationship between \( f(k) \) and the Fibonacci numbers \( F_k \), [11]. The Fibonacci numbers are defined by the recurrence

\[ F_0 = 0, \quad F_1 = 1, \quad \text{and} \quad F_n = F_{n-1} + F_{n-2}, \text{ for } n \geq 2. \]

It can be shown that

\[ f(k) = F_{2k+1}, \text{ for } k \geq 1. \]

Then, we obtain that

\[ \max_{P \in \mathcal{P}} \{|\det(A_P)|\} = f(r - 1) = F_{2r-1}. \]

That proves Theorem 6. \( \Box \)

**Acknowledgments**

We would like to thank the anonymous referees of this article for their many insightful and constructive comments. Y. Yang was supported by the U.S. Army Research Office under Grant No. DAAH04-96-1-0234 and by the U.S. National Science Foundation under Grant Nos. OSR-9350540 and MIP-9522532.

**References**


Yuanyuan Yang received the BS and MS degrees in computer engineering from Tsinghua University, Beijing, China, in 1982 and 1984, respectively, and the MSE and PhD degrees in computer science from The Johns Hopkins University, Baltimore, Maryland, in 1988 and 1992, respectively. In August 1992, Dr. Yang joined the Computer Science Faculty at the University of Vermont, where she is currently an associate professor. Dr. Yang’s research interests include parallel and distributed computing and systems, high speed networks, optical networks, high performance computer architecture, and fault-tolerant computing. She has published extensively in archival journals and conference proceedings in these areas. Dr. Yang also holds two U.S. patents in the area of multicast communication. Her research has been mainly supported by the U.S. Army Research Office and the National Science Foundation. Dr. Yang is on the organizing committee of the Fifth IEEE International Symposium on High-Performance Computer Architecture in 1999 and has served on the program committee of several international conferences. Dr. Yang is a senior member of the IEEE and a member of the IEEE Computer Society, the IEEE Communication Society, and the ACM.

Jianchao Wang received the BS degree in computer engineering from Tsinghua University, Beijing, China, in 1982, and the MS and PhD degrees in computer science from Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 1985 and 1988, respectively. He is currently a principal member of technical staff at GTE Laboratories Inc. in Waltham, Massachusetts. Before he joined GTE Labs, Dr. Wang worked at the Institute of Computing Technology, Chinese Academy of Sciences, The Johns Hopkins University, and Legent Corporation in Marlboro, Massachusetts. Dr. Wang has received a number of excellence/achievement awards from GTE and Legent and has four U.S. patents granted or pending. Dr. Wang’s research interests include IP telephony, databases, programming languages, computer communication networks, computer algorithms, and fault-tolerant computing. Dr. Wang is a member of the IEEE Computer Society and the ACM.