UNIVERSITY OF MINNESOTA

This is to certify that I have examined this bound copy of a doctoral dissertation by

HUNG QUANG NGO

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

DING-ZHU DU

Name of Faculty Adviser

Signature of Faculty Adviser

Date

GRADUATE SCHOOL

Issues in Interconnection Networks

A DISSERTATION

SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL

OF THE UNIVERSITY OF MINNESOTA

BY

HUNG QUANG NGO

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

MAY 2001

ACKNOWLEDGMENTS

Most people would feel lucky having a positive turning point in their academic lives. I was fortunate enough to have more than a couple. Obviously, a few people have played crucial roles on making these twists: my parents, professor Ding-Zhu Du – my advisor in Computer Science, and professor Dennis Stanton – my advisor in Mathematics. Many other professors and friends have been of tremendous help during the last 5 years, whose names I shall try to mention later in this section.

It is impossible to describe precisely in words my appreciation to all these people. Instead, I shall attempt to tell the modest story of my life thus far, and how these people have greatly improved it.

The first turning point was right after my ninth grade. I was very satisfied being exempted from taking the all important entrance exam to the tenth grade. Seeing my potential in Mathematics, my mother suggested me to take the test to enter the only specialized Mathematics class of Ho Chi Minh city at Le Hong Phong high school. My score was just enough to enter. It turned out that those three high school years were very successful as I excel to be one of the best. More important than any thing else, this boosted up my confidence in Mathematics and established the love for Mathematics in me.

The second turning point was right after I got my Bachelor degree in Computer Engineering from Ho Chi Minh city University of Technology. While I was clue-less on what I would like to do and was capable of doing with my life, my parents suggested me to take a shot at graduate study in the United States of America. This was a great move.

I was doing Multimedia Networking in the first two and a half years of my graduate life under Prof. Jaideep Srivastava. I would like to thank him for accepting me to his group of fine people. I have learned a lot during this period, and made a few very good friends. Also during this time, I started to take Mathematics courses. Quite a few of them. I was extremely fortunate that my first ever Combinatorics course was taught by Prof. Dennis Stanton. His lectures were so inspiring that my passion for Mathematics, and Combinatorics in particular, seemed to have been awaken. This was the third turning point. Many thanks to Prof. Stanton, I gradually realized that Multimedia Networking, although certainly an interesting area, is not exactly what I want to focus on. I needed something more combinatorial in nature.

Prof. Ding-Zhu Du got in just when I needed him the most. The fourth turning point was at the end of the summer of 1998 when I was working as an intern for Microsoft. All of a sudden, Prof. Du sent me an email asking me if I would like to pursue a Ph.D. under him. It turned out that officially he was still my assigned advisor, who was supposed to sign a form when I passed the qualifying exam. I have never really done any paper work to pick an advisor on my own. This is possibly the only time I find paper work truly helpful. Being a very responsible "faked" advisor, he browsed my homepage and found out that I had some background in Mathematics, which is very important to do his kind of research. The rest is history. Under the superb supervision of Prof. Du, I finished this dissertation in two years, and also accomplished many other things that I could never have dreamed of. I am very grateful for his tremendous help and advice. He was much more than just an advisor to all of us, his students. We often had long discussions on many aspects of the academic life, besides the technical matters. His encouragements, patience and strong belief in my ability have certainly played a huge part on my little run.

I would also like to thank several computer theorists and mathematicians, from whom I have taken classes which directly helped my research: professors R. Janardan, V. Reiner, W. Richter, and D. White.

I would like to thank Prof. Janardan for many other useful discussions and suggestions concerning various topics. He was very helpful also during my job seeking. Thanks are due to Prof. Heimdahl for writing recommendation letters, for being in the defense committee, and for various sporadic chats; Prof. Park for writing recommendation letters and encouragements.

I could not have gone through these 5 years without a lot of good friends, academic and non-academic alike, alphabetically including Xiuzhen Cheng, Raja Harinath, Manch Hon, Ivaylo Ilinkin, Sunny Kim, Bing Lu, Son Ngo, Minh Nguyen, Luan Nguyen, Thanh Pham, Lu Ruan, Srihari, Imelda Sutedjo, Cuong Tran, Hung Tran, Viet Tran, Thanh Trinh, Srivatsan Varadarajan. All of the little discussions, chit-chats, going out, happy hours at big-10, studying in the coffee shops Expressó Expose and European Grind, rollerblading around the lakes, skiing, etc. have been extremely helpful for me to overcome even the most difficult time.

Many many thanks to all of you. This dissertation would not have been completed without you.

DEDICATION

To mom Kim-Dinh Chu and dad Thai-Kien Ngo

ABSTRACT

Interconnection networks play a very important role in Computer Science. In the small scale, processors and memories and other computing devices are connected via these networks. While in the larger scale, we use networks to interconnect workstations, routers, or clusters of workstations, ...

Research on interconnection networks can be classified roughly into several layers: (a) application, (b) physical/network/transport, and (c) topological. This dissertation concerns mostly with issues at the topological layer.

At the topological layer, we study the designs of good topologies for interconnection networks which satisfy certain criteria. Four main classes of interconnection network topologies include shared medium, direct, indirect, and hybrid networks. Given the enormous number of application areas, the required features for network topologies could be very different. On the other hand, there are several features that are desired by many applications, such as low diameter, low degree, large number of nodes, and large connectivity for direct topologies; and rearrangeability or nonblockingness for indirect topologies, etc.

We shall address several problems at the topological layer, concerning topologies from all four major classes. This type of study has led to many beautiful developments in both Theoretical Computer Science and Mathematics. I hope that this dissertation shall strengthen this point. We mention a few of the problems addressed in this dissertation as follows. One problem we shall address is the Beneŝ conjecture on the rearrangeability of Shuffle-Exchange networks, which has not been solved for almost 40 years. Another

problem is on the multirate rearrangeability of Clos networks, a key building block for larger indirect networks. The dissertation addresses a 12 year old conjecture concerning this problem. On both conjectures, we shall provide new improvements toward solving them. Another sample problem is about the connectivity of a very general class of direct network: the consecutive-d digraphs. We give an almost complete characterization of the connectivity of these graphs, which are generalizations of many well-known direct network topologies such as the de Bruijn digraphs, and the Kautz digraphs. Yet another problem relates directly to non-adaptive group testing, where this dissertation provides the first non-constant error tolerant design.

At the application layer, a set of services have been provided by lower layers. Our task is then to design protocols on top of these services to satisfy our application needs. At this layer, we propose to study how to devise a protocol to transmit continuous media streams (video, audio) so that given the same network loss, perceptually users get better stream quality.

The dissertation, in sum, presents new results on many interesting problems of interconnection network topologies. The main theme is to apply well-established discrete mathematics techniques to solve a class of practical problems. From our results, there are several wide open directions for further research.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we shall attempt to draw a general picture depicting the field of interconnection networks and how this thesis fits into the big picture. In particular, the chapter shows how the rest of the chapters relate to the area of interconnection networks and to each other.

Section 1.1 gives a somewhat informal definition of interconnection networks, their uses and examples. The section also describes the main theme of this thesis. Section 1.2 presents a classification of interconnection network topologies. In section 1.3, we shall briefly describe the types of research typically done in this area via the well-known layering approach. Section 1.4 focuses on the topological layer, where most of the problems addressed in this thesis belong. Section 1.5 outlines the rest of the thesis and also describes the way each of the following chapters is organized.

1.1 Interconnection Networks and this Dissertation

At the abstract level, interconnection networks are nothing but graphs, be it undirected or directed. In general, the set of nodes of these graphs could represent virtually anything from "people" in relationship graphs to "processors" and "memories" in parallel computers. From a more computer engineering point of view, interconnection networks are networks which interconnect a set of computing devices such as processors, memories, workstations or even sets (clusters) of workstations. The term "networks" used here could be thought of as "intelligent topologies", since a computer network is essentially an interconnection topology plus the algorithms controlling data passing through it.

Interconnection network is an enormous field, which holds a very important position in both theory and practice.

From a practical point of view, the performance of most digital systems today is limited not by their logic or memory, but by their communication or interconnection capability. Communication speed and delay are the main bottleneck in all kinds of systems. Memories are larger and processors are faster, and they are becoming cheaper than ever, but the interconnection bandwidths are still several order of magnitude lagging behind. In the Internet, for example, we spend most of our time waiting for data to arrive, not to be processed. The speed and delay of the network depends almost solely on its interconnection capability, large or small scale. For instance, the speed of light could be nearly reached in optical fiber data transfer, but this speed will not be of much use if we do not have interconnection networks for optical switches which could switch broadband data at the same pace.

From a theoretical view point, various areas of Theoretical Computer Science and Mathematics have benefited from and have been of great use in studying interconnection networks. A famous example is the celebrated result of Ajtai, Komlós and Szemerédi [2] about sorting n number in parallel in $O(\log n)$ time. The proof was based on the notions of concentrators, superconcentrators and expanders, which are special graphs arising from studying classical (telephony) switching networks. In fact, the entire area of switching networks intertwines closely with Graph Theory, Algebraic Graph Theory, Combinatorics, Probability and Circuit Complexity [124, 134].

Given the huge number of applications, the requirements for interconnection networks could be very diverse. However, as we shall see later each class of networks has several outstanding requirements which are desired by most if not all applications. Good examples include rearrangeability and nonblockingness for indirect networks; and low degree, large number of nodes, low density for direct networks.

The results presented in this dissertation fit nicely into the overall picture described above. The problems addressed by the dissertation are well-defined theoretical problems which come directly from practical needs. In particular, most results concern with common desireable features of interconnection networks which thus have more potential applications than just the original motivations. Moreover, there are several difficult outstanding conjectures discussed in the dissertation, strengthening the point that our problems are theoretically interesting and difficult.

Each of the problems under investigation is abstracted out from the original practical concern to become a general mathematical problem. There are many reasons behind the generalizations, although we always have to be alert of the danger of solving an extremely difficult theoretical problem which has no practical implication whatsoever. Firstly, as mostly the case in theory (the term "theory" refers to either Theoretical Computer Science or Mathematics), the more general, the easier as we are not constrained to special cases. Secondly, a general solution has potential applications to different areas. As we shall see later, most problems in this dissertation are of this nature.

The main theme of the dissertation is to apply well known mathematical techniques, combinatorial ones in particular, to solve practical problems. The solutions in turn feed more problems and techniques back to theory. In particular, all of the problems and solutions are wide open, in a lot of sense, for further research.

1.2 A Taxonomy of Interconnection Network Topologies

In this section, we give a classification of interconnection network topologies based on a paper by Ni [125]. In each category, we shall also try to give sample networks and actual systems which use them if applicable. The data comes from the book by Duato, Yalamanchili and Ni [62].

Interconnection network topologies could be divided into four main classes: (a) Shared-Medium; (b) Direct Networks (also called Router-Based or Point-to-Point); (c) Indirect Networks (also called Switch-Based or Switching Networks); and (d) Hybrid Networks. The following subsections elaborate a little more on each class. Before getting into more details, we provide here a list which gives a little road map to the topologies. For each type of network, we shall give several real systems which use the topology. The examples are put in parentheses next to the topology.

(a) Shared-Medium Networks

- Local Area Networks or LANs
  - Contention Bus (Ethernet)
  - Token Bus (Arcnet)
  - Token Ring (FDDI Ring, IBM Token Ring)
- Backplane Bus (Sun Gigaplane, DEC AlphaServer8X00, SGI PowerPath-2)

(b) Direct Networks (Router-Based or Point-to-Point)

- Strictly Orthogonal Topologies
  - Mesh: 2D-mesh (Intel Paragon), 3D-mesh (MIT J-Machine)

        * Torus (or k-ary n-cube): 1D unidirectional torus or ring (KSR first-level ring), 2D bidirectional torus (Intel/CMU iWarp), 3D bidirectional torus (Cray T3D, Cray T3E)
        * Hypercube (Intel iPSC, nCUBE)
    - Regular Topologies: de Bruijn graphs, Kautz graphs, consecutive-d digraphs, circulant graphs, Cayley graphs of certain groups, ...
    - Other Direct Topologies: trees, generic graphs (which connect routers on the Internet, e.g.)

(c) Indirect Networks (Switch-Based)

- Regular Topologies
    * Crossbar (Cray X/Y-MP, DEC GIGAswitch, Myrinet)
    * Multistage Interconnection Networks or MINs
        · Blocking Networks: unidirectional MIN (NEC Cenju-3, IBM RP3), bidirectional MIN (IBM SP, TMC CM-5, Meiko CS-2)
        · Nonblocking Networks: strictly nonblocking (Clos), wide-sense nonblocking, rearrangeably nonblocking (Beneŝ)
- Irregular Topologies (DEC Autonet, Myrinet, ServerNet)

(d) Hybrid Networks

- Multiple-Backplane Buses (Sun XDBus)
- Hierarchical Networks (Bridged LANs, KSR, the Internet)
    * Cluster-Based Networks (Stanford DASH, HP/Convex Exemplar)
- Other Hypergraph Topologies: Hyperbuses, Hypermeshes, ...

1.2.1 Shared-Medium Networks

The most simple interconnection topology allows all parties to share a common medium, such as copper wires, optical fibers, the air or a bundle of wires realized by printed lines on circuit boards. Figure 1.1 shows typical examples of shared-medium networks. On the left of the figure is a shared-medium bus, the right a shared-medium ring. The most well-known real-world networks which use shared-medium topologies are the *local area networks* (LANs), which include *Ethernet*, *Token Bus* and *Token Ring*. At a smaller scale, processors sharing memories could all be interconnected using a *backplane buses* which usually have 50 to 300 wires printed on circuit boards or by discrete (backplane) wiring.

A very important feature of shared-medium networks is that only one node can use the network at a time. Thus, an *arbitration mechanism* is needed to decide who *owes* the common medium at a given time. This mechanism is essentially the thing that distinguishes different LAN technologies.

In *contention buses* like Ethernets, all nodes compete for a cable which allows *collisions* to occur. Some probabilistic back-off strategy is enforced at each node until a node gains access to the cable. This is possible since each node can "sense" if the cable is in use or not. The throughput is about 35% on average, which is not every high. But Ethernets are very popular due to its simplicity. Another plausible mechanism is to use time division multiplexing (TDM), assigning a time slot to each node to avoid conflicts. This strategy is very ineffective if applied blindly, but it is often used in combination with other strategies to form some sort of hierarchical mechanism. Yet another mechanism is to periodically query for nodes wanting access to the medium, and then assign time slots to these nodes, minimizing the ineffectiveness of TDM yet eliminating the indeterministic nature of Aloha-type mechanisms. We shall come back to this strategy in a later chapter.

In *token buses* and *token rings*, a “token” (data packet) is passed from one node to another according to a predefined protocol. The node holding the token has access to the medium.

Much more detailed information on shared-medium networks can be found on any typical networking and parallel computing books, see [103, 149], for example.

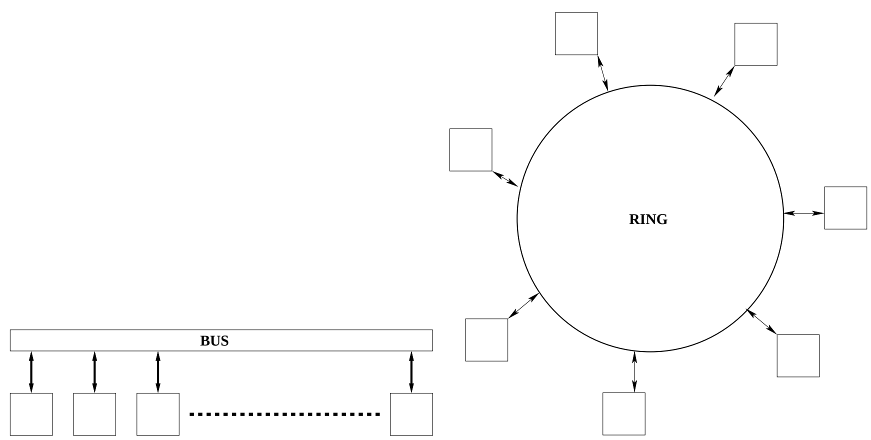


Figure 1.1: Shared-Medium LANs

1.2.2 Direct Networks

In general, routers in the Internet form a network where a pair of directly connected routers could communicate packets without going through other hops. This is a good example of a direct network. Formally, a direct network could be viewed as a graph (directed or undirected) whose nodes are programmable computers with their own memory, processor and other functional units. Figure 1.2 shows a generic node architecture for our direct networks. Although this architecture applies more to a node in multicomputers, not an Internet router, it is sufficient for our purposes. As we have mentioned, the processor and other units in a node makes the node intelligent, controlling the router for message routing.

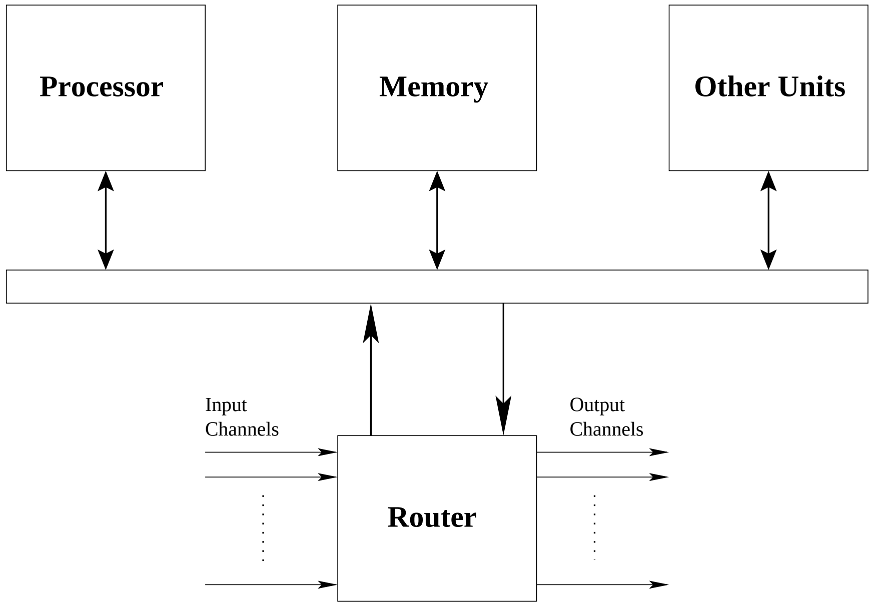


Figure 1.2: A generic node in direct networks

The router consists of a set of input channels and a set of output channels. Hence, we could view the whole network as a directed graph where each edge goes from an output channel of one node to an input channel of another node. When the channels are bidirectional, the graph becomes undirected.

The biggest advantage of direct networks over shared-medium networks is *scalability*. Adding more nodes into a shared-medium network decreases its throughput, which is not the case with direct networks. There are many other advantages as well, such as performance, incremental expandability, partitionability, ... Clearly, the superiority comes with a cost. Direct networks are considerably more complex to handle. Distributed routing, fair queueing, congestion handling are only a few issues.

Direct network topologies can be divided into three main classes: *strictly orthogonal*, *regular*, and *others*.

The definition of an orthogonal topology, as found in many engineering textbooks such as [62], is not mathematically rigorous. Consequently, we shall not attempt to give the definition here. Informally, orthogonal topologies consist of nodes which are members of $\mathbb{Z}_{k_1} \times \cdots \times \mathbb{Z}_{k_n}$, and each link can be arranged in such a way that it produces a displacement in a single dimension. The number n is called the dimension of the network. When $k_1 = \cdots = k_n = k$, the network is said to be k-ary. For example, the bidirectional k-ary n-cube has node set $\mathbb{Z}_k^n$, where there is a link between x and y iff x and y agrees in all coordinates but coordinate i, in which $y_i = (x_i \pm 1) \pmod k$. The hypercube (or n-cube) is nothing but a 2-ary n-cube. Meshes and tori are other examples of orthogonal topologies.

Regular topologies are regular (directed or undirected) graphs. Some well-known regular topologies include the Kautz graphs [96], de Bruijn graphs [44], their generalizations [17, 92], consecutive-d digraphs [57], ... Figure 1.3 shows the consecutive-d digraph $G(2, 10, 4, 7)$, whose precise definition shall be given in a later chapter.

A good theoretical resource on direct network topologies and related problems is [18]. The reader is also referred to [149] for routing and related issues. Direct networks are sometimes referred to as *point-to-point networks* or *router-based networks*.

1.2.3 Indirect Networks

Indirect networks, also called *switch-based networks* or *switching networks*, form an important class of interconnection networks. In this class of networks, the nodes are connected via switches instead of directly to each other. An indirect network can be thought of as a black box with m inputs and n outputs. In the classical sense (of telephone switching networks), the black box is configurable so that any one-to-one mapping between some subset of inputs and some equal-sized subset of output is *realizable*. The one-to-one mapping rep-

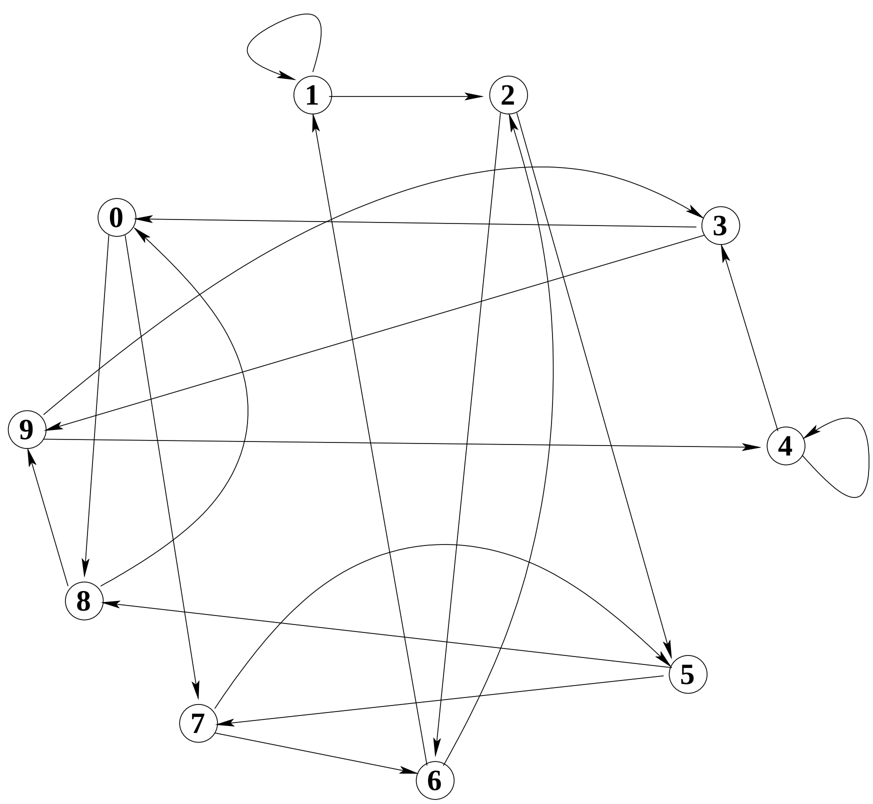
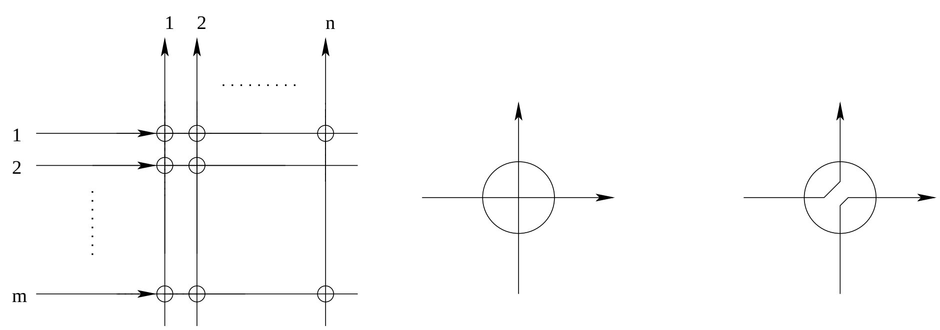


Figure 1.3: A direct network: the consecutive-d digraph $G(2, 10, 4, 7)$

resents simultaneous communication requests between inputs and outputs. This model is the *unicast* model, as opposed to the *multicast* model, where an input could request a set of outputs as long as no conflict arises. We shall see later that in the case of optical networks, more requirements could be imposed on the switching network, posing new challenging problems.

Before elaborating more on this class of networks, let us first look at the most basic components of switching networks: the crossbars and cross points as shown in figure 1.4. An $m \times n$ *crossbar* is denoted by X_{mn} with m inputs and n outputs, interconnected via



(a) Crossbar X_{mn} (b) A straight crosspoint (c) A bend crosspoint

Figure 1.4: Crossbar and crosspoint

mn cross-points as shown. Each crosspoint has two configurable states: the *cross state* and the *bend state*. By setting the states of mn cross-points correctly, any one-to-one mapping between inputs and outputs is realizable. Since there is a physical limitation on the number of cross-points, for large networks the crossbars are not practical. Hence, researchers have come up with switching network designs which implement a larger switch using smaller crossbars. Figure 1.5 shows one sample of such design. The figure shows a typical example of *multistage interconnection networks* (MINs), in which the crossbars are divided into stages with crossbars at a stage having identical dimension (like 3×2), and crossbars at a

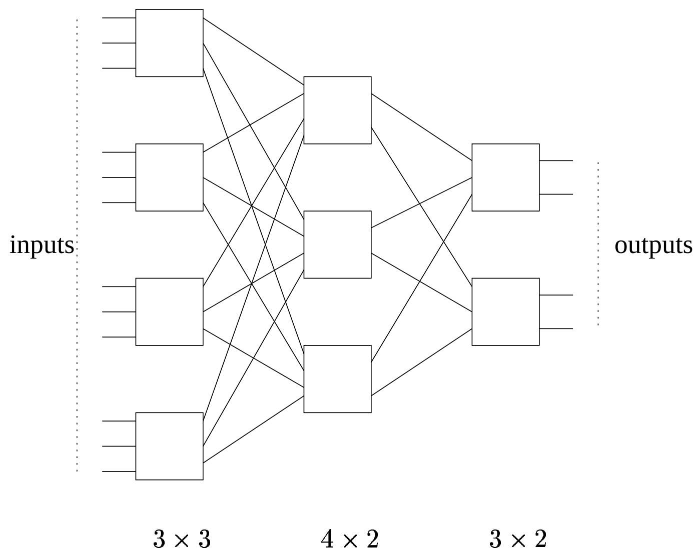


Figure 1.5: A 3-stage interconnection network

stage can only be connected to crossbars at two adjacent stages. Irregular MINs are hard to control, hence most MINs have regular connection pattern, such as the one shown in figure 1.6. We shall come back to this so-called Shuffle-Exchange networks in a later chapter.

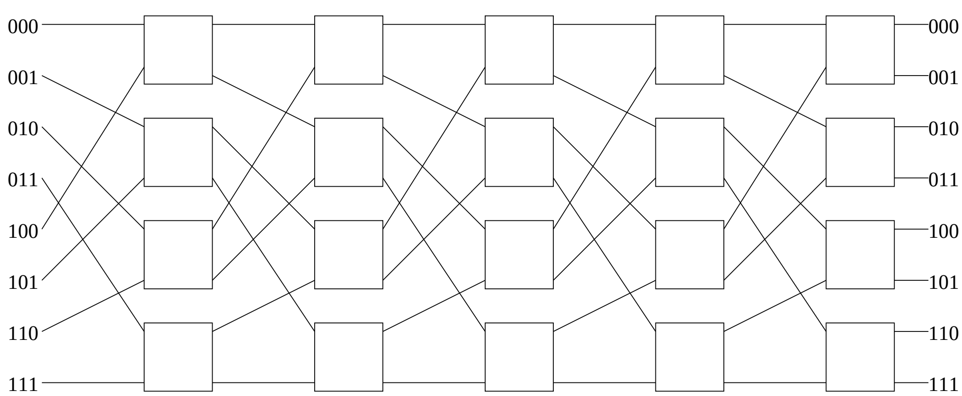


Figure 1.6: The 5-stage Shuffle-Exchange network $(SE_3)^5$

Switching networks, besides many other larger scale applications, are crucial in the designs of switches, be it telephony or optical. The classical reference on switching networks is Beneŝ [14]. More recent developments could be found in Hwang [90], Du and Hwang [60], Hui [85], and Hinton [79].

1.2.4 Hybrid Networks

Hybrid networks are networks whose topologies are combinations of several classes of topologies. A hybrid network is shown in figure 1.7. In this network, the nodes are parti-

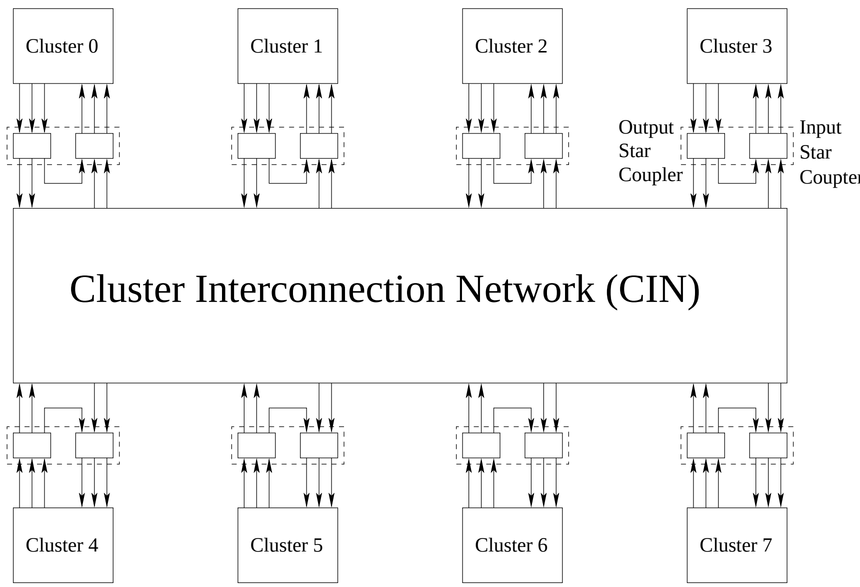


Figure 1.7: A clustered based optical interconnection network

tioned into clusters where nodes in each cluster are connected using the simple *star* topology. The stars, in turn, are interconnected via some direct topology called the *cluster interconnection network* (or CIN). The reason for this combination is clear: scalability. As the networks become larger, it is practically impossible to impose any particular topology to the nodes due to physical and geometrical constraints. It is much easier to implement some sort of hierarchical or clustered-based topology in the whole network. The Internet, for example, is a hybrid network where the LANs are connected using some shared-medium

	Shared-Medium	Direct	Indirect	Hybrid
Application				X
Transport/Network/Physical				
Topological	X	X	X	X

Table 1.1: A tentative subdivision of research on interconnection networks

topology and interfaced to the rest of the network via routers or switches which are interconnected via a direct topology. In fact, the routers and switches have their own cluster and hierarchy structure. Hyperbuses are another example of hybrid networks. Processors and memories sharing a huge common bus makes the bus a bottle neck. Instead, a small amount of processing elements could share a little bus, and the little buses are connected to a larger bus and so on and so forth. Clearly these topologies scale well with the cost of increasing the complexity of controlling the networks.

Several references on hybrid networks are in place. We mention only a few here: Bermond et al. [20], Duato et al. [62].

1.3 Research on Interconnection Networks

Studies on interconnection networks could be roughly classified into different layers as shown in table 1.3, where each cell in the table represents a rather large area of research. The columns in the table represent different network layers. For each type of the topologies, we could theoretically study the networks at each layer. Life would have been great if things are this simple. However, in reality, different technologies and their combinations make this table multidimensional. The layerings are different from one type of networking to another. The TCP/IP protocol stack, ATM protocol stack, optical network protocol stacks are only a few of them. Moreover, the rows and columns of the table are interdependent in a lot of cases. In a wide sense though, the three rows of the table are relatively independent, enough for this table to be meaningful. The X's in the table specify the places where the problems addressed in this dissertation belong to.

At the application layer, we assume that the lower layers provide a set of services to transmit data packets from one host to another. At this layer, there is no need to worry about checksums or routing tables. All we have to do is to ask the lower layers to send one packet from one host to another, given some minimal support for quality of service (QoS). For example, TCP services ensure that the packet actually arrives at the destination, but the delay time is theoretically unbounded; while UDP packets are to be sent without any guarantee. Since the QoS requirements for some applications could be very strict, such as multimedia applications, at the application layer we try to take advantage of the given services and also try to satisfy the application's QoS criteria. An example of this type of studies shall be presented in a later chapter.

The middle layer actually consists of many smaller sublayers. We shall not attempt to describe in details what type of studies there are on this layer, since the whole area is very large. Roughly speaking though, this layer concerns with packet/message routing, transmission at the bit level, physical and logical addressing of hosts, and so on, given some (dynamic) network topology. Clearly this is an extremely large layer, as there are so many different types of networks with diversely different physical and logical constraints. At the larger scale we have IP, ATM, and optical networks; while at the smaller scale we encounter multicomputer or multiprocessor systems.

At the topological layer, we study different topologies from a more theoretical point of view, addressing mostly abstract problems. More details on this layer shall be described in the next section. As can be seen from table 1.3, this dissertation deals mostly with topological issues.

1.4 The Topological Layer

In many theoretical disciplines, there are three main types of questions: (a) how good can we do a certain thing ? (b) how do we actually do it in the optimal sense ? and (c) how do we take advantage of this optimal solution ? Research at the topological layer of interconnection networks is no exception. There are three main classes of problems to be investigated. The first class concerns with determining how good such and such a network can be designed. Secondly, we need to actually design a network which is as close to optimal as possible. And lastly, generic routing algorithms are needed to take advantage of the optimal designs. This dissertation addresses mainly problems from the first and the second class. As we have mentioned and shall demonstrate later, this type of studies is very rewarding from both the theoretical and practical perspectives. It should be obvious that all these problems depend on the type of networks under consideration and the features we want to achieve. Moreover, we also need to keep an eye on various trade-offs imposed on the problem at hand, mostly due to practical reasons.

Given that the number of applications of interconnection networks is enormous, desired features could vary greatly from one application to another, and from one topology to the next. However, there are some common objectives for most applications:

- *Connectivity:* supports for large numbers of nodes and concurrent connections, supports for multicasting and broadcasting, ...

- *Performance:* high throughput, small delay, low error rate, low processing load at each node, adaptability to changing and unbalanced loads, efficient fault identification and recovery, ...
- *Structural features:* scalability, modularity, fault tolerant, ...
- *Simplicity/Cost effectiveness:* balance between complexity and cost.

More specifically, we describe briefly some typical problems for each class of topologies below. The problems are constrained by the desired features as mentioned above, and also by some other topologically dependent features.

In shared-medium networks, the main problem is to come up with an efficient *arbitration mechanism*, which basically helps determine which node has access to the common medium at a given time. Well-established protocols such as carrier-sense multiple access with collision detection (CDMA/CD) for contention bus, or frequency/time division multiple access (FDMA/TDMA) for wireless networks are good examples of solutions to this problem [98, 149].

For direct networks, most problems at the topological layer are graph theoretic problems. We often want networks with low diameter, low (average) node degree, small average distance, large number of nodes, high degree of node/link connectivity, ...

Indirect networks require a distinct set of features, including nonblockingness, multicast nonblockingness, multirate nonblockingness. There are three types of nonblocking networks: strictly nonblocking, rearrangeably nonblocking, and wide-sense nonblocking. We give here informal descriptions of these properties, leaving the formal definitions to the specific chapters where the property is of concern. An indirect network is *strictly nonblocking* iff for any existing valid configuration of the network which connects a set of inputs to a set of outputs, it is always possible to find a route from an idle input to an idle output without disturbing the existing routes. When only *rearrangeability* is required, we are free to rearrange the existing routes. In *wide-sense nonblocking* networks, we cannot disturb the existing routes, but the new request has to be routed according to certain algorithm. Similar description could be said about multicast nonblocking, in which case an input can request a set of outputs. Optical networks pose new challenging problems. As multiplexing technologies such as time division multiplexing (TDM) and wavelength division multiplexing (WDM) allow one link to carry multiple connections with different bandwidths. The three types of nonblocking networks now need to be generalized to the multirate case, where each request from an input to an output also specifies a bandwidth requirement. A link can carry multiple channels as long as the total channel bandwidths does not exceed the link's capacity.

Hybrid networks were proposed mainly for their scalability. However, these networks increase the overall complexity of the system. Hence, one of the main problems is to reduce this complexity in some specific sense which is application dependent.

Studies on interconnection networks at the topological layer are very theoretical in nature. More often than not, different mathematical areas provide elegant solutions to our problems. Combinatorics, Algebra, and Graph Theory, in particular, are extremely useful. On the other hand, these studies pose very interesting and challenging mathematical problems, which require us to develop new techniques in the corresponding mathematical area. Algebraic Graph Theory, for example, has been expanded when we try to construct good expander graphs, which originated from switching network problems [2, 4, 5, 124, 134]. The same could be said about the Probabilistic Method, Graph Theory, Linear Algebra, Boolean Circuit Complexity, Coding Theory, ...

	Shared-Medium	Indirect	Direct	Hybrid
Application				Chapter 9
Topological	Chapter 2	Chapters 3, 4, 5	Chapters 6, 7	Chapter 8
Corresponding Part	I	II	III	IV

Table 1.2: A road map to the rest of the chapters

1.5 Dissertation Outline

It should be noted again that all the problems in this dissertation, despite their practical roots, are often formulated at a much more theoretical level. Many problems involve outstanding and very interesting mathematical conjectures. The generality usually makes the problem easier to solve. Moreover, general problems have potential applications to different areas, as we shall see.

Except the last chapter, each of the remaining chapters of the dissertation has roughly four main parts: overview, preliminaries, main results, and discussions. The "overview" section defines the problem and gives an account of related works. The "preliminaries" section gives detailed formulation and define notations, terminologies, and formally quote related results. Our main results of each chapter are either presented in the "main results" section, or in several sections after the preliminaries. The "discussions" section summarizes the results and gives potential future investigations.

In view of the subdivision of studies on interconnection networks as shown in table 1.3, table 1.5 shows us where the rest of the chapters fit into the big picture. The main chapters of the dissertation is divided into four parts depending on the type of topologies addressed. Specifically, the following describes the topic of each chapter in more details.

Chapter 2 concerns a particular arbitration mechanism for share-medium networks. As it turns out, this mechanism could be formulated precisely as an non-adaptive combinatorial group testing problem, which is obviously much more general. The chapter went on to propose a new error tolerant non-adaptive pooling design, which is the first design to tolerate a non-constant number of errors. This is exactly what we have stressed several times: applications of general solutions to different areas. Non-adaptive group testings have direct applications to DNA library screening and data mining.

Chapter 3 addresses the famous Beneš conjecture. In the early 60's, Beneš conjectured that $2n-1$ stages are sufficient for the Shuffle-Exchange network with 2^n inputs and 2^n outputs to be rearrangeable. The problem can be formulated in many different ways, from group theoretic, graph theoretic to linear algebraic formulations. The conjecture is very difficult and has not been proved yet. People have tried to improve the bound on this minimum number of stages by a small amount after every decade or so. Each of the improvement requires a new technique. As often the case in Mathematics, some sort of combination of such techniques shall eventually helps prove the conjecture. This chapter improves the 12 year old bound further with a new technique, fitting to the line of research thus far.

Chapter 4 deals with another conjecture on the multirate rearrangeability of the symmetric Clos networks. The conjecture was made by Chung and Ross in 1991. The progress has also been slow. This chapter, like the previous one, improves the bounds for the function under investigation. This problem could be formulated as a generalized edge coloring problem for bipartite graphs, which somehow escaped the literature.

Chapter 5 links a conjecture of Hwang and Lin (1995) on the multicast rearrangeability and nonblockingness of the Clos network to several other conjectures, including one by

Paul Erdős. We shall make several generalizations of Erdős conjecture and proves some special cases.

Chapter 6 characterizes the connectivity of the consecutive-d digraphs. These are a very general class of directed topologies, which generalizes the well-known Kautz digraphs and de Bruijn digraphs. These digraphs have been proved to be the best in terms of diameter given the maximum degree and the number of nodes. Hence, it is useful to characterize their connectivity, a measure of fault-tolerance. It is shown that these graphs almost always have maximum possible connectivity. We also show how to modify these graphs to reach maximum connectivity in the cases where the maximum is not reached.

Chapter 7 fills the gap on a wrong proof of a conjecture about the maximum reliability of the cyclic consecutive-k-out-of-n: G system. This system represents a ring topology.

Chapter 8 is about a problem on the scalability of optical networks. The problem could be formulated as a generalized version of the vertex coloring problem. The chapter improves significantly several known bounds on this generalized chromatic number for the hypercube, using techniques from Coding Theory.

Chapter 9 is the only chapter at the application layer, proposing a new protocol for reducing consecutive losses in multimedia streaming. The heart of the solution is a purely combinatorial theorem. Experimentally, we also show that the new protocol improves significantly the perceptual quality of the transmitted stream.

Lastly, chapter 10 summarizes the dissertation and discusses future works in a broader context.

Part I

Shared Medium Networks

Chapter 2

Conflict Resolution Algorithms and Error-Tolerant Non-adaptive Group Testing

This chapter first introduces a class of arbitration mechanisms for multiaccess communications called conflict resolution algorithms. The algorithms are linked directly to group testing, non-adaptive group testing in particular. We then propose a new error-tolerant non-adaptive pooling design, which is the first in the literature to tolerate more than a constant number of errors (linear in this case).

2.1 Overview

2.1.1 Conflict Resolution Algorithms

Figure 2.1 shows a generic model for multiaccess channels (also called multiple access channels or multiple access communications), in which a set of users share a common channel for communication. The channel could be a single cable in contention buses or the air in wireless networks. Protocols such as CSMA/CD for contention buses, or FDMA/TDMA for wireless channels are well established technologies [149]. Although collisions do not occur with CSMA/CD once a user has seized the channel, they can still occur during the contention period. Clearly the collisions worsen the system performance, especially when the propagation delay is large and each data unit is small, which is exactly

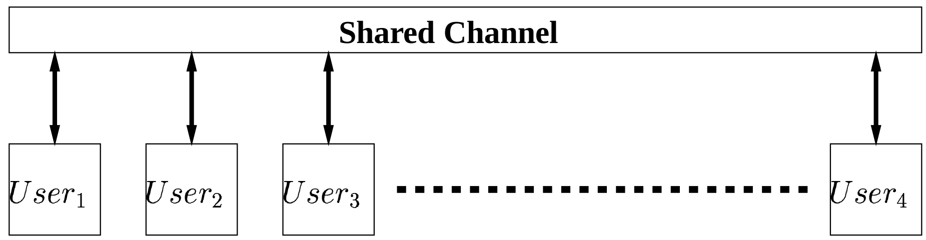


Figure 2.1: A generic model for multiaccess channels

the case with optical networks. Consequently, other methods which avoid collisions have been proposed. These algorithms are often named *Conflict Resolution Algorithms* (CRAs). The reader is referred to [121] for detailed descriptions and performance analyses of different CRAs. The first CRAs were first proposed almost simultaneously by several authors: Capetanakis [35–37], Tsybakov and Mikhailov [152], and Hayes [77]. Later, Berge et al. [16] and Wolf [165] noted that the idea is the same as that of group testing (see next subsection).

There are several models of multiple access communications in which non-adaptive group testing applies, one of which is as follows. (For another model, see Colbourn, Dinitz and Stinson [43].) Assume that data units have the same length and can be transmitted in one time slot. An arbitration method is to do the scheduling in epochs. At the start of an epoch, the users are to be classified as *active* or *inactive* as whether they have a message to send at that moment. An inactive user remains inactive even if the user generates some message during the epoch. Once all active users are known, we can arrange them to different time slots and no bandwidth is wasted. The epoch ends when all currently active users have used up their time slots. To determine the set of active users at the beginning of each epoch, some central device sends out *group queries*. Each query ask each member of the corresponding group of users to send a bit out if the user is active. If no user in the group is active, then the channel is silent. Otherwise, the device senses some signal. The queries could be *adaptive* in the sense that the second query could be dependent on the returned result of the first query. If the device is not as smart, a predetermined set of queries are sent out every time, in which case the algorithm is called *non-adaptive*. In some cases, there could be noises in the channel which makes the device mistaken a silent response with a signal response and vice versa. Consequently, it would be nice for the algorithm to be able to tolerate several errors. It is possible to assume that no more than some number of users want to transmit that the same time, as otherwise the channel sharing would be very ineffective in the first place. It shall be clear that this type of querying algorithms is nothing but non-adaptive error-tolerant group testing algorithms as described below.

2.1.2 Group Testing

The basic problem of group testing is to identify the set of defectives in a large population of items. As it is becoming more standard to use the term *positive* instead of *defective*, we shall use the former throughout the chapter. We assume some testing mechanism exists which if applied to an arbitrary subset of the population gives a *negative outcome* if the subset contains no positive and *positive outcome* otherwise. Objectives of group testing vary from minimizing the number of tests, limiting number of pools, limiting pool sizes to tolerating a few errors. It is conceivable that these objectives are often contradicting, thus testing strategies are application dependent.

Group testing algorithms can roughly be divided into two categories : *Combinatorial Group Testing* (CGT) and *Probabilistic Group Testing* (PGT). In CGT, it is often assumed that the number of positives among n items is equal to or at most d for some given positive integer d. In PGT, we fix some probability p of having a positive. Group testing strategies can also be either *adaptive* or *non-adaptive*. A group testing algorithm is non-adaptive if all tests must be specified without knowing the outcomes of other tests. A group testing algorithm is *error tolerant* if it can detect or correct some e errors in test outcomes. Test errors could be either $0 \to 1$, i.e. a negative pool is identified as positive, or $0 \to 1$ in the contrast.

In this chapter, we propose two new classes of non-adaptive and error-tolerant CGT algorithms. Non-adaptive algorithms found its applications in a wide range of practical areas such as DNA library screening [12, 32], besides multiple access communications described above. For a general reference on CGT, the reader is referred to a monograph by Du and Hwang [59]. Recently, Ngo and Du [123] gave a survey on non-adaptive pooling designs.

The rest of the chapter is organized as follows. Section 2.2 presents basic definitions, notations and related works. Section 2.3 provides our results and section 2.4 concludes the chapter.

2.2 Preliminaries

Throughout this chapter, for any positive integer v we shall use $[v]$ to denote $\{1, 2, \dots v\}$. Also, given any set X and $k \in \mathbb{N}$, $\binom{X}{k}$ denotes the collection of all k-subsets of X. Naturally, $[0] = \emptyset$ and $\binom{X}{k} = \emptyset$ if $k > |X|$.

2.2.1 The Matrix Representation

Consider a $v \times n$ 01-matrix M. Let R_i and C_j denote row i and column j respectively. Abusing notation, we also let R_i (resp. C_j) denote the set of column (resp. row) indices corresponding to the 1-entries. The *weight* of a row or a column is the number of 1's it has.

Definition 2.1. M is said to be *d-disjunct* if the union of any d columns does not contain another column.

A d-disjunct $v \times n$ matrix M can be used to design a non-adaptive group testing algorithm on n items by associating the columns with the items and the rows with the pools to be tested. If $M_{ij} = 1$ then item j is contained in pool i (and thus test i). If there are no more than d positives and the test outcomes are error-free, then it is easy to see that the test outcomes uniquely identify the set of positives. We simply identify the items contained in negative pools as *negatives* (good items) and the rest as *positives* (defected items). Notice that d-disjunct property implies that each set of at most d positives corresponds uniquely to a test outcome vector, thus decoding test outcomes involves only a table lookup. The design of a d-disjunct matrix is thus naturally called a *non-adaptive pooling design*. We shall use this term interchangeably with the long "non-adaptive combinatorial group testing algorithm".

Let $S(\bar{d}, n)$ denotes the set of all subsets of n items (or columns) with size at most d, called the set of *samples*. For $s \in S(\bar{d}, n)$, let $P(s)$ denote the union of all columns corresponding to s, i.e. $P(s) = \bigcup_{i \in s} C_i$. A pooling design is e-error-detecting (correcting) if it can detect (correct) up to e errors in test outcomes. In other words, if a design is e-error-detecting then the test outcome vectors form a v-dimensional binary code with minimum Hamming distance at least $e + 1$. Similarly, if a design is e-error-correcting then the test outcome vectors form a v-dimensional binary code with minimum Hamming distance at least $2e + 1$. The following remarks are simple to see, however useful later on.

Remark 2.2. Suppose M has the property that for any $s, s' \in S(\bar{d}, n), s \neq s'$, $P(s)$ and $P(s')$ viewed as vectors have Hamming distance $\geq k$. In other words, $|P(s) \oplus P(s')| \geq k$ where $\oplus$ denotes the symmetric difference. Then, M is $(k-1)$-error-detecting and $\lfloor \frac{k-1}{2} \rfloor$-

error-correcting.

Remark 2.3. M being d-disjunct is equivalent to the fact that for any set of $d+1$ distinct columns $C_{j_0}, \ldots C_{j_d}$ with one column (say C_{j_0}) designated, C_{j_0} has a 1 in some row R where all C_{j_k}'s, $1 \leq k \leq d$ contain 0's.

2.2.2 Related Works

Previous works on error-tolerance designs are those of Dyachkov, Rykov and Rashad [64], Aigner [1], Muthukrishnan [122], Balding and Torney [13] and Macula [113, 114]. Dyachkov, Rykov and Rashad [64] derived upper and lower bounds for the test to item ratio given the number of tolerable errors, maximum number of positives, and the size of the population. Aigner [1] and Muthukrishnan [122], discussed optimal strategies when $d = 1$ and the number of errors is small, although in a slightly more general setting where each test outcome could be q-ary instead of binary. Balding and Torney [13] studied several instances of the problem when $d \leq 2$. In some specific case, they showed that an optimal strategy is possible if and only if certain Steiner system exists. In [114] Macula showed that his construction is error-tolerant with high probability, while in [113] he constructed e-error-tolerant d-disjunct matrices for certain values of e.

On construction of disjunct matrices, the most well-known method is to construct the matrix from *set packing designs*. This method was introduced by Kautz and Singleton [97] in the context of superimposed codes. A t-(v, k, λ) packing is a collection $\mathcal{F}$ of k-subsets of $[v]$ such that any t-subset of $[v]$ is contained in at most λ members of $\mathcal{F}$. When $\lambda = 1$ we can construct a $v \times |\mathcal{F}|$ d-disjunct matrix M from a t-$(v, k, 1)$ packing if $k > d(t-1)$. We simply index M's rows by members of $[v]$ and M's columns by members of $\mathcal{F}$, where there is a 1 in row $i \in [v]$ and column $F \in \mathcal{F}$ iff $i \in F$. Little is known about optimal set packing designs except for the case $t < 4$ (see, for example, [23, 123] for more details). Besides taking results directly from Design Theory, other works known on directly constructing d-disjunct matrices are those of Macula [112], Dýachkov, Macula, and Rykov [63].

2.3 Main Results

We first describe our d-disjunct matrices. Given integers $m \geq k > d \geq 1$. A matching of size l (i.e. it has l edges) is called an l-matching.

Definition 2.4. Let $M(m, k, d)$ be the 01-matrix whose rows are indexed by the set of all d-matchings on K_{2m}, and whose columns are indexed by the set of all k-matchings on K_{2m}. All matchings are to be ordered lexicographically. $M(m, k, d)$ has a 1 in row i and column j if and only if the i^{th} d-matching is contained in the j^{th} k-matching.

For q being a prime power, let $\mathbb{F}_q$ denote $GF(q)$. Let $\begin{bmatrix} \mathbb{F}_q^m \\ l \end{bmatrix}$ denote the set of all l-dimensional subspaces (l-subspaces for short) of the m-dimensional vector space on $\mathbb{F}_q$.

Definition 2.5. Let $M_q(m, k, d)$ be the 01-matrix whose rows (resp. columns) are indexed by elements of $\begin{bmatrix} \mathbb{F}_q^m \\ d \end{bmatrix}$ (resp. $\begin{bmatrix} \mathbb{F}_q^m \\ k \end{bmatrix}$). We also order elements of these set lexicographically. $M_q(m, k, d)$ has a 1 in row i and column j if and only if the i^{th} d-subspace is a subspace of the j^{th} k-subspace of $\mathbb{F}_q^m$.

We now show that $M(m, k, d)$ and $M_q(m, k, d)$ are d-disjunct.

Theorem 2.6. *Let $g(m, l) = \binom{2m}{2l}\frac{(2l)!}{2^l l!}$, $v = g(m, d)$, and $n = g(m, k)$. For $m \geq k > d \geq 1$, $M(m, k, d)$ is a $v \times n$ d-disjunct matrix with row weight $g(m - d, k - d)$ and column weight $\binom{k}{d}$.*

Proof. It is easy to see that $g(m,l)$ is the number of l-matchings of K_{2m}. Thus, $M(m,k,d)$ is a $v \times n$ matrix with row weight $g(m-d,k-d)$ and column weight $\binom{k}{d}$.

To show $M(m,k,d)$ is d-disjunct, we recall Remark 2.3. Consider $d+1$ distinct columns $C_{j_0}, C_{j_1}, \dots C_{j_d}$ of $M(m,k,d)$. Since all these columns are distinct k-matchings, for each $i \in [d]$ there exists an edge e_i of K_{2m} such that $e_i \in C_{j_0} \setminus C_{j_i}$. Hence, there exists a d-matching $R \subset C_{j_0}$ which contains all e_i's. To form a row R in which C_{j_0} has a 1 and no other C_{j_i} has, we simply add more edges in C_{j_0} to the set $\{e_i : i \in [d]\}$ until we get a matching of cardinality d. Obviously, $R \not\subset C_{j_i}, \forall i \in [d]$, so C_{j_0} has a 1 in row R where all other C_{j_i} contains 0. ☐

Theorem 2.7. *Let* $\begin{bmatrix} m \\ l \end{bmatrix}_q := \frac{(q^m-1)(q^{m-1}-1)\dots(q^{m-l+1}-1)}{(q^l-1)(q^{l-1}-1)\dots(q-1)}$, $v = \begin{bmatrix} m \\ d \end{bmatrix}_q$, *and* $n = \begin{bmatrix} m \\ k \end{bmatrix}_q$. *For* $m \geq k > d \geq 1$, $M_q(m,k,d)$ *is a* $v \times n$ *d-disjunct matrix with row weight* $\begin{bmatrix} m-d \\ k-d \end{bmatrix}_q$ *and column weight* $\begin{bmatrix} k \\ d \end{bmatrix}_q$.

Proof. It is standard that the Gaussian coefficient $\begin{bmatrix} m \\ l \end{bmatrix}_q$ counts the number of l-subspaces of $\mathbb{F}_q^m$ (see, for example, Chapter 24 of [153]). The weight of any column C of $M_q(m,k,d)$ is the number of d-subspaces of C, hence it is $\begin{bmatrix} k \\ d \end{bmatrix}_q$. The weight $w(R)$ of any row R is the number of k-subspaces of $\mathbb{F}_q^m$ which contains the d-subspace R. To show $w(R) = \begin{bmatrix} m-d \\ k-d \end{bmatrix}_q$, we employ a standard trick, namely double counting. Let $I(m,k,d)$ be the number of ordered tuples $(v_1, \dots, v_{k-d})$ of $k-d$ vectors in $\mathbb{F}_q^m \setminus R$ such that each v_i is not in the span of R and other v_j's, $j \neq i$. Notice that $|\mathbb{F}_q^m| = q^m$ and $|R| = q^d$. Counting $I(m,k,d)$ directly, there are $q^m - q^d$ ways to choose v_1, then $q^m - q^{d+1}$ ways to choose v_2 and so on. Thus,

$$I(m,k,d) = (q^m - q^d)(q^m - q^{d+1}) \dots (q^m - q^{k-1}) \tag{2.1}$$

On the other hand, $(v_1, \dots, v_{k-d})$ can be obtained by first picking a k-subspace C of $\mathbb{F}_q^m$ which contains R in $w(R)$ ways, then $(v_1, \dots, v_{k-d})$ is chosen from $C \setminus R$ in $I(k, k, d)$ ways. This yields

$$I(m, k, d) = w(R)I(k, k, d) \tag{2.2}$$

Combining (2.1) and (2.2) gives $w(R) = \begin{bmatrix} m-d \\ k-d \end{bmatrix}_q$ as desired. The fact that $M_q(m, k, d)$ is d-disjunct can be shown in a similar fashion to the previous theorem. □

The following lemma tells us how to choose k so that the test to item ratio ($\frac{v}{n}$) is minimized. The proof is easy to see and we omit it here.

Lemma 2.8. *For l goes from 1 to m, we have*

(i) The sequence $g(m, l)$ is unimodal and gets its peak at $l = \lfloor m - \sqrt{\frac{m+1}{2}} \rfloor$.

(ii) The sequence $\begin{bmatrix} m \\ l \end{bmatrix}_q$ is unimodal and gets its peak at $l = \lfloor \frac{m}{2} \rfloor$.

Before exploring further properties of $M(m, k, d)$, we need a definition and a lemma.

Definition 2.9. Let $C_{j_0}, C_{j_1}, \dots C_{j_d}$ be any $d+1$ distinct columns of $M(m, k, d)$. A d-matching R is said to be *private for* C_{j_0} with respect to $C_{j_1}, \dots C_{j_d}$ if $R \in C_{j_0} \setminus \bigcup_{i \in [d]} C_{j_i}$. Let $p(C_{j_0}; C_{j_1}, \dots C_{j_d})$ denote the number of private d-matchings of C_{j_0} with respect to $C_{j_1}, \dots C_{j_d}$.

Lemma 2.10. *Given integers $m > d \geq 1$ and any labeled simple graph G with $|V(G)| = m$ and $|E(G)| = d$. Then, the number of vertex covers of size d (or d-covers for short) of G is at least $d + 1$.*

Proof. Decompose G into its connected components. Suppose $G_1, \ldots, G_x$ are connected components which are not trees, and $G'_1, \ldots, G'_y$ are the rest of the components. Isolated vertices are also considered to be trees, so that G'_i is a tree for all $i \in [y]$. For $i = 1, \ldots x$, let $v_i = |V(G_i)|$ and $e_i = |E(G_i)|$. For $i = 1, \ldots y$, let $v'_i = |V(G'_i)|$ and $e'_i = |E(G'_i)|$. The following equations are straight from the definitions :

$$\sum_{i \in [x]} v_i + \sum_{i \in [y]} v'_i = m \quad (2.3)$$

$$\sum_{i \in [x]} e_i + \sum_{i \in [y]} e'_i = d \quad (2.4)$$

hence,

$$0 \leq \sum_{i \in [x]} e_i - \sum_{i \in [x]} v_i = y - (m - d) \quad (2.5)$$

Observe that for any connected simple graph H, picking any $|V(H)| - 1$ vertices out of $V(H)$ gives us a vertex cover. Hence, the number of $(|V(H)| - 1)$-covers of H is at least $\binom{|V(H)|}{|V(H)|-1} = |V(H)|$. To this end, notice that a d-cover of G could be formed by two methods as follows.

(a) *Method 1.* For each $i \in [x]$, pick in v_i ways a $(v_i - 1)$-cover for G_i, then cover all other G_j, $j \neq i$, with all of their vertices. We have used up $(\sum_{i \in [x]} v_i) - 1$ vertices, and need $d - (\sum_{i \in [x]} v_i) + 1$ more to cover all edges of the G'_i's. Firstly, there should be enough number of vertices left. Indeed,

$$\sum_{i \in [y]} v'_i = m - \sum_{i \in [x]} v_i \geq d + 1 - \sum_{i \in [x]} v_i.$$

Secondly, since each G'_i can be covered by $v'_i - 1$ vertices, to cover all G'_i's we need at most $\sum_{i\in[x]}(v'_i - 1)$ vertices. Equations (3) and (5) assure that

$$\sum_{i\in[y]}(v'_i - 1) = m - \sum_{i\in[x]} v_i - y < d + 1 - \sum_{i\in[x]} v_i.$$

In conclusion, this method gives us at least $(\sum_{i=1}^{x} v_i)$ d-covers for G.

(b) *Method 2.* This time, we are greedier by first taking all vertices in G_i's, $i \in [x]$ to cover them. Then, $a = d - \sum_{i\in[x]} v_i$ more vertices are needed to cover the rest. These a vertices can be chosen as follows. For each $(m-d)$-subset Y of $[y]$, cover each $G'_i, i \in Y$ with $v'_i - 1$ vertices, then cover each $G'_i, i \notin Y$ using all of its vertices. Indeed, the total number of vertices used is

$$\sum_{i\in Y}(v'_i - 1) + \sum_{i\notin Y} v'_i = \sum_{i\in[y]} v'_i - |Y| = (m - \sum_{i=1}^{x} v_i) - (m-d) = a.$$

Moreover, obviously there are at least $\prod_{i\in Y} v'_i$ ways to pick d-covers for each particular Y. In total, the number of d-covers formed by Method 2 is at least $\sum_{Y\in\binom{[y]}{m-d}} \prod_{i\in Y} v'_i$.

Noticing that $y \geq m - d \geq 1$, we have

$$\begin{aligned}
\sum_{Y\in\binom{[y]}{m-d}}\prod_{i\in Y} v'_i &= \sum_{Y\in\binom{[y]}{m-d}}\prod_{i\in Y}(e'_i+1) \\
&\geq \sum_{i\in[y]} e'_i + \binom{y}{m-d} \\
&\geq (\sum_{i\in[y]} v'_i - y) + (y-m+d+1) \\
&= d+1-\sum_{i\in[x]} v_i
\end{aligned}$$

Hence, methods 1 and 2 combined yields at least $(d+1)$ different d-covers for G. □

Theorem 2.11. *Given $m > d \geq 1$, and any set of $d+1$ distinct columns $C_{j_0}, C_{j_1}, \dots C_{j_d}$ of $M(m,m,d)$, then $p(C_{j_0}; C_{j_1}, \dots C_{j_d}) \geq d+1$.*

Proof. Observe that for each $i \in [d]$, $C_{j_0} \cup C_{j_i}$ is a loopless multigraph which is 2-regular. In fact, $C_{j_0} \cup C_{j_i}$ consists of cycles with even lengths. Moreover, $C_{j_0} \neq C_{j_i}$ implies that $C_{j_0} \cup C_{j_i}$ must have a cycle of length at least 4; consequently, $|C_{j_0} \setminus C_{j_i}| \geq 2$, $\forall i \in [d]$.

For each $i \in [d]$, choose arbitrarily $E_i \subseteq C_{j_0} \setminus C_{j_i}$ so that $|E_i| = 2$. Let G be the graph with $V(G) = C_{j_0}$, $E(G) = \{E_1, \dots E_d\}$. Then, G is a simple graph having m vertices and $\leq d$ edges. $|E(G)| \leq d$ because the E_i's are not necessarily distinct. Any d-subset R of C_{j_0} such that $R \cap E_i \neq \emptyset$, $\forall i$ is a private d-matching of C_{j_0} with respect to $C_{j_1}, \dots C_{j_d}$. Note that R is nothing but a d-cover of G. To show $p(C_0; C_1, \dots C_d) \geq d+1$, we shall show that the number of d-covers of G is at least $d+1$. Since adding more edges into G can only decrease the number of d-covers, we can safely assume that G has exactly d edges and apply Lemma 2.10. □

Corollary 2.12. *Given integers $m > d \geq 1$, the following holds :*

(i) $M(m, m, d)$ *is* d*-error-detecting and* $\lfloor \frac{d}{2} \rfloor$*-error-correcting.*

(ii) Moreover, if the number of positives is known to be exactly d*, then* $M(m, m, d)$ *is* $(2d + 1)$*-error-detecting and* d*-error-correcting.*

Proof. For any $s, s' \in S(\bar{d}, n)$, $s \neq s'$, without loss of generality we can assume there exists $C_{j_0} \in s \setminus s'$. Theorem 2.11 implies $|P(s) \oplus P(s')| \geq d + 1$, hence Remark 3.18 shows (i). If the number of positives is exactly d, we need to only consider $|s| = |s'| = d$; hence there exist $C_{j_0} \in s \setminus s'$ and $C'_{j_0} \in s' \setminus s$. This time, Theorem 2.11 implies $|P(s) \oplus P(s')| \geq 2d + 2$. Again, Remark 3.18 yields (ii). ☐

Corollary 2.13. *Given integers* $m > d \geq 1$*, then there exists a binary error-correcting code of dimension* $g(m, d)$ *and size* $\binom{g(m,m)}{d}$ *with minimum Hamming distance* $2d + 2$*.*

Proof. The code can be constructed by taking all the unions of d columns in $M(m, m, d)$. Clearly, it is $(2d + 1)$-error-detecting and d-error-correcting. ☐

Borrowing an idea from Macula [114], we get the following algorithm which uses $M(m, k, 2)$ for the at most d positive problem, and show that with very high probability, our algorithm gives the correct answer. Notice that each row of $M(m, k, 2)$ is a 2-matching consisting of some two parallel edges (e_1, e_2) of K_{2m}. We pay attention to $M(m, k, 2)$ because it has good $\frac{v}{n}$ ratio.

Algorithm 2.14. Use $M(m, k, 2)$ to design the pools as usual. For each edge $e \in E(K_{2m})$ such that the total number of positive outcomes involving e is $k - 1$, i.e.

$$|\{\{e, e'\} : \text{ the test } \{e, e'\} \text{ is positive }\}| = k - 1,$$

identify the item

$$C = \{e\} \cup \{e' : \{e, e'\} \text{ is positive }\}$$

as a positive.

Theorem 2.15. *Algorithm 2.14 gives correct answer with probability $P(m,k,d)$ where*

$$P(m,k,d) \geq \left[\frac{\sum_{j=1}^{k} (-1)^{j+1} \binom{k}{j} \binom{\sum_{i=0}^{j} (-1)^{i} \binom{j}{i} g(m-i,k-i)}{d-1}}{\binom{g(m,k)-1}{d-1}} \right]^{d}$$

For example, $P(8,6,9) \geq 98.5\%$. This means that we could use $M(8,6,9)$, which has dimension 5460×18918900, to find at most 9 positives in a population of 18918900 items using only 5460 tests with 98.5% chance of success.

Proof. Given a set of d distinct columns $C_{j_1}, C_{j_2}, \ldots C_{j_d}$. An edge $e \in E(K_{2m})$ is called a *mark* of C_{j_i} if e is a private 1-matching of C_{j_i} with respect to $\{C_{j_l}, l \in [d] \setminus \{i\}\}$, in which case C_{j_i} is said to be *marked*. If C_{j_i} is marked by e then exactly $k-1$ tests involving e and another edge in C_{j_i} is positive. Consequently, Algorithm 2.14 gives correct answer if the set of d positives is a *marked set*, namely every element is marked.

The probability that Algorithm 2.14 gives a correct answer is thus the probability that a random d set of columns of $M(m,k,2)$ is marked. For a fixed C_{j_1}, there are $\binom{g(m,k)-1}{d-1}$ ways to pick the other $d-1$ columns. Let X_i be the event that C_i is marked relative to the other $d-1$ columns, then

$$P(m,k,d) = P(X_1)P(X_2|X_1)P(X_3|X_1,X_2)\ldots \geq (P(X_1))^d.$$

To calculate $P(X_1)$, we count number of ways to pick $d-1$ columns other than C_{j_1} such that C_{j_1} is marked by some $e \in C_{j_1}$. Let A_i be the collection of all $(d-1)$-sets of columns other than C_{j_1} such that $e_i \in C_{j_1}$ marks C_{j_1} with respect to A_i. The answer is then

$|\bigcup\{A_i, 1 \leq i \leq k\}|$. This number can be obtained by applying the inclusion-exclusion principle twice. Dividing it by $\binom{g(m,k)-1}{d-1}$ gives us $P(X_1)$ and proves the theorem. $\square$

2.4 Discussions

We have given the constructions of two different classes of pooling designs. Firstly, $M(m, k, d)$ has good performance when the number of positives is small comparing to the number of items. Deterministically, a larger ratio of positives to items is sometime preferred. Probabilistically, however, $M(m, k, 2)$ could be used to solve the $S(\bar{d}, n)$ problem with very high probability of success. The main strength of this construction is that $M(m, m, d)$ is d-error-detecting. $M_q(m, k, d)$ is the q-analogue of the construction given by Macula [112]. An interesting question is: "what is the q-analogue of a matching?"

One could think of several different variations of the matching idea. For example, a possible generalization is to index the rows (columns) of a matrix $M(m, k, d, l)$ with all graphs having d (k) edges whose vertex degrees are $\leq l$. $M(m, k, d)$ is nothing but $M(m, k, d, 1)$. Further investigations in this direction might lead to better designs.

Lastly, in reality given a specific problem with certain parameters, m and k have to be chosen appropriately to suit one's need. More careful analysis need to be done to help pick the *best* m and k given n, d and/or any other constraints from practice. We need some reasonably good asymptotic formulas to estimate them.

Part II

Indirect Networks

Chapter 3

Rearrangeability and the Beneš Conjecture

3.1 Overview

3.1.1 Rearrangeability

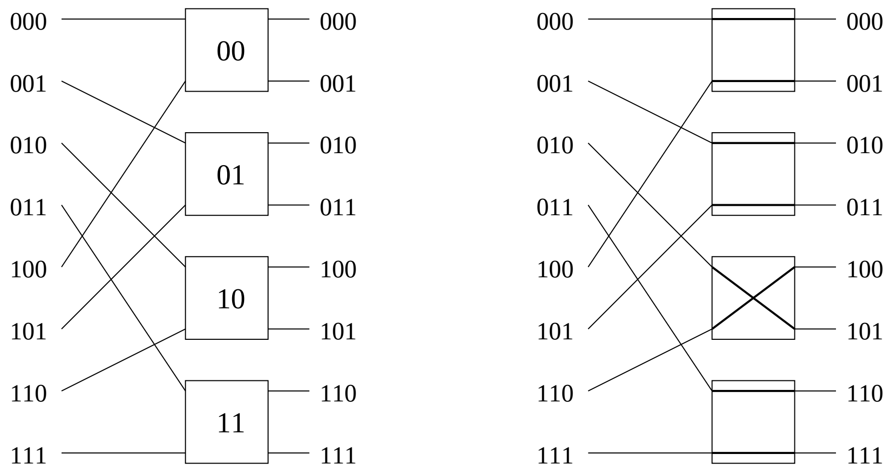
A standard question to be addressed on any indirect interconnection network is that if the network is *rearrangeable*. An N-input N-output network is rearrangeable if and only if any one to one mapping from the inputs to the outputs is routable by the network. *Universality* is another term that is often used synonymously with *rearrangeability*.

The notion of rearrangeability is fundamental from both the theoretical and practical point of view. Practically, any space switching device such as telephone switches [14] and optical cross-connects [79] (OXCs) has an internal rearrangeable network. Theoretically, the notions of expanding graphs (superconcentrators, concentrators and expanders) [132, 133] come from rearrangeable networks. During the past 30 years, the wave of research on the complexity of switching networks, central around expanding graphs, has enriched many areas of Theoretical Computer Science and Mathematics, such as Boolean Circuit Complexity, Algebraic Graph Theory, Probabilistic Method, ... The number of related publications is too large to be cited here. The reader is referred to Pippenger [134] and Ngo and Du [124] for further information.

3.1.2 Shuffle-Exchange Networks

Shuffle-Exchange networks (SE networks for short) were initially proposed by Stone (1971, [148]) to be an efficient interconnecting architecture for parallel processors. Various applications, especially from the field of parallel computing, have benefited from this interconnecting pattern. The literature on SE networks is enormous, we mention a few applications here. Many useful networks are topologically equivalent to SE networks, such as the *data manipulator*, *flip networks*, *regular Banyan networks*, *Omega networks*, *indirect binary n-cubes*, ... [101, 105, 126, 128, 166]. They have been proposed as the primary memory system for array processors accessing slices of data [105]. The SE connection pattern could be used for Fast Fourier Transform, polynomial evaluation, sorting, matrix operations [120, 148]; for solving 2 and 3-dimensional PDEs, radix-2 FFT, matrix multiplications [130]; for 2-dimensional convolution in image processing [65]; as fault-tolerant MIN [31, 102, 156]; as self-routing networks of linear permutations [84, 136]; for packet and message routings [131]; and for multicasting [107].

As SE networks consist of SE stages, let us first introduce the notion of an SE stage. For each natural number n, let $N = 2^n$. Each SE_n stage includes a *perfect shuffle* pattern followed by an array of $\frac{N}{2}$ 2×2 crossbars. Figure 3.1a shows SE_3. For convenience, we number the inputs, outputs and binary switches in binary format as shown. Each input $\overline{x_0 \dots x_n}$ is connected to the switch numbered $\overline{x_1 \dots x_n}$. We put a bar on top of a sequence of variables to signify the fact that it is a binary representation. The connection pattern is called a perfect shuffle because it is like shuffling two halves of a card deck so that the cards from two halves interleave perfectly. Recall that each binary switch has two valid states: crossing or straight, making $2^{N/2} = 2^{2^{n-1}}$ possible one-to-one perfect matchings between the inputs and the outputs. Part (b) of the figure shows one setting of the switches



(a) The numbering (b) Each switch has two valid states

Figure 3.1: A Shuffle-Exchange stage with parameter 3: SE_3

in which 100 is mapped to 001, 110 to 100 and so on. Each matching could be thought of as a permutation on $[N] = \{1, \ldots, N\}$. The matchings form a subset of the symmetric group of order N. (For group theoretic and algebraic notions, see the classic book by Artin [11].) Let σ be the permutation defined by the perfect shuffle, and E be the subgroup of S_N defined by the switches, then the set of permutations defined by SE_n is the left coset σE of E.

A k-stage SE network with parameter n, denoted by $(SE_n)^k$, is a network with N inputs and N outputs, consisting of k consecutive SE_n stages. A typical drawing of a 7-stage SE network with $n = 4$ (i.e. $(SE_4)^7$) is shown in Figure 3.2.

3.1.3 The Beneš Conjecture

In the context of SE networks, a long standing question was that how many SE stages are necessary and sufficient for an SE network to be rearrangeable. In fact, it is not entirely

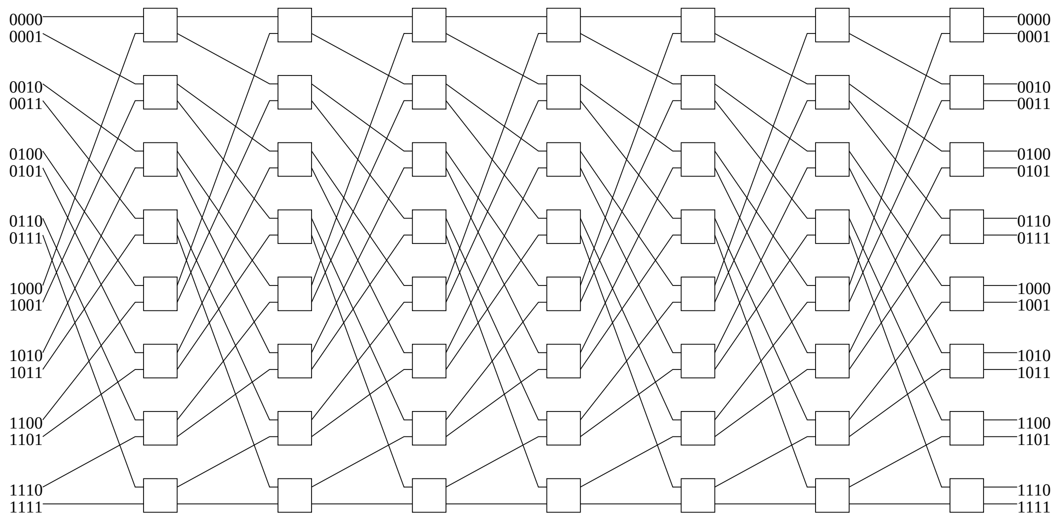


Figure 3.2: The 7-stage SE network for $n = 4$, i.e. $(SE_4)^7$

clear that increasing the number of stages would increase the rearrangeability of an SE network. For convenience, let us use $m(n)$ to denote the minimum positive integer so that $(SE_n)^{m(n)}$ is rearrangeable. Beneš conjectured in 1962 and published in 1975 [15] that $2n - 1$ SE stages is necessary and sufficient to route all $N!$ perfect matchings from the inputs to the outputs, i.e. $m(n) = 2n - 1$. Note that $m(n) = 2n - 1$ is equivalent to the fact that S_N could be decomposed into the product of $2n - 1$ left cosets σE of E, but not fewer.

There has been a very slow progress toward answering this question. The algorithm discussed by Stone (1971, [148]) showed that $m(n) \leq n^2$, thus $m(n)$ is well defined. Parker (1980, [128]) showed that $n + 1 \leq m(n) \leq 3n$, where the lower bound was obtained by a counting argument, and the upper bound by group calculations plus the rearrangeability of the Beneš network [14]. Wu and Feng (1981, [167]) gave an explicit algorithm to route all matchings, proving $m(n) \leq 3n - 1$. Huang and Tripathi (1986, [83]) improved the bound to $m(n) \leq 3n - 3$. Raghavendra and Varma (1987, [137]) verified the conjecture for $N = 8$. They used that result to show $m(n) \leq 3n - 4$ [155]. They also specified a permutation which $(SE_n)^k$ can not route if $k \leq 2n - 2$, in effect showing $2n - 1 \leq m(n)$. With a different formulation, Linial and Tarsi (1989, [110]) also verified the conjecture for $N = 8$ and showed $m(n) \leq 3n - 4$. From their formulation it is easy to see that at least $2n - 1$ stages are needed to route all permutations. Feng and Seo (1994, [67]) gave a proof of the conjecture, which was incomplete as pointed out by Kim, Yoon, and Maeng (1997, [100]). There have been, in fact, several other wrong proofs published in journals during the last 15 years.

In this chapter, we give a proof that $m(4) = 7$ using a new method, and then adapt Linial and Tarsi's results to show that $m(n) \leq 3n - 5$. As we shall see, the $n = 4$ case is considerably more difficult than the $n = 3$ case. I strongly believe that hidden in this proof there is some general technique(s) that would help improve the bound further. In particular, an algebraic formulation of the proof would be of tremendous interest.

3.2 Preliminaries

This section presents related concepts and previous results on the problem. Throughout the chapter, we shall assume that $n \in \mathbb{N}$ and $N = 2^n$. The following definitions and lemmas are from Linial and Tarsi [110].

Definition 3.1. For $k \in \mathbb{N}$, a $N \times k$ 01-matrix A, denoted by $A_{N\times k}$ is said to be *balanced* if

(i) Either $k \leq n$ and every row vector $v \in \mathbb{F}_2^k$ occurs exactly 2^{n-k} times as rows of A.

(ii) or $k > n$ and every n consecutive column vectors of A form a balanced matrix.

Definition 3.2. Given a balanced matrix $A_{N\times(n-1)}$, a column vector $x \in \mathbb{F}_2^N$ is said to *agree* with A if appending x into A yields an $N \times n$ balanced matrix (the matrix $[A, x]$).

Lemma 3.3. *If A and B are two $N \times (n-1)$ balanced matrices, then there exists a vector $x \in \mathbb{F}_2^N$ that agrees with both A and B.*

Lemma 3.4. *Let $A_{N\times n}$ be a 01-matrix such that deleting any column of A yields a balanced $N \times (n-1)$ matrix. Then, either (i) A is balanced, or (ii) each row of A has an even number of 1's, or (iii) each row of A has an odd number of 1's.*

Lemma 3.5. *Let $A_{N\times k}$ be a balanced matrix with $k \le n$, and let T be a non-singular $k\times k$ 01-matrix, then AT is also balanced, where all the arithmetic is done modulo 2.*

Notice that when x agrees with A, we can insert x into any position between the columns of A to get a balanced matrix. It is also easy to see that $(SE_n)^m$ ($m > n$) is rearrangeable if and only if for every two given balanced matrices $A_{N\times n}$ and $B_{N\times n}$ there exists an $N\times(m-n)$ balanced matrix M such that the matrix $[A, M, B]$ is balanced. Here the rows of A are binary representations of the inputs and the corresponding rows of B are binary representations of the matched outputs.

3.3 Main Results

To illustrate the idea and introduce notations needed for the main theorem ($m(4) = 7$), we first reproduce a known result (see [110, 137]) using the new method.

Lemma 3.6. $m(3) = 5$*, namely the network $(SE_3)^5$ is rearrangeable.*

Proof. We use the same approach as that of Raghavendra and Varma [137], namely from first principles. However, the method is different and more intuitive. Figure 3.3 shows a

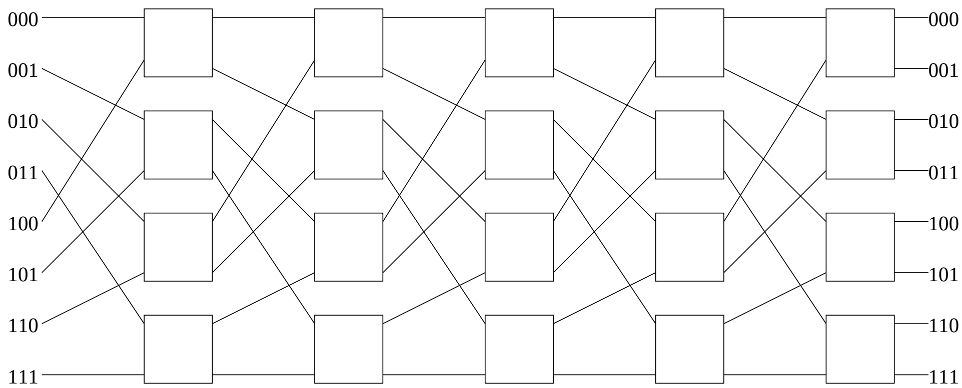


Figure 3.3: The 5-stage SE network for $N = 8$, i.e. $(SE_3)^5$

typical drawing of a 5-stage SE network for $N = 8$. For convenience, the network can be redrawn and the switches can be labeled as shown in Figure 3.4. In the figure, the

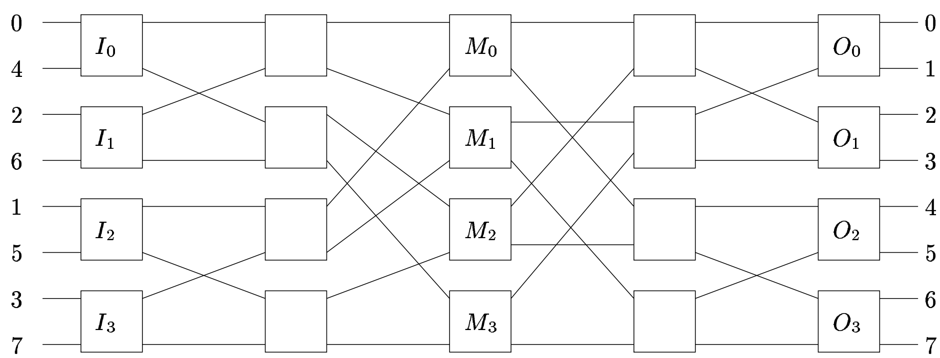


Figure 3.4: A redrawing of the $(SE_3)^5$ network

inputs and outputs have been numbered in decimals for convenience. We write $x \in I_i$ ($y \in O_j$) if input x (output y) is connected to input switch I_i (output switch O_j). Given a perfect matching π (or permutation) from the inputs $\{0, 4, 2, 6, 1, 5, 3, 7\}$ to the outputs $\{0, 1, 2, 3, 4, 5, 6, 7\}$, we first construct a 4×4, 2-regular multi-bipartite graph $G(\pi) = (I, O, E)$ whose bipartitions are $I = O = \{0, 1, 2, 3\}$. I and O correspond to the set of input and output switches respectively. We shall refer to $G(\pi)$ as G if π is clear from the context. $(i, j) \in E$ iff $\pi(x) = y$ for some $x \in I_i$ and $y \in O_j$, introducing multiple edges if necessary. We now need some notations. Suppose we have colored the edges of G with colors in $C = \{0, 1, 2, 3\}$. For each $c \in C$, let $L(c)$ ($R(c)$) be the multi-set of the vertices in I (O) which are incident to an edge colored c. For each subset $S \subseteq C$, let $L(S) = \bigcup_{c \in S} L(c)$ and $R(S) = \bigcup_{c \in S} R(c)$, where the union is multiset union. For each $e \in E$, let $l(e)$ ($r(e)$) denote the vertex in I (O) incident to e. Similarly, for any subset $A \subseteq E$, let $L(A) = \{l(e) \mid e \in A\}$ and $R(A) = \{r(e) \mid e \in A\}$.

To this end, we observe from Figure 3.4 that the realizability of the matching is equivalent to the existence of a coloring of G with colors in C such that

(P_1) For each $c \in C$, $L(c)$ has a representative from each of $\{0, 1\}$ and $\{2, 3\}$.

(P_2) $L(\{0, 1\}) = L(\{2, 3\}) = \{0, 1, 2, 3\}$. In other words, $L(\{0, 1\})$ and $L(\{2, 3\})$ have distinct elements.

(P'_1) For each $c \in C$, $R(c)$ has a representative from each of $\{0, 1\}$ and $\{2, 3\}$.

(P'_2) $R(\{0, 2\}) = R(\{1, 3\}) = \{0, 1, 2, 3\}$. In other words, $R(\{0, 2\})$ and $R(\{1, 3\})$ have distinct elements.

Note that these conditions imply that each color appears exactly twice. The conditions are chosen so that the two edges colored $c \in \{0, 1, 2, 3\}$ can be routed through middle switch M_c. We will not state and prove the correctness of any routing algorithm based on the coloring here, as it is straightforward.

We now describe a procedure to properly color all 4×4 2-regular multi-bipartite graphs G as follows. Along the way, we shall also prove that our procedure works.

Phase 1. As G is 2 regular and multi-bipartite, it is the union of even cycles. G thus can be decomposed into two 4×4 perfect matchings by taking alternate edges on each cycle.

Let the matchings be M_1 and M_2 (whose vertex sets are the same as G.)

Phase 2. From each M_i ($i = 1, 2$), construct a 2×2 2-regular bipartite graph G_i by combining within each bipartition of M_i the pairs of vertices $\{0, 1\}$ and $\{2, 3\}$. Figure 3.5 illustrates the results of our first two phases. Obviously, $L(E(G_i)) = R(E(G_i)) = \{0, 1, 2, 3\}$, for $i = 1, 2$. Here and henceforth the L and R functions are applied in the context of the original graph G.

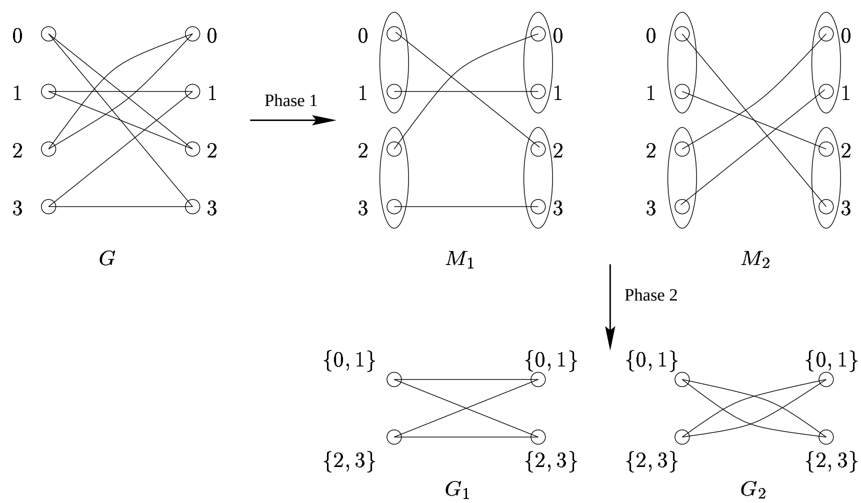
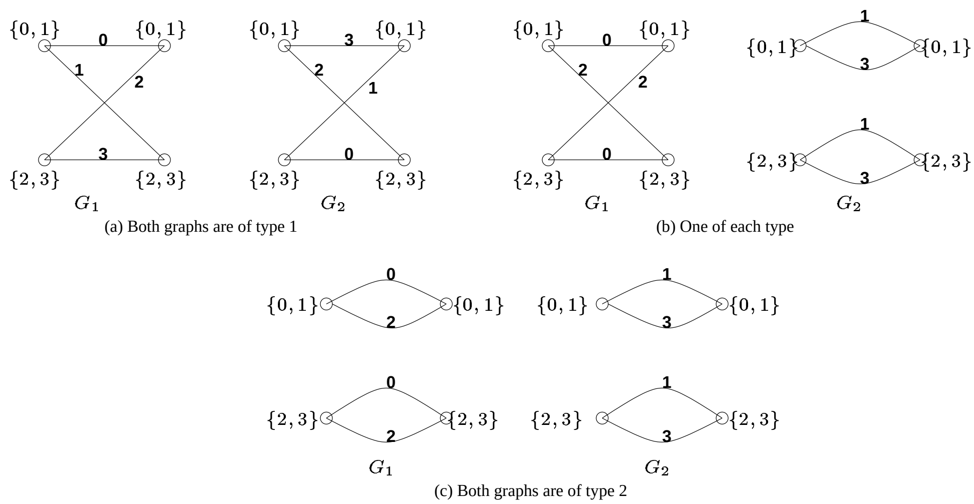


Figure 3.5: An illustration of the first two phases

We call the graphs G_1 and G_2 the *basic components* of G. Since the basic components are 2×2 2-regular bipartite graphs, they can only be either a 4-cycle or a union of two 2-cycles. A basic component is said to be of *type 1* if it is a 4-cycle and of *type 2* otherwise. In Figure 3.5, G_1 is of type 1 and G_2 is of type 2.

Phase 3. As each coloring of G_1 and G_2 induces uniquely a coloring of G, we are to color G_1 and G_2 so that the coloring satisfy conditions P_i and P_i', $1 \leq i \leq 2$. We call an edge whose color is $c \in C$ a *c-edge*. Consider two cases:

Case 1 Both G_1 and G_2 are of type 1. In this case we color the graphs as shown in Figure 3.6a. It is easy to see that the coloring satisfies all prescribed conditions. The basic idea is that as we have used each color exactly twice, to enforce P_1 and P_1' we need to make sure that if there is a c-edge going from $\{0,1\}$ to $\{0,1\}$, then the other c-edge must go from $\{2,3\}$ to $\{2,3\}$ in either basic components, and similarly if a c-edge going from $\{0,1\}$ to $\{2,3\}$ then the other c-edge must go from $\{2,3\}$ to $\{0,1\}$. To enforce P_2 and P_2', on the left side (the I side) we *separate* each color pair $\{0,1\}$ (i.e. $L(0) \cap L(1) = \emptyset$) and $\{2,3\}$ (i.e. $L(2) \cap L(3) = \emptyset$), while on the right (the O side) we separate the pairs $\{0,2\}$ ($R(0) \cap R(2) = \emptyset$) and $\{1,3\}$ ($R(1) \cap R(3) = \emptyset$).



(a) Both graphs are of type 1

(b) One of each type

(c) Both graphs are of type 2

Figure 3.6: Illustration of the colorings when $n = 3$

Case 2 There is one graph of type 2. Without loss of generality, assume G_2 is of type 2 as illustrated in Figures 3.6b and 3.6c. In this case we color G_1 with $\{0,2\}$ and G_2 with $\{1,3\}$. Notice that P_1, P_1', and P_2' are satisfied even if we switch colors in one (or both) 2-cycles of G_2. To ensure P_2, we do this switching if necessary at each 2 cycle of G_2 to separate each pair $\{0,1\}$ and $\{2,3\}$ on the left.

□

Secondly, we use the idea to derive a more elaborate proof for the case where $N = 16$. Firstly, we redraw the network as shown in Figure 3.7, so that it is easier to derive the conditions similar to the P_i and P'_i. Given any perfect matching π from the inputs to the

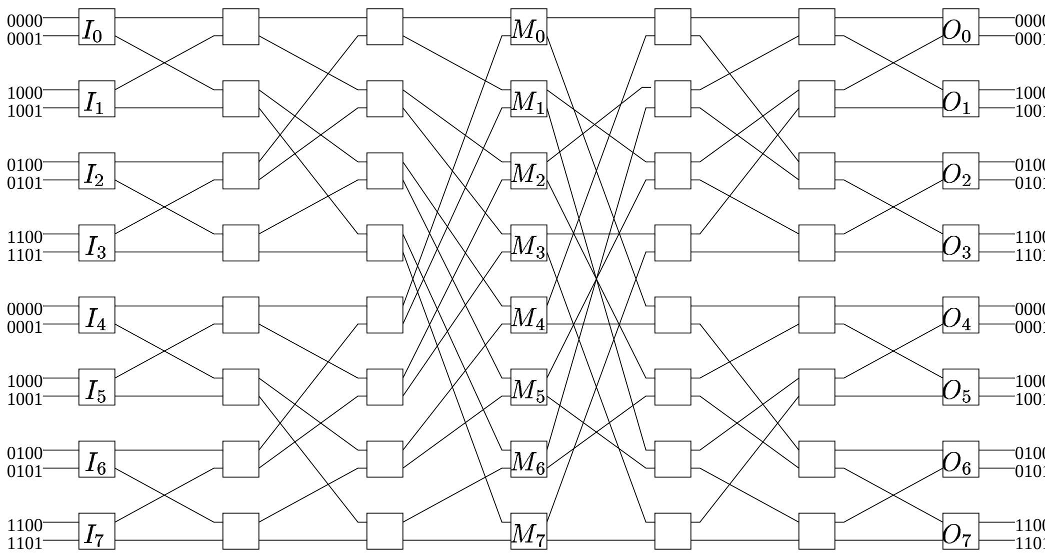


Figure 3.7: A redrawing of the $(SE_4)^7$ network

outputs, we first construct the 8×8 2-regular multi-bipartite graph G in a similar way as the G in Lemma 3.6. The bipartitions of G are $I = O = \{0, \ldots, 7\}$, and $(i, j) \in E(G)$ if for some $x \in \{0, \ldots, 15\}$ we have $x \in I_i$ and $\pi(x) \in O_j$. From Figure 3.7, the following proposition is easy to see. We reuse all notations introduced in the proof of Lemma 3.6. Again, as a valid coloring induces a routing algorithm in a straightforward way, we shall not describe the algorithm here.

Proposition 3.7. *The fact that $(SE_4)^7$ is rearrangeable is equivalent to the fact that for any 8×8 2-regular multi-bipartite graph $G = (I, O)$ with bipartitions $I = O = \{0, \ldots, 7\}$,* *there exists an edge coloring of G using colors in $C = \{0, \ldots, 7\}$ satisfying the following conditions:*

(P_1) For each $c \in C$, $L(c)$ has a representative from each of $\{0,1,2,3\}$ and $\{4,5,6,7\}$.

(P_2) For each pair $\{c_1, c_2\} \in \{\{0,1\},\{2,3\},\{4,5\},\{6,7\}\}$, $L(\{c_1,c_2\})$ has a representative from each of $\{0,1\}$, $\{2,3\}$, $\{4,5\}$, and $\{6,7\}$.

(P_3) $L(\{0,1,2,3\}) = L(\{4,5,6,7\}) = \{0,1,\ldots,7\}$. In other words, the elements of $L(\{0,1,2,3\})$ and $L(\{4,5,6,7\})$ are all distinct.

(P_1') For each $c \in C$, $R(c)$ has a representative from each of $\{0,1,2,3\}$ and $\{4,5,6,7\}$.

(P_2') For each pair $\{c_1, c_2\} \in \{\{0,4\},\{2,6\},\{1,5\},\{3,7\}\}$, $R(\{c_1,c_2\})$ has a representative from each of $\{0,1\}$, $\{2,3\}$, $\{4,5\}$, and $\{6,7\}$.

(P_3') $R(\{0,4,2,6\}) = R(\{1,5,3,7\}) = \{0,1,\ldots,7\}$. In other words, the elements of $R(\{0,4,2,6\})$ and $R(\{1,5,3,7\})$ are all distinct.

Note that these conditions imply that each color appears exactly twice. Again, the conditions were specifically chosen so that each pair of edges with the same color $c \in C$ can be routed through middle switch M_c without causing any conflict. From now on, we shall refer to a *valid* coloring of G as the coloring satisfying the prescribed conditions in Proposition 3.7. The following proposition further explores properties of a graph G which can be validly colored.

Proposition 3.8. *G can be validly colored if and only if the graph G' obtained from G by applying one of the following operations can also be validly colored. Let X be either I or O, the operations are:*

1. *Switch labels of the vertices* $\{v_1, v_2\}$ *in* X*, where*

$$\{v_1, v_2\} \in \{\{0,1\}, \{2,3\}, \{4,5\}, \{6,7\}\}.$$

2. *Switch labels of each pair* $\{0, 2\}$ *and* $\{1, 3\}$ *in* X.

3. *Switch labels of each pair* $\{4, 6\}$ *and* $\{5, 7\}$ *in* X.

4. *Switch labels of each pair* $\{0, 4\}$, $\{1, 5\}$, $\{2, 6\}$, *and* $\{3, 7\}$ *in* X.

5. *Flip* G *horizontally, i.e. switch labels of each pair* $\{0, 7\}$, $\{1, 6\}$, $\{2, 5\}$, *and* $\{3, 4\}$ *in both* I *and* O.

6. *Flip* G *vertically, i.e.* G' *is the mirror image of* G.

Proof. It's not difficult to see that the valid coloring of G induces a valid coloring of G' under all cases except the operation of flipping G vertically. In this case, from the coloring of G we can obtain a coloring of G' by the following mapping of colors: if in G an edge is colored c, whose binary representation is $\overline{c_1 c_2 c_3}$, then we use $\overline{c_3 c_2 c_1}$ to color the edge in G'. The verification that this is indeed a valid coloring of G' is mechanical and we shall not attempt to do so here. □

Theorem 3.9. $m(4) = 7$*, namely the network* $(SE_4)^7$ *is rearrangeable.*

Proof. Let G be an 8×8 2-regular multi-bipartite graph. To color G properly, i.e. the coloring satisfies the conditions of Proposition 3.7, we decompose G into 4 basic components. The decomposition is formally described below. Figure 3.8 illustrates the decomposition procedure.

Phase 1 Decompose G into two edge disjoint 8×8 perfect matchings M_1 and M_2.

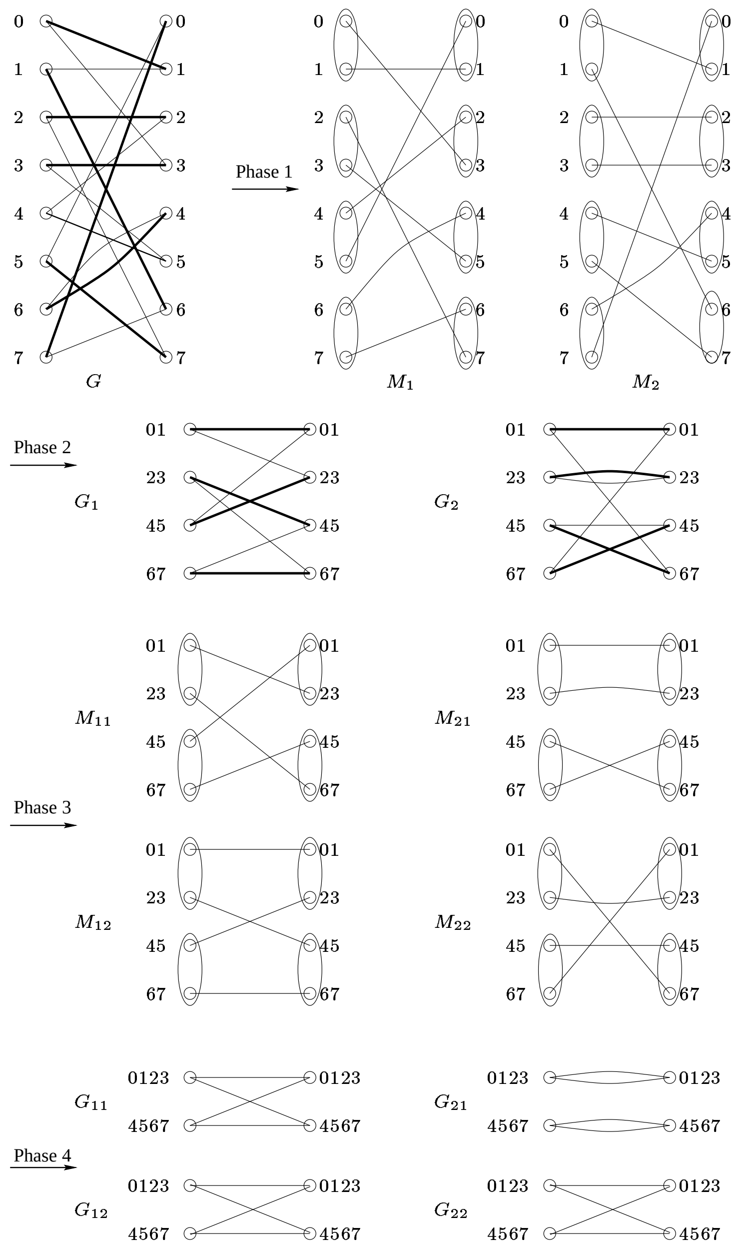


Figure 3.8: An illustration of the basic component decomposition

Phase 2 For each $i = 1, 2$, construct the graph G_i by collapsing the pairs of vertices $\{0,1\}$, $\{2,3\}$, $\{4,5\}$, and $\{6,7\}$ on each bipartition of M_i. It is clear that the graphs G_i are 4×4 2-regular bipartite graphs.

Phase 3 For each $i = 1, 2$, decompose G_i into two edge disjoint 4×4 perfect matchings M_{i1} and M_{i2}.

Phase 4 For each $i = 1, 2$ and $j = 1, 2$, construct the graph G_{ij} by collapsing the pairs of vertices $\{01, 23\}$ and $\{45, 67\}$ on each bipartition of M_{ij}. As before, the G_{ij} are called *basic components* of G, and can only be one of two types: (a) type 1 corresponds to a 4-cycle and (b) type 2 corresponds to two 2 cycles. We are now ready to color the basic components so that the (uniquely) induced coloring on G is valid.

As we have seen in the proof of Lemma 3.6, the number of type-2 basic components can roughly be thought of as the degree of flexibility in finding a valid coloring for G. When there is no type-2 basic component, we color the edges of G_{ij} as shown in Figure 3.9. The coloring clearly satisfies conditions P_i and P'_j $(0 \leq i \leq 3, 1 \leq j \leq 3)$.

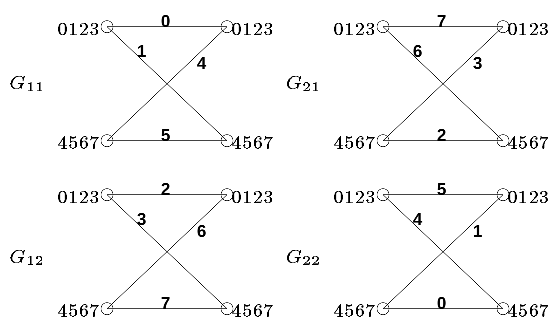


Figure 3.9: The coloring when there is no type-2 basic component

The rest of the cases are considered in Lemmas 3.11, 3.12, and 3.13 with the help of Lemma 3.10 and Proposition 3.8. The basic idea is that as these cases involve at least one type-2 component, Lemma 3.10 allows us to color the other three basic components using certain set of 6 colors, without worrying about coloring the last basic component. Proposition 3.8 helps simplify case analysis. Roughly, the graphs Gs could be partitioned into "equivalent" classes, where graphs from each of the class can be obtained from one another by applying a sequence of operations described in Proposition 3.8. $\square$

Lemma 3.10. *Assume G has at least one basic component of type 2, say G_{22}. Let*

$$C_L = \{\{0,1\},\{2,3\},\{4,5\},\{6,7\}\},$$
$$C_R = \{\{0,4\},\{2,6\},\{1,5\},\{3,7\}\}$$

and $\{c_1,c_2\} \in C_L \cup C_R$ be an arbitrary pair of colors in the set. If the other three basic components can be colored using colors in $C - \{c_1,c_2\}$ so that none of the properties P_i ($0 \le i \le 3$) and P'_j ($1 \le j \le 3$) are violated, then G_{22} can be colored with c_1 and c_2 to form a completely valid coloring of G.

Proof. We only show the lemma for the case where $\{c_1,c_2\} = \{0,1\}$. Other cases are easily seen to be similar. We shall try to color each 2-cycle of G_{22} with 0 and 1, switching the colors if necessary.

Firstly, we claim that by coloring each 2-cycle of G_{22} with 0 and 1, properties P_i ($0 \le i \le 3$), and P'_1 hold, no matter which edge in each 2 cycle gets which color. Indeed, P_1, and P'_1 hold trivially. Let e_1 and e_2 be two edges in any 2-cycle of G_{22}, then P_2 holds because $L(\{e_1,e_2\})$ has **either** a representative from each of $\{0,1\}$ and $\{2,3\}$ **or** $\{4,5\}$ and $\{6,7\}$. Since we have assumed that P_3 was not violated, before the new colors 0 and 1 arrives, $L(\{4,5,6,7\}) = \{0,1,\dots,7\}$ and $L(\{2,3\}) \subset \{0,1,\dots,7\}$. In fact, $L(\{2,3\})$ has 4 distinct members since P_2 was not violated. Thus, as G is 2-regular $L(\{0,1\}) = \{0,1,\dots,7\} - L(\{2,3\})$, no matter how we assign 0 and 1 to the edges of G_{22}. Hence, P_3 holds.

Notice that $R(4)$ and $R(5)$ each has a representative from each of $\{0,1,2,3\}$ and $\{4,5,6,7\}$. We claim that $R(\{4,5\})$ has a representative from each of $\{0,1\}$, $\{2,3\}$, $\{4,5\}$, and $\{6,7\}$. Assume for contradiction, without loss of generality, that $R(\{4,5\}) \cap \{0,1\} = \emptyset$. Let's look at the 4 edges whose right end points are 2 or 3. Two of them are colored 4 and 5 as $R(\{4,5\}) \cap \{0,1\} = \emptyset$. The third one is one of the 2 edges in a 2-cycle of G_{22}. The fourth edge must have had a color from $\{2,6,3,7\}$, say 2 or 6. Moreover, of the four edges whose right end points are 0 or 1, one of them is the other edge in the 2-cycle of G_{22}, one of them must have been colored 6 or 2 because P'_2 was not violated, the last two have to get colors 3 and 7 as P'_1 holds for 3 and 7. However, this makes P'_2 invalid for the color pair $\{3,7\}$. Contradiction!

Now, we try to switch colors in each 2-cycle of G_{22} if necessary to achieve P'_2. Let e_1, and e_2 be the two edges at the 2-cycle whose right end points are in $\{0,1,2,3\}$. By construction, $R(\{e_1,e_2\})$ has a representative from each of $\{0,1\}$ and $\{2,3\}$. Assign colors 0 and 1 to e_1 and e_2 so that $R(\{0,4\})$ has a representative from each of $\{0,1\}$ and $\{2,3\}$. Notice that this implies $R(\{1,5\})$ has a representative from each of $\{0,1\}$ and $\{2,3\}$, too. The same procedure is done with the other 2-cycle of G_{22}.

Lastly, we show that P'_3 holds automatically. Notice that P'_3 is equivalent to the condition that each vertex i of the bipartition O of G is incident to two edges whose colors have different parities. Let $\{v_1,v_2\} \in \{\{0,1\},\{2,3\},\{4,5\},\{6,7\}\}$ be a pair of right side vertices. It is easy to see that after we have colored the edges of G_{22} as above, of the four edges whose right end points are in $\{v_0,v_1\}$, two have even colors and the other two have odd colors. So if after the coloring either v_0 or v_1 is incident to colors of the same parity, then so is the other. However, this means that there must have been a violation of P'_3 even before the coloring of G_{22}, because there is only one edge of G_{22} whose right end point is in $\{v_0, v_1\}$. □

Lemma 3.11. *G can be validly colored if there is at most 1 basic component of type 1.*

Proof. G could either have one or zero type-1 basic component. Without loss of generality, we assume the three type-2 components are G_{12}, G_{21}, and G_{22}. The coloring is roughly shown in Figure 3.10. The idea is to fix the coloring of G_{11} by two colors $\{0, 4\}$ as shown

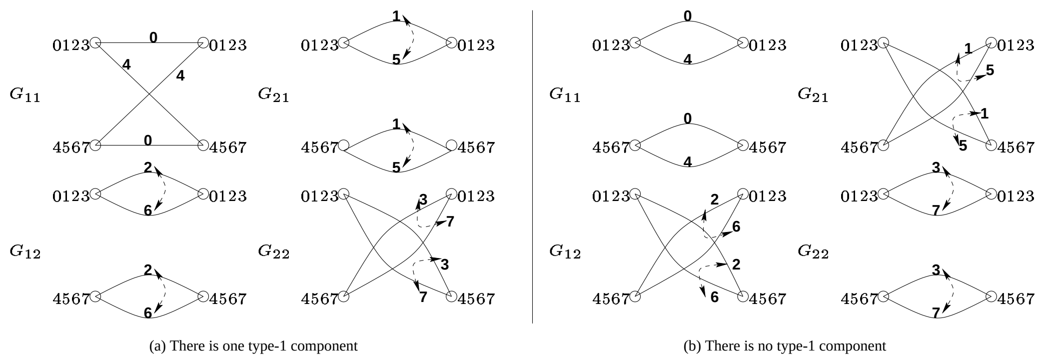


Figure 3.10: The cases when there is at most one type-1 component

in the figure, then switch the assigned pair of colors at each 2-cycle if necessary to form a valid coloring of G.

Obviously, properties P_1, P_i' $(1 \le i \le 3)$ hold. We now do the switching on each 2-cycle of G_{12} so that $|\{0,1\} \cap L(\{0,2\})|$ is either 0 or 2, and that $|\{4,5\} \cap L(\{0,2\})|$ is either 0 or 2. This is certainly possible. Notice that this implies $|\{0,1\} \cap L(\{4,6\})|$ and $|\{4,5\} \cap L(\{4,6\})|$ are also either 0 or 2. Secondly, we do the switching on each 2-cycle of G_{21} and G_{22} so that $|\{0,1\} \cap L(\{0,2\})| + |\{0,1\} \cap L(\{1,3\})| = 2$ and $|\{4,5\} \cap L(\{0,2\})| + |\{4,5\} \cap L(\{1,3\})| = 2$. Intuitively, we want to "separate" the left end points of the edges having colors in $\{0,2\}$ from the left end points of edges having colors in $\{1,3\}$, in order to maintain properties P_2. The same assertion holds for the pairs $\{4,6\}$ and $\{3,5\}$. Lastly, P_3

is assured by the fact that G_{i1} and G_{i2} have edges which form an 8×8 matching, for each $i = 1, 2$. $\square$

The proofs of the following lemmas are put in separate sections, as they are long and thus would distract the reader from the main line of reasoning.

Lemma 3.12. *G can be validly colored if there are exactly 2 basic components of type 1.*

Lemma 3.13. *G can be validly colored if there are exactly 3 basic components of type 1.*

Now, we use the formulation of Linial and Tarsi to first show an auxiliary lemma and then combine the lemma with Theorem 3.9 to improve the upper bound of $m(n)$. The following lemma has been shown by Varma and Raghavendra in [155], however the proof was rather long. We straightforwardly extend Theorem 3.1 in [110] to obtain a much shorter proof.

Lemma 3.14. *If $m(k) = 2k - 1$ for a fixed $k \in \mathbb{N}$, then $(SE_n)^{3n-k-1}$ is rearrangeable whenever $n \geq k$.*

Proof. The assertion in the lemma is equivalent to the fact that if we know $m(k) = 2k - 1$ for some fixed $k \in \mathbb{N}$, then for every two $N \times n$ balanced matrices $A = [a_1, \ldots, a_n]$ and $B = [b_1, \ldots, b_n]$, there exists an $N \times (2n - k - 1)$ balanced matrix M such that the matrix $[A, M, B]$ is balanced. Here a_i and b_i are the i^{th} columns of A and B respectively. We shall construct the $(2n - k - 1)$ column vectors of M. The construction takes several steps as follows.

Step 1 Repeatedly apply Lemma 3.3 to construct vectors $\{u_1, \ldots, u_{n-k}\}$ such that for $i = 1, \ldots, n-k$, u_i agrees with $[a_{i+1}, \ldots, a_n, u_1, \ldots, u_{i-1}]$ and $[u_{i-1}, \ldots, u_1, b_n, \ldots, b_{i+1}]$. Let $U = [u_1, \ldots, u_{n-k}]$ and $U^R = [u_{n-k}, \ldots, u_1]$, then after this step both $[A, U]$ and $[U^R, B]$ are balanced.

Step 2 We want to construct vectors $x_1, \ldots, x_{k-1}$ such that if we let $X = [x_1, \ldots, x_{k-1}]$, then $[A, U, X]$ and $[X, U^R, B]$ are both balanced. Notice that as U is an $N \times (n-k)$ balanced matrix, each row of U occurs exactly 2^k times, and so do the rows of U^R in the same positions. Hence, the rows of U and U^R can be partitioned into 2^{n-k} classes of 2^k identical row vectors in each partition. For v be any column of U or U^R, let $v^{(i)}$ be the sub-vector of v with entries in the i^{th} partition, where $0 \le i \le 2^{n-k} - 1$. Notice that $v^{(i)} \in \mathbb{F}_2^k$ for each i. Also, for each $i = 0, \ldots, 2^{n-k} - 1$, let

$$A^{(i)} = [a^{(i)}_{n-k+1}, \ldots, a^{(i)}_n]$$

and

$$B^{(i)} = [b^{(i)}_n, \ldots, b^{(i)}_{n-k+1}]$$

Then, since Beneš conjecture is true for k (i.e. $m(k) = 2k - 1$), there exist vectors $x_1^{(i)}, \ldots, x_{k-1}^{(i)}$ such that $[A^{(i)}, X^{(i)}, B^{(i)}]$ is balanced. The vectors $x_1, \ldots, x_{k-1}$ are obtained by pasting together the $x_j^{(i)}$ preserving the positions of the partitions.

After this step, $[A, U, X]$ is balanced because at the positions where the rows of U are identical we have $[A^{(i)}, X^{(i)}]$ being a $2^k \times k$ balanced matrix. The fact that $[X, U^R, B]$ is balanced follows similarly.

Step 3 Now we define an $N \times (n-k)$ matrix W from U such that $[A, W, X, U^R, B]$ is balanced. Define W as follows (all arithmetics are done over $\mathbb{F}_2$).

$$w_i = \begin{cases} u_i & 1 \le i \le \lfloor \frac{n-k}{2} \rfloor \\ u_i + u_{n-k-i} & \lfloor \frac{n-k}{2} \rfloor + 1 \le i \le n-k-1 \\ u_{n-k} + a_n & i = n-k \end{cases} \tag{3.1}$$

We are left to show that $[A, W, X, U^R, B]$ is balanced. The balancedness of $[X, U^R, B]$ has already been established, so we only need to show that $[A, W, X, U^R]$ is balanced. We do this by considering the following types of submatrices:

(a) Submatrices of the form $[a_i, \ldots, a_n, w_1, \ldots, w_{i-1}]$ where $2 \leq i \leq n-k+1$. We apply Lemma 3.5 and use the fact that $[a_i, \ldots, a_n, u_1, \ldots, u_{i-1}]$ is balanced. $[a_i, \ldots, a_n, w_1, \ldots, w_{i-1}]$ can be obtained from $[a_i, \ldots, a_n, u_1, \ldots, u_{i-1}]$ by an invertible linear transformation with the invert map preserves the a_j ($i \leq j \leq n$) and

$$u_j = \begin{cases} w_j & 1 \leq j \leq \lfloor \frac{n-k}{2} \rfloor \\ w_j + w_{n-j-k} & \lfloor \frac{n-k}{2} \rfloor + 1 \leq j \leq n-k-1 \\ w_{n-k} + a_n & j = n-k \end{cases} \tag{3.2}$$

(b) Submatrices of the form $[a_i, \ldots, a_n, w_1, \ldots, w_{n-k}, x_1, \ldots, x_{k+i-n-1}]$ where $n-k+2 \leq i \leq n$. Similarly, in this case we use Lemma 3.5 and the balancedness of the matrix $[a_i, \ldots, a_n, u_1, \ldots, u_{n-k}, x_1, \ldots, x_{k+i-n-1}]$. In this case, the a_j and x_j are preserved, while equations (3.1) and (3.2) are used to transform $u_1, \ldots, u_{n-k}$ to $w_1, \ldots, w_{n-k}$ and vice versa.

(c) Submatrices of the form $P = [w_i, \ldots, w_{n-k}, x_1, \ldots, x_{k-1}, u_{n-k}, \ldots, u_{n-k-i+1}]$ where $1 \leq i \leq n-k$. Here we use the fact that $Q = [a_n, u_1, \ldots, u_{n-k}, x_1, \ldots, x_{k-1}]$ is balanced. To get P from Q, we apply (3.1). To get Q from P, we fix all vectors x_j, and $u_{n-k-i+1}, \ldots, u_{n-k}$. Moreover, a_n and u_j ($1 \leq j \leq n-k-i$) is obtained by

$$\begin{aligned}
a_n &= u_{n-k} + w_{n-k} \\
u_j &= w_j + w_{n-k-j} \text{ for } i \leq j \leq \lfloor \tfrac{n-k}{2} \rfloor \leq n-k-i \\
u_j &= w_j \text{ for } i \leq \lfloor \tfrac{n-k}{2} \rfloor + 1 \leq j \leq n-k-i \\
u_j &= w_{n-k-j} + u_{n-k-j} \text{ when } j \leq \min(i-1, n-k-i)
\end{aligned}$$

□

Theorem 3.15. *For $n \in \mathbb{N}$ and $n \geq 4$, a SE network with $3n-5$ stages is rearrangeable.*

Proof. This is immediate from Theorem 3.9 and Lemma 3.14. □

3.4 Proof of Lemma 3.12

In this section we shall present a proof of Lemma 3.12 and also set up most of the basic techniques and notations needed for the proof of Lemma 3.13, which is more complicated. In the proof of this Lemma and the next, we assume that G_{22} is of type-2. Without loss of generality, we also assume that the 2-cycles of G_{12} go horizontally, i.e. one goes from 0123 to 0123, and the other from 4567 to 4567, applying operation 4 of Proposition 3.8 if necessary.

The basic idea behind the proofs of these Lemmas is to start from a coloring of G_{11}, G_{12}, and G_{21} using 6 colors $\{0, 4, 2, 6, 1, 5\}$ which does not violate P_1 and P_i' ($i = 1, 2, 3$), and then modify this coloring so that none of the properties in Proposition 3.7 is violated. The Lemmas then hold as a consequence of Lemma 3.10.

In the course of modifying the original coloring, we shall need 5 basic color transformations: $UU(c_1, c_2)$, $UL(c_1, c_2)$, $LU(c_1, c_2)$, $LL(c_1, c_2)$, and $A(c_1, c_2)$, where c_1 and c_2

are two distinct colors in C. Transformation $UU(c_1, c_2)$ (for Upper-Upper) switches colors of a c_1-edge e_1 and a c_2-edge e_2 where $l(e_i) \in \{0, 1, 2, 3\}$, and $r(e_i) \in \{0, 1, 2, 3\}$, $i = 1, 2$. $UL(c_1, c_2)$ (for Upper-Lower) switches colors of a c_1-edge e_1 and a c_2-edge e_2 where $l(e_i) \in \{0, 1, 2, 3\}$, and $r(e_i) \in \{4, 5, 6, 7\}$, $i = 1, 2$. LU and LL are defined similarly in the obvious way. $A(c_1, c_2)$ (for All) changes color of all c_1-edges to c_2 and vice versa. We will see that the transformations UU, UL, LU, LL are well defined from the associated context where they are applied.

We now need to introduce a concept called the *color incidence vectors* (or CIV for short) associated with a coloring of G. To each subset of vertices $\{0, 1, 2, 3\}$ and $\{4, 5, 6, 7\}$ of either I or O, we associate a CIV of 4 components, where the component corresponding to vertex i consists of 2 colors of the edges incident to i. For example, Figure 3.11 shows a coloring of a graph G and the four associated CIVs. Note that the coloring shown is a **valid** one. It is immaterial if the CIVs are row vectors or column vectors, so we will adopt the

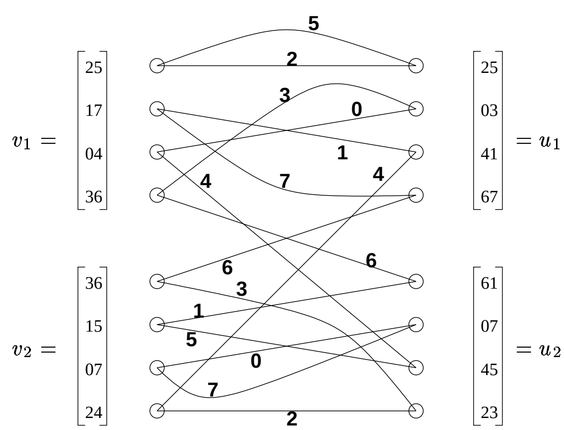


Figure 3.11: An example of color incidence vectors

convention that in the figures we use column vectors, and in the texts we use row vectors. Moreover, the order of the color pair in each component of a CIV is not important. In the figure, we have specifically chosen the order so that it is easier to see the validity of the coloring.

Corresponding to a coloring of G, there are 4 CIVs. We shall use v_1 and v_2 to denote the CIVs corresponding to the subsets $\{0,1,2,3\}$ and $\{4,5,6,7\}$ of I, respectively. Similarly, u_1 and u_2 are the CIVs corresponding to the subsets $\{0,1,2,3\}$ and $\{4,5,6,7\}$ of O, respectively. The following definitions relate the CIVs to the conditions P_i and P_i'.

Definition 3.16. Let v be a CIV corresponding to a coloring of G, then

- v is said to *respect* P_1 or P_1' if the components of v contain all 8 colors $\{0,\ldots,7\}$.
- Let A be the set of 4 colors in the first two components of v, and B be the set of 4 colors in the last two components of v. Then, v is said to *respect* P_2 if each pair $\{c_1,c_2\} \in \{\{0,1\},\{2,3\},\{4,5\},\{6,7\}\}$ has a representative from each of A and B. Similarly, v is said to *respect* P_2' if each pair $\{c_1,c_2\} \in \{\{0,4\},\{2,6\},\{1,5\},\{3,7\}\}$ has a representative from each of A and B.
- v is said to *respect* P_3 if each component of v has a representative from each of $\{0,1,2,3\}$ and $\{4,5,6,7\}$. Similarly, v respects P_3' if each component of v has a representative from each of $\{0,4,2,6\}$ and $\{1,5,3,7\}$.

The following Proposition is fairly straightforward, thus we omit the proof.

Proposition 3.17. *Let v_1, v_2, u_1, u_2 be the CIVs corresponding to some coloring of G as defined above. Then, the coloring is valid if and only if*

(i) v_1 *and* v_2 *respect* P_i *(*$i = 1,2,3$*).*

(ii) u_1 *and* u_2 *respect* P_i' *(*$i = 1,2,3$*).*

To show Lemma 3.12, we consider two cases as follows.

Case 1. *The two type-*1 *components are* G_{i1} *and* G_{i2} *for some* $i \in \{1, 2\}$. Without loss of generality, we assume G_{11} and G_{12} are of type-1. We start from the coloring shown in Figure 3.12, which clearly does not violate P_1 and P_i' ($i = 1, 2, 3$). Moreover, applying both $A(0, 6)$ and $A(2, 4)$ would yield another coloring where P_1 and P_i' are not violated.

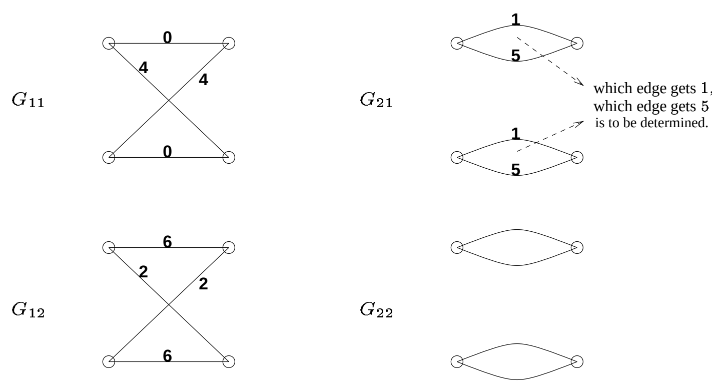


Figure 3.12: Case 1 of Lemma 3.12

After the tentative coloring introduced in Figure 3.12, and after applying a sequence of the first 3 operations in Proposition 3.8, the possible CIVs are the union of two sets:

$$F_1 = \left\{ \begin{bmatrix} \bullet 0 \\ \circ 2 \\ \bullet 4 \\ \circ 6 \end{bmatrix}, \begin{bmatrix} \bullet 0 \\ \circ 2 \\ \circ 4 \\ \bullet 6 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ \bullet 2 \\ \bullet 4 \\ \circ 6 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ \bullet 2 \\ \circ 4 \\ \bullet 6 \end{bmatrix} \right\} \quad F_2 = \left\{ \begin{bmatrix} \bullet 0 \\ \circ 6 \\ \bullet 2 \\ \circ 4 \end{bmatrix}, \begin{bmatrix} \bullet 0 \\ \circ 6 \\ \circ 2 \\ \bullet 4 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ \bullet 6 \\ \bullet 2 \\ \circ 4 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ \bullet 6 \\ \circ 2 \\ \bullet 4 \end{bmatrix} \right\}$$

Here, the $\bullet$'s stand for the colors of the edges of a 2-cycle of G_{21}, which are to be determined. In each CIV, one $\bullet$ gets colored 1 and the other gets 5. Obviously, the choice of 5 and 1 on one CIV will affect another. Specifically, the choice of 5 and 1 on one of the CIVs v_i will affect one of the u_j, but not the other v_i. Similarly, the $\circ$'s correspond to colors of

G_{22}, which we are not concerned about, as we shall apply Lemma 3.10. Strictly speaking, each F_i consists of four classes of CIVs. However, we shall not differentiate between individual CIV and its class as this is immaterial.

We shall also extend the notion of a condition P_i or P_i' being respected by a CIV to a class of CIV. A class of CIV *respects* P_i or P_i' if the corresponding condition in Definition 3.16 is not violated yet. In this sense, each class of CIVs in $F_1 \cup F_2$ respects all P_i and P_i'. Notice that no matter how we assign 1 and 5 in each 2-cycle of G_{21}, the resulting CIVs in $F_1 \cup F_2$ (u_1 and u_2 in particular) still respect P_i'. Moreover, u_1 and u_2 keep respecting P_i' even after we apply $A(2,6)$ or $A(0,4)$.

If both v_1 and v_2 belong to F_1, then in both v_i we choose the $\bullet$ that goes with 0 or 2 to be 5 and the other $\bullet$ to be 1. Clearly, the P_i are respected by v_1 and v_2.

If both v_1 and v_2 belong to F_2, then we apply $A(2,6)$ which move v_1 and v_2 to F_1 again.

Hence, we could now assume without loss of generality that $v_1 \in F_1$ and $v_2 \in F_2$, flipping G horizontally if necessary (operation 5 of Proposition 3.8). Furthermore, due to operation 6 of flipping G vertically, we can also assume that u_1 and u_2 don't belong to the same F_i.

As the $\bullet$'s colors in v_1 can be chosen easily so that v_1 respects P_i ($i = 1,2,3$), we try to modify the coloring so that v_2 does, too. If $v_2 = \begin{bmatrix}\bullet 0 & \circ 6 & \circ 2 & \bullet 4\end{bmatrix}$, we can pick colors so that $v_2 = \begin{bmatrix}\mathbf{5}0 & \circ 6 & \circ 2 & \mathbf{1}4\end{bmatrix}$. When $v_2 = \begin{bmatrix}\circ 0 & \bullet 6 & \bullet 2 & \circ 4\end{bmatrix}$, we apply $A(0,6)$ and $A(2,4)$ to make $v_2 = \begin{bmatrix}\bullet 0 & \circ 6 & \circ 2 & \bullet 4\end{bmatrix}$ and choose $\bullet$'s colors similarly. Note that the availability of the $\bullet$'s colors in v_1 are not affected.

If $v_2 = \begin{bmatrix}\bullet 0 & \circ 6 & \bullet 2 & \circ 4\end{bmatrix}$ and $u_1 \in F_1$, we apply $LU(2,4)$, making

$$v_2 = \begin{bmatrix}\bullet 0 & \circ 6 & \circ 2 & \bullet 4\end{bmatrix},$$

but keeping P_i respected by the v_j and P_i' respected by the u_j ($i = 1,2,3$, $j = 1,2$). The $\bullet$'s colors in v_2 could now be picked as before. When $v_2 = \begin{bmatrix} \bullet 0 & \circ 6 & \bullet 2 & \circ 4 \end{bmatrix}$ and $u_2 \in F_1$, we apply $LU(2,4)$, $UU(0,6)$, $UL(2,4)$, which turns v_2 into $\begin{bmatrix} \bullet 0 & \circ 6 & \circ 2 & \bullet 4 \end{bmatrix}$ but keeps v_1 in F_1, still.

Lastly, when $v_2 = \begin{bmatrix} \circ 0 & \bullet 6 & \circ 2 & \bullet 4 \end{bmatrix}$, we apply $A(0,6)$ and $A(2,4)$ and return to the previous case.

Case 2. *It is not the case that the two type-*1 *components are* G_{i1} *and* G_{i2} *for any* $i \in \{1,2\}$. Without loss of generality, we assume G_{11} and G_{21} are of type-1. We start from the coloring shown in Figure 3.13, which clearly does not violate P_1 and P_i' ($i = 1,2,3$).

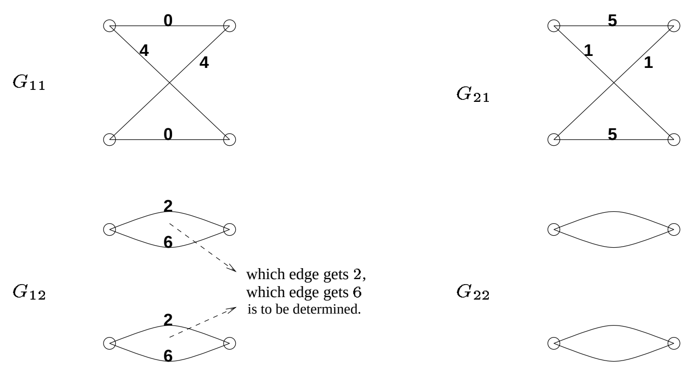


Figure 3.13: Case 2 of Lemma 3.12

Again, there are 8 possible CIVs divided into two classes:

$$F_1 = \left\{ \begin{bmatrix} 05 \\ \bullet\circ \\ 41 \\ \bullet\circ \end{bmatrix}, \begin{bmatrix} 05 \\ \bullet\circ \\ 4\circ \\ \bullet 1 \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 5 \\ 41 \\ \bullet\circ \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 5 \\ 4\circ \\ \bullet 1 \end{bmatrix} \right\} \quad F_2 = \left\{ \begin{bmatrix} 01 \\ \bullet\circ \\ 45 \\ \bullet\circ \end{bmatrix}, \begin{bmatrix} 01 \\ \bullet\circ \\ 4\circ \\ \bullet 5 \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 1 \\ 45 \\ \bullet\circ \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 1 \\ 4\circ \\ \bullet 5 \end{bmatrix} \right\}$$

Remark 3.18. A special property of F_1 is that for each vector $v \in F_1$, there is a *proper assignment* of colors 6 and 2 to the $\bullet$, i.e. one $\bullet$ gets 6, the other gets 2 such that v respect all P_i ($i = 1,2,3$). For each $v \in F_2$, a proper assignment exists if 5 and 1 get exchanged in

v. Moreover, for any vector $v \in F_1 \cup F_2$, if we exchange colors of one of the pairs $\{4, 1\}$ or $\{0, 5\}$, we could properly assign colors 6 and 2 to the $\bullet$'s of v to maintain P_i.

If both v_1 and v_2 are in F_1, then we properly assign 2 and 6 to the $\bullet$'s. While, if v_1 and v_2 are in F_2 we apply $A(5, 1)$ and pick $\bullet$'s colors in the same way.

Now, assume $v_1 \in F_1$ and $v_2 \in F_2$. Also, by flipping G vertically if necessary, we can assume that u_1 and u_2 are representatives from F_1 and F_2.

If $v_1 = \begin{bmatrix} 0\circ & \bullet 5 & 4\circ & \bullet 1 \end{bmatrix}$, then apply $A(5, 3)$ and $A(1, 7)$ so that $v_1 = \begin{bmatrix} 0\circ & \bullet \mathbf{3} & 4\circ & \bullet \mathbf{7} \end{bmatrix}$. In this case, F_2 becomes

$$F_2 = \left\{ \begin{bmatrix} 07 \\ \bullet\circ \\ 43 \\ \bullet\circ \end{bmatrix}, \begin{bmatrix} 07 \\ \bullet\circ \\ 4\circ \\ \bullet 3 \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 7 \\ 43 \\ \bullet\circ \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 7 \\ 4\circ \\ \bullet 3 \end{bmatrix} \right\}$$

Next, choose the $\bullet$'s colors in v_1 to turn it into $v_1 = \begin{bmatrix} 0\circ & \mathbf{6}3 & 4\circ & \mathbf{2}7 \end{bmatrix}$. While, in v_2 the $\bullet$ in the first or second component gets 2 and the other gets 6. We have used up 6 colors $\{0, 4, 2, 6, 3, 7\}$. Only 1 and 5 are left for the type-2 component G_{22}, hence Lemma 3.10 applies.

If $v_2 = \begin{bmatrix} 0\circ & \bullet 1 & 4\circ & \bullet 5 \end{bmatrix}$, then apply $A(5, 7)$ and $A(1, 3)$. The rest is similar.

Consequently, we only need to consider $v_1 \in A_1 := F_1 - \left\{ \begin{bmatrix} 0\circ & \bullet 5 & 4\circ & \bullet 1 \end{bmatrix} \right\}$, and $v_2 \in A_2 := F_2 - \left\{ \begin{bmatrix} 0\circ & \bullet 1 & 4\circ & \bullet 5 \end{bmatrix} \right\}$. u_1 and u_2 can be assumed to be representatives of A_1 and A_2 for the same reason as before.

If $v_1 = \begin{bmatrix} 05 & \bullet\circ & 41 & \bullet\circ \end{bmatrix}$, then we flip G vertically so that $u_1 = \begin{bmatrix} 05 & \bullet\circ & 41 & \bullet\circ \end{bmatrix}$, and that v_1 and v_2 are now representatives of A_1 and A_2. Apply $UU(0, 5)$ and $LU(1, 4)$, then P_i' are still respected by v_1 and v_2. Moreover, 0 and 5 get exchanged in v_1, while 1 and 4 get switched in v_2. Proper assignments of 2 and 6 now exist for v_1 and v_2 by Remark 3.18.

Similarly, when $v_2 = \begin{bmatrix} 01 & \bullet\circ & 45 & \bullet\circ \end{bmatrix}$ we apply $LL(0,5)$ and $UL(1,4)$ and proceed in the same manner.

Hence, we are left to consider the case where

$$v_1 \in B_1 := \left\{ \begin{bmatrix} 05 \\ \bullet\circ \\ 4\circ \\ \bullet 1 \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 5 \\ 41 \\ \bullet\circ \end{bmatrix} \right\} \quad v_2 \in B_2 := \left\{ \begin{bmatrix} 01 \\ \bullet\circ \\ 4\circ \\ \bullet 5 \end{bmatrix}, \begin{bmatrix} 0\circ \\ \bullet 1 \\ 45 \\ \bullet\circ \end{bmatrix} \right\}$$

and, u_1 and u_2 are representatives of B_1 and B_2.

If $v_1 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix}$ and $u_1 \in B_2$, $u_2 \in B_1$, then we flip G vertically, apply $UU(0,5)$ and Remark 3.18. While, if v_1 is the same but $u_1 = \begin{bmatrix} 0\circ & \bullet 5 & 41 & \bullet\circ \end{bmatrix} \in B_1$ and $u_2 \in B_2$, then we apply $LU(1,4)$ and Remark 3.18.

When $v_1 = \begin{bmatrix} 0\circ & \bullet 5 & 41 & \bullet\circ \end{bmatrix}$ and $u_1 \in B_1$, $u_2 \in B_2$, then we flip G vertically, apply $LU(1,4)$ and Remark 3.18. While, when v_1 is the same but $u_2 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix} \in B_1$ and $u_1 \in B_2$, then we apply $LL(0,5)$ and Remark 3.18.

Therefore, there are 8 cases left, 4 of which are when $v_1 = u_1 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix}$, and $v_2, u_2 \in B_2$. The other 4 cases are when $v_1 = u_2 = \begin{bmatrix} 0\circ & \bullet 5 & 41 & \bullet\circ \end{bmatrix}$, and $v_2, u_1 \in B_2$. We shall consider these 8 cases in turn as follows.

(2a) $v_1 = u_1 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix}$, and $v_2 = u_2 = \begin{bmatrix} 01 & \bullet\circ & 4\circ & \bullet 5 \end{bmatrix}$.

This case introduces a new technique which will be used in later cases and the proof of Lemma 3.13. Let us first take a look at Figure 3.14. The graph shown represents all graphs considered in this case. The two dashed edges at the upper half is a 2-cycle

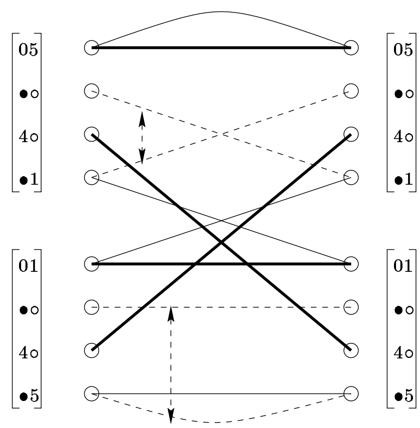


Figure 3.14: Representative figure for case 2a of Lemma 3.12

of G_{12}, and the other two dashed edges are from the other 2-cycle. The end points of the dashed edges go to the ●'s, but we don't know which edge goes to which ●.

Figure 3.15 presents our solution to this case. The figure shows a "proof without

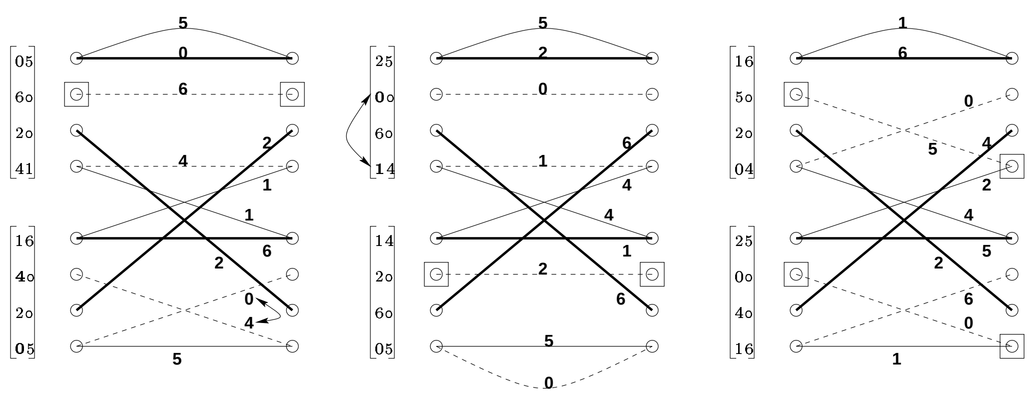


Figure 3.15: Case 2a of Lemma 3.12

words". Basically, we consider 3 sub-cases represented sequentially by 3 drawings from left to right.

In the first sub-case, we fix the end points (in the square boxes) of a dashed edge in the upper half, letting all other dashed edges go freely. The second dashed edge in the upper half has only one choice to go, of course. The doubly headed arrow specifies that we could exchange colors 0 and 4 of those two dashed edges so that v_2 looks as shown. It is straightforward to check that in this sub-case v_1 and v_2 respect the P_i and u_1 and u_2 respect the P_i'.

In the second sub-case, we fix the end points of a dashed edge in the lower half. Assign colors to all edges as shown. The doubly headed arrow on the side means that no matter which dashed edge goes to which $\bullet$, we still have v_1 respecting the P_i.

The last drawing considers the only sub-case left. The figure is self-explaining.

(2b) $v_1 = u_1 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix}$, $v_2 = \begin{bmatrix} 01 & \bullet\circ & 4\circ & \bullet 5 \end{bmatrix}$, and $u_2 = \begin{bmatrix} 0\circ & \bullet 1 & 45 & \bullet\circ \end{bmatrix}$.

Figure 3.16 presents our solution to this case.

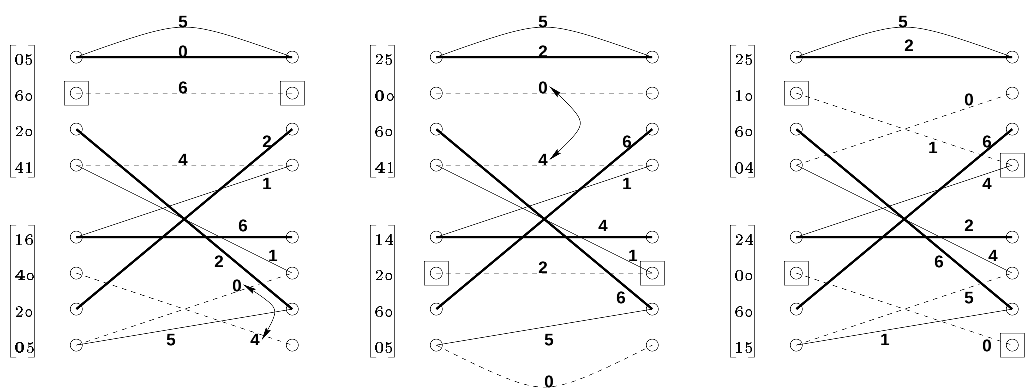


Figure 3.16: Case 2b of Lemma 3.12

(2c) $v_1 = u_1 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix}$, $v_2 = \begin{bmatrix} 0\circ & \bullet 1 & 45 & \bullet\circ \end{bmatrix}$, and $u_2 = \begin{bmatrix} 01 & \bullet\circ & 4\circ & \bullet 5 \end{bmatrix}$.

Figure 3.17 presents our solution to this case.

(2d) $v_1 = u_1 = \begin{bmatrix} 05 & \bullet\circ & 4\circ & \bullet 1 \end{bmatrix}$, $v_2 = u_2 = \begin{bmatrix} 0\circ & \bullet 1 & 45 & \bullet\circ \end{bmatrix}$.

Figure 3.18 presents our solution to this case.

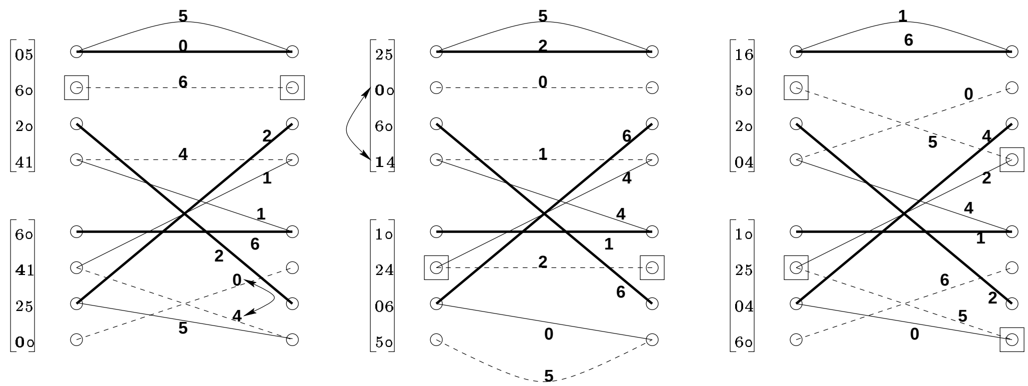


Figure 3.17: Case 2c of Lemma 3.12

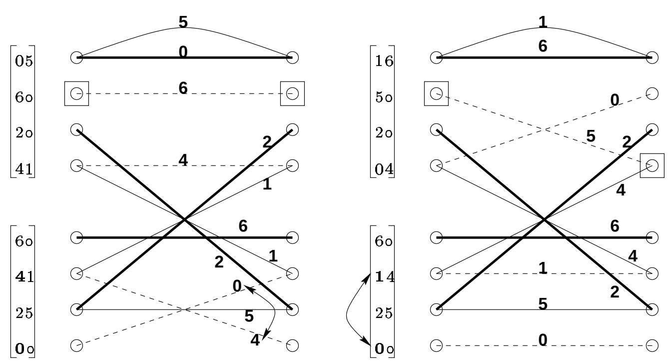


Figure 3.18: Case 2d of Lemma 3.12

Before proceeding to the proof of case (2e), we need another proposition, which is very useful later on in the proof of Lemma 3.13.

Proposition 3.19. *Let G be our graph to be colored as usual. For any $j \in \{1,2\}$, let $\bar{j}$ be the element in $\{1,2\} - \{j\}$. Assume the following hold:*

(i) G_{22} is a type-2 component of G with the two 2-cycles going horizontally, i.e. one 2-cycle of G_{22} goes from v_1 to u_1 and the other goes from v_2 to u_2.

(ii) There is a partial coloring of G, in which all edges of G_{11}, G_{12}, and G_{21} are colored using twice each color in $\{0,4,2,6,1,5\}$.

(iii) (The partially colored) u_1 and u_2 respect the P_i' ($i = 1,2,3$.)

(iv) (The partially colored) $v_{\bar{j}}$ respects the P_i ($i = 1,2,3$.)

(v) v_j respects P_1 and P_2 but does not respect P_3 because one component of v_j contains some $c \in \{2,6\}$ and another color c', but c and c' are not representatives of $\{0,1,2,3\}$ and $\{4,5,6,7\}$.

(vi) The edges get color c go horizontally from v_j to u_j and $v_{\bar{j}}$ to $u_{\bar{j}}$. Moreover, c goes with $\circ$ in a component of u_j, i.e. the edge e that gets colored c has the same right end point as another edge e' in the 2-cycle that goes from v_j to u_j.

Then, G can be properly colored.

The last drawing in Figure 3.19 is an example of such a situation. In this drawing, $j = 2$, $c = 2$ and $c' = 1$.

Proof. Firstly, assume $c = 2$. Clearly, just as in the proof of Lemma 3.10, we can assign 3 and 7 to edges of the 2-cycle going from $v_{\bar{j}}$ to $u_{\bar{j}}$ so that $v_{\bar{j}}$ respects the P_i and $u_{\bar{j}}$ respects the P'_i. Suppose

$$v_j = \begin{bmatrix} cc' & c_1\circ & c_2c_3 & c_4\circ \end{bmatrix}$$

where $\{c, c', c_1, c_2, c_3, c_4\} = \{0, 4, 2, 6, 1, 5\} = \{0, 1, 2, 4, 5, 6\}$ because v_j respects P_1. As cc' is the only component that does not respect P_3 (i.e. $c = 2$ and c' are in $\{0, 1, 2, 3\}$), c_2c_3 are representatives of $\{0, 1, 2, 3\}$ and $\{4, 5, 6, 7\}$. Thus, $c_1, c_4 \in \{4, 5, 6\}$. In fact, as v_j respects P_2, $c_1 \in \{4, 5\}$, $c' \in \{0, 1\}$ and thus $6 \in \{c_2, c_3, c_4\}$.

Now, assign 7 to e' and 3 to the other edge of the 2-cycle that goes from v_j to u_j. We claim that after exchanging colors (7 and 2) of e and e', we have a valid coloring of G. Indeed, the CIVs $v_{\bar{j}}$ and $u_{\bar{j}}$ are not affected. u_j respects P'_i after the first assignment of 7 and 3, and keeps respecting P'_i as we have just exchanged colors of edges having the same right end point. We only need to be concerned about v_j after this exchange. There are two cases for v_j before the exchange:

$$v_j = \begin{bmatrix} \mathbf{2}c' & c_1\mathbf{7} & c_2c_3 & c_4\mathbf{3} \end{bmatrix}$$

or

$$v_j = \begin{bmatrix} \mathbf{2}c' & c_1\mathbf{3} & c_2c_3 & c_4\mathbf{7} \end{bmatrix}$$

The reader can easily check that v_j does respect P_i after the exchange of 2 and 7.

The case where $c = 6$ is done similarly. The only difference is that e gets 3 this time. □

Remark 3.20. This proposition will prove to be very useful in the proof of Lemma 3.13. It could be stated in a much more general fashion, however we did not do so because we will need only this instance of the proposition, and because the general statement would be too notationally heavy, thus hard to grasp.

(2e) $v_1 = u_2 = \begin{bmatrix}0\circ & \bullet 5 & 41 & \bullet\circ\end{bmatrix}$, and $v_2 = u_1 = \begin{bmatrix}01 & \bullet\circ & 4\circ & \bullet 5\end{bmatrix}$.

Figure 3.19 presents the *partial* solution to this case. The reason this case was only partially solved is due to the last drawing, i.e. the last sub-case, in which there is a violation of P_3 in v_2. To resolve this violation, we reason as follows. If the 2-cycles

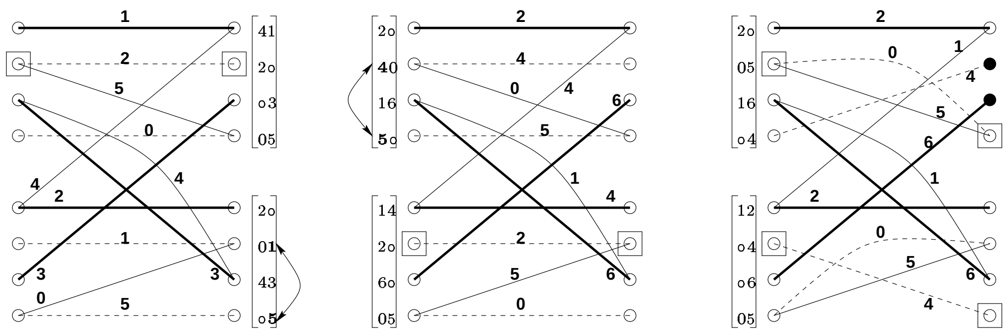


Figure 3.19: Case 2e of Lemma 3.12

of G_{22} go horizontally, we apply Proposition 3.19. Otherwise, the 2-cycle going from v_1 to u_2 could be colored with 3 and 7 properly, as in Lemma 3.10. The other 2-cycle has edges going from the blacken vertices at u_1 to the $\circ$ of v_2. We assign 7 to the edge whose right end point is the second component of u_1 and 3 to the other edge.

There are two cases after this assignment, depending on what v_2 ends up being. If v_2 ends up to be $v_2 = \begin{bmatrix}12 & \mathbf{7}4 & \mathbf{3}6 & 05\end{bmatrix}$, then we apply $LU(1,7)$. Obviously, u_1 still respects the P_i', and v_2 respects all the P_i now. If v_2 becomes $v_2 = \begin{bmatrix}12 & \mathbf{3}4 & \mathbf{7}6 & 05\end{bmatrix}$, then we apply $LU(1,7)$ and $LL(0,2)$ to get the same result. In these two situations, the relative positions of 4 and 5 are not important.

Lastly, note that we have used 6 colors $\{0,1,2,3,4,5\}$ in the first sub-case, leaving $\{6,7\}$. Lemma 3.10 still applies.

(2f,2g,2h) When $v_1 = u_2 = \begin{bmatrix}0\circ & \bullet 5 & 41 & \bullet\circ\end{bmatrix}$, we have three more cases, which are simple and close enough to be considered at once: (2f) $v_2 = \begin{bmatrix} 01 & \bullet\circ & 4\circ & \bullet 5 \end{bmatrix}$, and $u_1 = \begin{bmatrix} 0\circ & \bullet 1 & 45 & \bullet\circ \end{bmatrix}$; (2g) $v_2 = \begin{bmatrix} 0\circ & \bullet 1 & 45 & \bullet\circ \end{bmatrix}$, and $u_1 = \begin{bmatrix} 01 & \bullet\circ & 4\circ & \bullet 5 \end{bmatrix}$; and (2h) $v_1 = u_2 = \begin{bmatrix} 0\circ & \bullet 5 & 41 & \bullet\circ \end{bmatrix}$, $v_2 = u_1 = \begin{bmatrix} 0\circ & \bullet 1 & 45 & \bullet\circ \end{bmatrix}$.

Figure 3.20 presents the solutions all three cases.

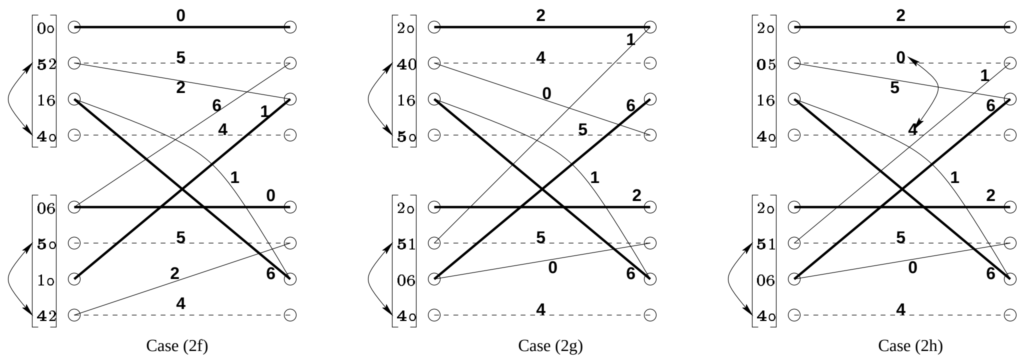


Figure 3.20: Case 2f, 2g and 2h of Lemma 3.12

3.5 Proof of Lemma 3.13

We assume that the 2-cycles of G_{22} go horizontally, i.e. one goes from 0123 to 0123, and the other from 4567 to 4567, applying operation 4 of Proposition 3.8 if necessary. We start from the coloring shown in Figure 3.21, whose CIV v_j respect P_1 and u_j respect P_i' ($i = 1, 2, 3$, $j = 1, 2$). Moreover, applying both $A(0, 2)$ and $A(4, 6)$ would yield another coloring where P_1, and P_i' are not violated. So does applying any of $A(0, 4)$, $A(1, 5)$, or $A(2, 6)$.

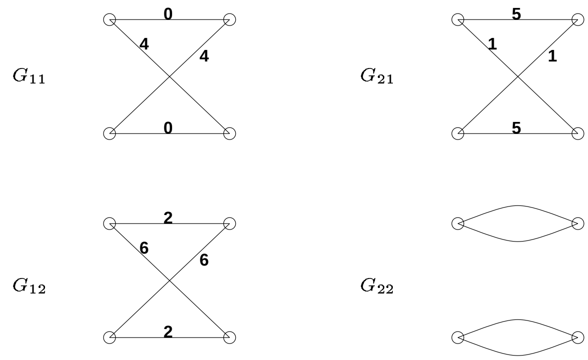


Figure 3.21: The initial coloring for Lemma 3.13

The possible CIVs are the union of four sets:

$$F_1 = \left\{ \begin{bmatrix} 50 \\ \circ 2 \\ 14 \\ \circ 6 \end{bmatrix}, \begin{bmatrix} 50 \\ \circ 2 \\ \circ 4 \\ 16 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 52 \\ 14 \\ \circ 6 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 52 \\ \circ 4 \\ 16 \end{bmatrix} \right\} \quad F_2 = \left\{ \begin{bmatrix} 10 \\ \circ 2 \\ 54 \\ \circ 6 \end{bmatrix}, \begin{bmatrix} 10 \\ \circ 2 \\ \circ 4 \\ 56 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 12 \\ 54 \\ \circ 6 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 12 \\ \circ 4 \\ 56 \end{bmatrix} \right\}$$

$$F_3 = \left\{ \begin{bmatrix} 50 \\ \circ 6 \\ 14 \\ \circ 2 \end{bmatrix}, \begin{bmatrix} 50 \\ \circ 6 \\ \circ 4 \\ 12 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 56 \\ 14 \\ \circ 2 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 56 \\ \circ 4 \\ 12 \end{bmatrix} \right\} \quad F_4 = \left\{ \begin{bmatrix} 10 \\ \circ 6 \\ 54 \\ \circ 2 \end{bmatrix}, \begin{bmatrix} 10 \\ \circ 6 \\ \circ 4 \\ 52 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 16 \\ 54 \\ \circ 2 \end{bmatrix}, \begin{bmatrix} \circ 0 \\ 16 \\ \circ 4 \\ 52 \end{bmatrix} \right\}$$

Notice that for all $i = 1, 2, 3$, the vectors in F_1 respect P_i, F_2 respect P_i after applying $A(1, 5)$, F_3 respect P_i after applying $A(2, 6)$, and F_4 respect P_i after applying $A(0, 4)$. Hence, if v_1 and v_2 belong to the same F_i then we are done. We could thus also assume u_1 and u_2 are from different F_i's. Consider 4 cases.

Case 1. $v_1 \in F_1$ *and* $v_2 \in F_3 \cup F_4$. If $v_2 \in F_4$, applying $A(0, 2)$ and $A(4, 6)$ would keep v_1 in F_1, while move v_2 to F_3. Hence, without loss of generality we can assume $v_1 \in F_1$ and $v_2 \in F_3$. Consider 4 sub-cases as follows, which are ordered in increasing in level of complexity.

(1a) $v_2 = \begin{bmatrix} 50 & \circ 6 & 14 & \circ 2 \end{bmatrix}$. In this case, v_2 and v_1 already respect the P_i.

(1b) $v_2 = \begin{bmatrix} \circ 0 & 56 & 14 & \circ 2 \end{bmatrix}$. In this case, we can not apply Lemma 3.10 directly, but have to go further. There is no problem with picking colors 3 and 7 for the 2 cycle of G_{22} connecting v_1 to u_1. For the other 2-cycle, we pick colors and do certain transformation as follows. Notice that no matter what u_2 is, the 5 in u_2 is not in the same component as a $\circ$. There is exactly an edge e of this 2-cycle where $r(e)$ and 5 are both in the first two components or the last two components of u_2. Assign 3 to e and 7 to the other edge. There are two cases depending on how v_2 ends up to be after this assignment.

If v_2 becomes $v_2 = \begin{bmatrix} \mathbf{3}0 & 56 & 14 & \mathbf{7}2 \end{bmatrix}$, then we apply $LL(5,3)$, which keeps u_2 respecting the P_i' and makes v_2 respect the P_i.

If v_2 becomes $v_2 = \begin{bmatrix} \mathbf{7}0 & 56 & 14 & \mathbf{3}2 \end{bmatrix}$, then we apply $LL(5,3)$, $LU(4,6)$ and $UU(0,2)$. Effectively, this transformation makes $v_2 = \begin{bmatrix} 70 & 34 & 16 & 52 \end{bmatrix}$, respecting the P_i, exchanges each pair $\{0,2\}$ and $\{4,6\}$ in u_1, and exchanges $\{0,2\}$ in v_1. Hence, after the transformation u_1 still respects the P_i', and v_1 is still in F_1, respecting the P_i.

(1c) $v_2 = \begin{bmatrix} 50 & \circ 6 & \circ 4 & 12 \end{bmatrix}$.

If u_2 has a $\circ 2$ as a component (we will write $\circ 2 \in u_2$ for short), then apply Proposition 3.19. If $52 \in u_2$ and $\circ 0 \in u_2$, then we apply $LL(5,2)$, then $A(0,2)$ and $A(4,6)$, so that $v_2 = \begin{bmatrix} 20 & \circ 4 & \circ 6 & 51 \end{bmatrix}$. Proposition 3.19 could be applied now with $c = 2$.

If $14 \in u_1$, then apply $LU(1,4)$. If $16 \in u_1$, then apply $LU(1,6)$, $LL(0,2)$, and

$UL(4,6)$, turning v_2 into $\left[52 \quad \circ 1 \quad \circ 4 \quad 06\right]$, while keeping all other CIVs respecting their corresponding conditions.

Hence, we are left to consider the case where $\circ 2 \notin u_2$, not both 52 and $\circ 0$ are in u_2, and 14 and 16 are not in u_1.

As we have seen, the CIVs are useful in classifying how each subset of vertices $\{0,1,2,3\}$ and $\{4,5,6,7\}$ of I and O are connected to the others. There is another way to classify the "shape" of these connecting patterns. Ignoring the edges of G_{22}, each subset $\{0,1,2,3\}$ and $\{4,5,6,7\}$ of I and O are connected to three horizontal edges (i.e. edges connecting v_j and u_j) and three diagonal edges (i.e. edges connecting v_j and $u_{\bar{j}}$). The connecting patterns to each of these vertex subsets can be classified based on the relative end points of the horizontal and diagonal edges, up to the application of the first 3 operations of Proposition 3.8. Figure 3.22 shows all possible *shapes* of u_1. In the figure, the thicken edges represent edges that go from u_1 to

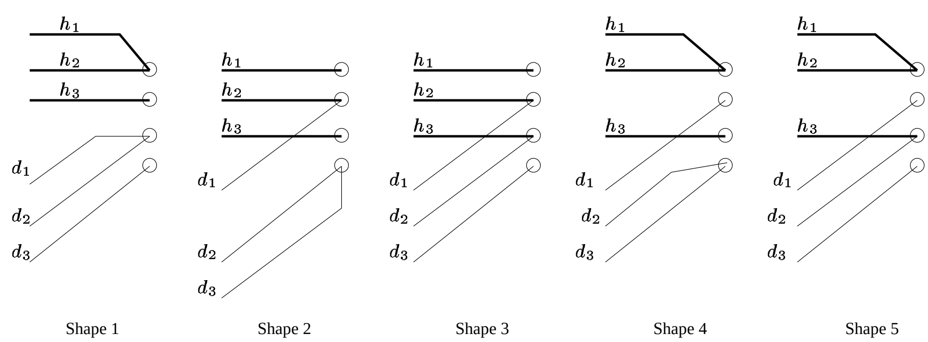


Figure 3.22: All possible shapes of u_1

v_1 (horizontally). The other three edges go from u_1 to v_2 (diagonally). It is straightforward to check that there are only 5 possible shapes as shown, up to applying the first three operations of Proposition 3.8. For example, if $u_1 = \left[50 \quad \circ 2 \quad \circ 4 \quad 16\right]$, then it is of shape 1. While, $\begin{bmatrix} \circ 0 & 12 & 54 & \circ 6 \end{bmatrix}$ is of shape 3, and $\begin{bmatrix} \circ 0 & 16 & 54 & \circ 2 \end{bmatrix}$ is of shape 2. Similarly, Figure 3.23 shows all possible shape of u_2. These are just mirror images of the shapes of u_1. We have tentatively label the edges in the shapes

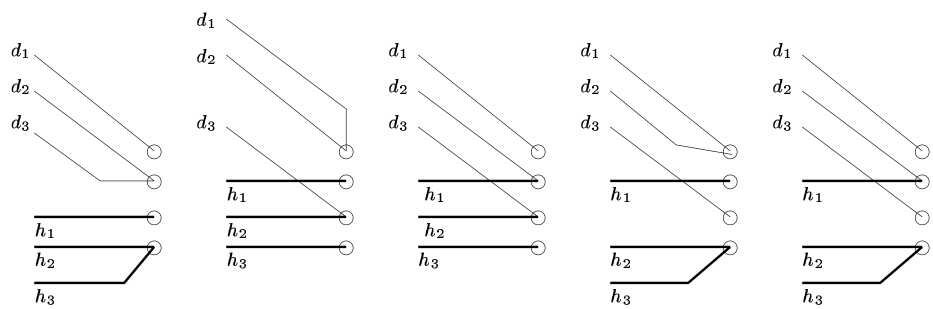


Figure 3.23: All possible shapes of u_2

shown. Clearly, some ambiguity exists with the labels of two horizontal or diagonal edges which share the same end point. However, as we shall see later, these ambiguities are not important when we refer to these labels.

Because 14 and 16 are not in u_1, u_1 can only be of shape 3 or 5, because given the way we construct and color G_{11} and G_{12}, the even-numbered edges can not share an end point. Similarly, as not both 52 and $\circ 0$ are in u_2 and $\circ 2 \notin u_2$, u_2 can only be of shape 3 or 5. For, if u_2 is of shape 1 or 4, then either h_1 was colored 2 which makes $\circ 2 \in u_2$, or h_1 was colored 0 and thus both 52 and $\circ 0$ are in u_2. If u_2 is of shape 2, then h_2 must have been colored 5, which makes $\circ 2 \in u_2$ (2 is the color of h_1 or h_3.)

Now, we describe a way to re-color the 6 edges of G in G_{11}, G_{12}, and G_{21}. The idea is to use color 1, 2 and 5 to color the horizontal edges, and 0,4,6 to color the diagonal edges. The reader might find it useful sketching several figures while we are discussing the coloring.

Let's start by coloring the edges connected to u_1. When u_1 is of shape 3, we assign 2 to h_1, and $\{5, 1\}$ to $\{h_2, h_3\}$. What we mean by this is that one of $\{h_2, h_3\}$ will get 1 and the other will get 5, but which edge actually gets which color is to be decided later. h_1, h_2 and h_3 are connected to the first two components of v_1 (vertices 0 and 1), because $v_1 \in F_1$. If some vertex $j \in \{0, 1\}$ is incident to two of these h_i ($i = 1, 2, 3$), j must be incident to some h_i where $i \in \{2, 3\}$. Assign 5 to h_i, 1 to the edge in $\{h_2, h_3\} - \{h_i\}$. Clearly, this way we get a (partially colored) v_1 respecting the P_i. Now we have to assign 0, 4, and 6 to the diagonal edges connected to u_1. Assign 0 to the edge d_i that was originally colored 6. If $i = 1$, then assign 4 and 6 to d_2 and d_3 arbitrarily. If $i \neq 1$, the assign 6 to the edge in $\{d_2, d_3\} - \{d_i\}$, and 4 to d_1. Either way, u_1 respects the P'_j ($j = 1, 2, 3$), and v_2 looks like

$$v_2 = \begin{bmatrix} c_1 c_2 \\ \circ\mathbf{0} \\ \circ\mathbf{4} \\ \mathbf{6} c_3 \end{bmatrix} \quad \text{or} \quad v_2 = \begin{bmatrix} c_1 c_2 \\ \circ\mathbf{0} \\ \circ\mathbf{6} \\ \mathbf{4} c_3 \end{bmatrix}$$

where $\{c_1, c_2, c_3\} = \{1, 2, 5\}$ are to be precisely determined when we color the horizontal edges connected to u_2. The case where u_1 is of shape 5 is done in exactly the same way.

Secondly, we need to color the edges connected to u_2. When u_2 is of shape 3, we assign 1 to the edge h_i that was originally colored 2. Clearly, $i \neq 3$ because $\circ 2 \notin u_2$. Assign 2 to h_3 and 5 to the last edge. When u_2 is of shape 5, we also assign 1 to the edge h_i that was originally colored 2. If $i \in \{2, 3\}$, then assign 2 to the edge in $\{h_2, h_3\} - \{h_i\}$ and 5 to the last edge. If $i = 1$, then assign 2 and 5 to h_2 and h_3

arbitrarily. Either way, u_2 still respects the P'_j and v_2 must either be

$$v_2 = \begin{bmatrix} \mathbf{52} \\ \circ 0 \\ \circ 4 \\ 6\mathbf{1} \end{bmatrix} \quad \text{or} \quad v_2 = \begin{bmatrix} \mathbf{52} \\ \circ 0 \\ \circ 6 \\ 4\mathbf{1} \end{bmatrix}$$

which clearly respects the P_j.

The diagonal edges of u_2 can be assigned 0,4 and 6 in much the same way. d_1 or d_2 will get 6, the other two get 0 or 4 which could be exchanged to ensure v_1 respects the P_j.

(1d) $v_2 = \begin{bmatrix} \circ 0 & 56 & \circ 4 & 12 \end{bmatrix}$.

If $14 \in u_1$, then apply $LU(1,4)$ to make $v_2 = \begin{bmatrix} \circ 0 & 56 & \circ 1 & 42 \end{bmatrix}$ and then proceed similar to case (1b).

If $16 \in u_1$, then apply $LU(1,6)$, then $LL(0,2)$ and $UL(4,6)$. Hence, we could now assume that u_1 is in shape 3 or 5.

If $52 \in u_2$, then applying $LL(2,5)$, $UU(0,2)$ and $LU(4,6)$ would do.

Now, suppose $\circ 2 \notin u_2$. As $52 \notin u_2$, u_2 can only be of shape 3 or 5. Proceeding in the same manner as case (1c) completes the case. Consequently, we could now assume that $52 \notin u_2$ and $\circ 2 \in u_2$. If $50 \in u_2$, then apply $LL(5,0)$, making $v_2 = \begin{bmatrix} \circ 5 & 06 & \circ 4 & 12 \end{bmatrix}$. Proposition 3.19 proves to be useful here again. When $52 \notin u_2$ and $\circ 2 \in u_2$ and $50 \notin u_2$, it is not hard to show that u_2 is of shape 2 or 3.

To summarize, we are left with the cases where u_1 is of shape 3 or 5, and u_2 is of shape 2 or 3. We will consider these 4 possible sub-cases in turn.

Firstly, suppose u_1 is of shape 3 and u_2 is of shape 3. We will color the horizontal edges with 0,2 and 4, the diagonal edges with 1, 5 and 6. For u_1's edges, assign 2 to h_1 and $\{0,4\}$ to $\{h_2,h_3\}$. Which edge gets 0 or 4 can be decided in the same manner as in case (1c) with 1 and 5. Assign 6 to d_3 and $\{1,5\}$ to $\{d_1,d_2\}$. Note that one of d_1 or d_2 must have been colored 1 originally. Assign 1 to that originally-1 edge and 5 to the other. After this assignment, v_2 looks like:

$$v_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{6} \\ \circ\mathbf{5} \\ \mathbf{1}c_3 \end{bmatrix} \quad \text{or} \quad v_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{5} \\ \circ\mathbf{6} \\ \mathbf{1}c_3 \end{bmatrix}$$

For u_2's diagonal edges, assign 6 to d_1, and $\{1,5\}$ to $\{d_2,d_3\}$. For u_2's horizontal edges, assign 2 to h_3 (which must have been originally colored 2 because $\circ 2 \in u_2$), and 4 to h_2 and 0 to h_1. After this assignment, v_2 can only be one of four forms:

$$v_2 = \begin{bmatrix} \circ 4 \\ 0\mathbf{6} \\ \circ\mathbf{5} \\ \mathbf{1}2 \end{bmatrix} \text{ or } v_2 = \begin{bmatrix} \circ 0 \\ 4\mathbf{6} \\ \circ\mathbf{5} \\ \mathbf{1}2 \end{bmatrix} \text{ or } v_2 = \begin{bmatrix} \circ 4 \\ 0\mathbf{5} \\ \circ\mathbf{6} \\ \mathbf{1}2 \end{bmatrix} \text{ or } v_2 = \begin{bmatrix} \circ 0 \\ 4\mathbf{5} \\ \circ\mathbf{6} \\ \mathbf{1}2 \end{bmatrix}$$

The first form is good due to Proposition 3.19. The second form is good after applying $LL(0,4)$ and Proposition 3.19. The third and fourth form is good after applying $LL(2,4)$.

Secondly, suppose u_1 is of shape 3 and u_2 is of shape 2. We can assume that the edge h_1 of u_2 was originally colored 2, otherwise h_3 was and hence the previous trick still applies. Moreover, h_2 must have been 5 originally as an originally odd-number edge must have shared an end point with another edge. Let us first try to assign colors to

u_1's edges. This time we use 0,2, and 4 for horizontal and 1,5, and 6 for diagonal edges. As "usual", assign 2 to h_1 of u_1, $\{0,4\}$ to $\{h_2,h_3\}$. Starting from here, there are two sub-cases: (i) when d_3 was originally 4; and (ii) when d_3 was originally 6.

If d_3 was originally 4, then assign 6 to d_3 and $\{1,5\}$ to $\{d_1,d_2\}$ appropriately so that

$$v_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{5} \\ \circ\mathbf{6} \\ \mathbf{1}c_3 \end{bmatrix}$$

Then, for the diagonal edges of u_2, assign 6 to d_1 and $\{1,5\}$ to $\{d_2,d_3\}$ with 1 and 5 exchangeable to make v_1 valid. For the horizontal edges of u_2, assign 4 to h_1 (which was originally 2) and $\{0,2\}$ to $\{h_2,h_3\}$ arbitrarily. After this assignment, v_2 becomes either

$$v_2 = \begin{bmatrix} \circ\mathbf{0} \\ \mathbf{2}5 \\ \circ 6 \\ 1\mathbf{4} \end{bmatrix} \quad \text{or } v_2 = \begin{bmatrix} \circ\mathbf{2} \\ \mathbf{0}5 \\ \circ 6 \\ 1\mathbf{4} \end{bmatrix}$$

which obviously respect the P_i.

When d_3 was originally 6, the situation is more complicated. We will use 0,6 and 5 for horizontal and $1,2$ and 4 for diagonal edges. For u_1's edges, assign 5 to h_3 and $\{0,6\}$ to $\{h_1,h_2\}$. 0 and 6 could be exchanged to make v_1 respect the P_i. Next, assign 2 to d_3, 1 to d_1 and 4 to d_2. After this assignment, v_2 can be one of two forms:

$$v_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{2} \\ \circ\mathbf{1} \\ \mathbf{4}c_3 \end{bmatrix} \quad \text{or } v_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{2} \\ \circ\mathbf{4} \\ \mathbf{1}c_3 \end{bmatrix}$$

Now it is u_2's edges' turn. Assign 2 to d_3, $\{1,4\}$ exchangeable to $\{d_1, d_2\}$. Next, assign 6 to h_1, 5 to h_2 (which was 5 originally), and 0 to h_3. v_2 could now only be:

$$v_2 = \begin{bmatrix} \circ\mathbf{0} \\ \mathbf{5}2 \\ \circ 1 \\ 4\mathbf{6} \end{bmatrix} \quad \text{or } v_2 = \begin{bmatrix} \circ\mathbf{0} \\ \mathbf{5}2 \\ \circ 4 \\ 1\mathbf{6} \end{bmatrix}$$

The second form already respects the P_i. The second form is good too, due to Proposition 3.19.

Thirdly, suppose u_1 is of shape 5 and u_2 is of shape 2. We handle this case differently: we will color the edges so that v_1 and v_2 respect the P_i', and u_1 and u_2 respect the P_i. A vertical flip of G would complete the case. Notice that at u_1 the original color of d_2 must have been 1, and at u_2 the original color of h_2 must have been 5. We will use $0, 2$ and 5 for horizontal edges and $1, 4$ and 6 for diagonal edges.

For u_1's diagonal edges, assign 4 to d_2, 1 to the edge that was 6 and 6 to the edge that was 4 originally. After this assignment, u_1 is of one of two forms:

$$u_1 = \begin{bmatrix} c_1 c_2 \\ \mathbf{1}\circ \\ c_3\mathbf{4} \\ \circ\mathbf{6} \end{bmatrix} \quad \text{or } u_1 = \begin{bmatrix} c_1 c_2 \\ \mathbf{6}\circ \\ c_3\mathbf{4} \\ \circ\mathbf{1} \end{bmatrix}$$

and v_2 must be $\begin{bmatrix}\circ 2 & 01 & \circ 6 & 45\end{bmatrix}$, which respects the P_i'. Where $\{c_1, c_2, c_3\} = \{1, 4, 6\}$ whose exact assignment will be decided later. If u_2 is of the first form, assign 0 to h_3 and $\{2, 5\}$ to $\{h_1, h_2\}$ exchangeable for v_1 to respect the P_i'. If u_2 is of the second form, assign 2 to h_3 and $\{0, 5\}$ to $\{h_1, h_2\}$ exchangeable for v_1 to respect the P_i'.

For u_2's horizontal edges, assign 0 to h_2, 5 to the edge that was 2, and 2 to the edge that was 0 originally. After this assignment, u_2 is one of two forms:

$$u_2 = \begin{bmatrix} c_1c_2 \\ \mathbf{2}\circ \\ c_3\mathbf{0} \\ \circ\mathbf{5} \end{bmatrix} \quad \text{or } u_2 = \begin{bmatrix} c_1c_2 \\ \mathbf{5}\circ \\ c_3\mathbf{0} \\ \circ\mathbf{2} \end{bmatrix}$$

Here $\{c_1, c_2, c_3\} = \{0, 2, 5\}$ whose exact assignment will be decided later. If u_1 is of the first form, assign 6 to d_3 and $\{1, 4\}$ to $\{d_1, d_2\}$ exchangeable for v_1 to respect the P_i'. If u_1 is of the second form, assign 4 to d_3 and $\{1, 6\}$ to $\{d_1, d_2\}$ exchangeable for v_1 to respect the P_i'. It is also easy to check that v_2 respects the P_i'.

Lastly, suppose u_1 is of shape 5 and u_2 is of shape 3. In this case, if d_2 and d_3 of u_2 don't share the left end point, then we can reuse the previous trick. If d_2 and d_3 of u_2 have the same left end point, we start coloring in the usual way, namely for the v_j to respect the P_i and u_j to respect the P_i'. Note that at u_1 the color of d_2 originally was 1, and at u_2 the original color of h_3 was 2 (because $\circ 2 \in u_2$). We have to consider two cases, depending on the original color of d_1 at u_1: (i) d_1 was 6; and (ii) d_1 was 4.

If at u_1, d_1 was originally 6 then we use 0,5, and 6 for the horizontal edges and $1, 2,$ and 4 for the diagonal edges. We start with u_1's horizontal edges, assigning 6 to h_3

and $\{5,0\}$ to h_1 and h_2. Then, 2 to d_1, 1 to d_2 and 4 to d_3. After this assignment, we must have

$$v_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{2} \\ \circ\mathbf{4} \\ \mathbf{1}c_3 \end{bmatrix}$$

Now, let's assign colors to the horizontal edges of u_2. We assign 6 to h_3, whose original color was 2, and $\{5,0\}$ to $\{h_1,h_2\}$, whose exact assignment is such that v_2 respects P_i. This is certainly possible. After the assignment, we either have

$$u_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{0} \\ c_3\mathbf{5} \\ \circ\mathbf{6} \end{bmatrix} \quad \text{or} \quad u_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{5} \\ c_3\mathbf{0} \\ \circ\mathbf{6} \end{bmatrix}$$

where, c_i is the color of d_i to be chosen from $\{1,2,4\}$. We assign 2 to d_1, 4 to d_2 and 1 to d_3 if u_2 is in the first form. Assign 2 to d_1, 1 to d_2 and 4 to d_3 if u_2 is in the second form. Recall that d_2 and d_3 shares the left end point, hence v_1 keeps respecting the P_i.

If at u_1, d_1 was originally 4 then we use 1,5, and 6 for horizontal and 0,2, and 4 for vertical edges. Starting first with u_1's edges, we assign 6 to h_1, $\{1,5\}$ to $\{h_2,h_3\}$, which are exchangeable to make v_1 respect the P_i. Then, assign 2 to d_3, 4 to d_1 and 0 to d_2, making

$$v_2 = \begin{bmatrix} \circ c_1 & c_2\mathbf{2} & \circ\mathbf{4} & \mathbf{0}c_3 \end{bmatrix}$$

With u_2's edges, assign 6 to h_3 and $\{5,1\}$ to $\{h_1,h_2\}$ such that

$$v_2 = \begin{bmatrix} \circ\mathbf{1} & \mathbf{5}2 & \circ 4 & 0\mathbf{6} \end{bmatrix}$$

which respects the P_i. After this assignment, u_2 is either

$$u_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{1} \\ c_3\mathbf{5} \\ \circ\mathbf{6} \end{bmatrix} \quad \text{or} \quad u_2 = \begin{bmatrix} \circ c_1 \\ c_2\mathbf{5} \\ c_3\mathbf{1} \\ \circ\mathbf{6} \end{bmatrix}$$

If u_2 is in the first form, assign 0 to d_3 and $\{2,4\}$ to $\{d_1, d_2\}$, which are exchangeable to make v_1 valid. If u_2 is in the second form, assign 4 to d_3 and $\{0,2\}$ to $\{d_1, d_2\}$ arbitrarily.

Case 2. $v_1 \in F_1$ *and* $v_2 \in F_2$. In this case, we flip G vertically so that $u_1 \in F_1$ and $u_2 \in F_2$. Clearly u_1 and u_2 respect P_i'. We need to modify the coloring so that v_1 and v_2 respect P_i. (P_1 is already respected.) Since $u_1 \in F_1$ and $u_2 \in F_2$, we can apply any sequence of $UU(0,2)$, $LU(4,6)$, $UL(4,6)$ and $LL(0,2)$ while keeping P_i' respected by the u_j. As case 1 has been done, we can assume that v_1 and v_2 are not representatives of F_1 and $F_3 \cup F_4$. Consider 3 sub-cases left:

(2a) $v_1 \in F_1$ and $v_2 \in F_2$. In this case, apply $UU(0,2)$ and/or $UL(4,6)$ if necessary to make $v_1 = \begin{bmatrix} 50 & \circ 2 & 14 & \circ 6 \end{bmatrix}$. Apply $LU(4,6)$ as needed so that 14 is one component of u_1. Apply $LL(0,2)$ if necessary so that v_2 is either $\begin{bmatrix} 10 & \circ 2 & 54 & \circ 6 \end{bmatrix}$ or $\begin{bmatrix} 10 & \circ 2 & \circ 4 & 56 \end{bmatrix}$. Next, as 14 is a component of u_1, we can apply $LU(1,4)$, so that v_2 is either $\begin{bmatrix} 40 & \circ 2 & 51 & \circ 6 \end{bmatrix}$ or $\begin{bmatrix} 40 & \circ 2 & \circ 1 & 56 \end{bmatrix}$. The first vector already respects P_i. The second vector respects P_i after we switch 2 and 6, so does v_1. Hence, we apply $A(2,6)$ if necessary to make both v_1 and v_2 respect P_i.

(2b) $v_1 \in F_2$ and $v_2 \in F_1$. This is quite similar to the previous case. Apply $LL(0,2)$ and/or $LU(4,6)$ if necessary to make $v_2 = \begin{bmatrix} 50 & \circ 2 & 14 & \circ 6 \end{bmatrix}$. Apply $UU(0,2)$ as needed so that 50 is one component of u_1. Apply $UL(4,6)$ if necessary so that v_1 is either $\begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$ or $\begin{bmatrix}\circ 0 & 12 & 54 & \circ 6\end{bmatrix}$. Next, as 50 is a component of u_1, we can apply $UU(0,5)$, so that v_1 is either $\begin{bmatrix}15 & \circ 2 & 04 & \circ 6\end{bmatrix}$ or $\begin{bmatrix}\circ 5 & 12 & 04 & \circ 6\end{bmatrix}$. The first vector already respects P_i. The second vector respects P_i after we switch 2 and 6, so does v_2. Hence, we apply $A(2,6)$ if necessary to make both v_1 and v_2 respect P_i.

(2c) v_1 and v_2 are representatives of F_2 and $F_3 \cup F_4$.

Suppose $v_1 \in F_3 \cup F_4$, so that $v_2 \in F_2$. Apply $LU(4,6)$ and/or $LL(0,2)$ as needed so that $v_2 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$. If $v_1 \in F_3$, then apply $UU(0,2)$ and $UL(4,6)$ to move v_1 to F_4. Finally, we apply $A(0,4)$, making v_1 and v_2 both respect P_i.

The situation when $v_1 \in F_2$ and $v_2 \in F_3 \cup F_4$ is done similarly.

Case 3. $v_1 \in F_2$ *and* $v_2 \in F_3 \cup F_4$. As $A(0,2)$ and $A(4,6)$ transform a vector from F_4 to F_3, while keeping a vector from F_2 in F_2, we only need to consider $v_1 \in F_2$ and $v_2 \in F_3$.

If $v_1 = \begin{bmatrix}\circ 0 & 12 & \circ 4 & 56\end{bmatrix}$, then we first apply $A(0,2)$ and $A(4,6)$. This would make $v_1 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$, and move v_2 into F_4. Secondly, $A(0,4)$ makes both v_1 and v_2 respect the P_i.

If $v_1 = \begin{bmatrix}10 & \circ 2 & \circ 4 & 56\end{bmatrix}$ and $u_1 \in F_2$, then apply these in order: $LU(4,6)$, $LL(0,2)$, $UL(4,6)$, and $A(0,4)$. While if v_1 is the same but $u_2 \in F_2$, we do $UL(4,6)$, and $A(0,4)$.

If $v_1 = \begin{bmatrix}\circ 0 & 12 & 54 & \circ 6\end{bmatrix}$ and $u_1 \in F_2$, we apply a sequence of transformations: $UU(0,2)$, then $A(0,2)$ and $A(4,6)$, then $A(0,4)$. While if v_1 is the same but $u_2 \in F_2$, we do $UU(0,2)$, $LU(4,6)$, $LL(0,2)$ and lastly $A(0,4)$.

Consequently, the cases left to be considered are:

- $v_1 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$, $v_2 \in F_3$, u_1 and u_2 are representatives of some F_i and F_j

with $i \neq j$, $i \neq 1$ and $j \neq 1$.

- $v_1 = \begin{bmatrix} 10 & \circ 2 & \circ 4 & 56 \end{bmatrix}$, u_1 and u_2 are representatives of F_3 and F_4.
- $v_1 = \begin{bmatrix} \circ 0 & 12 & 54 & \circ 6 \end{bmatrix}$, u_1 and u_2 are representatives of F_3 and F_4.

We consider all these sub-cases in turn, using the technique "proof without words" introduced in the previous section.

(3a) $v_1 = \begin{bmatrix} 10 & \circ 2 & 54 & \circ 6 \end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_2$, and $u_2 \in F_3$. Figure 3.24 shows our solution to this case. The dashed edges are from G_{21}, the thicken edges are from

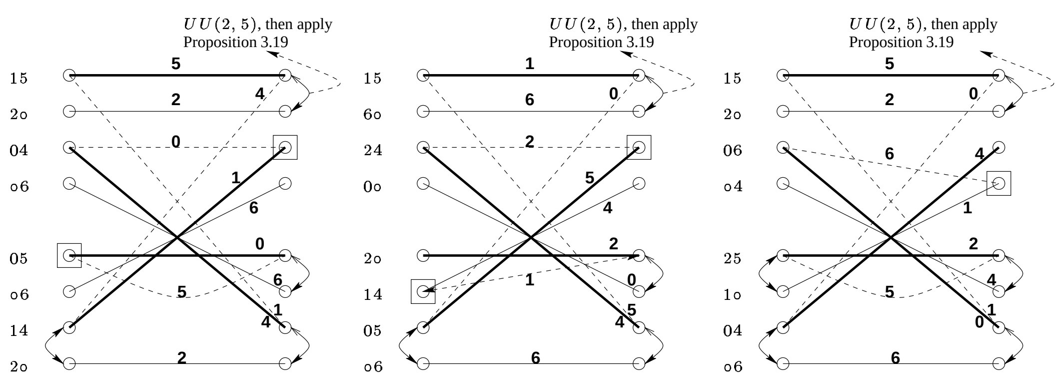


Figure 3.24: Case 3a of Lemma 3.13

G_{11}, and the rest of the edges are from G_{12}. As we are considering the case where $v_2 \in F_3$, $u_1 \in F_2$, and $u_2 \in F_3$, there are 6 degrees of flexibility for the 1 and 5 to move within either the first two components or the last two components of the associated CIVs. This corresponds precisely to moving the end points of the dashed edges, within the first two components or the last two components of their CIVs.

In reality, there are totally $2^6 = 64$ cases. In Figure 3.24, we consider only three cases where the end points of the dashed edges in the boxes are fixed. The doubly headed arrows indicate that no matter if this end point is at one head or the other, this coloring is still good. Occasionally, we have to explain why when we move the end point of the dashed edge to the other head of the arrow, the coloring is still OK. The explanations are given right on the figure itself, and they are self-explaining.

The CIVs v_1 and v_2 have been put next to the colorings for the ease of referencing which edge gets which color, and checking if the coloring is a valid one. Note that we only color the edges not in G_{22} as usual.

(3b) $v_1 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_3$, and $u_2 \in F_2$. Figure 3.25 shows our solution to this case.

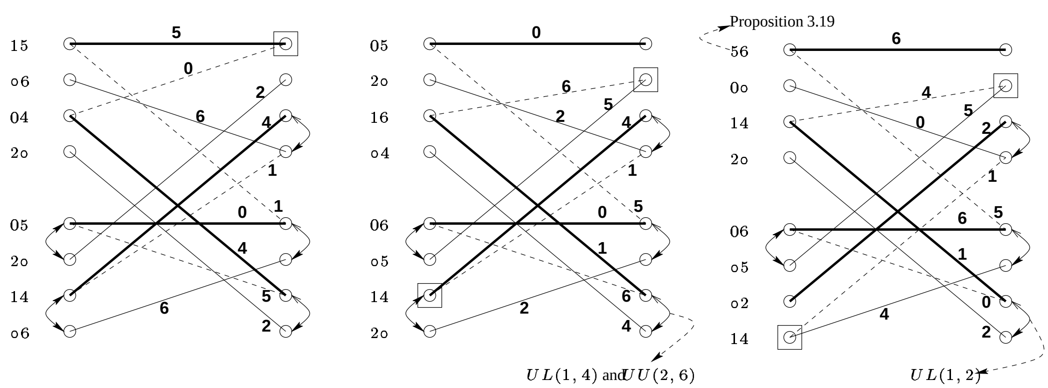


Figure 3.25: Case 3b of Lemma 3.13

(3c) $v_1 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_2$, and $u_2 \in F_4$. Figure 3.26 shows our solution to this case.

(3d) $v_1 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_4$, and $u_2 \in F_2$. This sub-case is a little special, as we have to consider 4 cases shown in Figure 3.27.

(3e) $v_1 = \begin{bmatrix}10 & \circ 2 & 54 & \circ 6\end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_3$, and $u_2 \in F_4$. This case is even more special. There are 4 sub-cases to be considered as shown in Figure 3.28. In the last

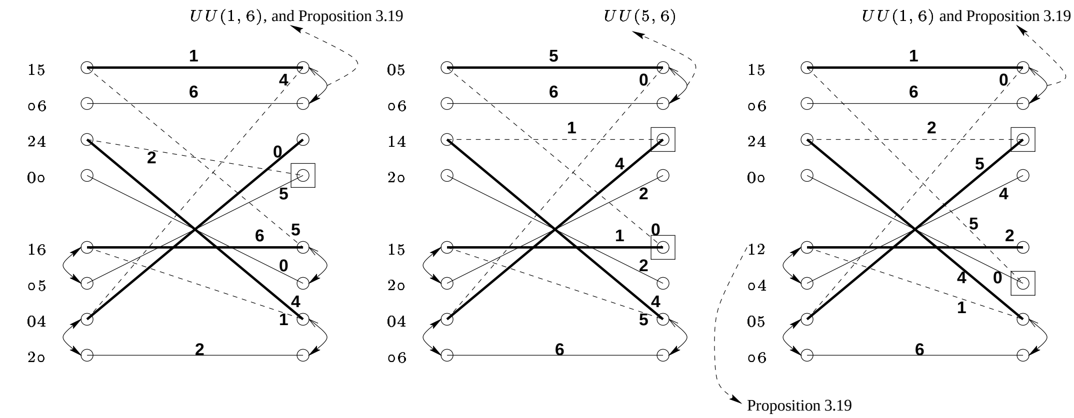


Figure 3.26: Case 3c of Lemma 3.13

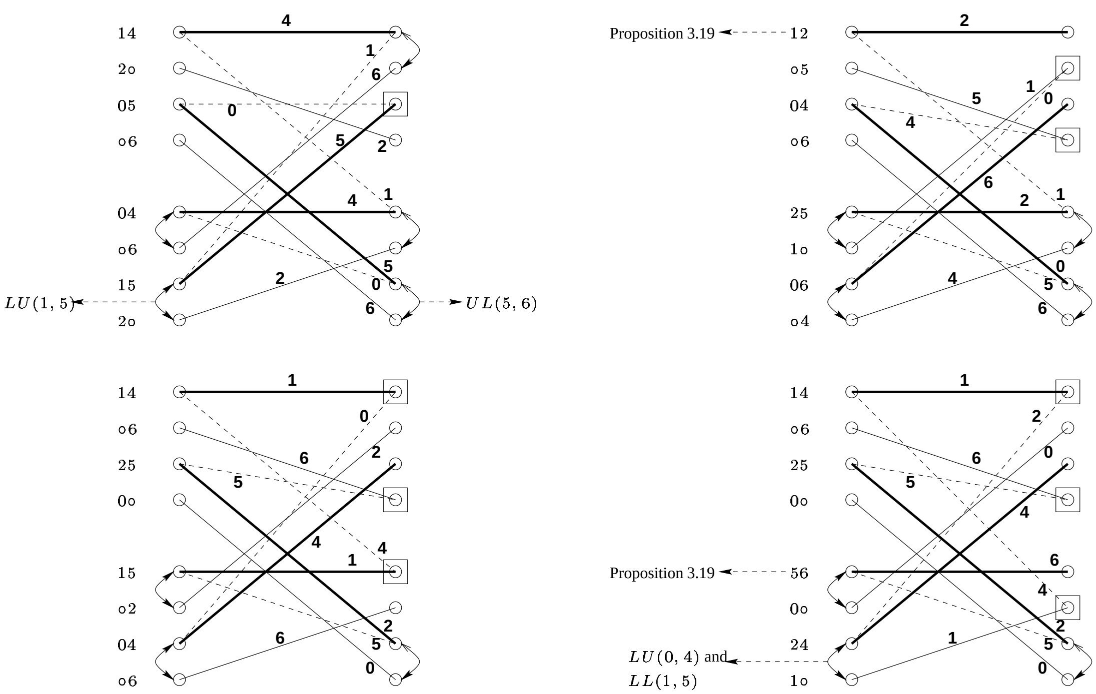


Figure 3.27: Case 3d of Lemma 3.13

sub-case (see the last drawing of Figure 3.28), we have provided the coloring so that the v_j respect the P_i' and the u_j respect the P_i. Flipping G vertically would complete the proof. The vectors u_1 and u_2 have been put on the right of the drawing, indicating that we are trying to flip G vertically.

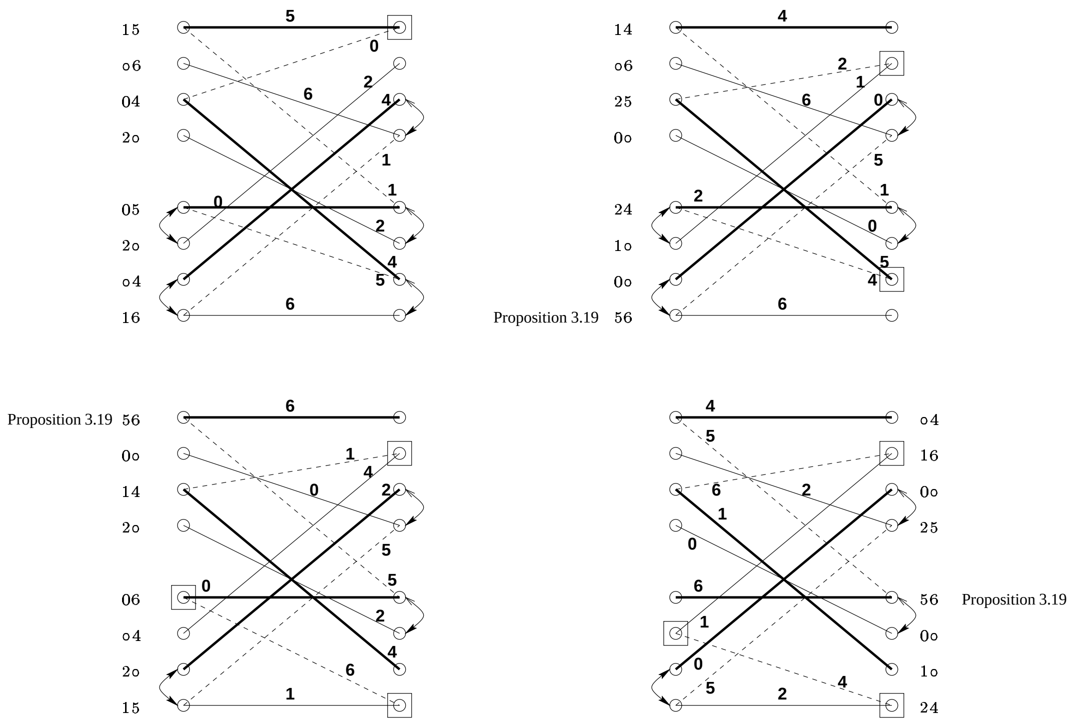


Figure 3.28: Case 3e of Lemma 3.13

(3f) $v_1 = \begin{bmatrix} 10 & \circ 2 & 54 & \circ 6 \end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_4$, and $u_2 \in F_3$. Figure 3.29 shows our solution to this case. In the first and second drawings, we use the vertical flip of G.

(3g) $v_1 = \begin{bmatrix} 10 & \circ 2 & \circ 4 & 56 \end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_3$, and $u_2 \in F_4$. Figure 3.30 shows our solution to this case.

(3h) $v_1 = \begin{bmatrix} 10 & \circ 2 & \circ 4 & 56 \end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_4$, and $u_2 \in F_3$. The coloring shown in case (3f) is also valid here.

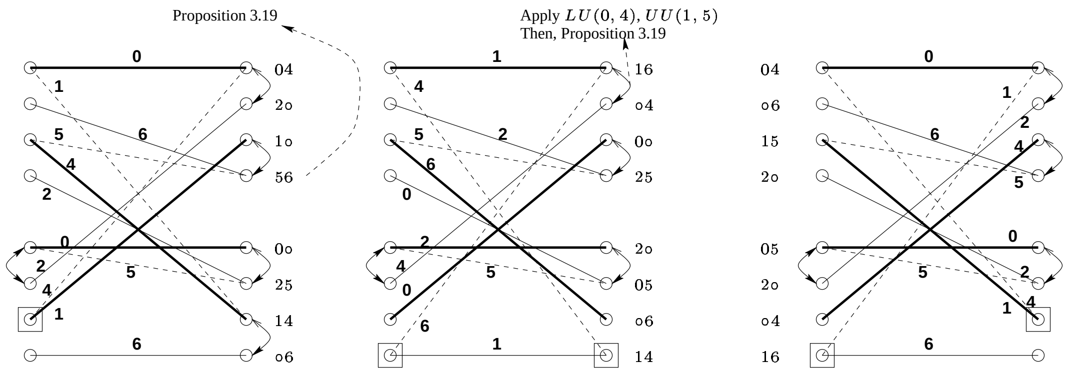


Figure 3.29: Case 3f of Lemma 3.13

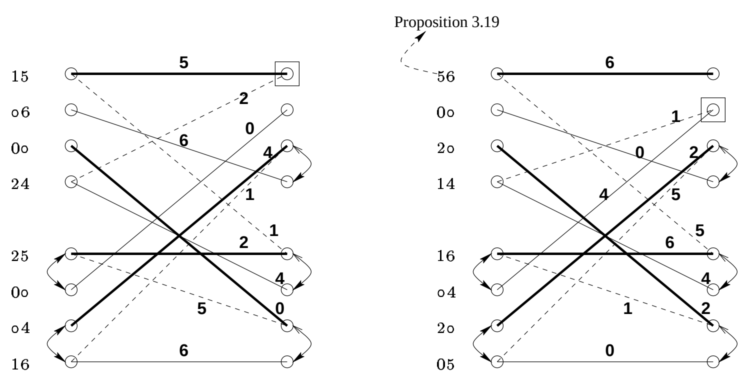


Figure 3.30: Case 3g of Lemma 3.13

(3i) $v_1 = \begin{bmatrix} \circ 0 & 12 & 54 & \circ 6 \end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_3$, and $u_2 \in F_4$. Figure 3.31 shows our solution to this case. We use the vertical flip of G in the second, third and fifth drawings.

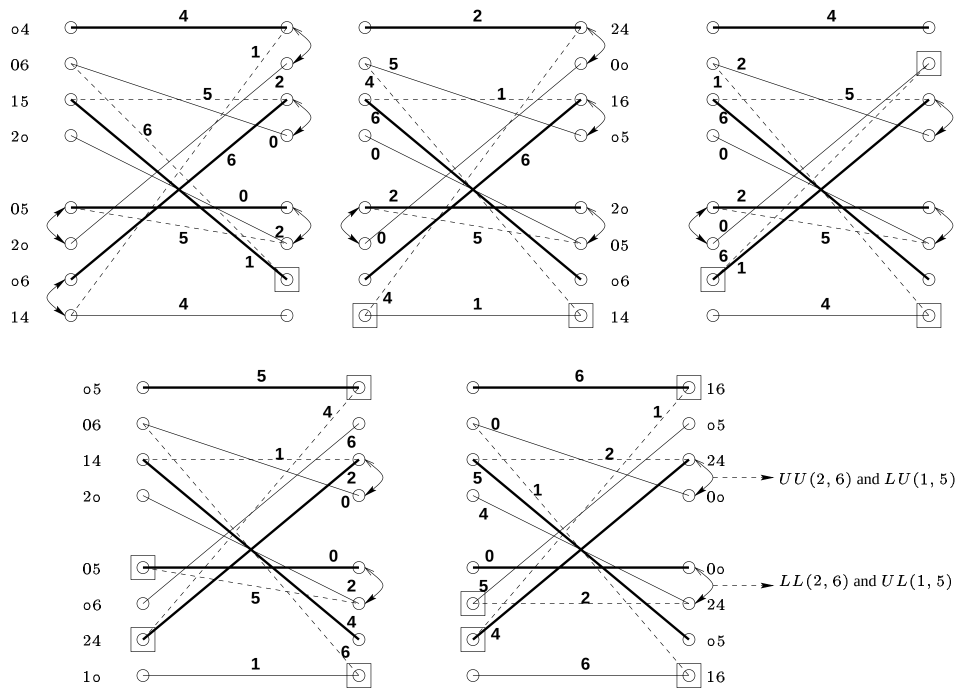


Figure 3.31: Case 3i of Lemma 3.13

(3j) $v_1 = \begin{bmatrix} \circ 0 & 12 & 54 & \circ 6 \end{bmatrix}$, $v_2 \in F_3$ $u_1 \in F_3$, and $u_2 \in F_4$. The coloring in case (3e) is also valid here.

Case 4. $v_1 \in F_3$ *and* $v_2 \in F_4$. Now that all cases 1,2, and 3 have been proven, we can assume that u_1 and u_2 are also representatives of F_3 and F_4. Otherwise, we flip G vertically and apply one of the previous cases. This case is surprisingly simple. There are 2 sub-cases depending on which of u_1 and u_2 comes from which of F_3 and F_4. Figure 3.32 shows the solution to both the sub-cases.

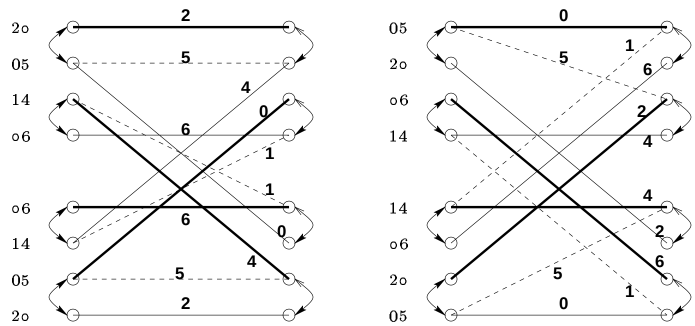


Figure 3.32: Case 4 of Lemma 3.13

3.6 Discussions

In this chapter, we have verified that the 7-stage SE network for $n = 4$ is rearrangeable. This result and an extension of another formulation were used to show that $3n-5$ SE stages are sufficient for the rearrangeability of the SE network with 2^n inputs and 2^n outputs.

It was conjectured that $2n-1$ SE stages are necessary and sufficient for the SE network to be rearrangeable. However, there has been very slow progress on proving the conjecture. Although the proof of the main theorem is tedious and fairly tricky, there are several tricks that are used very often. Some of the them were general enough to be put as lemmas and propositions. We were not able to generalize the others, which are interesting in their own right. The proof, in some sense, also “shows” why this conjecture is so difficult. There is no particular technique that could be used throughout, we need different tricks to solve different sub-cases. That is not to say there is no nice proof of the main theorem or the conjecture, we just were not able to see the “right” formulation, yet. In particular, any connection of our proof to the formulation of Linial and Tarsi should be helpful, beside the fact that the binary representation of our colors are precisely the row vectors of the matrix M to be constructed in Linial and Tarsi’s formulation.

We hope that our work, besides improving the bound, contributes to the effort of attacking this difficult problem. We believe that an algebraic formulation of the proof would yield better upper bound for $m(n)$.

Chapter 4

Multirate Rearrangeability and the Chung-Ross Conjecture

4.1 Overview

As we have mentioned in chapter 3, the notion of rearrangeability is fundamental in any space switching devices such as telephone switches or optical cross-connects. In fact, this notion of rearrangeability is referred to as being *classical*, as in *classical switching networks* for telephone technology. New multiplexing technologies on optical networks such as time/frequency/wavelength division multiplexing (TDM/FDM/WDM) allow a link to carry more than one channel with different bandwidths, often referred to as different *rates*. This environment is called the *multirate* environment. It also makes sense as *integrated broadband networks* are viewed by many experts to be the future technology [85], in which the network needs to provide services to many applications with different bandwidth requirements. New switches need to be able to switch multirate channels fast enough not to be a bottleneck of an optical network. Consequently, classical switching networks need to be extended to fit the new multirate paradigm. The first effort along this direction was done by Melen and Turner [119]. The authors partially generalized several concepts from classical switching networks to multirate switching networks. Chung and Ross [42] continued this theme and studied the "multirate rearrangeability" of the symmetric 3-stage

Clos network, one of the most basic building blocks of MINs. In this chapter, we study a conjecture made by Chung and Ross on the minimum number of middle switches for a symmetric Clos network to be multirate rearrangeable. We shall give the original formulation and two other equivalent formulations. Several related properties shall be shown and improvements on the upper and lower bounds of the number of middle switches shall be provided.

4.1.1 Original Formulation

The Clos network has been widely used for data communications and parallel computing systems. Quite a lot of research efforts [14, 38, 42, 52, 69, 72, 86, 106, 108, 109, 119, 151] have been put on investigating the non-blocking properties and rearrangeability of the Clos network. The 3-stage Clos network was paid special attention to since it can be expanded in a "straightforward" way to multi-stage Clos network. Our main interest is in the multirate rearrangeability of the symmetric 3-stage Clos network. Let us formally introduce some related concepts.

The 3-state Clos network $C(n_1, r_1, m, n_2, r_2)$ is a 3-stage MIN, where the first stage consists of r_1 crossbars of size $n_1 \times m$, the last stage has r_2 crossbars of dimension $m \times n_2$, and the middle stage has m crossbars of dimension $r_1 \times r_2$ (see figure 4.1). Each input switch I_i ($i = 1, \ldots, r_1$) is connected to each middle switch M_j ($j = 1, \ldots, m$). Similarly, the middle stage and the last stage are fully connected.

The *symmetric* 3-*stage Clos network* $C(n, m, r)$ is nothing but $C(n, r, m, n, r)$. A $C(2, 3, 4)$ is shown in Figure 4.2. Any switch is assumed to be nonblocking, i.e. any inlet can be connected to any outlet as long as there's no conflict on the outlet. This can be thought of as a crossbar of size $p \times q$ with pq crosspoints. Having too many cross-

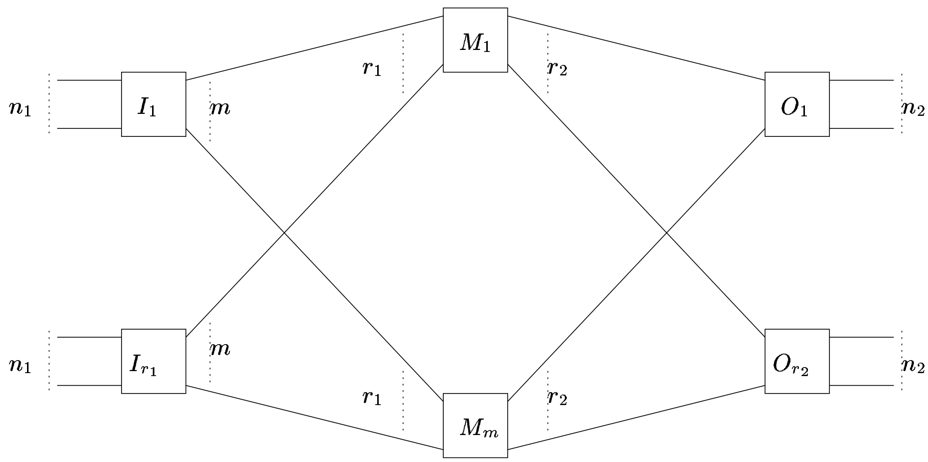


Figure 4.1: The 3-stage Clos network $C(n_1, r_1, m, n_2, r_2)$

points is expensive and we would like to design a huge switch using smaller switches with fewer number of crosspoints than when a brute-force design is used. We can also envision these designs to be some network (or bus) topology where a set of stations (processors, memories, etc.) can simultaneously communicate with another set of stations. Despite its practical roots and implications, we are only concerned about the mathematical aspects of this problem.

The inlets (outlets) of the input (output) switches are the *inputs* (*outputs*) of the network. Inputs and outputs are referred to as *external links*, while links between switches are referred to as *internal links*. When $C(n, m, r)$ is used in the multi-rate environment, each connection is associated with a weight which can be thought of as the bandwidth requirement of that connection. A link can carry any number of connections as long as the *load*, i.e. the total weight of these connections, does not exceed the link capacity. It is commonly assumed that links have uniform capacity which is normalized to be unity.

A *connection request* is a triple (i, j, w) where i is an inlet, j an outlet, and w the weight. A *request frame* is a collection of requests such that the total weight of all requests

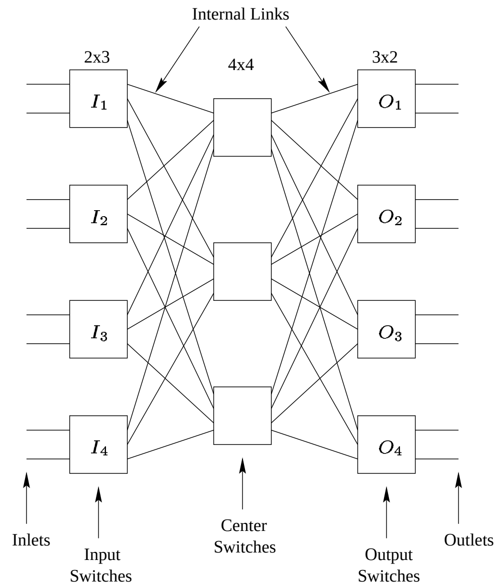


Figure 4.2: A 3-stage Clos network $C(2, 3, 4)$

in the frame involving a fixed inlet or outlet does not exceed unity. To discuss routing it is convenient to assume that all links are directed from left to right. Thus a *path* from an inlet to any outlet always consists of the sequence: an inlet link $\rightarrow$ an input switch $\rightarrow$ a link $\rightarrow$ a center switch $\rightarrow$ a link $\rightarrow$ an output switch $\rightarrow$ an outlet link. Furthermore, since the crossbar is assumed to be nonblocking, a request (i, j, w) is *routable* if and only if there exists a path from i to j such that every link on this path has unused capacity at least $1 - w$ before carrying out this request. A request frame is routable if there exists a set of paths, one for each request, such that for every link the sum of weights of all requests going through it does not exceed unity. $C(n, m, r)$ is called *multirate rearrangeable* (or just rearrangeable as the context is obvious) if *every* request frame is routable.

Let $M(n, r)$ denote the minimum value of m such that $C(n, m, r)$ is rearrangeable for given n and r. Our problem is to find $M(n, r)$. The problem seems to be difficult, as in 1991 Chung and Ross has conjectured that $M(n, r) = 2n - 1$ (independent of r!!) and so far no one has been able to prove or disprove the conjecture.

4.1.2 The Bipartite Graph Edge Coloring Formulation

Given a request frame $\mathcal{F}$, define a weighted bipartite graph $G_{\mathcal{F}} = (I, O)$ where I (resp. O) contains all the input (resp. output) switches. There is an edge with weight w between vertices X, Y of G for each request (x, y, w) where x (resp. y) is an inlet (resp. outlet) of X (resp. Y). $C(n, m, r)$ is rearrangeable iff for all $\mathcal{F}$ the edges of $G_{\mathcal{F}}$ can be m-colored such that at every vertex, the total weight of same color edges incident to this vertex is at most unity. To see this, just associate each color with a center switch.

In summary, our problem can be formulated as follows. Given positive integers n, r and an edge-weighted bipartite graph $G = (I, O)$ with the following properties:

1. $|I| = |O| = r$.

2. $\forall v \in V(G)$, all edges incident to v can be partitioned into n groups such that the total weight of edges from each group is at most 1.

For any edge-coloring of G, let $w_c(v)$ denotes the total weights of edges colored c which are incident to v. An edge-coloring of G is *valid* iff $w_c(v) \leq 1$ for all c and v. Our problem is to find the minimum $M(n, r)$ such that a valid $M(n, r)$-edge-coloring of G exists.

4.1.3 The Matrix Coloring Formulation

Number the inlets (outlets) consecutively from 1 to nr so that the input switch I_i (resp. output switch O_j) consists of all inlets numbered $(i-1)n+1, \dots in$ (resp. $(j-1)n+1, \dots jn$). Construct an $nr \times nr$ matrix M whose rows (columns) correspond to the inlets (outlets). Assign $M_{ij} = w$ if there exists a request (i, j, w), and $M_{ij} = 0$ otherwise. M

satisfies the condition that each of its row and column sums is between 0 and 1. Each input (resp. output) switch I_i (resp. O_j) corresponds to the set of rows $(i-1)n+1, \ldots in$ (resp. $(j-1)n+1, \ldots jn$) which we will call a *block row* (resp. *block column*). Our problem is to find the minimum integer $M(n,r)$ such that we can color all cells of M using $M(n,r)$ colors satisfying : (a) the sum of cells of the same color in each block row is at most unity; (b) the sum of cells of the same color in each block column is at most unity.

Remark 4.1. There is a subtle difference between this formulation and the other two. Here, by coloring each cell as a whole we force any two requests of the form (i,j,w_1) and (i,j,w_2) to be routed on the same route. However, we shall prove that this formulation is equivalent to the other two with respect to the minimum number of colors $M(n,r)$.

4.2 Existing Results

When all rates are either 0 or 1, the problem is called the *classical circuit switching problem* [14]. The following theorem can be found in [14].

Theorem 4.2. *When all weights are chosen from* $\{0,1\}$, $M(n,r) = n$.

We shall provide a slightly different proof from that in [14] of this theorem in a later section, making use of our matrix formulation. For the general multi-rate network, the results are quite fragmented. Let us introduce some more notations. When the weights are forced to be chosen from a given set K of k distinct weights, let $M^k(n,r)$ denote minimum number of middle switches so that $C(M^k(n,r),n,r)$ is rearrangeable in the sense that all requests have weights in K. If b and B are lower and upper bounds of all weights, where $0 < b \leq B \leq 1$, then $M(n,r)$ becomes $M_{[b,B]}(n,r)$. Similarly, we define $M_{(b,B]}(n,r)$, $M_{[b,B)}(n,r)$, and $M_{(b,B)}(n,r)$ in the obvious way.

Remark 4.3. The followings are immediate.

(i) $M(n,1) = n$.

(ii) $M(1,r) = 1$.

(iii) $M^1(n,r) = n$ by Theorem 4.2.

(iv) $M(n,r) \geq M^{k+1}(n,r) \geq M^k(n,r)\ \forall k \geq 1$.

(v) $M(n,r) = M_{[0,1]}(n,r) \geq M_{[b,B]}(n,r)$ for $0 < b \leq B \leq 1$

(vi) $M(n,r) \geq n$.

Thus, from now on we assume $n, r \geq 2$. Finding the exact values of $M(n,r)$ seems to be a difficult problem, just as what happens with our previous Beneš conjecture problem, although this problem has a completely different nature and seems to be more tractable.

4.2.1 Lower Bounds

There are very few known results on the lower bounds of $M(n,r)$. One plausible reason for the lack of attention to the lower bound is that from the practical point of view, the upper bound is more important. However, I think another reason is that the "only" way to get a lower bound l for $M(n,r)$ is to specify a request frame which requires at least l middle switches, while specifying this request frame gives so much insight into solving the problem which we don't quite have yet. Du, Gao, Hwang, and Kim [52] proved the following.

Theorem 4.4. *When $r \geq 3$,*

(i) $M^2(n,r) \geq n+1$.

(ii) $M^3(n,r) \geq \lceil \frac{11n}{9} \rceil$.

We shall improve all these bounds by showing that when $r \geq 2$, $M^3(n,r) \geq \lceil \frac{5n}{4} \rceil$.

4.2.2 Upper Bounds

Melen and Turner [119] gave an elegant algorithm called CAP to route requests. Based on CAP and a new tricky lemma, Du et al. [52] proved :

Theorem 4.5 (Du et al., 1998). *For* $0 < b \leq B \leq 1$,

(i) $M_{[0,B]}(n,r) \leq \frac{n-B}{1-B}$.

(ii) $M_{[b,1]}(n,r) \leq n\lfloor \frac{1}{b} \rfloor$.

(iii) $M(n,r) \leq \lceil \frac{41n-E_n}{16} \rceil$, *where* $E_n = 8, 5, 6, 3$ *if* $n \equiv 0, 1, 2, 3 \pmod 4$.

Recently, Lin, Du, Hu, and Xue [108] proved the followings.

Theorem 4.6. $M(n,r) \leq 2n-1$ *when each weight is chosen from a given finite set*

$$\{p_1, \ldots p_h, p_{h+1}, \ldots p_k\},$$

where

$$1 \geq p_1 > \ldots p_h \geq p_{h+1} > \ldots > p_k,$$

and

$$p_k | p_{k-1}, p_{k-1} | p_{k-2}, \ldots p_{h+2} | p_{h+1}.$$

Here h *and* k *are positive integers, and* $x|y$ *means* y *is an integer multiple of* x.

Theorem 4.7. *We have*

(i) $M^2(n, r) \leq 2n - 1$.

(ii) $M_{(\frac{1}{3},1]}(n, r) \leq 2n - 1$.

In another paper, Lin, Du, Wu, and Yoo [109] proved :

Theorem 4.8. *We have*

(i) $M^3(n, r) \leq \lceil \frac{9n}{4} \rceil$.

(ii) $M^3_{(\frac{1}{5},1]}(n, r) \leq 2n$.

4.3 Main Results

All of the results above were approached from the first two formulations. It is natural to start approaching the problem from the third formulation with the hope of obtaining some result(s) of a different nature. This is true as it turns out to be, although our results here are not as strong in most cases.

4.3.1 General Observations

First we show the equivalence of the matrix formulation to the other two. Let $M_{matrix}(n, r)$ and $M_{original}(n, r)$ denote $M(n, r)$ for the matrix and original formulation respectively. We prove

Proposition 4.9. $M_{matrix}(n, r) = M_{original}(n, r)$

Proof. Given any request frame $\mathcal{F}$, construct our matrix M as usual with $M_{ij} = \sum_{(i,j,w)\in\mathcal{F}} w$. Since we can use $M_{matrix}(n,r)$ colors to properly color our matrix, only $M_{matrix}(n,r)$ middle switches are needed to route the requests in $\mathcal{F}$. This shows $M_{matrix}(n,r) \geq M_{original}(n,r)$.

Conversely, any matrix M satisfying conditions as in the matrix formulation can be used to construct a valid request frame, which can be routed using $M_{original}(n,r)$ middle switches. Associating each middle switch with one color, this routing induces a valid coloring for our matrix. Thus, $M_{matrix}(n,r) \leq M_{original}(n,r)$. □

Proposition 4.10. *$M(n,r)$ is invariant under the following assumptions :*

(i) M is doubly stochastic.

(ii) M contains only rational numbers, i.e. M is rationally doubly stochastic.

Proof. (i) Let $M_{ds}(n,r)$ denote $M(n,r)$ when M is doubly stochastic. Given our matrix M, if M is not yet doubly stochastic then clearly there exist a row r and a column c of M whose sums $s(r)$ and $s(c)$ are < 1. Increase the cell at the intersection of r and c by $\min\{1 - s(r), 1 - s(c)\}$ then M is now closer to be doubly stochastic. Repeatedly doing this would make M doubly stochastic after a finite number of steps (at most $2n^2r$). Color the doubly stochastic matrix M' obtained from this procedure using $M_{ds}(n,r)$ colors. Clearly this coloring induces a valid $M_{ds}(n,r)$-coloring of the original M. Hence, $M(n,r) \leq M_{ds}(n,r)$. The other direction is trivial.

(ii) Let $M_{rat}(n,r)$ denote $M(n,r)$ when M contains only rational entries. Obviously $M_{rat}(n,r) \leq M(n,r)$. Conversely, given our matrix M we can reduce all non-rational entries of M by at most ϵ to make them all rational, where ϵ *is small enough* *so that at any block row and block column, if any subset of entries sum up to more than* 1 *then it remains to be more than* 1 *after the perturbation.*

$\square$

Corollary 4.11. *We can assume that* M *is an integral matrix each of whose rows and columns sum up to a common integer* N*. The only difference is that at each block row and block column, the sum of same color cells is now required to be* $\leq N$*.*

To illustrate some good aspect of this matrix approach, let us provide a proof of Theorem 4.2 using the matrix formulation.

A proof of Theorem 4.2. When all weights are chosen from $\{0, 1\}$, M can be assumed to be a permutation matrix by part (i) of Proposition 4.10. Recall that M has dimension $nr \times nr$, we can think of M as a $r \times r$ matrix, each of whose entry is an $n \times n$ matrix, called a block of M. Construct a matrix M' of dimension $r \times r$, where each entry of M' is the sum of all entries in the corresponding block of M. Then, M' is an integral matrix whose line sums are n. Using P. Hall's matching condition, it is easy to show that M' can be written as a sum of n permutation matrices. This decomposition of M' induces a decomposition of M into a sum of n (0,1)-matrices $M_1, M_2, \ldots M_n$ where for each i, no two 1-entries of M_i lie in the same block row or block column of M. Color all 1-entries of M which correspond to an M_i using one color, then we have a valid n-coloring of M. Thus, $M(n, r) \leq n$, which implies $M(n, r) = n$. $\square$

4.3.2 Lower Bounds

When looking for a lower bound of $M(n, r)$, the following proposition is helpful.

Proposition 4.12. *For integers* $k, n, r \geq 1$, $m \geq 0$, *we have* *(i)* $M(n, r+1) \geq M(n, r)$.

(ii) $M(n+1, r) \geq M(n, r) + 1$.

(iii) $M^{k+1}(n+m, r) \geq M^k(n, r) + m$.

Proof. We provide only the proof of (i). (ii) and (iii) can be proved in a similar manner. Given any doubly stochastic $nr \times nr$ matrix M, we construct a doubly stochastic matrix M' by adding to M another block row R and block column C of size n whose entries are all 0's except that at the intersection block of R and C we put an $n \times n$ permutation matrix. Any valid coloring of M' induces a valid coloring of M, thus $M(n, r+1) \geq M(n, r)$ as wanted. □

Theorem 4.13. $M^3(n, 2) \geq \lceil \frac{5n}{4} \rceil$

Proof. We construct a matrix of the following form. Here we give an example when $n = 6$.

$$\begin{array}{cccccc|cccccc}
.6 & 0 & 0 & 0 & 0 & 0 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & .6 & 0 & 0 & 0 & 0 & .4 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & .6 & 0 & 0 & 0 & 0 & .4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & .6 & 0 & 0 & 0 & .4 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & .6 & 0 & 0 & 0 & .4 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & .6 & 0 & 0 & .4 & 0 & 0 & 0 \\
\hline
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}$$

In general, we consider a $2n \times 2n$ matrix M as follows. M has n $.6$'s going along the main diagonal of the second quadrant, n $.4$'s going pair by pair down in the first quadrant, and $\frac{n}{2}$ 1's in the fourth quadrant. To validly color M, all .6's have to be colored differently with a set C of n colors, and all 1's have to be colored differently using a set C' of $\frac{n}{2}$ colors (C and C' are not necessarily disjoint). Let k be the number of $.4$'s which were colored by some colors in C. Since at most two $.4$'s can have the same color, the first block column needs at least $n + \frac{n-k}{2} = \frac{3n}{2} - \frac{k}{2}$ colors. Similarly, the second block column needs at least $\frac{n}{2} + k + \frac{n-k}{2} = n + \frac{k}{2}$ colors. Hence, in total we need at least

$$\max\{\frac{3n}{2} - \frac{k}{2}, n + \frac{k}{2}\} \geq \frac{(\frac{3n}{2} - \frac{k}{2}) + (n + \frac{k}{2})}{2} = \frac{5n}{4}$$

□

Although obvious from Proposition 4.12 and Theorem 4.13, the following is worth mentioning.

Corollary 4.14. $M(n,r) \geq \lceil \frac{5n}{4} \rceil$, *when* $r \geq 2$.

4.3.3 Upper Bounds

Now, we show a result of a different nature than all the existing results on this problem.

Theorem 4.15. $M(2^k, r) \leq 3^k$.

Proof. First, some notations are to be introduced. For any subset C of entries of M, let $s(C)$ denote the sum of all entries in C. For any submatrix A of M, let $(A)_{i*}$ (resp. $(A)_{*j}$) denote the i^{th} (resp. j^{th}) row (resp. column) of A.

We proceed by induction on k. The inequality holds trivially when $k = 0$ by Remark 4.3. Consider a $2^{k+1}r \times 2^{k+1}r$ matrix M. M can be thought of as a $r \times r$ block matrix where each block is a $2^{k+1} \times 2^{k+1}$ matrix. For $p, q \in \{1, \ldots r\}$, let A^{pq} be the $2^{k+1} \times 2^{k+1}$ matrix at the intersection of the p^{th} block row and q^{th} column of M. A^{pq} consists of 4 quadrants of dimensions $2^k \times 2^k$: B_1^{pq}, B_2^{pq}, B_3^{pq}, B_4^{pq}.

Note that $M(n,r)$ is unchanged when permuting rows (columns) within the same block row (block column) of the corresponding matrix. We call these permutations *valid*. Now, validly permute rows and columns of M to maximize the following sum :

$$\sum_{1 \leq p,q \leq r} (s(B_2^{pq}) + s(B_4^{pq}))$$

For convenience, we abuse notation by also denoting the new matrix by M. Now, we construct the following $2^k r \times 2^k r$ matrices :

1. M_1 consists of all blocks B_2^{pq}.

2. M_2 consists of all blocks B_4^{pq}.

3. M_3 consists of all blocks $B_1^{pq} + B_3^{pq}$.

Clearly, we can apply induction on M_1 and M_2. To apply induction on M_3 also, we need to show that each row and column of M_3 sums up to at most 1. This is equivalent to the following conditions :

(i) $\forall 1 \leq p \leq r, 1 \leq i \leq 2^k$,

$$\sum_{q=1}^{r}(s((B_1^{pq})_{i*}) + s((B_3^{pq})_{i*})) \leq 1$$

(ii) $\forall 1 \leq q \leq r, 1 \leq j \leq 2^k$,

$$\sum_{p=1}^{r}(s((B_1^{pq})_{*j}) + s((B_3^{pq})_{*j})) \leq 1$$

To see (i), Let M' be the matrix obtained from M by interchanging rows $(p-1)2^{k+1}+i$ and $(p-1)2^{k+1}+2^k+i$. These are the rows corresponding to $(B_1^{pq})_{i*}$ and $(B_3^{pq})_{i*}$ respectively. $s(M) \geq s(M')$ implies that

$$\sum_{q=1}^{r}(s((B_1^{pq})_{i*}) + s((B_3^{pq})_{i*})) \leq \sum_{q=1}^{r}(s((B_2^{pq})_{i*}) + s((B_4^{pq})_{i*}))$$

But, M is doubly stochastic. Thus,

$$\sum_{q=1}^{r}(s((B_1^{pq})_{i*}) + s((B_3^{pq})_{i*})) + \sum_{q=1}^{r}(s((B_2^{pq})_{i*}) + s((B_4^{pq})_{i*})) = 2$$

This proves (i). Condition (ii) can be proved similarly. Now, applying induction hypothesis on M_1, M_2 and M_3, we can use 3×3^k colors to color M. $\square$

Corollary 4.16. $M(n, r) \leq 3^{\lceil \log_2 n \rceil}$.

Proof. Given a doubly stochastic $nr \times nr$ matrix M. Extend M in the straightforward way to obtain a doubly stochastic matrix M' of order $2^{\lceil \log_2 n \rceil} r$. Any valid coloring of M' induces a valid coloring of M. ☐

Note that Corollary 4.16 beats Theorem 4.5 for $n = 2$ and $n = 4$. In fact, applying Theorem 4.15 with a good example we can show that $M(2, r) = 3$. For practical purposes, small values of n make sense. However, from the theoretical perspective, this is not good at all. The upper bound in Theorem 4.5 is linear, while ours is super linear. However, in the same endeavor to beat existing results when n is small, we also show the following, which is better than all existing results.

Proposition 4.17. $M(3, r) \leq 6$

Proof. The proof is based on a method similar to that of Theorem 4.15's proof. It is simple but tedious to put formally, and thus omitted here. ☐

4.4 Discussions

There is one big aspect of this problem which has not been explored, namely the algorithms to route the requests. Clearly the CAP algorithm [119] is one of them. There are several works along this line such as those in [38, 106]. I have to admit that I have not looked carefully into this. I took one route to go about solving this problem–the route of finding the "best" possible result first, and then look for an algorithm to achieve it later. Clearly, on my way I might come up with some sort of an algorithm (not necessarily polynomial though). After all, I'm not yet so much into *Constructive Mathematics*. Traditional Mathematics (with existential proofs) still makes perfect sense.

One might wonder why Corollary 4.11 was stated but has not been used anywhere. Actually, since all those integral matrices satisfying conditions of Corollary 4.11 can be written as a sum of (at most $N^2 - 2N + 1$) permutation matrices, one might attempt to devise an algorithm which colors our matrix M inductively from these matrices. To show $M(n, r) \leq 2n - 1$ as conjectured by Chung and Ross [42], I actually did this and generated random matrices to test my algorithm. It worked in all the random cases but I haven't been able to show that the algorithm works in general. Basically, the algorithm starts from one permutation matrix (which can be colored with n colors), and inductively modifies its coloring upon adding a new permutation matrix into the current matrix until the entire M is obtained.

There are several natural extensions to this problem that one might consider. Firstly, Du et al. [52] extended the bipartite graph formulation to the following problem. Let $G = (V, E, W)$ denote a weighted multigraph where each $e \in E$ is associated with a weight $w(e) \in W$, $b < w(e) \leq B$, and b and B are constants satisfying $0 < b \leq B \leq 1$. The edges are to be colored such that for each vertex v and each color c, $w_c(v) \leq 1$. The problem is to minimize the number of colors. When $b = B = 1$, this number is $\chi(G)$, the chromatic number of G. They quoted : "Surprisingly, this very natural extension of the edge-coloring problem seems to have been neglected in the literature". They also showed that this number is $\leq \lceil \frac{17n-5}{6} \rceil$.

In the same line of thought, I think there is another "natural" extension of this problem in the bipartite case, which would lead to generalizations of several classical results in Matching Theory. Before this report, all the results as those in [52,108,119] made extensive use of the following theorem by König. The book by Lovász and Plummer [111] contains most of the references related to the theorems stated in this section.

Theorem 4.18 (König's Line Coloring Theorem, 1916). *For every bipartite graph G, $\chi_e(G) = \Delta(G)$.*

Here $\chi_e(G)$ is the chromatic index of G, i.e. $\chi_e(G)$ is the minimum integer so that a $\chi_e(G)$-edge-coloring of G exists, and $\Delta(G)$ is the maximum degree of all vertices in G. This theorem can be proved "easily" using one of the following three theorems.

Theorem 4.19 (König's Minimax Theorem, 1975). *If G is bipartite, then $\tau(G) = \nu(G)$.*

Here $\tau(G)$ is the matching number, i.e. the size of the maximum matching, and $\nu(G)$ is the vertex covering number, i.e. the size of the minimum vertex cover, of G.

Theorem 4.20 (P. Hall, 1935). *Let $G = (A, B)$ be a bipartite graph. Then G has a matching of A into B if and only if $|\Gamma(X)| \geq |X|$ for all $X \subseteq A$.*

Theorem 4.21 (Frobenius, 1917). *Let $G = (A, B)$ be a bipartite graph. Then G has a complete matching if and only if $|A| = |B|$ and $|\Gamma(X)| \geq |X|$ for all $X \subseteq A$.*

Frobenius' Theorem is often called the *Marriage Theorem*. It is interesting to note that all three theorems are equivalent, and the proof of their equivalences isn't so hard to find. It is even more interesting to know that they are all equivalent to the celebrated Dilworth's theorem :

Theorem 4.22 (Dilworth, 1950). *In any finite poset, the size of any largest antichain equals the size of any smallest chain decomposition.*

Although these are interesting results, one might wonder what they have to do at all with our problem. Well, may be not all of them are related (for now). But first, let's look at Theorem 4.18. An obvious corollary of this theorem would be that for any m-regular bipartite graph G, the minimum number of matchings G can be decomposed into is m (and thus these matchings have to be all perfect). Our problem adds weights to everything. Let's define a w-matching of a weighted bipartite graph G to be a subgraph H of G such that at any $v \in V(H)$, total weights of edges incident to v is ≤ 1. Our problem in general is to look for the minimum number of w-matchings a weighted bipartite graph can be decomposed into. There is another constraint to the problem, which says that at any vertex v of G, all edges can be partitioned into n groups of weight sums ≤ 1. If each of these n groups sums up to 1 (as in the doubly stochastic case), we obviously have an w-n-regular bipartite graph. As we have seen, the minimum number of w-matchings that G can be decomposed into is $\geq \frac{5n}{4}$.

Theorem 4.18 can be obtained from any of P. Hall's, König's Minimax, Frobenius' or Dilworth's Theorems. Hence, a natural question to raise is : *is there any weighted version of one of these theorem which would give us the exact value of $M(n, r)$?* P. Hall's Theorem is the most obvious candidate to start with. For example, given a weighted bipartite graph $G = (A, B)$ with all weights ≤ 1, we define a w-matching from A into B to be a w-matching H of G such that $A \subseteq V(H)$ and for all $v \in A$, $0 < w_H(v) \leq 1$. Here $w_H(v)$ denotes the sum of all edges incident to v in H. To extend P. Hall's theorem, we would like to find a condition on G for a w-matching from A into B to exist. It is not obvious how to extend the condition $|\Gamma(X)| \geq |X|,\ \forall X \subseteq A$ to a weighted version, let alone using that extension to show $M(n, r) = 2n - 1$. Some studies along this line would be interesting.

Lastly, a weighted version of Dilworth's theorem would be really amazing to see. To conclude, *I strongly believe that $M(n, r) = 2n - 1$.*

Chapter 5

Multicast Nonblocking and the DHH-Erdős Conjecture

5.1 Overview

In this chapter, we go back to the 1-rate model and study the relationship between a conjecture by Hwang and Lin (1995, [91]) about the multicast nonblocking properties of 3-stage Clos networks and a conjecture by Du, Hsu, Hwang [47] and its extension by Paul Erdős. We shall avoid the already boring path of presenting network related material first and mathematical material later. Instead, we jump right into these fascinating conjectures and give the application on interconnection networks later.

Consider n disjoint triangles and a cycle on the $3n$ vertices of the n triangles. The union of the n triangles and the cycle is called a *cycle-plus-triangle graph*. In 1986, Du, Hsu, and Hwang [48][1] conjectured that every cycle-plus-triangle graph has independent number n. i.e. the maximum independent set contains n vertices. Soon later, Paul Erdős got interested in this conjecture and improved it to a stronger version that every cycle-plus-triangle graph is 3-colorable. Due to Erdős' promotion in his frequent traveling, this conjecture becomes quite well-known during the past ten years. There were several efforts [6, 66] to attack the conjecture and it was finally proved by H. Fleischner and M. Stiebitz [70].

In this chapter, we give an extension of the above conjecture with an application in

[1] This paper was accepted in 1986 but published in 1993.

switching networks.

5.2 Extension

Let us first consider a slightly more general situation. Instead of a cycle, consider a union of disjoint cycles on the $3n$ vertices of the n triangles. That is, we consider a graph G constructed by taking the union of n disjoint triangles and a disjoint union of cycles on the $3n$ vertices of the n triangle.

Is G still 3-colorable? The answer is MAY BE NOT. In fact, the graph in Figure 5.1 is not 3-colorable since it contains a clique of size four. But it can be obtained by taking union of four disjoint triangles and a union of three cycles of size four.

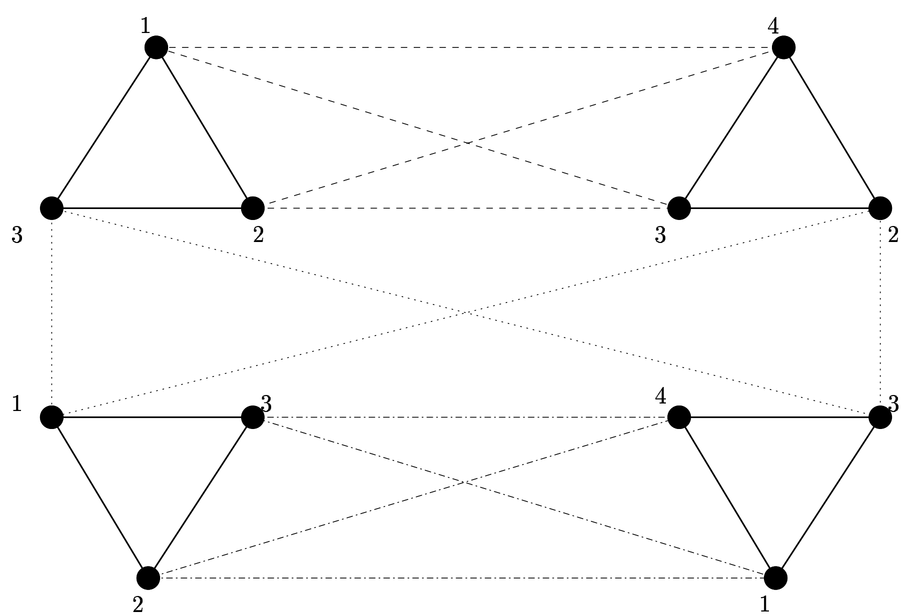


Figure 5.1: Not 3-colorable but 4-colorable

This example also shows that a similar conjecture made in [51] is false. The conjecture is as follows. Consider the line graph G of a d-regular graph. Partition the vertex set of G

into disjoint subsets of size exactly n with $d \le n \le 2d-1$ and for each subset, construct a clique on it, with possible reuses of original edges of G. Then, the resulting graph G^* is $2d-1$ vertex-colorable. In the above counterexample, we have $d = 2$ and $n = 3$. G consists of three cycles of size 4 and its vertices are divided into four subsets of size exactly n. But, G^* is not $(2d-1)$-colorable.

The graph in Figure 5.1 is 4-colorable. In general, is G 4-colorable? The answer is YES. In fact, every vertex in G has degree four. It is well-known that a connected 4-regular graph is 4-colorable unless it is a complete graph of order five [24]. Clearly, G cannot have a connected component of size five. Therefore, G is 4-colorable.

The above observation suggests the following conjecture.

Conjecture 5.1. Consider a graph H with maximum degree m. Let $L(H)$ be the line graph of H. Divide all vertices of $L(H)$ into disjoint groups of size at most n. Connect all vertices in each group into a clique. If $m \le n$, then the resulting graph is $(m+n)$-colorable.

The reader may have question on the coloring number $m+n$. Why do we use $m+n$ instead of $m+n-1$? In fact, for the above example, we have $m = 2$ and $n = 3$ and the resulting graph is $(m+n-1)$-colorable. The following example may provide an explanation.

Let H be a complete graph of order four. Let a, b, c, d be vertices of H. The line graph $L(H)$ of H contains six vertices ab, cd, ac, bd, ad, bc. Now, we divide them into three groups $\{ab, cd\}$, $\{ac, bd\}$, $\{ad, bc\}$. Connect every two vertices in the same group with an edge. The resulting graph G is a complete graph of order six. Thus, it cannot be 5-colorable. However, we can have $m = 3$, $n = 3$ and $m + n - 1 = 5$ (note: $m \le n$). Actually, in this example, each group has size 2 (less than 3). Therefore, this is also an example to explain why we need condition $m \le n$. In fact, if we remove the condition

$m \leq n$, then the example fits the condition $m = 3$ and $n = 2$. In this case, $m + n = 5$. However, G is not 5-colorable.

Theorem 5.2. *Conjecture 5.1 holds for $m = 2$ and 3.*

Proof. It is a well-known fact that every graph with maximum degree $\Delta \geq 3$ must be $(\Delta + 1)$-colorable and furthermore, it is Δ-colorable unless the graph contains a subgraph isomorphic to the complete graph of order $\Delta + 1$, i.e. a clique of size $\Delta + 1$ [24]. Note that the resulting graph in Conjecture 5.1 has maximum degree $2(m - 1) + n - 1$. For $m = 2$, $(2(m - 1) + n - 1) + 1 = m + n$ and for $m = 3$, $2(m - 1) + n - 1 = m + n$. Thus, it suffices to show that for $m = 3$, the resulting graph does not contain a clique of size $n + 4$. For contradiction, suppose that the resulting graph contains a clique Q of size $n + 4$. Since it has maximum degree $n + 3$, the clique Q must be a connected component of it. Thus, we may assume, without loss of generality, that the resulting graph itself is the clique Q. Now, we want to prove that Q can not be obtained in the way described in Conjecture 5.1. To do so, we consider the problem of removing disjoint cliques of size at most n to obtain a graph with maximum degree at most $2(m - 1) = 4$. Since every vertex in Q has degree $m + n = n + 3$, each removed clique has to have size n in order to have degree $n - 1$ at each vertex. It follows that $n \mid (n + 4)$. Since $n \geq m = 3$, we must have $n = 4$. Thus, Q is a clique of size 8. Removing two disjoint cliques of size 4 from Q results in a graph P as shown in Figure 5.2. This graph P cannot be the line graph $L(H)$ of a graph H with maximum degree at most three. In fact, P is 4-regular. If $P = L(H)$, then H must be 3-regular. So, each vertex of P must be adjacent to four vertices which can be divided into two pairs such that vertices in the same pair are adjacent. However, this is not true to P, a contradiction. □

Now, we propose a direct generalization of DHH-Erdös conjecture as follows.

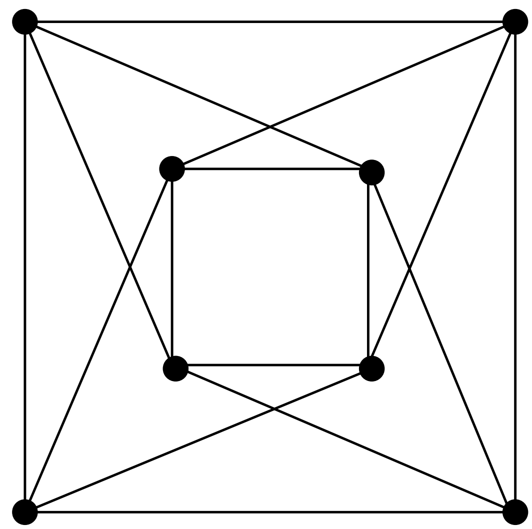


Figure 5.2: Graph P.

Conjecture 5.3. Consider a m-regular m-connected graph H. Let $L(H)$ be the line graph of H. Suppose all vertices of $L(H)$ can be divided into disjoint groups of size exactly n. Add a clique of size n on vertices in each group. The resulting graph is $(m+n-2)$-colorable.

The results in [66, 70] shows that this conjecture holds for $m = 2$.

5.3 An Application to Switching Networks

Conjecture 5.1 has an application in switching networks. To see it, let us first introduce some concepts in switching networks. Part of the following discussion has been presented in the previous chapter. However, to keep this chapter relatively self-contained we restate some definitions here.

Consider the three stage Clos network $C(n_1, r_1, m, n_2, r_2)$ as defined in section 4.1. See figure 5.3 There are totally $r_1 n_1$ inlets in the first stage and totally $r_2 n_2$ outlets in the

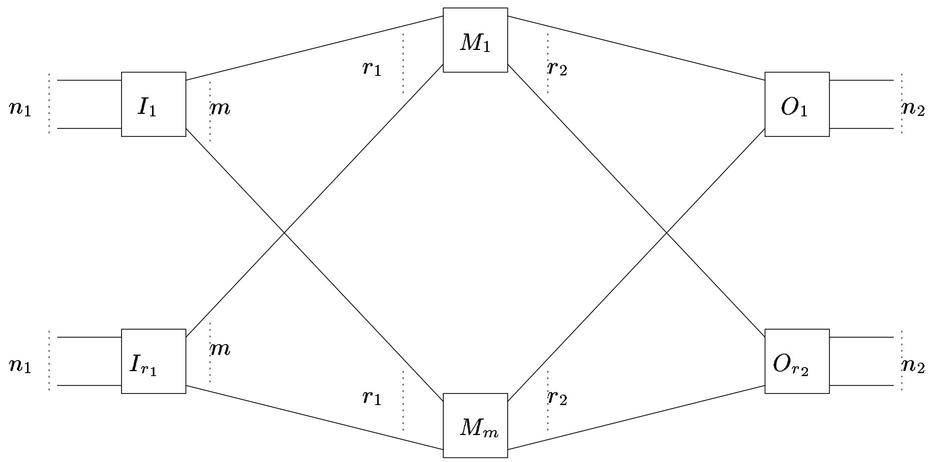


Figure 5.3: The 3-stage Clos network $C(n_1, r_1, m, n_2, r_2)$

third stage. Denote by I the set of all r_1n_1 inlets in the first stage and by O the set of all r_2n_2 outlets in the third stage. Then a *connection* in a three-stage Clos network is a pair (x, y), where $x \in I$ and $y \in O$. A *route* is a path in the network joining an input crossbar (i.e., a crossbar in the first stage) to an output crossbar (i.e., a crossbar in the third stage) and a route r *realizes* a connection (x, y) if x and y belong to the input crossbar and the output crossbar joined by r, respectively.

A set of connections is *compatible* if for every input switch I_i, there are at most n_1 connections involving I_i, and for every output switch O_j, there are at most n_2 connections involving O_j. In other words, *multicasting* is involved. A *configuration* is a set of routes and it is *compatible* if every edge in the network is used only once. A set of connections is said to be *realizable* if there exists a compatible configuration which contains routes realizing all connections in the set. A network is said to be *rearrangeable* if every compatible set of connections is realizable. It is well-known that a three-stage Clos network $C(n_1, r_1, m, n_2, r_2)$ is rearrangeable if and only if $\min(n_1, n_2) \le m$ [90]. A connection c is said to be *compatible* with a compatible set C of connections if $C \cup \{c\}$ is still compatible. A route r is said to be *compatible* with a compatible configuration R if $R \cup \{r\}$ is still compatible. A network is said to be *strictly nonblocking* if for every compatible configuration R realizing a connection set C and every connection c compatible with C, there exists a route r such that r realizes c and is compatible with R. A three-stage Clos network $C(n_1, n_3, r_1, r_2, r_3)$ is strictly nonblocking if and only if $m \geq n_1 + n_3 - 1$.

The above concepts can be easily extended to one-to-many connections, i.e. when *multicasting* is involved. A 1-to-k connection is a $(k+1)$-tuple $(x; y_1, y_2 \cdots, y_k)$ where $x \in I$ and $y_1, y_2, \cdots, y_k \in O$.

Note that for those y_j's lying in the same output crossbar, the path from x can branch at that output crossbar to reach these y_j's. But for y_j's lying in different output crossbars, the branching has to take place either in the input crossbar or at a center crossbar. It is well known [90] that if branching at input crossbars is allowed, then the symmetric 3-stage Clos network $C(n, m, r) = C(n, r, m, n, r)$ is rearrangeable for 1-to-k connections if $m \geq kn$. However, the case that input switches do not have the branching capability remains an open problem. Hwang and Lin [91] conjectured that $C(n, m, r)$ is rearrangeable for 1-to-2 connections and meanwhile strictly nonblocking for 1-to-1 connection if $m \geq 2n$. This conjecture can be extended to asymmetric three-stage Clos networks. In fact, this extension has connection to Conjecture 1.

Theorem 5.4. *Suppose Conjecture 5.1 holds. If $m \geq n_1 + n_2$ and $n_1 \geq n_2$, then $C(n_1, r_1, m, n_2, r_2)$ is rearrangeable for 1-to-2 connections.*

Proof. Suppose $\{(x_i; y_{2i-1}, y_{2i})\}$ is a set of compatible 1-to-2 connections. That is, at most n_1 x_i's are the same and at most n_3 y_j's are the same.

First, consider the case that for each 1-to-2 connection $(x_i; y_{2i-1}, y_{2i})$, $y_{2i-1} \neq y_{2i}$. Let H be the graph with vertex set O and edge set $\{(y_{2i-1}, y_{2i})\}$. Then H has maximum degree at most n_3. Let $L(H)$ be the line graph of H. Divide all vertices of $L(H)$ into disjoint groups such that two vertices (y_{2i-1}, y_{2i}) and (y_{2j-1}, y_{2j}) are in the same group if and only if $x_i = x_j$. Thus, each group has size at most n_1. Connect all vertices in each group into a clique. Since Conjecture 5.1 is assumed to be true, the resulting graph is $(n_1 + n_2) - colorable$. Note that if two vertices (y_{2i-1}, y_{2i}) and (y_{2j-1}, y_{2j}) are of the same color, then we must have $x_i \neq x_j$. Note that each vertex (y_{2i-1}, y_{2i}) represents a connection $(x_i; y_{2i-1}, y_{2i})$. Therefore, if we arrange all 1-to-2 connections in the same color to pass through the same middle switch, then each input switch has at most one connection to this middle switch. Moreover, the middle switch has at most one connection to each output switch since two 1-to-2 connections have the same component in output switches must be adjacent in $L(H)$. Therefore, if $r_2 \geq n_1 + n_3$, $C(n_1, n_3, r_1, r_2, r_3)$ is rearrangeable for 1-to-2 connections.

Now, we consider the general case. If $y_{2i-1} = y_{2i}$ for some 1-to-2 connection $(x_i; y_{2i-1}, y_{2i})$, then we may add a new output switch and change y_{2i} to the new output switch. In this way, we can reduce the general case to the first case. □

5.4 Discussions

The connection of Conjecture 5.1 to the rearrangeability for 1-to-2 connections may suggest a generalization of Conjecture 5.1, corresponding to the rearrangeability for 1-to-k connections.

Let V_1 and V_2 be two disjoint sets of vertices. A (h, k)-bipartite hypergraph (V_1, V_2, E) is a hypergraph such that each hyper-edge $e \in E$ contains at most h vertices in V_1 and at most k vertices in V_2. The degree of each vertex is the number of hyper-edges containing the vertex. The generalization can be stated as follows: Consider a $(1, k)$-bipartite hypergraph $G(V_1, V_2, E)$. Suppose each vertex in V_1 has degree at most n and each vertex in V_2 has degree at most m. If $m \leq n$, then G is $(n+1+(k-1)(m-1))$-edge-colorable, i.e. all hyper-edges of G can be in $(n+1+(k-1)(m-1))$ colors such that any two hyper-edges in the same color are not adjacent.

Unfortunately, this generalization is false. The following is a counterexample.

Choose $V_1 = \{I_1, I_2, I_3\}$ and $V_2 = \{O_1, O_2, \cdots, O_9\}$. Consider the following edge set E:

$$\begin{array}{ccc} (I_1; O_1, O_2, O_3) & (I_1; O_4, O_5, O_6) & (I_1; O_7, O_8, O_9) \\ (I_2; O_1, O_4, O_7) & (I_2; O_2, O_5, O_8) & (I_2; O_3, O_6, O_9) \\ (I_3; O_1, O_5, O_8) & (I_3; O_2, O_4, O_9) & (I_3; O_3, O_5, O_7) \end{array}$$

Then we have $k = m = n = 3$. But, G is not 8-edge-colorable. In fact, the edge graph of G is a complete graph of order 9.

Part III

Direct Networks

Chapter 6

Connectivity of Consecutive-d Digraphs

6.1 Overview

De Bruijn graphs [44], Kautz graphs [96] and their generalizations have been extensively studied [17, 19–22, 30, 41, 47–50, 53, 68, 81, 82, 87, 92, 93]. It was stated in [17] that these graphs are competitive topological structures for interconnection networks of computers and multiprocessor systems. For a nice survey, the reader is referred to [20] and [19].

For integers d, n, q, r satisfying $0 < d \leq n$, $-n/2 < q \leq n/2$, and $q \neq 0$, a consecutive-d digraph $G(d, n, q, r)$ (as defined in [48]) has n nodes, labeled by integers mod n, with edges from each node i to d consecutive nodes, which are those with labels $qi+r+k \pmod{n}$. The concept of the consecutive-d digraph generalizes many interconnection networks of computers and multiprocessor systems. The generalized de Bruijn digraphs [92, 139] and the generalized Kautz digraphs [93] are its two useful subclasses consisting of $G_B(d, n) = G(d, n, d, 0)$ and $G_I(d, n) = G(d, n, n-d, n-d)$, respectively. The following results on connectivity of $G_B(d, n)$, $G_I(d, n)$, and $G(d, n, q, r)$ are known.

(1) If $n \geq d^3$, $G_B(d, n)$ and $G_I(d, n)$ are $(d-1)$-connected (Imase, Soneoka and Okada [94].)

(2) If $n \geq d^4$, then $G_I(d, n)$ is d connected iff $(d+1) \mid n$ and $gcd(d, n) > 1$ (Homobono and Peyrat [82].)

(3) If $gcd(n,q)=d$ and $n>d^2$, then $G(d,n,q,r)$ is at least ($d-1$)-connected and it is d-connected iff it has no loop. (Du, Hsu, and Peck [50])

In this chapter, we determine the connectivity of $G(d,n,q,r)$ in almost all cases, and as corollaries, remove condition $n \geq d^4$ on n from the result of Homobono and Peyrat [82], significantly relax condition $n \geq d^3$ from the result of Imase, Soneoka and Okada [94]. In addition, we also study how to modify $G_B(d,n)$ to get a d-connected digraph by replacing all loops with a cycle or a set of disjoint cycles.

6.2 Preliminaries

Let $\psi = gcd(q-1,n)$. (note: $\psi = n$ if $q=1$.) Denote by $(x)_n$ the residue of x modulo n, represented by a number in $\{0,1,\ldots,n-1\}$. An edge is said to be with k-value i, where $0 \leq i < d$, if it is contained in the subgraph $G(1,n,q,r+i)$. The following lemma can be found in [54].

Lemma 6.1. *$G(d,n,q,r)$ has the following properties:*

(a) Each node has at most one loop.

(b) $G(d,n,q,r)$ has no loop iff $0 < (r)_\psi \leq \psi - d$.

(c) If $d < \psi$, then all loops of $G(d,n,q,r)$ are with the same k-value.

(d) If $\psi = 1$, then for each k-value there exists exactly one loop with the k-value. If $\psi > 1$, then for each k-value, either there is no loop or there are exactly ψ loops with the k-value. Moreover, if i is a loop-node, then the ψ loop-nodes are i, $i+n/\psi$, $\cdots$, $i+(\psi-1)n/\psi$. In particular, if $d \geq 2$ then $G(d,n,q,r)$ has either no loop or at least two loops.

(e) *If $|q-1| \leq d$ and x is a loop-node, then either $x + \lfloor n/(q-1) \rfloor$ or $x + \lceil n/(q-1) \rceil$ is a loop-node.*

In particular, notice that if $d \geq \psi$ then there always exists a loop. Next, we show a new lemma concerning loops in $G_B(d, n)$.

Lemma 6.2. *Consider $G_B(d, n)$, where $d > 1$. Let x and y be two distinct loop-nodes. Then either $|x - y| = 1$ or $\frac{n}{(d-1)} - 1 \leq |x - y| \leq n - (\frac{n}{(d-1)} - 1)$. Moreover, if $|x - y| = 1$, then the loop at x and the loop at y are with k-values 0 and $d - 1$.*

Proof. Since x and y are loop-nodes, we have

$$(d-1)x + k \equiv 0 \pmod{n} \tag{6.1}$$

$$(d-1)y + k' \equiv 0 \pmod{n} \tag{6.2}$$

for $0 \leq k, k' \leq d-1$. Without loss of generality, assume $k \leq k'$. Then $0 \leq k' - k \leq d-1$. Subtracting (6.2) from (6.1), we obtain

$$(d-1)(x-y) \equiv k' - k \pmod{n}$$

If $|x-y| = 1$, then $k' - k = d-1$ and we must have $k' = d-1$ and $k = 0$. If $|x-y| > 1$, then $(d-1)(x-y) = k' - k + \ell n$ for some nonzero integer ℓ and hence $|x-y| \geq \frac{n}{(d-1)} - 1$. It is also easy to see that $|l| \leq (d-2)$, so that $|x-y| \leq \frac{|l|}{d-1} + 1 \leq n - (\frac{n}{(d-1)} - 1)$. □

From the above two lemmas, it is easy to see that $G_B(d, n)$ has exactly $d - 1 + \psi$ loop-nodes. 2ψ of them form ψ pairs of adjacent nodes. The rest of them are isolated. These $d - 1$ "groups" of size 1 or 2 are almost evenly distributed in Z_n with "distance" at least $\frac{n}{(d-1)} - 1$ apart.

6.3 Consecutive Runs

In this section we show two lemmas which are important in studying the connectivity of consecutive-d digraphs.

A subset of $\mathbb{Z}_n$ is called a *consecutive run* if its elements can be consecutively numbered mod n. For convenience, we call the d out-edges from the node a *claw* and the set of end points of the claw a *claw-end*. In a consecutive-d digraph a node's claw-end forms a consecutive run of size d. Let $g = gcd(n, q)$. Denote $\bar{i} = \{i, i+n/g, \cdots, i+(g-1)n/g\}$. Then all nodes in $\bar{i}$ have the same set of successors. Each $\bar{i}$ will be called an *orbit*. Denote $\hat{i} = \{ig+r, ig+r+1, \cdots, ig+r+g-1\}$. Then all nodes in $\hat{i}$ have the same set of predecessors. Each $\hat{i}$ will be called a *block*.

A good way to visualize orbit, claws, blocks and consecutive runs is to think of $\mathbb{Z}_n$ as n points $\{0, 1, \ldots n-1\}$ put clockwise on a circle, spaced equally. An orbit is then a set of g equally-spaced points on the circle with interval n/g. Note that $\mathbb{Z}_n$ is partitioned uniformly into orbits. The circle is also partitioned into $\frac{n}{g}$ arcs (or consecutive runs) of size g, which are our blocks. The start of each block ($ig + r$, for some $i \in \mathbb{Z}_n$) is also the start of some claw-end (because there is always an $j \in \mathbb{Z}_n$ such that $jq + r \equiv ig + r \pmod{n}$). Claw-ends from different orbits start at different positions on the $\mathbb{Z}_n$-circle.

Since each orbits is of size g, the in-degree of a node of $G(d, n, q, r)$ must be divisible by g. Thus, if the in-degree of a node is d, then we must have $g \mid d$. It was proved in [50] that $g \mid d$ iff the in-degree of every node is d. Throughout this chapter, we assume $g \mid d$, i.e., the in-degree of every node is d. To emphasize this, we may still mention this condition in the statements of lemmas and theorems. Also note that when $g \mid d$, each claw-end contains exactly d/g blocks.

Lemma 6.3 (Consecutivity Lemma). *Suppose $g \mid d$ and $g < d$. Let C, D, E be a parti-*

tion of the node set of $G(d,n,q,r)$ such that removal of all nodes in E leaves no path from any node in C to any node in D. Let S be the union of all claw ends from nodes in C. If $|E| < d$, then S ($\subseteq C \cup E$) is a consecutive run of size at least $|C| + d - g$.

Proof. Suppose there are y orbits which intersect with C. Each such orbit contributes a consecutive run of size d in S, which we shall call a C*-run*. A consecutive run R in S is *maximal* if no other consecutive run in S properly contains R. Let x be the number of maximal consecutive runs in S. Let R be any maximal consecutive run in S, and k be the number of different C-runs which are *entirely* contained in R. Then since each C-run starts at the beginning of some block, we must have $1 + (k-1)g + d - 1 \leq |R|$. In other words, each maximal consecutive run R in S contains at most $|R|/g - (d/g - 1)$ different C-runs. Summing over all maximal consecutive runs in S, we get $|S|/g - x(d/g-1) \geq y$. Hence, S has at least $gy + x(d-g)$ elements. Since $S \subseteq C \cup E$ and $|E| \leq d-1$, we have

$$gy + x(d-g) \leq |C| + d - 1.$$

Note that $gy \geq |C|$. If $g = 1$, then it is clear that $x = 1$. If $d > g > 1$, then $d - g \geq d/2$ since $g \mid d$. Thus, $x = 1$. Finally, $x = 1$ implies that S is a consecutive run and $|S| \geq gy + d - g \geq |C| + d - g$. $\square$

From the consecutivity lemma, it is easy to see that $|E| \geq d - g$. This means that if $g \mid d$ and $g < d$, then $G(d,n,q,r)$ is at least ($d-g$)-connected.

Lemma 6.4. *Let R be a consecutive run of $\mathbb{Z}_n$. Suppose $S = \{a, a+h, \cdots, a+(c-1)h\} \subseteq R$ for some natural numbers a, c and h. Let f be any function so that for each $x \in S$, $f(x)$ is a consecutive run of size d in R. Further assume that $a + ih \in \bigcup_{j=0}^{c-1} f(a+jh)$, $\forall i = 0, \ldots c-1$. Then we have* *(i) If* $|f(a+ih) \cap f(a+(i+1)h)| \geq h$ *for* $i = 0, \cdots, c-2$*, then there exists an* $i \in \mathbb{Z}_{c-1}$*, such that* $a + ih \in f(a + ih)$*.*

(ii) If $|f(a + ih) \cap f(a + (i + 1)h)| \geq h - 1$ *for* $i = 0, \cdots, c - 2$*, then there exists an* $i \in \mathbb{Z}_{c-1}$ *such that* either $a+ih \in f(a+ih)$ or $f(a+ih) = \{a+ih+1, \cdots, a+ih+d\}$ *and* $f(a + (i + 1)h) = \{a + (i + 1)h - d, \cdots, a + (i + 1)h - 1\}$*.*

Proof. (i) Let

$$\begin{aligned} A &= \{a + ih \mid x < a + ih,\ \forall x \in f(a + ih)\} \\ B &= \{a + ih \mid x > a + ih,\ \forall x \in f(a + ih)\} \end{aligned}$$

Suppose to the contrary that such an i does not exist; then since $a \in f(a + ih)$ for some i and $a+(c-1)h \in f(a+jh)$ for some j, both A and B are not empty. It follows that there exists an i such that $a+ih$ and $a+(i+1)h$ are not both in A nor both in B. If $a+ih \in A$ and $a+(i+1)h \in B$, then $f(a+ih) \cap f(a+(i+1)h) = \emptyset$, contradicting $|f(a + ih) \cap f(a + (i + 1)h)| \geq |h| > 0$. If $a + ih \in B$ and $a + (i + 1)h \in A$, then $f(a + ih) \cap f(a + (i + 1)h)$ lies between $a + ih + 1$ and $a + (i + 1)h - 1$, so that $|f(a + ih) \cap f(a + (i + 1)h)| < h$, contradicting $|f(a + ih) \cap f(a + (i + 1)h)| \geq h$, too.

(ii) Similarly, we can prove the second half of the lemma.

□

6.4 Connectivity

In this section we determine the connectivity of consecutive-d digraphs. The results are described by two theorems, as consequences of which the results of Imase, Soneoka and Okada [94] and Homobono and Peyrat [82] are extended to smaller n. The approaches we use here differ very much from theirs.

Theorem 6.5. *If $g \mid d$ and $1 < g < d$, then $G(d, n, q, r)$ is at least ($d - g$)-connected and it is d-connected iff it has no loop.*

Proof. The first half has been proven in the last section. We prove the second half here.

If our graph has a loop at i, then removing $d - 1$ nodes other than i from the claw-end of i disconnects i to the rest of the graph. In other words, if $G(d, n, q, r)$ is d-connected then it has no loop.

For the other direction, let E be a node-cut of the smallest size, which disconnects D from C, i.e. removal of all the nodes in E leaves no path from nodes in C to those in D. Assume $|E| \leq d - 1$. We will prove the existence of a loop.

Let S be the union of claw-ends from nodes in C. By Lemma 6.3, S is a consecutive run of size at least $|C| + d - g$. Without loss of generality, we may assume that all nodes not in S are in D, since otherwise they can be moved into D without increasing the size of the node-cut E. Thus, $S = C \cup E$. Since S is a consecutive run, so is its complement D. The following facts are important in the remainder of the proof.

(i) For any consecutive run R of $\mathbb{Z}_n$ and any orbit O, there are at least $\lfloor \frac{|R|g}{n} \rfloor$ elements of O in R, and at least $g - \lceil \frac{|R|g}{n} \rceil$ elements of O not in R.

(ii) Every claw from E catches some node in D. (Otherwise, E can be decreased, contradicting the minimality of E.)

(iii) If an orbit contains an element of C, then it contains no element of E. (Otherwise, E can be decreased by putting such elements into C.) An orbit having an element in C (E) is called a C-*orbit* (E-orbit). Notice that no claw-end from any C-orbit intersects D.

(iv) D has at most $g-1$ elements in C-orbits. (Otherwise, putting all such elements into C does not change the set E, but makes $|E|+|C|-|S|>g-1$, contradicting $|S|\geq|C|+d-g$.)

Now, for any $k\in\mathbb{Z}_n$, let k^* denote the integer between 0 and $n/2$ such that $k\equiv k^*$ or $-k^*\pmod n$. k^* is called the *magnitude* of k. To prove the existence of a loop, we may assume $q^*\geq d$ since if $q^*<d$, then certainly $\psi<d$, so that a loop must exist by Lemma 6.1. We consider three cases based on where $|D|$ lies between 0 and n.

Case 1: $(g-a)n/g\leq|D|<(g-a+1)n/g$ for some $a=0,\cdots,\lfloor g/2\rfloor$. By (i), each C-orbit contains at least $g-a$ elements in D. Let y be the number of C-orbits, then D has at least $y(g-a)$ elements in C-orbits. By (iv), $y(g-a)\leq g-1$. So, $y=1$. It follows that $|S|=d$. Since $S=C\cup E$, $|E|\leq d-1$ and the claw-end of our C-orbit has size d, every node in C has a loop.

Case 2: $an/g<|D|\leq(a+1)n/g$ for some $a=0,\cdots,\lfloor g/2\rfloor-1$. Again, by (i) we have that of g elements in an orbit, at least a must be in D and at least $g-a-1$ must not be in D. It is not hard to see that there are at least $\lceil\frac{(|D|+d-1)}{g}\rceil$ orbits whose claw-ends intersect D, because each starting point of a block in D is also the starting point of a claw-end and the size of a claw-end, d, is greater than the size of a block, g. These orbits have to be disjoint from C, hence E has at least $(g-a-1)\lceil(|D|+$

$d-1)/g\rceil$ elements. So, $d-1 \geq (g-a-1)\lceil(|D|+d-1)/g\rceil$, i.e.

$$|D| \leq (d-1)(a+1)/(g-a-1) \leq d-1.$$

We will prove that D must contain a loop node. Notice that since S is a union of claw-ends and $g \mid d$, S is also a union of (consecutive) blocks. Hence, D is a union of consecutive blocks, too. Let B be the rightmost block of D (clockwise). If B contains a loop node, then we are done. Thus, we may assume that B has no loop node. For any node i, let $ce(i)$ denote the claw-end from i. The inequality $d \leq q^* \leq n/2$ implies that if for some $i \in B$, $ce(i)$ has its right end point in $D \setminus B$ then the right end point of $ce(i+1)$ is not in B, neither is the left end point of $ce(i+1)$ since B has no loop. In other words, no two consecutive claw-ends from B both intersect D. Hence, B has at most $\lceil g/2\rceil$ elements whose claw-ends intersect D. Let O be the orbit whose claw-end intersects D only in B. Then $O \subset (D-B) \cup E$. As we have noticed, there are at least $g-a-1$ elements of O not in D. These elements have to be in E, so $|O \cap E| \geq g-a-1 \geq g-(\lfloor \frac{g}{2} \rfloor -1)-1 = \lceil g/2\rceil$.

When B is removed from D, only the nodes whose claw-ends intersect $D \setminus B$ (there are at most $\lceil g/2\rceil$ of these nodes) have to be moved into E and others can be moved into C. However, all elements in $O \cap E$ can be moved from E to C. Thus, this move does not increase $|E|$. In this way, we can reduce D to have only one block. However, as $g \mid d$ each node in this block has d in-edges. One of them must be from a node in D, which forms a loop.

Case 3: $\lfloor g/2\rfloor n/g \leq |D| < \lceil g/2\rceil n/g$. This case exists only for g odd and at most two C-orbits exist by the same argument as that in Case 1. If there exists only one orbit, we can prove, as in Case 1, that each of the nodes in C has a loop. If there are two C-orbits, then each C-orbit must have at most $(g+1)/2$ elements in C since it has at least $(g-1)/2$ elements in D. So, $|E \cup C| \leq d - 1 + 2\frac{g+1}{2} = d + g$. It follows that the claw from each node in a C-orbit can miss only one block in $E \cup C$. If C has no loop, then we must have $|E \cup C| = |S| = d + g$ and hence each C-orbit has exactly $(g+1)/2$ elements in C. Furthermore, $(g+1)/2$ elements of C in a C-orbit must fit in a block which is not contained in any claw-end of this C-orbit. Thus, $1 + (\frac{g+1}{2} - 1)n/g \leq g$. So, $n \leq 2g \leq d$. Thus, $q^* < d$, a loop must exist.

□

Theorem 6.6. *If $g = 1$, then $G(d, n, q, r)$ is at least $(d-1)$-connected. Moreover, if $n > 3d$ then it is d-connected if and only if none of the following occurs :*

(1) It has a loop.

(2) $r \equiv 1 \pmod{(d+1)}$ *and* $q \equiv -d \pmod{n}$.

(3) $r \equiv 1 \pmod{(d+1)}$ *and* $qd \equiv -1 \pmod{n}$.

Proof. Let E be any node-cut such that removal of all nodes in E leaves no path from C to D. When $d = 1$, the theorem trivially holds; thus, we assume $d > 1 = g$.

We first show that $G(d, n, q, r)$ is $(d-1)$-connected. Suppose $|E| \leq d - 1$, then by the consecutivity lemma we have $|C| + d - 1 \leq |S| \leq |C| + |E|$; hence, $|E| = d - 1$ and S is exactly the union of C and E. So, $G(d, n, q, r)$ is at least $(d-1)$-connected. Moreover, both S and D are consecutive runs. It also follows that each claw from D or E contains at least one node in D and each claw from C contains at least a node in C. For the sake of description, let us first introduce some notation.

For any node i, let $l(i)$ (respectively $r(i)$) be the left (right) end point of $ce(i)$ looking clockwise on the $\mathbb{Z}_n$-circle. Let $i, j \in \mathbb{Z}_n$; then we use the phrase *nodes between i and*

j to mean all nodes from i to j or from j to i clockwise around the circle, depending on which one has fewer nodes. Let m be the multiplicative inverse of $q \pmod n$ and m^* be the magnitude of m. We will prove the theorem by showing the following two claims.

Claim 1. *If $n \geq 3d$, $|E| \leq d-1$ and $|D| \leq |C|$, then $(q^*-1)(|D|-1) < d$ where q^* is magnitude of q. Furthermore, D has a loop-node unless $q \equiv -d \pmod n$.*

Claim 2. *If $n > 3d$, $|E| \leq d-1$ and $|C| \leq |D|$, then $(m^*-1)(|C|-1) < d$. Furthermore, C has a loop-node unless $qd \equiv -1 \pmod n$.*

Before proving these facts, let us show how the claims enable us to prove the second part of our theorem.

For the forward direction, if $G(d,n,q,r)$ is d-connected, then clearly it has no loop. Furthermore, if $r \equiv 1 \pmod{(d+1)}$ and $q \equiv -d \pmod n$, then we can assume $r = x(d+1)+1$ for some $x \in \mathbb{Z}_n$. By definition, x is connected to $\{x+1,\ldots,x+d\}$ and $x+1$ is connected to $\{x-d+1,\ldots,x\}$. Of the d claws containing x, there is exactly one claw not containing $x+1$ which is the claw from $x+1$. Similarly, of the d claws containing $x+1$, there is exactly one claw not containing x which is the claw from x. The remaining $d-1$ claws contain both x and $x+1$. No other claw intersects x or $x+1$. Hence, removing all $d-1$ nodes whose claws intersect both x and $x+1$ disconnects x and $x+1$ from the rest of the nodes, contradicting $G(d,n,q,r)$ being d-connected. We are left to show that condition (3) does not hold. Again, let $r = x(d+1)+1$ for some $x \in \mathbb{Z}_n$. Notice that if $qd \equiv -1 \pmod n$, then xd connects to $\{xd+1,\ldots,xd+d\}$ and $xd+d$ connects to $\{xd,\ldots,xd+d-1\}$. Thus, removing $d-1$ nodes $\{xd+1,\ldots,xd+d-1\}$ disconnect the rest of the nodes from xd and $xd+d$. So if $G(d,n,q,r)$ is d-connected then (3) cannot happen either.

For the backward direction, if none of the three conditions holds and $G(d,n,q,r)$ is still not d-connected, then there exists a node cut E with size less than d. When $|C| \le |D|$, by Claim 1 it must be the case that $q \equiv -d \pmod{n}$. However, by part (b) of Lemma 6.1 our graph has no loop only if $0 < (r)_\psi \le \psi - d$. But $\psi = gcd(q-1,n) = gcd(d+1,n)$ which is less than d unless $d+1 = \psi$. It follows that $(r)_\psi = 1$ or $r \equiv 1 \pmod{(d+1)}$, a contradiction. When $|D| \le |C|$, by Claim 2 it must be the case that $qd \equiv -1 \pmod{n}$, thus $(q-1)d \equiv -d-1 \pmod{n}$. So $\psi \mid d+1$. Similar to the previous case, we conclude that $d+1=\psi$, which implies $r \equiv 1 \pmod{(d+1)}$, another contradiction. □

Proof of Claim 1. Without loss of generality, we assume $q > 0$. The case when $q < 0$ is symmetric. Notice that $q^* < n/2$ since $g = 1$. Moreover, as we have discussed, $|E| \le d-1$ implies that both $S = C \cup E$ and D are consecutive runs and $|E| = d-1$.

We first show that for any two nodes i and $i+1$ of D, S cannot fit between $l(i)$ and $r(i+1)$. (When $q < 0$, we will show that S cannot fit between $l(i+1)$ and $r(i)$.) The number of nodes between $l(i)$ and $r(i+1)$ is $q^* + d$. If S fits between them then $|S| = |C \cup E| \le q^* + d - 2$ because any claw-end from D must also intersect D. If follows that $q^* > |C|$. Moreover, $|D| \le |C|$ and $n = |D|+|C|+d-1$ implies that $|C| \ge \frac{n-d+1}{2} \ge |D|$, and $n \ge 3d$ gives us

$$q^* > |C| \ge \frac{n-d+1}{2} \ge (n - \frac{n}{3} + 1)/2 > n/3.$$

Let l and r be the left and right end point of S, respectively. Consider the nodes between $l(i+2)$ and $r(i+3)$ if $i+2$ and $i+3$ are both in D. It follows from $l(i) \notin S$, $r(i+1) \notin S$, and $n/3 < q^* < n/2$ that $l(i+2)$ lies between $r(i+1)$ and $l(i)$, thus $l(i+2) \notin S$. Now, if $l(i+3) \notin S$ then it must be the case that $|D|$ is at least as large as the number of points from $r(i+1)$ to $l(i+3)$, which is $2q^* - (d-2)$. Thus, $2q^* - (d-2) \le |D| \le \frac{n-d+1}{2}$, which leads to $q^* \leq \frac{n}{3} - \frac{3}{4}$, contradicting $q^* > \frac{n}{3}$. Consequently, $l(i+3) \in S$. Again, as each claw-end from D intersects D, $r(i+3) \notin S$; hence, the elements of S also lie between $l(i+2)$ and $r(i+3)$. The same conclusion holds if we consider $i-1$ and $i-2$.

Continuing this way, it is obvious that there are at least $k = \lfloor \frac{|D|-1}{2} \rfloor$ different adjacent pairs $(i, i+1)$ of nodes in D such that the elements of S lie between $l(i)$ and $r(i+1)$. Let these pairs be $(i_1, i_1+1), \ldots, (i_k, i_k+1)$. Without loss of generality, assume $l(i_k)$ is closest to l, which means that $r(i_k+1)$ is furthest from r. Since all $r(i_j)$'s are different and not in S, considering the the points between $l(i_k)$ and $r(i_k+1)$ we obtain $q^* + d - 1 - k \geq |S| = n - |D|$. So,

$$q^* \geq n - |D| - d + 1 + k \geq n - |D| - d + 1 + \frac{|D|}{2} - 1 \geq \frac{n}{2} - \frac{1}{4}$$

contradicting $q^* < n/2$.

Since S cannot fit between $l(i)$ and $r(i+1)$ for any $i, i+1 \in D$, all nodes between $r(i)$ and $l(i+1)$ (if $q^* \geq d$) or between $l(i+1)$ and $r(i)$ (when $q^* < d$) must be in D. Counting this way and taking into account the fact that both D and S are consecutive runs, it is easy to see that D must have at least $2 + q^* - d + (|D| - 2)q^*$ nodes. Thus $|D| > 1 + q^* - d + (|D| - 2)q^*$, or $(q^* - 1)(|D| - 1) < d$.

If $|D| = 1$, then the node in D is clearly a loop-node. If $|D| \geq 2$, then $q^* - 1 < d$. When $q^* < d$, every pair of claws from adjacent nodes overlap. By Lemma 6.4, D has a loop-node. When $q^* = d$, D has either a loop-node or a pair of nodes i and $i+1$ such that claws from i and $i+1$ end with $f(i) = \{i+1, \cdots, i+d\}$ and $f(i+1) = \{i-d+1, \cdots, i\}$, respectively. The latter one implies that $r + qi \equiv i + 1 \pmod{n}$ and $r + q(i+1) \equiv i - d + 1 \pmod{n}$. Thus, $q \equiv -d \pmod{n}$, the exceptional case. ☐

Proof of Claim 2. Since $g = 1$, the claw-ends coming out from C can be ordered so that the second node of each is the first node of the next claw-end. Then, C must consist of nodes with indices $a, a + m^*, \ldots, a + (|C| - 1)m^*$, and these must all lie among the $d - 1 + |C|$ consecutive nodes in $S = C \cup E$. If $m^* = 1$, then the claim holds trivially. If $m^* > 1$, then either all the nodes lying between these nodes of C are in E, so that $(m^* - 1)(c - 1) < d$, or the size of D, $|D|$, is at most $m^* - 1$ so that D can fit between adjacent nodes of C in this order. The lemma is proven if we show that this latter case cannot happen when $n > 3d$.

We use the word "interval" to mean the nodes lying between adjacent nodes of C exclusively in the order above. Clearly $m^* < \frac{n}{2}$, thus *either* every other interval of C contains D, so that $m^* \geq |D| + \lfloor \frac{|C|}{2} \rfloor$, *or* there are 3 consecutive intervals such that the first one or the third one contains D and the other two do not, so that $n - 2m^* \geq |D|$. If the former occurs, then $n/2 \geq |D| + |C|/2 \geq (3/2)|C|$, so that $n \leq |D| + |C| + d - 1 < n/2 + n/6 + n/3 - 1 < n$, a contradiction. If the latter case occurs, then $n \geq 2m^* + |D| \geq 3|D| \geq 3|C|$, so $n \leq |D| + |C| + d - 1 \leq n/3 + n/3 + n/3 - 1 < n$, again a contradiction.

If $|C| = 1$, then the node in C is obviously a loop-node. If $|C| \geq 2$, then $m^* - 1 < d$. When $m^* < d$, by Lemma 6.4 C contains a loop-node. When $m^* = d$, also by Lemma 6.4 C has either a loop or a pair of nodes i and $i + m^*$ such that the claws from i and $i + m^*$ end with $\{i + 1, \cdots, i + d\}$ and $\{i, \cdots, i + d - 1\}$, respectively. The latter one implies $r + qi \equiv i + 1 \pmod{n}$ and $r + q(i + m^*) \equiv i \pmod{n}$. Thus, $qd = qm^* \equiv -1 \pmod{n}$, the exceptional case. □

The following corollary removes the condition $n \geq d^4$ from the result of Homobono and Peyrat [82].

Corollary 6.7. *$G_I(d, n)$ is d-connected iff $gcd(n, d) > 1$ and $(d + 1) \mid n$.*

Proof. Note that $G_I(d, n)$ has no loop iff $(d + 1) \mid n$ by Lemma 6.1 part (b). When $g = gcd(d, n) > 1$ and $(d + 1) \mid n$ we must have $g < d$, thus by Theorem 6.5, $G_I(d, n)$ is d-connected. Conversely, if $d + 1$ does not divide n, then $G_I(d, n)$ is not d-connected because it has a loop; and if $g = 1$ and $(d+1) \mid n$, then by Theorem 6.6, $G_I(d,n)$ is not d-connected. Notice that we proved this direction independent of $n > 3d$. Therefore, $G_I(d,n)$ is d-connected iff $gcd(n,d) > 1$ and $(d+1) \mid n$. □

The following corollary uses a weaker condition, namely $n > d \cdot gcd(n,d)$, instead of the condition $n \geq d^3$ in the result of Imase, Soneoka and Okada [94].

Corollary 6.8. *If $n > d \cdot gcd(n,d)$, then $G_B(d,n)$ and $G_I(d,n)$ are at least ($d-1$)-connected.*

Proof. When $gcd(n,d) = d$, $G(d,n,q,r)$ is the line-graph of $G(d,n/d,q,r)$. To see this, consider a digraph $\bar{G}$ with nodes $\bar{0},\ldots,\overline{n'-1}$ where $n' = n/d$, and with edges labeled by $0,\ldots,n$; each node $\bar{i}$ has in-edges $i, i+n', \ldots, i+(d-1)n'$ and out-edges $qi+r, qi+r+1, \ldots, qi+r+d-1$. Clearly, there is an edge from $\bar{i}$ to $\bar{j}$ iff $j \equiv qi+r+k \pmod{n'}$ for some $k = 0,\ldots,d-1$. Thus, $\bar{G}$ is isomorphic to $G(d,n',q,r)$. On the other hand, the line graph of $\bar{G}$ is $G(d,n,q,r)$.

It was proved in [50] that if $g \mid d$, then $G(d,n,q,r)$ is at least $(d-1)$-line-connected and it is d-line-connected iff it has no loop. This implies that if $g = d$ and $n > d^2$, then $G(d,n,q,r)$ is at least ($d-1$)-connected and it is d-connected iff it has no loop. This fact and Theorem 6.6 allow us to assume that $1 < gcd(n,d) < d$.

Consider the proof of Theorem 6.5. We show $|E| \geq d-1$. In Case 1, if $|C| = 1$, $|E| \geq d-1$; if $|C| \geq 2$, then we must have $1 + n/g \leq d$ since only one C-orbit exists. Thus, $n \leq g(d-1)$, a contradiction. In Case 2, by the reduction, we may assume that D has only one block. Since $n > gd \geq (g-1)d$, exactly one claw from D intersects D. However, there are d claws intersecting D. Realize that $d-1$ of them must come from E, i.e. $|E| \geq d-1$. In Case 3, if there is only one C-orbit, then it is similar to that in Case 1. If two C-orbits exist, then each C-orbit contains at least $(g-1)/2$ elements of C. Clearly

$E \cup C$ must have at least n/g elements in this case. Any consecutive run of size n/g in $E \cap C$ contains exactly 2 elements of C and the rest are in E. So, if $|E| \leq d - 2$, then $n/g \leq 2 + |E| = d$, a contradiction. □

6.5 Modification of $G_B(d, n)$

A purpose of this study is to find good candidates for the topological structure of communication networks. Here is a basic problem: Given the number of nodes and an upper bound on degree, find a digraph to achieve the smallest diameter and largest connectivity. Suppose that G is a digraph with n nodes and each node of G has in-degree and out-degree at most d. By a simple calculation, it was shown that the diameter of G is at least $\lceil \log_d n(d-1) + 1 \rceil - 1$ [41]. In general, for given n and d, determining whether a digraph exists to achieve this lower bound of diameter is not an easy job. However, if we allow a difference of one from the optimal value, then the generalized de Bruijn digraphs $G_B(d, n)$ and the generalized Kautz digraphs $G_I(d, n)$ meet the requirement (see [92, 93, 139]). A question is, could these graphs be modified to have largest connectivity? In fact, loops do nothing to contribute to the connectivity. One can "improve" them by deleting whatever loops occur according to the defining formulae, replacing them by a single cycle or several disjoint cycles. This improvement has been studied for $d = 2$ in [53, 140].

From Theorem 6.6, we see that $G_I(d, n)$ can be at most $(d-1)$-connected without a loop. So, the improvement does not always exist for $G_I(d, n)$. However, it almost always exists for $G_B(d, n)$. We give this result in this section. We first prove a lemma.

Lemma 6.9. *Let $n > \max(5d, d^2 + 1)$ and $d \geq 2$. Suppose that E is a node-cut of size at most $d - 1$ in $G_B(d, n)$, such that removal of the nodes in E leaves no path from any node in C to any node in D; then* either

(1) *C has a loop node, and the number of nodes between any loop node in E and any loop node in C is at most $2d - 1$,* or

(2) *D has a loop node, and the number of nodes between any loop node in E and any loop node in D is at most $2d - 1$.*

Proof. First, assume $gcd(n, d) = 1$. Since $1 < d < d^2 + 1 < n$, we have $q^* \neq 1$, $m^* \neq 1$, $d \not\equiv -d \pmod{n}$ and $d^2 \not\equiv -1 \pmod{n}$. Moreover, by the consecutivity lemma, the set S has $|C| + d - 1$ elements and both $S = C \cup E$ and D are consecutive runs. In this case, $|E|$ is indeed minimum, so every claw-end from E must intersect D. Consider the proof of Theorem **??**. If $|D| \leq |C|$, then by Claim 1 D has a loop node and $|D| \leq d$. A claw-end from an E-loop node has to intersect D, so that the number of nodes between an E-loop node and any node in D is at most $2d - 1$. If $|C| \leq |D|$, then by Claim 2 C has a loop node and $|C| \leq d$. Hence $|S| \leq 2d - 1$. Consequently, the number of nodes between any two nodes in $E \cup C$ is at most $2d - 1$.

Now, assume $1 < gcd(n, d) < d$. Notice that in the proof of Theorem 6.5, the minimality of $|E|$ is assumed. Here, we do not assume it. However, by Corollary 6.8, $|E| = d - 1$ is indeed minimum. The difference is that D may not be consecutive. To meet the assumption $S = C \cup E$ in the proof of Theorem 6.5, we have to move at most $g - 1$ elements from C into D. Those elements are in $A = (C \cup E) \setminus S$ and cannot have a loop. So, the movement affects only the sizes of C and D. Let $C' = C \setminus A$ and $D' = D \cup A$. Now, consider the proof of Theorem 6.5 applied to E, C' and D'. In case 1, every node in C' has a loop, so that C has a loop. Moreover, $|S| = d$ so $|C'| = 1$ and $|C| \leq 1 + g - 1 < d$. Clearly the loop node of C' is within d of every node in E. The claw-end from a loop node i in A has to intersect S, so i is also within $2d - 1$ of every node in E. In sum, the number of nodes between a C-loop node and an E-loop node is at most $2d - 1$. In case 2, $|D'| \leq d - 1$ and

D' has a loop, so that $|D| \le d-1$ and D has a loop. The cardinality of E is minimum so every claw-end from E intersects D'. It follows that every E-loop node is within $2d-1$ from every loop node in D ($\subseteq D'$). In case 3, of g elements in an orbit, $\lceil \frac{g}{2} \rceil - 1$ must not be in D' and clearly are elements of E. Moreover, just as we have noted in the proof of Theorem 6.5, there are at least $\lceil \frac{|D'|+d-1}{g} \rceil$ orbits whose claw-ends intersect D'; therefore, $|E| \ge \lceil \frac{|D'|+d-1}{g} \rceil$. This gives us

$$\lfloor \frac{g}{2} \rfloor \frac{n}{g} \le |D'| \le \frac{g|E|}{\lceil \frac{g}{2} - 1 \rceil} - (d-1).$$

Since g in this case must be odd, $g = gcd(d,n) > 1$, and $|E| \le d-1$, it is easy to see that this contradicts $n > \max(5d, d^2+1)$.

Finally, we consider the case of $gcd(n,d) = d$. Note that $G_B(d,n)$ is the line-graph of $G_B(d,n/d)$. Thus, E gives a line-cut of size at most $d-1$ for $G_B(d,n/d)$. However, it was proven in [54] that such a line-cut must be incident to a node of $G_B(d,n/d)$, which implies that C or D is a singleton. So, the lemma holds. □

A digraph is called a *modified* $G(d,n,q,r)$ if it is constructed from $G(d,n,q,r)$ by connecting all loop-nodes into disjoint cycles with sizes at least two and deleting all loops. The modification is said to be *cyclic* if all loop-nodes are connected into a single cycle. The modification is said to be *simple* if there is no multiple edge in the resultant simple graph.

Theorem 6.10. *When* $n > 2d(d-1)$ *and* $d \ge 4$*, there exists a cyclically-modified* $G_B(d,n)$ *of connectivity* d*.*

Proof. Consider two loop-nodes x and y where the number of nodes between them is at least $2d$. When (1) in Lemma 6.9 occurs, $x \in C$ implies $y \in D$. When (2) in Lemma 6.9 occurs, $x \in D$ implies $y \in C$. This means that as long as all loop-nodes are connected by a cycle (or disjoint cycles) with edges of “distance” at least $2d-1$, the node-cut E of size less than d will no longer exist in the modified graph. Hence, the connectivity becomes d. We next show the existence of such a modification. Consider a graph H with node set consisting of all loop-nodes of $G_B(d,n)$ and an edge between x and y exists iff x and y are at a distance at least $2d-1$ from each other. If H is Hamiltonian, then the theorem is proved. We prove the Hamiltonian property of H by showing that minimum degree $\delta(H)$ of H is at least half the number of its nodes. Consider any loop-node i of $G_B(d,n)$. As we have mentioned in section 6.2, except for the possible loop node right next to i, all other loop nodes are at least $n/(d-1)-1 \geq 2d-1$ from it, i.e the number of nodes between them is at least $2d$. Hence, $\delta(H) \geq d-2+\psi$. It is easy to see that when $d \geq 4$, $d-2+\psi \geq \frac{d-1+\psi}{2}$. □

Notice that as $n > \max(5d, d^2+1)$, our cyclic modification is also simple. The next theorem relaxes the conditions on n and d a bit further.

Theorem 6.11. *Let $\psi = gcd(d-1,n)$. If $1 < \psi < d-1$, then for $n \geq d^2$ and $d \geq 2$, there exists a simply-modified $G_B(d,n)$ of connectivity d.*

Proof. For each k-value such that $\psi \mid k$, there are exactly ψ loop-nodes which are evenly distributed with distance n/ψ. Note that $n/\lambda \geq 2n/d \geq 2d$. We connect each loop-node x to another loop-node $x+n/\psi$. Then, all loop-nodes are connected by several disjoint cycles of size ψ, and all edges are in the graph H of the proof of Theorem 6.10. Finally, we notice that the above connections produce no multiple edges. The details are easy to verify. □

6.6 Discussions

In this chapter, we have determined the connectivity of consecutive-d digraphs $G(d, n, q, r)$ in almost all cases, and studied how to modify these graphs to maximize connectivity. Our results generalized and improved existing results on de Bruijn digraphs, Kautz digraphs, and their generalizations.

There are still, however, a few small gaps in our characterization of the connectivity of $G(d, n, q, r)$. In particular, several problems remained to be solved:

(a) When $\gcd(q, n) = d$ and $n \leq d^2$, what are the necessary and sufficient conditions for $G(d, n, q, r)$ to be d-connected.

(b) When $\gcd(q, n) = 1$ and $n \leq 3d$, what are the necessary and sufficient conditions for $G(d, n, q, r)$ to be d-connected.

(c) When $1 < \gcd(q, n) < d$, what are the necessary and sufficient conditions for $G(d, n, q, r)$ to be $(d - i)$-connected, where $0 < i < \gcd(q, n)$.

Chapter 7

Reliability of Cyclic Systems

7.1 Overview

A cyclic consecutive-k-out-of-n: G system $con_C(k, n : G)$ is a cycle of $n(\geq k)$ components such that the system works if and only if some k consecutive components all work. Suppose n components with working probabilities $p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[n]}$ are all exchangeable. How can they be assigned to the n positions on the cycle to maximize the reliability of the system? Kuo, Zhang, and Zuo [104] showed that if $k = 2$, then the optimal assignment is *invariant*, i.e. it depends solely on the ordering of working probabilities of the components, independent of their values. They also claimed that for $k \geq 3$ and $n > 2k + 1$, $Con_C(k, n : G)$ has no invariant optimal assignment. For $n \leq 2k + 1$, Zuo and Kuo [172] claimed that there exists an invariant optimal assignment

$$(p_{[1]}, p_{[3]}, p_{[5]}, \cdots, p_{[6]}, p_{[4]}, p_{[2]}).$$

However, Jalali, Hawkes, Cui, and Hwang [95] found that their proof is incomplete. A proof in case $n = 2k + 1$ was given in [61]. In this chapter, we give a complete proof for this invariant optimal assignment with $n \leq 2k + 1$.

7.2 Main Results

In this section, we show the following.

Theorem 7.1. *For $k \leq n \leq 2k+1$, there exists an invariant optimal assignment*

$$(p_{[1]}, p_{[3]}, p_{[5]}, \cdots, p_{[6]}, p_{[4]}, p_{[2]}).$$

Let $p_1, p_2, \cdots, p_n$ be reliabilities of the n components on the cycle in counterclockwise direction. To simplify the proof, we first assume that

$$0 < p_{[1]} < p_{[2]} < \cdots < p_{[n]} < 1.$$

Our proof is based on the following representation of the reliability of consecutive-k-out-of-n: G cycle for $n \leq 2k+1$:

Lemma 7.2. *The reliability of consecutive-k-out-of-n: G cycle for $n \leq 2k+1$ under assignment C can be represented as*

$$\begin{aligned} R(C) &= p_1 \cdots p_n + \sum_{i=1}^{n} q_i p_{i+1} \cdots p_{i+k} \\ &= p_1 \cdots p_n + \sum_{i=1}^{n} p_i \cdots p_{i+k-1} - \sum_{i=1}^{n} p_i \cdots p_{i+k} \end{aligned} \tag{7.1}$$

where $q_i = 1 - p_i$ and $p_{n+i} = p_i$.

Proof. The system works if and only if all components work or for some i, the ith component fails and the next k components all work. Since $n \leq 2k+1$, there exists at most one such i. Therefore,

$$R(C) = p_1 \cdots p_n + \sum_{i=1}^{n} q_i p_{i+1} \cdots p_{i+k}.$$

□

This representation is key to show the main theorem. It explains why an invariant optimal assignment exists for $n \leq 2k+1$, but does not exist for $n > 2k+1$.

For $n = k, k+1$, by Lemma 7.2, $R(C)$ has the same value for all assignment C and hence Theorem 7.1 is trivially true. Next, we assume $k+2 \le n \le 2k+1$.

To prove Theorem 7.1, it suffices to show that in any optimal assignment,

$$(p_i - p_j)(p_{i-1} - p_{j+1}) > 0 \text{ for } 1 < i < j < n. \tag{7.2}$$

That is, selecting any component to be labeled p_1, we always have

$$(p_i - p_{n-i+1})(p_{i+1} - p_{n-i}) > 0 \text{ for } i = 1, \cdots, h. \tag{7.3}$$

where $h = \lfloor n/2 \rfloor$. For simplicity of representation, we denote $i' = n - i + 1$. When n is odd, $(\frac{n+1}{2})' = \frac{n+1}{2}$. Furthermore, without loss of generality, we assume $p_1 > p_{1'}$ throughout this proof. Then, the condition (7.3) can be rewritten as

$$p_i > p_{i'} \text{ for } i = 1, \cdots, h. \tag{7.4}$$

Let $I = \{i \mid 1 < i \le h, p_i < p_{i'}\}$. Let C_I be the assignment obtained from C by exchanging components i and i' for all $i \in I$. To prove (7.4), it suffices to show that for any assignment C, if $I \neq \emptyset$, then

$$R(C) < R(C_I).$$

Denote $I' = \{i' \mid i \in I\}$ and

$$(y_{i_1} \cdots y_{i_d})^I = \left(\prod_{\substack{1 \le j \le d \\ i_j \notin I \cup I'}} y_{i_j} - \prod_{\substack{1 \le j \le d \\ i_j \notin I \cup I'}} y_{i'_j} \right) \left(\prod_{\substack{1 \le j \le d \\ i_j \in I \cup I'}} y_{i'_j} - \prod_{\substack{1 \le j \le d \\ i_j \in I \cup I'}} y_{i_j} \right),$$

where $y_i = p_i$ or q_i. It is easy to verify that

$$\left(y_{i_1} \cdots y_{i_d} \prod_{\substack{1 \le j \le d \\ i_j \in I \cup I'}} \frac{y_{i'_j}}{y_{i_j}} + y_{i'_1} \cdots y_{i'_d} \prod_{\substack{1 \le j \le d \\ i_j \in I \cup I'}} \frac{y_{i_j}}{y_{i'_j}} \right) - (y_{i_1} \cdots y_{i_d} + y_{i'_1} \cdots y_{i'_d})$$
$$= (y_{i_1} \cdots y_{i_d})^I. \tag{7.5}$$

Intuitively, consider the sum $s = (y_{i_1} \cdots y_{i_d} + y_{i'_1} \cdots y_{i'_d})$. Let s' be the sum obtained from s by exchanging y_{i_j} with $y_{i'_j}$ whenever $i_j \in I \cup I'$, then $s' - s = (y_{i_1} \cdots y_{i_d})^I$.

Denote $Q_k(C) = \sum_{i=1}^{n} p_i \cdots p_{i+k-1}$. Then

$$R(C) = p_1 \cdots p_n + Q_k(C) - Q_{k+1}(C), \tag{7.6}$$

and

$$R(C_I) - R(C) = \big(Q_k(C_I) - Q_k(C_I)\big) - \big(Q_{k+1}(C_I) - Q_{k+1}(C)\big). \tag{7.7}$$

Let $a = \lfloor k/2 \rfloor$ and

$$s = \begin{cases} h & \text{if } n \text{ is even and } k \text{ is odd} \\ h+1 & \text{otherwise.} \end{cases} \tag{7.8}$$

Then, we have

Lemma 7.3. $Q_k(C_I) - Q_k(C) = \sum_{i=1}^{s} (p_{-a+i} \cdots p_{-a+i+k-1})^I$.

Proof. Consider four cases.

Case 1. n and k both are even. In this case, $s = h + 1 = 1 + n/2$ and $a = k/2$. Note that

$$\begin{aligned}
\sum_{i=h+2}^{n} p_{-a+i} \cdots p_{-a+i+k-1} &= \sum_{i=2}^{h} p_{-a+(2h+2-i)} \cdots p_{-a+(2h+2-i)+k-1} \\
&= \sum_{i=2}^{h} p_{n-(-a+i+k-1)+1} \cdots p_{n-(-a+i)+1} \\
&= \sum_{i=2}^{h} p_{(-a+i+k-1)'} \cdots p_{(-a+i)'}.
\end{aligned}$$

Thus,

$$\begin{aligned} Q_k(C) &= p_{-a+1}\dots p_a + \sum_{i=2}^{h}(p_{-a+i}\dots p_{-a+i+k-1} + p_{(-a+i)'}\dots p_{(-a+i+k-1)'}) \\ &\quad + p_{-a+h+1}\cdots p_{-a+h+k} \\ &= \prod_{j=1}^{a}(p_j p_{j'}) + \sum_{i=2}^{h}(p_{-a+i}\cdots p_{-a+i+k-1} + p_{(-a+i)'}\cdots p_{(-a+i+k-1)'}) \\ &\quad + \prod_{j=1}^{a}(p_{h-a+j}p_{(h-a+j)'}). \end{aligned}$$

So,

$$Q_k(C_I) - Q_k(C) = \sum_{i=2}^{h}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

However,

$$(p_{-a+1}\cdots p_{-a+k})^I = (p_{-a+h+1}\cdots p_{-a+h+k})^I = 0.$$

Therefore,

$$Q_k(C_I) - Q_k(C) = \sum_{i=1}^{h+1}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

Case 2. n is even and k is odd. In this case, $s = h = n/2$ and $a = (k-1)/2$. Note that

$$Q_k(C) = \sum_{i=1}^{h}(p_{-a+i}\cdots p_{-a+i+k-1} + p_{(-a+i)'}\cdots p_{(-a+i+k-1)'}).$$

Therefore,

$$Q_k(C_I) - Q_k(C) = \sum_{i=1}^{h}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

Case 3. n and k both are odd. In this case, $s = h + 1 = (n+1)/2$ and $a = (k-1)/2$.

Note that

$$\begin{aligned}
Q_k(C) &= \sum_{i=1}^{h}(p_{-a+i}\cdots p_{-a+i+k-1}+p_{(-a+i)'}\cdots p_{(-a+i+k-1)'}) \\
&\quad +p_{-a+h+1}\cdots p_{-a+h+k} \\
&= \sum_{i=1}^{h}(p_{-a+i}\cdots p_{-a+i+k-1}+p_{(-a+i)'}\cdots p_{(-a+i+k-1)'}) \\
&\quad +p_{-a+h-1}p_{(-a+h-1)'}\cdots p_h p_{h'} p_{h+1}.
\end{aligned}$$

Thus,

$$Q_k(C_I)-Q_k(C)=\sum_{i=1}^{h}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

However,

$$(p_{-a+h+1}\cdots p_{-a+h+k})^I=0.$$

Therefore,

$$Q_k(C_I)-Q_k(C)=\sum_{i=1}^{h+1}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

Case 4. n is odd and k is even. In this case, $s=h+1=(n+1)/2$ and $a=k/2$. Note that

$$\begin{aligned}
Q_k(C) &= p_{-a+1}\cdots p_{-a+k}+\sum_{i=2}^{h+1}(p_{-a+i}\cdots p_{-a+i+k-1}+p_{(-a+i)'}\cdots p_{(-a+i+k-1)'}) \\
&= \prod_{j=1}^{a}(p_j p_{j'})+\sum_{i=2}^{h+1}(p_{-a+i}\cdots p_{-a+i+k-1}+p_{(-a+i)'}\cdots p_{(-a+i+k-1)'})
\end{aligned}$$

Thus

$$Q_k(C_I)-Q_k(C)=\sum_{i=2}^{h+1}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

However,

$$(p_{-a+1}\cdots p_{-a+k})^I = 0.$$

Therefore,

$$Q_k(C_I) - Q_k(C) = \sum_{i=1}^{h+1}(p_{-a+i}\cdots p_{-a+i+k-1})^I.$$

$\square$

Define

$$t = \begin{cases} h & \text{if } n \text{ is even and } k+1 \text{ is odd} \\ h+1 & \text{otherwise} \end{cases}$$

and $b = \lfloor (k+1)/2 \rfloor$. We have a useful representation of $R(C_I) - R(C)$ as follows.

Lemma 7.4.

$$R(C_I) - R(C) = \sum_{i=2}^{t}(p_{-b+i}\cdots p_{-b+i+k-1})^I - \sum_{i=1}^{t}(p_{-b+i}\cdots p_{-b+i+k})^I.$$

Proof. By Lemma 7.3, we have

$$R(C_I) - R(C) = \sum_{i=1}^{s}(p_{-a+i}\cdots p_{-a+i+k-1})^I - \sum_{i=1}^{t}(p_{-b+i}\cdots p_{-b+i+k})^I.$$

Note that if k is even and n is odd, then $a = b$, $s = t$, and

$$(p_{-a+1}\cdots p_{-a+k})^I = (p_{-k/2+1}\cdots p_{k/2})^I = 0;$$

if k is odd and n is odd, then $a = b - 1$, $s = t$, and

$$(p_{-a+s}\cdots p_{-a+s+k-1})^I = (p_{-(n-k)/2}\cdots p_{-(n+k)/2-1})^I = 0;$$

if k is even and n is even, then $a = b$, $s = t + 1$, and

$$(p_{-a+1} \cdots p_{-a+k})^I = (p_{-a+s} \cdots p_{-a+s+k-1})^I = 0;$$

if k is odd and n is even, then $a = b - 1$ and $s = t - 1$. Therefore, we always have

$$\sum_{i=1}^{s} (p_{-a+i} \cdots p_{-a+i+k-1})^I = \sum_{i=2}^{t} (p_{-a+i} \cdots p_{-a+i+k-1})^I.$$

□

Note that $(p_{-b+i} \cdots p_{-b+i+k-1})^I \geq 0$ for $2 \leq i \leq t$ and $(p_{-b+i} \cdots p_{-b+i+k})^I \geq 0$ for $1 \leq i \leq t$. Therefore, to prove $R(C_I) > R(C)$, we need to compare $(p_{-b+i} \cdots p_{-b+i+k-1})^I$ with $(p_{-b+i} \cdots p_{-b+i+k})^I$. Before looking in details at the proofs of Lemmas 7.5,7.6, and 7.7, the reader should take a look at equation (7.10) to see what we are shooting for.

Lemma 7.5. *Suppose $I = \{i \mid 1 < i \leq h, p_i < p_{i'}\}$. Then for $i = 1, \cdots, b$,*

$$(q_{-b+i} p_{-b+i+1} \cdots p_{-b+i+k})^I = (p_{-b+i+1} \cdots p_{-b+i+k})^I - (p_{-b+i} p_{-b+i+1} \cdots p_{-b+i+k})^I \geq 0, \tag{7.9}$$

and the strict inequality sign holds if and only if

$$\begin{aligned} \{j \mid b - i + 1 \leq j \leq \min(-b + i + k, n + b - i - k), j \in I\} &\neq \emptyset \\ \{j \mid b - i + 1 \leq j \leq \min(-b + i + k, n + b - i - k), j \notin I\} &\neq \emptyset. \end{aligned}$$

Proof. First, assume $-b+i \notin I \cup I'$. Then we have

$$\begin{aligned}
&(q_{-b+i}p_{-b+i+1}\cdots p_{-b+i+k})^I \\
= &\left(q_{-b+i}\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\notin I\cup I'}} p_j - q_{(-b+i)'}\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\notin I\cup I'}} p_{j'}\right) \\
&\cdot\left(\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\in I\cup I'}} p_{j'} - \prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\in I\cup I'}} p_j\right) \\
= &\left(\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\notin I\cup I'}} p_j - \prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\notin I\cup I'}} p_{j'}\right) \\
&\cdot\left(\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\in I\cup I'}} p_{j'} - \prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\in I\cup I'}} p_j\right) \\
&-\left(\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\notin I\cup I'}} p_j - \prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\notin I\cup I'}} p_{j'}\right) \\
&\cdot\left(\prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\in I\cup I'}} p_{j'} - \prod_{\substack{-b+i+1\le j\le -b+i+k\\ j\in I\cup I'}} p_j\right) \\
= &(p_{-b+i+1}\cdots p_{-b+i+k})^I - (p_{-b+i}\cdots p_{-b+i+k})^I
\end{aligned}$$

The same equation holds when $-b+i \in I \cup I'$ similarly.

Now, suppose $-b+i \in I \cup I'$. If $-b+i+k \le n+b-i-k$, then $-b+i+k \le h$ and hence we have

$$\begin{aligned}
&(q_{-b+i}p_{-b+i+1}\cdots p_{-b+i+k})^I \\
= &\left(\prod_{j=1}^{b-i} p_j p_{j'}\right)\left(\prod_{\substack{b-i+2\le j\le -b+i+k\\ j\notin I}} p_j - \prod_{\substack{b-i+2\le j\le -b+i+k\\ j\notin I}} p_{j'}\right) \\
&\cdot\left(p_{(b-i+1)'}q_{(-b+i)'}\prod_{\substack{b-i+2\le j\le -b+i+k\\ j\in I}} p_{j'} - p_{(b-i+1)}q_{(-b+i)}\prod_{\substack{b-i+2\le j\le -b+i+k\\ j\in I}} p_j\right) \\
\ge &\ 0
\end{aligned}$$

since $-b+i \in I \cup I'$ implies that

$$\begin{aligned} p_{(b-i+1)'}q_{(-b+i)'} - p_{b-i+1}q_{-b+i} &= p_{-b+i}q_{b-i+1} - p_{b-i+1}q_{-b+i} \\ &= p_{-b+i} - p_{b-i+1} \\ &= p_{(b-i+1)'} - p_{b-i+1} \\ &> 0 \end{aligned}$$

On the other hand, if $-b+i \notin I \cup I'$, and $-b+i+k < n+b-i-k+1$, then

$$\begin{aligned} & (q_{-b+i}p_{-b+i+1}\cdots p_{-b+i+k})^I \\ = \; & \left(\prod_{j=1}^{b-i} p_j p_{j'}\right)\left(p_{b-i+1}q_{-b+i}\prod_{\substack{b-i+2\leq j\leq -b+i+k \\ j\notin I}} p_j - p_{(b-i+1)'}q_{(-b+i)'}\prod_{\substack{b-i+2\leq j\leq -b+i+k \\ j\notin I}} p_{j'}\right) \\ & \cdot\left(\prod_{\substack{b-i+2\leq j\leq -b+i+k \\ j\in I}} p_{j'} - \prod_{\substack{b-i+2\leq j\leq -b+i+k \\ j\in I}} p_j\right) \\ \geq \; & 0 \end{aligned}$$

since $-b+i \notin I \cup I'$ implies that

$$\begin{aligned} p_{b-i+1}q_{-b+i} - p_{(b-i+1)'}q_{(-b+i)'} &= p_{b-i+1} - p_{(b-i+1)'} \\ &> 0 \end{aligned}$$

Moreover, it is easy to verify that in both cases

$$(q_{-b+i}p_{-b+i+1}\cdots p_{-b+i+k})^I > 0$$

if and only if

$$\begin{aligned} \{j \mid b-i+1 \leq j \leq -b+i+k, j \in I\} &\neq \emptyset \\ \{j \mid b-i+1 \leq j \leq -b+i+k, j \notin I\} &\neq \emptyset \end{aligned}$$

Now we consider the case when $-b+i+k \geq n+b-i-k+1$, which implies $n+b-i-k \leq h$. If $-b+i \in I \cup I'$, then

$$
\begin{aligned}
& (q_{-b+i}p_{-b+i+1}\cdots p_{-b+i+k})^I \\
= \quad & \left(\prod_{j=1}^{b-i} p_j p_{j'}\right)\left(\prod_{\substack{b-i+2\leq j\leq n+b-i-k \\ j\notin I}} p_j - \prod_{\substack{b-i+2\leq j\leq n+b-i-k \\ j\notin I}} p_{j'}\right) \\
& \cdot\left(p_{(b-i+1)'}q_{(-b+i)'}\prod_{\substack{b-i+2\leq j\leq n+b-i-k \\ j\in I}} p_{j'} - p_{(b-i+1)}q_{(-b+i)}\prod_{\substack{b-i+2\leq j\leq n+b-i-k \\ j\in I}} p_j\right) \\
& \cdot\left(\prod_{n+b-i-k+1\leq j\leq h} p_j p_{j'}\right)\gamma \\
\geq \quad & 0
\end{aligned}
$$

where

$$
\gamma = \begin{cases} 1 & \text{if } n \text{ is even} \\ p_{h+1} & \text{if } n \text{ is odd} \end{cases}
$$

The case when $-b+i \notin I \cup I'$ is completely analogous. Moreover, in both cases

$$
(q_{-b+i}p_{-b+i+1}\cdots p_{-b+i+k})^I > 0
$$

if and only if

$$
\begin{aligned}
\{j \mid b-i+1 \leq j \leq n+b-i-k, j \in I\} &\neq \emptyset \\
\{j \mid b-i+1 \leq j \leq n+b-i-k, j \notin I\} &\neq \emptyset
\end{aligned}
$$

□

Similarly, we can show the following two lemmas.

Lemma 7.6. *Suppose $I = \{i \mid 1 \le i \le h, p_i < p_{i'}\}$. Then for $i = b, \cdots, t$,*

$$(p_{-b+i} \cdots p_{-b+i+k-1} q_{-b+i+k})^I = (p_{-b+i} \cdots p_{-b+i+k-1})^I - (p_{-b+i} p_{-b+i+1} \cdots p_{-b+i+k})^I \ge 0.$$

Lemma 7.7. *Suppose $I = \{i \mid 1 \le i \le h, p_i < p_{i'}\}$. Then*

$$\begin{aligned} & (q_0 p_1 \cdots p_k q_{k+1})^I \\ = \; & (p_1 \cdots p_k)^I - (p_0 \cdots p_k)^I - (p_1 \cdots p_{k+1})^I + (p_0 \cdots p_{k+1})^I \\ \ge \; & 0 \end{aligned}$$

and the strict inequality sign holds if and only if

$$\begin{aligned} \{j \mid 1 \le j \le n-k, j \in I\} & \ne \emptyset \\ \{j \mid 1 \le j \le n-k, j \notin I\} & \ne \emptyset. \end{aligned}$$

By Lemmas 7.4–7.7, we have

$$\begin{aligned} R(C_I) - R(C) = & \sum_{i=1}^{b-1} (q_{-b+i} p_{-b+i+1} \cdots p_{-b+i+k})^I + \sum_{i=b+2}^{t} (p_{-b+i} \cdots p_{-b+i+k-1} q_{-b+i+k})^I \\ & + (q_0 p_1 \cdots p_k q_{k+1})^I - (\prod_{j=0}^{k+1} p_j)^I. \end{aligned} \tag{7.10}$$

To this end, let $d = \lceil (n-k)/2 \rceil - 1$. Note that $(n-k-2-d)$ is either d or $d-1$, thus

$$\begin{aligned} (\prod_{j=0}^{k+1} p_j)^I &= (q_{-1} \prod_{j=0}^{k+1} p_j)^I + (\prod_{j=-1}^{k+1} p_j)^I \\ &= (q_{-1} \prod_{j=0}^{k+1} p_j)^I + (q_{1+k+1} \prod_{j=-1}^{1+k} p_j)^I + (\prod_{j=-1}^{k+1+1} p_j)^I \\ &= \ldots \\ &= \sum_{i=1}^{d} (q_{-i} \prod_{j=-i+1}^{k+i} p_j)^I + \sum_{i=1}^{n-k-2-d} (q_{i+k+1} \prod_{j=-i}^{i+k} p_j)^I + (\prod_{j=-d}^{n-d-1} p_j)^I, \end{aligned}$$

and

$$(\prod_{j=-d}^{n-d-1} p_j)^I = 0.$$

Hence, we have

$$\begin{aligned}
&R(C_I) - R(C) = \\
&\quad \sum_{i=1}^{b-d-1} (q_{-b+i} \prod_{j=-b+i+1}^{-b+i+k} p_j)^I + \sum_{i=1}^{d} \Big[(q_{-i} \prod_{j=-i+1}^{-i+k} p_j)^I - (q_{-i} \prod_{j=-i+1}^{i+k} p_j)^I \Big] + \\
&\sum_{i=b+n-k-d}^{t} (q_{-b+i+k} \prod_{j=-b+i}^{-b+i+k-1} p_j)^I + \sum_{i=1}^{n-k-2-d} \Big[(q_{i+k+1} \prod_{j=i+1}^{i+k} p_j)^I - (q_{i+k+1} \prod_{j=-i}^{i+k} p_j)^I \Big] + \\
&\qquad (q_0 p_1 \cdots p_k q_{k+1})^I. \qquad (7.11)
\end{aligned}$$

This representation suggests us to show Lemmas 7.8 and 7.9.

Lemma 7.8. *For $i = 1, \cdots, d$,*

$$(q_{-i} \prod_{j=-i+1}^{-i+k} p_j)^I \geq (q_{-i} \prod_{j=-i+1}^{i+k} p_j)^I$$

and the inequality holds strictly if and only if

$$(q_{-i} \prod_{j=-i+1}^{-i+k} p_j)^I > 0.$$

Proof. First, consider the case that $-i \in I'$ and n is even. Denote

$$\begin{aligned}
A &= \prod_{-i+1 \leq j \leq -i+k, j \notin I \cup I'} p_j - \prod_{-i+1 \leq j \leq -i+k, j \notin I \cup I'} p_{j'} \\
B &= \prod_{-i+1 \leq j \leq i+k, j \notin I \cup I'} p_j - \prod_{-i+1 \leq j \leq i+k, j \notin I \cup I'} p_{j'} \\
A' &= q_{(-i)'} \prod_{-i+1 \leq j \leq -i+k, j \in I \cup I'} p_{j'} - q_{-i} \prod_{-i+1 \leq j \leq -i+k, j \notin I \cup I'} p_j \\
B' &= q_{(-i)'} \prod_{-i+1 \leq j \leq i+k, j \in I \cup I'} p_{j'} - q_{-i} \prod_{-i+1 \leq j \leq -i+k, j \notin I \cup I'} p_j
\end{aligned}$$

Then

$$(q_{-i}\prod_{j=-i+1}^{-i+k}p_j)^I-(q_{-i}\prod_{j=-i+1}^{i+k}p_j)^I=AA'-BB'.$$

If $-i+k<n+i-k+1$, then $-i+k\leq h$. Hence

$$\begin{aligned}
& A-B \\
= & (\prod_{1\leq j\leq i,j\notin I}p_jp_{j'})[(\prod_{i+1\leq j\leq n-i-k,j\notin I}p_j)\alpha(1-\beta\delta) \\
& -(\prod_{i+1\leq j\leq n-i-k,j\notin I}p_{j'})\beta(1-\alpha\delta)]
\end{aligned}$$

where

$$\begin{aligned}
\alpha &= \prod_{n-i-k+1\leq j\leq -i+k,j\notin I}p_j, \\
\beta &= \prod_{n-i-k+1\leq j\leq -i+k,j\notin I}p_{j'}, \\
\delta &= \prod_{-i+k+1\leq j\leq h,j\notin I}p_jp_{j'}
\end{aligned}$$

and

$$\alpha(1-\beta\delta)-\beta(1-\alpha\delta)=\alpha-\beta\geq 0.$$

Thus, $A\geq B$.

If $-i+k\geq n+i-k+1$, then $n+i-k\leq h$. Hence

$$\begin{aligned}
& A-B \\
= & (\prod_{1\leq j\leq i,j\notin I}p_jp_{j'})(\prod_{n+i-k+1\leq j\leq h,j\notin I}p_jp_{j'}[(\prod_{i+1\leq j\leq n-i-k,j\notin I}p_j)\alpha(1-\beta) \\
& -(\prod_{i+1\leq j\leq n-i-k,j\notin I}p_{j'})\beta(1-\alpha)]
\end{aligned}$$

where

$$\begin{aligned}\alpha &= \prod_{n-i-k+1\le j\le n+i-k, j\notin I} p_j,\\ \beta &= \prod_{n-i-k+1\le j\le n+i-k, j\notin I} p_{j'},\end{aligned}$$

and

$$\alpha(1-\beta)-\beta(1-\alpha)=\alpha-\beta\ge 0.$$

Thus, $A \ge B$.

Similarly, we have $A' \ge B'$. Therefore, $AA' \ge BB'$. By similar arguments, we can prove the inequalities in other cases.

Now, it is easy to verify that $AA' > BB'$ if and only if

$$\begin{aligned}\{j \mid i+1 \le j \le \min(-i+k, n+i-k), h \in I\} &\ne \emptyset\\ \{j \mid i+1 \le j \le \min(-i+k, n+i-k), h \notin I\} &\ne \emptyset\end{aligned}$$

if and only if, or by Lemma 7.5,

$$(q_{-i}\prod_{j=-i+1}^{-i+k} p_j)^I > 0.$$

□

Similarly, we can prove

Lemma 7.9. *For* $i = 1, \cdots, n-k-2-d$,

$$(q_{i+k+1}\prod_{j=i+1}^{i+k} p_j)^I \ge (q_{i+k+1}\prod_{j=-i}^{i+k} p_j)^I.$$

By Lemmas 7.5-7.9, all terms in (7.11) are non-negative. Next, we show that if $I \neq \emptyset$, then at least one term in (7.11) is positive.

Note that $1 \notin I$. Since $I \neq \emptyset$, there exists a positive integer r such that $1 \leq r < h$, $r \notin I$, and $r+1 \in I$. If $r+1 \leq n-k$, then

$$\begin{aligned} r &\in \{j \mid 1 \leq j \leq n-k, j \notin I\} \\ r+1 &\in \{j \mid 1 \leq j \leq n-k, j \in I\} \end{aligned}$$

and hence

$$(q_0 p_1 \cdots p_k q_{k+1})^I > 0.$$

If $r+1 > n-k$, then choose $i = b+n-k-(r+1) < b$ and we have

$$b-i+1 \leq r, r+1 \leq \min(-b+i+k, n+b-i-k).$$

Hence, by Lemma 7.5,

$$(q_i p_{i+1} \cdots p_{i+k})^I > 0.$$

Finally, we deal with the case that some equality signs hold in $0 \leq p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[n]} \leq 1$. If $p_{[1]} = p_{[2]} = \cdots = p_{[n]}$, then Theorem 7.1 is trivially true. If $p_{[i]} < p_{[i+1]}$, then for sufficiently small $\varepsilon > 0$, we have

$$0 < p_{[1]} + \varepsilon < \cdots < p_{[i]} + i\varepsilon < p_{[i+1]} - (n-i)\varepsilon < \cdots < p_{[n]} - \varepsilon < 1.$$

For them, we already proved the optimality of assignment C^* in Theorem 7.1, that is, for any assignment C, $R(C^*) \geq R(C)$. Now, we can complete our proof of Theorem 7.1 by setting $\varepsilon \to 0$.

7.3 Discussions

An invariant optimal assignment is a nice thing to have in practice, and also an interesting mathematical problem to solve. The existence of invariant optimal assignment has been widely studied for the consecutive-k-out-of-n: F systems and G systems. Usually, the nonexistence of invariant optimal assignments was demonstrated [88, 117, 172]. There are only four nontrivial cases that invariant optimal assignments may exist. The first is an invariant optimal assignment for the consecutive-2-out-of-n: F line conjectured by Derman, Lieberman and Ross [45], and independently proved by Du and Hwang [55], and Malon [116]. In fact, the former proved the harder cycle version which is the second case of existence. Note that the cycle version implies the line version since by setting $p_{[n]} = 1$ ($p_{[1]} = 0$ in the G system), the line problem is reduced to the cycle problem. The third case is an invariant optimal assignment for the consecutive-k-out-of-n: G line for $n \leq 2k$ conjectured by Kuo, Zhang and Zuo [104], and proved by Jallai, Hawkes, Cui and Hwang [95]. The fourth case is its cycle version, the current case. Note that again the cycle version implies the line version, but is much harder. In the line version, one needs only to prove the case $n = 2k$, and the $n < 2k$ case can be reduced to the $n = 2k$ case. No similar reduction is possible for the cycle case. One may wonder whether a simpler proof exists by considering other pairings. In the current chapter, we break the term $q_i p_{i+1} \cdots p_{i+k}$ into two parts $p_{i+1} \cdots p_{i+k}$ and $-p_i \cdots p_{i+k}$. Use the pairing of $p_{-a+i} \cdots p_{-a+i+k-1}$ with $p_{(-a+i)'} \cdots p_{(-a+i+k-1)'}$ for the first part and a similar one for the second part, then compare the C assignment with the C_I assignment. But since the comparison of one part is positive and the other is negative, we have to further compare their sizes and thus complicating the proof. Can we not break the term $q_i p_{i+1} \cdots p_{i+k}$ and find a pairing to work? One such possibility is also to consider the clockwise representation of $R(C)$, namely,

$$R(C) = p_1 \dots p_n + \sum_{i=1}^{n} q_i p_{i-1} \dots p_{i-k}.$$

We then pair each term $q_i p_{i+1} \cdots p_{i+k}$ from the counter-clockwise representation with the term $q_{i'} p_{(i+1)'} \cdots p_{(i+k)'}$ from the clockwise representation. The C_I assignment is better than the C assignment in all cases except when $q_i < q_i'$ and i' not belonging to $i, \cdots, i+k$.

The determination of invariant optimal assignments on lines and cycles is an application of the broader problem of finding an optimal permutation, linear or cyclic, under a certain objective function. This type of problem has been considered before [39, 89] when the arguments of the objective function are $|x_i - x_{i+1}|$ for all i. In the optimal assignment problem, the arguments are products like $q_i p_{i+1} \cdots p_{i+k}$, which seems to raise a new type of optimal permutation problem. In this chapter we give a solution to one such problem, and hope the approach may work for other similar problems [39, 56, 58, 89].

Part IV

Hybrid Networks

Chapter 8

Scalability of an Optical Topology

8.1 Overview

8.1.1 A scalable optical network topology

Emerging high bandwidth applications, such as voice/video services, distributed databases, and network super-computing, are driving the use of single-mode optical fibers as the communication media for the future [33]. Optical passive stars [46] provide a simple medium to connect nodes in a local or metropolitan area network. The single-star optical network with time and/or wavelength division multiplexing have been extensively studied in the past [40, 80, 159]. However, the scalability of the single-star configuration is constrained by the number of wavelengths that can be coupled and separated while maintaining acceptable crosstalk and power budget levels. Recently, a multi-star configuration which efficiently combines space with time and/or wavelength division was proposed in [7] to overcome this limit. In this class of networks, nodes are grouped into clusters with time and/or wavelength multiplexing. Clusters are further interconnected via fiber links to form a cluster interconnection network (CIN) according to some interconnection topology (see figure 8.1). If the cluster size is more than one, self cluster links are provided to enable connectivity among nodes in the same cluster. Wavelength spatial reuse is exploited in the channel set assignment to clusters. This network class has several advantages including

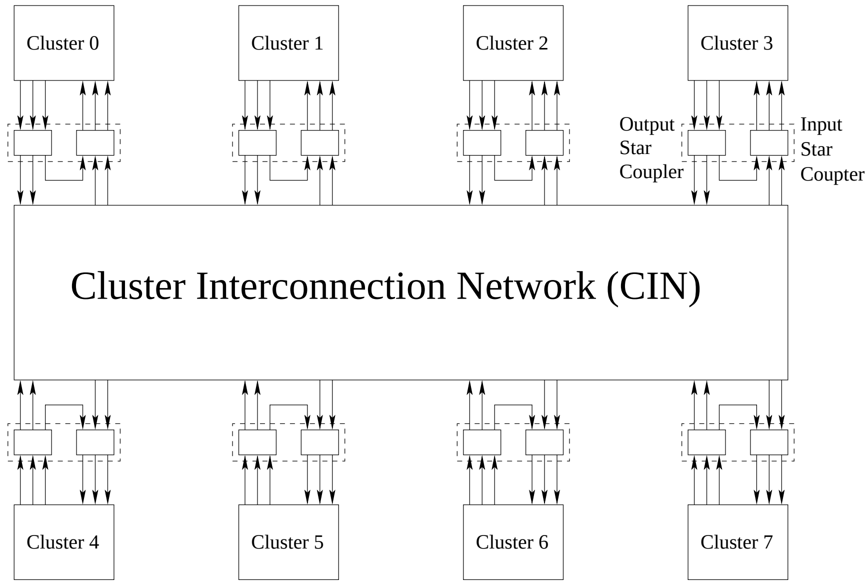


Figure 8.1: A Cluster-based Optical Network Topology

low link density, nice scalability and desirable reconfigurability [7, 9].

The key design issue of this class of networks is the conflict-free channel set assignment to the output star couplers. As the channels sets are valuable resources, it's desirable to share the channel sets among the output star couplers while maintaining the conflict-free transmission. The objective of the conflict-free channel set assignment problem is to find the minimum number of disjoint channel sets required by the conflict-free communication. This optimal conflict-free channel set assignment problem as been studied for various CIN topologies, such as the perfect shuffle [7, 8], hypercube [99, 160], rotator graphs [161] and star graphs [158].

More specifically, consider the case where the CIN topology is undirected. Any two clusters C_1, C_2 which connect to the coupler of another cluster C_3 must be assigned disjoint channel sets which are also disjoint from the channel set of C_3. Otherwise, we get a channel conflict. The problem is then to assign channel sets to clusters (vertices of the CIN) so that any two clusters whose distance is at most 2 have different channel sets.

Generalizing this problem to a graph coloring problem, for each CIN topology G we need to determine the minimum number of colors to color vertices of G so that vertices within distance k from each other have different colors. The most well-understood topology with many good properties is the hypercube. Hence, we shall address this problem on the hypercube, as more precisely discussed in the next subsection.

8.1.2 Related works

An n-cube (or n dimensional hypercube) is a graph whose vertices are vectors of the n dimensional vector space over the field $\{0, 1\}$. There's an edge between two vertices of the n-cube whenever their Hamming distance is exactly 1, where the Hamming distance of two vectors is the number of coordinates they differ. Given n and k, our problem is to find $\chi_{\bar{k}}(n)$, the minimum number of colors needed to color the vertices of an n-cube so that any two vertices of (Hamming) distance at most k have different colors.

Wan [160] proved that

$$n + 1 \leq \chi_{\bar{2}}(n) \leq 2^{\lceil \log_2 (n+1) \rceil} \tag{8.1}$$

and conjectured that $\chi_{\bar{2}}(n) = 2^{\lceil \log_2 (n+1) \rceil}$

Kim et al. [99] showed that

$$2n \leq \chi_{\bar{3}}(n) \leq 2^{\lceil \log_2 n \rceil + 1} \tag{8.2}$$

$$\left(\binom{n}{k/2} \right) \leq \chi_{\bar{k}}(n) \leq (k+1) \left(\frac{k+2}{2} \right)^{\frac{k(k+2)}{8} \lceil \log_2 n \rceil} \tag{8.3}$$

$$2\left(\binom{n-1}{\frac{k-1}{2}}\right) \leq \chi_{\bar{k}}(n) \leq (k+1)\left(\frac{k+2}{2}\right)^{\frac{k(k+2)}{8}\lceil \log_2 n\rceil} \quad (8.4)$$

where $\left(\binom{n}{m}\right) = \sum_{i=0}^{m}\binom{n}{i}$

The upper bounds in (8.1) and (8.2) are fairly tight. In (8.1), the upper and lower bounds coincide when $n+1$ is an exact power of 2, while if n is an exact power of 2 upper and lower bounds of (8.2) meet. However, the uppers bounds in (8.3) and (8.4) are not tight. In fact, when $k = 2$ and $k = 3$ they are different from that of (8.1) and (8.2). A natural approach to get an upper bound of $\chi_{\bar{k}}(n)$ is to find a coloring of the n-cube with as few colors as possible. We shall use this idea and properties of linear codes (to be introduced in the next section) to give tighter bounds for general k which imply (8.1) when $k = 2$ and (8.2) when $k = 3$. In fact, the upper bounds in (8.1) and (8.2) are straight application of the Hamming code [141].

All existing lower bounds can be improved slightly by applying existing results on the main coding theory problem [141].

The remaining of the paper is organized as follows. Section 8.2 introduces concepts and results from coding theory needed for the rest of the paper. Section 8.3 discusses our results and section 8.4 concludes the chapter.

8.2 Preliminaries

The following concepts and results can be found in many standard texts on coding theory such as [141].

Let $A = \{0, 1, \ldots q-1\}$ where $q \geq 2$ is an integer, and A^n be the set of all n-dimensional vectors (or strings of length n) over A. Any non-empty subset C of A^n is called a *q-ary block code*. Our main concern is when $A = \{0, 1\}$, in which case C is called a *binary code*. From now on, the term codes refers to binary codes unless specified otherwise. Each element of C is called a *codeword*. Let $M = |C|$ then C is called an *(n, M)-code*. The *Hamming distance* between any two codewords $c = c_1c_2 \ldots c_n$ and $d = d_1d_2 \ldots d_n$ are defined to be $d(c, d) = |\{i : c_i \neq d_i\}|$. For $c \in C$, the *weight* of c denoted by $w(c)$ is the number of 1's in c. The *minimum distance* $d(C)$ of a code C is the least Hamming distance between two different codewords in C. If $C \subset A^n$, $|C| = M$, and $d(C) = d$ then C is called an *(n, M, d)-code*.

One of the most important problem in coding theory is to find $A_q(n, d)$, the largest size M such that a q-ary (n, M, d)-code exists. This problem is so important that it is referred to as *the main coding theory problem*. In case $q = 2$, we will write $A(n, d)$ instead of $A_2(n, d)$. The following theorems are standard results in coding theory and the reader is referred to [141] for proofs.

Theorem 8.1. $A(n, 2t + 1) = A(n + 1, 2t + 2)$

Theorem 8.2.

$$A(n, 2t+1) \leq \frac{2^n}{\sum_{i=0}^{t} \binom{n}{i} + \frac{1}{\left\lfloor \frac{n}{t+1} \right\rfloor} \binom{n}{t} \left(\frac{n-t}{t+1} - \left\lfloor \frac{n-t}{t+1} \right\rfloor \right)}$$

Theorem 8.2 is a special case of the Johnson bound [141].

It is clear that all n-dimensional vectors over $\{0, 1\}$ form an n-dimensional vector space, which we denote by $V_n(2)$. A code $C \subset V_n(2)$ is called a *linear code* if C is a linear subspace of $V_n(2)$. Moreover, C is called a *$[n, m]$-code* if it has dimension m. As expected, if C also has minimum distance d then it is called an *$[n, m, d]$-code*. Notice that the square brackets automatically refer to linear codes. A $m \times n$ matrix G is called the *generator matrix* of an $[n, m]$-code C if its rows form a basis of C. In other words, every codewords in C is a linear combination of some rows of G. Given an $[n, m]$-code C, an $(n - m) \times n$ matrix H is called the *parity check matrix* of C if $c \in C \Leftrightarrow cH^T = 0$. From coding theory, we know that specifying a linear code using generator matrix and using parity check matrix are equivalent. In fact, there are ways to construct the parity check matrix from the generator matrix of a code and vice versa. For a vector $x \in V_2(n)$, the *syndrome* of x associated with a parity check matrix H is defined to be $synd(x) = xH^T$.

Given an $[n, m, d]$-code C, the *standard array* of C is a $2^{n-m} \times 2^m$ table where each row is a (left) coset of C. This table is well defined since elements of C form an Abelian subgroup of $V_2(n)$ under addition, and from basic algebra we know that the cosets of a group partition the group uniformly. The first row of the standard array contains C itself. The first column of the standard array contains the minimum weight elements from each coset. These are called *coset leaders*. Each entry in the table is the sum of the codeword on the top of its column and its coset leader. Since each pair of distinct codewords has Hamming distance at least d, each pair of elements in the same row also has Hamming distance at least d. It is a basic fact from coding theory that all elements on the same row of the standard array have the same syndrome and different rows have different syndromes.

We conclude this section by an important theorem. Again, the reader is referred to [141] for a proof.

Theorem 8.3. *If H is a $(n - m) \times n$ matrix where any $d - 1$ columns of H are linearly independent and there exists d linearly dependent columns in H, then H is the parity check matrix of an $[n, m, d]$-code.*

8.3 Main Results

Lemma 8.4. *If $k = 2t$, then*

$$\chi_{\bar{k}}(n) \geq \sum_{i=0}^{t} \binom{n}{i} + \frac{1}{\left\lfloor \frac{n}{t+1} \right\rfloor} \binom{n}{t} \left(\frac{n-t}{t+1} - \left\lfloor \frac{n-t}{t+1} \right\rfloor \right)$$

When $k = 2t + 1$, we have

$$\chi_{\bar{k}}(n) \geq 2 \left(\sum_{i=0}^{t} \binom{n-1}{i} + \frac{1}{\left\lfloor \frac{n-1}{t+1} \right\rfloor} \binom{n-1}{t} \left(\frac{n-1-t}{t+1} - \left\lfloor \frac{n-1-t}{t+1} \right\rfloor \right) \right)$$

Proof. Given a proper m-coloring of the n-cube with parameters n and k, let $S_i, 1 \leq i \leq m$ be the set of vertices which were colored i. Clearly for each i, S_i forms an $(n, |S_i|, d)$-code where $d \geq k + 1$. With the note that $A(n, d)$ is a decreasing function in d, we have

$$2^n = \sum_{i=1}^{m} |S_i| \leq \sum_{i=1}^{m} A(n, k+1) = mA(n, k+1)$$

In particular, $\chi_{\bar{k}}(n) \geq \frac{2^n}{A(n,k+1)}$. When $k = 2t$, theorem 8.2 yields

$$\chi_{\bar{k}}(n) \geq \sum_{i=0}^{t} \binom{n}{i} + \frac{1}{\left\lfloor \frac{n}{t+1} \right\rfloor} \binom{n}{t} \left(\frac{n-t}{t+1} - \left\lfloor \frac{n-t}{t+1} \right\rfloor \right)$$

When $k = 2t + 1$, combining theorems 8.1 and 8.2 gives us

$$
\begin{aligned}
\chi_{\bar{k}}(n) &\geq \frac{2^n}{A(n,k+1)} \\
&= \frac{2^n}{A(n,2t+2)} \\
&= \frac{2^n}{A(n-1,2t+1)} \\
&\geq 2\left(\sum_{i=0}^{t}\binom{n-1}{i} + \frac{1}{\left\lfloor\frac{n-1}{t+1}\right\rfloor}\binom{n-1}{t}\left(\frac{n-1-t}{t+1} - \left\lfloor\frac{n-1-t}{t+1}\right\rfloor\right)\right) \square
\end{aligned}
$$

Lemma 8.5. *Let* $\left(\binom{n}{m}\right)$ *denotes* $\sum_{i=0}^{m}\binom{n}{i}$*, then we have*

$$\chi_{\bar{k}}(n) \leq 2^{\left\lfloor \log_2\left(\left(\binom{n-1}{k-1}\right)\right)\right\rfloor + 1} \text{ when } k \text{ is even}$$

$$\chi_{\bar{k}}(n) \leq 2^{\left\lfloor \log_2\left(\left(\binom{n-2}{k-2}\right)\right)\right\rfloor + 2} \text{ when } k \text{ is odd}$$

Proof. Let C be an $[n, m, k+1]$-code. The key to our proof is the observation that every two elements on the same row of C's standard array are at least $k+1$ apart. Thus, coloring each row of of C's standard array by one separate color would give us a valid coloring. The number of colors used is 2^{n-m} – the number of rows of C's standard array. Consequently, one way to obtain a good coloring of the n-cube is to find a linear $[n, m, k+1]$ code with as large an m as possible. Moreover, by theorem 8.3 we can construct a linear $[n, m, d]$ code by trying to build its parity check matrix H, which is an $(n-m) \times n$ matrix with the property that d is the largest number for any $d-1$ columns of H to be linearly independent. Also, since all elements of a coset of the code (a row of its standard array) have the same syndrome, we can use H to color each vector $x \in V_2(n)$ with $synd(x) = xH^T$.

Let $p = \left\lfloor \log_2 \left(\binom{n-1}{d-2} \right) \right\rfloor + 1$, then clearly

$$\binom{n-1}{1} + \binom{n-1}{2} + \ldots \binom{n-1}{d-2} < 2^p - 1$$

Now, we describe a procedure to construct a $p \times n$ parity check matrix H by trying to choose the column vectors of H. The first column vector can be any non-zero vector. Suppose we already had a set V of i vectors so that any $d-1$ of them are linearly independent. The $(i+1)^{th}$ vector can be picked as long as it is not in the span of any $d-2$ vectors in V. In other words, since we're working over the field $\mathbb{F}_2$, the new vector can't be the sum of any $d-2$ or less vectors in V. The total number of *invalid* vector is at most $\binom{i}{1} + \binom{i}{2} + \ldots \binom{i}{d-2}$ (this is an increasing function in i). Consequently, as long as $\binom{i}{1} + \binom{i}{2} + \ldots \binom{i}{d-2} < 2^p - 1$ then we can still add a new column into H.

As we've noticed,

$$\binom{n-1}{1} + \binom{n-1}{2} + \ldots \binom{n-1}{d-2} < 2^p - 1$$

so we can choose n column vectors of H. This bound in coding theory literature is a special case of the *Gilbert-Varshamov Bound* on the existence of linear codes.

The linear code C whose parity check matrix is H has minimum distance at least d (and size $|C| = 2^{n-p}$). The number of rows of the standard array of C is 2^p. For our problem, we want $d = k + 1$. The linear code C constructed gives a valid coloring using 2^p colors, so

$$\chi_{\bar{k}}(n) \leq 2^p = 2^{\left\lfloor \log_2 \left(\binom{n-1}{k-1} \right) \right\rfloor + 1}$$

This inequality holds regardless of k being odd or even and thus it proves our lemma for the even k case. However, when k is odd we are able to do better than that.

Notice that adding an even parity bit to each vector of $V_2(n-1)$ gives us half of $V_2(n)$, and adding an odd parity bit would give us the other half. When k is odd, we just showed that the $(n-1)$-cube can be colored using with $a = 2^{\lfloor \log_2(\binom{n-2}{k-2}) \rfloor + 1}$ colors so that if two vertices have the same color then their distance is at least k. From this, we can obtain a coloring of the n-cube as follows. First add an even parity bit to each vertex of the $(n-1)$-cube and color the cube with a colors, then add an odd parity bit and use a completely different set of a colors to color the cube. This is clearly a coloring of the n-cube using $2a = 2^{\lfloor \log_2(\binom{n-2}{k-2}) \rfloor + 2}$ colors. What remained to be shown is that this coloring is valid with parameters n and k.

For $x \in V_2(n)$, let $x' \in V_2(n-1)$ be the vector obtained from x by deleting the parity bit added. By the construction of the coloring, if two vertices x and y of the n-cube have the same color then $d(x', y') \geq k$, and the same type of parity bit (even or odd) was added to them to get x and y. It is clear that if $d(x', y') \geq k+1$, then $d(x, y) \geq k+1$. If $d(x', y') = k$, then since k is odd, x' and y' must have had different bits added to; consequently, $d(x, y) = k+1$. In sum, if two vertices x and y of the n-cube have the same color then $d(x, y) \geq k+1$, and so we had a valid coloring with parameters n and k. □

Lemma 8.4 and 8.5 can be summarized by the following theorem.

Theorem 8.6. *Let $t = \lfloor \frac{k}{2} \rfloor$ and $(\binom{n}{m})$ denotes $\sum_{i=0}^{m} \binom{n}{i}$, then*

when k is even, we have

$$\sum_{i=0}^{t} \binom{n}{i} + \frac{1}{\lfloor \frac{n}{t+1} \rfloor} \binom{n}{t} \left(\frac{n-t}{t+1} - \left\lfloor \frac{n-t}{t+1} \right\rfloor \right) \leq \chi_{\bar{k}}(n) \leq 2^{\lfloor \log_2(\binom{n-1}{k-1}) \rfloor + 1},$$

and when k is odd, we have

$$2\left(\sum_{i=0}^{t}\binom{n-1}{i}+\frac{1}{\left\lfloor\frac{n-1}{t+1}\right\rfloor}\binom{n-1}{t}\left(\frac{n-1-t}{t+1}-\left\lfloor\frac{n-1-t}{t+1}\right\rfloor\right)\right)$$
$$\leq \chi_{\bar{k}}(n) \leq 2^{\left\lfloor\log_2\left(\binom{n-2}{k-2}\right)\right\rfloor+2}$$

Also note that since

$$2^{\left\lfloor\log_2\left(\binom{n-1}{2-1}\right)\right\rfloor+1} = 2^{\lfloor\log_2 n\rfloor+1} = 2^{\lceil\log_2(n+1)\rceil},$$

and

$$2^{\left\lfloor\log_2\left(\binom{n-2}{3-2}\right)\right\rfloor+2} = 2^{\lfloor\log_2(n-1)\rfloor+2} = 2^{\lceil\log_2 n\rceil+1}$$

inequalities (8.1) and (8.2) are direct consequences of this theorem.

8.4 Discussions

The key to get a good coloring is to find the parity check matrix H when k is even. As can be seen, the proof of theorem 8.6 implicitly gave us an algorithm to construct H, but it is still not very constructive. However, in the case $k = 2$ (and thus in case $k = 3$) we can explicitly construct H. To see this, consider the Hamming code $\mathcal{H}_2(r)$, which is a $[2^r - 1, 2^r - 1 - r, 3]$ code. Its parity check matrix $H(r, 2)$ has dimensions $r \times (2^r - 1)$. Let $r = \lceil\log_2(n + 1)\rceil$, then $2^r - 1 \geq n$. So, if we remove the last $2^r - 1 - n$ columns of $H(r, 2)$, then we get a parity check matrix of an $[n, n - \lceil\log_2(n + 1)\rceil, 3]$ code. This code gives us a coloring of the n-cube with parameters n and 2 using $2^{\lceil\log_2(n+1)\rceil}$ colors. This proves the upper bound of (8.1).

Besides the *Johnson bound* we used, other known upper bounds of $A(n,d)$ might give us better lower bound of $\chi_{\bar{k}}(n)$ such as the *Plotkin bound*, the *Elias bound* and the *Linear Programming bound*. However, applying these bounds breaks the problem into various cases and doesn't give us a significantly better result.

For some special values of n and k, we can get better results by considering some specially good linear codes. The Golay $\mathcal{G}_{24}$ code is a binary $[24,12,8]$-code whose generator matrix has the form $G = [I_{12} \mid A]$ where A was *magically* given by Golay in 1949 [115].

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

This shows that $\chi_{\bar{7}}(24) \leq 2^{12}$, while our theorem gives

$$2^{12} \leq \chi_{\bar{7}}(24) \leq 2^{17}$$

Thus in fact $\chi_{\bar{7}}(24) = 2^{12}$ (!!!), and our upper bound is far off in this case. Puncturing $\mathcal{G}_{24}$ (i.e. removing any coordinate position) at any coordinate gives us $\mathcal{G}_{23}$, a (linear) $[23, 12, 7]$-code. This implies $\chi_{\bar{6}}(23) \leq 2^{11}$, while again our theorem gives too large an upper bound :

$$2^{11} \leq \chi_{\bar{6}}(23) \leq 2^{15}$$

However, again we obtain $\chi_{\bar{6}}(23) = 2^{11}$. We summarize the cases where the exact values of $\chi_{\bar{k}}(n)$ are known as follows

- $\chi_{\bar{7}}(24) = 2^{12}$ (shown above).
- $\chi_{\bar{6}}(23) = 2^{11}$ (shown above).
- $\chi_{\bar{2}}(2^m - 1) = 2^m$ (immediate from theorem 8.6)
- $\chi_{\bar{2}}(2^m - 2) = 2^m$ (immediate from theorem 8.6)
- $\chi_{\bar{3}}(2^m) = 2^{m+1}$ (immediate from theorem 8.6)
- $\chi_{\bar{3}}(2^m - 1) = 2^{m+1}$ (immediate from theorem 8.6)

One might wonder if we can get more exact values of $\chi_{\bar{k}}(n)$'s using the same method. Our lower bound was proven using Johnson's bound, a slight extension of the sphere packing bound. To construct a linear code that matches the sphere packing bound, the code has to be perfect, namely there exist a radius r such that the spheres $S(c, r) = \{a \mid d(c, a) \leq r\}$ around each codeword c covers the whole space. The binary $[2^r - 1, 2^r - 1 - r, 3]$ Hamming code $\mathcal{H}(r)$ and the Golay $[23, 12, 7]$-code are perfect. This is why we were able to obtain the exact values as above. Thus, the question comes down to “does there exist any other binary perfect codes besides the Hamming codes and the Golay codes ?". The answer was given by Tietavainen (1973) with most of the work done previously by van Lint :

Theorem 8.7. *A nontrivial perfect q-ary code C, where q is a prime power, must have the same parameters as either a Hamming code or one of the Golay codes $\mathcal{G}_{23}$ (binary) or $\mathcal{G}_{11}$ (ternary).*

However, as we have mentioned the Johnson bound is slightly better than the sphere packing bound. That is how we got two additional values $\chi_{\bar{2}}(2^m - 2) = 2^m$ and $\chi_{\bar{3}}(2^m - 1) = 2^{m+1}$. This comes from the fact that the *shorten* Hamming $[2^m - 2, 2^m - 2 - m, 3]$ is *nearly perfect*, i.e. it gives equality in the Johnson bound. Again, does there exist any other nearly perfect codes ? Lindstrom (1975) answered in the affirmative : the only other nearly perfect code is the *punctured Preparata code*. This code has parameter $(2^m - 1, 2^{2^m - 2m}, 5)$. Unfortunately, this is not a binary *linear* code, so our coloring doesn't quite work.

That is not the end of our hope. Hammons, Kumar, Calderbank, Sloane and Sole [34, 74, 75] showed that the Preparata code is Z_4-linear, namely it can be constructed easily from a linear code over Z_4 as the binary image under the Gray map. This map transforms Lee distance in Z_4 to Hamming distance in Z_2. The mapping is simple, but the construction of the code is quite involved and I'm still working on it. I would hope to get a very nice upper bound for $\chi_{\bar{4}}(n)$ and $\chi_{\bar{5}}(n)$ from this break-through paper in coding theory.

Chapter 9

Error Spreading for Multimedia Streaming

9.1 Overview

Due to the phenomenal growth of multimedia systems and their applications, there have been numerous research efforts directed at providing a *continuous media (CM)* service over varying types of networks. With the boom of the Internet, continuous media like audio and video are using the Internet as the principal medium for transmission. However, the Internet provides a *single class best effort* service, and does not provide any sort of guarantees [28]. A network characteristic of special concern to this chapter is transmission errors, and specifically the dropping of data packets. Packets are dropped when the network becomes congested, and given the nature of this phenomenon, strings of successive packets are often dropped [25, 129], leading to significant bursty errors [168].This bursty loss behavior has been shown to arise from the drop-tail queuing discipline adopted in many Internet routers [127].

Handling bursty errors has always been problematic, especially since no good models exist for its prediction. On the other hand, most applications, especially realtime Multimedia applications, do not tolerate bursty error, making it imperative that they be handled in a good manner. Perceptual studies on continuous media viewing have shown that user dissatisfaction rises dramatically when bursty error goes beyond a certain threshold [162–164]. This is especially so for audio, where the threshold is quite small, and hence this issue is quite pressing for applications like Internet phone.

These observations point quite solidly to the need for development of efficient mechanisms to control bursty errors in continuous media streaming through networks. Redundancy is the key to handling packet loss/damage in standard communication protocols. There are two main classes of schemes, namely the *reactive* schemes and the *proactive* schemes.

Reactive schemes respond by taking some action once transmission error has been detected, while pro-active schemes take some action in advance to avoid errors . A protocol such as TCP is reactive since the receiver sends a feedback to the sender upon detecting an error, in response to which the sender will re-transmit the lost or corrupted packet. The reaction can be initiated by the source or the sink. *Source initiated reaction* occurs in schemes based on feedback combined with retransmission like [28, 29, 138]. The feedback control can be based on stream rate [118, 171], bandwidth [73], loss/delay [28] and a wide variety of network QoS parameters [10, 135, 150]. *Sink initiated reaction* occurs in reconstruction based schemes like [3, 157]. Coding data in an error correcting manner before transmission is a pro-active scheme where corrupted packets can be reconstructed ([76] and *Forward Error Correction Codes* [26, 27]).

Each of these classes of schemes requires extra bandwidth, for feedback and retransmission in the first, and for extra bits in the second category. This need for extra bandwidth can exacerbate the problem, especially since network congestion is the principal culprit for the bursty errors. One more approach that has been proposed, is to fulfill the real time needs of CM applications with other services like *RSVP and RTP*, which offer varying degree of performance guarantees [142, 169]. Services like RTP/RSVP require that some resource allocation and/or reservation mechanism be provided by the network [28]. Since these mechanisms are not yet widely deployed in the Internet, our focus has not been on them.

Recent work ([143, 170]) has proposed schemes where the overall characteristics of the data being transmitted can be used to control the transmission error. For example, [170] has proposed selectively dropping video frames on the sender side, based on a *cost-benefit analysis* which takes into account the desired Quality of Service (QoS). This is quite effective in a LAN (senders are known and cooperative) or the Internet using RED gateways where during congestion, the probability that the gateway notifies a particular connection to reduce its window is roughly proportional to that connection's share of the bandwidth through the gateway [71]. While drop-tail queuing discipline is still adopted in many routers [127], this scheme may not be directly applicable yet.

In this chapter we propose a new type of scheme for handling bursty errors, which we call *error spreading*. A key advantage of this scheme is that it is not based on redundancy, and hence requires absolutely no extra bandwidth. The main idea is that we do not try to reduce overall error, but rather tradeoff bursty error (the *bad error*) for average error (the *good error*). Perceptual study of continuous media viewing [162–164] has shown that a reasonable amount of overall error is acceptable, as long as it is spread out, and not concentrated in spots. A similar approach has been taken by [168]. But they have used a random scrambling techniques with redundant reconstruction for audio, and as stated by them they have not investigated the buffer requirements. We have established clearly the bounds and the relationship between buffer requirement and user perceived quality in a bounded bursty network error scenario.

In this chapter we make several contributions. First, we formulate the problem of error handling in continuous media transmission as a tradeoff between the user QoS requirements, network characteristics, and sender resource availability. Second, we provide a complete analytical solution for the special case where the network errors are bounded. While this solution may be of actual use in some specialized networks, e.g., a tightly controlled real-time network, its principal use is in providing important mathematical relationships that can be used as the basis of protocols for general networks. Next, we use this analysis to develop such a protocol for networks where there is no bound on the error. Finally, we present results of an experimental evaluation that illustrates the benefits of the proposed scheme.

This chapter is organized as follows: section 9.2 formulates the problem and section 9.3 presents a mathematical analysis of the bounded network error case. Section 9.4 presents the transmission protocol, while section 9.5 describes its evaluation. Section 9.6 concludes the chapter.

9.2 Problem Formulation

This section briefly discusses the content based continuity QoS metrics introduced in [162]. Then, we define our problem based on this metrics.

9.2.1 QoS metrics

For the purpose of describing QoS metrics for lossy media streams, CM stream is envisioned as a flow of data units (referred to as logical data units, or LDUs, in the uniform framework of [147]). In our case, we take a video LDU to be a frame, and an audio LDU to consist of 8000/30, i.e. 266 samples of audio[1]. Given a rate for streams consisting of

[1] SunAudio has 8-bit samples at 8kHz, and an audio frame constitutes of 266 such samples equivalent to a play time of one video frame, i.e. 1/30 seconds.

these LDUs, we envision that there is a time slot for each LDU to be played out. In the ideal case a LDU should appear at the beginning of its time slot. This section uses only the content based continuity metrics proposed in [162]. Issues on rate and drift factors (discussed in [162]) are not considered. Note also that we shall use the term LDU and frame interchangeably, since *frame* is very commonly used in Multimedia community.

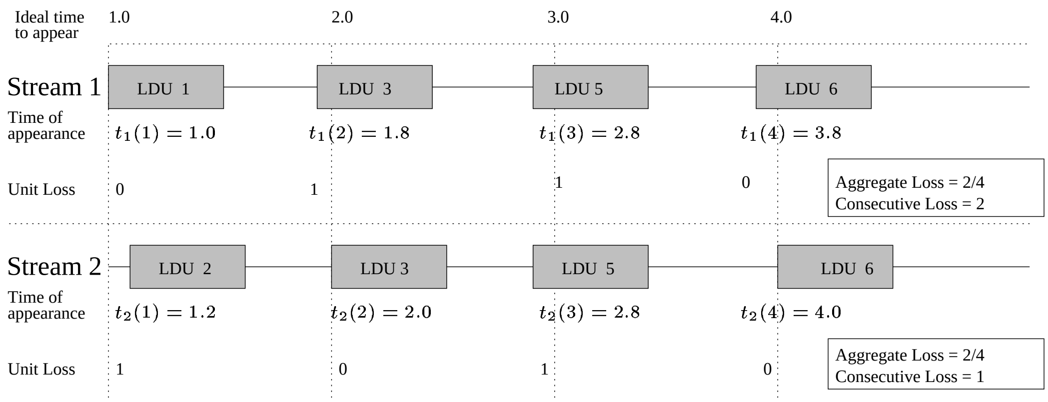


Figure 9.1: Two Sample Streams Used to Explain the QoS Metrics

The above figure is from [162]. Ideal contents of a CM stream are specified by the ideal contents of each LDU. Due to loss, delivery error or resource over-load problems, appearance of LDUs may deviate from this ideal, and consequently lead to discontinuity. The metrics of continuity are designed to measure the average and bursty deviation from the ideal specification. A loss or repetition of a LDU is considered a unit loss in a CM stream. (A more precise definition is given in [162].) The aggregate number of such unit losses is the *aggregate loss (ALF)* of a CM stream, while the largest consecutive non-zero loss is its *consecutive loss (CLF)*. In the example streams of Fig. 9.1, stream 1 has an aggregate loss of 2/4 and a consecutive loss of 2, while stream 2 has an aggregate loss of 2/4 and a consecutive loss of 1. The reason for the lower consecutive loss in stream 2 is that its losses are more spread-out than those of stream 1. Note that the metrics already takes care of losses (both consecutive and aggregate) that arise due timing drifts [162].

In a user study [163] it has been determined that the tolerable value for consecutive video frame losses were determined to be two frames. For audio this limit was about three frames.

9.2.2 Problem Statement

One of the most important factors that affect the quality of a CM stream is the *Consecutive Loss Factor* [162] (CLF). Network often lose frames in bursts, alternating between lossy burst and successful burst [25, 127, 129, 168]. This often causes unacceptably high CLF. Our objective is to reduce CLF given the same network characteristics. The main idea is *loss spreading*, or distributing consecutive loss over some time period.

For example, suppose we sent a sequence of 17 consecutive video frames numbered 1 to 17. During transmission, a network bursty error of size 7 occurs which causes the loss of frames numbered 7 to 13, as shown in the first row of Table 9.1. This causes the stream to have a CLF of $7/17$.

Now suppose we permute this sequence of frames before transmission so that consecutive frames become far apart in the sequence, the CLF can be reduced significantly. To illustrate this idea, consider the frame transmission order shown in the second row of Table 9.1. With exactly the same bursty error once again consecutive frames are lost, except this time they are consecutive only in the permuted domain. In the original domain these are spread far apart.

Clearly, if the 17 frames were sent in this order, we would have had a CLF of only $1/17$. Table 9.1 summarizes our example by giving three sequences and their corresponding CLFs. The first sequence is the natural order of frames, the second is the permuted

	Frame sequence	CLF
In order	01 02 03 04 05 06 [07 08 09 10 11 12 13] 14 15 16 17	$7/17$
Permuted	01 06 11 16 04 09 [14 02 07 12 17 05 10] 15 03 08 13	$1/17$
Un-permuted	01 [02] 03 04 [05] 06 [07] 08 09 [10] 11 [12] 13 [14] 15 16 [17]	$1/17$

Table 9.1: An example of how the order of frames sent affects CLF

order, and the third is the un-permuted order observed at the receiver's side. The third sequence was presented to show how the loss has been spread out over the original sequence. The boxed numbers represent lost frames.

This example motivates the following problem.

Bursty Error Reduction Problem (BERD)

- **Objective :** to reduce the bursty error, i.e. CLF, to an perceptually acceptable level (by spreading it out over the stream).

- **Input parameters:**

  - m is the sender's buffer size, in terms of LDUs. m is determined by the sender's operating environment and its current status.

  - p is the upper bound on the size of a bursty loss in the communication channel, within a window of m LDUs (we relax this assumption in section 9.4).

  - k is the user's maximum acceptable CLF.

- **Output :** a permutation function π on $S = \{1, 2, 3, \ldots, m\}$ which decides the order in which a set of m consecutive LDUs must be sent. Moreover, the system is expected to give the lower bound k_0 which is the minimum CLF that can be supported in this constrained environment.

- **Assumption :** two consecutive bursty loss are at least m LDUs apart.

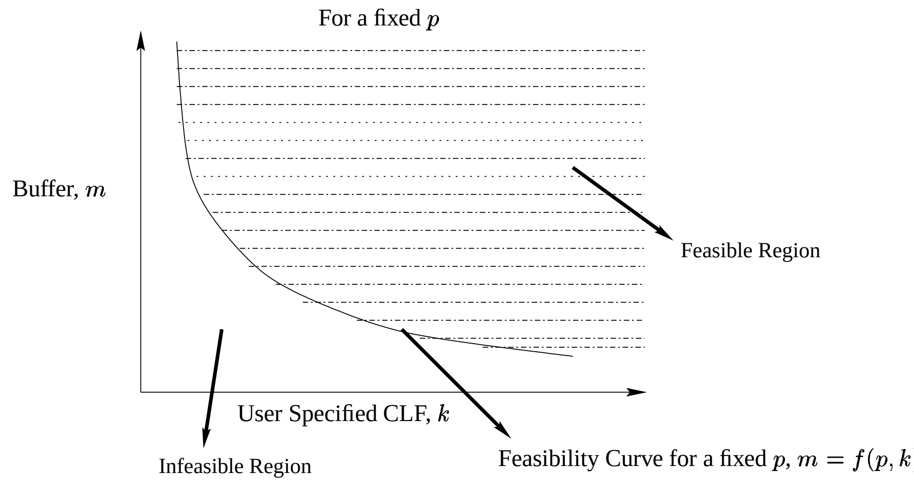


Figure 9.2: Part of Deterministic Solution Space

Figure 9.2 visualizes how the solution space for a particular value of p would appear. The boundary of the curve is essentially what we found. Above it is the feasible region, where intuitively if we increase m, then k should still be the same or less. There is a typical trade off between buffer size m and CLF k. The greater m is, the less k we can support but also the greater memory requirement and initial delay time. Given m_0, line $m = m_0$ cuts the boundary curve at k_0 at or above which we can support. Conversely, given k_0, line $k = k_0$ intersects the curve at m_0 at or above which the buffer size must be in order to support k_0.

There are several points worth noticing. Firstly, we deal only with data streams that have no inter-frame dependency such as Motion-JPEG or uncompressed data streams (audio, video, sensor data, $\ldots$). The reason for this is that this allows us to consider every frame to be equally important; thus, we can permute the frames in any way we would like to. Secondly, the frames in these types of streams have relatively comparable sizes. For example, a sequence of MJPEG frames only has a change in size significantly when the scene switches. So, no matter if it is us who send the frames by breaking them up into equal size UDP packets or it is the transport layer interface (TLI) which does so, a consecutive packet loss implies a proportional consecutive frame loss. Finally, to satisfy our assumption that two consecutive lost windows are at least m frames apart, closer lost windows can be combined and consider to be a larger lost window.

9.3 Bounded Network Error Case

In this section, we will discuss the case where the bound p of continuous network loss is known. For convenience, we first state the problem in purely mathematical terms and establish some notations to be used throughout the proof.

We are given positive integers m and p. Let S_m denotes the set of all permutations acting on $[m] = \{1, 2, \ldots m\}$. For any permutation $\pi \in S_m$, the sets

$$W_i^\pi = \{\pi_i, \pi_{i+1}, \ldots \pi_{i+p-1}\}, 1 \leq i \leq m$$

are called the *sliding windows* of size p (of π), where the indices are calculated modulo m, then plus 1. Thus, when $1 \leq i \leq m - p + 1$, $W_i^\pi = \{\pi_i, \pi_{i+1}, \ldots \pi_{i+p-1}\}$, and when $m - p + 1 < i \leq m$, $W_i^\pi = \{\pi_i, \ldots, \pi_m, \pi_1, \ldots, \pi_{i+p-m-1}\}$

For any pair of integers k and l such that $1 \leq k < l \leq m$, let $[k, l]$ denotes the set $\{k, k+1, \ldots l\}$. Let $\{c_i^\pi\}_1^m$ be the sequence of integers defined as follows.

$$c_i^\pi = \begin{cases} \max\{|[k,l]|,\ [k,l] \subseteq W_i^\pi\} & \text{if } 1 \le i \le m-p+1 \\ \max\{|[k,m]| + |[1,l]|, & \text{if } m-p+1 < i \le m \\ \quad [k,m] \subseteq \{\pi_i, \dots \pi_m\}, & \\ \quad [1,l] \subseteq \{\pi_1, \dots \pi_{p+i-m-1}\}\} & \end{cases}$$

Let $C^\pi = \max\{c_i^\pi | 1 \le i \le m\}$. Then k_0 is defined to be

$$k_0 = \min\{C^\pi, \pi \in S_m\}$$

Our objective is to find k_0 as a function of m and p. Moreover, we also wish to specify a permutation π so that $C^\pi = k_0$.

Informally, when $1 \le i \le m-p+1$, c_i^π is the maximum number of consecutive integers in W_i^π. While if $m-p+1 < i < m$, c_i^π is the sum of two quantities x and y, where x is the length of the longest consecutive integer sequence in $\{\pi_i, \dots, \pi_m\}$ which ends in m, and y is the length of the longest consecutive integer sequence in $\{\pi_1, \dots, \pi_{i+p-m-1}\}$ which starts at 1. The reason for this is that suppose we apply our permutation to two adjacent buffers of size m, we would like our permutation to also deal with the case where the network loss burst occurs across these two buffers.

The value of k_0 and permutation π depends tightly on the relationship between p and m. Section 9.3.1 presents our solution based on two cases divided into two lemmas: (a) the case where $p \le \frac{m}{2}$, in which we have more freedom to choose our permutation; (b) the case where $\frac{m}{2} < p < m$. Lastly, section 9.3.2 summarizes our work on the bounded error case by a theorem and an algorithm which gives us a *good* permutation given m and p.

Remark 9.1. If p is known and m is fixed, then $k_0 = 0$ when $p = 0$ and $k_0 = m$ when $p \ge m$.

Throughout this section, we assume m and p were given, unless specified otherwise.

9.3.1 Deterministic Solutions

Lemma 9.2. *If* $0 < p \leq \frac{m}{2}$ *then* $k_0 = 1$

Proof. Since $p > 0$, we have $k_0 \geq 1$. So, to prove $k_0 = 1$ it is sufficient to specify a permutation π on $S = \{1, 2, 3, \ldots m\}$ so that $C^\pi = 1$. To avoid possible confusion, we would like to point out that for any $\pi \in S_m$, $\pi(i)$ specifies the position of i in the permuted sequence where i is sent to, while π_i is the number at position i in one line notation of π. For example, if $\pi = 4\ 1\ 3\ 6\ 2\ 5$ then $\pi_1 = 4$, $\pi_2 = 1$, but $\pi(1) = 2$ and $\pi(2) = 5$.

We consider two cases based on the parity of m as follows.

- **Case 1 :** m **is odd**

  Let $m = 2m' + 1$, $p' = \min\{j, j \geq p \wedge \gcd(m, j) = 1\}$. Since $p \leq \frac{m}{2}$, we have $p \leq m'$. Moreover, $\gcd(m, m') = 1$, hence $p' \leq m' < \frac{m}{2}$. We now construct a permutation π such that $C^\pi = 1$. Let π be defined as follows.

  $$\pi(i) = ((i-1).p' \bmod m) + 1,\ 1 \leq i \leq m \tag{9.1}$$

  We first need to prove that π is indeed a permutation. As $\gcd(m, p') = 1$, p' generates the group $\mathbb{Z}_m$ of integers modulo m ($\mathbb{Z}_m = \{0, 1, \ldots m-1\}$). Thus the sets $\{0, 1, \ldots m-1\}$ and $\{0, p', 2p' \bmod m, \ldots (m-1)p' \bmod m\}$ are identical. So (9.1) defines a valid $\pi \in S_m$.

  Secondly, we need show that $C^\pi = 1$. Let $\pi = \pi_1, \pi_2, \ldots \pi_m$ in one line notation. If $C^\pi \geq 2$ then there exists i and j, $1 \leq i < j \leq m$ such that **either** $|\pi_i - \pi_j| = 1$ and both π_i and π_j belong to the same sliding window W_k^π for some $1 \le k \le m - p + 1$ **or** $\{\pi_i, \pi_j\} = \{1, m\}$ and both π_i and π_j belong to the same sliding window W_k^π for some $m - p + 1 < k \le m$.

Note that by definition of π, we have $i = ((\pi_i - 1).p' \bmod m) + 1$ and $j = ((\pi_j - 1).p' \bmod m) + 1$, thus,

$$j - i = (\pi_j - \pi_i).p' \bmod m \tag{9.2}$$

– If $|\pi_i - \pi_j| = 1$ and both π_i and π_j belong to the same W_k^π for some $1 \le k \le m-p+1$, then we must have $j-i < p$. Moreover, $|\pi_i - \pi_j| = 1$ and (9.2) imply $j-i$ is either p' or $m-p'$. As stated earlier, $p \le p' < m/2$, so $m-p' > p' \ge p$, making $j - i < p$ impossible.

– Otherwise, suppose $\{\pi_i, \pi_j\} = \{1, m\}$ and both π_i and π_j belong to the same sliding window W_k^π for some $m - p + 1 < k \le m$. Notice that we must have $m - j + i < p$ for both π_i and π_j to be in W_k^π. Moreover, $\{\pi_i, \pi_j\} = \{1, m\}$ and (9.2) imply that $j - i$ is either p' or $m - p'$. So $m - j + i \in \{p', m - p'\}$, and similar to the previous case, this makes $m - j + i < p$ impossible

In sum, we have just shown that $c_i^\pi = 1, \forall 1 \le i \le m$, so $C^\pi = 1$. Notice that the choice of p' could have been any integer between p and $m/2$, as long as $\gcd(m, p') = 1$. In particular, $p' = m'$ clearly works. However, we have chosen p' to be the minimum of these because we would like the permuted sequence to be spreaded out in a "better manner". To illustrate this, consider the case where $m = 17$ and $p = 5$. Our choice of p' is 5 in this case, thus in two line notation we have

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ 1 & 8 & 15 & 5 & 12 & 2 & 9 & 16 & 6 & 13 & 3 & 10 & 17 & 7 & 14 & 4 & 11 \end{pmatrix}$$

While if we choose $p' = m' = 8$, the permutation becomes

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 \\ 1 & 16 & 14 & 12 & 10 & 8 & 6 & 4 & 2 & 17 & 15 & 13 & 11 & 9 & 7 & 5 & 3 \end{pmatrix}$$

Although this permutation does satisfy $C^{\pi} = 1$, it has the alternative numbers being too close to each other, thus we might end up in a situation where we lose every other frame for a while and then a chunk of successfully arrived frames. This clearly creates worse affect perceptually on the media stream comparing to the previous choice of p'.

- **Case 2 : m is even**

  If $\{j, p \leq j < \frac{m}{2} \wedge \gcd(m, j) = 1\} \neq \emptyset$ then we can just use exactly the same permutation as when m is odd.

  If the set is empty, let $m = 2p' \geq 2p$, so $p' \geq p$. As the previous case, we seek a permutation π so that $C^{\pi} = 1$. Let π be defined as follows.

$$\pi(i) = p'.(i \bmod 2) + \left\lceil \frac{i}{2} \right\rceil,\ 1 \leq i \leq m \qquad (9.3)$$

  If i is even, then $\pi(i) = \frac{i}{2}$, namely all the even numbers will be placed from the first position to the $\left(\frac{m}{2}\right)^{th}$ position in increasing order. When i is odd, $\pi(i) = p' + \frac{i+1}{2}$, so all the odd numbers will be placed from the $(\frac{m}{2}+1)^{st}$ position to the m^{th} position. It is clear from the above observation that π is a proper permutation on S.

We are left to prove that $C^\pi = 1$. Write $\pi = (\pi_1, \pi_2, \ldots \pi_m)$ in one line notation. Firstly, notice that for any $1 \leq i < j \leq m$, we have

$$\begin{aligned} i &= p'.(\pi_i \bmod 2) + \left\lceil \frac{\pi_i}{2} \right\rceil \\ j &= p'.(\pi_j \bmod 2) + \left\lceil \frac{\pi_j}{2} \right\rceil \end{aligned}$$

So,

$$j - i = p'.((\pi_j - \pi_i) \bmod 2) + \left\lceil \frac{\pi_j}{2} \right\rceil - \left\lceil \frac{\pi_i}{2} \right\rceil \tag{9.4}$$

Similar to case 1, if $C^\pi \geq 2$ then we consider two subcases :

– If $|\pi_i - \pi_j| = 1$ and both π_i and π_j belong to the same W_k^π for some $1 \leq k \leq m - p + 1$ then we must have $j - i < p$. Moreover, since $|\pi_i - \pi_j| = 1$ and $j > i$, it must be the case that π_j is odd and π_i is even. Combining with (9.4), we have

$$j - i = p' + \left\lceil \frac{\pi_j}{2} \right\rceil - \left\lceil \frac{\pi_i}{2} \right\rceil \geq p' \geq p$$

This makes $j - i < p$ impossible.

– Otherwise, if $\{\pi_i, \pi_j\} = \{1, m\}$ and both π_i and π_j belong to the same W_k^π for some $m - p + 1 < k \leq m$, then we must have $m - j + i < p$ for both π_i and

π_j to be in W_k^π. By (9.3), $\pi(1) = p' + 1$ and $\pi(m) = p'$, so it must be the case that $j = p' + 1$ and $i = p'$. Thus, $m - j + i = m - 1 \geq p' \geq p$, contradiction !

Example 9.3. Let $m = 16$ and $p = 8$, then

$$\pi = \begin{pmatrix} 2 & 4 & 6 & 8 & 10 & 12 & 14 & 16 & 1 & 3 & 5 & 7 & 9 & 11 & 13 & 15 \end{pmatrix}$$

□

Lemma 9.4. *If* $\frac{m}{2} < p < m$ *then* $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$

Proof. In order to prove that $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$, we will first prove that

$$k_0 \geq \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1 \tag{9.5}$$

then specify a permutation π so that $C^\pi = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$.

Let $\sigma = (\sigma_1, \sigma_2, \ldots \sigma_m)$ be any permutation on $S = \{1, 2, \ldots m\}$. Let $Y = y_1, y_2, \ldots y_{m-p}$ be the sequence obtained from sorting $\sigma_1, \sigma_2, \ldots \sigma_{m-p}$ in increasing order. Intuitively, Y is the complement of W^σ_{m-p+1} with respect to S.

Now, for convinience let $y_0 = 0$ and $y_{m-p+1} = m + 1$. Let $S_i = \{x : y_{i-1} < x < y_i, x \in S\}$, where $1 \leq i \leq m - p + 1$. S_i is simply the set of numbers between y_{i-1} and y_i in S.

$$y_0 = 0 \underbrace{1\ 2\ 3\ \ldots}_{S_1} y_1 \underbrace{\ldots}_{S_2} y_2 \cdots \underbrace{\ldots}_{S_i} y_i \ \cdots \underbrace{\ldots}_{S_{m-p}} y_{m-p} \underbrace{\ldots\ m}_{S_{m-p+1}} m + 1 = y_{m-p+1}$$

It is clear that

$$\sum_{i=1}^{m-p+1} |S_i| = p$$

Since $S_1 \uplus S_2 \uplus \ldots \uplus S_{m-p+1} = W^{\sigma}_{m-p+1}$, where $\uplus$ denotes disjoint union, it must be the case that $c^{\sigma}_{m-p+1} \geq |S_i|,\ \forall i \in \{1, 2 \ldots, m-p+1\}$. Thus, we have

$$p = \sum_{i=1}^{m-p+1} |S_i| \leq \sum_{i=1}^{m-p+1} c^{\sigma}_{m-p+1} = (m-p+1).c^{\sigma}_{m-p+1}$$

This implies

$$C^{\sigma} \geq c^{\sigma}_{m-p+1} \geq \left\lceil \frac{p}{m-p+1} \right\rceil$$

this inequality holds for all $\sigma \in S_m$'s, consequently

$$k_0 \geq \left\lceil \frac{p}{m-p+1} \right\rceil \tag{9.6}$$

Inequality (9.6) is not as tight as (9.5). To prove that (9.5) holds, we need to consider two cases.

Case (i) : $(m-p+1) \not| \, p$. In this case, $\lceil \frac{p}{m-p+1} \rceil = \lfloor \frac{p}{m-p+1} \rfloor + 1$, so (9.6) implies (9.5).

Case (ii) : $(m-p+1) \mid p$. In this case, let $y = \lceil \frac{p}{m-p+1} \rceil = \lfloor \frac{p}{m-p+1} \rfloor$.

Suppose $k_0 = y$ and let $\sigma = \sigma_1, \sigma_2, \ldots \sigma_m$ be a particular permutation on S so that $C^{\sigma} = y$. From the definition of C^{σ}, for any $1 \leq i \leq m-p+1$, we must have $c^{\sigma}_i \leq y$. Consider two special sliding windows $W^{\sigma}_1 = \{\sigma_1, \sigma_2, \ldots \sigma_p\}$ and $W^{\sigma}_{m-p+1} = \{\sigma_{m-p+1}, \sigma_{m-p+2}, \ldots \sigma_m\}$. Table 9.2 illustrates the situation.

We have $2p > m$, so $p > m-p$. Hence, there is no overlapping between $S - W^{\sigma}_1$ and $S - W^{\sigma}_{m-p+1}$. From the previous analysis, $k_0 = y$ holds if and only if $|S_i| = k_0, \forall i \in \{1, 2, \ldots, m-p+1\}$. So it must be the case that

$S - W^{\sigma}_{m-p+1}$	W^{σ}_{m-p+1}	
$\sigma_1, \sigma_2, \ldots \sigma_{m-p}$	$\sigma_{m-p+1}, \ldots \sigma_p$	$\sigma_{p+1}, \ldots \sigma_m$
W^{σ}_1		$S - W^{\sigma}_1$

Table 9.2: Pictorial illustration of $W^{\sigma}_1, W^{\sigma}_{m-p+1}, S - W^{\sigma}_1$ and $S - W^{\sigma}_{m-p+1}$

$$y_i = i.(k_0 + 1), \forall i = 1, 2, \ldots m - p \tag{9.7}$$

because if $\exists i, |S_i| < k_0$, then

$$\begin{aligned} m &= \sum_{i=1}^{m-p+1} |S_i| + (m-p) \\ &< (m-p+1).k_0 + (m-p) \\ &= (m-p+1).\frac{p}{m-p+1} + (m-p) \\ &= m, \text{ contradiction!} \end{aligned}$$

The previous analysis was done based on the window W^{σ}_{m-p+1} containing the last p elements of σ. So we have $\{y_1, y_2, \ldots ...y_{m-p}\} = S - W^{\sigma}_{m-p+1}$. Notice that exactly the same result holds if we do the analysis with respect to W^{σ}_1, in which case the y_i's are also determined by (9.7). So, we have the following :

$$\{1(k_0+1), 2(k_0+1), \ldots, (m-p)(k_0+1)\} = S - W^{\sigma}_{m-p+1} = S - W^{\sigma}_1$$

This is impossible since $(S - W^{\sigma}_1)$ and $(S - W^{\sigma}_{m-p+1})$ are disjoint. Consequently $k_0 \neq y$, so $k \geq y + 1 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$.

Finally, to show that $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$, it is sufficient to specify a permutation π on S such that $C^\pi = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$.

Before specifying π, we need to observe some facts. Let $q = m - p$, $r = \left\lfloor \frac{p}{q+1} \right\rfloor$, and $t = \left\lfloor \frac{m}{r+2} \right\rfloor$, then $q \geq 1$ because $q = m - p > 0$. Moreover, since $p > m - p = q$, we have $p \geq q + 1$. Thus $r \geq 1$ and we can write p as :

$$p = \quad (q+1)r + r', \quad 0 \leq r' \leq q$$

In addition, $m = p + q = (q+1)r + r' + q \geq qr + q + r \geq r + 2$, so $t \geq 1$ and we can write m as :

$$m = \quad (r+2)t + t', \quad 0 \leq t' \leq r + 1$$

Now, we specify π in one line notation by considering 2 cases as follows.

Case (i) : $t' = r + 1$.

We have

$$\begin{aligned} m &= p + q \\ \Rightarrow \quad (r+2).t + (r+1) &= (q+1).r + r' + q \\ &= (q-1).(r+2) + r + r' - q + r + 2 \\ \Rightarrow \quad (r+2).t + (r+1) &= (q-1).(r+2) + (r' - q) + 2.r + 2 \\ \Rightarrow \quad (r+2).t &= (q-1).(r+2) + (r' - q) + r + 1 \end{aligned}$$

Moreover, since $r' \leq q$, we get

$$\begin{aligned} (r+2).t &\leq (q-1).(r+2)+r+1 \\ \Rightarrow \qquad t &\leq (q-1)+\tfrac{r+1}{r+2} \end{aligned}$$

t and q are natural numbers, so

$$t \leq q-1 \tag{9.8}$$

We are ready to describe our π now. Notice that we would like C^π to be $r+1$, making $k_0 = r+1$ as desired. First we define 2 special arrays $A = \{a_i\}_1^{t+1}$ and $B = \{b_i\}_1^{t+1}$ as follows.

$$\begin{aligned} a_i &= 1+(i-1).(r+2), & 1 \leq i \leq t+1 \\ b_i &= (r+1)+(i-1).(r+2), & 1 \leq i \leq t+1 \end{aligned}$$

It is obvious that $1 \leq a_i \leq m$ and $1 \leq b_i \leq m$, thus A and B are subsequences of S. Let $C = \{c_i\}_1^{m-2.(t+1)}$ be the increasing sequence of numbers in S but not in A and B. Notice that if we write numbers $1, 2, \ldots m$ in increasing order, we have the following picture :

$$\begin{array}{ccccccccccccc} & & b_1 & & & & b_2 & & & & & & b_{t+1} \\ 1 & .. & r+1 & .. & r+3 & .. & 2r+3 & .. & 2r+5 & .... & 1+t(r+2) & .. & m \\ a_1 & & & & a_2 & & & & a_3 & & a_{t+1} & & \end{array}$$

and the rest of the numbers (the dots in the picture) contribute sequence C. Finally, here is our π in one line notation :

$$\pi = (\underbrace{a_{t+1}, a_t, \dots a_1}_{A}, \underbrace{c_{m-2.(t+1)}, \dots\dots\dots\dots, c_1}_{C=S-A-B} \underbrace{b_{t+1}, b_t, \dots b_1}_{B})$$

To prove that this is a valid permutation, it suffices to observe that the sequences $\{a_i\}$ and $\{b_i\}$ don't intersect, since $a_i \equiv 1 \pmod{r+2}$, while $b_j \equiv r+1 \pmod{r+2}$.

We are left to prove that for every $1 \leq k \leq m$, W^π doesn't contain more than $r+1$ *consecutive* integers, where as usual, when $m-p+1 < k \leq m$ we interprete *consecutiveness* a little differently.

Notice that $\forall i \in \{1,\dots,t\}$, the numbers of integers between a_i and a_{i+1} exclusively is $r+1$, which we shall call the *internal distance* between a_i and a_{i+1}. Similarly, the internal distance between b_i and b_{i+1} is also $r+1$. Now, consider the first sliding window W_1^π of size p of π :

$$\pi = \overbrace{\underbrace{a_{t+1}, a_t, \dots a_1, c_{m-2.(t+1)}, \dots\dots,}_{W_1^\pi \longrightarrow} \dots, c_1}^{m-(t+1)\ numbers}, b_{t+1}, b_t, \dots b_1$$

This window can not contain any b_i's, since from (9.8) we have $|W_1^\pi| = p = m-q \leq m-(t+1)$. As we have noticed earlier, the internal distance between b_i and b_{i+1} is $r+1$ for all $i \in \{1,2,\dots t\}$, thus it is easy to see that $c_1^\pi \leq r+1$. We will prove that this property holds for all sliding windows W_k^π by induction on k, where $1 \leq k \leq m-p+1$, namely $\forall k \in \{1,2,\dots m-p+1\}, c_k^\pi \leq r+1$.

- Base case : $c_1^\pi \leq r+1$ as discussed.
- Suppose we have $c_k^\pi \leq r+1$ for some $1 \leq k < m-p+1$.

- Let's look at W^{π}_{k+1}. If W^{π}_{k+1} does not contain any element of B then $c^{\pi}_{k+1} \leq r+1$ holds trivially. Otherwise, suppose W^{π}_{k+1} does contain some set of more than $r+1$ consecutive integers. Let i be the least integer such that $b_i \in W^{\pi}_{k+1}$. It is clear that W^{π}_{k} doesn't contains b_i, because intuitively we have just moved W^{π}_{k} one step to the right of π and added b_i into W^{π}_{k} to obtain W^{π}_{k+1}. The only way for W^{π}_{k+1} to contain a set of more than $r+1$ consecutive integers is when this set contains b_i, since by induction hypothesis, just before this point, $c^{\pi}_{k} \leq r+1$. Additionally, since the size of W^{π}_{k+1} is p, it's easy to see that if $b_i \in W^{\pi}_{k+1}$ then $a_j \notin W^{\pi}_{k+1}, \forall j \geq i$.

  However, if b_i is contained in some set X of more than $r+1$ consecutive integers, then X has to either contain a_i or a_{i+1} because $a_i < b_i < a_{i+1}$ and the internal distance between a_i and a_{i+1} is $r+1$. Contradiction !

  So W^{π}_{k+1} does not contain any set of more than $r+1$ consecutive integers. In other words, $c^{\pi}_{k+1} \leq r+1$.

We also have to show that

$$c^{\pi}_{k} \leq r+1, \text{ when } m-p+1 < k \leq m \tag{9.9}$$

- If $r = 1$, then $m = 3t+2$; thus, $2(t+1) = m-t \geq p+1$. Consequently, none of $W^{\pi}_{k}, m-p+1 < k \leq m$ contains both $a_1 = 1$ and $b_{t+1} = m$. (9.9) follows trivially.

- If $r > 1$, then $c_1 = 2$ and $c_{m-2(t+1)} = m-1$. Moreover, clearly $c_{m-2(t+1)} \notin \{\pi_k, \ldots \pi_m\}$ and $c_1 \notin \{\pi_1, \ldots \pi_{p+k-m-1}\}$ whenever $m-p+1 < k \leq m$. Thus, for these values of k, c_k^π is at most 2, which is at most $r+1$, which completes our proof.

Example 9.5. Let $m = 17$, $p = 9$. Thus, $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1 = \left\lfloor \frac{9}{17-9+1} \right\rfloor + 1 = 2$. $r = 1$ and $17 = 3.5 + 2$, so $t' = 2 = r + 1$ in this case. The sequences A and B are as follows. $A = 1, 4, 7, 10, 13, 16$. $B = 2, 5, 8, 11, 14, 17$. And lastly, applying our scheme gives us permutation

$$\pi = \begin{pmatrix} 16 & 13 & 10 & 7 & 4 & 1 & 15 & 12 & 9 & 6 & 3 & 17 & 14 & 11 & 8 & 5 & 2 \end{pmatrix}$$

Case (ii) : $0 \le t' < r + 1$.

We have

$$\begin{aligned} m &= p + q \\ \Rightarrow \quad (r+2).t + t' &= (q+1).r + r' + q \\ &= q.(r+2) + r + r' - q \\ \Rightarrow \quad (r+2).t &= q.(r+2) + (r' - q) + r - t' \end{aligned}$$

Moreover, since $r' \le q$, we get

$$\begin{aligned} (r+2).t &\le q(r+2) + r - t' \\ \Rightarrow \quad t &\le q + \tfrac{r-t'}{r+2} \end{aligned}$$

t and q are natural numbers, so

$$t \le q \tag{9.10}$$

Similar to the case (i), we define 2 special arrays $A = \{a_i\}_1^t$ and $B = \{b_i\}_1^t$ as follows.

$$\begin{aligned} a_i &= t' + 1 + (i-1).(r+2) \quad 1 \le i \le t \\ b_i &= i.(r+2) \quad\quad\quad\quad\quad\quad 1 \le i \le t \end{aligned}$$

It is obvious that $1 \le a_i \le m$ and $1 \le b_i \le m$, thus A and B are subsequences of S. $A \cap B = \emptyset$ since $(r+2)$ divides a_i and does not divide b_j for all i and j. Let $C = \{c_i\}_1^{m-2t}$ be the increasing sequence of numbers in S but not in A and B. If we write numbers $1, 2, \ldots m$ in increasing order, this time we have the following picture :

$$\begin{array}{ccccccccccccc} & & & & b_1 & & & & & & b_t & & \\ 1 & .. & t'+1 & .. & r+2 & .... & t'+1+(t-1)(r+2) & .. & t(r+2) & .. & m \\ & & a_1 & & & & a_t & & & & \end{array}$$

and the rest of the numbers (the dots in the picture) contribute sequence C. π in one line notation is as follows.

$$\pi = (\underbrace{a_t, a_{t-1}, \ldots a_1}_{A}, \underbrace{c_{m-2t}, \ldots\ldots\ldots\ldots, c_1}_{C=S-A-B} \underbrace{b_t, b_{t-1}, \ldots b_1}_{B})$$

The proof goes almost the same as in case (i). Due to $t \le q$, W_1^π does not contain any of the b_i's. Straightforward induction shows that when $1 \le k \le m - p + 1$, none of W_k^π contain more than $r+1$ consecutive integers. On the other hand, when $m - p + 1 < k \le m$, we always have

$$c_1 \notin \{\pi_1, \ldots, \pi_{p+k-m-1}\} \text{ and } c_{m-2t} \notin \{\pi_k, \ldots, \pi_m\}$$

If $t' = 0$, then $c_1 = 2$ and $c_{m-2t} = m - 1$, so $c_k^\pi \leq 2 \leq r + 1$. If $t' > 0$, then $c_1 = 1$ and $c_{m-2t} = m$, so $c_k^\pi = 0$.

For example, let $m = 17$, $p = 12$. $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1 = \left\lfloor \frac{12}{17-12+1} \right\rfloor + 1 = 3$. $r = 2$ and $17 = 4.4 + 1$, so $t' = 1 < r + 1$ in this case. The sequences A and B are as follows. $A = 4, 8, 12, 16$. $B = 2, 6, 10, 14$. Applying our scheme gives us permutation

$$\pi = \begin{pmatrix} 16 & 12 & 8 & 4 & 17 & 15 & 13 & 11 & 9 & 7 & 5 & 3 & 1 & 14 & 10 & 6 & 2 \end{pmatrix}$$

□

9.3.2 Summary and Benefits of the Bounded Error Case

The following theorem summarizes our work on the deterministic cases.

Theorem 9.6. *If p and m are both determined, then*

- $k_0 = 0$ *when* $p = 0$ *and* $k_0 = m$ *when* $p \geq m$
- $k_0 = \left\lfloor \frac{p}{m-p+1} \right\rfloor + 1$ *when* $0 < p < m$.

Proof. Since if $0 < p \leq \frac{m}{2}$ then $\left\lfloor \frac{p}{m-p+1} \right\rfloor = 0$, this is immediate from the preceding lemmas and remark. Also note that if the desired k_0 is given, these formulas allow us to find the minimum buffer size m_0 to achieve k_0. □

Algorithm $calculatePermutation(m, p)$ is a permutation generator which generates permutation π with $C^{\pi} = k_0$ on input m and p. Notice that it takes only linear time.

calculatePermutation(m, p)

if $p \leq 0$ or $p \geq m$ **then**

output the identity permutation

end if

if $p \leq \frac{m}{2}$ **then**

$M \leftarrow \{j,\, p \leq j \leq \frac{m}{2} \wedge \gcd(m, j) = 1\}$

if $M \neq \emptyset$ **then**

$p' \leftarrow \min\{j,\ j \in M\}$

for $i \leftarrow 1$ to m **do**

$\pi(i) \leftarrow ((i-1)p' \bmod m) + 1$

end for

else

** $M = \emptyset$, m must be even **

$p' = \frac{m}{2}$

for $i \leftarrow 1$ to m **do**

$\pi(i) \leftarrow p'.(i \bmod 2) + \lceil \frac{i}{2} \rceil$

end for

end if

else

$q \leftarrow m - p$

$r \leftarrow \left\lfloor \frac{p}{q+1} \right\rfloor$

$t \leftarrow \lfloor \frac{m}{r+2} \rfloor$

$t' \leftarrow m \bmod (r+2)$

if $t' = r + 1$ **then**

for $i \leftarrow 1$ to $t + 1$ **do**

$a_i \leftarrow 1 + (i-1).(r+2)$

$b_i \leftarrow (r+1) + (i-1).(r+2)$

end for

$C \leftarrow \{1, 2, \ldots m\} - \{a_i\} - \{b_i\}$

C is extracted in increasing order.

for $i \leftarrow 1$ to $t + 1$ **do**

$\pi(a_{t+2-i}) \leftarrow i$

end for

for $i \leftarrow t + 2$ to $m - (t+1)$ **do**

$\pi(c_{m-i-t}) \leftarrow i$

end for

for $i \leftarrow m - t$ to m **do**

$\pi(b_{m-i+1}) \leftarrow i$

end for

else

** i.e. $0 \leq t' \leq r$ **

for $i \leftarrow 1$ to t **do**

$a_i \leftarrow t' + 1 + (i-1).(r+2)$

$b_i \leftarrow i.(r+2)$

end for

$C \leftarrow \{1, 2, \ldots m\} - \{a_i\} - \{b_i\}$

C is extracted in increasing order.

for $i \leftarrow 1$ to t **do**

  $\pi(a_{t+1-i}) \leftarrow i$

end for

for $i \leftarrow t+1$ to $m-t$ **do**

  $\pi(c_{m-i-t+1}) \leftarrow i$

end for

for $i \leftarrow m-t+1$ to m **do**

  $\pi(b_{m-i+1}) \leftarrow i$

end for

end if

end if

Benefits of solving the bounded error case

The assumption that p is known can be envisioned in future networks where some sort of QoS guarantees are provided, such as ATM, Internet2, etc. . More importantly, it gives us a rigid background to solve the unbounded error case.

9.4 Unbounded Network Error Case

9.4.1 Feedback based permutation adjustment protocol

Our protocol is a simple feedback based protocol. Some CM systems use TCP/IP for communication [78]. But it has been shown in [146] that CM applications based on TCP are unstable when the real time bandwidth requirements fall below available bandwidth. Thus in this protocol, we use the UDP communication model (like [144,145]). We dynamically use the solution provided in the deterministic cases as a mechanism for the non-deterministic scenario presented here. We assume that m, the buffer size, is known in advance by both client and server. This can also be part of a initial negotiation.

At the server side, a buffer of size m is kept. Server permutes frames (actually frame indices) based on current set of parameters, then initiates transmission of the frames in the buffer. Server changes the permutation scheme based on client's feedback. The permutation scheme changes only at the start of the next buffer of frames.

At the client side, the client waits for a period of $m/frameRate$ (time needed for the client's buffer to be filled up) and calculates consecutive network loss for this buffer window. The client keeps track of the previous window's estimated network consecutive loss and sends its next estimation back to the server. It sends feedback (ACK) in a UDP packet. Note that ACK packet is also given a sequence number so that out of order ACK packets will be ignored. The server makes decision based on the maximum sequence numbered ACK.

Given a buffer of size m, initially the server assumes the average case where $p = \left\lfloor \frac{m}{2} \right\rfloor$. Denote p_i as the actual consecutive network loss, and p_i^* as the estimated network loss in the i^{th} window. We use exponential averaging to estimate next loss. Suppose we are currently at the n^{th} window, p_n^* is determined by

$$p_n^* = \left\lceil \alpha . p_n + (1 - \alpha) . p_{n-1}^* \right\rceil$$

In this experiment, we have picked $\alpha = \frac{1}{2}$. This value turns out to work just fine, as shown in section 9.5. Whether or not there exists an optimal value for α is subject to further investigation. Basically, α measures how much weight we would like to give to the current network status. The larger α is, the less weight we give to the history of network behavior. p_n^* is rounded up because we want to assume the worse error.

9.4.2 Illustration of the protocol

Figure 9.3 illustrate an example of how client and server interact. $< j, \pi_i >$ is the time where server sends the i^{th} frame of the j^{th} buffer window. ACK_j containing the estimated

p_j^* sent back by the client. By the time server gets ACK_j, it could be in the $(k-1)^{th}$ buffer window. So, it uses ACK_j for the k^{th} buffer window. Lastly, ACK_{j-1} is lost, so we have not used it for transmission of any of the buffer window subsequently.

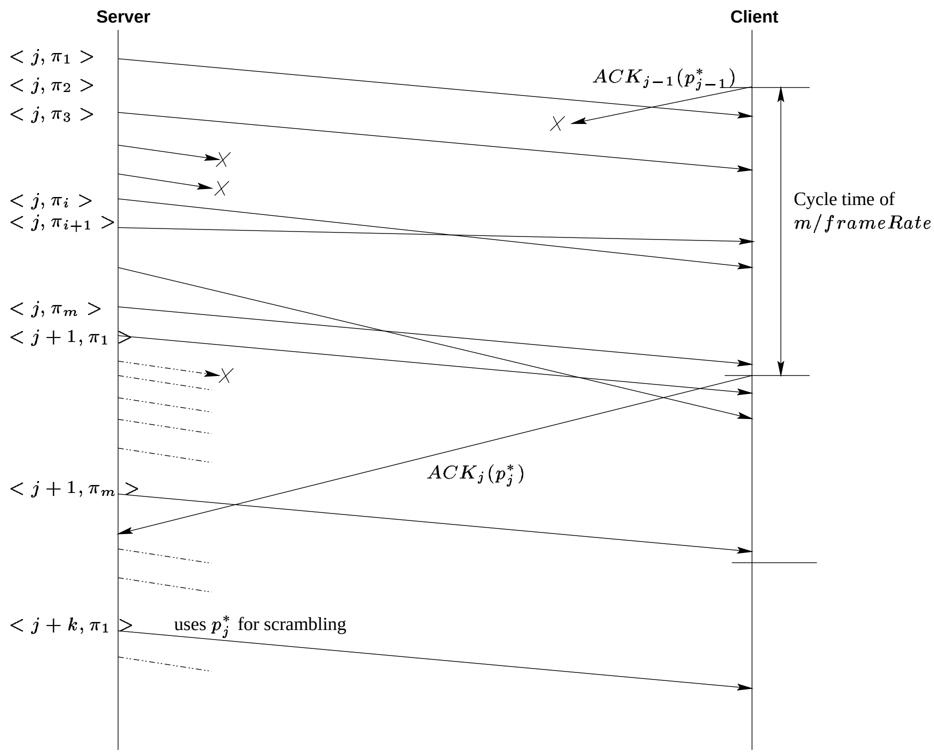


Figure 9.3: A sample session

9.5 Experimental Evaluation

The following two sections presents the evaluation of our scheme in two scenarios. In one case the protocol described in section 9.4.1 has been implemented and tested over a long haul network. In the second case, we use a data set extracted from a real-time application such as *Internet Phone* and simulate our protocol. We show the reduction in CLF in both the cases. Our protocol has smoothed out CLF to be within the range of perceptually acceptable tolerance. Also, it adapts quite well with abnormality in network loss pattern. Moreover, *almost all* of CLF values are within the range of perceptual tolerance (see section 9.2.1. Thus this approach of of using End User QoS as a direct means to control Media Delivery shows a lot of promise. There are a number of extensions to the protocol which we have been and are currently looking into. These are briefly discussed in section 9.6;

9.5.1 Video Experiment : Actual Media Delivery over a Long Haul Network

We have conducted experiments of sending two MJPEG video clips over LAN and WAN. Due to limited space, only the result of WAN is shown here. However, the behavior of our protocol is the same in both cases. We transfered data from a UltraSparc 1 (`rawana.cs.umn.edu`) in Computer Science department, University of Minnesota to another SunSparc (lombok.cs.uwm.edu) in Computer Science department, University of Wisconsin, Milwaukee [2]. The experiment was conducted at 9:45am when network traffic is expected to be average. Both clips have resolution 512×384. Clip 1 frame sizes varies from 5276 to 36364 bytes with 9544 bytes as the median, 10845 bytes as the mean and 4450.7 is the standard variation. These numbers for clip 2 respectively are 5072, 34408, 10282, 10916 and 3642.8. Clip 1 contains 2607 frames and clip 2 contains 1736 frames. Our buffer window is of size 50. Three times "traceroute" [3] told us that the packets typically go through 14 hops in between. A sample traceroute session is as follows.

```
 1  eecscix.router.umn.edu (160.94.148.254)  2 ms  1 ms  1 ms
 2  tc8x.router.umn.edu (128.101.192.254)  23 ms  4 ms  3 ms
 3  tc0x.router.umn.edu (128.101.120.254)  6 ms  1 ms  1 ms
 4  t3-gw.mixnet.net (198.174.96.5)  1 ms  1 ms  1 ms
```

[2] Thanks to Mr. Thanh C. Nguyen at the Department of Computer Science, University of Wisconsin, Milwaukee for helping us in conducting this experiment

[3] Thanks to Mr. Luan V. Nguyen by the time was a system staff at the Department of Computer Science, University of Minnesota, for providing us with this traceroute result

```
 5  border5-hssi1-0.Chicago.cw.net (204.70.186.5)  11 ms  11 ms  29 ms
 6  core2-fddi-0.Chicago.cw.net (204.70.185.49)  11 ms  11 ms  11 ms
 7  core2-hssi-3.WillowSprings.cw.net (204.70.1.225)  13 ms  13 ms  15 ms
 8  core3.WillowSprings.cw.net (204.70.4.25)  310 ms  52 ms  123 ms
 9  * ameritech-nap.WillowSprings.cw.net (204.70.1.198)  245 ms  35 ms
10  aads.nap.net (198.32.130.39)  18 ms  18 ms  21 ms
11  r-milwaukee-hub-a9-0-21.wiscnet.net (207.112.247.5)  25 ms  22 ms  27 ms
12  205.213.126.39 (205.213.126.39)  19 ms  20 ms  23 ms
13  miller.cs.uwm.edu (129.89.139.22)  24 ms  25 ms  21 ms
14  lombok.cs.uwm.edu (129.89.142.52)  24 ms *  25 ms
```

Figure 9.4 shows the result. As can be seen from the figure, our scheme has done quite well smoothing network consecutive losses. In a few cases our CLF is 1 higher (clip 2) but that was due to rapid changes in network loss behavior and it is expected. Most of the time CLF is well below and also within tolerable perceptual limits (see section 9.2.1).

9.5.2 Simulation: Using data from a real time application like Internet Phone

The data [4] was collected for an *Internet Voice or Voice on Networks (VON)* application. The server is *vermouth.ee.umanitoba.ca (Canada)* and the Client is *rawana.cs.umn.edu (Minnesota, USA)*. *vermouth* and *rawana* is a SUN UltraSparc 1, running Solaris V2.6 and V2.5 respectively. Each host is on a 10 Mbps Ethernet (LAN). The transmission is over the Internet and the data set was collected on a Saturday, from 10 am to 2 pm. The two files presented here are of voice packets of sizes 160 and 480 bytes. As can be seen from figure 9.5 the actual CLF's of network losses are varying while the CLF based on our protocol always has lower CLF (in this case CLF=1, implying no consecutive losses).

[4] Thanks to Mr. Difu Su, Computer Science Department, University of Minnesota, for providing us with the data set

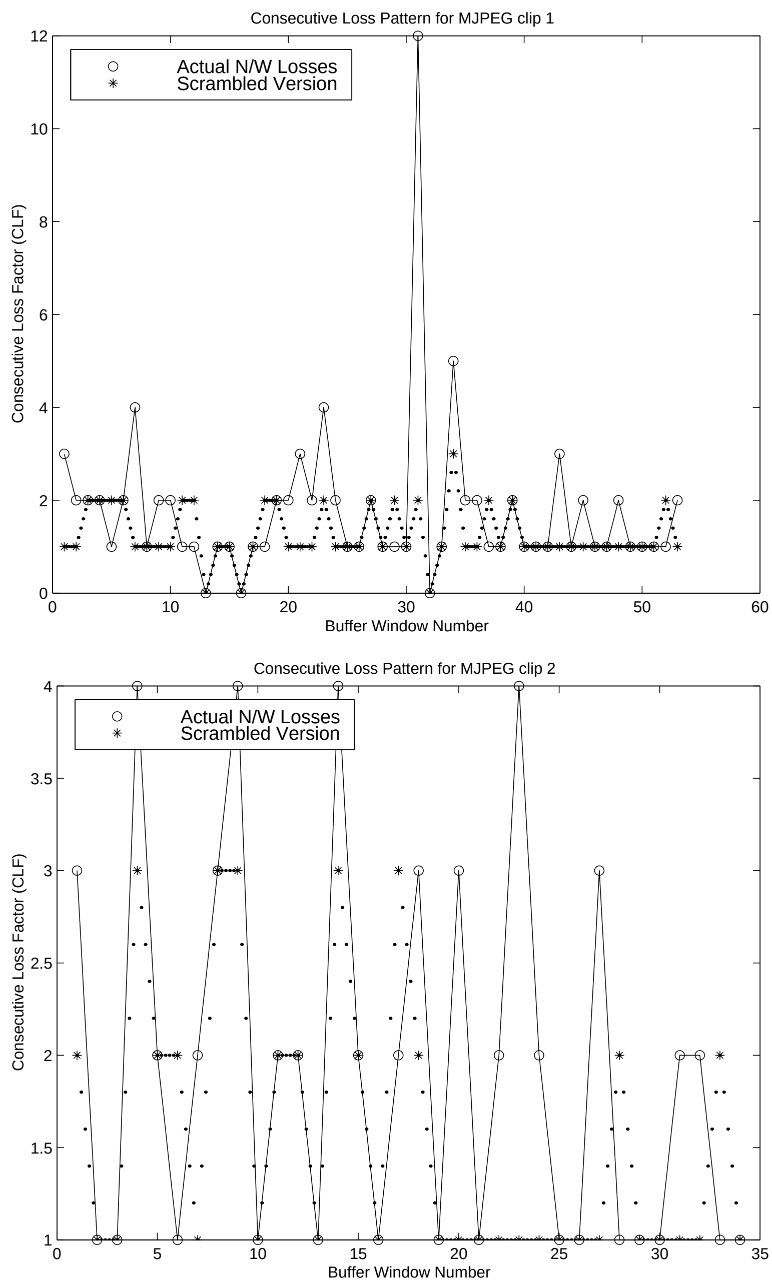


Figure 9.4: Performance of our protocol when transmitting video over long haul network

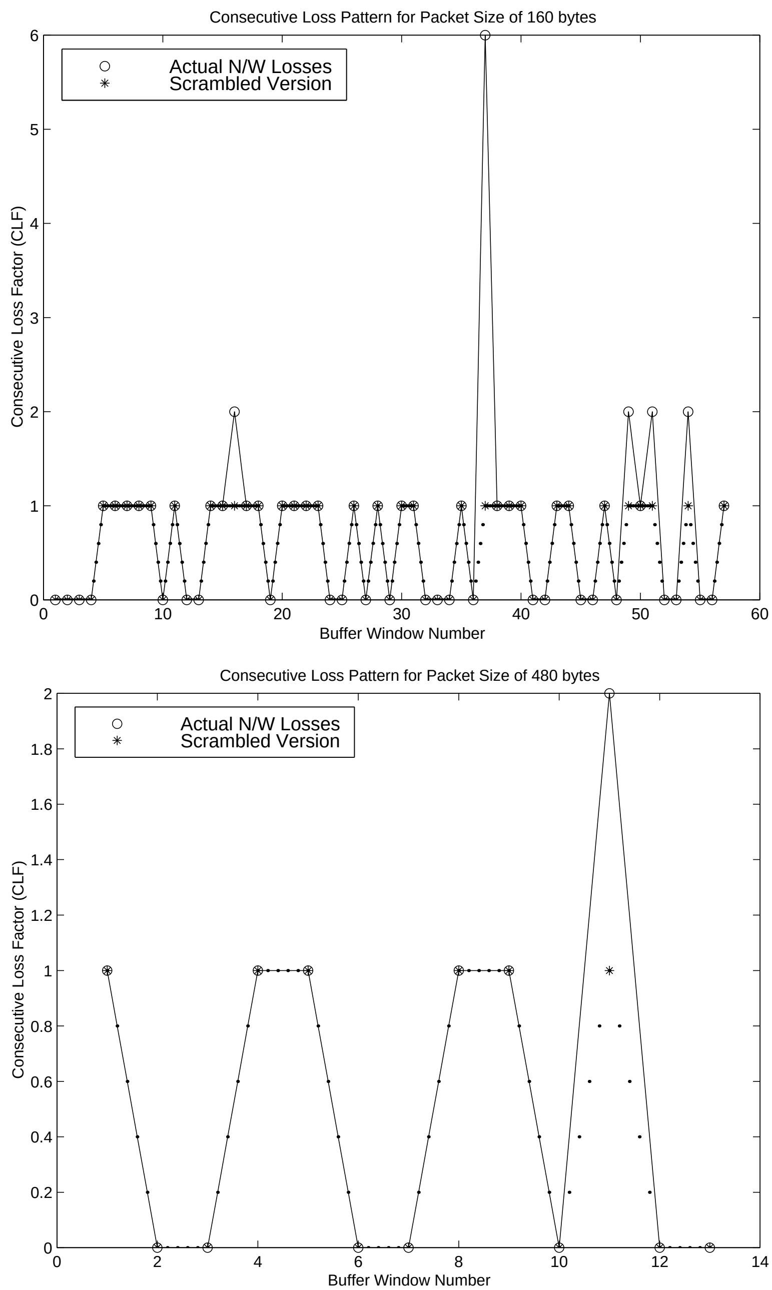


Figure 9.5: Performance of our protocol for a real-time application such as Internet Phone

9.6 Discussions

In this chapter we have addressed the problem of handling bursty losses in continuous media transmission. We formulated the problem in terms of a number of parameters including user QoS requirements, sender resource availability, and network loss behavior. We introduced the idea of *error spreading*, which takes spots of concentrated bursty losses and spreads it evenly over the entire stream. This makes the stream more acceptable from a perceptual viewpoint [163]. Our experiments over the Internet show that the scheme is quite effective.

Our ongoing work is addressing a number of issues. First, we want to develop an analytical formulation for the unbounded network error case. Second, we want to develop metrics which help us to choose which permutation to be used when the first level parameter (k_0) is not enough to break the tie. Finally, we want to extend this idea to groups of synchronized streams.

An extension of this work has already been done and is presented in [154]. In the paper, we extended this idea to hanlde streams with inter-frame dependency such as MPEG and shows that error spreading could be used in conjunction with other existing protocols without changing the underlining protocol.

From a Combinatorial point of view, it would be interesting to answer the following question : *given* m *and* p*, how many* $\pi \in S_m$ *are there such that* $C^{\pi} = k_0$ *?* The answer to this is probably not so difficult to find, but it will be tedious with various boundary cases involved.

Chapter 10

Conclusions

Interconnection networks play a crucial role in various branches of Computer Science. This thesis has addressed many interesting and difficult problems of interconnection network from the topological layer. Progresses were made on several outstanding conjectures. All four major classes of interconnection networks were included. Just like it has been for the last 40 years, the problems in this dissertation require a good deal of mathematical tools, ranging from Combinatorics, Linear Algebra, Graph Theory, Coding Theory, to even Number Theory ...

Admittedly, some of the real-world applications are still looking somewhat superficial. However, this is due to the fact that this field is enormous and is still in its early stage of development. In fact, no good comprehensive book on the theory of interconnection network topology exists; although from an engineering point of view there are several good ones.

All the problems and results in the thesis are combinatorial in nature. This fact fits well to the overall theme of applying well established discrete mathematics techniques to solve a class of practical problems. What missing are the probabilistic models on interconnection networks, which form the second major class of problem solving techniques for interconnection topology problems. One of the reasons for not having any probabilistic arguments on our results is that all of our problems require some very precise estimates on the objective functions. While, probabilistic arguments often result in values with hidden constants. That is not to assert that probabilistic methods are out of question, however. We might just not know yet how to apply them in the right way.

From all the results presented, there are wide open problems and directions for further research, as discussed at the end of each chapter. One major direction is about the multi-rate complexity of switching networks. The complexity of classical (i.e. single rate) switching networks has been a major area of research during the last 40 years. These studies have enriched many areas of Mathematics and Theoretical Computer Science with fascinating problems and new techniques. Hence, we expect that the natural extension from single-rate to multi-rate could provide similar effects. The problems seem to be more difficult with the extension, one of whose examples we have seen in Chapter 4.

BIBLIOGRAPHY

[1] M. AIGNER, *Searching with lies*, J. Combin. Theory Ser. A, 74 (1996), pp. 43–56.

[2] M. AJTAI, J. KOMLÓS, AND E. SZEMERÉDI, *Sorting in $c \log n$ parallel steps*, Combinatorica, 3 (1983), pp. 1–19.

[3] A. ALBANESE, J. BLOMER, J. E. M. LUBY, AND M. SUDAN, *Priority Encoding Transmission*, Proceeding of 35th Annual Symposium on Foundations of Computer Sciences, Santa Fe, New Mexico, (1994).

[4] N. ALON, *Eigenvalues and expanders*, Combinatorica, 6 (1986), pp. 83–96. Theory of computing (Singer Island, Fla., 1984).

[5] ——, *Eigenvalues, geometric expanders, sorting in rounds, and Ramsey theory*, Combinatorica, 6 (1986), pp. 207–219.

[6] N. ALON AND M. TARSI, *Colorings and orientations of graphs*, Combinatorica, 12 (1992), pp. 125–134.

[7] K. A. ALY, *Reconfigurable WDM Interconnection of Large-scale Multiprocessor System*, PhD thesis, Department of Electrical and Computer Engineering, SUNY at Buffalo, Buffalo, NY, 1993.

[8] ——, *Conflict-free channel assignment for an optical cluster-based shuffle network configuration*, in Proceedings, 1994 SIGCOMM Conference, London, UK, Aug. 31st - Sept. 2nd 1994, pp. 201–210.

[9] K. A. ALY AND P. W. DOWD, *A class of scalable optical interconnection networks through discrete broadcast-select multi-domain WDM*, in Proceedings of the 13th Annual Joint Conference of the IEEE Computer and Communications Societies on Networking for Global Communications. Volume 1, Los Alamitos, CA, USA, June 1994, IEEE Computer Society Press, pp. 392–399.

[10] R. APTEKAR, J. FISHER, V. KISIMOV, AND H. NIESHLOS, *Distributed Multimedia:User Perception and Dynamic QoS*, in SPIE , vol. 2188, SPIE, 1994.

[11] M. ARTIN, *Algebra*, Prentice-Hall Inc., Englewood Cliffs, NJ, 1991.

[12] D. J. Balding, W. J. Bruno, E. Knill, and D. C. Torney, *A comparative survey of non-adaptive pooling designs*, in Genetic mapping and DNA sequencing (Minneapolis, MN, 1994), Springer, New York, 1996, pp. 133–154.

[13] D. J. Balding and D. C. Torney, *Optimal pooling designs with error detection*, J. Combin. Theory Ser. A, 74 (1996), pp. 131–140.

[14] V. E. Beneš, *Mathematical theory of connecting networks and telephone traffic*, Academic Press, New York, 1965. Mathematics in Science and Engineering, Vol. 17.

[15] ——, *Proving the rearrangeability of connecting networks by group calculations*, Bell System Tech. J., 54 (1975), pp. 421–434.

[16] T. Berger, N. Mehravari, D. Towsley, and J. Wolf, *Random multiple-access communications and group testing*, IEEE Trans. Commun., 32 (1984), pp. 769–779.

[17] J. Bermond and C. Peyrat, *de bruijn and kautz networks: a competitor for the hypercube?*, in Proc. of the First European Workshop on Hypercube and Distributed Computers, Rennes, Elsevier (North-Holland), Oct 1989, pp. 279–293.

[18] J.-C. Bermond, ed., *Interconnection networks*, North-Holland Publishing Co., Amsterdam, 1992. Reprint of Discrete Appl. Math. **37/38** (1992).

[19] J. C. Bermond, F. Comellas, and D. F. Hsu, *Distributed loop computer networks: A survey*, Journal of Parallel and Distributed Computing, 24 (1995), pp. 2–10.

[20] J.-C. Bermond, N. Homobono, and C. Peyrat, *Large fault-tolerant interconnection networks*, Graphs Combin., 5 (1989), pp. 107–123.

[21] ——, *Connectivity of Kautz networks*, Discrete Math., 114 (1993), pp. 51–62. Combinatorics and algorithms (Jerusalem, 1988).

[22] J.-C. Bermond and C. Peyrat, *Broadcasting in de Bruijn networks*, Congr. Numer., 66 (1988), pp. 283–292. Nineteenth Southeastern Conference on Combinatorics, Graph Theory, and Computing (Baton Rouge, LA, 1988).

[23] T. Beth, D. Jungnickel, and H. Lenz, *Design theory*, Cambridge University Press, Cambridge, 1986.

[24] B. Bollobás, *Extremal graph theory*, Academic Press Inc. [Harcourt Brace Jovanovich Publishers], London, 1978.

[25] J. C. BOLOT, *End-to-End Packet Delay and Loss Behaviour in the Internet*, ACM SIGCOMM '93, Ithaca, NY, (1993).

[26] J. C. BOLOT AND A. V. GARCIA, *The case for FEC-based error control for packet audio in the Internet*, To appear in ACM Multimedia Systems.

[27] J. C. BOLOT AND TURLETTI, *Adaptive Error Control for Packet Video in the Internet*, Proceedings IEEE International Conference on Image Processing, ICIP'96, Laussane, Switzerland, (1996).

[28] J. C. BOLOT AND T. TURLETTI, *Experience with Control Mechanisms for Packet Video*, ACM Communication Review, Jan '98, 28 (1998).

[29] J. C. BOLOT, T. TURLETTI, AND I. WAKEMAN, *Scalable Feedback Control for Multicast Video Distribution in the Internet*, ACM SIGCOMM'94, London, UK, (1994), pp. 58–67.

[30] J. BOND AND A. IVANYI, *Modeling of interconnection networks using de bruijn graphs*, in Proc. Third Conference of Program Designers, Budapest, A. Vitanyi, ed., 1987, pp. 75–88.

[31] J. BRUCK, R. CYPHER, AND C.-T. HO, *Fault-tolerant de Bruijn and shuffle-exchange networks*, IEEE Transactions on Parallel and Distributed Systems, 5 (1994), pp. 548–553.

[32] W. J. BRUNO, E. KNILL, D. J. BALDING, D. C. BRUCE, N. A. DOGGETT, W. W. SAWHILL, R. L. STALLINGS, C. C. WHITTAKER, AND D. C. TORNEY, *Efficient pooling designs for library screening*, Genomics, (1995), pp. 21–30.

[33] W. E. BURR, *The FDDI optical data link*, IEEE commun. mag., 24, 5 (1986), pp. 18–23.

[34] A. R. CALDERBANK, A. R. HAMMONS, JR., P. V. KUMAR, N. J. A. SLOANE, AND P. SOLÉ, *A linear construction for certain Kerdock and Preparata codes*, Bull. Amer. Math. Soc. (N.S.), 29 (1993), pp. 218–222.

[35] J. CAPETANAKIS, *The Multiple Access Broadcast Channel: Protocol and Capacity Considerations*, PhD thesis, Massachusetts Institute of Technology, 1977.

[36] ——, *Generalized tdma: The multi-accessing tree protocol*, IEEE Trans. Commun., 27 (1979), pp. 1476–1484.

[37] ——, *Tree algorithm for packet broadcasting channel*, IEEE Trans. Inform. Theory, IT-25 (1979), pp. 505–515.

[38] J. D. CARPINELLI AND A. Y. ORUÇ, *Applications of matching and edge-coloring algorithms to routing in Clos networks*, Networks, 24 (1994), pp. 319–326.

[39] C. C. CHAO AND W.-Q. LIANG, *Arranging n distinct numbers on a line or a circle to reach extreme total variations*, European J. Combin., 13 (1992), pp. 325–334.

[40] M. S. CHEN, N. R. DONO, AND R. RAMASWAMI, *A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks*, IEEE Journal on Selected Areas in Communications, SAC-8, 6 (1990), pp. 1048–1057.

[41] F. CHUNG, *Diameter of communication networks*, in Proceedings of Symposia in Applied Mathematics , vol. 34, 1986, pp. 1–18.

[42] S.-P. CHUNG AND K. W. ROSS, *On nonblocking multirate interconnection networks*, SIAM J. Comput., 20 (1991), pp. 726–736.

[43] C. J. COLBOURN, J. H. DINITZ, AND D. R. STINSON, *Applications of combinatorial designs to communications, cryptography, and networking*, in Surveys in combinatorics, 1999 (Canterbury), Cambridge Univ. Press, Cambridge, 1999, pp. 37–100.

[44] N. DE BRUIJN, *A combinatorial problem*, in Koninklijke Nederlandse Academie van Wetenschappen Proc. , vol. A49, 1946, pp. 758–764.

[45] C. DERMAN, G. J. LIEBERMAN, AND S. M. ROSS, *On the consecutive-2-out-of-n: f system*, IEEE Trans. Rel., 31 (1982), pp. 57–63.

[46] C. DRAGONE, *Efficient n x n star coupler based on fourier optics*, Electronics Letters, 24 (1988), pp. 942–944.

[47] D. DU, D. HSU, AND F. HWANG, *Doubly-linked ring networks*, IEEE Trans. on Computers , C-34 (1985), pp. 853–855.

[48] ——, *Hamiltonian property of d-consecutive digraphs*, Math. Comput. Modeling 17, 11 (1993), pp. 61–63.

[49] D. DU, D. HSU, F. HWANG, AND X. ZHANG, *The hamiltonian property of generalized de bruijn digraphs*, Journal of Combinatorial Theory, Ser B 52 (1991), pp. 1–8.

[50] D. DU, D. HSU, AND G. W. PECK, *Connectivity of consecutive-d digraphs*, Discrete Applied Mathematics, 38 - 38 (1992), pp. 169–178.

[51] D.-Z. DU, F. CAO, AND D. F. HSU, *de Bruijn digraphs, Kautz digraphs, and their generalizations*, in Combinatorial network theory, Kluwer Acad. Publ., Dordrecht, 1996, pp. 65–105.

[52] D. Z. DU, B. GAO, F. K. HWANG, AND J. H. KIM, *On multirate rearrangeable Clos networks*, SIAM J. Comput., 28 (1999), pp. 464–471 (electronic).

[53] D. Z. DU AND D. F. HSU, *On hamiltonian consecutive-d digraphs*, Combinatorics and Graph Theory, 25 (1989), pp. 47–55.

[54] D.-Z. DU, D. F. HSU, AND D. J. KLEITMAN, *Modification of consecutive-d digraphs*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, (1991).

[55] D. Z. DU AND F. K. HWANG, *Optimal consecutive-2-out-of-n systems*, Math. Oper. Res., 11 (1986), pp. 187–191.

[56] ——, *Optimal assignments for consecutive-2 graphs*, SIAM J. Algebraic Discrete Methods, 8 (1987), pp. 510–518.

[57] D. Z. DU AND F. K. HWANG, *Generalized de Bruijn digraphs*, Networks, 18 (1988), pp. 27–38. 64 n 2.

[58] D. Z. DU AND F. K. HWANG, *Optimal assembly of an s-stage k-out-of-n system*, SIAM J. Discrete Math., 3 (1990), pp. 349–354.

[59] D.-Z. DU AND F. K. HWANG, *Combinatorial group testing and its applications*, World Scientific Publishing Co. Inc., River Edge, NJ, 1993.

[60] ——, eds., *Advances in switching networks*, American Mathematical Society, Providence, RI, 1998. Papers from the workshop held as part of the DIMACS Special Year on Network Switching at Princeton University, Princeton, NJ, July 7–9, 1997.

[61] D.-Z. DU, F. K. HWANG, Y. JUNG, AND H. Q. NGO, *Optimal consecutive-k-out-of-$(2k+1)$: G cycle*, J. Global Optim., 19 (2001), pp. 51–60.

[62] J. DUATO, S. YALAMANCHILL, AND L. NI, *Interconnection Networks*, IEEE Computer Society, Washington, 1997.

[63] A. G. D′YACHKOV, A. J. MACULA, JR., AND V. V. RYKOV, *New constructions of superimposed codes*, IEEE Trans. Inform. Theory, 46 (2000), pp. 284–290.

[64] A. G. D′YACHKOV, V. V. RYKOV, AND A. M. RASHAD, *Superimposed distance codes*, Problems Control Inform. Theory/Problemy Upravlen. Teor. Inform., 18 (1989), pp. 237–250.

[65] Z. FANG, X. LI, AND L. M. NI, *On the communication complexity of generalized* 2*-D convolution on array processors*, IEEE trans. on comp., C-38, 2 (1989), pp. 184–194.

[66] M. R. FELLOW, *Transversals of vertex partitions of graphs*, SIAM J. Disc. Math., 3 (1990), pp. 206–215.

[67] T.-Y. FENG AND S.-W. SEO, *A new routing algorithm for a class of rearrangeable networks*, IEEE Trans. Comput., 43 (1994), pp. 1270–1280.

[68] M. A. FIOL, J. L. A. YEBRA, AND I. A. DE MIQUEL, *Line digraph iterations and the (d, k) digraph problem*, IEEE Trans. on Comp., C-33, 5 (1984), pp. 400–403.

[69] P. FISHBURN, F. K. HWANG, D. Z. DU, AND B. GAO, *On* 1*-rate wide-sense non-blocking for* 3*-stage Clos networks*, Discrete Appl. Math., 78 (1997), pp. 75–87.

[70] H. FLEISCHNER AND M. STIEBITZ, *A solution to a colouring problem of P. Erdős*, Discrete Math., 101 (1992), pp. 39–48. Special volume to mark the centennial of Julius Petersen's "Die Theorie der regulären Graphs", Part II.

[71] S. FLOYD AND V. JACOBSON, *Random Early Detection gateways for Congestion Avoidance*, IEEE/ACM Transactions on Networking, 1 (1993), pp. 397–413.

[72] B. GAO AND F. K. HWANG, *Wide-sense nonblocking for multirate* 3*-stage Clos networks*, Theoret. Comput. Sci., 182 (1997), pp. 171–182.

[73] W. C. GENG, F. JAHANIAN, AND S. SECHREST, *An Optimal Bandwidth Allocation Strategy for the Delivery of Compressed Prerecorded Video*, ACM/Springer-Verlag Multimedia Systems Journal, (1996).

[74] A. R. HAMMONS, JR., P. V. KUMAR, A. R. CALDERBANK, N. J. A. SLOANE, AND P. SOLÉ, *On the apparent duality of theKerdock and Preparata codes*, in Applied algebra, algebraic algorithms and error-correcting codes (San Juan, PR, 1993), Springer, Berlin, 1993, pp. 13–24.

[75] ——, *The $\mathbf{z}_4$-linearity of Kerdock, Preparata, Goethals, and related codes*, IEEE Trans. Inform. Theory, 40 (1994), pp. 301–319.

[76] R. HASIMOTO-BELTRAN AND A. KHOKHAR, *Pixel Level Interleaving Schemes for Robust Image Communication*, 17th IEEE Symposium on Reliable Distributed Systems, '98, (1998), pp. 455–460.

[77] J. HAYES, *An adaptive technique for local distribution*, IEEE Trans. Commun., 26 (1978), pp. 1178–1186.

[78] B. Heinrichs and R. Karabek, *Tcp/ip supporting real-time applications: The predictive delay control algorithm*, in Second International Workshop on Network and Operating Systems Support for Digital Audio and Video, 1992, pp. 45–47.

[79] H. Hinton, *An Introduction to Photonic Switching Fabrics*, Plenum Press, New York, NY, 1993.

[80] M. G. Hluchyj and M. J. Karol, *ShuffleNet: An application of generalized perfect shuffles to multihop lightwave networks*, IEEE Transactions on Communications, COM-37 (1990).

[81] N. Homobono, *Connectivity of generalized de bruijn and kautz graphs*, in Proc. 11-th British Conference, Ars Combinatoria , 1988.

[82] N. Homobono and C. Peyrat, *Connectivity of imase and itoh digraphs*, IEEE Transactions on Computers, C-37 (1988), pp. 1459–1461.

[83] S. T. Huang and S. K. Tripathi, *Finite state model and compatibility theory: New analysis tools for permutation networks*, IEEE Trans. Comput., 35 (1986), pp. 509–601.

[84] S. T. Huang, S. K. Tripathi, N. S. Chen, and Y. C. Tseng, *An efficient routing algorithm for realizing linear permutations on p^t-shuffle-exchange networks*, IEEE Trans. Comput., 40 (1991), pp. 1292–1298.

[85] J. H. Hui, *Switching and traffic theory for integrated broadband networks*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1990.

[86] F. K. Hwang, *Rearrangeability of multiconnection three-stage Clos networks*, Networks, 2 (1972), pp. 301–306.

[87] F. K. Hwang, *The hamiltonian property of linear functions*, Operation Research Letters, 6 (1987), pp. 125–127.

[88] F. K. Hwang, *Invariant permutations for consecutive-k-out-of-n cycles*, IEEE Trans. Rel., 38 (1989), pp. 65–67.

[89] ——, *Extremal permutations with respect to weak majorizations*, European J. Combin., 17 (1996), pp. 637–645.

[90] F. K. Hwang, *The mathematical theory of nonblocking switching networks*, World Scientific Publishing Co. Inc., River Edge, NJ, 1998.

[91] F. K. HWANG AND C. H. LIN, *Broadcasting in a 3-stage point-to-point nonblocking network*, Inter. J. Rel. Qual. and Safety Eng., 2 (1995), pp. 299–307.

[92] M. IMASE AND M. ITOH, *Design to minimize a diameter on building block network*, IEEE Transactions on Computers, C-30 (1981), pp. 439–442.

[93] ——, *A design for directed graphs with minimum diameter*, IEEE Transactions on Computers, C-32 (1983), pp. 782–784.

[94] M. IMASE, T. SONEOKA, AND K. OKADA, *Connectivity of Regular Directed Graphs with Small Diameters*, IEEE Transactions on Computers, c-34 (1985), pp. 267–273.

[95] A. JALALI, A. G. HAWKES, L. CUI, AND F. K. HWANG, *The optimal consecutive-k-out-of-n: g line for $n \leq 2k$*, Naval Res. Logist. to appear.

[96] W. KAUTZ, *Bounds on Directed (d, k) Graphs*, Theory of Cellular Logic Networks and Machines, AFCRL-68-0668 Final Report, (1968), pp. 20–28.

[97] W. H. KAUTZ AND R. C. SINGLETON, *Nonrandom binary superimposed codes*, IEEE Trans. Inf. Theory, 10 (1964), pp. 363–377.

[98] S. KESHAV, *An engineering approach to computer networking: ATM networks, the Internet, and the telephone network*, Addison-Wesley, Reading, MA, USA, 1997.

[99] D. S. KIM, D.-Z. DU, AND P. M. PARDALOS, *A coloring problem on the n-cube*, Discrete Appl. Math., 103 (2000), pp. 307–311.

[100] M. KIM, H. YOON, AND S. MAENG, *On the correctness of inside-out routing algorithm*, IEEE Trans. Comput., 46 (1997), pp. 820–823.

[101] C. P. KRUSKAL, M. SNIR, AND A. WEISS, *The distribution of waiting times in clocked multistage interconnection networks*, IEEE Trans. Comput., 37 (1988), pp. 1337–1352.

[102] M. KUMAR, D. DIAS, AND J. JUMP, *Switching strategies in shuffle-exchange packet-switched networks*, IEEE trans. on comp., C-34, 2 (1985), pp. 180–186.

[103] V. KUMAR, A. GRAMA, A. GUPTA, AND G. KARYPIS, *Introduction to Parallel Computing*, Benjamin/Cummings, 1993.

[104] W. KUO, W. ZHANG, AND M. ZUO, *A consecutive-k-out-of-n: g system: The mirror image of a consecutive-k-out-of-n: f system*, IEEE Trans. Rel., 39 (1990), pp. 244–253.

[105] D. H. LAWRIE, *Access and alignment of data in an array processor*, IEEE Transactions on Computers, C-24 (1975), pp. 1145–1155.

[106] T. T. LEE AND P. P. TO, *Non-blocking routing properties of Clos networks*, in Advances in switching networks (Princeton, NJ, 1997), Amer. Math. Soc., Providence, RI, 1998, pp. 181–195.

[107] S. C. LIEW, *A general packet replication scheme for multicasting in interconnection networks*, in Proceedings of the 14th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'95), Los Alamitos, CA, USA, Apr. 1995, IEEE Computer Society Press, pp. 394–401.

[108] G.-H. LIN, D.-Z. DU, X.-D. HU, AND G. XUE, *On rearrangeability of multirate Clos networks*, SIAM J. Comput., 28 (1999), pp. 1225–1231 (electronic).

[109] G.-H. LIN, D.-Z. DU, W. WU, AND K. YOO, *On 3-rate rearrangeability of Clos networks*, in Advances in switching networks (Princeton, NJ, 1997), Amer. Math. Soc., Providence, RI, 1998, pp. 315–333.

[110] N. LINIAL AND M. TARSI, *Interpolation between bases and the shuffle exchange network*, European J. Combin., 10 (1989), pp. 29–39.

[111] L. LOVÁSZ AND M. D. PLUMMER, *Matching theory*, North-Holland Publishing Co., Amsterdam, 1986. Annals of Discrete Mathematics, 29.

[112] A. J. MACULA, *A simple construction of d-disjunct matrices with certain constant weights*, Discrete Math., 162 (1996), pp. 311–312.

[113] ——, *Error-correcting nonadaptive group testing with d^e-disjunct matrices*, Discrete Appl. Math., 80 (1997), pp. 217–222.

[114] ——, *Probabilistic nonadaptive group testing in the presence of errors and dna library screening*, Annals of Combinatorics, (1999), pp. 61–69.

[115] F. J. MACWILLIAMS AND N. J. A. SLOANE, *The theory of error-correcting codes. I*, North-Holland Publishing Co., Amsterdam, 1977. North-Holland Mathematical Library, Vol. 16.

[116] D. M. MALON, *Optimal consecutive-2-out-of-n: f component sequencing*, IEEE Trans. Rel., 33 (1984), pp. 414–418.

[117] ——, *Optimal consecutive-k-out-of-n: f component sequencing*, IEEE Trans. Rel., 34 (1985), pp. 46–49.

[118] S. McCane, V. Jacobson, and M. Vetterli, *Receiver-driven layered multicast*, ACM SIGCOMM '96, (1996), pp. 117–130.

[119] R. Melen and J. S. Turner, *Nonblocking multirate networks*, SIAM J. Comput., 18 (1989), pp. 301–313.

[120] Miel, *Constant geometry fast fourier transforms on array processors*, IEEETC: IEEE Transactions on Computers, 42 (1993).

[121] M. L. Molle and G. C. Polyzos, *Conflict resolution algorithms and their performance analysis*, Tech. Rep. TR-CS93-300, Department of Computer Science and Engineering, University of California, San Diego, Jul 1980.

[122] S. Muthukrishnan, *On optimal strategies for searching in presence of errors*, in Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (Arlington, VA, 1994), New York, 1994, ACM, pp. 680–689.

[123] H. Q. Ngo and D.-Z. Du, *A survey on combinatorial group testing algorithms with applications to dna library screening*, in DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, vol. 55, Providence, RI, 2000, Amer. Math. Soc., pp. 171–182.

[124] ——, *Notes on the complexity of switching networks*, in Advances in Switching Networks, D.-Z. Du and H. Q. Ngo, eds., Kluwer Academic Publisher, 2001. to appear.

[125] Ni, *Issues in designing truly scalable interconnection networks*, in ICPP: 25th International Conference on Parallel Processing, 1996.

[126] A. Y. Oruc and M. Y. Oruc, *Equivalence relations among interconnection networks*, Journal of Parallel and Distributed Computing, 2 (1985), pp. 30–49.

[127] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, *Modeling TCP Throughput: A Simple Model and its Empirical Evaluation*, ACM SIGCOMM '98, Vancouver, CA, (1998).

[128] D. S. Parker, Jr., *Notes on shuffle/exchange-type switching networks*, IEEE Trans. Comput., 29 (1980), pp. 213–222.

[129] V. Paxson, *End-to-End Internet Packet Dynamics*, ACM SIGCOMM '97, Cannes, France, (1997).

[130] M. C. Pease, III, *The indirect binary n-cube microprocessor array*, IEEE Transactions on Computers, C-26 (1977), pp. 458–473.

[131] G. D. Pifarre, L. Gravano, S. A. Felperin, and J. L. C. Sanz, *Fully adaptive minimal deadlock-free packet routing in hypercubes, meshes, and other networks: Algorithms and simulations*, IEEE Transactions on Parallel and Distributed Systems, 5 (1994), pp. 247–263.

[132] M. S. Pinsker, *On the complexity of a concentrator*, in Proceedings of the 7th International Teletraffic Conference, Stockholm, June 1973, 1973, pp. 318/1–318/4.

[133] N. Pippenger, *Superconcentrators*, SIAM J. Comput., 6 (1977), pp. 298–304.

[134] ——, *Communication networks*, in Handbook of theoretical computer science, Vol. A, Elsevier, Amsterdam, 1990, pp. 805–833.

[135] S. V. Raghavan, S. K. Tripathi, and B. Prabhakaran, *On QoS Parameters and Services for MultiMedia Applications*, Tech. Rep. 3167, Department of Computer Science, University of Maryland, College Park, MD 20742 USA, 1994.

[136] C. S. Raghavendra and R. V. Boppana, *On self-routing in benes and shuffle-exchange networks*, IEEE Transactions on Computers, 40 (1991), pp. 1057–1064.

[137] C. S. Raghavendra and A. Varma, *Rearrangeability of the five-stage shuffle/exchange network for $N = 8$*, IEEE Trans. Comm., 35 (1987), pp. 808–812.

[138] S. Ramanathan and P. V. Rangan, *Feedback Techniques for Intra-Media Continuaity and Inter-Media Synchronization in Distributed Multimedia Systems*, The Computer Journal, 36 (1993), pp. 19 – 33.

[139] S. Reddy, J. Kuhl, S. Hosseini, and H. Lee, *On digraphs with minimum diameter and maximum connectivity*, in Proc. of 20-th Annual Allerton Conference , 1982, pp. 1018–1026.

[140] S. Reddy, D. Pradhan, and J. Kuhl, *Directed graphs with minimal diameter and maximal connectivity*, tech. rep., School of Engineering Oakland Univ. Tech. Rep., 1980.

[141] S. Roman, *Coding and Information Theory*, Springer-Verlag, New York, 1992.

[142] H. Schulzrinne, S. Casner, R. Frederick, and S. McCane, *RTP: A Transport Protocol for Real-Time Applications* , RFC 1889, (1994).

[143] S. Sedigh, J. Joshi, A. Bashandy, and A. Ghafoor, *Evaluation of Filtering Mechanisms for MPEG Video Communications*, 17th IEEE Symposium on Reliable Distributed Systems, '98, (1998), pp. 449–454.

[144] B. SMITH, *Cyclic-UDP: A Priority Driven Best-Effort Protocol.* http://www.cs.cornell.edu/Info/Faculty/Brian_Smith.html/Publications.

[145] B. SMITH, L. ROWE, AND S. YEN, *A Tcl/Tk Continuous Media Player*, in Proceedings Tcl 1993 Workshop, June 1993.

[146] B. C. SMITH, *Implementation Techniques for Continuous Media Systems and Applications*, PhD thesis, University of California, Berkeley, 1994.

[147] R. STEINMETZ AND G. BLAKOWSKI, *A media synchronization survey: Reference model, specification and case studies*, IEEE Journal on Selected Areas in Communication, 14 (1996), pp. 5–35.

[148] H. STONE, *Parallel processing with perfect shuffle*, IEEE Trans. Comput., 20 (1971), pp. 153–161.

[149] A. S. TANENBAUM, *Computer Networks*, Prentice-hall International, Inc., 1996.

[150] D. TOWSLEY, *Providing quality of service in packet switched networks*, in Performance Evaluation of Computer Communication Systems, L. Donatiello and R. Nelson, eds., Springer Verlag, 1993, pp. 560–586.

[151] K.-H. TSAI AND D.-W. WANG, *Lower bounds for wide-sense non-blocking Clos network*, in Computing and combinatorics (Taipei, 1998), Springer, Berlin, 1998, pp. 213–218.

[152] B. TSYBAKOV AND V. MIKHAILOV, *Free synchronous packet access in a broadcast channel with feedback*, Problemy Peredachi Informatsii, 14 (1978), pp. 32–59 (259–280).

[153] J. H. VAN LINT AND R. M. WILSON, *A course in combinatorics*, Cambridge University Press, Cambridge, 1992.

[154] S. VARADARAJAN, H. Q. NGO, AND J. SRIVASTAVA, *Error spreading: A perception-driven approach to error handling in continuous media streaming*, in Proceedings of the 20th International Conference on Distributed Computing Systems (ICDCS), Taipei, Taiwan, Republic of China, Apr 2000, IEEE Comp. Soc., pp. 475–483.

[155] A. VARMA AND C. RAGHAVENDRA, *Rearrangeability of multistage shuffle/exchange networks*, IEEE Trans. Comm., COM-36, 10 (1988), pp. 1138–1147.

[156] ——, *Fault-tolerant routing in multistage interconnection networks*, IEEE trans. on comp., C-38, 3 (1989), pp. 385–393.

[157] B. WAH AND D. LIN, *Transformation-based Reconstruction for Audio Transmissions over the Internet*, 17th IEEE Symposium on Reliable Distributed Systems, ’98, (1998), pp. 211–217.

[158] P.-J. WAN, *Conflict-free channel set assignment for an optical cluster-based interconnection network based on star graphs*, Informatica, to appear, (1997).

[159] ——, *Multichannel Lightwave Networks*, PhD thesis, Computer Science Department, University of Minnesota, Minneapolis, MN 55455, 1997.

[160] P.-J. WAN, *Near-optimal conflict-free channel set assignments for an optical cluster-based hypercube network*, J. Comb. Optim., 1 (1997), pp. 179–186.

[161] ——, *Conflict-free channel set assignment for an optical cluster interconnection network based on rotator digraphs*, Theoret. Comput. Sci., 207 (1998), pp. 193–201. In memoriam of Ronald V. Book.

[162] D. WIJESEKERA AND J. SRIVASTAVA, *Quality of Service (QoS) Metrics for Continuous Media*, Multimedia Tools and Applications, 3 (1996), pp. 127–166.

[163] D. WIJESEKERA, J. SRIVASTAVA, A. NERODE, AND M. FORESTI, *Experimental Evaluation of Loss Perception in Continuous Media*, to appear in ACM - Springer Verlag, (1997).

[164] D. WIJESEKERA, S. VARADARAJAN, S. PARIKH, J. SRIVASTAVA, AND A. NERODE, *Performance evaluation of media losses in the continuous media toolkit*, in International Workshop on Multimedia Software Engineering , MSE’98, Kyoto, Japan, 1998.

[165] J. K. WOLF, *Born again group testing: multiaccess communications*, IEEE Transaction on Information Theory, IT-31 (1985), pp. 185–191.

[166] C. L. WU AND T. Y. FENG, *The reverse-exchange interconnection network*, IEEE Trans. Comput., 29 (1980), pp. 801–811.

[167] ——, *The universality of the shuffle-exchange network*, IEEE Trans. Comput., 30 (1981), pp. 324–332.

[168] J. H. YAO, Y. M. CHEN, AND T. VERMA, *MPEG-Based Audio-on-demand experiment for the Internet*, Internetworking ’96, Nara, Japan, (1996).

[169] L. ZHANG, S. DEERING, D. ESTRIN, S. SHENKER, AND D. ZAPPALA, *RSVP: A New Resource Reservation Protocol*, IEEE Network, 5 (1993), pp. 8 – 18.

[170] Z. L. Zhang, S. Nelakuditi, R. Aggarwal, and R. P. Tsang, *Efficient Selective Frame-Discard Algorithms for Stored Video Delivery across resource Constrained Networks*, To Appear in IEEE INFOCOM '99, (1999).

[171] Z. L. Zhang, J. D. Salehi, J. F. Kurose, and D. Towsley, *Supporting Stored Video: Reducing Rate Variability and End-to-End Resource Requirements through Optimal Smoothing*, ACM SIGMETRICS, (1996).

[172] M. Zuo and W. Kuo, *Design and performance analysis of consecutive-k-out-of-n structure*, Naval Res. Logist., 37 (1990), pp. 203–230.