INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

U·M·I

University Microfilms International
A Bell & Howell Information Company
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
313/761-4700 800/521-0600

Order Number 9315520

Fault tolerant node disjoint routing in n-dimensional hypercube interconnection network

Yoon, Kisong, Ph.D.

City University of New York, 1993

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

A

FAULT TOLERANT NODE DISJOINT ROUTING IN n-DIMENSIONAL HYPERCUBE INTERCONNECTION NETWORK

by

Kisong Yoon

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

1993

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy

12/22/92
Date

Michael Anshel
Chair of Examining Committee

12/22/92
Date

Stanley Habib
Executive Officer

Theodore Brown

Christina Zamfirescu

Igidor Gertner
Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

FAULT TOLERANT NODE DISJOINT ROUTING IN n-DIMENSIONAL HYPERCUBE INTERCONNECTION NETWORK

by

Kisong Yoon

Advisor: Professor Michael Anshel

One property of n-dimensional hypercube interconnection network is that there exist n node disjoint paths between any two arbitrary nodes of Hamming distance k in n-dimensional hypercube, where k paths are of length k and n-k paths are of length k+2. A hypercube with faulty elements is called an injured hypercube. There are algorithms which generate those n node disjoint paths. But it is not guaranteed that this algorithms can be used for an injured hypercube. In a hypercube, there are redundant elements which are not used for n node disjoint paths. This redundancy is used for fault tolerant node disjoint routing. In this dissertation, we are looking for conditions of faulty elements guaranteeing the existence of n node disjoint paths using k x k matrix and (n-k) x (k+2) matrix for the set of paths of length k and k+2 respectively, where the entries of the matrices are the bit positions of the address of the nodes in a hypercube.

To my wife

Contents

List of Tables

List of Illustrations

Fault Tolerant Node Disjoint Routing in a Hypercube Network

Kisong Yoon

Abstract - One property of n-dimensional hypercube network is that there exist n node disjoint paths between any two arbitrary nodes of Hamming distance k in n dimensional hypercube, where k paths are of length k and n-k paths are of length k+2. In this paper, we are interested in how to find n node disjoint paths when a certain number of elements, nodes and/or edges, are faulty in the hypercube. A hypercube with faulty elements is called an injured hypercube. Given two nodes in a hypercube, it can be evaluated how many nodes are actually, possibly and never used for n node disjoint paths between two nodes. Since there exist a certain number of redundant elements for n node disjoint paths, with appropriate routing we might be able to maintain the same number of node disjoint paths for the same source and destination nodes in the hypercube in the presence of injured elements. Given two arbitrary nodes, some of faulty nodes can be precluded for n node disjoint paths by simply examining the addresses of faulty nodes. We show how this preclusion can be done. Then we discuss methods to generate n node disjoint paths in two different cases of length k and $k+2$. In the case of length k, we propose a node disjoint routing algorithm for a non-injured hypercube by constructing $k \times k$ matrix, called shifted paths generating matrix(SPGM), and show how to use this matrix for fault tolerant node disjoint routing of length k. Then we improve this algorithm by rearranging the entries of the matrix with some restrictions. From a $k \times k$ SPGM, $[(k-2)!\cdot(k-3)!\cdot\cdot\cdot 3!\cdot 2!\cdot 1!]^2$ different sets of node disjoint paths of length k can be generated. In the case of length $k+2$, we also propose a node disjoint routing algorithm for a non-injured hypercube by constructing $(n-k) \times (k+2)$ matrix, called paths generating matrix(PGM) and also show that how this matrix can be used for fault tolerant node disjoint routing of length $k+2$ in the same fashion. From this routing matrix, $(k!)^{n-k}$ all possible sets of node disjoint paths of length $k+2$ can be generated. Finally, we study a broadcast algorithm for n-cube using a SPGM.

I. Introduction

When a computer application requires a very large amount of computation that must be completed in a reasonable amount of time, it becomes necessary to use machines with correspondingly large computing capacity. One approach to achieve this purpose is to configure a system that contains a large number of conventional processors. The attractive feature of this is that the individual processors do not have to be complex, high-performance units. But efficient routing of messages between processors is a very important point for the performance of the multi computer system.

Among various interconnection structures, hypercube multi computers have been attractive because each node has precisely the same types of connections to the remainder of the system as all the other nodes. For example, each node has the same number of adjacent nodes. A hypercube also has other topological properties[3]. As the use of multi computer system increases, fault tolerant routing becomes very important. Fault tolerant routing means the successful routing of messages between two arbitrary non injured nodes in the presence of injured elements. Several researches related to hypercube architecture have been reported[5]-[9]. It is known that many algorithms for a regular hypercube can be handled by a systematic procedures. Those systematic algorithms cannot be applied directly to an injured hypercube. We will study one systematic algorithm which generates k node disjoint paths of length k. As mentioned before, it is not guaranteed that this algorithm can be used for an injured hypercube. In this paper, we consider node disjoint routing algorithms for an hypercube in the presence of injury. If all injury information of the injured hypercube network is known in advance, we might be able to find n node disjoint paths in a systematic way for the injured hypercube network by avoiding the injury. Several researches related to the fault diagnosis in multi-processor architecture have been reported[10]-[36]. The basic assumption in this paper is that all injury information is known beforehand. But since injury can occur randomly in the hypercube network, it is not guaranteed that there exist n node disjoint paths in n dimensional hypercube in the presence of injury. In this paper, several node disjoint routing algorithms are discussed for certain types of injuries.

Injured hypercubes can be divided into two categories, node injured and edge injured hypercubes. A node injured hypercube contains faulty nodes. In other words, these injured nodes cannot transmit and receive any information. These injured nodes must be avoided in node disjoint routing. An edge injured hypercube contains broken connections between two arbitrary nodes. These two nodes containing broken edge are said to be partially injured nodes. This broken edge must be avoided in node disjoint routing. The node injury can be considered as the special case of the edge injury, saying that all connections of the node are broken. Generally speaking, an edge injured hypercube has more flexibility than a node injured hypercube in node disjoint routing because the partially injured node in the edge injured hypercube can be selected for node disjoint routing by choosing non broken connection of the node.

The basic idea of node disjoint routing in an injured hypercube is that there are a certain number of unused nodes in the hypercube network which might be able to replace injured nodes used in n node disjoint paths between two nodes of distance n in the n dimensional hypercube. Suppose n node disjoint paths between two arbitrary nodes in n dimensional hypercube are being used. If injury occurs to nodes unused for n node disjoint paths, clearly there still exist n node disjoint paths. And if injury occurs to nodes which are used for any node disjoint paths and they can be replaced with non injured and unused nodes, there still exist n node disjoint paths. The main topic in this paper is how to examine in a systematic manner whether or not the injured nodes used in n node disjoint paths can be replaced with non injured and unused nodes.

We use matrices to generate node disjoint paths. k x k matrices are used for the set of the paths of length k and (n-k) x (k+2) matrices are used for the set of the paths of length k+2, where k is Hamming distance of two nodes in n-dimensional hypercube.

The contents of this paper is as follows.

II. Preliminaries

Definition) An n-dimensional hypercube, denoted by n-cube or Q_n, is defined recursively as follows.

i) $Q_1 = K_2$

ii) $Q_n = K_2 \times Q_{n-1}$

,where K_2 is the complete graph with two nodes and x is the product operation of two graphs[1]. Thus Q_n contains 2^n nodes which can be labeled $a_1a_2 \ldots a_n$, where each a_i is either 0 or 1 and $n \cdot 2^{n-1}$ links because each node has n adjacent nodes. The addresses of any two neighbors differ in exactly one bit position.

Hypercube can be viewed in two different ways. The first view is shown in fig. 2-1. In this way, it is very hard to draw a picture of hypercubes for dimension five and above. The second view is shown in fig. 2-2. In this way, all nodes in n dimensional hypercube are partitioned into n+1 groups according to their weights of relative addresses to the source node.

fig.2-1

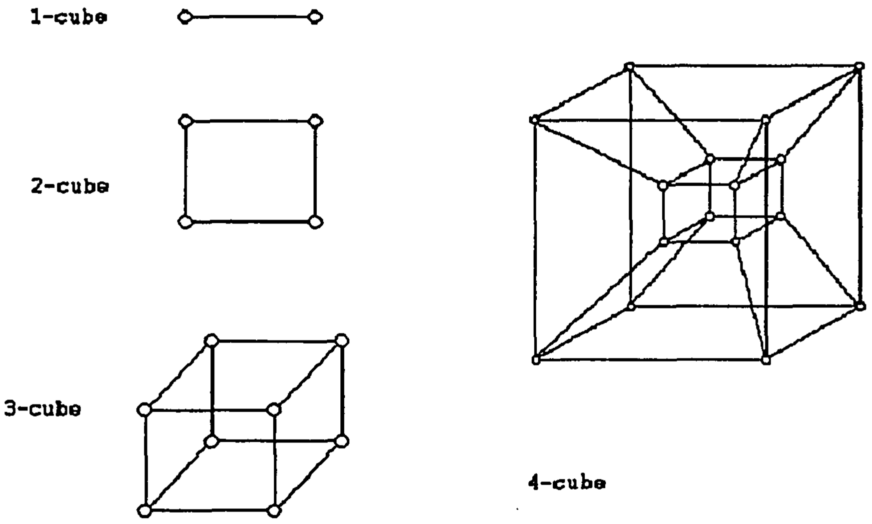
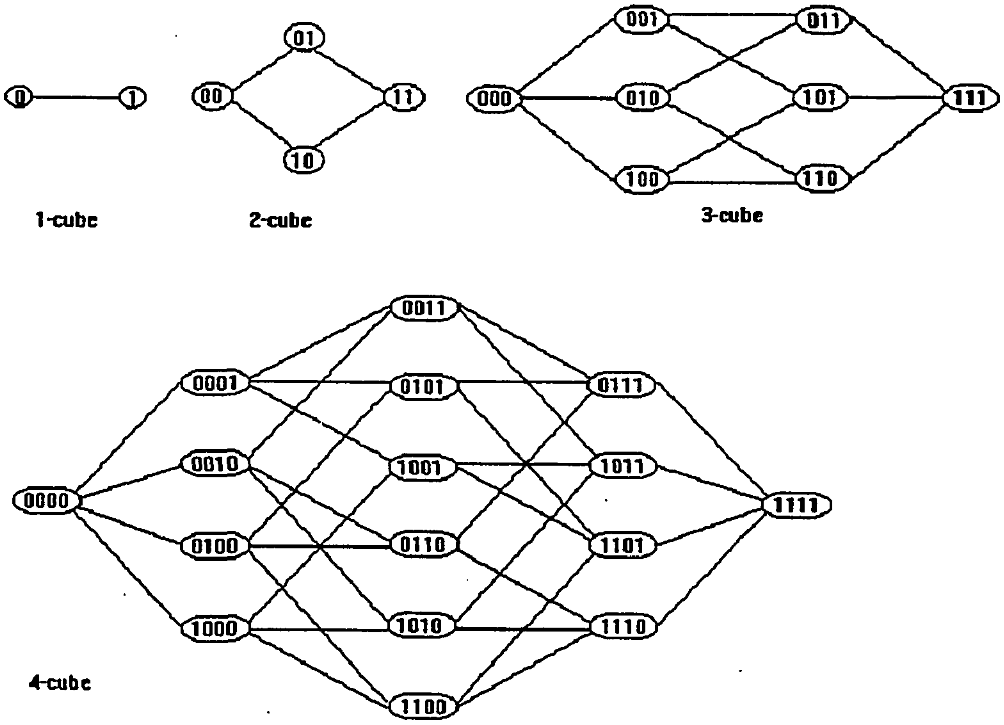


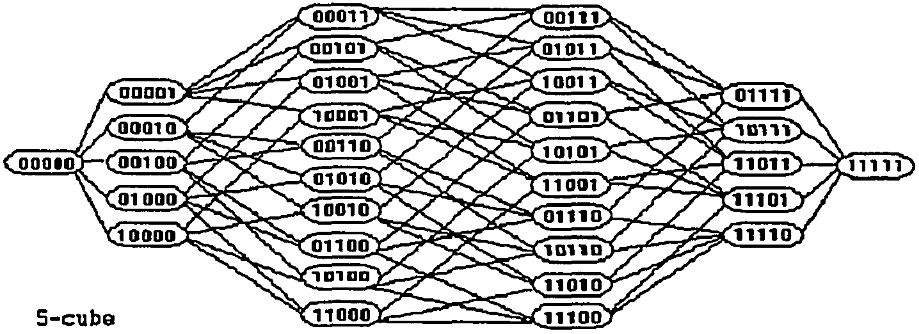
fig.2-2



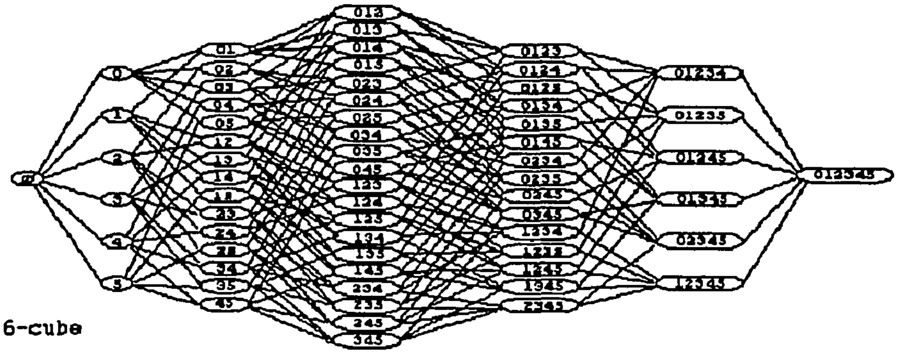
1-cube 2-cube 3-cube

4-cube

fig.2-2(continued)



5-cube



6-cube

Definition) The Hamming distance between a pair of nodes u and w is denoted by H(u,w) and defined as the number of digits that must change in one node so that the other node results[2].

example) For n = 4

$H(0011,1101) = 3$

$H(0000,1111) = 4$

$H(0001,0000) = 1$

Definition) The relative address of a node u with respect to another node to w, denoted by $u/_w$, is defined as $u/_w = u \oplus w$, where $\oplus$ represents EXCLUSIVE-OR operation.

example)

The relative address of node 0011 to 1101 is 1110.

The relative address of node 0101 to 1010 is 1111.

The relative address of a hypercube with respect to a node u can be determined by the relative addresses of all the nodes it contains.

Definition) The weight of a node u, denoted by W(u), is defined as the number of 1's in the address of the node.

example) For n = 5

$W(00000) = 0$

$W(00100) = 1$

$W(11111) = 5$

All nodes in a hypercube can be distributed to stages according to the weights of them. There are n+1 stages because $0 \le W(u) \le n$ for an arbitrary node u. Stage i represents a stage where all nodes have same weight i. A node at a stage has no connections with other nodes at the same stage. A node at stage i has connections to stage i-1 and i+1 only. We discuss this in more detail later.

<u>Definition)</u> N_k and N_{k+2} represent the sets of bit position numbers where the binary value of $A \oplus B$ is 1 and 0 respectively.

example) for n = 10

$A = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$B = 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1$

$A \oplus B = 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1$

bit position 9 8 7 6 5 4 3 2 1 0

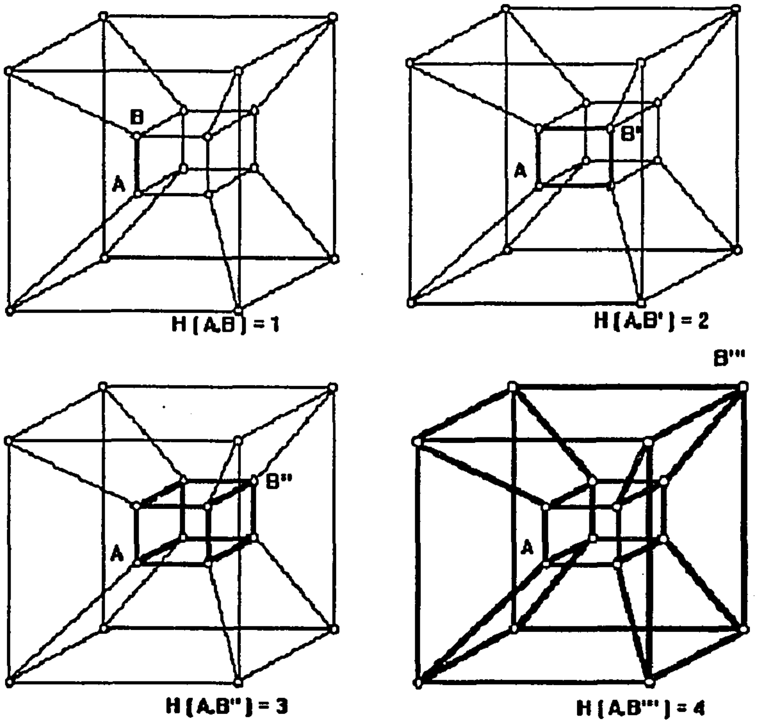
$N_k = \{ 0, 2, 5, 6, 7, 8 \}$

$N_{k+2} = \{ 1, 3, 4, 9 \}$

<u>Definition)</u> The subcube of two arbitrary nodes u and w in n-cube, denoted by SQ(u,w), defined as the smallest hypercube including u and w. k-dimensional subcube and k-subcube also represent a subcube of diameter k.

For example, when u = 0011 and w = 1101, we get Hamming distance H(u,w) = 3. There are eight nodes in this subcube SQ(u,w). Since a subcube is another hypercube, it has all same properties of hypercube. See fig. 2-3 and fig. 2-4.

fig.2-3

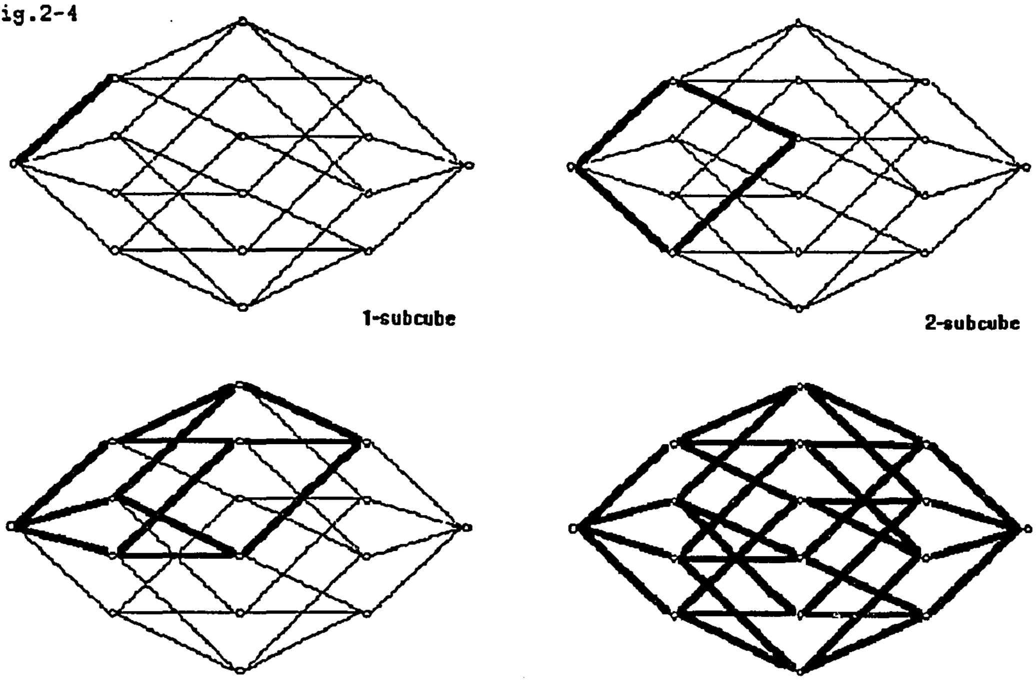


H(A,B)=1 H(A,B')=2

H(A,B'')=3 H(A,B''')=4

A path in a hypercube is represented by a sequence of nodes in which every two consecutive nodes are physically adjacent to each other in the hypercube. The number of links on a path is called the length of the path.

fig.2-4



1-subcube

2-subcube

3-subcube

4-subcube

One important point is that there are different paths between two different nodes u and w. The existence of such paths might be useful for speeding up transfers of large amounts of data between two nodes. It also provides a way of selecting alternative routes in case a given node in a path is failing. In order for this to be possible, the paths must not cross each other, i.e., they must not have common nodes, except for two nodes u and w. We refer to such paths as node disjoint paths.

One property of n dimensional hypercube is as follows. Let u and w be two arbitrary nodes such that $H(u,w) = k$. Then there are exactly n node disjoint paths from u to w. These paths are composed of k disjoint paths of length k and n-k disjoint paths of length k+2.

Let two arbitrary nodes u and w be the source and destination nodes respectively in the subcube SQ(u,w) with Hamming distance $H(u,w) = k$. There are 2^k nodes in the subcube SQ(u,w). For the convenience,all addresses of the used nodes are relative to the source node. Thus the weight of the source is 0 and the weight of the destination node is k.

All nodes used in a node disjoint paths in n-cube except the source and destination nodes are called intermediate nodes. There are k-1 intermediate nodes in each path of length k and there are k+1 intermediate nodes in each path of length k+2. Obviously, all intermediate nodes of all paths must be distinct.

We take four dimensional hypercube as a specific example to describe fault tolerant routing. For four dimensional hypercube, there are four different node disjoint paths between 0000 and 1111. Suppose the following four node disjoint paths are being used between 0000 and 1111.

0000 - 0001 - 0101 - 0111 - 1111

0000 - 0010 - 1010 - 1110 - 1111

0000 - 0100 - 1100 - 1101 - 1111

0000 - 1000 - 1001 - 1011 - 1111

In this example, there are sixteen nodes in the hypercube but only fourteen nodes are used. There are always two unused nodes no matter how we select four node disjoint paths between two arbitrary nodes of Hamming distance four in four dimensional hypercube. See fig. 2-5 and fig. 2-6. In our example, nodes 0011 and 0110, are not used. If those two unused nodes are injured, the above four node disjoint paths still can be used. Suppose two nodes, 0101 and 1010, are injured. But four node disjoint paths still can be taken as follows.

0000 - 0001 - 0011 - 0111 - 1111

0000 - 0010 - 0110 - 1110 - 1111

0000 - 0100 - 1100 - 1101 - 1111

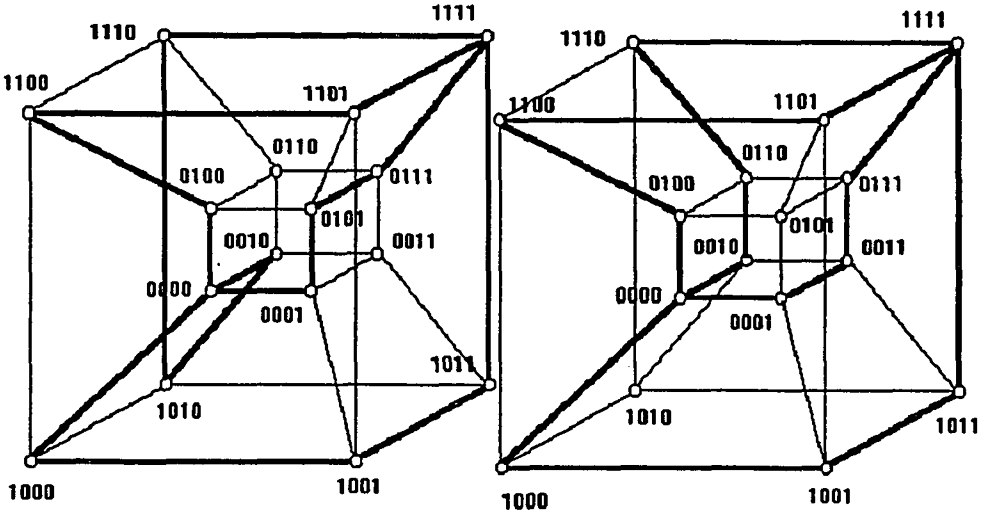
0000 - 1000 - 1001 - 1011 - 1111

In this case, two injured nodes 0101 and 1010 have been replaced with non injured and unused nodes 0011 and 0110 respectively.

Suppose a node 1000 is injured. Even though there are two unused nodes in the network, no one can replace the injured node 1000. As the result, there can not be four node disjoint paths with the injured node 1000. Suppose 0011, 1100 and 1001 are injured. Clearly, there can not be four node disjoint paths because there are only two unused nodes in the network. This shows that it should be examined whether or not there still exist n node disjoint paths in n dimensional hypercube in the presence of injury.

Let S_k and S_{k+2} represent two sets of the paths of length k and k+2 respectively. Next two chapters discuss how to find n node disjoint paths for S_k and S_{k+2} in the presence of injury.

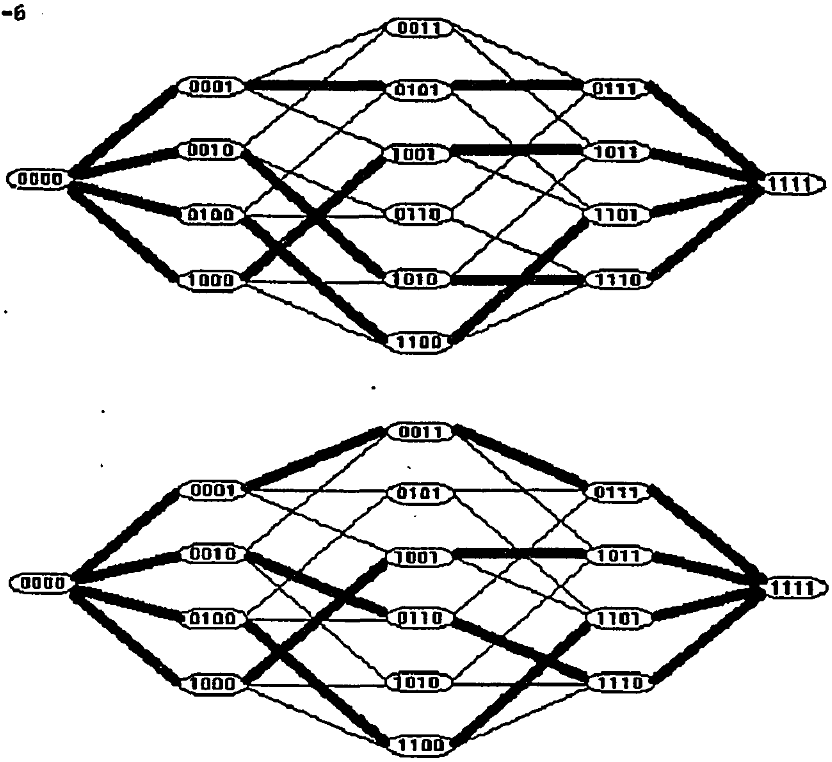
fig.2-5



Nodes 0110 and 0011 are not used.

Nodes 0101 and 1010 are not used.

fig.2-6


0011
0001
0101
0111
0010
1001
1011
0000
0110
1111
0100
1101
1000
1010
1110
1100
0011
0001
0101
0111
0010
1001
1011
0000
0110
1111
0100
1101
1000
1010
1110
1100

III. Fault Tolerant Routing for the paths of length k

Let me repeat one of the properties for n-dimensional hypercube network. There are k node disjoint paths of length k between two arbitrary nodes, u and v, of Hamming distance k in the n dimensional hypercube network, where $k \le n$. Since subcube SQ(u,v) is also a hypercube, it has all properties of a hypercube. From now on, we consider k node disjoint paths between two arbitrary nodes in the SQ(u,v), whose Hamming distance is k and all addresses we use in the following context mean the relative addresses to the source node.

This chapter is composed of the following three sections.

A) Section A describes the numbers of unused nodes and edges in the k-subcube.

B) Section B describes the algorithm using shifted routing matrix in the presence of injury.

C) Section C describes the entry exchange algorithms which expand the shifted routing matrix in the presence of injury.

A. Redundant Elements in k-dimensional subcube.

First of all, we have to know how many redundant nodes and edges in the k-dimensional subcube in order to have k node disjoint paths of length k between two arbitrary node, u and v, of $H(u,v) = k$. Now, we have the following Lemma.

Lemma 1. There are $2^k - k^2 + k - 2$ unused nodes in the k-subcube in order to have k node disjoint paths of length k between two arbitrary nodes whose Hamming distance is k.

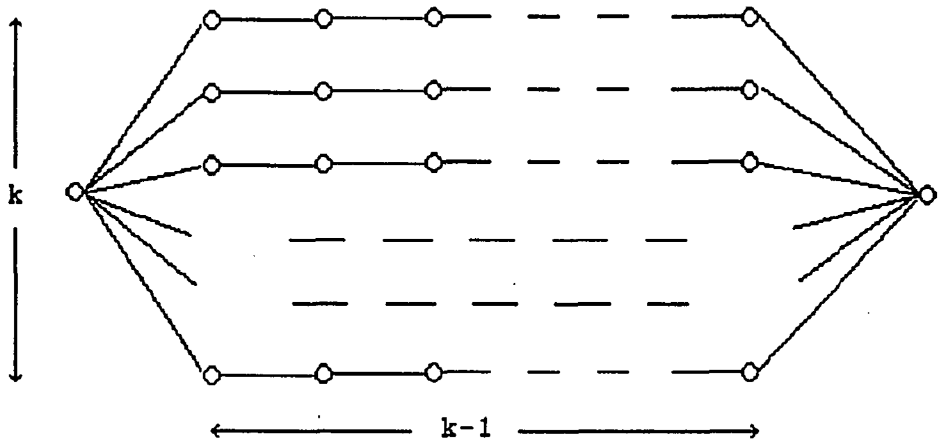
Proof) From the properties of hypercube, there are 2^k nodes in k-subcube and there are k node disjoint paths between two arbitrary nodes, u and v, whose Hamming distance is k. Let's say that node u is the source and node v is the destination. Clearly, u and v are common for all k node disjoint paths. Since there are $k-1$ intermediate nodes for each path, there are $k \cdot (k-1)$ intermediate nodes for all paths. See fig. 3-1. The number of all nodes used for k node disjoint paths is $k \cdot (k-1) + 2$. Therefore, the number of unused nodes in k-subcube is $2^k - k^2 + k - 2$.

∎

Table 3-1 shows how many nodes are unused in k-subcube. As shown in the table, there is no unused nodes for dimension 1, 2 and 3. This means that all nodes in those subcubes must be used for k node disjoint paths, where k = 1,2 and 3. From the dimension 4, we start to have unused nodes. The higher dimension we have, the more unused nodes we have. We will see that higher dimensional hypercube has more possibility for replacement. Similarly, we can generate the formula for unused edges.

fig.3-1

k node disjoint paths in k-subcube



There are k-1 intermediate nodes in each path.

There are k edges in each path.

Lemma 2. There are $k\cdot(2^{k-1} - k)$ unused edges in k-subcube in order to have k node disjoint paths of length k between two arbitrary nodes whose Hamming distance is k.

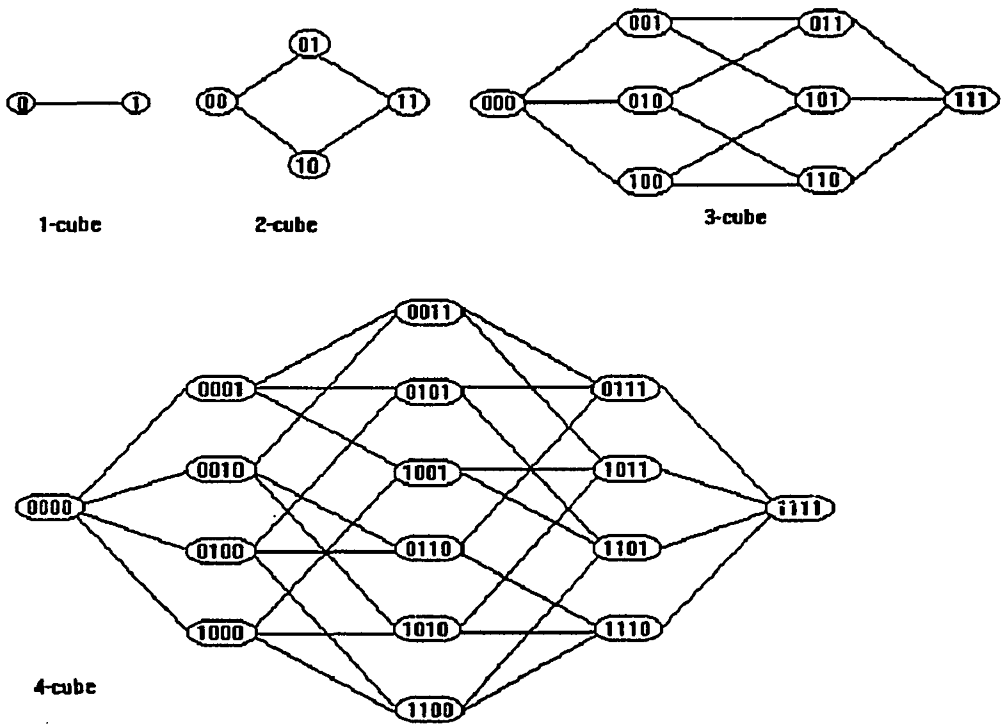
Proof) Since there are 2^k nodes in SQ(u,v) and each node has degree k, there are $2^k\cdot k / 2$ or $2^{k-1}\cdot k$ edges. There are k node disjoint paths between two arbitrary nodes, u and v, whose length is k. Again, let's say that node u is the source and node v is the destination. Since there are k edges for each path, there are k^2 edges for all paths. See fig. 3-1. Therefore, the number of unused edges in k-subcube is $2^{k-1}\cdot k - k^2$ or $k\cdot(2^{k-1} - k)$.

∎

Table 3-2 shows how many edges are unused in the dimension k. As shown in the table, there is no unused nodes for dimension 1 and 2. This means that all edges in those hypercube must be used for k node disjoint paths, where k = 1,2. From the dimension 3, we start to have unused edges. The higher dimension we have, the more unused edges we have.

As mentioned before, hypercube can be viewed in two different ways. The second view is shown in fig. 3-2. In this way, all nodes in n dimensional hypercube are partitioned into n+1 groups according to their weights of relative addresses to the source node. We call these groups stages. There are n+1 stages and stage number represents the weight of the nodes at that stage. There is no connections among the nodes at same stage. Since only one bit is different between two adjacent nodes, the nodes at stage i have connections to stage i-1 and stage i+1 only. A node at stage i has i connections to stage i-1 and has n-i connections to stage i+1, where $1 \le i \le n$. This is also true for k-dimensional subcube.

fig.3-2



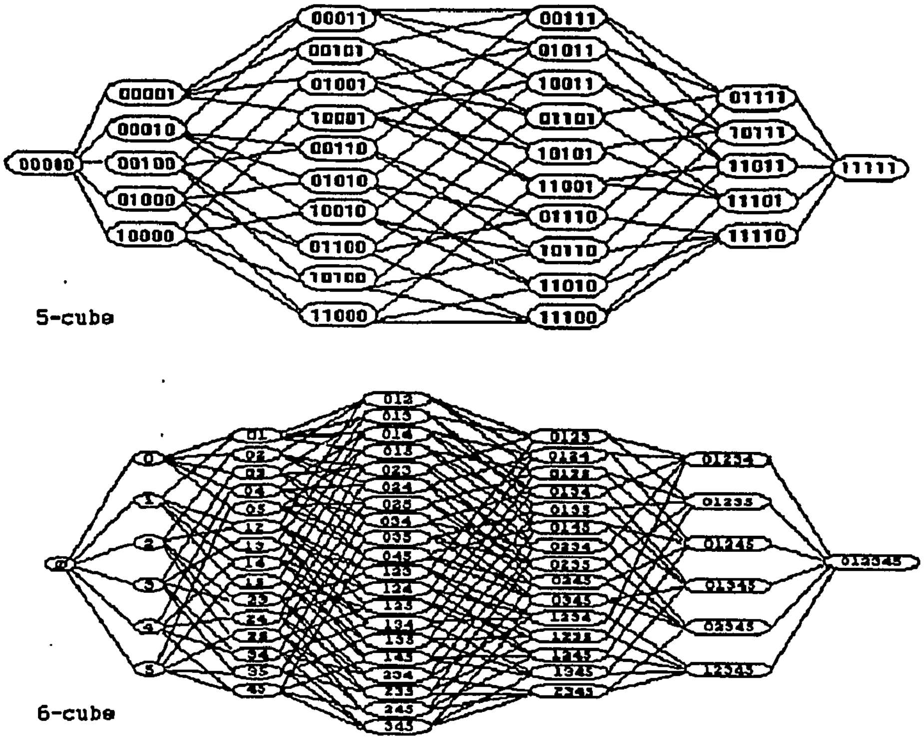
1-cube

2-cube

3-cube

4-cube

fig.3-2(continued)



<u>Lemma 3.</u> There are ${}_kC_i - k$ unused nodes at each stage i, where $1 \le i \le k-1$, in order to have k node disjoint paths of length k in k-subcube.

Proof) There are ${}_kC_i$ nodes at each stage i because each node has k digit address and i number of 1's. k nodes are used for k node disjoint paths at each stage i, where $1 \le i \le k-$

1. Since there is only one node at stage 0 and k which are the source and destination nodes respectively, there cannot be unused node at stage 0 and n. Therefore, the number of unused nodes at each stage i is ${}_kC_i - k$, where $1 \le i \le k-1$.

■

Clearly, the sum of nodes at all stages must be 2^k. This can be proved by the following mathematical formula.

$$\sum_{i=0}^{k} {}_kC_i = 2^k$$

The sum of unused nodes must be $2^k - k^2 + k - 2$ which is derived from Lemma 1.

$$\sum_{i=1}^{k-1} ({}_kC_i - k)$$

$$= \sum_{i=1}^{k-1} {}_kC_i - (k-1) \cdot k$$

$$= \sum_{i=0}^{k} {}_kC_i - (k-1) \cdot k - 2$$

$$= 2^k - (k-1) \cdot k - 2$$

$$= 2^k - k^2 + k - 2$$

Table 3-3 shows how many nodes each stage has for hypercube of dimension one through ten. Table 3-4 shows how many unused nodes each stage has for hypercubes of dimension one through ten. Notice that there is no unused node at stage 0, 1, k-1 and k. Since there is only one node at stage 0 and stage n, the source and destination nodes respectively, they can not be injured for k node disjoint paths. Since there are only k nodes at stage 1 and k-1, there can not be injured nodes for k node disjoint paths in k-dimensional subcube.

Table 3-1

dimension k	# of nodes in the k-cube	# of nodes for k node disjoint paths	# of unused nodes
1	2	2	0
2	4	4	0
3	8	8	0
4	16	14	2
5	32	22	10
6	64	32	32
7	128	44	84
8	256	58	198
9	512	74	438
10	1024	92	932
20	1048576	382	1048194

Table 3-2

dimension k	# of edges in the k-cube	# of edges for k node disjoint paths	# of unused edges
1	1	1	0
2	4	4	0
3	12	9	3
4	32	16	16
5	80	25	55
6	192	36	156
7	448	49	399
8	1024	64	960
9	2304	81	2223
10	5120	100	5020

Table 3-3 Each entry represents the number of nodes at each stage. (X is undefined.)

k	stage number										
	0	1	2	3	4	5	6	7	8	9	10
1	1	1	X	X	X	X	X	X	X	X	X
2	1	2	1	X	X	X	X	X	X	X	X
3	1	3	3	1	X	X	X	X	X	X	X
4	1	4	6	4	1	X	X	X	X	X	X
5	1	5	10	10	5	1	X	X	X	X	X
6	1	6	15	20	15	6	1	X	X	X	X
7	1	7	21	35	35	21	7	1	X	X	X
8	1	8	28	56	70	56	28	8	1	X	X
9	1	9	36	84	126	126	84	36	9	1	X
10	1	10	45	120	210	252	210	120	45	10	1

Table 3-4 Each entry represents the number of unused nodes at each stage. (X is undefined.)

k	stage number										
	0	1	2	3	4	5	6	7	8	9	10
1	0	0	X	X	X	X	X	X	X	X	X
2	0	0	0	X	X	X	X	X	X	X	X
3	0	0	0	0	X	X	X	X	X	X	X
4	0	0	2	0	0	X	X	X	X	X	X
5	0	0	5	5	0	0	X	X	X	X	X
6	0	0	9	14	9	0	0	X	X	X	X
7	0	0	14	28	28	14	0	0	X	X	X
8	0	0	20	48	62	48	20	0	0	X	X
9	0	0	27	75	117	117	75	27	0	0	X
10	0	0	35	110	200	242	200	110	35	0	0

B. A Shifted Path Generating Matrix for S_k(SPGM).

In this section, we discuss an algorithm which generates k node disjoint paths in k-dimensional subcube, denoted by k-subcube, in the presence of a certain type of injury.

Recall the following definition.

<u>Definition)</u> N_k and N_{k+2} represent the sets of bit position numbers where the binary value of $A \oplus B$ is 1 and 0 respectively.

example) for n = 10

$$
\begin{array}{rcccccccccc}
A = & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
B = & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\hline
A \oplus B = & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\
\text{bit position} & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0
\end{array}
$$

$N_k = \{ 0, 2, 5, 6, 7, 8 \}$

$N_{k+2} = \{ 1, 3, 4, 9 \}$

For node disjoint paths of length k in k-subcube, we need the elements of N_k.

An address of a node in a n-dimensional hypercube is usually denoted by n digit binary number. Here, we introduce a different notation for the address of a node in n-cube.

<u>Definition)</u> (a_1, a_2, a_3, , , , a_p) ,where $0 \le p \le n$ and $0 \le a_i \le n-1$, is defined as the address of the node where the value of bit position a_i is 1.

The position number of the least significant bit is 0.
The position number of the most significant bit is n-1.

example) For ten dimensional hypercube,
Node 0011001101 is denoted by (0,2,3,6,7).
Node 0001010101 is denoted by (0,2,4,6).
Node 0000000011 is denoted by (0,1).
Node 0000000000 is denoted by ().
Node 1111111111 is denoted by (0,1,2,3,4,5,6,7,8,9).

We call the numbers a_1 through a_p <u>node numbers</u>. This notation has the following properties.

property1) The number of node numbers of a node is the weight of the address of the node.
property2) All node numbers of a node are distinct.
property3) All node at stage i have i node numbers.
property4) A node can be denoted by any permutation of node numbers of the node.

example) (1,3,6), (3,1,6) and (6,3,1) represent the same node 0001001010.

<u>Definition)</u> Routing Matrix(RM) is defined as follows, where $1 \le p,q \le n$.

$$RM = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots\cdots & a_{0,q-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots\cdots & a_{1,q-1} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots\cdots & a_{2,q-1} \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ a_{p-1,0} & a_{p-1,1} & a_{p-1,2} & \cdots\cdots & a_{p-1,q-1} \end{bmatrix}$$

RM is a matrix where the following conditions are satisfied

1) Each entry represents a node in n-cube.
2) Each row represents one of n node disjoint paths in n-cube.
3) $a_{i,j}$ represents an intermediate node for one of n node disjoint path in n-cube.
4) The contents of each entry is a relative address to the source node.
5) All entries in the last column represent the same node

which is the destination node.

6) The source node is not shown in this matrix.

7) At row i, node numbers of the node at column j must be included to node numbers of the node at column j+1.

8) Since an entry represents an intermediate node of one of node disjoint paths, all entry must be distinct except entries in the last column.

example) For our example, n = 10 and k = 6.

$$M = \begin{bmatrix} (0) & (0,2) & (0,2,5) & (0,2,5,6) & (0,2,5,6,7) & (0,2,5,6,7,8) \\ (2) & (2,8) & (2,8,7) & (2,8,7,6) & (2,8,7,6,5) & (2,8,7,6,5,0) \\ (5) & (5,8) & (5,8,6) & (5,8,6,7) & (5,8,6,7,0) & (5,8,6,7,0,2) \\ (6) & (6,8) & (6,8,7) & (6,8,7,0) & (6,8,7,0,2) & (6,8,7,0,2,5) \\ (7) & (7,8) & (7,8\ 0) & (7,8,0,5) & (7,8,0,5,2) & (7,8,0,5,2,6) \\ (8) & (8,0) & (8,0,5) & (8,0,5,6) & (8,0,5,6,2) & (8,0,5,6,2,7) \end{bmatrix}$$

The above matrix M is a RM because each row represents one of six node disjoint paths of length six between ( ) and (0,2,5,6,7,8) in 10-cube.

Definition) A Path Generating Matrix (PGM) is defined as follows, where $1 \le p,q \le n$.

$$PGM = \begin{bmatrix} a_{0,0} & a_{0,1} & a_{0,2} & \cdots\cdots & a_{0,q-1} \\ a_{1,0} & a_{1,1} & a_{1,2} & \cdots\cdots & a_{1,q-1} \\ a_{2,0} & a_{2,1} & a_{2,2} & \cdots\cdots & a_{2,q-1} \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot \\ a_{p-1,0} & a_{p-1,1} & a_{p-1,2} & \cdots\cdots & a_{p-1,q-1} \end{bmatrix}$$

PGM is a matrix where the following conditions are satisfied.

1) $0 \le a_{i,j} \le n-1$
2) From the Path Generating Matrix(PGM) , Routing Matrix(RM) is constructed by the following algorithm.

$$b_{i,j} = (a_{i,0}, a_{i,1}, a_{i,2}, \ldots, a_{i,j})$$

,where $PGM = [\, a_{i,j} \,]$ and $RM = [\, b_{i,j} \,]$.

example) For our example, $n = 10$ and $k = 6$.

$$M = \begin{bmatrix} 0 & 2 & 5 & 6 & 7 & 8 \\ 2 & 8 & 7 & 6 & 5 & 0 \\ 5 & 8 & 6 & 7 & 0 & 2 \\ 6 & 8 & 7 & 0 & 2 & 5 \\ 7 & 8 & 0 & 5 & 2 & 6 \\ 8 & 0 & 5 & 6 & 2 & 7 \end{bmatrix}$$

This matrix M is a PGM because it generates the following RM.

$$M = \begin{bmatrix} (0) & (0,2) & (0,2,5) & (0,2,5,6) & (0,2,5,6,7) & (0,2,5,6,7,8) \\ (2) & (2,8) & (2,8,7) & (2,8,7,6) & (2,8,7,6,5) & (2,8,7,6,5,0) \\ (5) & (5,8) & (5,8,6) & (5,8,6,7) & (5,8,6,7,0) & (5,8,6,7,0,2) \\ (6) & (6,8) & (6,8,7) & (6,8,7,0) & (6,8,7,0,2) & (6,8,7,0,2,5) \\ (7) & (7,8) & (7,8\ 0) & (7,8,0,5) & (7,8,0,5,2) & (7,8,0,5,2,6) \\ (8) & (8,0) & (8,0,5) & (8,0,5,6) & (8,0,5,6,2) & (8,0,5,6,2,7) \end{bmatrix}$$

Before describing the routing algorithm for injured hypercube, let's discuss Shifted Path Generating Matrix(SPGM). It has been proved that SPGM is a PGM[4].

<u>Definition)</u> Shifted Path Generating Matrix(SPGM) is a matrix such that given any sequence of the elements of N_k, the SPGM is obtained according to the following algorithm.

Let a selected sequence be $a_0, a_1, a_2, , , , a_{k-2}, a_{k-1}$.

```
for i = 0 to k-1
     for j = 0 to k-1
          $b_{i,j} = a_{(i+j) \bmod k}$
     next
next
```

, where SPGM = $[\ b_{i,j}\ ]$.

In other words, SPGM is obtained by placing the selected sequence at row 0 and then shifting it to the left circularly k-1 times with placing it to the next rows of the SPGM matrix.

$$SPGM = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{k-2} & a_{k-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{k-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{k-1} & a_0 & a_1 & \cdots\cdots & a_{k-3} & a_{k-2} \end{bmatrix}$$

example) For our example, $n = 10$, $k = 6$ and

$N_k = \{ 0, 2, 5, 6, 7, 8 \}$

Selected Sequence is 0, 2, 5, 6, 7, 8.

$$SPGM = \begin{bmatrix} 0 & 2 & 5 & 6 & 7 & 8 \\ 2 & 5 & 6 & 7 & 8 & 0 \\ 5 & 6 & 7 & 8 & 0 & 2 \\ 6 & 7 & 8 & 0 & 2 & 5 \\ 7 & 8 & 0 & 2 & 5 & 6 \\ 8 & 0 & 2 & 5 & 6 & 7 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,2) & (0,2,5) & (0,2,5,6) & (0,2,5,6,7) & (0,2,5,6,7,8) \\ (2) & (2,5) & (2,5,6) & (2,5,6,7) & (2,5,6,7,8) & (2,5,6,7,8,0) \\ (5) & (5,6) & (5,6,7) & (5,6,7,8) & (5,6,7,8,0) & (5,6,7,8,0,2) \\ (6) & (6,7) & (6,7,8) & (6,7,8,0) & (6,7,8,0,2) & (6,7,8,0,2,5) \\ (7) & (7,8) & (7,8\ 0) & (7,8,0,2) & (7,8,0,2,5) & (7,8,0,2,5,6) \\ (8) & (8,0) & (8,0,2) & (8,0,2,5) & (8,0,2,5,6) & (8,0,2,5,6,7) \end{bmatrix}$$

Since N_k has k elements and the first and last elements of the selected sequence are consider to be adjacent, (k-1)! different SPGM's can be obtained for k-subcube.

example) For n = 4, k = n and $N_k = \{ 0, 1, 2, 3 \}$, there are (4 - 1)! or 6 different SPGM's.

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 & 3 & 2 \\ 1 & 3 & 2 & 0 \\ 3 & 2 & 0 & 1 \\ 2 & 0 & 1 & 3 \end{bmatrix} \quad \begin{bmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 3 & 0 \\ 1 & 3 & 0 & 2 \\ 3 & 0 & 2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 2 & 3 & 1 \\ 2 & 3 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 1 & 0 & 2 & 3 \end{bmatrix} \quad \begin{bmatrix} 0 & 3 & 1 & 2 \\ 3 & 1 & 2 & 0 \\ 1 & 2 & 0 & 3 \\ 2 & 0 & 3 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 3 & 2 & 1 \\ 3 & 2 & 1 & 0 \\ 2 & 1 & 0 & 3 \\ 1 & 0 & 3 & 2 \end{bmatrix}$$

Four node disjoint path from the first SPGM are shown in fig. 3-3.

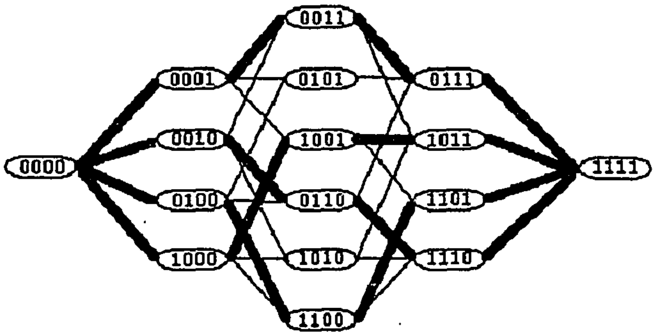
Let's consider the following three matrices.

$$\begin{bmatrix} 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \end{bmatrix} \qquad \begin{bmatrix} 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

Though these matrices look different, they produce the same four node disjoint paths as the first matrix of above six different matrices. Thus if two SPGM's produce the same node disjoint paths, they are considered as the same SPGM's.

fig 3-3

$$\text{SPGM} \begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix} \qquad \text{RM} \begin{bmatrix} (0) & (0,1) & (0,1,2) & (0,1,2,3) \\ (1) & (1,2) & (1,2,3) & (1,2,3,0) \\ (2) & (2,3) & (2,3,0) & (2,3,0,1) \\ (3) & (3,0) & (3,0,1) & (3,0,1,2) \end{bmatrix}$$



Now let's discuss the node disjoint routing algorithm for injured hypercube. From the construction of SPGM, we can find one property that all node numbers of an entry $a_{i,j}$ of a Routing Matrix derived from SPGM are adjacent in the selected sequence considering that the first and last elements of the sequence considered to be adjacent. In other words, nodes whose node numbers are not adjacent in the selected sequence are not used for n node disjoint paths. It is easier to examine the adjacency by distributing the elements of N_k on the circle.

example) For n = 5, k = n and N_k = { 0, 1, 2, 3, 4 }

The selected sequence is 0, 1, 2, 3, 4.

0

4 1

3 2

(0,2), (1,3), (0,1,3) and (0,2,3) do not appear in the SPGM because all node numbers of the given nodes are not adjacent. (0,1), (3,4), (2,3,4) and (4,0,1) appear in the SPGM because all node numbers are adjacent.

As mentioned before, there cannot be injury at stage 1 and k-

1. This can be also shown on the circle. From the definition, all elements must appear on the circle. This means that the injury cannot happen to a node of weight one in order to have k node disjoint paths. And It is impossible to keep node numbers of nodes of weight k-1 not adjacent on the circle. This also means that the injury cannot happen to a node of weight k-1 in order to have k node disjoint paths.

Again there are (k-1)! different sequences with N_k for k-subcube. If one node of weight i, $2 \le i \le k-2$, is injured, there are $(k-1)! - (k-i)! \cdot i!$ sequences where node numbers of the injured node are not adjacent.

This can be proved by simple mathematical formula.

$(k-1)!$	is the number of sequences with the elements of N_k.
$(k-i)! \cdot i!$	is the number of sequences with k elements where i node numbers of the injured node are adjacent.

We have proved the following Lemma from this property.

<u>Lemma 1.</u> If a SPGM can be constructed avoiding the adjacency of the node numbers for all injured nodes for two arbitrary nodes of Hamming distance k in k-subcube, there still exist k node disjoint paths of length k between those two nodes.

example) Suppose we are looking for six node disjoint paths between 0000000000 and 0111100101 in 6-subcube and 0000100100, 0011000000, 0011100000 and 0110000101 are injured.

0000000000 = ( )
0111100101 = (0,2,5,6,7,8)
N_k = { 0, 2, 5, 6, 7, 8 }

0000100100 = (2,5)
0011000000 = (6,7)
0011100000 = (5,6,7)
0110000101 = (7,8,0,2)

If the sequence of our example, 0, 2, 5, 6, 7, 8 is used,

0 2
8 5
7 6

$$\text{SPGM} = \begin{bmatrix} 0 & 2 & 5 & 6 & 7 & 8 \\ 2 & 5 & 6 & 7 & 8 & 0 \\ 5 & 6 & 7 & 8 & 0 & 2 \\ 6 & 7 & 8 & 0 & 2 & 5 \\ 7 & 8 & 0 & 2 & 5 & 6 \\ 8 & 0 & 2 & 5 & 6 & 7 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,2) & (0,2,5) & (0,2,5,6) & (0,2,5,6,7) & (0,2,5,6,7,8) \\ (2) & \underline{(2,5)} & (2,5,6) & (2,5,6,7) & (2,5,6,7,8) & (2,5,6,7,8,0) \\ (5) & (5,6) & \underline{(5,6,7)} & (5,6,7,8) & (5,6,7,8,0) & (5,6,7,8,0,2) \\ (6) & \underline{(6,7)} & (6,7,8) & (6,7,8,0) & (6,7,8,0,2) & (6,7,8,0,2,5) \\ (7) & (7,8) & (7,8\ 0) & \underline{(7,8,0,2)} & (7,8,0,2,5) & (7,8,0,2,5,6) \\ (8) & (8,0) & (8,0,2) & (8,0,2,5) & (8,0,2,5,6) & (8,0,2,5,6,7) \end{bmatrix}$$

All four injured nodes appear in the RM (underlined above) because all node numbers of all injured nodes are adjacent in the sequence. But with the same injury, we can select a different sequence, 2, 6, 0, 5, 8, 7, where all node numbers of all injured nodes are not adjacent. Then there still exist six node disjoint paths of length six in the presence of four injured nodes. Here we have different sequence, SPGM and RM.

7 2

8 6

5 0

$$\text{SPGM} = \begin{bmatrix} 2 & 6 & 0 & 5 & 8 & 7 \\ 6 & 0 & 5 & 8 & 7 & 2 \\ 0 & 5 & 8 & 7 & 2 & 6 \\ 5 & 8 & 7 & 2 & 6 & 0 \\ 8 & 7 & 2 & 6 & 0 & 5 \\ 7 & 2 & 6 & 0 & 5 & 8 \end{bmatrix}$$

RM

$$\begin{bmatrix} (2) & (2,6) & (2,6,0) & (2,6,0,5) & (2,6,0,5,8) & (2,6,0,5,8,7) \\ (6) & (6,0) & (6,0,5) & (6,0,5,8) & (6,0,5,8,7) & (6,0,5,8,7,2) \\ (0) & (0,5) & (0,5,8) & (0,5,8,7) & (0,5,8,7,2) & (0,5,8,7,2,6) \\ (5) & (5,8) & (5,8,7) & (5,8,7,2) & (5,8,7,2,6) & (5,8,7,2,6,0) \\ (8) & (8,7) & (8,7\ 2) & (8,7,2,6) & (8,7,2,6,0) & (8,7,2,6,0,5) \\ (7) & (7,2) & (7,2,6) & (7,2,6,0) & (7,2,6,0,5) & (7,2,6,0,5,8) \end{bmatrix}$$

Though there are four injured nodes, there still exist six node disjoint paths of length six. Since none of node numbers of injured nodes are adjacent in the sequence, four injured nodes (2,5), (6,7), (5,6,7) and (7,8,0,2) don't appear in the RM.

Now we consider the edge injured hypercube with the sequence of the elements of N_k. In node injured hypercube, the order of node numbers is not important. For example, (3,0,2,1,4) and (2,1,4,3,0) represent the same node which is the destination node 11111 in five dimensional hypercube. But those two different representations for one node show two different paths from node () which is 00000.

(3,0,2,1,4) : ()-(3)-(3,0)-(3,0,2)-(3,0,2,1)-(3,0,2,1,4)

00000 - 01000 - 01001 - 01101 - 01111 - 11111

(2,1,4,3,0) : ()-(2)-(2,1)-(2,1,4)-(2,1,4,3)-(2,1,4,3,0)

00000 - 00100 - 00110 - 10110 - 11110 - 11111

Thus the node notation with node numbers represents binary relative address and intermediate path from the source node to the specified node.

<u>Definition)</u> Perm(e_0, e_1, e_2, . . . e_q) is defined as any permutation with e_0, e_1, e_2, . . . e_q.

For example, the node 001101 can be represented by 3! or six different ways.

(0,2,3) (0,3,2) (2,0,3) (2,3,0) (3,0,2) (3,2,0)

These are all possible paths of length three from the source node 000000 to 001101. Thus the node 001101 can be denoted by Perm(0,2,3).

Since there is only one bit difference between two arbitrary adjacent nodes in n dimensional hypercube, Hamming distance of the two nodes represents the length of the shortest path between the nodes. From now on, we talk about the shortest path between two arbitrary nodes.

Suppose one edge between node u and node v , u = $(a_1, a_2, a_3, \ldots, a_i)$ and v = $(a_1, a_2, a_3, \ldots, a_i, a_{i+1})$ is injured where $1 \le i \le k-2$. Since these two nodes are partially injured, they might be used for node disjoint paths.

There are i! different paths to reach the node v via injured edge u -> v because there are i node numbers for node u. The node v is denoted by Perm$(a_1, a_2, a_3, \ldots, a_i, a_{i+1})$. The node v which is reachable from node u is denoted by (Perm$(a_1, a_2, a_3, \ldots, a_i)$, a_{i+1}) and Perm$(a_1, a_2, a_3, \ldots, a_i)$ is obtained according to the path from the source node to the node u. If these orders of node numbers of node u are avoided, the selection of the injured edge u -> v is avoided. There are $i! \cdot i$ different orders to avoid the injured edge. This can be proved as follows.

# of orders to represent node v	$(i + 1)!$
# of orders to represent node v via edge u -> v	$i!$

# of orders to represent node v not via u -> v is

$$(i + 1)! - i!$$
$$= (i + 1) \cdot i! - i!$$
$$= i \cdot i!$$

example) Suppose that the edge 0001101 -> 0011101 is injured in 7-cube. In our notation, the edge (2,3,0) -> (2,3,0,4) is injured. There are $3 \cdot 3!$ or 18 different orders to each 0011101 not via injured edge from the node 0000000.

(4,2,3,0) (2,4,3,0) (2,3,4,0)
(4,2,0,3) (2,4,0,3) (2,0,4,3)
(4,0,2,3) (0,4,2,3) (0,2,4,3)
(4,0,3,2) (0,4,3,2) (0,3,4,2)
(4,3,0,2) (3,4,0,2) (3,0,4,2)
(4,3,2,0) (3,4,2,0) (3,2,4,0)

All of the above nodes represents the same node 0011101 but they don't visit the node 0001101 which means they don't visit the injured edge 0001101 -> 0011101. In other words, as long as node number 4 does not appear in the end, the injured edge can be avoided.

We have proved the following Lemma from the above discussion.

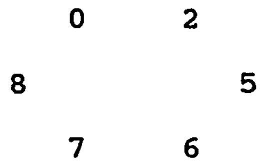
Lemma 2. If a sequence can be selected avoiding the orders of numbers of all injured edges for two arbitrary nodes of Hamming distance k in k-subcube, there still exist k node disjoint paths between them.

example) Suppose we are looking for six node disjoint paths between 0000000000 and 0111100101 in 6-subcube and the following edges are injured.

0010000000 -> 0110000000 <=> (7) -> (7,8)
0000000101 -> 0000100101 <=> (0,2) -> (0,2,5)
0000100100 -> 0001100101 <=> (2,5) -> (2,5,6)
0100000101 -> 0100100101 <=> (8,0,2) -> (8,0,2,5)
0111000001 -> 0111000101 <=> (6,7,8,0) ->(6,7,8,0,2)

0000000000 = ( )
0111100101 = (0,2,5,6,7,8)
$N_k = \{ 0, 2, 5, 6, 7, 8 \}$

If the sequence of our example, 0, 2, 5, 6, 7, 8 is used,



$$\text{SPGM} = \begin{bmatrix} 0 & 2 & 5 & 6 & 7 & 8 \\ 2 & 5 & 6 & 7 & 8 & 0 \\ 5 & 6 & 7 & 8 & 0 & 2 \\ 6 & 7 & 8 & 0 & 2 & 5 \\ 7 & 8 & 0 & 2 & 5 & 6 \\ 8 & 0 & 5 & 6 & 2 & 7 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & \underline{(0,2)} & \underline{(0,2,5)} & (0,2,5,6) & (0,2,5,6,7) & (0,2,5,6,7,8) \\ (2) & \underline{(2,5)} & \underline{(2,5,6)} & (2,5,6,7) & (2,5,6,7,8) & (2,5,6,7,8,0) \\ (5) & (5,6) & (5,6,7) & (5,6,7,8) & (5,6,7,8,0) & (5,6,7,8,0,2) \\ (6) & (6,7) & (6,7,8) & \underline{(6,7,8,0)} & \underline{(6,7,8,0,2)} & (6,7,8,0,2,5) \\ \underline{(7)} & \underline{(7,8)} & (7,8\ 0) & (7,8,0,2) & (7,8,0,2,5) & (7,8,0,2,5,6) \\ (8) & (8,0) & \underline{(8,0,2)} & \underline{(8,0,2,5)} & (8,0,2,5,6) & (8,0,2,5,6,7) \end{bmatrix}$$

All five injured edges appear in the RM (underlined above) because all orders of all injured edges appear in the sequence. But with the same injury, we can select a different sequence, 0, 5, 6, 8, 7, 5, where all orders of all injured edges don't appear. Then there still exist six node disjoint paths of length six in the presence of five injured edges. Here we have different sequence, SPGM and RM.

$$\begin{array}{cccc} & 7 & 5 & \\ 8 & & & 0 \\ & 6 & 2 & \end{array}$$

$$\text{SPGM} = \begin{bmatrix} 5 & 0 & 2 & 6 & 8 & 7 \\ 0 & 2 & 6 & 8 & 7 & 5 \\ 2 & 6 & 8 & 7 & 5 & 0 \\ 6 & 8 & 7 & 5 & 0 & 2 \\ 8 & 7 & 5 & 0 & 2 & 6 \\ 7 & 5 & 0 & 2 & 6 & 8 \end{bmatrix}$$

RM

$$\begin{bmatrix} (5) & (5,0) & \underline{(5,0,2)} & (5,0,2,6) & (5,0,2,6,8) & (5,0,2,6,8,7) \\ (0) & \underline{(0,2)} & (0,2,6) & (0,2,6,8) & \underline{(0,2,6,8,7)} & (0,2,6,8,7,5) \\ (2) & (2,6) & (2,6,8) & (2,6,8,7) & (2,6,8,7,5) & (2,6,8,7,5,0) \\ (6) & (6,8) & (6,8,7) & (6,8,7,5) & (6,8,7,5,0) & (6,8,7,5,0,2) \\ (8) & \underline{(8,7)} & (8,7\ 5) & (8,7,5,0) & (8,7,5,0,2) & (8,7,5,0,2,6) \\ \underline{(7)} & (7,5) & (7,5,0) & (7,5,0,2) & (7,5,0,2,6) & (7,5,0,2,6,8) \end{bmatrix}$$

Though there are five injured nodes, there still exist six node disjoint paths of length six. Since no orders of injured edges appears in the sequence, five injured edges don't appear in the RM. Notice that there are ten partially injured nodes but five of them (underlined in RM) are still used in the six node disjoint paths of length six.

From Lemma 1 and 2, we can derive the following theorem.

<u>Theorem 3.1</u> If a sequence can be selected from N_k avoiding the conditions of all faulty elements for two arbitrary nodes of Hamming distance k in k-subcube, there still exist k node disjoint paths between them.

C. PGM's derived from a SPGM for S_k using entry exchanges.

In the previous section, we studied SPGM for an injured hypercube. If a SPGM can be constructed avoiding all injury condition, injured nodes and/or edges, there still exist k node disjoint paths of length k in k-subcube. This algorithm using SPGM has one weak point that one injured node can cause another non injured node not to be used. One property of the SPGM method is that an adjacency of nodes always produces another adjacency. From the SPGM method, the node numbers of an injured node can not be adjacent in the selected sequence. In other words, there is always a couple of nodes which appear or don't appear in the RM derived from a SPGM at the same time. We call one node the counter node of the other. The address of one node is the complement of the other. Since injure nodes cannot appear from the selected sequence, the counter nodes of the injured nodes also cannot appear from the same sequence whether the counter nodes are injured or not.

example) Suppose node 0011010 is injured in seven dimensional hypercube.
This node is denoted by (1,3,4).
The counter node of the node (1,3,4) is 1100101 and denoted by (0,2,5,6).

Since node (1,3,4) is injured, node numbers 1,3 and 4 can not be adjacent in the sequence. Node numbers of the counter node 0,2,5 and 6 are automatically not adjacent which means that node 1100101 can not be used whether or not it is injured.

Now, we are going to study other PGM's which have more flexibility than SPGM's. Let's consider the following PGM.

For $n = 5$, $k = n$ and $N_k = \{ 0, 1, 2, 3, 4 \}$

$$PGM = \begin{bmatrix} 0 & 2 & 1 & 3 & 4 \\ 1 & 4 & 2 & 3 & 0 \\ 2 & 4 & 3 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 2 & 1 & 3 \end{bmatrix}$$

This PGM generates the following RM.

$$RM = \begin{bmatrix} (0) & (0,2) & (0,2,1) & (0,2,1,3) & (0,2,1,3,4) \\ (1) & (1,4) & (1,4,2) & (1,4,2,3) & (1,4,2,3,0) \\ (2) & (2,4) & (2,4,3) & (2,4,3,0) & (2,4,3,0,1) \\ (3) & (3,4) & (3,4,0) & (3,4,0,1) & (3,4,0,1,2) \\ (4) & (4,0) & (4,0\ 2) & (4,0,2,1) & (4,0,2,1,3) \end{bmatrix}$$

Clearly, the above PGM is not a SPGM because each element in a column is not distinct. But this PGM generates a RM for the node disjoint paths of length five. Let's take a close look at the above RM. Consider node (0,2) and its counter node (1,3,4). In the above RM, node (0,2) appears in the RM but the counter node (1,3,4) doesn't, which is impossible in a SPGM. From the above example, we can say that there exist PGM's which are not SPGM's. In this section, we talk about PGM's which are not SPGM's but derived from SPGM.

<u>Definition)</u> $\mathrm{Perm}_{i,j,l}$ represents a permutation with j^{th} entry through l^{th} entry of row i in (p x q) matrix where $0 \le i \le p-1$ and $0 \le j,l \le q-1$ and $j \le l$. $\mathrm{Perm}_{i,j,l}$ is undefined when $j > l$.

example) For n = 5

$$M = \begin{bmatrix} 0 & 3 & 2 & 1 & 4 \\ 1 & 3 & 2 & 4 & 0 \\ 2 & \underline{4} & \underline{3} & \underline{0} & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 2 & 1 & 3 \end{bmatrix}$$

$\mathrm{Perm}_{2,1,3}$, underlined above, represents one of the followings.

$$2\ \ \underline{4\ \ 3\ \ 0}\ \ 1$$
$$2\ \ \underline{4\ \ 0\ \ 3}\ \ 1$$
$$2\ \ \underline{3\ \ 4\ \ 0}\ \ 1$$
$$2\ \ \underline{3\ \ 0\ \ 4}\ \ 1$$
$$2\ \ \underline{0\ \ 3\ \ 4}\ \ 1$$
$$2\ \ \underline{0\ \ 4\ \ 3}\ \ 1$$

$Perm_{i,j,l}$ represents one of $(l - j + 1)!$ possibilities.

From the algorithm of SPGM construction, once a sequence is selected, the SPGM can start with any position of the sequence. As mentioned in the previous section, all SPGM's from the same sequence are considered all same because they generate the same node disjoint paths of length k.

example) For $n = 4$, $k = n$ and $N_k = \{ 0, 1, 2, 3 \}$

Selected sequence is 0, 1, 2, 3.

0

3 1

2

From this sequence, the following matrices can be derived.

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 2 & 3 & 0 & 1 \\ 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \end{bmatrix} \begin{bmatrix} 3 & 0 & 1 & 2 \\ 0 & 1 & 2 & 3 \\ 1 & 2 & 3 & 0 \\ 2 & 3 & 0 & 1 \end{bmatrix}$$

These four different matrices are considered as the same SPGM, because all these matrices generate the same node disjoint paths.

<u>Lemma 1.</u> Let M be a SPGM for k-subcube.

$$M = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{k-2} & a_{k-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{k-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{k-1} & a_0 & a_1 & \cdots\cdots & a_{k-3} & a_{k-2} \end{bmatrix}$$

The matrix with $Perm_{i,1,k-2}$, where $0 \le i \le k-1$, is a PGM.

Proof) We can prove this Lemma by showing that the matrix with $Perm_{i,1,k-2}$ generates Routing Matrix where all intermediate nodes are distinct.

We prove this Lemma by a contradiction.

Let's consider row 0 and row i, where $1 \le i \le k-1$.

Suppose there is a common intermediate node for row 0 with $Perm_{0,1,k-2}$ and row i, where $1 \le i \le k-1$.

In other to have common node for row 0 with $Perm_{0,1,k-2}$ and row i, a_0 must be one of the common node number because row 0 starts with a_0.

a_{k-1} must be one of the common node numbers because a_{k-1} always appears right before a_0 for row i, where $1 \le i \le k-1$.

But since a_{k-1} is the last entry of row 0 with $Perm_{0,1,k-2}$, a_{k-1} can not be one of intermediate node numbers for row 0.

This is a contradiction. Therefore, there is no common intermediate node for row 0 with $Perm_{0,1,k-2}$ and row i, where $1 \le i \le k-1$.

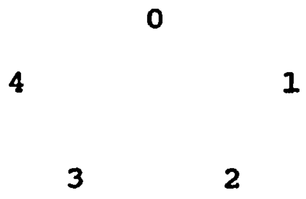
Since a SPGM can start with any row from the algorithm of SPGM construction, this is true for all other rows. This completes the proof.

■

This Lemma means that a SPGM with any permutation with all elements except the first and the last of any row in the SPGM is a PGM. From this Lemma, (k-2)! different PGM's can be derived from a k x k SPGM.

example) For n = 5, k = n and $N_k = \{ 0, 1, 2, 3, 4 \}$

Selected sequence is 0, 1, 2, 3, 4.



$$\text{SPGM} \quad = \quad \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & \underline{3} & \underline{4} & \underline{0} & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

PGM's can be obtained by permuting the underlined part. One of them and Routing Matrix are shown below.

$$\text{PGM} \quad = \quad \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & \underline{4} & \underline{3} & \underline{0} & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

$$\text{RM} = \begin{bmatrix} (0) & (0,1) & (0,1,2) & (0,1,2,3) & (0,1,2,3,4) \\ (1) & (1,2) & (1,2,3) & (1,2,3,4) & (1,2,3,4,0) \\ \underline{(2)} & \underline{(2,4)} & \underline{(2,4,3)} & \underline{(2,4,3,0)} & \underline{(2,4,3,0,1)} \\ (3) & (3,4) & (3,4,0) & (3,4,0,1) & (3,4,0,1,2) \\ (4) & (4,0) & (4,0,1) & (4,0,1,2) & (4,0,1,2,3) \end{bmatrix}$$

In this example, (5 - 2)! or 6 different PGM's can be derived from the given SPGM.

We improve this Lemma as follows.

Lemma 2. Let M be a SPGM for k-subcube.

$$M = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{k-2} & a_{k-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{k-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{k-1} & a_0 & a_1 & \cdots\cdots & a_{k-3} & a_{k-2} \end{bmatrix}$$

The matrix with $\mathrm{Perm}_{i,1,k-2}$ and $\mathrm{Perm}_{(i+1) \bmod k,1,k-2}$, where $0 \le i \le k-1$, is a PGM of the node disjoint paths of length k.

Proof) We can prove this Lemma by showing that the matrix with $\mathrm{Perm}_{i,1,k-2}$ and $\mathrm{Perm}_{(i+1) \bmod k,1,k-2}$ generates Routing matrix where all intermediate nodes are distinct.

We also prove this Lemma by a contradiction.

Let's consider row 0, row 1 and row i, where $2 \le i \le k-1$.

From the previous Lemma, there is no common intermediate node for row 0 and row i, where $2 \leq i \leq k-1$.

Now we show that there is no common intermediate node from row 0 with $Perm_{0,1,k-2}$ and row 1 with $Perm_{1,1,k-2}$ and there is no common intermediate node from row 1 with $Perm_{1,1,k-2}$ and row i, where $2 \leq i \leq k-1$.

(i) There is no common intermediate node for row 0 with $Perm_{0,1,k-2}$ and row 1 with $Perm_{1,1,k-2}$.

Suppose there is a common intermediate node from row 0 with $Perm_{0,1,k-2}$ and row 1 with $Perm_{1,1,k-2}$.

a_0 must be included in the node numbers of the common intermediate node because row 0 starts with a_0.

a_0 can not be included in the node numbers of the common intermediate node because row 1 ends with a_0.

This is a contradiction. This completes the proof of (i).

ii) There is no common intermediate node from row 1 with $Perm_{1,1,k-2}$ and row i, where $2 \leq i \leq k-1$.

Suppose there is a common node from row 1 and row i, where 2

$\leq i \leq k-1$.

a_1 must be included in the node numbers of the common intermediate node because row 1 starts with a_1.

For row i, where $2 \leq i \leq k-1$, a_0 must be included in the node numbers of the common node, because a_0 always appears right before a_1.

But a_0 can not be included in the node numbers of the common node because row 1 ends with a_0.

This is a contradiction. This completes the proof of (ii).

Since a SPGM can start with any row from the algorithm of the SPGM construction, this is true for all other rows. This completes the proof.

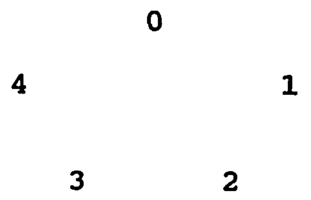
∎

This Lemma means that a SPGM with any permutations with all elements except the first and the last of any two adjacent rows in the SPGM is a PGM.

From this Lemma, $[\ (k-2)!\ ]^2$ different PGM's can be derived from a k x k SPGM.

example) For $n = 5$, $k = n$ and $N_k = \{\ 0,\ 1,\ 2,\ 3,\ 4\ \}$

Selected Sequence is 0, 1, 2, 3, 4.



$$\text{SPGM} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & \underline{3} & \underline{4} & \underline{0} & 1 \\ 3 & \underline{4} & \underline{0} & \underline{1} & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

PGM's can be obtained by permuting the underlined parts. One of them and Routing Matrix are shown below.

$$\text{PGM} = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & \underline{4} & \underline{0} & \underline{3} & 1 \\ 3 & \underline{1} & \underline{0} & \underline{4} & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

$$\text{RM} = \begin{bmatrix} (0) & (0,1) & (0,1,2) & (0,1,2,3) & (0,1,2,3,4) \\ (1) & (1,2) & (1,2,3) & (1,2,3,4) & (1,2,3,4,0) \\ \underline{(2)} & \underline{(2,4)} & \underline{(2,4,0)} & \underline{(2,4,0,3)} & \underline{(2,4,0,3,1)} \\ \underline{(3)} & \underline{(3,1)} & \underline{(3,1,0)} & \underline{(3,1,0,4)} & \underline{(3,1,0,4,2)} \\ (4) & (4,0) & (4,0,1) & (4,0,1,2) & (4,0,1,2,3) \end{bmatrix}$$

In this example, $[\ (5-2)!\ ]^2$ or 36 different PGM's can be derived from the given SPGM.

Since row 0 and row k-1 are considered being adjacent, the matrix with the permutation of the underlined parts in the following matrix is a PGM.

$$\text{SPGM} \quad = \quad \begin{bmatrix} 0 & \underline{1 \quad 2 \quad 3} & 4 \\ 1 & 2 \quad 3 \quad 4 & 0 \\ 2 & 3 \quad 4 \quad 0 & 1 \\ 3 & 4 \quad 0 \quad 1 & 2 \\ 4 & \underline{0 \quad 1 \quad 2} & 3 \end{bmatrix}$$

We also improve the previous Lemma as follows.

<u>Lemma 3.</u> Let M be a SPGM for k-subcube.

$$M \quad = \quad \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{k-2} & a_{k-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{k-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{k-1} & a_0 & a_1 & \cdots\cdots & a_{k-3} & a_{k-2} \end{bmatrix}$$

The matrix with

$\text{Perm}_{0,1,k-2}$

$\text{Perm}_{i,1,k-i-1}$

, for all i where $1 \le i \le k-1$, is a PGM for the node disjoint paths of length k.

Proof) We prove this Lemma by showing that the matrix with $\text{Perm}_{0,1,k-2}$ and $\text{Perm}_{i,1,k-i-1}$, where $1 \le i \le k-1$, generates Routing matrix where all intermediate nodes are distinct.

We prove this Lemma by showing that each row with the given permutation doesn't generate a common intermediate node from all lower rows with the given permutation.

Let's consider row 0 and all lower rows with the given permutation.

Suppose there is a common intermediate node from row 0 with $\text{Perm}_{0,1,k-2}$ and all lower rows with $\text{Perm}_{i,1,k-i-1}$, where $1 \le i \le k-1$.

Since row 0 starts with a_0, a_0 must be included in the node numbers of the common intermediate node.

Since a_{k-1} always appears before a_0 in the all lower rows with the given permutation, a_{k-1} must be included in the node numbers of the common intermediate node.

Since row 0 ends with a_{k-1}, a_{k-1} can not be included in the node numbers of the common intermediate node.

This is a contradiction. Thus there is no common intermediate node between row 0 and any one of lower rows with the given permutations.

Let's consider row i and all lower rows with the given permutation, where $1 \leq i \leq k-1$.

Suppose there is a common intermediate node from row i and all lower rows with the given permutation.

Since row i starts with a_i, a_i must be included in the node numbers of the common intermediate node.

Since a_{i-1} always appears right before a_i in all lower rows with the given permutation, a_{i-1} must be included in the node numbers of the common intermediate node.

Since row i ends with a_{i-1}, a_{i-1} can not be included in the node numbers of the common intermediate node.

This is a contradiction. Thus, there is no common intermediate node with the given permutation from row i and any one of its lower rows.

This completes the proof.

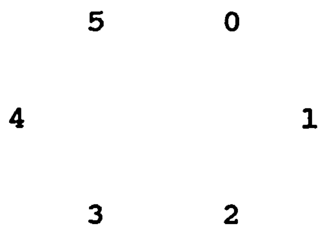
■

From this Lemma, the number of different PGM's derived from a k x k SPGM is

$$(k - 2)! \cdot \overset{k-1}{\underset{i=2}{\pi}} (k - i)!$$

example) For n = 6, k = n and N_k = { 0, 1, 2, 3, 4, 5 }

Selected Sequence is 0, 1, 2, 3, 4, 5.



$$\text{SPGM} = \begin{bmatrix} 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\ 1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & 0 \\ 2 & \underline{3} & \underline{4} & \underline{5} & 0 & 1 \\ 3 & \underline{4} & \underline{5} & 0 & 1 & 2 \\ 4 & \underline{5} & 0 & 1 & 2 & 3 \\ 5 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

PGM's can be obtained by permuting the underlined parts. One of them and Routing Matrix are shown below.

$$\text{PGM} = \begin{bmatrix} 0 & \underline{2} & \underline{3} & \underline{4} & \underline{1} & 5 \\ 1 & \underline{5} & \underline{4} & \underline{3} & \underline{2} & 0 \\ 2 & \underline{5} & \underline{3} & \underline{4} & 0 & 1 \\ 3 & \underline{5} & \underline{4} & 0 & 1 & 2 \\ 4 & \underline{5} & 0 & 1 & 2 & 3 \\ 5 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,2) & (0,2,3) & (0,2,3,4) & (0,2,3,4,1) & (0,2,3,4,1,5) \\ (1) & (1,5) & (1,5,4) & (1,5,4,3) & (1,5,4,3,2) & (1,5,4,3,2,0) \\ (2) & (2,5) & (2,5,3) & (2,5,3,4) & (2,5,3,4,0) & (2,5,3,4,0,1) \\ (3) & (3,5) & (3,5,4) & (3,5,4,0) & (3,5,4,0,1) & (3,5,4,0,1,2) \\ (4) & (4,5) & (4,5,0) & (4,5,0,1) & (4,5,0,1,2) & (4,5,0,1,2,3) \\ (5) & (5,0) & (5,0,1) & (5,0,1,2) & (5,0,1,2,3) & (5,0,1,2,3,4) \end{bmatrix}$$

In this example, $4! \cdot 4! \cdot 3! \cdot 2!$ or 6912 different PGM's can be derived.

Since SPGM can start with any row from the algorithm of SPGM construction, row 0 and row k-1 are considered being adjacent.

Therefore the matrix with the permutations of the underlined parts in the following matrix is also a PGM.

$$SPGM = \begin{bmatrix} 0 & \underline{1} & \underline{2} & 3 & 4 & 5 \\ 1 & \underline{2} & 3 & 4 & 5 & 0 \\ 2 & 3 & 4 & 5 & 0 & 1 \\ 3 & \underline{4} & \underline{5} & \underline{0} & \underline{1} & 2 \\ 4 & \underline{5} & \underline{0} & \underline{1} & \underline{2} & 3 \\ 5 & \underline{0} & \underline{1} & \underline{2} & 3 & 4 \end{bmatrix}$$

example) for $n = 10$, $k = n$ and $N_k = \{0,1,2,3,4,5,6,7,8,9\}$

$$\begin{bmatrix} 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 \\ 1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 \\ 2 & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & 1 \\ 3 & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & 1 & 2 \\ 4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & 1 & 2 & 3 \\ 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & 1 & 2 & 3 & 4 \\ 6 & \underline{7} & \underline{8} & \underline{9} & 0 & 1 & 2 & 3 & 4 & 5 \\ 7 & \underline{8} & \underline{9} & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 8 & \underline{9} & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{bmatrix} \begin{bmatrix} 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & 6 & 7 & 8 & 9 \\ 1 & \underline{2} & \underline{3} & \underline{4} & 5 & 6 & 7 & 8 & 9 & 0 \\ 2 & \underline{3} & \underline{4} & 5 & 6 & 7 & 8 & 9 & 0 & 1 \\ 3 & \underline{4} & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 \\ 4 & 5 & 6 & 7 & 8 & 9 & 0 & 1 & 2 & 3 \\ 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 \\ 6 & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\ 7 & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & 6 \\ 8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & 6 & 7 \\ 9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & 6 & 7 & 8 \end{bmatrix}$$

We again improve the previous Lemma.

Theorem 3.2 Let M be a SPGM for k-subcube.

$$M = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{k-2} & a_{k-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{k-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{k-1} & a_0 & a_1 & \cdots\cdots & a_{k-3} & a_{k-2} \end{bmatrix}$$

The matrix with

$Perm_{0,1,k-2}$,

$Perm_{i,1,k-i-1}$

$Perm_{i,k-i+1,k-2}$

, for all i where $1 \le i \le k-1$, is a PGM for the node disjoint paths of length k.

Proof) We have to show that the matrix with $Perm_{0,1,k-2}$, $Perm_{i,1,k-i-1}$ and $Perm_{i,k-i+1,k-2}$, where $1 \le i \le k-1$, generates routing matrix where all intermediate nodes are distinct.

We prove this theorem by showing that each row with the given permutation doesn't generate a common intermediate node from all lower rows with the given permutations.

Let's consider row 0 and all lower rows with the given permutations.

Suppose there is a common intermediate node from row 0 and all lower rows with the given permutations.

Since row 0 starts with a_0, a_0 must be included in the node numbers of the common intermediate node.

Since a_{k-1} always appears before a_0 in the all lower rows with the given permutations, a_{k-1} must be included in the node numbers of the common intermediate node.

Since row 0 ends with a_{k-1}, a_{k-1} can not be included in the node numbers of the common intermediate node.

This is a contradiction. Thus, there is no common intermediate node from row 0 and any one of lower rows with the given permutations.

Let's consider row i and all lower rows, say row j , with the given permutations, where $1 \le i \le k-1$ and $i < j \le k-1$.

Suppose there is a common intermediate node from row i and row j, where $i < j \leq k-1$, with the given permutations.

Since row i starts with a_i, a_i must be included in the node numbers of the common intermediate node.

Since a_0 always appears before a_i in row j with the given permutations, a_0 must be included in the node numbers of the common intermediate node.

The column number of a_0 at every row is (k-i) mod k. This number also says what numbers appear before a_0. For row i, numbers, a_i, a_{i+1}, a_{i+2}, , , a_{k-1}, appear before a_0.

Since a_0 is included in the node numbers of the common intermediate node, a_i through a_{k-1} in row i must be included.

Since row j ends with one of numbers, a_i through a_{k-1}, it is impossible that row j has the common intermediate node with row i.

This completes the proof.

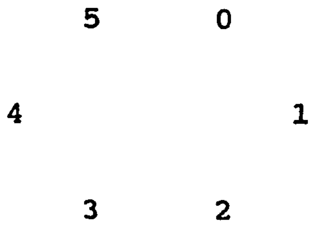
∎

From this theorem, the number of different PGM's derived from a k x k SPGM is

$$\left[\prod_{i=2}^{k-1} (k - i)! \right]^2$$

example) For n = 6, k = n and N_k = { 0, 1, 2, 3, 4, 5 }

Selected Sequence is 0, 1, 2, 3, 4, 5.



$$\text{SPGM} = \begin{bmatrix} 0 & \underline{1 \quad 2 \quad 3 \quad 4} & & & & 5 \\ 1 & \underline{2 \quad 3 \quad 4 \quad 5} & & & & 0 \\ 2 & \underline{3 \quad 4 \quad 5} & & & 0 & 1 \\ 3 & \underline{4 \quad 5} & & 0 & \underline{1} & 2 \\ 4 & \underline{5} & 0 & \underline{1 \quad 2} & & 3 \\ 5 & 0 & \underline{1 \quad 2 \quad 3} & & & 4 \end{bmatrix}$$

PGM's can be obtained by permuting the underlined parts. One of them and Routing Matrix are shown below.

$$PGM = \begin{bmatrix} 0 & \underline{2} & \underline{3} & \underline{4} & \underline{1} & 5 \\ 1 & \underline{5} & \underline{4} & \underline{3} & \underline{2} & 0 \\ 2 & \underline{5} & \underline{3} & \underline{4} & 0 & 1 \\ 3 & \underline{5} & \underline{4} & 0 & \underline{1} & 2 \\ 4 & \underline{5} & 0 & \underline{2} & \underline{1} & 3 \\ 5 & 0 & \underline{3} & \underline{1} & \underline{2} & 4 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,2) & (0,2,3) & (0,2,3,4) & (0,2,3,4,1) & (0,2,3,4,1,5) \\ (1) & (1,5) & (1,5,4) & (1,5,4,3) & (1,5,4,3,2) & (1,5,4,3,2,0) \\ (2) & (2,5) & (2,5,3) & (2,5,3,4) & (2,5,3,4,0) & (2,5,3,4,0,1) \\ (3) & (3,5) & (3,5,4) & (3,5,4,0) & (3,5,4,0,1) & (3,5,4,0,1,2) \\ (4) & (4,5) & (4,5,0) & (4,5,0,2) & (4,5,0,2,1) & (4,5,0,2,1,3) \\ (5) & (5,0) & (5,0,3) & (5,0,3,1) & (5,0,3,1,2) & (5,0,3,1,2,4) \end{bmatrix}.$$

In this example, $[4!\cdot 3!\cdot 2!]^2$ or 82944 different PGM's can be derived.

Since a SPGM can start with any row from the algorithm of the SPGM construction, row 0 and row k-1 are considered being adjacent. Therefore the matrix with the permutations of the underlined parts in the following matrix is also a PGM.

$$\text{SPGM} = \begin{bmatrix} 0 & \underline{1} & \underline{2} & 3 & \underline{4} & 5 \\ 1 & \underline{2} & 3 & \underline{4} & \underline{5} & 0 \\ 2 & 3 & \underline{4} & \underline{5} & \underline{0} & 1 \\ 3 & \underline{4} & \underline{5} & \underline{0} & \underline{1} & 2 \\ 4 & \underline{5} & \underline{0} & \underline{1} & \underline{2} & 3 \\ 5 & \underline{0} & \underline{1} & \underline{2} & 3 & 4 \end{bmatrix}$$

example) for n = 10, k = n and $N_k = \{0,1,2,3,4,5,6,7,8,9\}$

$$\begin{bmatrix} 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 \\ 1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 \\ 2 & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & 1 \\ 3 & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & 2 \\ 4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & 3 \\ 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & 4 \\ 6 & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\ 7 & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & 6 \\ 8 & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & 7 \\ 9 & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8 \end{bmatrix} \begin{bmatrix} 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & \underline{6} & \underline{7} & \underline{8} & 9 \\ 1 & \underline{2} & \underline{3} & \underline{4} & 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 \\ 2 & \underline{3} & \underline{4} & 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & 1 \\ 3 & \underline{4} & 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & 2 \\ 4 & 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & 3 \\ 5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 \\ 6 & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\ 7 & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & 6 \\ 8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & \underline{6} & 7 \\ 9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 & \underline{6} & \underline{7} & 8 \end{bmatrix}$$

$$\begin{bmatrix}
0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 \\
1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 & 0 \\
2 & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 & \underline{0} & 1 \\
3 & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 & \underline{0} & \underline{1} & 2 \\
4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 & \underline{0} & \underline{1} & \underline{2} & 3 \\
5 & \underline{6} & \underline{7} & \underline{8} & 9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 \\
6 & \underline{7} & \underline{8} & 9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\
7 & \underline{8} & 9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & 6 \\
8 & 9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & 7 \\
9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8
\end{bmatrix}
\begin{bmatrix}
0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8 & 9 \\
1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8 & \underline{9} & 0 \\
2 & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8 & \underline{9} & \underline{0} & 1 \\
3 & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8 & \underline{9} & \underline{0} & \underline{1} & 2 \\
4 & \underline{5} & \underline{6} & \underline{7} & 8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & 3 \\
5 & \underline{6} & \underline{7} & 8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 \\
6 & \underline{7} & 8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\
7 & 8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & 6 \\
8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & 7 \\
9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8
\end{bmatrix}$$

This theorem can be restated as follows.

From a SPGM for k-subcube, a k x k matrix with any permutations of partitions which are partitioned by column 0, column k-1 and a_i's at each row is a PGM.

So far we have considered SPGM's and a PGM's obtained from a SPGM. Now we examine whether a SPGM and a RM obtained from the SPGM can have every node as an intermediate node for k node disjoint paths of length k in k-subcube.

We take a sequence 0,1,2,3,4 for n = 5 and k = n as an example.

$$\text{SPGM} \quad = \quad \begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 0 \\ 2 & 3 & 4 & 0 & 1 \\ 3 & 4 & 0 & 1 & 2 \\ 4 & 0 & 1 & 2 & 3 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,1) & (0,1,2) & (0,1,2,3) & (0,1,2,3,4) \\ (1) & (1,2) & (1,2,3) & (1,2,3,4) & (1,2,3,4,0) \\ (2) & (2,3) & (2,3,4) & (2,3,4,0) & (2,3,4,0,1) \\ (3) & (3,4) & (3,4,0) & (3,4,0,1) & (3,4,0,1,2) \\ (4) & (4,0) & (4,0,1) & (4,0,1,2) & (4,0,1,2,3) \end{bmatrix}$$

In a Routing Matrix derived from a SPGM, only the nodes whose node numbers are adjacent in the selected sequence are used for node disjoint paths. In other words, the nodes whose node numbers are not adjacent in the selected sequence are never used for node disjoint paths.

example) For the above example, (0,2,3) is never used because node numbers 0,2 and 3 are not adjacent in the selected sequence.

We take a Routing Matrix derived from a SPGM by Lemma 3 as another example.

$$\text{SPGM} = \begin{bmatrix} 0 & \underline{2} & \underline{3} & \underline{4} & \underline{1} & 5 \\ 1 & \underline{5} & \underline{4} & \underline{3} & \underline{2} & 0 \\ 2 & \underline{5} & \underline{3} & \underline{4} & 0 & 1 \\ 3 & \underline{5} & \underline{4} & 0 & 1 & 2 \\ 4 & \underline{5} & 0 & 1 & 2 & 3 \\ 5 & 0 & 1 & 2 & 3 & 4 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,2) & (0,2,3) & (0,2,3,4) & (0,2,3,4,1) & (0,2,3,4,1,5) \\ (1) & (1,5) & (1,5,4) & (1,5,4,3) & (1,5,4,3,2) & (1,5,4,3,2,0) \\ (2) & (2,5) & (2,5,3) & (2,5,3,4) & (2,5,3,4,0) & (2,5,3,4,0,1) \\ (3) & (3,5) & (3,5,4) & (3,5,4,0) & (3,5,4,0,1) & (3,5,4,0,1,2) \\ (4) & (4,5) & (4,5,0) & (4,5,0,1) & (4,5,0,1,2) & (4,5,0,1,2,3) \\ (5) & (5,0) & (5,0,1) & (5,0,1,2) & (5,0,1,2,3) & (5,0,1,2,3,4) \end{bmatrix}.$$

Some nodes can not be used in any PGM's derived from Lemma 2. For example node (0,2,5[illegible] can not appear in any paths.

<u>Theorem 3.3</u> Let M be a k x k PGM which is derived from a k x k SPGM with

$\text{Perm}_{0,1,k-2}$

$\text{Perm}_{i,1,k-i-1}$

$\text{Perm}_{i,k-i+1,k-2}$, for all i where $1 \le i \le k-1$.

i) Every node in the k-subcube except the source and the destination nodes can be used by PGM.

ii) Given PGM, every node in the k-subcube except the source and the destination nodes can appear in one path only.

Proof) We prove this theorem by showing that the node numbers of an arbitrary node except the source and the destination can appear at only one row of the Routing Matrix derived from M.

Since all k entries at column 0 of M are distinct, all k nodes of weight 1 appear at the row where node number is same as the entry at column 0.

Since all k entries at column k-1 of M are distinct, all nodes of weight k-1 are distinct and appear in only one row.

Now we consider an arbitrary node whose weight is greater than 1 and smaller than k-1.

$$M = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{k-2} & a_{k-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{k-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{k-1} & a_0 & a_1 & \cdots\cdots & a_{k-3} & a_{k-2} \end{bmatrix}$$

All nodes can be divided into three groups.

i) a group of nodes where a_0 is not included in their node numbers.

ii) a group of nodes where a_0 is included and a_{k-1} is not included in their node numbers.

iii) a group of nodes where a_0 is included and a_{k-1} is included in their node numbers.

We show that a node in each group can appear at only one row of M.

Let $(b_1, b_2, b_3 \ldots b_q)$ represent an arbitrary node u, where $2 \leq q \leq k-2$ and b_i appears before b_{i+1} in the row 0 of the original SPGM.

i) A node, u, in the group where a_0 is not included in their node numbers appears at one row of M only.

Since row 0 starts with a_0 and a_0 is not included in the node numbers of u, node u cannot come from row 0.

Rows which start with b_i, $1 \le i \le q$, are candidates so that node u can appear in the node disjoint paths.

We declare that node u appears at the row which starts with b_1, the leftmost node number of the node u..

Suppose node u appears at the row starting with b_m where there exists b_n such that $n < m$.

a_0 divides all entries in row i of M, $1 \le i \le k-1$, into two sets, $S_{i,l}$ and $S_{i,r}$, where $S_{i,l}$ contains all node numbers appearing before a_0 and $S_{i,r}$ contains all node numbers appearing after a_0.

For row starting with b_m which is same as a_t, $S_{i,l}$ contains a_t through a_{k-1} and $S_{i,r}$ contains a_1 through a_{t-1}.

Since $n < m$, b_n must be included in $S_{i,r}$.

Since b_n must be included in the node numbers of node u and a_0

must appear before b_n in row starting with b_m, a_0 must be included in the node numbers of node u.

This is a contradiction. This completes the proof of i).

ii) A node, u, in the group where a_0 is included and a_{k-1} is not included in their node numbers appears at one row of M only.

We declare that these nodes appear at row 0.

Since row 0 ends with a_{k-1} and begins with a_0, a_0 must be included and a_{k-1} must not included in the node numbers of node u.

Thus, the node, u, can come from row 0 with the given permutation.

Now we show that a node with the given condition can not come from row j, where $1 \le j \le k-1$.

From the construction of the matrix and the given permutations, a_{k-1} must appear before a_0 for row j, $1 \le j \le k-1$.

Thus, if a_0 is included in the node numbers of node u, then a_{k-1} must be included.

This shows that a node with the given condition can not come from row j, where $1 \le j \le k-1$.

This completes the proof of ii).

iii) Nodes in the group where a_0 is included a_{k-1} is also included in their node numbers appears at one row of M only.

a_{k-1} cannot be included in the number of intermediate nodes which appears in the row 0, because row 0 ends with a_{k-1}.

a_0 cannot be included in the number of intermediate nodes which appears in the row 1, because row 1 ends with a_0.

Now, we show that a node u with the given condition can come from only one row i, where $2 \le i \le k-1$.

All node numbers of node u can be divided into two sets, u_a and u_b, where u_a contains a_0 and the node numbers whose subscripts of a are decreasingly consecutive from k-1 and u_b contains the rest of the node numbers.

example) for n = 10 and k = n

$u = (a_0, a_1, a_3, a_6, a_8, a_9)$

$u_a = \{ a_0, a_9, a_8 \}$

$u_b = \{ a_1, a_3, a_6 \}$

$u = (a_0, a_1, a_3, a_6, a_7, a_8, a_9)$

$u_a = \{ a_0, a_9, a_8, a_7, a_6 \}$

$u_b = \{ a_1, a_3 \}$

$u = (a_0, a_9)$

$u_a = \{ a_0, a_9 \}$

$u_b = \{ \}$

From the construction of a SPGM and the given permutation, a_i, a_{i+1}, . . . , a_{k-1} always appear before a_0 and a_1, a_2, . . . a_{i-1} always appear after a_0 at row i, where $2 \le i \le k-1$.

All entries at row i of M, $2 \le i \le k-1$, can be divided into two groups, $S'_{i,l}$ and $S'_{i,r}$, where $S'_{i,l}$ contains a_0 and all entries appearing before a_0 and $S'_{i,r}$ contains the rest of entries appearing after a_0 with the given permutations.

From the construction of a SPGM and the given permutations, there is only one row where $S'_{i,l} = u_a$ and $2 \le i \le k-1$.

We have to show that node u can come from row i where $S'_{i,l} = u_a$ and node u can't come from row j where $S'_{j,l} \neq u_a$, $2 \leq i,j \leq k-1$.

First we show that a node u can come from row i where $S'_{i,l} = u_a$ and $2 \leq i \leq k-1$.

Let's examine row i where $S'_{i,l} = u_a$ and $2 \leq i \leq k-1$.

Node numbers of node u consist of two sets, u_a and u_b.

Since row i ends with a_{i-1}, all intermediate nodes from row i cannot have node number a_{i-1}.

Since $u_a = S'_{i,l}$, a_{i-1} is not included in node numbers of node u from the construction of the set u_a

Thus, u_b is a subset of $S'_{l,r} - a_{i-1}$.

The given permutation $Perm_{i,k-i+1,k-2}$ represents the permutation with elements of the set $S'_{l,r} - a_{i-1}$.

By the given permutation $Perm_{i,k-i+1,k-2}$, all elements of u_b can appear before all other elements in $S'_{i,r} - a_{i-1}$.

Since all elements of u_a and u_b represent the node numbers of the node u, the node u can be derived from row i as an intermediate node.

We also show that a node u can not come from row i where $S'_{i,l} \neq u_a$ and $2 \leq i \leq k-1$.

We show this in two different cases.

1) A node u can't come from row j, where $S'_{j,l} \neq u_a$ and $2 \leq j \leq i-1$.
2) A node u can't come from row j', where $S'_{j',l} \neq u_a$ and $i+1 \leq j' \leq k-1$.

1) All elements of $S'_{j,l}$ must appear in the node numbers of any intermediate node with the given permutation because a_0 must be included.

Since $| S'_{j,l} | > | u_a |$, there must be at least one number which is not the node number of node u from the construction of $S'_{j,i}$ and u_a.

Therefore, node u can't come from row j, where $S'_{j,l} \neq u_a$ and $2 \leq j \leq i-1$.

2) Each row j', where $S'_{j',l} \neq u_a$ and $i+1 \leq j' \leq k-1$, ends

with one of $a_i, a_{i+1}, . . . , a_{k-2}$.

In other words, row j', $i+1 \leq j' \leq k-1$, can not produce an intermediate node whose node numbers include $a_i, a_{i+1}, , , a_{k-2}$ because each row ends with one of them.

But $a_i, a_{i+1}, , , a_{k-2}$ must be included in the node numbers of node u from the construction of u_a.

This is a contradiction.

Thus, node u can't come from row j', where $S'_{j',l} \neq u_a$ and $i+1 \leq j' \leq k-1$.

This completes the proof of iii).

∎

example) From the following matrix with permutation of the underlined parts, We show that any node can be derived from only one row, where $n = 10$ and $k = n$.

$$\begin{bmatrix}
0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 \\
1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 \\
2 & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & 1 \\
3 & \underline{4} & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & 2 \\
4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & 3 \\
5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & 4 \\
6 & \underline{7} & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\
7 & \underline{8} & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & 6 \\
8 & \underline{9} & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & 7 \\
9 & 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & \underline{5} & \underline{6} & \underline{7} & 8
\end{bmatrix}$$

(0,9) is from row 9.
(3,8) is from row 3.
(0,4,7,8) is from row 0.
(2,4,6,8,9) is from row 2.
(0,3,6,7,8,9) is from row 6.
(0,1,2,3,4,9) is from row 9.

Although we change the permutation areas, every node can also come from only one row.

$$\begin{bmatrix}
0 & \underline{1} & \underline{2} & \underline{3} & 4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & 9 \\
1 & \underline{2} & \underline{3} & 4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & 0 \\
2 & \underline{3} & 4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & 1 \\
3 & 4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & 2 \\
4 & \underline{5} & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & 3 \\
5 & \underline{6} & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 \\
6 & \underline{7} & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 & 5 \\
7 & \underline{8} & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 & \underline{5} & 6 \\
8 & \underline{9} & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 & \underline{5} & \underline{6} & 7 \\
9 & \underline{0} & \underline{1} & \underline{2} & \underline{3} & 4 & \underline{5} & \underline{6} & \underline{7} & 8
\end{bmatrix}$$

(0,9) is from row 9.
(3,8) is from row 8.
(0,4,7,8) is from row 4.
(2,4,6,8,9) is from row 4.
(0,3,6,7,8,9) is from row 6.
(0,1,2,3,4,9) is from row 9.

So far we have studied how to derive PGM's from a SPGM with proper permutations. These PGM's can be used for fault tolerant node disjoint routing in an injured hypercube.

Again, one property of a hypercube is that there are k different paths of length k between two arbitrary nodes of Hamming distance k. Thus, there are two different paths,p_1 and p_2, of length two between two arbitrary nodes,u and v, where they are not in the same stage and their Hamming distance is two. Since the length of the paths is two, there is only one intermediate node between u and v. Let's say that p_1 and p_2 visit intermediate nodes x and y respectively.

example) For five dimensional hypercube, there are two paths of length 2 between 00110 and 01111.

u = 00110
v = 01111
p_1 00110 - 00111 - 01111
p_2 00110 - 01110 - 01111

These two paths, p_1 and p_2, cannot be used at same time in node disjoint routing between 00000 and 11111 because nodes u and v are common for p_1 and p_2. But one path can replace the other in case of the injury that either x or y is injured. Clearly, if both x and y are injured, there is no path of length two between u and v. This property can be used in fault tolerant node disjoint routing in an injured hypercube. We study how this property can be shown in a PGM which is derived from a SPGM. Let's take the previous example again with different notation.

u = (1,2)
v = (0,1,2,3)

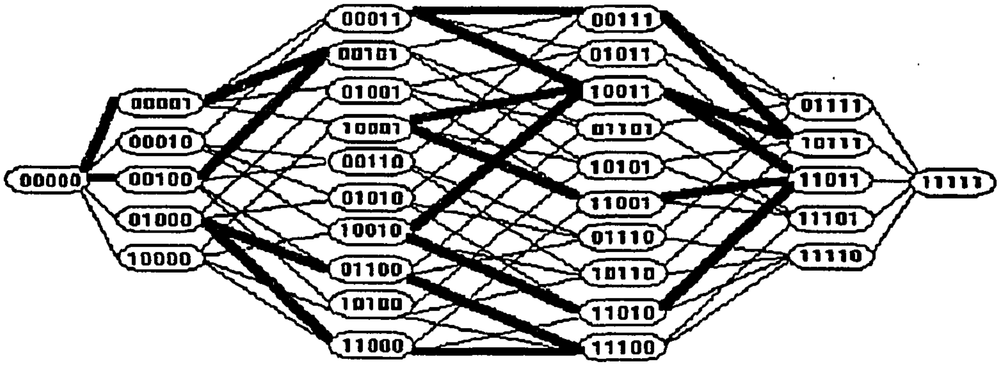
p_1 (1,2) - (0,1,2) - (0,1,2,3)
p_2 (1,2) - (1,2,3) - (0,1,2,3)

Nodes (0,1,2) and (1,2,3) are intermediate for p_1 and p_2 respectively. Suppose there is a row (1 2 0 3 4) of a SPGM for five dimensional hypercube. the row (1 2 0 3 4) generates one node disjoint path, () - (1) - (1,2) - (1,2,0) - (1,2,0,3) - (1,2,0,3,4). p_1 appears in this path because (1,2,0) and (0,1,2) represent the same node 00111 and (0,1,2,3) and (1,2,0,3) represent the same node 01111. Suppose node (0,1,2) is injured. Then p_1 can be replaced with p_2. Thus, new path is () - (1) - (1,2) - (1,2,3) - (1,2,0,3) - (1,2,0,3,4). This path can be generated by a row (1 2 3 0 4) of a PGM. Now let's compare the two rows.

row1 $(1\ 2\ \underline{0\ 3}\ 4)$
row2 $(1\ 2\ \underline{3\ 0}\ 4)$

Row2 can be obtained by swapping the underlined part of row1. As discussed so far, replacement for injured nodes can be examined by swapping two adjacent entries in a row of a SPGM with a certain restriction. Without any restriction, a swap does not guarantee the node disjoint routing. See fig. 3-4. In fig. 3-4, node 10011 can replace three paths.

fig.3-4



This restriction has been shown in theorem 3.2. The given permutations can be achieved with a series of swaps of two adjacent elements of the row. Suppose a SPGM has a row (0 1 2 3 4 5 6) with a permutation of underlined part in seven dimensional hypercube. (0 2 4 1 5 3 6) can be obtained by a certain number of swaps (indicated below by the underlined) and it is shown that each swap represents a replacement.

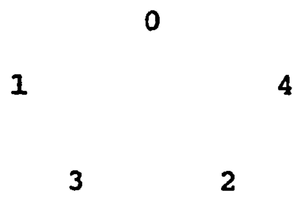
$(0\ \underline{1\ 2}\ 3\ 4\ 5\ 6)$	no replacement
$(0\ 2\ 1\ \underline{3\ 4}\ 5\ 6)$	(0,1) has been replaced with (0,2).
$(0\ 2\ \underline{1\ 4}\ 3\ 5\ 6)$	(0,2,1,3) has been replaced with (0,2,1,4).
$(0\ 2\ 4\ 1\ \underline{3\ 5}\ 6)$	(0,2,1) has been replaced with (0,2,4).
$(0\ 2\ 4\ 1\ 5\ 3\ 6)$	(0,2,4,1,3) has been replaced with (0,2,4,1,5).

Furthermore, with this method, theorem 3.3 guarantees that a node substituting another injured node never appears in other paths. This algorithm gives much more flexibility than the simple construction of a SPGM discussed in section B.

Suppose nodes (0,1), (0,2) and (0,3) are injured in five dimensional hypercube. In this case, it is impossible to select a proper sequence with numbers 0,1,2,3 and 4 where the node numbers of injured nodes are not adjacent. Thus, we cannot produce five node disjoint paths with the SPGM algorithm. But we can have five node disjoint paths by modifying a SPGM with proper permutations as follows.

First of all, we select a sequence of the elements of N_k avoiding the adjacency of node numbers of injured nodes as much as possible.

Selected sequence is 0, 4, 2, 3, 1.



$$\text{SPGM} = \begin{bmatrix} 0 & 4 & 2 & 3 & 1 \\ 4 & 2 & 3 & 1 & 0 \\ 2 & 3 & 1 & 0 & 4 \\ 3 & 1 & 0 & 4 & 2 \\ 1 & \underline{0 \quad 4} & & 2 & 3 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,4) & (0,4,2) & (0,4,2,3) & (0,4,2,3,1) \\ (4) & (4,2) & (4,2,3) & (4,2,3,1) & (4,2,3,4,0) \\ (2) & (2,3) & (2,3,1) & (2,3,1,0) & (2,3,1,0,4) \\ (3) & (3,1) & (3,1,0) & (3,1,0,4) & (3,1,0,1,2) \\ (1) & \underline{(1,0)} & (1,0,4) & (1,0,4,2) & (1,0,4,2,3) \end{bmatrix}$$

In this Routing Matrix, injured node (0,1) still appears. But we can avoid this injured node by swapping these two underlined numbers in the previous SPGM.

$$\text{SPGM} = \begin{bmatrix} 0 & 4 & 2 & 3 & 1 \\ 4 & 2 & 3 & 1 & 0 \\ 2 & 3 & 1 & 0 & 4 \\ 3 & 1 & 0 & 4 & 2 \\ 1 & \underline{4 \quad 0} & & 2 & 3 \end{bmatrix}$$

RM

$$\begin{bmatrix} (0) & (0,4) & (0,4,2) & (0,4,2,3) & (0,4,2,3,1) \\ (4) & (4,2) & (4,2,3) & (4,2,3,1) & (4,2,3,4,0) \\ (2) & (2,3) & (2,3,1) & (2,3,1,0) & (2,3,1,0,4) \\ (3) & (3,1) & (3,1,0) & (3,1,0,4) & (3,1,0,1,2) \\ (1) & \underline{(1,4)} & (1,4,0) & (1,4,0,2) & (1,4,0,2,3) \end{bmatrix}$$

In this Routing Matrix, injured node (0,1) doesn't appears. Now we have five node disjoint paths in the presence of three injured nodes. We studied how to avoid an injured edge in section B using SPGM. PGM's derived from a SPGM also can be used for an edge injured hypercube with much more flexibility.

In section B, we studied the injured edge and how to avoid the condition of injured edges using SPGM method. With the PGM derived from a SPGM, different edges used in node disjoint paths also can be selected by using different permutations. Therefore, we can use the PGM derived from a SPGM for fault tolerant routing for the node disjoint paths of length k in k-subcube in the presence of node and/or edge injury. Steps for fault tolerant routing in k-subcube are shown below.

step1 Select a proper sequence of the elements of N_k to avoid all injury conditions as much as possible.

step2 Construct a SPGM with the selected sequence.

step3 Construct a PGM with the given permutation, shown in theorem 3.2, avoiding additional injury condition.

IV. Fault Tolerant Routing for the paths of length k+2.

In chapter III, we studied fault tolerant routing methods for the node disjoint paths of length k between two arbitrary nodes of Hamming distance k in n-cube using k x k PGM's. In this chapter, we study a fault tolerant routing method for the paths of length k+2 between two arbitrary nodes of Hamming distance k in n-cube using (n-k) x (k+2) PGM's. We already know that in addition to the k paths of length k, there are n-k node disjoint paths of length k+2 between two arbitrary nodes of Hamming distance k in n-cube. In this chapter, we also introduce a method to generate n-k node disjoint paths of length k+2 and show how this method can be used for fault tolerant routing for the paths of length k+2. In this chapter, by the paths we mean the node disjoint paths of length k+2 unless otherwise mentioned.

Recall the following definition.

<u>Definition)</u> N_k and N_{k+2} represent the sets of bit position numbers where the binary value of $A \oplus B$ is 1 and 0 respectively.

example) for n = 10

$A = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$B = 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1$

$$A \oplus B = 0\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 1$$

bit position $9\ 8\ 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0$

$$N_k = \{ 0, 2, 5, 6, 7, 8 \}$$

$$N_{k+2} = \{ 1, 3, 4, 9 \}$$

For node disjoint paths of length k+2 out of k-subcube, we need the elements of N_k and N_{k+2}. For the routing algorithm for the paths of length k+2, , we introduce another different notation for the address of a node in a hypercube.

<u>Definition)</u> $[a_1, a_2, a_3, , , , a_q]$,where $0 \leq q$ and $0 \leq a_i \leq n-1$, is defined as the address of the node where the value of bit position a_i is complemented with respect to the source node..

The position number of the least significant bit is 0.
The position number of the most significant bit is n-1.

example) For ten dimensional hypercube, assume that the address of the source node is 0000000000.

Node 0011001101 is denoted by [0,2,3,6,7].
Node 0001010101 is denoted by [0,2,4,6].
Node 0000000011 is denoted by [0,1] or [0,1,0,1,0,1]
Node 0000000000 is denoted by [], [1,1] or [1,1,2,2,6,6].

Node 1111111111 is denoted by [0,1,2,3,4,5,6,7,8,9].

We also call the numbers a_1 through a_q node numbers. This notation has the following properties.

property1) The address of any node can have more than n node numbers.

property2) All node numbers of a node need not be distinct.

property3) A node can be denoted by any permutation of node numbers of the node.

example) (1,1,0), (1,0,1) and (0,1,1) represent the same node 0000000001.

example) For 10-cube, we are looking for ten node disjoint paths between two nodes, 0000000000 and 0100100110.

Since Hamming distance of the two nodes is four, there are four node disjoint paths of length four and six additional node disjoint paths of length six. We can generate node disjoint paths of length four by using a SPGM or a PGM derived from a SPGM. Now we show a 6 x 6 matrix as an example.

$$\begin{bmatrix} [0] & [0,5] & [0,5,8] & [0,5,8,1] & [0,5,8,1,2] & [0,5,8,1,2,0] \\ [3] & [3,1] & [3,1,2] & [3,1,2,5] & [3,1,2,5,8] & [3,1,2,5,8,3] \\ [4] & [4,8] & [4,8,5] & [4,8,5,2] & [4,8,5,2,1] & [4,8,5,2,1,4] \\ [6] & [6,2] & [6,2,5] & [6,2,5,8] & [6,2,5,8,1] & [6,2,5,8,1,6] \\ [7] & [7,1] & [7,1\ 8] & [7,1,8,2] & [7,1,8,2,5] & [7,1,8,2,5,7] \\ [9] & [9,5] & [9,5,2] & [9,5,2,8] & [9,5,2,8,1] & [9,5,2,8,1,9] \end{bmatrix}$$

The above matrix is a RM because it satisfies all conditions for a RM. Now, we show how to construct (n-k) x (k+2) routing matrix for the node disjoint paths of length k+2 between two nodes of Hamming distance k in n-cube.

Let S_k and S_{k+2} represent the sets of the node disjoint paths of length k and k+2 between two arbitrary nodes of Hamming distance k in n-cube.

Theorem 4.1 Let A and B be two arbitrary node of H(A,B) = k in n-cube. Let W(A) = 0 and W(B) = k. Then the first intermediate nodes of the paths in S_k and S_{k+2} must be the nodes of weight one in SQ(A,B) and out of SQ(A,B) respectively.

proof) It is previously defined that N_k and N_{k+2} represent the sets of bit position numbers where the binary value of $A \oplus B$ is 1 and 0 respectively.

If a path in S_k uses a node of weight one out of SQ(A,B) meaning that bit position in one of N_{k+2} is complemented, this bit position must be complemented again later.

The reason is following.

Because the values of the positions in N_{k+2} are same in nodes A and B, we need even number of uses of elements of N_{k+2} in order to have the same value in the positions.

Therefore, once an element of N_{k+2} is used, this element must be used at least once more.

Since the addresses of two nodes A and B are different in the positions of N_k, the values of the positions indicated by the elements in N_k must be used at least one time.

As a result, the path in S_k starting with a node out of SQ(A,B) must be of length k+2. This is a contradiction.

This completes the proof of the theorem.

∎

<u>Theorem 4.2</u> A path in S_{k+2} must start and end by complementing the value of the bit position pointed by an element of N_{k+2}.

proof) From the theorem 4.1, it has been proved that a path in S_{k+2} must start by complementing the value of the bit position pointed by an element i of N_{k+2}.

We have to show that a path in S_{k+2} must end by complementing the value of the bit position pointed by the same element i of N_{k+2}.

Suppose that complementing the value of the bit position pointed by the same element i of N_{k+2} occurs not in the end.

In chapter III, it has been shown that all the nodes represented by any n-1 distinct elements of N_k are used by all paths in S_k.

Therefore, if a path in S_{k+2} starts and doesn't end by complementing the value of the bit position pointed by an element of N_{k+2}, the path and one of the path in S_k must have at least common intermediate node which breaks the existence of n node disjoint paths.

This completes the proof of the theorem.

∎

example) For 10-cube, consider ten node disjoint paths between two nodes [] and [1,2,5,8].

One of node disjoint paths of length six is

[]-[3]-[3,1]-[3,1,2]-[3,1,2,5]-[3,1,2,5,8]-[3,1,2,5,8,3].

In this path, the bit position 3 has been complemented at the first and last. Bit position 3 is an element of N_{k+2}. From the discussion in chapter III, four nodes [1,2,5], [1,2,8], [1,5,8] and [2,5,8] must be used as the last intermediate nodes for four node disjoint paths of length four in S_k in this example. Let's complement the bit position not in the end then the result is shown below.

[]-[3]-[3,1]-[3,1,3]-[3,1,3,5]-[3,1,3,5,8]-[3,1,3,5,8,2]

Clearly, this path is of length six. But let's take look at the last intermediate node [3,1,3,5,8]. This node represents the same node [1,5,8] which is the last intermediate node for one of the node disjoint paths of length four. Since node [1,5,8] is common for two paths, we can not have ten node disjoint paths in this way.

With the result of the theorem 4.2, we have the following theorem.

<u>Theorem 4.3</u> Let $N_k = \{ a_0, a_1, a_2, , , a_{k-1} \}$
and $N_{k+2} = \{ a_k, a_{k+1}, a_{k+2}, , , a_{n-1} \}$.
Let M be a (n-k) x (k+2) matrix.

$$M = \begin{bmatrix} a_k & a_0 & a_1 & \cdots\cdots & a_{k-1} & a_k \\ a_{k+1} & a_0 & a_1 & \cdots\cdots & a_{k-1} & a_{k+1} \\ a_{k+2} & a_0 & a_1 & \cdots\cdots & a_{k-1} & a_{k+2} \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdots\cdots & \cdot & \cdot \\ a_{n-1} & a_0 & a_1 & \cdots\cdots & a_{k-1} & a_{n-1} \end{bmatrix}$$

The matrix with $Perm_{i,1,k}$, for all i where $0 \le i \le n-k-1$, is a PGM of the node disjoint paths of length k+2.

proof) We can prove by showing that the matrix with the given permutations generates a Routing Matrix for n-k node disjoint paths of length k+2 where all intermediate nodes are distinct.

From the theorem 4.2, if all the complements for all paths in S_{k+2} occur for the first and last intermediate nodes, there is no common intermediate nodes with all node disjoint path of length k except the source and destination nodes.

The same algorithm is applied to construct a routing matrix from a PGM, as discussed in chapter III.

Since each entry in the column 0 are distinct and never appear in other rows, it must be contained in the node numbers in all intermediate nodes for each row.

Therefore, all the entries of the constructed routing matrix must be distinct with the given permutations except the entries in the last column which represent the destination node.

This completes the proof of the theorem.

■

From this method, $(k!)^{n-k}$ different sets of S_{k+2}.

example) For 10-cube, we are looking for six node disjoint paths of length six between 0000000000 and 0100100110.

$A = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$B = 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0$

$A \oplus B = 0\ 1\ 0\ 0\ 1\ 0\ 0\ 1\ 1\ 0$

bit position 9 8 7 6 5 4 3 2 1 0

$N_k = \{ 1, 2, 5, 8 \}$

$N_{k+2} = \{ 0, 3, 4, 6, 7, 9 \}$

We can construct a PGM and a RM with N_k and N_{k+2}.

$$
\text{PGM} = \begin{bmatrix}
0 & \underline{1\ \ 2\ \ 5\ \ 8} & 0 \\
3 & \underline{1\ \ 2\ \ 5\ \ 8} & 3 \\
4 & \underline{1\ \ 2\ \ 5\ \ 8} & 4 \\
6 & \underline{1\ \ 2\ \ 5\ \ 8} & 6 \\
7 & \underline{1\ \ 2\ \ 5\ \ 8} & 7 \\
9 & \underline{1\ \ 2\ \ 5\ \ 8} & 9
\end{bmatrix}
$$

From the theorem 4.3, we can make different PGM's with the permutations of underlined parts. One of them is shown below.

$$
\text{PGM} = \begin{bmatrix}
0 & \underline{5\ \ 8\ \ 1\ \ 2} & 0 \\
3 & \underline{1\ \ 2\ \ 5\ \ 8} & 3 \\
4 & \underline{8\ \ 5\ \ 2\ \ 1} & 4 \\
6 & \underline{2\ \ 5\ \ 8\ \ 1} & 6 \\
7 & \underline{1\ \ 8\ \ 2\ \ 5} & 7 \\
9 & \underline{5\ \ 2\ \ 8\ \ 1} & 9
\end{bmatrix}
$$

From this PGM, we have the following routing matrix which is shown in the beginning of this chapter.

RM

$$
\begin{bmatrix}
[0] & [0,5] & [0,5,8] & [0,5,8,1] & [0,5,8,1,2] & [0,5,8,1,2,0] \\
[3] & [3,1] & [3,1,2] & [3,1,2,5] & [3,1,2,5,8] & [3,1,2,5,8,3] \\
[4] & [4,8] & [4,8,5] & [4,8,5,2] & [4,8,5,2,1] & [4,8,5,2,1,4] \\
[6] & [6,2] & [6,2,5] & [6,2,5,8] & [6,2,5,8,1] & [6,2,5,8,1,6] \\
[7] & [7,1] & [7,1\ 8] & [7,1,8,2] & [7,1,8,2,5] & [7,1,8,2,5,7] \\
[9] & [9,5] & [9,5,2] & [9,5,2,8] & [9,5,2,8,1] & [9,5,2,8,1,9]
\end{bmatrix}
$$

In this example, the number of different PGM's is $(4!)^6$ or 24^6.

Definition $A_{c,i}$ is defined as the address of a node whose bit value pointed by position i is the complement of node A all others are same.

example)   Let A = 00000 in 5-cube.

Then $A_{c,0}$ is 0001 and $A_{c,3}$is 01000.

Theorem 4.4   The method mentioned in theorem 4.3 generates all possible sets of S_{k+2}.

proof)   Consider the a path of length k between two arbitrary node of Hamming distance k in k-subcube.

Suppose that the source node A has the address of k 0's and the destination node B has the address of k 1's.

The number of different path of length k between two nodes A and B is k!, because it is same as the number of different orders in which complement of each position of the address of source node

From the theorem 4.2, a path in S_{k+2} must start and end by complementing the value of the bit position pointed by an element of N_{k+2}.

As a result, every path of length k+2 must contain two intermediate nodes, $A_{c,i}$ and $B_{c,i}$, where $i \in N_{k+2}$.

Then, the Hamming distance between two nodes $A_{c,i}$ and $B_{c,i}$ is k.

Now, consider the path of length k between $A_{c,i}$ and $B_{c,i}$.

As mentioned before, there are k! different ways in order to have a path of length k between $A_{c,i}$ and $B_{c,i}$.

And this is true for all elements in N_{k+2}.

therefore, the method mentioned in theorem 4.3 generates all possible sets of S_{k+2}.

■

The result of theorem 4.3 and 4.4 is used for fault tolerant node disjoint routing for S_{k+2} in the same fashion as the PGM derived from a SPGM discussed in chapter III. Since rearrangements of the entries of (n-k) x (k+2) PGM under the given restriction change the selections of nodes and edges, we can check all possible ways to avoid all injuries.

Notice that the method discussed in theorem 4.3 can generate all possible sets of S_{k+2}. But the method derived a SPGM in chapter III does not generate all possible PGM's.

example) For n = 5 and k = n, $N_k = \{ 0, 1, 2, 3, 4 \}$

Consider the following PGM.

$$PGM = \begin{bmatrix} 0 & 1 & 4 & 2 & 3 \\ 1 & 2 & 0 & 3 & 4 \\ 2 & 3 & 1 & 4 & 0 \\ 3 & 4 & 2 & 0 & 1 \\ 4 & 0 & 3 & 1 & 2 \end{bmatrix}$$

This PGM generates five node disjoint paths of length five, can not be derived from any SPGM.

Finally, we take an example to generate ten node disjoint paths in 10-cube between two nodes 0000000000 and 0100100110 by showing two PGM with allowed permutation areas. PGM1 is for S_k and PGM2 is for S_{k+2}.

$$\mathrm{PGM1} = \begin{bmatrix} 1 & \underline{2 \quad 5} & & 8 \\ 2 & \underline{5 \quad 8} & & 1 \\ 5 & \underline{8} & 1 & 2 \\ 8 & 1 & \underline{2} & 5 \end{bmatrix}$$

$$\mathrm{PGM2} = \begin{bmatrix} 0 & \underline{1 \quad 2 \quad 5 \quad 8} & 0 \\ 3 & \underline{1 \quad 2 \quad 5 \quad 8} & 3 \\ 4 & \underline{1 \quad 2 \quad 5 \quad 8} & 4 \\ 6 & \underline{1 \quad 2 \quad 5 \quad 8} & 6 \\ 7 & \underline{1 \quad 2 \quad 5 \quad 8} & 7 \\ 9 & \underline{1 \quad 2 \quad 5 \quad 8} & 9 \end{bmatrix}$$

V. Additional Properties for Node Disjoint Routing

In n-cube, there are 2^n nodes. But some of them are actually used, others are possibly used and the others are never used according to the Hamming distance of the source and destination nodes. In this chapter, we discuss how many nodes of n-cube are actually, possibly and never used for S_k and S_{k+2}.

Since there are k node disjoint paths of length k, there are k-1 intermediate nodes in each path of length k. Likewise, since there are n-k node disjoint paths of length k+2, there are k-1 intermediate nodes for each path of length k+2. Therefore, the number of the actually used nodes is

$$\begin{aligned} & (n-k)\cdot(k+1) + k\cdot(k-1) + 2 \\ = \quad & n\cdot k - k^2 + n - k + k^2 - k + 2 \\ = \quad & n\cdot k + n - 2\cdot k + 2 \\ = \quad & n\cdot(k+1) - 2\cdot(k-1). \end{aligned}$$

Let's consider two nodes A and B of $H(A,B) = k$. From the discussion of chapter III, all nodes in SQ(A,B) are possibly used for one of node disjoint paths of k. And from the discussion of chapter IV, all nodes in $SQ(A_{c,i},B_{c,i})$ for all i, where $i \in N_{k+2}$, are possibly used for one of node disjoint paths of length k+2. There are 2^k node in SQ(A,B). Since

$H(A_{c,i},B_{c,i})$ for all i, where $i \in N_{k+2}$, is k, there are also 2^k nodes in $SQ(A_{c,i},B_{c,i})$ for all i, where $i \in N_{k+2}$. Therefore, the number of possibly used nodes for S_k and S_{k+2} is

$$2^{k}\cdot(n-k+1).$$

Since there are 2^n nodes in n-cube, the number of never used nodes for two arbitrary nodes of Hamming distance k is

$$2^n - 2^{k}\cdot(n-k+1).$$

<u>Theorem 5.1</u> Let Hamming distance of two nodes A and B be k. A node is never used for S_k or S_{k+2} if and only if the address of a node contains at least two elements of N_{k+2}, where every node in n-cube is represented by distinct node numbers except the source and destination nodes.

Proof) Since the paths of S_k consist of only the nodes inside SQ(A,B), all intermediate nodes used for S_k must have no element of N_{k+2}.

Since the paths of S_{k+2} consist of only the nodes inside $SQ(A_{c,i},B_{c,i})$ for all i, where $i \in N_{k+2}$, all intermediate nodes used for S_{k+2} must have only one element of N_{n+2}.

If a node is never used, it has at least two elements of N_{k+2}.

Next, suppose a node X has x distinct elements of N_{k+2}, where $x \geq 2$.

Let's find the shortest path from the source to destination including the node X.

The source is k bit different from the destination indicated by the elements of N_k.

The values indicated by the elements of N_{k+2} are same in the addresses of the source and destination.

In order to have the node X in the shortest path, the value of each position indicated the elements of N_{k+2} must be complemented two times.

Therefore, $2 \cdot x$ complements are needed in addition to k complements for the shortest path, which produce the shortest path of length $k + 2 \cdot x$.

Since $x \geq 2$ from the assumption, the shortest path including the node X must be at least $k+4$.

Therefore the node X cannot be used for S_k or S_{k+2}.

∎

We show how many nodes are actually, possibly and never used for n = 5,10 and 20 according to the Hamming distance as follows.

Notice that if k = n or n-1, all the nodes in n-cube are possibly used for S_k or S_{k+2}. Also notice that if k = 1, the numbers of actually used and possibly used nodes are same.

Table 5-1 for n = 5

k	# of nodes actually used for n paths	# of nodes possibly used for n paths	# of nodes never used for n paths	# of node in n-cube
5	22	32	0	32
4	19	32	0	
3	16	24	8	
2	13	16	16	
1	10	10	22	

Table 5-2 for n = 10

k	# of nodes actually used for n paths	# of nodes possibly used for n paths	# of nodes never used for n paths	# of node in n-cube
10	92	1024	0	1024
9	84	1024	0	
8	76	768	256	
7	68	512	512	
6	60	320	704	
5	52	192	832	
4	44	112	912	
3	36	64	960	
2	28	36	988	
1	20	20	1004	

Table 5-3 for n = 20

k	# of nodes actually used for n paths	# of nodes possibly used for n paths	# of nodes never used for n paths	# of node in n-cube
20	382	1048576	0	1048576
19	364	1048576	0	
18	346	786432	262144	
17	328	524288	524288	
16	310	327680	720896	
15	292	196608	851968	
14	274	114688	933888	
13	256	65536	983040	
12	238	36864	1011712	
11	220	20480	1028096	
10	202	11264	1037312	
9	184	6144	1042432	
8	166	3328	1045248	
7	148	1792	1046784	
6	130	960	1047616	
5	112	512	1048064	
4	94	272	1048304	
3	76	144	1048432	
2	58	76	1048500	
1	40	40	1048536	

VI. Broadcast Algorithm using SPGM

In this chapter, we provide a broadcast algorithm which broadcasts a message to all nodes in n-cube in n time units.

We take an example to describe the algorithm. We show a 6 x 6 SPGM for 6-cube and all nodes generated by each row of the SPGM with the permutations of the underlined parts. Assume that the source = 000000 and the destination = 111111. We call the underlined parts the permutation parts.

$$\mathrm{SPGM} = \begin{bmatrix} 0 & \underline{1} & \underline{2} & \underline{3} & \underline{4} & 5 \\ 1 & \underline{2} & \underline{3} & \underline{4} & \underline{5} & 0 \\ 2 & \underline{3} & \underline{4} & \underline{5} & 0 & 1 \\ 3 & \underline{4} & \underline{5} & 0 & \underline{1} & 2 \\ 4 & \underline{5} & 0 & \underline{1} & \underline{2} & 3 \\ 5 & 0 & \underline{1} & \underline{2} & \underline{3} & 4 \end{bmatrix}$$

$0 \ \underline{1 \ 2 \ 3 \ 4} \ 5$

(0)	(0,1) (0,2) (0,3) (0,4)	(0,1,2) (0,1,3) (0,1,4) (0,2,3) (0,2,4) (0,3,4)	(0,1,2,3) (0,1,2,4) (0,1,3,4) (0,2,3,4)	(0,1,2,3,4)	(0,1,2,3,4,5)

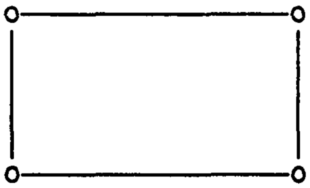
4-subcube

```
1  2  3  4  5  0

(1)   (1,2)  (1,2,3)  (1,2,3,4)
      (1,3)  (1,2,4)  (1,2,3,5)  (1,2,3,4,5)   (1,2,3,4,5,0)
      (1,4)  (1,2,5)  (1,3,4,5)
      (1,5)  (1,3,4)  (2,3,4,5)
             (1,3,5)
             (1,4,5)
                                      4-subcube

2  3  4  5  0  1

(2)   (2,3)  (2,3,4)
      (2,4)  (2,3,5)  (2,3,4,5)   (2,3,4,5,0)  (2,3,4,5,0,1)
      (2,5)  (2,4,5)
                         3-subcube

3  4  5  0  1  2

(3)   (3,4)  (3,4,5)    (3,4,5,0)  (3,4,5,0,1)    (3,4,5,0,1,2)
      (3,5)
                                        1-subcube
               2-subcube

4  5  0  1  2  3

(4)   (4,5)    (4,5,0)  (4,5,0,1)  (4,5,0,1,2)   (4,5,0,1,2,3)
                        (4,5,0,2)
   1-subcube
                                      2-subcube

5  0  1  2  3  4

(5)   (5,0)  (5,0,1)  (5,0,1,2)
             (5,0,2)  (5,0,1,3)  (5,0,1,2,3)   (5,0,1,2,3,4)
             (5,0,3)  (5,0,2,3)
                                     3-subcube
```

As shown above, all nodes except the source in 6-cube are shown in one of six groups derived from each row of the SPGM. In a group of nodes, some nodes are grouped by one or two subcubes which are shown above by boxes. The number of subcubes is same as the number of permutaion parts and the subcube dimension is the number of elements in the permutation part. But then, since each subcube is also a hypercube, same procedure can be applied to each subcubes recursively until all subcubes are of one or two dimension. In 1-cube and 2-cube, it is clear that messages are broadcast in one and two time units respectively.

1-cube

2-cube



Let's take the subcube from the second row of the given SPGM.

		(1,2,3)		
	(1,2)	(1,2,4)	(1,2,3,4)	
(1)	(1,3)	(1,2,5)	(1,2,3,5)	(1,2,3,4,5)
	(1,4)	(1,3,4)	(1,3,4,5)	
	(1,5)	(1,3,5)	(2,3,4,5)	
		(1,4,5)		

All these sixteen nodes are in SQ((1), (1,2,3,4,5)). Since H((1), (1,2,3,4,5)) is four and N_k = { 2, 3, 4, 5 }, a SPGM can be constructed by selecting any sequence of the elements of N_k. Consider the following SPGM with permutation parts.

$$\text{SPGM} = \begin{bmatrix} 2 & \underline{3 \quad 4} & & 5 \\ 3 & \underline{4 \quad 5} & & 2 \\ 4 & \underline{5} & 2 & 3 \\ 5 & 2 & \underline{3} & 4 \end{bmatrix}$$

$2 \quad \underline{3 \quad 4} \quad 5$

$$\underset{\text{2-subcube}}{\boxed{(1,2) \qquad \begin{matrix}(1,2,3)\\(1,2,4)\end{matrix} \qquad (1,2,3,4)}} \qquad (1,2,3,4,5)$$

$3 \quad \underline{4 \quad 5} \quad 2$

$$\underset{\text{2-subcube}}{\boxed{(1,3) \qquad \begin{matrix}(1,3,4)\\(1,3,5)\end{matrix} \qquad (1,3,4,5)}} \qquad (1,3,4,5,2)$$

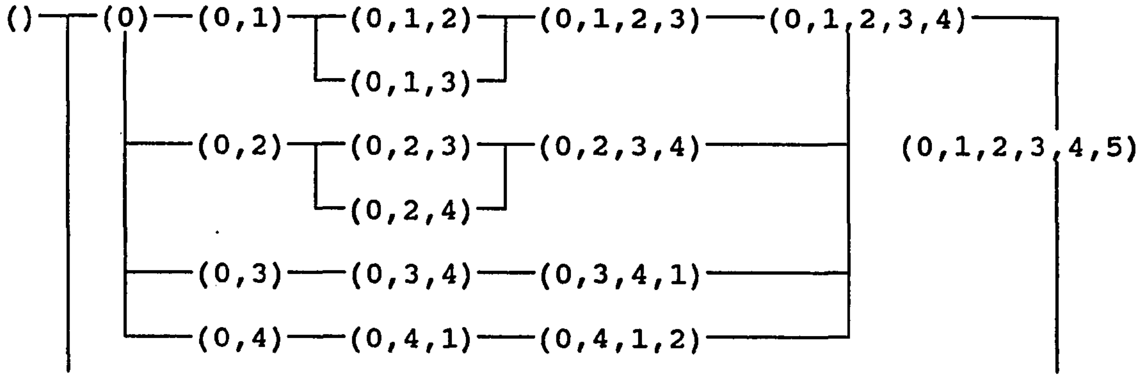
$4 \quad \underline{5} \quad 2 \quad 3$

$$\underset{\text{1-subcube}}{\boxed{(1,4) \qquad (1,4,5)}} \qquad (1,4,5,2) \qquad (1,4,5,2,3)$$

$5 \quad 2 \quad \underline{3} \quad 4$

$$(1,5) \qquad \underset{\text{1-subcube}}{\boxed{(1,5,2) \qquad (1,5,2,3)}} \qquad (1,5,2,3,4)$$

All other rows can be processed in the same way. And broadcasting from the source to all nodes takes six time units. One of six broadcast paths is shown below.



Now, let's discuss the broadcast algorithm for n-cube. This algorithm sends the message from the source node to all other nodes in n time units for n-cube. If we know the address of the source node, then we can decide the destination node by exclusive or operation. In order to have n time units for broadcast in n-cube, we have to take advantage of the property of a hypercube that there are n node disjoint paths of length n between two arbitrary nodes of Hamming distance n. We discussed a node disjoint routing algorithm in chapter III. We show how that algorithm can be used for broadcast in n-cube. Our broadcast algorithm requires an n x n SPGM with permutation parts discussed in theorem 3.2. By a PGM we mean the PGM derived from a SPGM unless otherwise mentioned in this chapter.

The matrix from the following SPGM

$$\text{SPGM} = \begin{bmatrix} a_0 & a_1 & a_2 & \cdots\cdots & a_{n-2} & a_{n-1} \\ a_1 & a_2 & a_3 & \cdots\cdots & a_{n-1} & a_0 \\ a_2 & a_3 & a_4 & \cdots\cdots & a_0 & a_1 \\ . & . & . & \cdots\cdots & . & . \\ . & . & . & \cdots\cdots & . & . \\ . & . & . & \cdots\cdots & . & . \\ a_{n-1} & a_0 & a_1 & \cdots\cdots & a_{n-3} & a_{n-2} \end{bmatrix}$$

with $\text{Perm}_{0,1,n-2}$, $\text{Perm}_{i,1,n-i-1}$ and $\text{Perm}_{i,n-i+1,n-2}$, for all i where $1 \le i \le n-1$, is a PGM for the node disjoint paths of length n.

We divide permutaion parts into four different types. Suppose a row i is $a_0 \quad a_1 \quad a_2 \quad \cdots\cdots \quad a_{n-2} \quad a_{n-1}$.

Type 1 $\text{Perm}_{i,1,n-2}$
Type 2 $\text{Perm}_{i,1,n-3}$
Type 3 $\text{Perm}_{i,2,n-2}$
Type 4 $\text{Perm}_{i,1,j}$ and $\text{Perm}_{i,j+2,n-2}$
,where $0 \le i \le n-1$ and $1 \le j \le n-4$

Since we need an n x n SPGM, $N_k = \{ 0, 1, 2, , , n-1\}$ and $N_{k+2} = \{ \}$. From the SPGM construction algorithm, there must be one element of N_k in front of all permutation parts. All nodes generated by this node and a permutation part construct a subcube. Let a row be $a_f \ \underline{a_1 \ a_2 \ . \ . \ . \ a_p} \ . \ . \ .$ The subcube constructed by a_f and $\underline{a_1\ \ a_2\ \ .\ \ .\ \ .\ \ a_p}$ is denoted by SQ($[a_f],[a_f,a_1,a_2\ \ .\ .\ .\ \ a_p]$). By a RM construction algorithm, the first node generated by this row is $[a_f]$ and the node obtained by a_f $\underline{a_1\ \ a_2\ \ .\ \ .\ \ .\ \ a_p}$ is $[a_f,a_1,a_2\ \ .\ .\ .\ \ a_p]$. As a result, the Hamming distance of these two nodes is p. In other words, the dimension of the subcube is the number of the elements in the permutation part.

example)

For 10-cube, a row $0\ \ \underline{1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ 7\ \ 8}\ \ 9$ generates SQ([0],[0,1,2,3,4,5,6,7,8] whose dimension is eight.

For 10-cube, a row $0\ \ \underline{1\ \ 2\ \ 3}\ \ 4\ \ \underline{5\ \ 6\ \ 7\ \ 8}\ \ 9$ generates SQ([0],[0,1,2,3] whose dimension is three and SQ([0,1,2,3,4],[0,1,2,3,4,5,6,7,8]) whose dimension is four.

From the above discussion, we can evaluate the number of nodes each type of permutation parts can generates.

Permutation parts of type 1 generate 2^{n-2} nodes.
Permutation parts of type 2 generate $2^{n-3} + 1$ nodes.
Permutation parts of type 3 generate $2^{n-3} + 1$ nodes.
Permutation parts of type 4 generate $2^{j} + 2^{n-j-3}$ nodes,
$1 \leq j \leq n-4$.

But here, the source and destination nodes are not included.

Since there are two permutation parts of type 1, the number of nodes generated from these is

$$2 \cdot 2^{n-2}.$$

Since there are one permutation part of type 2, the number of nodes generated from this is

$$(2^{n-3} + 1).$$

Since there are one permutation part of type 3, the number of nodes generated from this is

$$(2^{n-3} + 1).$$

Since there are n-4 permutation parts of type 4, the number of nodes generated from these is

$$\sum_{j=1}^{n-4} (2^{j} + 2^{n-j-3})$$

Therefore, the sum of all these numbers and the source and destination nodes must be 2^n.

$$2 \cdot 2^{n-2} + (2^{n-3} + 1) + (2^{n-3} + 1) + \sum_{j=1}^{n-4} (2^{j} + 2^{n-j-3}) + 2$$

$$= \quad 2^{n-1} + 2 \cdot (2^{n-3} + 1) + \sum_{j=1}^{n-4} (2^{j} + 2^{n-j-3}) + 2$$

$$= \quad 2^{n-1} + 2^{n-2} + \sum_{j=1}^{n-4} (2^{j} + 2^{n-j-3}) + 4$$

$$= \quad 2^{n-1} + 2^{n-2} + 4 \cdot (2^{n-4} - 1) + 4$$

$$= \quad 2^{n-1} + 2^{n-2} + 2^{n-2}$$

$$= \quad 2^{n}$$

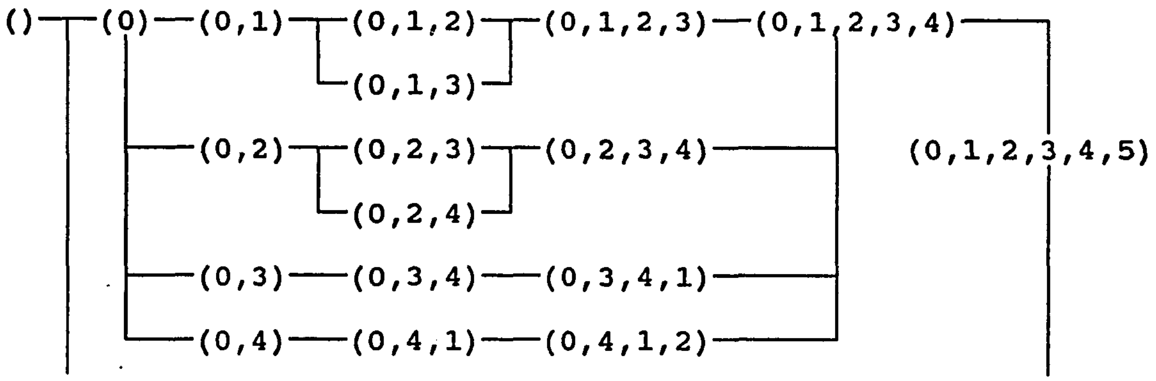
We call this process the subcube division algorithm. A subcube division is shown in fig. 6-1. After we apply this algorithm for n-cube, we can find a set of the next nodes to which the source can sends the message in parallel. Since n-cube is a subcube of itself, we can generalize as follows. After the subcube division algorithm is applied for k-subcube, where k > 2, there will be pairs of k-2, k-3, , , 2 and 1 dimensional subcubes. Since each subcube is another hypercube, this algorithm can be applied recursively. In order to broadcast a message in n time units, this recursive algorithm must be applied in parallel for all rows of PGM's of all possible subcubes, until there is no subcube of dimension 2. By doing this, all nodes generated by each row of the n x n PGM will be properly distributed for the broadcast. For every subcubes, we have its own source and destination nodes. We call them the intermediate source and destination nodes respectively. As mentioned before, it is clear that messages are broadcast in one and two time units in 1-subcube and 2-subcube respectively.

Here we have the subcube division algorithm.

```
for i = 0 to n-1 do
     if i = 0
          . Construct a PGM properly for n-cube
          . Decide the next nodes to send the messages.
          . Send the messages to the nodes in parallel.
     else
          . Construct PGM's for all possible subcubes.
          . Decide the next nodes to send the messages
            for all possible SQ's.
          . Send the messages to the nodes in parallel
            for all possible subcubes.
     endif
```

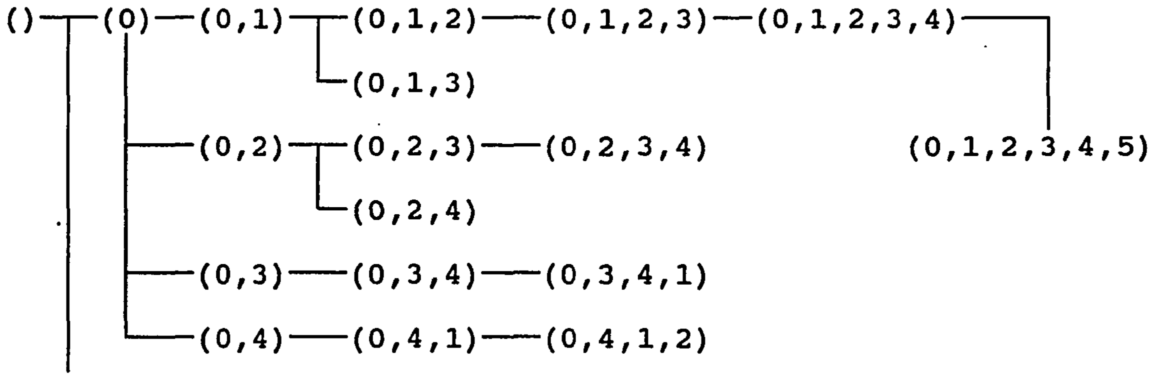
When we are talking about broadcasting, it is assumed that all nodes in n-cube are not faulty. If there are faulty edges in n-cube, we can avoid them by selecting a proper PGM.

With this algorithm, all destination nodes of subcubes of dimension > 1 receive redundant messages. Consider the following example.



In this example, nodes (0,1,2,3), (0,2,3,4), (0,1,2,3,4) and (0,1,2,3,4,5) receive the redundant messages.

But we can avoid redundancy by sending the message to all destinations of subcubes of dimension > 1 only if the route is derived from the first row of the PGM's for the subcube. Then the previous example will change as follows. Four broadcast trees for 4-cube and one tree for 6-cube are shown in fig. 6.2 and fig 6.3.



Theorem 6.1 The number of subcube divisions of n-cube using permutations parts shown in theorem 3.2 is

$$\begin{array}{ll} 1 & \text{if } n = 1,2 \\ 2 & \text{if } n = 3 \\ n! & \text{if } n \geq 4. \end{array}$$

proof) We can prove the cases of $n = 1$ and 2 by observation.

In case of $n = 3$, we can apply the theorem 3.2. But Since there are two permutation areas and each of them has only one entry, there cannot be different matrix.

Since there are $(3-1)!$ or 2 different sequences, there are two different subcube divisions.

Let's consider the cases of $n \geq 4$. Since there are $(n-1)!$ different sequences and there are n different set of permutation parts, there are $n!$ different subcube divisions for n-cube, where $n \geq 4$.

∎

example) For 4-cube, there are 4! different subcube divisions.

$$\left|\begin{array}{cccc} 0 & \underline{1} & \underline{2} & 3 \\ 1 & \underline{2} & \underline{3} & 0 \\ 2 & \underline{3} & 0 & 1 \\ 3 & 0 & \underline{1} & 2 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & 1 & \underline{2} & 3 \\ 1 & \underline{2} & \underline{3} & 0 \\ 2 & \underline{3} & \underline{0} & 1 \\ 3 & \underline{0} & 1 & 2 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{1} & 2 & 3 \\ 1 & 2 & \underline{3} & 0 \\ 2 & \underline{3} & \underline{0} & 1 \\ 3 & \underline{0} & \underline{1} & 2 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{1} & \underline{2} & 3 \\ 1 & \underline{2} & 3 & 0 \\ 2 & 3 & \underline{0} & 1 \\ 3 & \underline{0} & \underline{1} & 2 \end{array}\right|$$

$$\left|\begin{array}{cccc} 0 & \underline{1} & \underline{3} & 2 \\ 1 & \underline{3} & \underline{2} & 0 \\ 3 & \underline{2} & 0 & 1 \\ 2 & 0 & \underline{1} & 3 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & 1 & \underline{3} & 2 \\ 1 & \underline{3} & \underline{2} & 0 \\ 3 & \underline{2} & \underline{0} & 1 \\ 2 & \underline{0} & 1 & 3 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{1} & 3 & 2 \\ 1 & 3 & \underline{2} & 0 \\ 3 & \underline{2} & \underline{0} & 1 \\ 2 & \underline{0} & \underline{1} & 3 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{1} & \underline{3} & 2 \\ 1 & \underline{3} & 2 & 0 \\ 3 & 2 & \underline{0} & 1 \\ 2 & \underline{0} & \underline{1} & 3 \end{array}\right|$$

$$\left|\begin{array}{cccc} 0 & \underline{2} & \underline{1} & 3 \\ 2 & \underline{1} & \underline{3} & 0 \\ 1 & \underline{3} & 0 & 2 \\ 3 & 0 & \underline{2} & 1 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & 2 & \underline{1} & 3 \\ 2 & \underline{1} & \underline{3} & 0 \\ 1 & \underline{3} & \underline{0} & 2 \\ 3 & \underline{0} & 2 & 1 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{2} & 1 & 3 \\ 2 & 1 & \underline{3} & 0 \\ 1 & \underline{3} & \underline{0} & 2 \\ 3 & \underline{0} & \underline{2} & 1 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{2} & \underline{1} & 3 \\ 2 & \underline{1} & 3 & 0 \\ 1 & 3 & \underline{0} & 2 \\ 3 & \underline{0} & \underline{2} & 1 \end{array}\right|$$

$$\left|\begin{array}{cccc} 0 & \underline{2} & \underline{3} & 1 \\ 2 & \underline{3} & \underline{1} & 0 \\ 3 & \underline{1} & 0 & 2 \\ 1 & 0 & \underline{2} & 3 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & 2 & \underline{3} & 1 \\ 2 & \underline{3} & \underline{1} & 0 \\ 3 & \underline{1} & \underline{0} & 2 \\ 1 & \underline{0} & 2 & 3 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{2} & 3 & 1 \\ 2 & 3 & \underline{1} & 0 \\ 3 & \underline{1} & \underline{0} & 2 \\ 1 & \underline{0} & \underline{2} & 3 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{2} & \underline{3} & 1 \\ 2 & \underline{3} & 1 & 0 \\ 3 & 1 & \underline{0} & 2 \\ 1 & \underline{0} & \underline{2} & 3 \end{array}\right|$$

$$\left|\begin{array}{cccc} 0 & \underline{3} & \underline{1} & 2 \\ 3 & \underline{1} & \underline{2} & 0 \\ 1 & \underline{2} & 0 & 3 \\ 2 & 0 & \underline{3} & 1 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & 3 & \underline{1} & 2 \\ 3 & \underline{1} & \underline{2} & 0 \\ 1 & \underline{2} & \underline{0} & 3 \\ 2 & \underline{0} & 3 & 1 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{3} & 1 & 2 \\ 3 & 1 & \underline{2} & 0 \\ 1 & \underline{2} & \underline{0} & 3 \\ 2 & \underline{0} & \underline{3} & 1 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{3} & \underline{1} & 2 \\ 3 & \underline{1} & 2 & 0 \\ 1 & 2 & \underline{0} & 3 \\ 2 & \underline{0} & \underline{3} & 1 \end{array}\right|$$

$$\left|\begin{array}{cccc} 0 & \underline{3} & \underline{2} & 1 \\ 3 & \underline{2} & \underline{1} & 0 \\ 2 & \underline{1} & 0 & 3 \\ 1 & 0 & \underline{3} & 2 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & 3 & \underline{2} & 1 \\ 3 & \underline{2} & \underline{1} & 0 \\ 2 & \underline{1} & \underline{0} & 3 \\ 1 & \underline{0} & 3 & 2 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{3} & 2 & 1 \\ 3 & 2 & \underline{1} & 0 \\ 2 & \underline{1} & \underline{0} & 3 \\ 1 & \underline{0} & \underline{3} & 2 \end{array}\right| \qquad \left|\begin{array}{cccc} 0 & \underline{3} & \underline{2} & 1 \\ 3 & \underline{2} & 1 & 0 \\ 2 & 1 & \underline{0} & 3 \\ 1 & \underline{0} & \underline{3} & 2 \end{array}\right|$$

fig.6-1

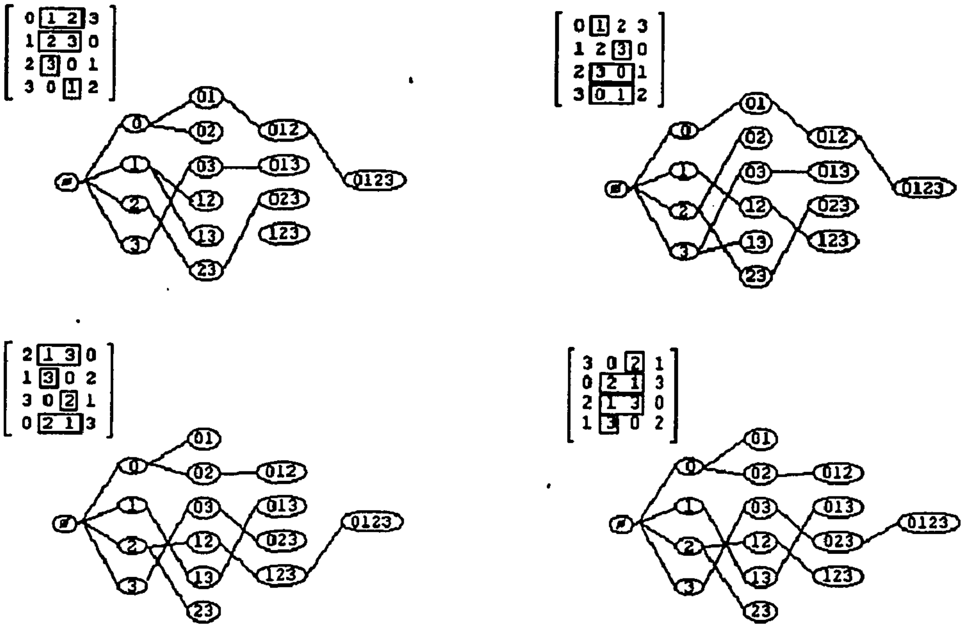
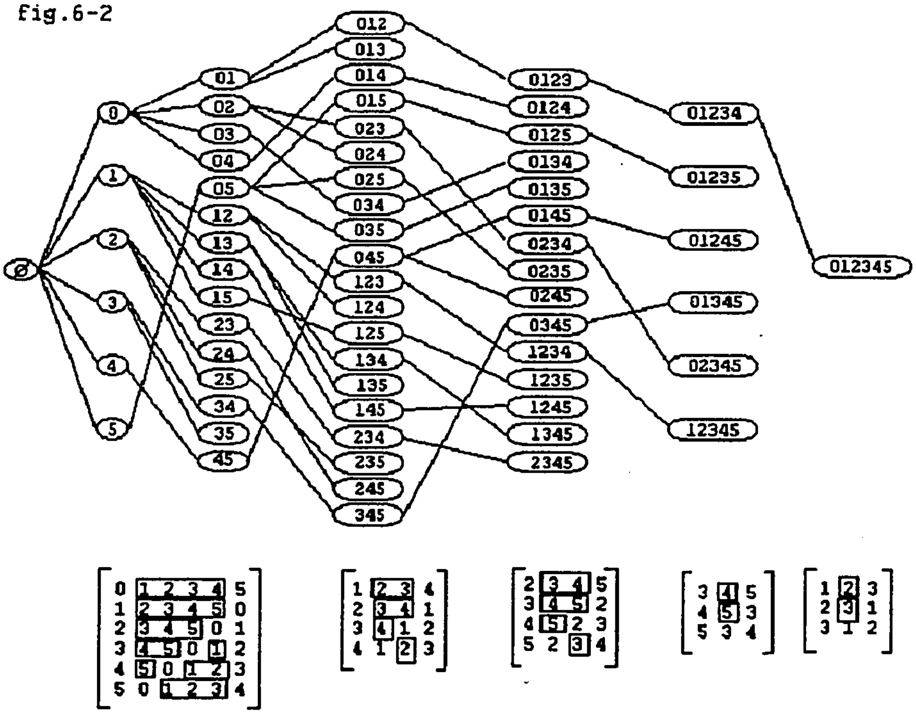

0 1 2 3
1 2 3 0
2 3 0 1
3 0 1 2
0 1 2 3
1 2 3 0
2 3 0 1
3 0 1 2
2 1 3 0
1 3 0 2
3 0 2 1
0 2 1 3
3 0 2 1
0 2 1 3
2 1 3 0
1 3 0 2
∅
0
1
2
3
01
02
03
12
13
23
012
013
023
123
0123

fig.6-2


∅
0
1
2
3
4
5
01
02
03
04
05
12
13
14
15
23
24
25
34
35
45
012
013
014
015
023
024
025
034
035
045
123
124
125
134
135
145
234
235
245
345
0123
0124
0125
0134
0135
0145
0234
0235
0245
0345
1234
1235
1245
1345
2345
01234
01235
01245
01345
02345
12345
012345
0 1 2 3 4 5
1 2 3 4 5 0
2 3 4 5 0 1
3 4 5 0 1 2
4 5 0 1 2 3
5 0 1 2 3 4
1 2 3 4
2 3 4 1
3 4 1 2
4 1 2 3
2 3 4 5
3 4 5 2
4 5 2 3
5 2 3 4
3 4 5
4 5 3
5 3 4
1 2 3
2 3 1
3 1 2

VII. Conclusions

In this paper, we have proposed node disjoint routing algorithms for the paths of length k and k+2 between two arbitrary nodes of Hamming distance k in n-cube. These algorithms use the bit positions of the addresses of the two nodes. The algorithms for the paths of length k use k x k matrices, called the SPGM, constructed with the bit positions where the values of the addresses of the source and destination nodes are not same. The algorithms for the paths of length k+2 use (n-k) x (k+2) matrices constructed with all bit positions. Fault tolerant node disjoint routing can be achieved by constructing proper matrices with the elements of N_k and N_{k+2} and rearranging the entries of the matrices with certain restrictions shown in previous chapters. Especially, k x k SPGM's with the given permutations of the entries of SPGM's have one property that it can use every node in the k-cube for one of the k paths of length k. From this special property, a broadcast algorithm sending a message to all other nodes in n time units for n-cube can be obtained by using proper SPGM's to subcubes recursively. We also showed how many and which nodes are never used for n node disjoint paths between two arbitrary nodes in n-cube. We show the steps to find n node disjoint paths between two arbitrary nodes in n-dimensional hypercube.

1. Evaluate two sets of N_k and N_{k+2} from the addresses of the source and destination nodes.

2. Preclude the nodes which contain more than one element of N_{k+2} by examining node numbers of the faulty nodes.

3. For the k node disjoint paths of length k,

    a. select a sequence with elements of N_k avoiding faulty conditions as much as possible.
    b. construct k x k SPGM.
    c. rearrange the entries of the SPGM with the given restrictions to avoid the rest of faulty conditions.

4. For the n-k node disjoint paths of length k+2, rearrange the entries of the (n-k) x (k+2) PGM with the given restrictions to avoid all faulty conditions.

Now, we conclude by mentioning our future research. We are going to analyze the algorithms shown in chapter III. The SPGM construction algorithm has special properties. For example, if all weights of faulty nodes are distinct and greater than 1 and smaller than k-1, then there exist k node disjoint paths of length k. In this case, a sequence of the elements of N_k can be selected avoiding all faulty nodes. For some other cases, it might not be needed to select a sequence and to construct PGM in order to avoid faulty elements. Thus, we are looking for specific conditions among faulty elements so that we can find node disjoint paths without selecting or constructing matrices. Furthermore, we are looking for conditions guaranteeing the existence of less than k node disjoint paths of length k for k-subcube. For example, if there is only one faulty node of weight 1, then there cannot be k node disjoint paths between two given nodes but there must exist k-1 node disjoint paths. The broadcast algorithm shown in chapter VI can be used for fault tolerant broadcast algorithm, where by fault tolerant broadcast algorithm we mean the broadcast from a non faulty node to all non faulty nodes in n-dimensional hypercube. Since the algorithm shown in chapter VI generates more balanced broadcast trees than typical broadcast algorithm, we can improve the performance of the all to all broadcast algorithm.

VIII. References

[1] F. Harary, Graph Theory. Addison-Wesley, 1969 pp 22-23.

[2] Zvi Kohavi "Switching and Finite Automata Theory" second edition pp 14-16

[3] Y. Saad and M.H. Schultz "Topological properties of hypercubes" IEEE Tran. Comput., vol. C-37, no. 7, pp 867-872, July 1988.

[4] Ilyong Chung "Dynamic Security" Ph.d Thesis from Graduate School and University Center of City University of New York. pp 64 - 81, Dec. 1990.

[5] Ming-Syan Chen and Kang G. Shin, "Adaptive Fault-Tolerant Routing in Hypercube Multicomputers" IEEE Trans. Comput., vol. 39. NO. 12. pp 1406-1416, Dec. 1990.

[6] M.S. Chen and K. G. Shin, "Message routing in an injured hypercube" in Proc. Third Conf. Hypercube Concurrent Compu. Appl., Jan 1988, pp.312-317.

[7] M.S. Chen and K.G. Shin "On hypercube fault-tolerant routing using global information" in Proc. Fourth Conf. Hypercube Concurrent Comput. Appl., Mar. 1989 pp 83-86.

[8] C.K. Kim and D.A. Reed, "Adaptive packet routing in a hypercube" in Proc., Third Conf., Hypercube Concurrent Compt. Appl., Jan 1988, pp 625-630.

[9] H. Katseff, "Incomplete hypercube" IEEE Trans. Comput., vol.c-37, no. 5, pp 604-608, May 1988.

[10] J.R. Armstrong and F.G. Gray, "Fault diagnosis in a Boolean n-cube array of microprocessors" IEEE trans. Computer, vol. C-30, no. 8, pp. 587-590, Aug. 1981

[11] G.G.L. Meyer and G.M. Masson "An efficient fault diagnosis algorithm for symmetric multiple processor architectures" IEEE trans. Computers, vol C-27, pp 1059-1063, Nov. 1978

[12] S.L. Hakami and A.T. Amin "Characterization of the connection assignment of diagnosable systems "IEEE trans. Computers, vol. c-23, pp. 86-88, Jan. 1974

[13] J.R. Armstrong and F.G. Gray "Fault diagnosis in a Boolean n-cube array of microprocessors" Dep. Elec. Eng. Virginia Polytech. Inst. and State Univ. ech. Rep. EE-

8103

[14] R.P. Bianchini, Jr. and R.W. Buskens "Implementation of On-Line Distributed System-Level Diagnosis Theory" IEEE Transactions on Computers May 1992 pp 616-626

[15] R.P. Bianchini, Jr. K. Goodwin and D.S. Nydick "Practical Application and Implementation of Distributed System-Level Diagnosis Theory" Int'l Symp. on Fault Tolerant Computing Jun. 1990 pp 332-339

[16] A.T. Dabura, K.K. Sabnani and W.J. Hery "Spare Capacity as a Means of Fault Detection and Diagnosis in Multiprocessor Systems" IEEE Transactions on Computers Jun 1989 pp 881-890

[17] P.Ramanathan and K.G. Shin "Reliable Broadcast in Hypercube Multicomputers" IEEE Transactions on Computers Dec. 1988 pp 1654-1656

[18] C.Aykanat and F. Ozguner "A Concurrent Error Detecting Conjugate Gradient Algorithm on a Hypercube multiprocessor" Int'l Symp. on Fault Tolerant Computing Jul. 1987 pp 204-209

[19] A. hafoor and P. Sole "Performance of Fault Tolerant Diagnostics in the Hypercube Systems" IEEE Transactions on Computers Aug. 1989 pp 1164-1172

[20] P.Banerjee "An Evaluation of System-Level Fault Tolerance on the Intel Hypercube Multiprocessor" Int'l Symp. on Fault Tolerant Computing Jun. 1988 pp 362-367

[21] A.H. Esfahanian "Generalized Measures of Fault Tolerance with Application to N-cube Networks" IEEE Transactions on Computers Nov. 1989 pp 1586-1590

[22] D.A. Rennels "On Implementing Fault Tolerance in Binary Hypercubes" Int'l Symp. on Fault Tolerant Computing Jul. 1986 pp 344-349

[23] E.Dilger and E. Ammann "System-Level Self-Diagnosis in n-cube Connected Multiprocessor Networks" Int'l Symp. on Fault Tolerant Computing Jun. 1984 pp 184-189

[24] D.A. Rennels "Fault Tolerant Computing - Concepts and Examples" IEEE Transactions on Computers Dec. 1984 pp 1116-1129

[25] S.L. Hakimi and K. Nakajama "On Adaptive System Diagnosis" IEEE Transactions on Computers Mar. 1984 pp 234-240

[26] P. Banerjee, J.T. Rahmeh, C.Stunkel, V.S. Nair, K. Roy, V. Balasubramanian and J.A. Abraham "Algorithm-Based Fault Tolerance on a Hypercube Multiprocessor" IEEE Transactions on Computers Sep. 1990 pp 1132-1145

[27] T.H. Lin and K.G. Shin "Locations of a Fault Module in a Computing System" IEEE Transactions on Computers Feb. 1990 pp 182-194

[28] F.Baris, F.Grandomi and P.Maestrini "A Theory of Diagnos ability without Repairs" IEEE Transactions on Computers Feb. 1976 pp 585-593

[29] K.Y.Chwa and S.L.Hakimi "On Fault Identification in Diagnosable Systems" IEEE Transactions on Computers Jun. 1981 pp 414-422

[30] A.avianpour and K.H.Kim "Diagnosabilities of Hypercubes under the Pessimistic One-Step Diagnosis Strategy" IEEE Transactions on Computers Feb. 1991 pp 232-236

[31] D.Brahme and J.A.Abraham "Functional Testing of Microprocessors" IEEE Transactions on Computers Jun. 1984 pp 475-485

[32] K.V.S.Bhat "An Efficient Approach for Fault Diagnosis in a Boolean n-cube Array of Microprocessors" IEEE Transactions on Computers Nov. 1983 pp 1070-1071

[33] B.S. Chllebus, A.Diks and A.Pelc "Optimal Broadcasting in Faulty Hypercube" Int'l Symp. on Fault Tolerant Computing Jun. 1991 pp 266-273

[34] P.Banerjee "Strategies for Reconfiguring Hypercubes under Faults" Int'l Symp. on Fault Tolerant Computing Jun. 1990 pp 210-217

[35] M.Peercy and P.Banerjee "Distributed Algorithms for Shortest Path, Deadlock-free routing and broadcasting in Arbitrary Faulty Hypercubes" Int'l Symp. on Fault Tolerant Computing Jun. 1990 pp 218-225

[36] V.Raghavan and A.R.ripathi "Improved Diagnosability Algorithms" IEEE Transactions on Computers Feb. 1991 pp 143-153