

# 분산/병렬 시스템을 위한 최소화의 오류-허용 방사형 그래프 설계

전 문 석<sup>†</sup> · 이 문 구<sup>††</sup>

## 요 약

순열 방사형 그래프는 병렬 또는 분산 시스템의 상호 연결망 구조로서 n-큐브의 새로운 대안으로 제시되고 있다. 그러나 최근까지 제시된 구조(메쉬, 하이퍼큐브 등)에 대한 오류-허용 설계 모델은 많이 연구되어왔지만 순열 방사형 그래프에 적합한 오류-허용 설계 모델은 연구되고 있지 않다. 따라서 본 논문에서는 순열 방사형 그래프에 적합한 새로운 오류-허용 설계 방법을 제안하였다. 이 방법은 현재 수행중인 구성 요소 중에서 오류가 발생할 때 기존 구조를 유지하기 위해서, 예비 구성 요소를 추가하여 적절히 오류 요소를 대체하는 기법이다. 먼저, 순열 방사형 그래프를 순환 그래프로 변환한 다음 순환 그래프의 순-동형 성질을 이용하였다. 또한 k개 예비프로세서를 추가하고 각 프로세서 당 최대 통신 링크를 최소화함으로써 오류-허용 설계구조의 비용을 최적화 하였다. 특히, 최대 통신 링크의 수를 최소화하는 새로운 연구 방법을 제안하였다.

## Minimum Design of Fault-Tolerant Arrangement Graph for Distributed & Parallel System

Moon-Seog Jun<sup>†</sup> · Moon-Gu Lee<sup>††</sup>

## ABSTRACT

The arrangement graph, which is a viable interconnection scheme for parallel and distributed systems, has been proposed as an attractive alternative to the n-cube. However, a fault-tolerant design model which is well suitable for the arrangement graph doesn't have been proposed until recently, but fault-tolerant design models for many schemes have been proposed in a large number of papers. So, our paper presents a new fault-tolerant design technique suited for the arrangement graph. To maintain the previous structures when it occurs a fault in the current processing, the scheme properly substitutes a fault-component into the existing structures by adding a spare component. The first of all, it converts arrangement graph into a circulant graph using the hamiltonian property and then uses automorphism of circulant graph to tolerate faults. Also, we optimize the cost of fault-tolerant architectures by adding exactly k spare processors (while tolerating up to k processor) and minimizing the maximum number of links per processor. Specially, we propose a new technique to minimize the maximum number of links.

### 1. 서 론

방사형 그래프(arrangement graph)는 현재까지 널리

사용되고 있는 n-큐브보다 여러 가지 측면에서 우수한 특성을 보이면서 n-큐브를 대신할 수 있는 새로운 대안으로 제시되고 있다[1,2]. 그러나 n-차원 방사형 그래프는 차원 n이 증가함에 따라 노드의 수가 n!로 급격히 증가하는 단점을 갖고 있다. 따라서 이러한 n차원 방사형 그래프의 단점을 제거한 순열 방사형 그래프가 최근 제안되어 많은 관심을 갖고 연구되어지고

※ 본 연구는 1997년 11월 한국과학기술재단의 자유공모과제(1997-001-1500106)로 지원받았음.

† 종신회원 : 숭실대학교 컴퓨터학부 교수

†† 준 회원 : 명지전문대학 전산과 겸임교수

논문집수 : 1998년 3월 13일, 심사완료 : 1998년 10월 13일

있다[3].

순열방사형 그래프  $A_{nk}$ 는  $1 \leq k \leq n-1$ 을 만족하는 두 개의 파라미터  $n$ 과  $k$ 에 의해서 정의된다.  $A_{nk}$ 의 노드는  $n$ 개(1, 2, ...,  $n$ )원소 중에서  $k$ 개의 원소를 배열하는 것이다.  $A_{nk}$ 의 간선은  $k$ 개의 위치 중에서 정확히 한 위치만 다른 노드와 연결한다. 예를 들면  $A_{52}$ 에서 노드  $p = 24321$ , 23125, 14134, 54와 연결된다. 또한 이 순열방사형 그래프는 노드/간선 대칭성, 강한 계층성, 최대 오류 허용, 강한 회복성과 같은 장점을 가지고 있을 뿐만 아니라 파라미터인  $n$ 과  $k$ 를 적절히 조절함으로써 노드의 수와 차수/직경의 절충을 좀더 적절히 선택할 수 있다. 예를 들어 5040개의 노드를 갖는 순열 방사형 그래프가 있을 때  $n=7$ 이고  $k=6$ 인 것과  $n=10$ 이고  $k=4$ 인 것으로 두 가지 조합이 가능하다. 첫 번째 조합 ( $n=7, k=6$ )은 차수가 6이고 직경이 9이며 두 번째 조합( $n=10, k=4$ )은 차수가 24이고 직경이 6이다. 따라서 만약 차수가 증가하더라도 직경이 낮은 것을 원한다면 두 번째 조합을 선택하면 된다[3].

한편 VLSI 기술의 발전과 병렬 알고리즘에 사용되는 구성요소(프로세서, 링크...)의 수가 증가하면서 여러 구성요소에 대한 고장의 기획 또한 증가하게 되었다. 임의의 구성요소의 고장에 대해서도 원래의 상호 연결망 구조를 유지하면서 성능의 저하가 없는 오류 허용 시스템의 요구가 증가되고 있으며, 이러한 요구에 따라 오류 허용 병렬 구조를 만들기 위한 수많은 연구가 진행되어 왔다. 이들 연구에서 사용된 기법들은 크게 두 가지 범주로 나누어질 수 있다. 첫째는 예비 구성 요소를 추가하지 않는 방법으로 오류 발생 부분에서 수행해야 할 일을 정상 부분에서 대신 수행토록 하는 기법으로 작은 성능 저하를 감수하여야 할 한다. 그러나 이러한 성능의 저하는 실제 대단히 심각한 문제가 될 수도 있다. 둘째는 예비 구성요소를 추가하는 방법으로 링크나 스위치 집합의 어떤 연결을 허락하지 않음으로써 오류 부분을 고립시키는 것이다. 현재는 이 분야에 대한 연구가 활발히 진행되고 있다.

본 논문은 그래프 모델을 기본으로 한다. 이러한 모델에서 분산 메모리 병렬 컴퓨터는 노드를 프로세서로 표현하고 간선을 통신 링크로 표현하는 그래프이다.

그래프 모델에서 오류 허용이란  $n$ 개의 노드를 갖는 대상 그래프가 있을 때  $q$ 개의 노드를 추가한 오류 허용 그래프로부터  $q$ 개의 임의의 노드 집합을 제거했을

때 나머지 노드로써 구성된 그래프가 원 대상 그래프임을 보장하는 것이다. 이러한 그래프 모델은 많은 연구가 있어 왔는데, Hayes[8]은 상호 연결망이 비동질성 내칭 트리일 때 하나 또는 두 개의 오류에 대해서 오류 허용이 가능하게 하는 설계 방법을 설명하기 위해 이 모델을 사용하였으며, Rosenberg[13]은 임의의 그래프  $G$ 가  $m$ 개의 와이어 그룹에 의해서  $m$ 개의 외부 평면 그래프로 분해되며 이 그래프에 대한 오류 허용 그래프를 얻기 위해서 디오게네스(Diogenes) 방법을 신형 배열에 대해서 조사하였다. Wong, Wong[16], 과 PAail[17]는 사이클을 연구하였다. 최근의 연구로는 Dutt와 Hayes가 트리[18], 하이퍼큐브[14], 순환 그래프[14]에 대해서 준 동형 구조 성질을 이용하여 오류 허용을 위한 설계 방법을 설명하였으며 Bruck, Cypher, Hof[19]는 예비 구성 요소의 수를 최소화하기 위해서 순환 그래프를 이용하여 매쉬 및 하이퍼큐브에 대해서 오류 허용을 위한 설계 방법을 설명하였다.

또한 본 논문에서는 1)  $k$ 개 프로세서의 오류 허용을 위해 정확히  $k$ 개 예비 프로세서를 추가하고, 2) 각 프로세서당 최대 통신 링크를 최소화함으로써 오류 허용 설계 구조의 비용을 최소화 하였다. 먼저 1)을 위해서 우리는 순열 방사형 그래프를 해밀턴 사이클을 이용하여 순환 그래프로 변환한 다음 순환 그래프의 준 동형 성질을 사용하여 설명하였으며 다음으로 2)에 대해서는 “대각-반 대각”이라는 새로운 방법을 사용하여 순환 그래프로 변환된 순열 방사형 그래프의 노드 당 최대 간선의 수를 최소화하였다.

이후의 논문 구성은 2장에서 순열 방사형 그래프 및 관련 용어에 대해서 살펴보고 또한 이 논문의 기본 개념이 되는 순환 그래프를 살펴본다. 3장에서는 순열 방사형 그래프를 순환 그래프로 변환하기 위해서 해밀턴 사이클 특성을 이용하여 모든 노드를 한 위치에 배치하고 그것으로부터 각 노드당 간선의 수를 최소화할 수 있는 방법을 살펴 본다. 4장은 결론이다.

## 2. 순열방사형 그래프와 순환 그래프

본 장에서는 순열 방사형 그래프의 정의와 특성을 연구하고, 필요한 용어를 정의한다. 그런 다음 오류 허용 그래프의 생성에 기본 개념이 되는 순환 그래프와 그것의 응용 방법에 대해서 살펴보겠다.

2.1 순열 방사형 그래프

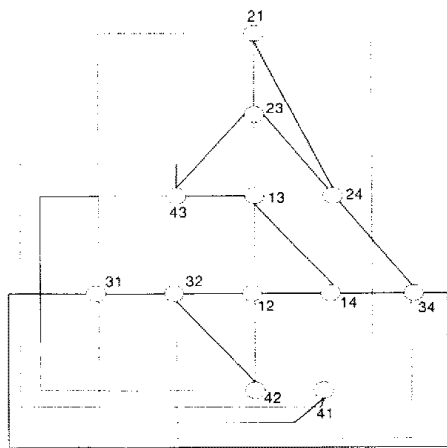
$1 \leq k \leq n-1$ 을 만족하는 두 정수  $n$ 과  $k$ 라고 하고 집합  $\langle n \rangle$ 과  $\langle k \rangle$ 는  $\langle n \rangle = \{1, 2, \dots, n\}$ ,  $\langle k \rangle = \{1, 2, \dots, k\}$ 라고 하자. 또한 집합  $\langle n \rangle$ 에 있는  $n$ 개의 순열(permutation)들의 집합 중에서 한번에  $k$ 를 취하는 순열의 집합을  $p \binom{n}{k}$ 라 하자. 한 순열  $p$ 에 있는  $k$ 개의 구성 요소들은  $p_1, p_2, \dots, p_k$  표시하고  $p = p_1, p_2, \dots, p_k$  라고 쓴다.

**정의 2.1 :**  $(n, k)$ -순열방사형 그래프  $A_{nk} = (V, E)$ 는 다음과 같이 구성된다.

$$V = \{ p_1, p_2, \dots, p_k \mid \langle n \rangle \text{의 } p_i \text{ 그리고 } i, j \text{일 때 } p_i \neq p_j \}$$

$$E = \{ (p, q) \mid p, q \in V \text{의 원소, } \langle k \rangle \text{에 있는 } i \text{에 대해서 } j \neq 1 \text{일 때 } p_i \neq q_i, p_i = q_j \}$$

즉,  $A_{nk}$ 의 노드들은 집합  $\langle n \rangle$ 의  $n$ 개의 구성 요소에서  $k$ 개의 구성 요소의 배열이고  $A_{nk}$ 의 간선들은  $k$ 위치 중에서 정확히 한 위치가 다른 두 배열사이를 연결한다. 이때 위치  $i$ 만 다른 두 배열을 연결하는  $A_{nk}$ 의 간선을  $i$ -간선이라 한다.  $n$ 과  $k$ 의 모든 값에 대한  $A_{nk}$ 는  $\frac{n!}{(n-k)!}$ 의 노드와  $k(n-k)$ 의 간선,  $\lfloor \frac{3k}{2} \rfloor$ 의 직경을 갖는 정규(regular)그래프이다. 또한 순열방사형 그래프가  $A_{n,n-1}$ 이면  $n$ 차원 방사형 그래프와 동형이고  $A_{n-1}$ 은  $n$ 개의 노드를 갖는 완전 그래프와 동형이다. 그러한 순열 방사형 그래프의 예가 그림 1에 보이고 있다.



(그림 1) 순열 방사형 그래프의 예( $A_{4,2}$ )  
(Fig. 1) An example( $A_{4,2}$ ) of arrangement graph

$A_{nk}$ 에서  $i$ 와  $j(1 \leq i \leq n, 1 \leq j \leq k)$ 가 정해졌을 때  $j$ 번째 위치에 구성 요소  $i$ 를 갖는  $A_{nk}$ 상의 노드 수는  $\frac{(n-1)!}{(n-k)!}$  개다. 이 노드들은  $A_{n-1,k-1}$  그래프에 대해서 동일한 방법으로 연결된다.  $A_{nk}$ 에서  $j$ 번째 위치에 구성 요소  $i$ 를 갖는 모든 노드를  $i_j$ 로 나타낸다.  $j$ 번째 위치를 고정함으로써 만들어지는  $1_j, 2_j, \dots, n_j$ 는 서로 독립적인 노드의 집합으로써  $A_{nk}$ 에 있는 노드를 분할하는 것과 같은 효과를 갖는다.

$A_{nk}$ 를  $n$ 개의  $A_{n-1,k-1}$ 으로 나누는 것은  $k$ 개의 다른 방법으로 수행할 수 있다. 따라서  $A_{nk}$ 는  $1 \leq p \leq k-1$ 에 대해서  $\frac{n!}{p!(n-p)!}$  가지의 방법으로  $\frac{n!}{(n-p)!}$  개의  $A_{n-p,k-p}$ 로 나눌 수 있으며 결국 어떤  $p(1 \leq p \leq k-1)$ 에 대해서도  $A_{nk}$ 는  $\binom{k}{p} \left( \frac{n!}{(n-p)!} \right)$ 개의  $A_{n-p,k-p}$ 를 얻을 수 있다.

그래프 내부 준 동형 구조(automorphism) [45]를 사용하여 기본적인 용어를 정의하면, 두 그래프  $G$ 와  $H$ 가 존재할 때  $G$ 의 간선집합  $\{x, y\} \in E(G)$ 이  $H$ 의 간선집합에 대해서  $\{\varphi(x), \varphi(y)\} \in E(H)$ 이 성립될 수 있는  $G$ 에서  $H$ 로의 매핑  $\varphi(\varphi: V(G) \rightarrow V(H))$ 가 존재한다면 동형이라고 할 수 있으며 여기서  $\varphi$ 는  $G$ 에서  $H$ 로의 동형 구조이다.  $G$ 에서 자기 자신  $G$ 로의 동형 구조를  $G$ 의 내부 준 동형 구조라고 한다.

오류 허용 설계의 목적은 기본 그래프라고 하는 어떤 구조  $G$ 를 보존하는 것이다. 즉 임의의  $G$ 에  $q$ 개의 예비 노드를 추가함으로써 그래프  $G' [q, G]$ 를 생성하여 어떠한 오류 형태( $Q = \{q_1, q_2, \dots, q_t\} (t \leq q)$ )에 대해서도  $G' - Q$ 가  $G$ 에 대해서 부그래프 동형인 것을 의미한다.  $Q$  오류 형태라고 하는 것은 최대  $\{q_1, q_2, \dots, q_t\}$ 로 구성되는 모든 오류 형태를 포함한다. 그리고 만약 임의의 그래프  $G$ 에  $t$ 개의 오류가 발생했다고 한다면 필요한 예비 노드  $q$ 의 수는  $t \leq q$ 를 만족하여야 할 것이다. 따라서  $t$ 개의 오류를 다른 정상 노드로 대체하기 위해서는 최소한  $q$ 개의 노드가 필요하며 이것이 요구되는 예비 노드의 하한이 된다. 만약 이러한 예비 노드에 대한 하한 조건인 최소 요구 노드의 수에 대한 제한을 경감시킨다면 좀더 다양한 형태의 오류 허용 설계가 가능하다.

2.2 순환 그래프

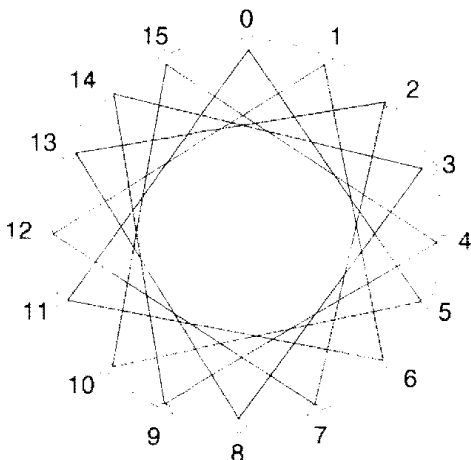
이 장에서는 순환 그래프라고 하는 그래프 부류에 대해서 설명한다.

**정의 2.2 :**  $p$ 가 양의 정수이고  $S$ 가 1에서  $\frac{(p+1)}{2}$  사이에 있는 정수들의 집합

$$(S = \{1 < a_1, a_2, \dots, a_k < \frac{(p+1)}{2}\})$$

이라고 할 때, 연결 집합  $S$ 를 갖는  $p$  노드 순환 그래프  $(C_{16})$ 는  $p$ 개의 노드들로 구성되어 있으며,  $C_{16}$ 의 각 노드들은 0에서  $p-1$ 사이의 유일한 레이블을 갖는다.

각 노드  $i$ 는 연결 집합  $S$ 의 원소  $s$ 에 대해서  $(i \pm s) \pmod p$ 인 모든 노드와 연결된다. 이때 연결 집합  $S$ 의 원소들을 분기된다. 그림 2는 순환 그래프에 대한 간단한 예를 보이고 있다. 이때  $a_k = \frac{p}{2}$ 이면 순환 그래프의 각 노드들은 동일하게  $2k$ 개의 간선을 갖으며,  $a_k = \frac{p}{2}$ 이면  $2k - 1$ 개의 간선을 갖는다. 한 노드로부터 임의의 노드와 연결된 간선으로부터 생성된 현 중에서 작은쪽 현위에 있는 노드의 수를 순환거리라 한다. 예를 들면 그림 2에서 노드 0과 노드 5를 연결하는 간선은 전체 원을 두 개의 부분으로 나누게 되는데 이때 작은 영역부분의 현위에 있는 노드(즉 1, 2, 3, 4)들의 수가 순환거리가 되며 이때 순환거리 값은 4이다. 이것을 순환거리(0, 5) = 4로 표현한다.



(그림 2) 16개의 노드와 변위가 1과 5인 순환 그래프 (Fig. 2) Circulant graph with 16 nodes and displacement on 1 and 5

**(정의 2.3)**  $S$ 를 정수 집합이라 하고  $q$ 가 음이 아닌 정수라고 할 때  $q$ 에 대한  $S$ 의 확장 연결집합(확장연결  $(S, q)$ )은 다음과 같은 집합  $T$ 로 나타낼 수 있다.

$$T = \bigcup_{s \in S} \{s, s+1, \dots, s+q\}$$

이때 위의 집합은 1확장 연결( $S, q$ ) 또는  $(q+1)S$ 인 관계를 갖는다.

다음 정리는 Dutt와 Hayes[14]에 의해서 증명된 결과이다.

**(정리 2.1)[14]**  $n$ 이 양수이고,  $S$ 가 0보다 크고  $\frac{(p+1)}{2}$  보다 작은 값을 갖는 정수의 집합이라고 하자. 또한  $q$ 가 음이 아닌 정수라고 하고  $T =$  확장연결  $(S, q)$ 이라 한다면 순환 그래프  $C_{n, T}$ 는 "오류허용( $q, C_{n, S}$ )"이다.

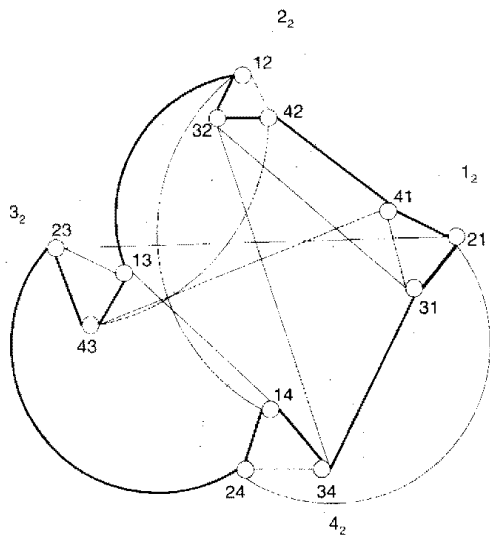
정리 2.1의 기본 개념은  $C_{n, T}$ 에 임의의  $q$ 개의 오류 노드들이 주어졌을 때, 오류 허용 그래프  $C_{n, q, T}$ 의 정상 노드 부분을 이용하여 원래의 그래프인  $C_{n, S}$ 를 부그래프로 임베딩할 수 있다고 하는 것으로 대상 그래프가 순환 그래프일 때 오류 허용 그래프를 생성할 수 있는 일반적인 기법이다. 따라서 본 연구에서도 오류 허용 순열 방사형 그래프를 생성하기 위해서 위의 정리 2.1을 사용하고자 한다. 먼저 순열 방사형 그래프를 순환 그래프로 변환하고 다음 오류 허용 순열 방사형 그래프를 생성하기 위해서 정리 2.1을 이용할 것이다.

3. 순열 방사형 그래프의 순환 그래프화 연구

본 장에서는 앞에서 설명한 정리 2.1을 이용하여 오류 허용 순열 방사형 그래프를 위한 수퍼 그래프를 생성하기 위해서, 주어진 순열 방사형 그래프를 순환 그래프로 변환하는 방법을 설명하고자 한다. 먼저 개략적으로 살펴보면, 첫째 주어진 그래프를 순환 그래프로 변환하기 위해 해밀턴 성질을 이용하여 하나의 원 위에 모든 노드를 배치한다. 그런 다음 구해진 결과를 이용해서 순환 그래프를 생성한다. 이때 순환 그래프의 간선의 수를 최소화하기 위해서 대각-반 대각 순서 방법을 이용하여 노드를 재배치한다. 마지막으로 정리 2.1을 이용하여 오류 허용 순열 방사형 그래프를 생성한다.

3.1  $A_{n,k}$ 의 해밀턴 사이클  $H_{(n-k)}^m$

순열 방사형 그래프  $A_{n,k}$ 를 순환 그래프로 변환하기 위해서는  $A_{n,k}$ 의 모든  $\frac{n!}{(n-k)!}$  개 원소를 포함하는 한 사이클이 존재함을 보이면 된다. 이것은  $A_{n,k}$ 가 해밀턴 그래프임을 증명하는 것과 같다. 순열 방사형 그래프  $A_{n,k}$ 는 만약 모든 노드를 포함하는 한 사이클  $H_{(n-k)}^m$ 가 존재하면 해밀턴이다. 이때  $H_{(n-k)}^m$ 를 해밀턴 사이클이라고 한다. 이제 우리는 순열 방사형 그래프  $A_{n,k}$ 를 순환 그래프로 변환하기 위하여 해밀턴 사이클  $H_{(n-k)}^m$ 를 찾아야 하는데, 이것은 [6, 7]에서 사용하는 유사한 방법을 사용한다.



(그림 3)  $A_{3,2}$ 의 한 해밀턴 순서의 예  
(Fig. 3) An example of hamiltonian order for  $A_{3,2}$

$H_{(n-k)}^m$ 는  $A_{n,k}$ 를 임의의  $j$ 번째 위치에 특정한 값을 고정시킨  $A_{n-1,k}$ 의  $1, 2, \dots, n_j$ 로 구성되는 총  $n$ 개의 그룹으로 나누는 방법을 반복적으로 사용하여 본 연구가 설계되고, 수행된 것이다. 다음 단계로서  $A_{n,k}$ 의 모든 노드를 포함하는 해밀턴 사이클을 찾는 방법에 대해서 구체적으로 연구를 수행하며, 해밀턴 사이클을 찾기 위해서는 주어진 그래프를 그룹화하는 분할 단계와 그 분할된 부 그래프로부터 해밀턴 사이클을 찾는 단계와 해밀턴 사이클을 결합하는 단계로 나누어 수행한다. 첫째 분할 단계에서는 본 연구에서 이미 설명한

것처럼 특정한 위치에 특정한 값을 고정시켜 독립적인 노드의 집합으로 노드를 분할한다. 즉  $A_{n,k}$ 가 있을 때  $k$ 번째 위치에 특정한 값을 고정시킨  $1, 2, \dots, n_k$ 으로 분할되어질 수 있으며 이 분할된 각 그룹들은 곧  $A_{n-1,k-1}$ 을 의미한다. 계속해서  $n$ 개의  $A_{n-1,k-1}$ 으로부터  $k-1$ 번째 위치에 특정한 값을 고정시킨  $1, 2, \dots, n-1, k-1$  ( $n_k$ 인 경우)로 분할하여  $n-1$ 개의  $A_{n-2,k-2}$ 를 구한다. 이러한 과정을 각 부그룹에 대해서 반복적으로 수행하여 각 그룹이  $A_{n-2,k-2}$ 이 될 때까지 수행한다. 예를 들면  $A_{6,1}$ 는  $1, 2, 3, 4, 5, 6$ 인 6개의  $A_{5,0}$ 으로 그룹화 할 수 있고  $A_{5,0}$ 중  $6_1$ 는 다시  $1, 2, 3, 4, 5$ 인 5개의  $A_{4,0}$ 으로 그룹화 할 수 있다. 여기서  $A_{n-k,2}$ 는 해밀턴 사이클을 구할 수 있는 가장 기본적인 부그룹이다. 다음 단계는 분할로부터 생성된 부 그룹에 대해서 해밀턴 사이클을 찾아내는 것으로 이해를 쉽게 하기 위해서 예를 들어 먼저 설명 하겠다. 그림 3을  $A_{3,2}$ 의 부 그래프중 3번째 위치에 5를 고정시킨 하나의  $A_{4,2}$ 라고 할 때 2번째 위치에 특정한 값을 고정시킨 4개의 부 그룹  $1_2, 2_2, 3_2, 4_2$ 로 나눌 수 있다. 이때 간선은 각 그룹과 그룹의 노드를 상호 연결하는 외부 간선과 그룹 내부에서 노드와 노드를 연결하는 외부 간선으로 나눌 수 있다. 최초 가장 낮은 번호를 갖는 그룹  $1_2$ 의 가장 큰 레이블을 갖는 노드 41을 찾고 41로부터 그 다음으로 낮은 번호 그룹  $2_2$ 과 연결되는 간선  $4_2$ 를 찾는다. 그 다음  $2_2$ 과  $3_2$ 의 연결은 앞 그룹의 번호를 레이블의 첫 위치 값으로 갖는 노드를 선택한다.  $2_2$ 의 시작점은 12가 된다. 끝점은 12와 연결된  $3_2$ 의 노드로 13이 된다. 계속해서  $3_2$ 과  $4_2$  그리고  $4_2$ 과  $1_2$ 에 대해서 위와 같은 방법을 반복 수행하여 외부간선 집합  $\{(41, 42), (12, 13), (23, 24), (34, 31)\}$ 을 구한다. 그런 다음 각 그룹 별로 외부 간선의 집합에서 구한 노드를 양끝으로 하는 내부 간선 집합  $\{(41, 42), (12, 13), (23, 24), (34, 31)\}, \{(24, 14), (14, 34)\}, \{(31, 21), (21, 41)\}$ 을 구한다. 이 내부 간선은 외부 간선에 의해서 알려진 노드를 양 끝점으로 그 그룹 속에 있는 모든 노드의 순서를 찾는 것으로 간단히 순환 올림 순서대로 구한다. 그런 다음 이렇게 구해진 내부 간선 집합과 외부 간선 집합을 레이블에 따라 순차적으로 병합한  $\langle 41, 42, 32, 12, 13, 43, 23, 24, 14, 34, 31, 21 \rangle$ 이  $A_{4,2}$ 의 모든 노드를 포함한 해밀턴 사이클이 된다. 이때

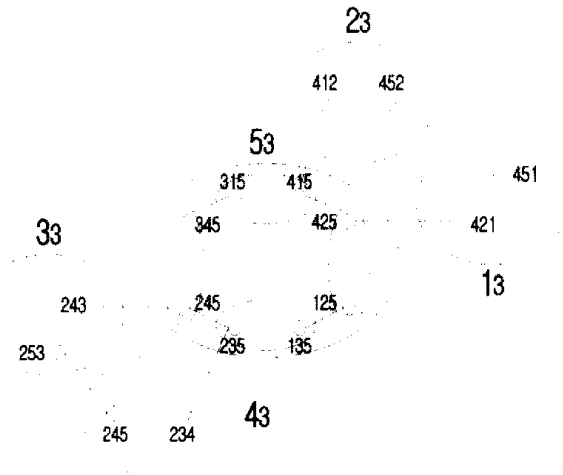
외부 간선  $e_1$ 과  $e_2$ , 그리고  $e_3$ 과  $e_4$ 를 연결한 간선 (41,42), (23,24)를 특히 외부 확장 간선이라고 한다. 그러나  $A_{5,3}$ 을 제외한 모든 그룹은 마지막 외부 간선이 외부 확장 간선이 된다. 외부 확장 간선은 다음 단계의 해밀턴 사이클 확장에서 해밀턴 사이클들을 연결하는 연결고리 역할을 수행한다. 한편 다른 모든 부 그래프의 동일 레벨에서 위와 같은 방법을 사용하여 동시에 해밀턴 사이클을 구할 수 있다. 알고리즘 FIND\_BASIC\_HAMILTONIAN은 위의 과정을 일반화한 것이다.

**PROCEDURE FIND\_HAMILTONIAN( $A_{n, k+2}$ )**

```

begin
  for {  $i_1, \dots, (n-k+3)$  } do
  {
    for {  $i_2, \dots, (n-k+2)$  }, 각 레이블 중  $n-k+3$  제외 do
      { /* 외부 간선을 찾는다 */
        외부간선 := {(n-k+2)i, (n-k+2)(i+1),
                    (i+1), (i+2)}, ..., {(n-k)(n-k+1), (n-k)
                    (n-k+2)}, {(n-k+1)(n-k+2), (n-k+1)i});
        /* 내부 간선을 찾는다 */
        내부간선 := {  $E_i = \{(n-k+2)(i+1), \dots, i(i+1), \dots,
        E_{n-k+2} = \{(n-k+1)i, \dots, (n-k+2)i\}$  };
        /* 내부 간선과 외부 간선을 합명한다 */
        해밀턴 사이클 := < (n-k+2)i, (n-k+2)(i+1),
         $E_i, \dots, (n-k+1)i,
        E_{n-k+2} >$ 
      }
    if 그룹이  $(n-k+2)$  이면
    then 외부 확장간선 := {(n-k+2)i, (n-k+2)(i+1),
                        (i+1), (i+2), (i+1)(i+3)} else 외부 확장간선 :=
                        {(n-k+1)(n-k+2), (n-k+1)i});
      }
  }
end;
```

다음 단계는 동일 레벨의 다른 부 그래프의 해밀턴 사이클을 결합하는데 많은 연구가 수행된 것이다. 이러한 결합에는 두 가지 형태의 연결이 존재한다. 하나는 3개의 부 그룹을 연결한 삼각 연결과 4개의 부 그룹을 연결한 사각 연결이 있다. 일반적으로  $A_{n,k}$ 에서  $n$ 이 홀수이면  $\frac{(n-1)}{2}$  개의 삼각연결이 존재하고,  $n$ 이



(그림 4)  $A_{5,3}$ 의 부 그룹의 상호 연결 예  
(Fig. 4) An example of interconnection of subgroup for  $A_{5,3}$

짝수인 경우  $\frac{(n-4)}{2}$  개의 삼각 연결과 하나의 사각 연결이 존재한다. 결합 방법은 가장 큰 번호를 갖는 부 그래프를 중심으로 상호 외부 확장 간선을 이용하여 삼각 또는 사각 연결을 수행한다. 그림 4는 1, 2, 3, 4, 5를 중심으로 연결되는 과정을 보이고 있다. 이러한 결합은 분할의 역순으로 홀수 그룹결합과 짝수 그룹 결합을 교환하면서 모든 노드를 포함한 해밀턴 사이클을 구할 때까지 반복 수행한다. 이때 외부 확장 간선은 삼각이나 사각 연결에서 부 그룹끼리 연결하는 간선이 된다.

**(보조정리 3.1)**  $A_{n,k}$ 가 해밀턴 사이클을 가질 때 한 해밀턴 사이클 상에 있는 모든 노드는 동일한 성질을 갖는다.

**(증명)** 노드 동질성의 증명은  $H_{(n-k)}$  상에 있는 임의의 두 노드가 있을 때 한 노드에서 또 다른 노드로 매핑되는 동형구조가 존재함을 증명하면 된다.  $H_{(n-k)}$  상에 있는  $A_{n,k}$ 의 두 정점을  $p, q$ 라 하자.  $1 \leq i \leq k$  에 대해  $M(p_i) = q_i$  이고  $(p_1, p_2, \dots, p_k)$ 에 있지 않은  $x$ 에 대해  $M(x) = x$ 를 만족하는  $\langle n \rangle$ 에서의 일대일 매핑을  $M$ 이라 하자.  $P_i$ 의 모든  $t = t_1, t_2, \dots, t_k$ 에 대해  $M(t) = M(t_1), M(t_2), \dots, M(t_k)$ 를 만족하는  $P_i$ 의 일대일 매핑  $M_i$ 는 노드  $p$ 를 노드  $q$ 로 매핑시킨다. 또한  $M_i$ 는 내부 준 동형 구조이다. 왜냐하면 두

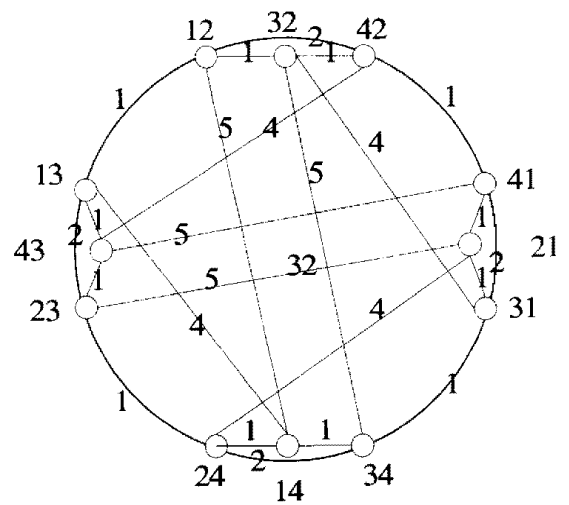
노드  $s$ 와  $t$ 가  $A_{n,k}$ 에서  $i$  인접이면  $M(s)$ 와  $M(t)$ 도  $i$  인접이다.

해밀턴 사이클상의 순서를  $HS = \langle v_1, \dots, v_i, v_j, v_k, v_{i+1}, \dots, v_k \rangle$ 라 하고 이 HS에 예비노드의 집합  $\{s_1, \dots, s_n\}$ 을 포함할 것을 FTHS라 정의해서 이용될 것이다. 예를 들면 그림 3에서  $A_{12}$ 의 한 해밀턴 순서는 굵은 선으로 표시한  $\langle 41, 42, 43, 12, 13, 43, 23, 24, 14, 34, 31, 21 \rangle$  될 수 있다. 또한 해밀턴 순서 그룹(HSG)은 해밀턴 순서 상에서  $j$ 번째 위치의 값이 동일한 그룹들을 표현한 것으로 예를 들면 HSG는  $\{31, 21, 41\}$ ,  $\{42, 32, 12\}$ ,  $\{13, 43, 23\}$ ,  $\{24, 14, 34\}$ 이다. 이때  $|HSG|$ 는 4가 된다.

3.2 간선의 최소화

앞 절에서 우리는 순열 방사형 그래프로부터 해밀턴 사이클 특성을 찾아냄으로써 한 원 위에 모든 노드를 배정할 수 있음을 알았다. 이제 우리는 순열 방사형 그래프를 순환 그래프로 정의하기 위해서는 연결 집합  $S$ 를 찾아야 한다. 앞에서 설명한 것처럼 연결 집합  $S$ 는 각 노드와 연결된 간선의 순환 거리를 모두 합한 것을 말한다. 일반적으로 간선은 정의 2.3에 의하여 예비 노드의 수가  $q$  이고 연결집합이  $S$ 라고 할 때 추가되는 예비 노드에 대한 간선의 수가 최대  $(q+1)|S|$ 만큼 증가한다. 이것은 추가되는 노드당 간선이 2배씩 증가된다는 것을 의미한다. 따라서 예비 노드의 최소화에서처럼 만약 간선이 추가되는 노드당 기존 그래프에 연결하기 위한 간선의 증가율이 추가되는 노드 수와 같다면 즉  $|S|+q$  라면 가장 이상적인 설계라 할 수 있다. 이때  $|S|+q$ 는 추가되는 노드  $q$ 에 대한 최소한의 요구 간선의 증가 분이 된다. 그런데 3.1절에서 얻어진 연결 집합  $S$ 를 가지고 순환 그래프를 생성하면 후에 오류 허용 순열 방사형 그래프를 형성하기 위해서 추가되는 매우 작은 노드 집합  $k$ 에 대해서도 곧 완전 그래프가 된다. 예를 들면 그림 5에는  $A_{4,2}$ 의 각 간선에 대한 순환 거리를 보여주고 있다. 이때 구해진 연결 집합  $S$ 는 1,2,4,5가 된다. 여기에 하나의 노드를 추가하면 연결 집합  $S$ 는 1,2,3,4,5,6이 되어 완전 그래프가 된다. 이렇게 되는 이유는 순열 방사형 그래프를 해밀턴 사이클 특성을 사용하여 순환 그래프화 할 때 노드들이 그룹화 개념으로 구성되어졌기 때문에 각 노드와 그 노드와 연결된 간선들이 임의의 순환 거리를 갖기

때문이다. 즉 연결 집합  $S$ 가 임의로 분포되어 있기 때문이다. 따라서 이러한 연결집합  $S$ 가 어떤 특정한 연결 거리를 중심으로 연속되어 있다면 이러한 문제를 해결할 수 있다. 특히 연결 집합  $S$ 가  $\frac{n}{2}$ 를 중심으로 연속된다면 각 노드에 부가되는 간선의 수를 줄일 수 있을 뿐만 아니라, 추후에 오류 허용 순열 방사형 그래프를 생성할 때 추가되는 노드들에 대한 간선의 수를 최소화함으로써 간선의 증가율을 경감시킬 수 있다.



(그림 5) 순열 방사형 그래프  $A_{4,2}$ 의 순환 거리  $(\{1,2,4,5\})$

(Fig. 5) Circular distance  $(\{1,2,4,5\})$  of arrangement graph  $A_{4,2}$

이제부터 우리는 이러한 간선 증가율을 최소화할 수 있는 대각-반대각 방법을 설명하고자 한다. 그림 6에서 보인 것처럼 해밀턴 순서 그룹들을 구하여 각 그룹내의 원소들은 오름차순으로 정렬  $(a_1, a_2, \dots, a_k | a_k < \dots < c_1, c_2, \dots, c_k | c_k)$ 하여 한 수평선상에 배열한다. 이 과정을 모든 그룹에 대해서 수행한 다음 각 그룹을 그림 6에서처럼 위쪽에서 아래쪽으로 해밀턴 순서에서 얻은 그룹 순서대로 배열한다. 이러한 과정이 완료되면 먼저 이러한 과정을 통해서 다음과 같은 방법으로 최적해를 얻을 수 있도록 연구가 될 것이다.

- 1) 첫 번째 그룹의 처음 원소  $(a_1, a_2, \dots, a_k | a_k)$ 로부터 대각 방향으로 각 그룹당 원소를 하나씩 선택한다. 마지막 열의 원소가 선택되면 2) 마지막 그룹의 처음 위치에 있는 원소  $(p_1, p_2, \dots, p_k | p_k)$ 를 선택한다. 3) 첫 번째 그룹의 두 번째 원소로부터 대각 방향으로 각

그룹에서 원소를 하나씩 선택하고 마지막 열의 원소가 선택되어지면 다시 마지막 그룹 두 번째 원소로부터 반 대각 방향으로 첫 번째 열까지 각 그룹에서 원소를 하나씩 선택한다.

$$\begin{aligned}
 & (1)\text{번째} \quad \left(2\left(\frac{(n-1)!}{(n-k)!}\right)-1\right)\text{번째} \\
 & \left(2\left(\frac{(n-1)!}{(n-k)!}\right)-1\right)\text{번째} \quad a_1, \dots, a_{\lfloor \frac{n-1}{2} \rfloor}, \dots, a_{\lfloor \frac{n-1}{2} \rfloor+1}, \dots, a_{n-1} \\
 & \quad a_{2\lfloor \frac{n-1}{2} \rfloor+1}, \dots, a_{2\lfloor \frac{n-1}{2} \rfloor+2}, \dots, a_{2\lfloor \frac{n-1}{2} \rfloor+2}, \dots, a_{n-2} \\
 & \quad a_{3\lfloor \frac{n-1}{2} \rfloor+1}, \dots, a_{3\lfloor \frac{n-1}{2} \rfloor+2}, \dots, a_{3\lfloor \frac{n-1}{2} \rfloor+2}, \dots, a_{n-1} \\
 & (2)\text{번째} \quad b_1, \dots, b_{\lfloor \frac{n-1}{2} \rfloor}, \dots, b_{\lfloor \frac{n-1}{2} \rfloor+1}, \dots, b_{n-1}
 \end{aligned}$$

(그림 6) 순열 방향형 그래프의 노드들에 대한 순서화 (Fig. 6) Ordering for nodes of arrangement graph

이러한 대각-반 대각 과정을 대각 방향일 때는 아래에서 위쪽으로 모든 그룹의 원소가 선택될 때까지 반복적으로 수행하며 모든 원소가 선택될 때 종료된다. 따라서 모든 원소를 선택하기 위해서는  $2\left(\frac{(n-1)!}{(n-k)!}\right)$  번 위의 방법을 반복하면 된다. 그런 다음 연결 집합에 속한 원소들이 연속되는 값을 갖도록 하기 위해 개선된 해밀턴 순서에 적절한 예비 노드를 추가시킨다. 이것에 대한 상세한 설명은 아래의 보조 정리를 통해서 서술하였다. 따라서 이러한 연결 집합을 갖는 순환 그래프는 매우 효과적으로 오류 허용 순열 방향형 그래프를 쉽게 생성하도록 하여준다.

**(보조정리 3.2)**

HSG가 2보다 큰 양수이고  $\langle l_1, l_2, \dots, l_{\lfloor \frac{n!}{(n-k)!} \rfloor} \rangle$  이  $\lfloor \frac{n!}{(n-k)!} \rfloor$ 인 순서이고  $\{SN\} = |HSG|-3$ 이라고 할 때 순서  $\langle l_1, l_2, \dots, l_{\lfloor \frac{n!}{(n-k)!} \rfloor}, \{SN\} \rangle$ 은  $S = \{|HSG|-1, \dots, \lfloor \frac{H}{2} \rfloor\}$ 인 연결집합을 갖는다.

**(증명)**  $\langle l_1, l_2, \dots, l_{\lfloor \frac{n!}{(n-k)!} \rfloor} \rangle$ 을 대각-반대각 순서로 배열한 순서이고 예비노드 집합  $\{SN\} = |HSG|-3$ 을 추가한 순서 집합이  $\langle l_1, l_2, \dots, l_{\lfloor \frac{n!}{(n-k)!} \rfloor}, \{SN\} \rangle$ 이라고 하자. 또한  $l_i$ 와 연결된 집합을  $\{l_{adj}\}$ 라고 할 때 한 순서 상에서  $|HSG|-1$ 보다 같거나 작은 인접 거리에 있는 임의의 두 노드  $l_i, l_m (0 < l_i, l_m \leq \frac{n!}{(n-k)!})$

의 순환거리가  $(l_i, l_m) \leq |HSG|-1 \leq \lfloor \frac{H}{2} \rfloor$ 임을 보일 것이다.

$|HSG|-1$ 보다 같거나 작은 순서 상의 임의의 두 노드  $l_i$ 와  $l_m$ 이 순환거리  $(l_i, l_m) < |HSG|-1$ 이라면  $l_i$ 의 순열  $p_{i1} p_{i2} \dots p_{ik}$ 와  $l_m$ 의 순열  $p_{m1} p_{m2} \dots p_{mk}$ 가 있을 때 반드시 순열의 임의 위치에서  $p_{ir} = p_{mr} (1 \leq r \leq k)$ 인 경우가 존재하고 또한  $l_i$ 와  $l_m$ 은 인접 노드 집합  $\{l_{adj}\}$ 중의 노드인  $l_{adj}$ 의 순열  $p_{adj1} p_{adj2} \dots p_{adjk}$ 에 대해서  $|HSG|-1$ 내에  $p_{ir}$  (또는  $p_{mr}) = p_{adjr}$ 이 존재한다. 그리고  $|HSG|-1$ 내의  $l_i, l_m$ 을 제외한 노드  $l_x (l_i < l_x < l_m)$ 의 순열  $p_{x1} p_{x2} \dots p_{xk}$ 에 대해서 순열의 임의의 위치에서도  $p_{ir}$  (또는  $p_{mr}) \neq p_{xr}$ 이다. 따라서  $l_i$ 와  $l_m$ 의 순환거리는  $|HSG|-1-1 \leq$  순환거리  $(l_i, l_m) \leq \lfloor \frac{H}{2} \rfloor$ 이며,  $l_i$ 와  $l_m$  및  $\{l_{adj}\}$ 은 모두  $|HSG|-1$ 보다 항상 크다. 다음으로  $|HSG|-1$ 보다 큰 경우에는 어떤 임의의 두 노드  $l_i, l_m$ 에 대해서  $p_{ir} = p_{mr}$  또는  $p_{ir} \neq p_{mr} (1 \leq r \leq k)$ 인 경우가 존재할 수 있는데  $p_{ir} \neq p_{mr}$ 인 경우는 직관적으로 이해될 수 있으며  $p_{ir} = p_{mr}$ 인 경우는 순환거리  $(l_i, l_m) > |HSG|-1$ 이므로 결국 노드간의 순환거리는  $|HSG|-1 \leq$  순환거리  $(l_i, l_m) \leq \lfloor \frac{H}{2} \rfloor$ 안에 존재하게된다. 우리는 지금까지  $|HSG|-1$ 보다 같거나 작은 경우 또는 큰 경우에 대해서 항상 순환거리가  $|HSG|-1$ 에서  $\lfloor \frac{H}{2} \rfloor$ 까지 범위 내에 존재함을 보였으며, 결국 순서  $\langle l_1, l_2, \dots, l_{\lfloor \frac{n!}{(n-k)!} \rfloor}, \{SN\} \rangle$ 은 연결집합이  $S = \{|HSG|-1, \dots, \lfloor \frac{H}{2} \rfloor\}$ 이다.

**(보조정리 3.3)**

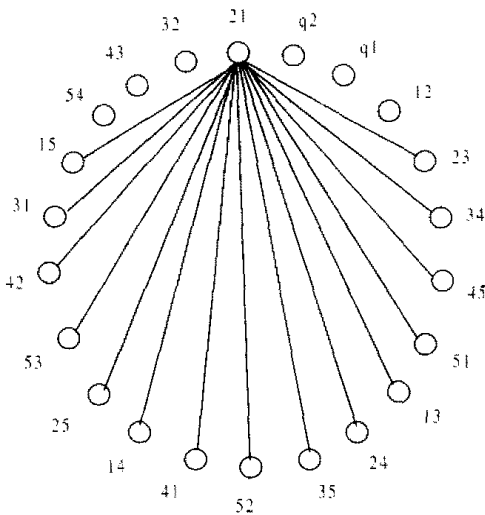
HSG가 2보다 큰 양수이고  $S = \{|HSG|-1, \dots, \lfloor \frac{H}{2} \rfloor\}$ 이라고 할 때  $ANet_{n,k}$ 는 순환 그래프  $C_{\lfloor \frac{n!}{(n-k)!} \rfloor, |HSG|-3, S}$ 의 부분 그래프이다.

**(증명)** 순열 방향형 그래프의 노드와 순환 그래프의 노드가  $\frac{n!}{(n-k)!} \leq \frac{n!}{(n-k)!} + |HSG|-3$ 이고 또한 간선이  $k(n-k) \leq 2$ 일 때 보조정리 5.2에 따라



$A_{n,k}$ 의 모든 노드는 변경된 해밀턴 순서로 순환 그래프  $C_{\frac{n!}{(n-k)!} + HSG - 3, S}$ 에 유일하게 매핑된다.

(그림 7)은 순열 방사형 그래프를 순환 그래프로 변환하기 위해서 3.1절에 의해 해밀턴 사이클을 구하고 간선의 수를 최소화하기 위해서 대각-반대각 방법을 사용하여 새로운 해밀턴 사이클을 구한 다음 보조 정리 3.2와 보조 정리 3.3에 따라 예비 노드를 추가한 결과를 보이고 있다. 이 순환 그래프의 노드



(그림 7) 순열 방사형 그래프  $A_{5,2}$ 의 순환 그래프 (Fig. 7) Circulant graph of arrangement graph  $A_{5,2}$

수는  $22 = 20 + 2$ (추가 노드)이고 연결집합은  $S = \{4, 5, 6, 7, 8, 9, 10, 11\}$ 로서 모든 노드를 표현할 경우 이해가 어렵기 때문에 임의의 노드 21에 대해서만 연결 간선을 보이고 있다.

**(보조정리 3.4)**

$|HSG-1 \leq s \leq \lfloor \frac{d}{2} \rfloor$  이고  $0 \leq x \leq s$ 인 정수를  $HSG, p, s, x$ 라 하자.  $S = \{s-x, s-x+1, \dots, s\}$ 이라 하고  $T =$  확장연결( $S, q$ )라 할 때 순환 그래프  $C_{\frac{n!}{(n-k)!} + HSG + q - 3, T}$ 의 최대 간선의 수는 다음과 같다.

$$d = \begin{cases} q \leq |HSG-3 \text{ 경우} & , 2|S| \\ q > |HSG-3 \text{ 경우} & , 2|S| + q - |HSG| + 3 \end{cases}$$

**(증명)** 먼저  $q \leq |HSG-3$ 인 경우는 보조 정리 5.2에 의한 모든  $0 < q < q+1 < q+2 < \dots \leq |HSG|-3$

에 대해서 확장 연결 집합  $T$ 가  $|HSG|-1 \leq l \leq \lfloor \frac{d}{2} \rfloor$ 로 항상  $S = T$ 이므로 간선  $d$ 는  $2|T| = 2|S|$ 라는 것을 쉽게 알 수 있다. 다음으로  $q > |HSG-3$ 인 경우를 살펴보면  $q = |HSG-2$ 일 때 확장 연결 집합은  $T = \{s-x, s-x+1, \dots, s, s+1\}$ 이며  $q = |HSG-1$ 일 때 확장 연결 집합은  $T = \{s-x, s-x+1, \dots, s, s+1, s+2\}$ 이 된다. 계속해서 매우 큰  $q$ 일 때 확장 연결 집합은  $T = \{s-x, s-x+1, \dots, s, s+1, \dots, s+q-|HSG|+3\}$ 이 되므로 간선은  $2|T|$ 가 될 것이다. 그러나  $T$ 의 집합 원소들이 연속적이므로 정의 5.2에 의해서 간선은  $2|S|$ 와  $q$ 에 의한 간선의 증가 분인  $q-|HSG|+3$ 을 더한 것과 같다. 결국  $q > |HSG-3$ 일 때 간선  $2|S| + q - |HSG| + 3$ 이 된다.

**(정리 3.5)**  $S = \{|HSG-1, \dots, \lfloor \frac{d}{2} \rfloor\}$ 이라 하고  $T =$  확장연결( $S, q$ )라 할 때 순환 그래프  $C_{\frac{n!}{(n-k)!} + HSG + q - 3, T}$ 는 오류 허용( $q, A_{n,k}$ )이다.

**(증명)** 정리 2.1에 의해 그래프  $C_{\frac{n!}{(n-k)!} + HSG + q - 3, T}$ 는 오류 허용( $q, C_{\frac{n!}{(n-k)!} + HSG - 3, S}$ )이다. 또한 보조정리 3.3에 의해 그래프  $A_{n,k}$ 는  $C_{\frac{n!}{(n-k)!} + HSG - 3, S}$ 의 부분 그래프이다. 따라서 그래프  $C_{\frac{n!}{(n-k)!} + HSG + q - 3, T}$ 는 오류 허용( $q, A_{n,k}$ )이다.

우리는 지금까지 오류 허용 순열 방사형 그래프를 설계하기 위해서 순열 방사형 그래프를 순환 그래프로 변환하기 위한 새로운 설계방법을 살펴보았으며 이러한 과정의 결과들을 통해서 다음을 고려할 수 있었다. 첫째는 보조정리 3.2의 연결 집합  $S$ 가 전체 그룹에 의존적임을 보이고 있으므로 만약 동일한 노드 수를 갖는 여러 순열 방사형 그래프 중에서 그룹 수가 많은 즉  $n$ 이 큰 그래프를 선택한다면 연결 집합  $S$ 를 최소화 할 수 있다. 예를 들면  $A_{7,6}$ 과  $A_{10,4}$ 는 모두 5040개의 노드를 갖는 순열 방사형 그래프이다. 이때  $A_{10,4}$ 를 선택하면 연결 집합  $S$  및 간선의 수를 줄일 수 있다. 둘째는 변경된 해밀턴 순서에 추가된 예비 노드들은 오류 허용 순열 방사형 그래프를 설계할 때 간선의 수를 줄이는데 이용되었을 뿐만 아니라 순환 그래프로의 변환이 용이하도록 하였다.

#### 4. 결 론

지금까지 우리는 순열 방사형 그래프에 적합한  $offdms$  오류 허용 설계방법을 제안 하였다. 이러한 오류 허용 설계방법에서는  $k$ 개 프로세서의 오류 허용을 위해 정확히  $k$ 개 예비 프로세서를 추가하고 각 프로세서당 최대 통신 링크를 최소화함으로써 오류 허용 순열 방사형 그래프는 정리 1에 의해서  $q$ 개의 오류를 수용할 수 있으며 간선은  $q \leq |HSG| - 3$ 일 때  $d = 2|S|$ 이고  $q > |HSG| - 3$ 일 때  $d = 2|S| + q - |HSG| + 3$ 을 갖는다. 또한 이것으로부터 간선의 수를 좀 더 감소시킬 수 있는 절충안을 얻을 수 있었다. 이 방법은 순열 방사형 그래프라고 하는 정규 그래프를 중심으로 설명하였으나 비정규 그래프에 대해서도 알고리즘의 작은 변화를 통해서 쉽게 적용되어질 수 있을 것이라고 생각하며 추후의 새로운 그래프에 대한 오류 허용 그래프를 생성할 때 효과적으로 이용될 수 있을 것이라고 기대한다. 앞으로의 연구 방향은 재구성 속도와 신뢰성 향상을 위해서 예비 노드를 지역화하여 재구성하는 분산 재구성 방법에 대한 연구가 있어야 한다.

#### 참 고 문 헌

- [1] Akers, S.B. Harel, D. and Krishnamurthy, B., "The star graph : An attractive to the  $n$  cube," in Proc. Int. Conf. Parallel Processing, 1987, pp. 393-400.
- [2] Akers, S. B. and Krishnamurthy, B., "A group theoretical model for symmetric interconnection networks," in Proc. Int. Conf. Parallel Processing Lett, 1986, pp.216-233.
- [3] Day, K. and Tripathi, A., "Arrangement graphs : A class of generalize star graphs," Inform. Processing Lett., No.42, pp.235-241, 1992.
- [4] Biggs, N., Finite Group of Automorphisms, Cambridge Univ. Press, Cambridge, 1971.
- [5] Yap, H. P., Some Topics in Graph Theory, Cambridge Univ. Press, Cambridge, 1986, pp.96-97.
- [6] Jwo, J. S. Lakshminarayanan, S. and Dhall, S.K., "Embedding of cycles and grids in star graphs," J.Circuit. Syst., Compt. < Vol.1, No.1 pp.43-74, 1991.
- [7] Day, K., and Tripathi, A., "Embedding of cycles in arrangement graphs," IEEE Trans. Comput., Vol.42, No.8, pp.1002-1006, Aug., 1993.
- [8] Hayes, J. P. A graph model for fault tolerant computing systems, IEEE Trans. Comput. C-25 (Sept. 1976), pp.875-883.
- [9] Kwan, C. L., and Todia, S., An optimal 2-fault tolerant realization of symmetric hierarchical tree systems, Networks 12(1982), pp.231-239.
- [10] Raghavendra, C. S., Avizienis, A., and Ercegovic, M. D. Fault Tolerance in binary tree architecture, IEEE Trans. Comput. C-33(June 1984), pp.568-572.
- [11] Bancrjee, P., Kuo, S.Y., and Fuchs, W. K. Reconfigurable cube connected cycles architecture, Proc. Sixteenth Fault Tolerant Computing Symposium, June 1986, pp.286-291.
- [12] Lombardi, F., Negrini, R., Sami, M.G., and Stefanelli, R. Reconfiguration of VLSI arrays : A covering approach. Proc. Seventeenth Fault Tolerant Computing Symposium., pp.251-256, June 1987.
- [13] Resenberg, A. L., The Diogenes approach to testable fault tolerant arrays of processor. IEEE Trans. Comput. C-32(Oct., 1983), pp.902-910
- [14] Dutt, S. and Hayes, J. P., "Designing fault tolerant systems using automorphisms," J. Parallel Distributed Computing, Vol.12, pp.249-268, 1991.
- [15] Rosenberg, A. L., "On designing fault-tolerant VLSI processor arrays." Advance in Computing Research, Vol.2, pp.181-204, 1984.
- [16] Wong, W. W., Wong, C.K., "Minimum  $k$ -Hamiltonian graphs," J. Graph Theory, Vol.8, pp.155-165, 1984.
- [17] Paoli, M., Wong, W.W., Wong, C.K., "Minimum  $k$  Hamiltonian graphs II," J. Graph Theory, Vol.10, pp.79-95, 1986.
- [18] Dutt, S., Hayes, J.P., "On designing and configuring  $k$ -fault-tolerant tree architecture," IEEE Trans. Comput., Vol.C-39, No.4, pp.490-503, 1990.
- [19] Jehoshua B., Robert C., Ching Tien H., "Fault Tolerant Meshes and Hypercubes with Minimal Numbers of Spares," IEEE Trans. Comput., Vol. 42, No.9, pp.1089-1103, 1993.



### 전 문 석

e-mail : mjun@david.soongsil.ac.kr  
1980년 송실대학교 전산과(학사)  
1986년 Univ. of Maryland 전산  
과(석사)  
1988년 Univ. of Maryland 전산  
과(박사)

1989년 Morgan State Univ. 부설 Physical Science  
Lab. 책임 연구원  
1991년~현재 송실대학교 컴퓨터학부 부교수  
1997년 학술진흥 심사위원  
관심분야 : 병렬 알고리즘, 병렬 컴퓨터 구조, 대규모  
집적회로, 병렬처리 이론, 포트 볼러런스



### 이 문 구

e-mail : yeon0330@hotmail.com  
1984년 송실대학교 전산과(학사)  
1993년 이화여자대학교 교육대학  
원 전산학과(석사)  
1996년~현재 송실대학교 대학원  
전산과 박사수료

1997년~현재 명지전문대학 전산과 겸임교수  
관심분야 : 병렬 알고리즘 설계, Multimedia 통신, 정보  
통신 보호, 방화벽 시스템 설계