

# 재귀원형군과 토러스에서 쌍형 다대다 서로소인 경로 커버

김 유상, 박 정흠

가톨릭대학교 컴퓨터공학과  
420-743 경기도 부천시 원미구 역곡 2동 산 43-1  
eusankun@naver.com, j.h.park@catholic.ac.kr  
전화 (02) 2164-{4799, 4366}, 팩스 (02) 2164-4777

## Paired Many-to-Many Disjoint Path Covers in Recursive Circulants and Tori

Eusang Kim and Jung-Heum Park

Department of Computer Science  
The Catholic University of Korea  
Yokkok 2-dong 43-1, Wonmi-gu, Puchon City, Kyonggi-do 420-743  
eusankun@naver.com, j.h.park@catholic.ac.kr  
TEL (02) 2164-{4799, 4366}, FAX (02) 2164-4777

---

\* 본 연구는 2007년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

(KCC2008 우수논문)

# 재귀원형군과 토러스에서 쌍형 다대다 서로소인 경로 커버

## Paired Many-to-Many Disjoint Path Covers in Recursive Circulants and Tori

### 요약

그래프  $G$ 의 쌍형 다대다  $k$ -서로소인 경로 커버(쌍형  $k$ -DPC)는  $k$ 개의 서로 다른 소스-싱크 쌍을 연결하며 그래프에 있는 모든 정점을 지나는  $k$ 개의 서로소인 경로 집합이다. 이 논문에서는 재귀원형군  $G(cd^m, d)$ ,  $d \geq 3$ 과 토러스에서 서로소인 경로 커버를 고려하여, 이 분 그래프가 아니고 분지수가  $\delta$ 인 재귀원형군과 토러스는 고장 요소(정점이나 에지)가  $f$ 개 이하일 때  $f + 2k \leq \delta - 1$ 을 만족하는 임의의  $f$ ,  $k \geq 1$ 에 대하여 쌍형  $k$ -DPC를 가짐을 보인다.

**키워드:** 고장 감내, 서로소인 경로 커버, 상호연결망, 강한 해밀톤 성질, 고장 해밀톤 성질

### Abstract

A *paired many-to-many  $k$ -disjoint path cover (paired  $k$ -DPC)* of a graph  $G$  is a set of  $k$  disjoint paths joining  $k$  distinct source-sink pairs in which each vertex of  $G$  is covered by a path. In this paper, we investigate disjoint path covers in recursive circulants  $G(cd^m, d)$  with  $d \geq 3$  and tori, and show that provided the number of faulty elements (vertices and/or edges) is  $f$  or less, every nonbipartite recursive circulant and torus of degree  $\delta$  has a paired  $k$ -DPC for any  $f$  and  $k \geq 1$  with  $f + 2k \leq \delta - 1$ .

**Key words:** Fault tolerance, disjoint path covers, interconnection networks, strong hamiltonicity, fault-hamiltonicity.

## 1. 서론

노드들 사이의 라우팅(routing)이나 선형 배열(linear arrays)의 임베딩(embedding) 등과 관련하여 여러 상호연결망(interconnection networks)에서 다루는 중요한 문제 중의 하나는 노드가 서로소인 경로(node-disjoint paths)를 찾는 것이다. 노드가 서로소인 경로는 노드들 사이에 효율적인 데이터 라우팅을 위한 병렬 경로로 사용될 수 있다. 또한 노드가 서로소인 경로에서 각 경로는 파이프라인(pipeline) 계산에 이용될 수도 있다. 상호연결망은 자주 그래프로 모델되는데, 정점과 에지는 각각 노드와 통신 링크에 대응한다. 이 논문에서는 노드가 서로소인 경로를 간단히 서로소인 경로라고 부르기로 한다.

서로소인 경로는 다음과 같이 세 가지로 분류할 수 있다: 일대일(*one-to-one*), 일대다(*one-to-many*), 그리고 다대다(*many-to-many*). 연결도(connectivity)가  $k$  이상인 그래프  $G$ 가 있다고 하자. Menger의 정리에 따르면  $G$ 의 서로 다른 두 정점 사이에  $k$ 개의 서로소인 경로가 존재하는데[1], 이들은 일대일 서로소인 경로를 이룬다. Menger의 정리를 확장한 소위 Fan Lemma는  $G$ 의 한 정점에서 서로 다른  $k$ 개의 정점 사이에  $k$ 개의 서로소인 경로가 존재함을 말하고 있고[1], 이 경로들은 일대다 부류가 된다. 마찬가지로 확장하여  $G$ 의 정점 수가  $2k$  이상이면 임의의  $k$ 개의 정점으로부터 다른  $k$ 개의 정점 사이에 다대다 부류에 속하는  $k$ 개의 서로소인 경로가 존재함이 알려져 있다[2].

세 유형의 서로소인 경로가 그래프  $G$ 에 있는 모든 정점을 지나는 경우를 생각해 볼 수 있다. 서로소인 경로 커버(*disjoint path cover; DPC*)는 그래프의 모든 정점을 지나는 서로소인 경로들의 집합을 말한다. 서로소인 경로 커버에 대한 문제는 모든 노드를 완전히 활용(utilization)하는 것이 중요한 응용과 관련된다. 선형 배열의 임베딩에서 커버는 모든 노드가 파이프라인 계산에 참여한다는 것을 의미한다. 다대다 서로소인 경로 커버를 찾는 문제가 가장 일반화된 문제이며, 이 논문에서는 다대다 서로소인 경로 커버를 다룬다.

그래프  $G$ 에서  $S \cap T = \emptyset$ 를 만족하는  $k$ 개의 소스(source)  $S = \{s_1, s_2, \dots, s_k\}$ 와  $k$ 개의 싱크(sink)  $T = \{t_1, t_2, \dots, t_k\}$ 가 주어져 있을 때, 소스와 싱크를 잇는 경로  $P_i (1 \leq i \leq k)$ 들의 집합이 서로소이면서 그래프의 모든 정점을 커버하면, 즉 모든  $i \neq j$ 에 대하여  $V(P_i) \cap V(P_j) = \emptyset$  이고  $\cup_{1 \leq i \leq k} V(P_i) = V(G)$ 이면 이들 경로들의 집합을 *다대다  $k$ -서로소인 경로 커버(many-to-many  $k$ -disjoint path cover)*라고 부른다. 여기서  $V(P_i)$ 는 경로  $P_i$ 에 속한 정점들의 집합이고,  $V(G)$ 는 그래프  $G$ 의 정점 집합이다. 소스와 싱크를 *터미널(terminal)*이라고 부른다. 이때 소스  $s_i$ 가 항상 싱크  $t_i$ 와 짝지어지면 *쌍형 다대다 DPC(paired many-to-many DPC)*라고 부르고, 하나의 소스가 임의의 한 싱크와 짝지어질 수 있으면 *비쌍형 다대다 DPC(unpaired many-to-many DPC)*라고 한다.

서로소인 경로 커버를 찾는 문제는 널리 알려진 해밀톤 경로(hamiltonian path)를 찾는 문제와 밀접하게 관련되어 있다. 해밀톤 경로 문제는 서로소인 경로 커버 문제의 특별한 경우라고 볼 수 있다. 일반적인 그래프에서 임의의 고정된(fixed)  $k$ 에 대하여 일대일, 일대다, 다대다  $k$ -서로소인 경로 커버가 존재하는지를 판별하는 문제는 NP-complete라고 밝혀져

있다[3]. 이 문제들은 모두 주어진 두 정점 사이에 해밀톤 경로가 존재하는지를 판별하는 문제로부터 다항 시간에 변환된다.

한편 고장이 있는 상호 연결망에 선형 배열과 링을 임베딩(embedding)하는 것은 병렬 처리에서 중요한 이슈의 하나이다. 그것은 정점이나 에지에 고장이 있는 그래프에서 길이가 가장 긴 경로와 사이클을 찾는 문제로 모델된다. 그래프  $G$ 에 있는  $f$ 개 혹은 그 이하의 요소(정점이나 에지)에 고장이 발생하여 이들 고장난 요소를 삭제하더라도  $G$ 가 해밀톤 사이클을 가지면,  $G$ 를  $f$ -고장 해밀톤 그래프( $f$ -fault hamiltonian)라고 말한다.  $f$ 개 혹은 그 이하의 요소에 고장이 발생하더라도 해밀톤 연결되어 있으면,  $f$ -고장 해밀톤 연결된 그래프( $f$ -fault hamiltonian-connected)라고 한다.

고장 요소 집합  $F$ 를 가진 그래프  $G$ 에서도 다대다 서로소인 경로 커버를 정의할 수 있다. 주어진  $k$ 개의 소스  $S = \{s_1, s_2, \dots, s_k\}$ 와  $k$ 개의 싱크  $T = \{t_1, t_2, \dots, t_k\}$  ( $S \cap T = \emptyset$ )를 연결하는 쌍형 다대다  $k$ -서로소인 경로 커버는 다음 세 조건을 만족하는  $s_i$ 와  $t_i$  ( $1 \leq i \leq k$ )를 잇는  $k$ 개의 서로소 경로  $P_i$ 들의 집합이다: (a)  $\cup_{1 \leq i \leq k} V(P_i) = V(G) \setminus F$ , (b) 모든  $i \neq j$ 에 대해  $V(P_i) \cap V(P_j) = \emptyset$ , (c) 각 경로  $P_i$ 가 지나가는 모든 에지는 고장이 아니다. 이러한 다대다  $k$ -DPC를  $k$ -DPC $[\{(s_1, t_1), (s_2, t_2), \dots, (s_k, t_k)\} | G, F]$ 로 나타낸다. 비쌍형 다대다  $k$ -DPC도 이와 유사하게 정의할 수 있다.

서로소인 경로 커버에 대하여 연구된 것들로는, 일대일 경로 커버에 대하여 정점이나 에지에 고장이 없는 재귀원형군(recursive circulant)[4-5], 에지 고장이 있는 하이퍼큐브(hypercubes)[6]에 대한 연구가 있고, 하이퍼큐브형 상호 연결망에 대한 일대다 서로소인 경로 커버[7]와 다대다 서로소인 경로 커버[3,8-10]가 발표되어 있다.  $f$ 개 이하의 고장 요소를 가진 모든  $m$ -차원 제한된 HL-graph[11]와 재귀원형군  $G(2^m, 4)$ 에는  $f+2k \leq m-1$ 을 만족하는 임의의  $f, k \geq 1$ 에 대하여  $f$ -고장 쌍형 다대다  $k$ -DPC가 존재하고[3],  $f+k \leq m-2$ 를 만족하는 임의의  $f, k \geq 1$ 에 대하여  $f$ -고장 비쌍형 다대다  $k$ -DPC가 존재한다[8]. 최근에 [9]에서 하이퍼큐브에 대한 쌍형 DPC가 존재할 충분조건이 연구되었고, [10]에서 분지수가 5 이상인 몇몇 하이퍼큐브형 상호연결망에서 길이가 같은 쌍형 다대다 2-DPC가 존재함이 밝혀졌다.  $k$ 개의 경로로 이루어진 일대일 서로소인 경로 커버는  $k^*$ -container라고도 알려져 있다[5-6].

재귀원형군(recursive circulant)은 [12]에서 제안된 다중컴퓨터의 연결망 구조이다. 재귀원형군  $G(N, d)$ 는  $N$ 개의 노드  $\{v_0, v_1, \dots, v_{N-1}\}$ 을 가지고  $a+d^i \equiv b \pmod{N}$ 을 만족하는 정수  $i$  ( $0 \leq i \leq \lceil \log_d N \rceil - 1$ )가 존재하면 두 노드  $v_a, v_b$ 를 잇는 에지가 있다.  $G(N, d)$ 는  $N$ 개의 정점을 가지고  $N$ 보다 작은 모든  $d$ 의 거듭 제곱을 점프로 가지는 circulant 그래프라고 정의할 수도 있다. 이 논문에서는  $N = cd^m, 1 \leq c < d (d \geq 3)$ 으로 제한된 재귀원형군  $G(cd^m, d)$ 를 다룬다.

$m$ -차원 토러스(torus)  $C_{k_1} \times C_{k_2} \times \dots \times C_{k_m}$ 에서 모든  $k_i$  ( $1 \leq i \leq m$ )는 3 이상인 정수이고,  $k_1 k_2 \dots k_m$  개의 노드로 구성되어 있으며, 각 노드는  $0 \leq i_j \leq k_j - 1$  ( $1 \leq j \leq m$ )인

$v_{i_1, i_2, \dots, i_m}$ 으로 나타낸다. 어떤  $p(1 \leq p \leq m)$ 에 대하여  $i'_p \equiv i_p + 1 \pmod{k_p}$ 이고,  $p$ 가 아닌 모든  $j(1 \leq j \neq p \leq m)$ 에 대하여  $i'_j = i_j$ 이면, 두 노드  $v_{i_1, i_2, \dots, i_m}$ 과  $v_{i'_1, i'_2, \dots, i'_m}$ 는 서로 인접하다.  $m$ -차원 토러스는 길이가 3 이상인 사이클  $m$ 개의 그래프 곱(graph product)라고 정의할 수 있다.  $m$ -차원 토러스는 분지수가  $2m$ 인 정규 그래프로서 연결도도  $2m$ 이다.

정점의 수가  $n$ 으로 같은 두 그래프  $G_0$ 와  $G_1$ 이 있을 때,  $G_0$ 에 속한 정점과  $G_1$ 에 속한 정점을 이으면서 서로 인접하지 않도록  $n$ 개의 에지를 추가하여 두 그래프를 하나로 묶을 수 있다. 이때 얻어지는 임의의 한 그래프를  $G_0 \oplus G_1$ 이라고 나타낸다. 각각  $n$ 개의 정점을 가진  $d(\geq 3)$  개의 그래프  $G_0, G_1, \dots, G_{d-1}$ 이 있다고 할 때,  $0 \leq i < d$ 인 모든  $i$ 에 대하여  $G_i$ 와  $G_{i+1 \bmod d}$ 의 모든 쌍에 대해  $\oplus$ 연산을 적용하면,  $nd$ 개의 정점을 가진 그래프를 얻게 된다. 이것을 *사이클 기반 설계(cycle-based construction)*라고 부르고, 이때 생성되는 임의의 그래프를  $\bigoplus_{0 \leq i < d} G_i$ 로 나타낸다. 그래프  $\bigoplus_{0 \leq i < d} G_i$ 에서  $G_i$ 들을 *요소 그래프(component)*라고 부른다. 재귀원형군  $G(cd^m, d)$ 과 토러스  $C_{k_1} \times C_{k_2} \times \dots \times C_{k_m}$ 는 한 단계 낮은 차원의 연결망에 사이클 기반 설계를 적용하여 얻을 수 있다.

이 논문에서는 먼저 각 요소 그래프  $G_i$ 의 고장 쌍형 DPC 성질과 고장 해밀톤 성질을 이용하여 어떻게  $\bigoplus_{0 \leq i < d} G_i$ 의 고장 쌍형 DPC를 설계할 수 있는지를 고찰한다. 그리고 이 결과를 재귀원형군과 토러스에 적용하여 분지수  $\delta$ 가 3 이상이고 이분 그래프가 아닌 경우,  $f+2k \leq \delta-1$ 인 모든  $f, k \geq 1$ 에 대하여  $f$ -고장 쌍형  $k$ -DPC를 설계할 수 있음을 보인다.

## 2. 서로소인 경로 커버 설계

이 절에서는 요소 그래프  $G_0, G_1, \dots, G_{d-1}$ 의 쌍형 다대다 서로소 경로 커버와 고장 해밀톤 성질을 이용하여  $\bigoplus_{0 \leq i < d} G_i$ 의 쌍형 다대다 서로소 경로 커버를 설계할 수 있음을 보인다. 즉, 아래 정리 1을 증명한다. 설계를 간단히 하기 위하여  $G_i$ 는 충분히 많은 정점을 가진다고 가정하고 있다.  $\delta(G)$ 는 그래프  $G$ 의 최소 분지수를 나타낸다.

**정리 1.**  $G_0, G_1, \dots, G_{d-1}$ 을 모두 정점의 수가  $n$ 인 그래프라고 하자. 또한  $\delta = \min_{0 \leq i < d} \delta(G_i) \geq 3$ 이며,  $\delta$ 가 홀수이면  $n \geq 2 \cdot 3^{(\delta-1)/2}$ 이고 짝수이면  $n \geq 3^{\delta/2}$ 라고 하자. 만약 각 요소 그래프  $G_i$ 가  $f+2k \leq \delta-1$ 을 만족하는 임의의  $f, k \geq 1$ 에 대하여  $f$ -고장 쌍형 다대다  $k$ -DPC를 가지고  $\delta-2$ -고장 해밀톤 그래프이면,  $\bigoplus_{0 \leq i < d} G_i$ 는  $f+2k \leq \delta(\bigoplus_{0 \leq i < d} G_i) - 1$ 을 만족하는 임의의  $f, k \geq 2$ 에 대하여  $f$ -고장 쌍형 다대다  $k$ -DPC를 가진다.

이 절에서는  $G_0, G_1, \dots, G_{d-1}$ 에 사이클 기반 설계를 적용하여 얻어지는 그래프  $\bigoplus_{0 \leq i < d} G_i$ 를 간단히  $H$ 로 나타내기로 한다.  $H$ 의 최소 분지수는  $\delta+2$ 가 된다. 앞으로 별

도로 언급하지 않더라도 요소 그래프의 인덱스에서 연산은 모듈로(modulo)  $d$  연산이다. 이후로 쌍형 다대다  $k$ -DPC를 줄여서  $k$ -DPC라고 부른다. 이 논문에서 경로는 정점의 열로 나타내고,  $u$ 와  $v$ 를 잇는 경로를  $u-v$  경로라고 부른다.

일반성을 잃지 않고  $f+2k=\delta(H)-1$ 이라고 가정한다. 즉,  $f+2k=\delta+1$ 을 가정한다.  $f+2k<\delta(H)-1$ 이라면  $\delta(H)-(f+2k)-1$ 개의 가상의 고장 에지를 추가하여  $k$ -DPC를 찾으려면 충분하기 때문이다.

위 정리를 증명하기 위한  $k$ -DPC 설계의 개요는 다음과 같다:

- (a) 소스  $s_p$ 와 싱크  $t_p$ 가 같은 요소 그래프  $G_i$ 에 포함되면  $s_p$ 와  $t_p$ 를 잇는  $s_p-t_p$  경로는  $G_i$ 에 포함되도록 설계한다.
- (b)  $s_p$ 는  $G_i$ 에  $t_p$ 는  $G_j(j \neq i)$ 에 포함되면  $s_p-t_p$  경로는  $G_i, G_{i+1}, G_{i+2}, \dots, G_j$ 의 정점들을 순서로 지나거나 혹은  $G_i, G_{i-1}, G_{i-2}, \dots, G_j$ 의 정점들을 순서로 지나도록 설계한다. 이때 인접한 요소 그래프간의 에지는 하나씩만 이용한다.
- (c) 만약 어떤 경로도 지나지 않는 요소 그래프가 있으면 이들 정점들을 모두 지나도록 이미 설계한 경로를 적절히 변형한다.

$H$ 의 고장 요소 집합을  $F$ 라고 둔다.  $G_i$ 에 속한 고장 요소의 수를  $f_i$ 라고 하자. 고장이 아니고 터미널이 아닌 정점을 *자유 정점*(free vertex)이라고 하고, 두 자유 정점을 잇는 에지가 고장이 아니면 *자유 에지*(free edge)라고 부른다.

**정의 1.** 소스  $s_p$ 는  $G_i$ 에 싱크  $t_p$ 는  $G_j(j \neq i)$ 에 포함된다고 하자. 모든  $l$ 에 대하여  $x_l, y_l$ 을  $G_l$ 의 자유 정점이라고 할 때, 자유 에지의 열  $(y_l, x_{l+1}), (y_{l+1}, x_{l+2}), \dots, (y_{j-1}, x_j)$ 을  $s_p, t_p$  쌍에 대한 *시계 방향 자유 브리지*(free bridge)라고 하고, 자유 에지의 열  $(y_l, x_{l-1}), (y_{l-1}, x_{l-2}), \dots, (y_{j+1}, x_j)$ 을 *반시계 방향 자유 브리지*라고 한다. 자유 브리지에서  $x_l, y_l$ 이 같은 정점일 수도 있고 다른 정점일 수도 있다.  $x_l=y_l$ 이면  $x_l$ (혹은  $y_l$ )를 *접점*(junction)이라 한다.

$G_i$ 에 속한 소스  $s_p$ 와  $G_j(j \neq i)$ 에 속한 싱크  $t_p$ 를 잇는  $s_p-t_p$  경로는  $s_p, t_p$  쌍에 대한 자유 브리지를 이용하여 설계한다. 이때 자유 브리지는 시계 방향 혹은 반시계 방향 어떤 것이 적절한지를 결정할 필요가 있고, 또한 방향이 결정되었더라도 자유 브리지가 지나는 요소 그래프 각각에 대해서 접점을 둘 것인지 혹은 그렇지 않을 것인지도 결정해야 한다.

소스-싱크 쌍이 어떤 요소 그래프에 포함되어 있는지를 그 관계를 나타내기 위하여, 요소 그래프  $G_0, G_1, \dots, G_{d-1}$ 을 단위원(unit circle) 상에 일정한 간격으로 놓인  $d$ 개의 점에 대응시킨다.  $G_i$ 가 어떤 한 점이 대응되면,  $G_{i+1}$ 은 시계 방향으로 만나는 다음 점에 대응시킨다. 서로 다른 요소 그래프에 속한 소스-싱크 쌍은  $s_p$ 는  $G_i$ 에  $t_p$ 는  $G_j(j \neq i)$ 에 포함되어 있다면  $G_i$ 에 대응하는 점과  $G_j$ 에 대응하는 점을 잇는 코드(chord)로 나타낸다. 소스  $s_p$ 와 싱크  $t_p$ 가 같은 요소 그래프  $G_i$ 에 포함되면 코드는  $G_i$ 에 대응하는 한 점이 된다.

(그림 1 참조)

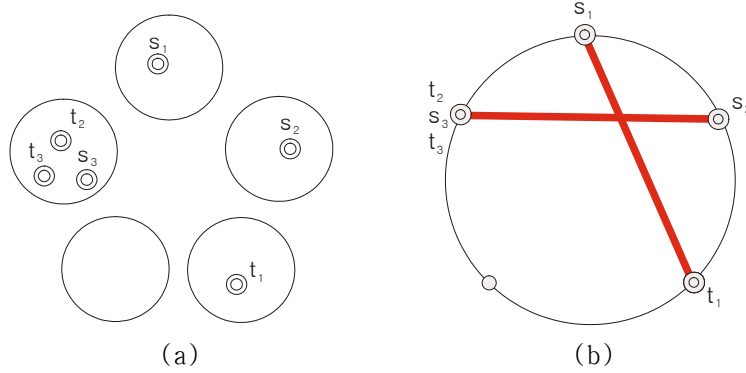


그림 1.  $d = 5, k = 3$

앞으로 2.1절에서 먼저 서로 다른 요소 그래프에 속한 소스-싱크 쌍에 대하여, 방향(시계, 반시계)과 각 요소 그래프에서 접점의 유무가 주어질 때,  $\delta \geq 5$ 이면 조건을 만족하는 자유 브리지가 항상 존재함을 보인다. 2.2와 2.3절에서 소스-싱크 쌍의 위치를 나타내는 코드들의 관계에 따른 각 경우에 대해서  $f$ -고장  $k$ -DPC를 설계한다.

## 2.1 자유 브리지의 존재

$G_i$ 의 임의의 한 정점  $v$ 에서 시작하여  $G_{i+1}, G_{i+2}, \dots, G_j$ 의 정점을 하나씩 순서대로 지나는 길이  $j-i$ 인 경로를  $G_i$ 에서  $G_j$ 까지의 *트랙(track)*이라고 한다.  $G_i$ 에서  $G_j$ 까지  $n$ 개의 트랙이 존재하고 트랙들은 서로소이다. 이들 중 터미널이나 고장 요소를 지나지 않는 트랙을 *자유 트랙(free track)*이라고 한다.

**보조 정리 1.**  $\delta \geq 5$ 라고 하자.  $H$ 의 서로 다른 요소 그래프에 속한 모든 소스-싱크 쌍에 대해서, 방향(시계, 반시계)과 각 요소 그래프에서의 접점 유무가 주어질 때, 조건을 만족하는 자유 브리지는 항상 존재한다. 더구나 자유 브리지는 서로소이다.

**증명.** 소스-싱크 쌍을 하나씩 고려하면서 주어진 조건을 만족하는 자유 브리지가 존재함을 보인다.  $G_i$ 에 속한 소스  $s_p$ 와  $G_j$ 에 속한 싱크  $t_p$ 에 대한 시계방향 자유 브리지를 설계한다. 일반성을 잃지 않고  $i < j$ 임을 가정한다. 반시계 방향 자유 브리지도 마찬가지로 설계된다.  $G_i$ 에서  $G_j$ 까지의 자유 트랙이 둘 이상 존재함을 증명하면 충분하다. 왜냐하면 자유 트랙 둘을 이용하여 한 자유 트랙에서 시작하여  $G_k (i < k < j)$ 에 접점이 있으면 기존의 자유 트랙을 이용하여  $G_{k+1}$  진행하고, 접점이 없으면 다른 자유 트랙을 이용하여  $G_{k+1}$  진행하면 되기 때문이다. 자유 트랙의 후보 수는  $n$ 이다. 트랙을 자유 트랙이 되지 못하도록 막는 방해 요소는 고장 요소 최대  $f$ 개, 터미널 최대  $2k$ 개, 그리고 이미 설계한 다른 소스-싱크 쌍에 대한 자유 브리지가 있다. 다른 소스-싱크 쌍에 대한 자유 브리지는 각각 최대 4개의 트랙을 무효화하므로 방해 요소는 최대  $4(k-1)$ 개이다. ( $G_{i+2}$ 에서  $G_{j-2}$ 까지 반시계 방향 자유 브리지를 생각해 보라.) 이제  $n \geq f + 2k + 4(k-1) + 2$ 임을 보인다. 그런데  $f + 2k = \delta(H) - 1 = \delta + 1$ 이므로  $f + 2k + 4(k-1) + 2 = f + 3(\delta + 1 - f) - 2 \leq 3\delta + 1$ 이다.  $\delta$ 가 홀수이

면  $n \geq 2 \cdot 3^{(\delta-1)/2}$ 이고 짝수이면  $n \geq 3^{\delta/2}$ 이므로, 모든  $\delta \geq 5$ 에 대하여  $n \geq 3\delta+1$ 임을 수학적 귀납법으로 쉽게 증명할 수 있고 그 증명은 생략한다. 이것으로 보조 정리의 증명이 끝났다.  $\square$

## 2.2 교차하지 않는 코드 쌍이 존재하는 경우

각각  $s_1, t_1$  쌍과  $s_2, t_2$  쌍에 대응하는 코드  $L_1, L_2$ 가 서로 교차하지 않는다고 가정하자. 이 때 코드의 길이가 0일수도 있다.  $L_1, L_2$ 가 교차하지 않으므로 단위원의 원주 상에 끝점이  $L_1$ 과 같은 호와 끝점이  $L_2$ 와 같은 호를 서로 교차하지 않게 그릴 수 있고, 이들에 대응되는 자유 브리지를 두면 두 자유 브리지가 동시에 지나는 요소 그래프는 존재하지 않게 된다. 어떤 요소 그래프  $G_i$ 에 속한 한 정점  $v$ 에 대해서  $v$ 에 인접하면서  $G_{i+1}$ 에 속한 정점을  $v^+$ 로 나타내고,  $v$ 에 인접한  $G_{i-1}$ 의 정점을  $v^-$ 로 나타낸다.  $v$ 에 인접하면서  $G_i$ 에 속하지 않는 정점은 항상 둘이다.

### Procedure DPC-A

1. 교차하지 않는 두 코드에 대응하는 자유 브리지를 찾는데, 어떤 요소 그래프도 두 자유 브리지가 동시에 지나지 않도록 한다.
2. 각 소스-싱크 쌍에 대한 자유 브리지를 만들 때, 터미널이 있거나 혹은 이미 다른 자유 브리지가 지나가고 있는 요소 그래프에는 항상 접점을 두도록 한다.
3. 나머지 소스-싱크 쌍들은 임의의 방향으로 자유 브리지를 둔다. 물론 위 단계 2의 조건을 만족하도록 설계한다.
4. 터미널을 가지거나 자유 브리지가 지나는 요소 그래프  $G_i$  각각에 대해서,  $G_i$ 에 속한 접점을 가상 고장으로 두고  $G_i$ 에 속한 모든 소스-싱크 쌍  $s_p-t_p$ , 자유 브리지 끝점과 터미널쌍  $s_q-y_q$ 와  $x_r-t_r$ , 그리고 자유 브리지에 속한 접점이 아닌 두 자유 정점쌍  $x_l-y_l$ 를 잇는 DPC를 찾는다.
5. 터미널을 가지지 않고 어떤 자유 브리지도 지나지 않는 요소 그래프가 있으면, 이미 DPC를 이미 찾은 요소 그래프와 인접한 것부터 고려하여 다음을 반복한다.  $G_i$ 에서 DPC를 찾았고  $G_{i+1}$ 에 DPC를 찾지 않았다고 하자.  $G_i$ 의 DPC가 지나는 에지  $(x,y)$  중에서 각각  $x, y$ 에 인접한  $G_{i+1}$ 의 정점  $x^+, y^+$ 가 고장이 아니며  $(x,x^+), (y,y^+)$ 도 고장이 아닌 것을 찾는다.  $(x,y)$ 를  $(x,x^+), (y,y^+)$  그리고  $G_{i+1}$ 에서  $x^+$ 와  $y^+$ 을 잇는 해밀톤 경로로 대체한다.

아래 그림 2는 그림 1(a)와 같이 주어진 소스와 싱크를 잇는 서로소인 경로 커버를 설계하는 과정을 보여준다.  $s_1, t_1$  쌍과  $s_3, t_3$  쌍에 대응하는 코드가 서로 교차하지 않으므로 Procedure DPC-A를 적용한다.



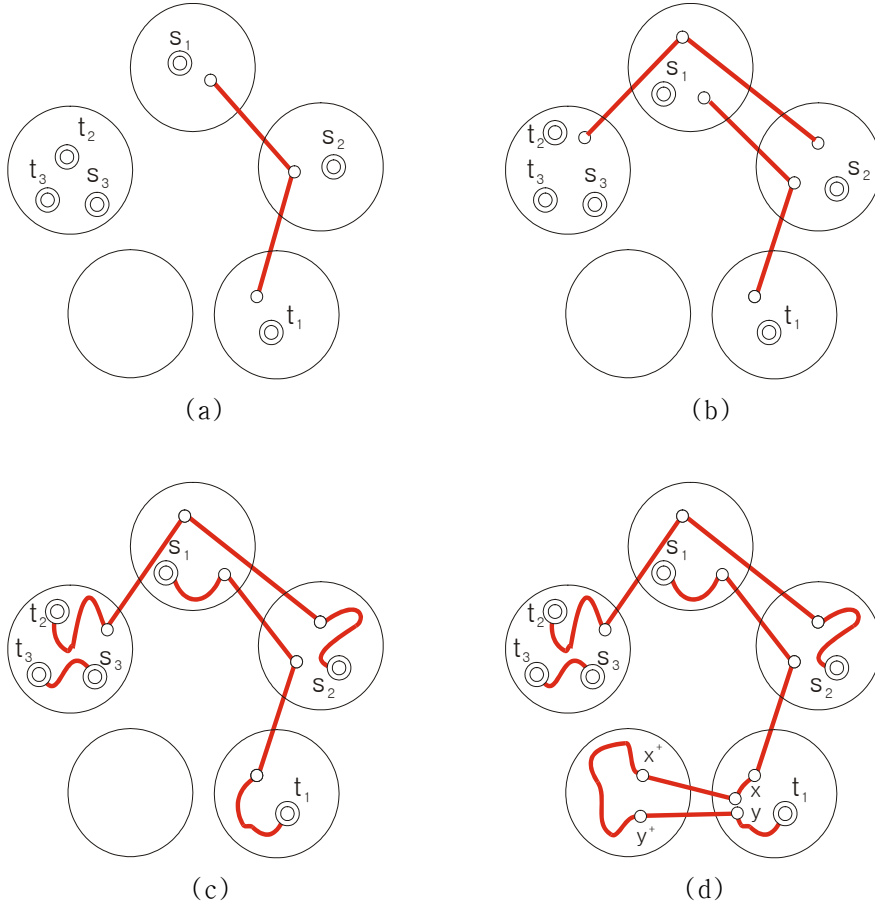


그림 2. Procedure DPC-A의 동작 예

**보조 정리 2.** 교차하지 않는 코드 쌍이 존재하는 경우 Procedure DPC-A는 항상  $f$ -고장  $k$ -DPC를 설계한다.

**증명.**  $\delta \geq 5$ 인 경우에는 보조 정리 1에 의해서 모든 소스-싱크 쌍에 대한 자유 브리지가 존재한다.  $\delta = 3, 4$ 일 경우는  $k=2$ 가 되며 각 요소 그래프에는 최대 하나의 자유 브리지만 지난다는 사실을 이용하여 자유 브리지가 존재함을 쉽게 증명할 수 있다. 그 증명은 여기에서 생략한다. 각 요소 그래프에는 최대  $f$ 개의 고장이 존재하며, 접점의 수를  $f'$ 개라고 하고 찾아야 하는 서로소인 경로의 수를  $k'$ 개라고 하면  $f' + k' \leq k - 1$ 이 성립한다. 따라서  $f + f' + 2k' \leq f + 2(k - 1) = \delta - 1$ 이 되어  $f + f'$ -고장  $k'$ -DPC가 존재하게 된다. 각 요소 그래프의 DPC와 자유 브리지를 모두 연결하면 터미널을 가지거나 자유 브리지가 지나는 요소 그래프로 유도된(induced) 부그래프의  $f$ -고장  $k$ -DPC가 설계된다. 이제 단계 5의 조건을 만족하는 에지  $(x, y)$ 의 존재만 보이면 충분하다. 그런 에지의 후보 수는 최소한  $n - f_i - k$ 개이고, 후보를 방해하는 요소는 고장 요소뿐이므로 최대  $f - f_i$ 개이다. 각 방해 요소가 최대 두 후보를 무력화시키므로, 남은 후보의 수는 최소한  $n - f_i - k - 2(f - f_i) \geq n - 2f - k = n - 2(f + 2k) + 3k \geq n - 2(\delta + 1) + 6 \geq n - 2\delta + 4 > 0$ 이다. 임의의  $\delta \geq 3$ 에 대하여  $n > 2\delta - 4$  임은 수학적 귀납법으로 쉽게 증명된다.  $\square$

이후로 2절에서 자유 브리지를 만들 때 항상 단계 2와 같은 방식으로 접점을 두며, 터미널을 가지지 않고 어떤 자유 브리지도 지나지 않는 요소 그래프가 있다면 단계 5와 같은 방식으로 처리한다. 앞으로 DPC를 설계하는 procedure를 기술할 때 단계 2와 5를 생략하기로 한다.

### 2.3 모든 코드 쌍이 교차하는 경우

모든 코드쌍이 서로 교차하는 경우는 아래와 같이 3가지 경우로 나누어 볼 수 있다:  
 경우 A: 그림 3(a)와 같이 코드 끝점은 아니고 내부에서만 교차하는 쌍이 존재하는 경우,  
 경우 B: 내부에서만 교차하는 쌍이 없고, 그림 (b), (c), (d)와 같이 코드의 끝점에서만 교차하는 쌍이 존재하는 경우, 그리고  
 경우 C: 그림 (e)와 같이 모든 코드가 끝점과 내부에서 동시에 교차하는, 즉 모든 코드가 서로 같고 그 코드의 길이가 0이 아닌 경우이다.

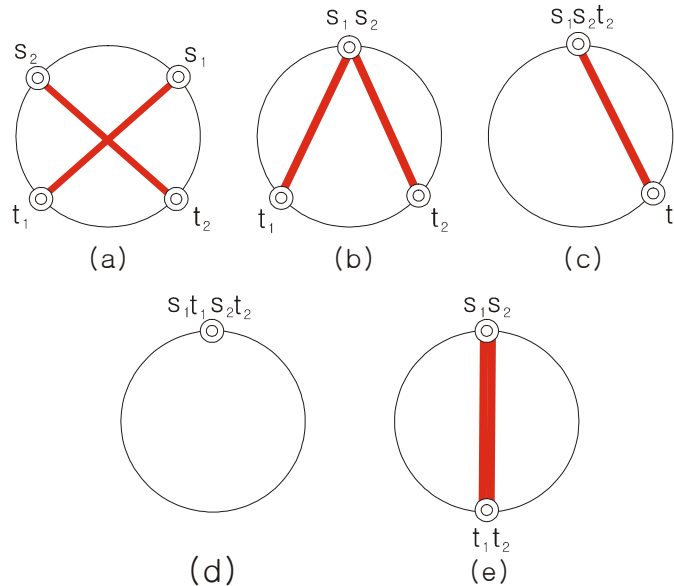


그림 3. 두 코드가 교차하는 유형

#### 2.3.1 어떤 두 코드 쌍이 내부에서 교차하는 경우 (경우 A)

교차하는 두 코드를 호에 대응시키면 대응되는 두 호는 항상 교차하게 된다. 두 호가 모두 지나는 점들에 대응되는 요소 그래프들로 유도된 부그래프를 *교차 영역*이라고 부르기로 하자. 내부에서 교차하는 두 코드 쌍에 대한 서로 다른 교차 영역은 4개 존재한다. 먼저 어떤 내부에서 교차하는 두 코드의 교차 영역에 포함되지 않는 소스-싱크 쌍이 존재하는 경우를 고려한다. 이 경우  $k \geq 3$ 이다.  $s_1, t_1$  쌍과  $s_2, t_2$  쌍이 만드는 어떤 교차 영역  $R$ 에  $s_3, t_3$ 가 모두 포함되지 않는다고 가정하자.

#### Procedure DPC-B

1. 교차 영역  $R$ 을 지나도록  $s_1, t_1$  쌍과  $s_2, t_2$  쌍에 대한 자유 브리지를 찾는다.  $s_3, t_3$  쌍에

- 대해서는 교차 영역  $R$ 을 지나지 않는 방향으로 자유 브리지를 찾는다.
2. 나머지 쌍들은 임의의 방향으로 자유 브리지를 둔다.
  3. 터미널을 가지거나 자유 브리지가 지나는 요소 그래프  $G_i$  각각에 대해서,  $G_i$ 에 속한 접점을 가상 고장으로 두고  $G_i$ 에 속한 모든 소스-싱크 쌍  $s_p-t_p$ , 자유 브리지 끝점과 터미널쌍  $s_q-y_q$ 와  $x_r-t_r$ , 그리고 자유 브리지에 속한 접점이 아닌 두 자유 정점쌍  $x_l-y_l$ 를 잇는 DPC를 찾는다.

위의 procedure를 기술할 때 Procedure DPC-A의 단계 2와 5를 생략하여 기술하고 있음에 주의한다.

**보조 정리 3.** 어떤 소스-싱크 쌍 모두를 포함하지 않는 교차 영역이 존재하고 경우 A를 만족할 때, Procedure DPC-B는 항상  $f$ -고장  $k$ -DPC를 설계한다.

**증명.**  $k \geq 3$ 이고 따라서  $\delta \geq 5$ 이므로 보조 정리 1에 의하여 자유 브리지가 존재한다. 단계 1과 같이 자유 브리지를 두면 세 자유 브리지가 동시에 지나는 요소 그래프는 존재하지 않게 된다. 따라서 어떤 요소 그래프에서도 최대  $k-1$ 개의 서로소인 경로를 찾으면 되는데, 그런 서로소인 경로가 존재함은 보조 정리 2와 같은 방법으로 증명된다.  $\square$

$s_1, t_1$ 쌍과  $s_2, t_2$ 쌍이 코드 내부에서 교차한다고 하자.  $s_1, t_1$ 은 각각  $G_0, G_b$ 에 속하고,  $s_2, t_2$ 는 각각  $G_a, G_c$ 에 속하며  $0 < a < b < c$ 라고 하자.  $G_{i,j}$ 는  $G_i, G_{i+1}, \dots, G_j$ 에 속한 정점들로 유도된 부그래프를 나타낸다. 먼저  $k \geq 3$ 인 경우를 다룬다. 교차 영역은  $G_{0,a}, G_{a,b}, G_{b,c}, G_{c,0}$ 가 될 수 있고, 이들 교차 영역은 모든  $j \geq 3$ 에 대하여  $s_j, t_j$  중 최소한 하나를 포함하여야 한다. 이 말은  $s_j, t_j$ 는  $G_0, G_b$ 에 하나씩 포함되거나 혹은  $G_a, G_c$ 에 하나씩 포함됨을 의미한다. 일반성을 잃지 않고  $s_j, t_j$ 는 각각  $G_0, G_b$ 에 포함되거나 혹은 각각  $G_a, G_c$ 에 포함된다고 가정한다. 따라서 모든 소스는  $G_0$ 나  $G_a$ 에 포함되며 모든 싱크는  $G_b$ 나  $G_c$ 에 포함된다. 그리고  $I_0 = I_b, I_a = I_c$ 이다. 여기서  $I_j$ 는  $G_j$ 에 속한 소스나 싱크의 인덱스 집합을 말한다.  $|I_0| \geq |I_a|$ 라고 가정한다.

### Procedure DPC-C

1.  $|I_a| \geq 2$ 인 경우:

$s_1, t_1$ 쌍에 대한 자유 브리지와  $s_p, t_p$ 쌍( $p \in I_0, p \neq 1$ )에 대한 자유브리지는 서로 방향이 반대가 되도록 둔다. 마찬가지로  $s_2, t_2$ 쌍에 대한 자유 브리지와  $s_q, t_q$ 쌍( $q \in I_a, q \neq 2$ )에 대한 자유브리지는 서로 방향이 반대가 되도록 둔다.

2.  $|I_a| = 1, f \geq 1$ 인 경우:

$s_1, t_1$ 쌍에 대한 자유 브리지와  $s_p, t_p$ 쌍( $p \in I_0, p \neq 1$ )에 대한 자유브리지는 서로 방향이 반대가 되도록 둔다.  $s_2, t_2$ 쌍에 대한 자유 브리지는  $f_0 \leq f_b$ 이면  $G_0$ 를 그렇지 않으면  $G_b$ 를 지나도록 둔다.

3.  $|I_a| = 1, f = 0$ 인 경우:

$s_p$ 를  $s_p^+ \neq s_2$ 인  $G_0$ 의 소스라고 하자.  $s_p^+ - t_p$  쌍에 대하여  $G_0$ 를 지나지 않는 자유 브리

지를 찾는다. 이 자유 브리지와  $(s_p, s_p^+)$ 를 합하여  $s_p - t_p$  쌍에 대한 자유 브리지라고 간주한다. 모든  $s_q, t_q$  쌍( $q \in I_0, q \neq p$ )에 대한 자유 브리지는  $G_c$ 를 지나도록 한다.  $s_2, t_2$  쌍에 대한 자유 브리지는  $G_0$ 를 지나도록 둔다.

4. 터미널을 가지거나 자유 브리지가 지나는 요소 그래프  $G_i$  각각에 대해서,  $G_i$ 에 속한 접점을 가상 고장으로 두고  $G_i$ 에 속한 모든 소스-싱크 쌍  $s_p - t_p$ , 자유 브리지 끝점과 터미널쌍  $s_q - y_q$ 와  $x_r - t_r$ , 그리고 자유 브리지에 속한 접점이 아닌 두 자유 정점쌍  $x_l - y_l$ 를 잇는 DPC를 찾는다.

**보조 정리 4.** 내부에서 교차하는 두 코드쌍이 이루는 교차 영역에 포함되지 않는 소스-싱크쌍이 존재하지 않고  $k \geq 3$ 일 때, Procedure DPC-C는 항상  $f$ -고장  $k$ -DPC를 설계한다.

**증명.** 자유 브리지가 존재함은 보조 정리 1에 의하여 알 수 있다.  $|I_a| \geq 2$ 인 경우는 어떤 요소 그래프에서도 접점의 수  $f'$ 과 찾아야 하는 서로소인 경로의 수  $k'$ 의 합이  $k-1$  이하가 되어 DPC를 설계할 수 있다.  $|I_a|=1, f \geq 1$ 인 경우에  $s_2, t_2$  쌍에 대한 자유 브리지가  $G_0$ 를 지난다고 하자.  $f_0 < f$ 이고 또한  $s_2, t_2$  쌍에 대한 자유 브리지는  $G_0$ 에 접점을 두므로  $f_0 + f' + 2k' \leq \delta - 1$ 이 성립되어 DPC를 설계할 수 있다. 다른 요소 그래프를 지나는 자유 브리지는  $k-1$ 개 이하이다.  $|I_a|=1, f=0$ 인 경우는  $|I_a|=1, f \geq 1$ 인 경우와 유사하게 증명된다.  $s_2, t_2$  쌍에 대한 자유 브리지는  $G_0$ 에 접점을 두며,  $s_p - t_p$  쌍에 대해서는 경로를 찾지 않고  $s_p$ 를 가상 고장으로 두고 DPC를 찾을 수 있으므로 단계 4가 성립한다. 다른 요소 그래프를 지나는 자유 브리지는  $k-1$ 개 이하이다. □

이제  $k=2$ 인 경우를 다룬다.  $f \geq 1$ 이면 두 소스-싱크 쌍에 대한 자유 브리지를 둘 때 자유 브리지가 지나는 교차 영역에 속하는 모든 요소 그래프  $G_i$ 에 대해서  $f_i < f$ 가 성립하도록 한다.  $G_i$ 에서 DPC가 존재하는데 그 이유는  $f_i < f$ 이고 두 자유 브리지 중에서 하나는  $G_i$ 에 접점을 만들기 때문이다. 이제  $f=0$ 이고, 이 경우는 유일하게 요소 그래프가  $\delta-2$ -고장 해밀톤 그래프라는 조건을 이용하여 설계한다. 아래 procedure의 단계 1, 2에서는  $w^+ \neq t_1$ 과  $z^- \neq t_2$ 를 만족하는  $G_a$ 의 정점  $w$ 와  $G_0$ 의 정점  $z$ 에 대하여  $G_{0,a}$ 를 커버하는  $s_1 - w$  경로와  $s_2 - z$  경로를 설계한다. 이 procedure가 2-DPC를 설계함은 자명하므로 그 증명은 생략한다.

#### Procedure DPC-D

1. 각 요소 그래프의 정점을 하나씩 지나고 또한 모든 터미널을 포함하는 길이  $d$ 인 사이클이 존재하는 경우:

$G_0$ 에서  $G_a$ 까지의 트렉 중  $s_1$ 을 지나는 것이  $s_2$ 도 지난다.  $G_0$ 에서  $G_a$ 까지의 임의의 자유 트렉 하나를  $(u_0, u_1, \dots, u_a)$ 라고 하자. 모든  $0 \leq i < a$ 에 대하여  $G_i$ 에서  $u_i$ 를 가상 고장으로 두고 해밀톤 사이클을 찾는다.  $v_0 = s_1$ 이라고 두고,  $G_i$ 의 해밀톤 사이클에서  $v_i$ 에 인접한 두 정점중 하나에 인접한  $G_{i+1}$ 의 정점을  $v_{i+1}$ 이라고 둔다. 이때  $v_{i+1} \neq s_2$ 이어야 한다.  $G_{0,a-1}$ 에는 자유 트렉  $(u_0, u_1, \dots, u_a)$ 에 포함된 정점을 제외한 모든 정점을 지나는

$s_1-v_a^-$  경로를 찾았다.  $G_a$ 에서  $v_a$ 와  $u_a$ 를 잇는 해밀톤 경로를 찾는다. 이 경로를  $(v_a, P_1, w, s_2, P_2, u_a)$ 라고 하자.  $s_1-v_a^-$  경로와  $(v_a, P_1, w)$ 를 병합하여  $s_1-w$  경로를 설계하고,  $(s_2, P_2, u_a)$  경로와 자유 트렉  $(u_a, u_{a-1}, \dots, u_0)$ 을 병합하여  $s_2-z$  경로를 설계한다. 여기서  $z = u_0$ 이다.

2. 위 단계 1의 조건을 만족하는 사이클이 존재하지 않을 경우:

일반성을 잃지 않고  $G_0$ 에서  $G_a$ 까지의 트렉 중  $s_1$ 을 지나는 것이  $s_2$ 를 지나지 않는다고 가정한다.  $s_2$ 를 지나는 트렉을  $(u_0, u_1, \dots, u_a)$ ,  $u_a = s_2$ 라고 하자. 모든  $0 < j \leq a$ 에 대하여  $u_j$ 를 가상 고장으로 두고 해밀톤 사이클을 찾는다.  $v_0 = s_1$ 이라고 두고  $G_i$ 의 해밀톤 사이클 상에서  $v_{i-1}^+$ 에 인접한 두 정점 중  $t_1$ 에 인접하지 않는 것을  $v_i$ 라고 하자.  $G_{1,a}$ 에서  $(u_0, u_1, \dots, u_a)$ 에 속한 정점을 제외한 다른 모든 정점을 지나는  $s_1^+ - v_a$  경로를 찾았다.  $v_a = w$ 라고 두고 이 경로를  $s_1$ 에서 시작하면  $s_1-w$  경로가 된다. 자유 트렉  $(u_a, u_{a-1}, \dots, u_0)$ 과  $G_0$ 에서  $s_1$ 을 가상 고장으로 두고 찾은 해밀톤 사이클을 이용하여  $s_2-z$  경로를 설계할 수 있다.

3.  $G_{a+1,b}$ 에서  $w^+$ 와  $t_1$ 을 잇는 해밀톤 경로를 찾아  $s_1-w$  경로와 합병하고,  $G_{c,d-1}$ 에서  $z^-$ 와  $t_2$ 를 잇는 해밀톤 경로를 찾아  $s_2-z$  경로와 합병한다.  $s_1-t_1$  경로를  $G_{b+1,c-1}$ 의 모든 정점을 지나도록 확장한다.

### 2.3.2 내부에서만 교차하는 코드 쌍이 없고, 끝점에서만 교차하는 쌍이 존재하는 경우 (경우 B)

끝점에서만 교차하는 어떤 두 코드 쌍의 교차 영역을  $G_0$ 라고 가정하자. 만약 두 끝점 모두가  $G_0$ 가 아닌 코드가 존재한다면, Procedure DPC-B와 같은 방식으로 세 코드에 대한 자유 브리지를 찾는데, 어떤 요소 그래프도 세 자유 브리지가 모두 지나지는 않도록 둘 수 있다. 따라서  $f$ -고장  $k$ -DPC를 설계할 수 있다.

이제 모든 코드의 최소한 한 끝점은  $G_0$ 라고 가정한다. 모든 소스는  $G_0$ 에 포함된다고 가정한다. 이 경우  $|L_0| = k$ 이고 모든  $j (1 \leq j < d)$ 에 대하여  $|L_j| < k$ 임을 관찰한다.  $G_0$  외부에 있는 싱크의 수를  $k'$ 이라고 하고,  $G_0$ 에 있는 싱크의 수를  $k''$ 이라고 하자.  $k = k' + k''$ 이다.

먼저  $k' = k$  ( $k'' = 0$ )인 경우를 고려한다.  $G_{1,j-1}$ 과  $G_{j,d-1}$ 이 모두 하나 이상의 싱크를 포함하는  $j (2 \leq j < d)$ 가 존재한다. 일반성을 잃지 않고  $t_1$ 은  $G_{1,j-1}$ 에  $t_2$ 는  $G_{j,d-1}$ 에 포함된다고 가정한다.

#### Procedure DPC-E

1.  $p \geq 3$ 이고  $t_p$ 가  $G_{1,j-1}$ 에 속하는 각 소스  $s_p$ 에 대하여  $s_p^+$ 와  $(s_p, s_p^+)$ 가 모두 고장이 아니고 터미널도 아니면  $(x_p, y_p) = (s_p, s_p^+)$ 라고 두고, 그렇지 않으면  $(x_p, y_p)$ 는 자유 에지로서  $x_p$ 는  $s_p$ 에 고장이 아닌 에지로 인접한  $G_0$ 의 정점이고  $y_p$ 는  $G_1$ 에 속하도록 둔다. 마

찬가지로  $p \geq 3$ 이고  $t_p$ 가  $G_{j,d-1}$ 에 속하는 각 소스  $s_p$ 에 대하여  $s_p^-$ 와  $(s_p, s_p^-)$ 가 모두 고장이 아니고 터미널도 아니면  $(x_p, y_p) = (s_p, s_p^-)$ 라고 두고, 그렇지 않으면  $(x_p, y_p)$ 는 자유 에지로서  $x_p$ 는  $s_p$ 에 고장이 아닌 에지로 인접한  $G_0$ 의 정점이고  $y_p$ 는  $G_{d-1}$ 에 속하도록 둔다. 이때 에지  $(x_p, y_p)$ 들을 서로 같거나 인접하지 않도록 한다.

2.  $p \geq 3$ 인 모든  $p \in I_0$ 에 대하여  $x_p, s_p$ 를 가상 고장으로 두고,  $G_0$ 에서  $s_1$ 과  $s_2$ 를 잇는 해밀톤 경로를 찾는다. 이  $s_1-s_2$  해밀톤 경로가 지나는 에지 중에서 다음 조건을 만족하는 에지  $(x_1, x_2)$ 를 찾는다: (a)  $x_1^+, (x_1, x_1^+)$ 은 터미널이 아니고 고장도 아니며, (b)  $x_2^-, (x_2, x_2^-)$ 도 터미널이 아니고 고장이 아니다.  $y_1 = x_1^+, y_2 = x_2^-$ 라고 둔다.
3.  $G_{1,j-1}$ 에 속한 모든  $y_p$ 와 싱크  $t_p$ 를 잇는 DPC를 찾고,  $G_{j,d-1}$ 에 속한 모든  $y_p$ 와 싱크  $t_p$ 를 잇는 DPC를 찾는다. 이 둘을  $G_0$ 의  $s_1-s_2$  해밀톤 경로, 에지  $(s_p, x_p), (x_p, y_p)$ 들과 합병한다.

**보조 정리 5.**  $k' = k$ 이고 경우 B를 만족할 때, Procedure DPC-E는 항상  $f$ -고장  $k$ -DPC를 설계한다.

**증명.** 먼저 단계 1의 조건을 만족하는 에지  $(x_p, y_p)$ 들이 존재함을 보인다. 한 소스  $s_p$ 에 대하여  $(x_p, y_p)$ 의 후보는  $\delta+1$ 개이고, 후보의 방해 요소는 고장 요소 최대  $f$ 개,  $s_p$ 를 제외한 터미널  $2k-1$ 개가 있다.  $x_q \neq s_q$ 이면  $x_q$ 도 후보를 방해할 수 있지만, 이 경우는  $t_q$ 가  $G_{1,j-1}$ 에 속하면  $s_q^+$ 나  $(s_q, s_q^+)$ 가 고장이거나 혹은  $s_q^+$ 가 터미널이고,  $t_p$ 가  $G_{j,d-1}$ 에 속하면  $s_q^-$ 나  $(s_q, s_q^-)$ 가 고장이거나 혹은  $s_q^-$ 가 터미널이다. 따라서 방해 요소의 수는 최대  $f+2k-1$ 이고 후보의 수보다 작으므로  $(x_p, y_p)$ 들이 존재한다.  $G_0$ 에  $s_1-s_2$  해밀톤 경로가 존재함은 자명하고, 정점의 수가 충분히 많으므로 단계 2의 조건을 만족하는 에지  $(x_1, x_2)$ 가 존재함을 쉽게 보일 수 있다. 각 요소 그래프가  $f$ -고장  $k-1$ -DPC를 가지므로  $G_{1,j-1}$ 와  $G_{j,d-1}$ 도  $f$ -고장  $k-1$ -DPC를 가진다. 따라서 단계 3의 DPC도 존재한다.  $\square$

이제  $k' < k$  ( $k'' \geq 1$ )인 경우를 고려한다.  $G_0$ 에 속한 싱크들의 인덱스 집합을  $I_0''$ ,  $I_0' = I_0 - I_0''$ 이라고 둔다. 당연히  $k' = |I_0'|$ ,  $k'' = |I_0''|$ 이다. 이 경우에  $G_{1,d-1}$ 이  $f$ -고장  $k-1$ -DPC를 가진다는 사실을 DPC 설계에 이용한다.

#### Procedure DPC-F

1. 각각의  $p \in I_0'$ 에 대하여  $s_p^+$ 가 자유 정점이고  $(s_p, s_p^+)$ 가 고장이 아니면  $(x_p, y_p) = (s_p, s_p^+)$ 라고 둔다; 그렇지 않고  $s_p^-$ 가 자유 정점이고  $(s_p, s_p^-)$ 가 고장이 아니면  $(x_p, y_p) = (s_p, s_p^-)$ 라고 둔다; 그렇지 않으면  $x_p$ 가  $G_0$ 에 속하고  $y_p$ 는  $G_{1,d-1}$ 에 속하는 임의의 자유 에지를 선택하여  $(x_p, y_p)$ 라고 둔다.
2. 각각의  $p \in I_0'$ 에 대하여  $x_p = s_p$ 이면  $s_p$ 를 모두 가상 고장으로 두고,  $G_0$ 에서  $p \in I_0'$ 이면서  $x_p \neq s_p$ 인  $s_p-x_p$  쌍과  $q \in I_0''$ 인  $s_q-t_q$  쌍을 잇는 DPC를 찾는다.
3.  $G_{1,d-1}$ 에서 모든  $p \in I_0'$ 에 대하여  $y_p-t_p$  쌍을 잇는 DPC를 찾고, 단계 2의 DPC와 병합

한다.

**보조 정리 6.**  $f_0 \leq f-2$ 이거나,  $k' \geq 3$ , 또는  $k'=2$ 이고  $f_0 = f-1$ 이면서  $k' < k$ 이고 경우 B를 만족할 때, Procedure DPC-F는 항상  $f$ -고장  $k$ -DPC를 설계한다.

**증명.**  $p \in I_0'$ 에 대하여  $x_p = s_p$ 인 소스의 수를  $f'$ 이라고 하자.  $G_0$ 에서  $f_0 + f'$ -고장  $k'' + (k' - f')$ -DPC가 존재함을 보이면 충분하다. 즉,  $f_0 + f' + 2(k'' + (k' - f')) \leq \delta - 1$ 임을 보인다.  $f_0 \leq f-2$ 이면 당연히 성립한다.  $f_0 \geq f-1$ 이라고 하자.  $k' \geq 3$ 이면  $f' \geq 2$ 이거나 혹은  $f_0 = f-1$ 이고  $f' \geq 1$ 이어서 성립한다. 마지막으로  $k'=2$ 이고  $f_0 = f-1$ 이면  $f' \geq 1$ 이므로 성립한다.  $\square$

위의 procedure를 직접 적용하지 못한 경우는  $k'=2$ 이고  $f_0 = f$ 이거나 혹은  $k'=0,1$ 이고  $f_0 \geq f-1$ 인 경우이다. 먼저  $k'=2$ 이고  $f_0 = f$ 인 경우는  $I_0' = \{1,2\}$ 라고 가정한다.  $x_1 = s_1$ 이고  $x_2 = s_2$ 이라면 Procedure DPC-F를 직접 적용할 수 있다. 그렇지 않으면, 일반성을 잃지 않고  $s_2$ 가  $t_1, t_2$ 와 모두 인접하다고 가정한다.  $(x_2, y_2) = (s_2, t_2)$ 라고 두고 단계 2를 적용한 다음,  $G_{1,d-1}$ 에서  $t_2$ 를 가상 고장으로 두고  $y_1 - t_1$  쌍을 잇는 1-DPC를 찾으면 된다.

$k'=1$ 이고  $f_0 = f-1$ 인 경우를 고려한다.  $I_0' = \{1\}$ 이라고 가정하자.  $x_1 = s_1$ 이면 Procedure DPC-F를 직접 적용한다. 그렇지 않으면  $(x_1, y_1) = (s_1, t_1)$ 이라고 두고 단계 2를 적용하여 DPC를 찾는다.  $s_1 - t_1$  경로는  $(s_1, t_1)$ 이라고 둔다.  $G_1$ 과  $G_{d-1}$ 에 고장과 가상 고장  $t_1$ 을 하나씩 나누어가지고 있고, 또한 요소 그래프  $G_i$ 는,  $1 \leq i < d$ ,  $f$ -고장 해밀톤 연결된 그래프이며  $f \geq 1$ 이기 때문에 Procedure DPC-A의 단계 5와 같은 방식으로 나머지 정점들을 지나도록 DPC를 확장할 수 있다.

$k'=1$ 이고  $f_0 = f$ 인 경우는 아래 Procedure DPC-G에서  $f$ -고장  $k$ -DPC를 설계한다.  $I_0' = \{1\}$ 이라고 가정한다. 단계 2의  $G_0$ 에서  $s_1 - t_2$  해밀톤 경로의 존재나 단계 4의  $G_{1,d-1}$ 에서 0-고장  $k-1$ -DPC의 존재는 자명하다.

### Procedure DPC-G

1.  $j \geq 3$ 인 모든 터미널  $s_j$ 와  $t_j$ 에 대하여  $s_j'$ 과  $t_j'$ 을  $t_1$ 과 다르고 각각  $s_j$ 와  $t_j$ 에 인접한  $G_{1,d-1}$ 의 정점이라고 하자.
2.  $j \geq 3$ 인 모든 터미널  $s_j$ 와  $t_j$ 들을 가상 고장으로 두고,  $G_0$ 에서  $s_1$ 과  $t_2$ 를 잇는 해밀톤 경로를 찾는다. 그 해밀톤 경로는  $s_2$ 를 지나고  $(s_1, P_1, x, s_2, P_2, t_2)$ 라고 둔다.
3.  $t_1$ 과 다르고  $x$ 에 인접한  $G_{1,d-1}$ 의 정점을  $x'$ 이라고 하자.
4.  $G_{1,d-1}$ 에서  $x' - t_1$ 쌍과  $j \geq 3$ 인 모든  $s_j' - t_j'$ 쌍을 잇는 DPC를 찾고,  $G_0$ 의 해밀톤 경로와 병합한다.

마지막으로  $k' = 0$ ,  $f_0 = f, f-1$ 인 경우를 고려한다. 이 경우는 위의 Procedure DPC-G와 매우 유사하게 DPC를 설계할 수 있다.

#### Procedure DPC-H

1.  $j \geq 3$ 인 모든 터미널  $s_j$ 와  $t_j$ 에 대하여  $s_j'$ 과  $t_j'$ 을 각각  $s_j$ 와  $t_j$ 에 인접한  $G_{1,d-1}$ 의 정점이라고 하자. 이때  $s_j'$ ,  $(s_j, s_j')$ ,  $t_j'$ , 그리고  $(t_j, t_j')$ 은 모두 고장 요소가 아니도록 선택한다.
2.  $j \geq 3$ 인 모든 터미널  $s_j$ 와  $t_j$ 들을 가상 고장으로 두고,  $G_0$ 에서  $s_1$ 과  $t_1$ 을 잇는 해밀톤 경로를 찾는다. 그 해밀톤 경로는  $s_2, t_2$ 를 지나므로  $(s_1, P_1, x, s_2, P_2, t_2, y, P_3, t_1)$ 이라고 둔다.
3.  $x'$ 과  $y'$ 을 각각  $x$ 와  $y$ 에 인접한  $G_{1,d-1}$ 의 정점으로서  $x'$ ,  $(x, x')$ ,  $y'$ , 그리고  $(y, y')$ 이 모두 고장이 아닌 것으로 선택한다.
4.  $G_{1,d-1}$ 에서  $x'-y'$ 쌍과  $j \geq 3$ 인 모든  $s_j'-t_j'$ 쌍을 잇는 DPC를 찾고,  $G_0$ 의 해밀톤 경로와 병합한다.

#### 2.3.3 모든 코드가 끝점과 내부에서 동시에 교차하는 경우 (경우 C)

이 경우에는 모든 소스-싱크 쌍에 대한 코드가 서로 같고 그 길이가 0이 아니다. 일반성을 잃지 않고 모든 소스는  $G_0$ 에 모든 싱크는  $G_i$ 에 존재하고  $d \geq 3$ 이므로  $i \neq d-1$ 이라고 가정한다.

#### Procedure DPC-I

1.  $j \geq 3$ 인 각 소스  $s_j$ 에 대하여  $s_j^-$ 와  $(s_j, s_j^-)$ 가 모두 고장이 아니면  $(x_j, y_j) = (s_j, s_j^-)$ 라고 두고, 그렇지 않으면  $(x_j, y_j)$ 는 자유 에지로서  $x_j$ 는  $s_j$ 에 고장이 아닌 에지로 인접한  $G_0$ 의 정점이고  $y_j$ 는  $G_{d-1}$ 에 속하도록 둔다.  $Y = \{y_j | j \geq 3\}$ 라고 하자.
2.  $j \geq 3$ 인 각 싱크  $t_j$ 에 대하여  $t_j^+$ 와  $(t_j, t_j^+)$ 가 모두 고장이 아니고  $t_j^+ \notin Y$ 이면  $(a_j, b_j) = (t_j, t_j^+)$ 이라고 두고, 그렇지 않으면  $(a_j, b_j)$ 는 자유 에지로서  $a_j$ 는  $t_j$ 에 고장이 아닌 에지로 인접한  $G_i$ 의 정점이고  $b_j$ 는  $G_{i+1}$ 의 정점으로  $Y$ 에 속하지 않도록 둔다.  $B = \{b_j | j \geq 3\}$ 라고 하자.
3.  $G_0$ 에서  $j \geq 3$ 인 모든  $s_j$ 와  $x_j$ 를 가상 고장으로 두고  $s_1$ 과  $s_2$ 를 잇는 해밀톤 경로를 찾는다. 이 경로에 속한 에지  $(x_1, x_2)$ 를 선택하는데  $x_1^-$ ,  $(x_1, x_1^-)$ ,  $x_2^+$ ,  $(x_2, x_2^+)$ 가 모두 고장이 아니고,  $x_1^-$ 는  $B$ 에 속하지 않도록 하며,  $t_2$ 에 인접한  $G_i$ 의 모든 정점  $v$ 에 대하여  $x_1^- \neq v^+$ 가 되도록 한다. 또한  $x_2^+ \neq t_1$ 이고, 모든  $j \geq 3$ 에 대하여  $x_2^+ \neq t_j$ ,  $a_j$ 를 만족하도록 한다. 찾은 해밀톤 경로를  $(s_1, P_1, x_1, x_2, P_2, s_2)$ 라고 하고,  $y_1 = x_1^-$ ,  $y_2 = x_2^+$ 라고 둔다.
4.  $i = 1$ 이면  $w = y_2$ 라고 두고, 그렇지 않으면  $w$ 는  $G_i$ 에 속한 자유 정점으로  $G_{1,i-1}$ 에서  $y_2$ 와  $w^-$ 를 잇는 해밀톤 경로가 존재하고  $(w, w^-)$ 가 고장이 아닌 에지가 되도록 둔다.
5.  $F'$ 은  $t_2$ 에 인접한  $G_i$ 의 에지  $(t_2, v)$ 들의 집합으로서  $v$ 는 싱크가 아니며 동시에  $v^+$ 나



$(v, v^+)$ 가 고장이거나 혹은  $v^+ \in Y$ 라는 조건을 만족한다.  $G_i$ 에서  $F$ 과  $j \geq 3$ 인 모든  $t_j$ 와  $a_j$ 를 가상 고장으로 두고  $w$ 과  $t_1$ 을 잇는 해밀톤 경로를 찾는다. 이것을  $(w, Q_1, t_2, z, Q_2, t_1)$ 라고 둔다.  $(a_1, b_1) = (z, z^+)$ 라고 둔다.

6.  $G_{i+1, d-1}$ 에서 모든  $j \neq 2$ 에 대하여  $y_j - b_j$ 를 잇는  $k-1$ -DPC를 찾는다. 이 DPC와  $G_0$ 와  $G_i$ 의 해밀톤 경로, 그리고 에지들  $(x_j, y_j)$ ,  $(a_j, b_j)$ 와 병합한다.

**보조 정리 7.** 경우 C를 만족할 때, Procedure DPC-I는 항상  $f$ -고장  $k$ -DPC를 설계한다.

**증명.** 단계 1의 조건을 만족하는  $(x_j, y_j)$ 들의 존재는 쉽게 알 수 있다. 단계 2의  $(a_j, b_j)$ 들이 존재함을 보이려고 한다. 한 싱크  $t_j$ 에 대하여  $(a_j, b_j)$ 의 후보는  $\delta+1$ 개이고, 후보의 방해 요소는 고장 요소 최대  $f$ 개,  $Y$ 의 원소들  $k-2$ 개, 그리고  $t_j$ 를 제외한 싱크들이다. 어떤 싱크  $t_p$ 가  $a_p \neq t_p$ 이면 후보 둘을 방해할 수도 있지만 이 경우는  $a_p^+$ 나  $(a_p, a_p^+)$ 가 고장이거나 혹은  $a_p^+ \in Y$ 가 된다. 따라서 방해 요소의 수는 최대  $f + (k-2) + (k-1)$ 이고 후보의 수보다 작으므로  $(a_j, b_j)$ 들이 존재한다.

단계 3의 해밀톤 경로 존재는 자명하다. 조건을 만족하는 에지  $(x_1, x_2)$ 의 존재를 보인다. 해밀톤 경로의 길이는  $n - f - 2(k-2) - 1 = n - \delta + 2$  이상이고 경로상의 에지는 모두  $(x_1, x_2)$ 의 후보이다. 후보에 대한 방해 요소는 고장 요소 최대  $f$ 개,  $B$ 의 원소  $k-2$ 개, 그리고  $t_2$ 에 인접한  $G_i$ 의 정점  $v$ 에 대한  $v^+$ 들  $\delta$ 개, 그리고 ( $i=1$ 인 최악의 경우에)  $t_1$ 과  $j \geq 3$ 인  $t_j$ ,  $a_j$ 들  $2k-3$ 개이다. 총 방해 요소의 수는  $f + (k-2) + \delta + (2k-3) = (f+2k) + k + \delta - 5 \leq (\delta+1) + \lfloor (\delta+1)/2 \rfloor + \delta - 5 = 2\delta + \lfloor (\delta+1)/2 \rfloor - 4$ 이다. 후보 에지의 수가 이것보다 크다고 보이면, 즉 모든  $\delta \geq 3$ 에 대하여  $n > 3\delta + \lfloor (\delta+1)/2 \rfloor - 6$ 임을 보이는데, 이는 수학적 귀납법으로 쉽게 증명되어 생각한다.

정점  $w$ 의 존재는 자명하다. 단계 5의 해밀톤 경로가 존재함을 알기 위해서 고장과 가상 고장의 수가  $\delta-3$  이하임을 보인다. 고장의 수는  $f_i$ 이다.  $F$ 에 속한 가상 고장 에지의 수는  $G_i$  외부의 최대  $f - f_i$ 개 고장과  $Y$ 의 원소  $k-2$ 개이다.  $j \geq 3$ 인  $a_j$ 와  $t_j$ 들이 가상 고장이 되는데  $a_j \neq t_j$ 라면  $t_j^+$ 나  $(t_j, t_j^+)$ 가 고장이거나 혹은  $t_j^+ \in Y$ 이므로, 총 가상 고장의 수는  $(f - f_i) + (k-2) + (k-2)$ 이다. 이를 고장의 수  $f_i$ 와 합하면 모두  $f + 2k - 4 = \delta - 3$ 개이다. 설계에 의하여  $z^+$ 와  $(z, z^+)$ 는 고장이 아니며  $z^+ \notin YU\{y_1\}$ 이다. 각 요소 그래프가  $f$ -고장  $k-1$ -DPC를 가지므로 단계 6의 DPC는 존재한다.  $\square$

### 3. 재귀원형군과 토러스의 서로소인 경로 커버

이 절에서는 정리 1의  $\bigoplus_{0 \leq i < d} G_i$ 에서 쌍형 다대다 서로소 경로 커버의 설계를 이용하여 재귀원형군과 토러스에서 쌍형 다대다 서로소인 경로 커버를 설계할 수 있음을 보인다. 재귀원형군  $G(cd^m, d)$ ,  $d \geq 3$ 과 토러스  $C_{k_1} \times C_{k_2} \times \dots \times C_{k_m}$ 은 각각 하나 낮은 차원의 재귀원형

군  $G(cd^{m-1}, d)$ 와 토러스  $C_{k_1} \times C_{k_2} \times \dots \times C_{k_{m-1}}$ 에 사이클 기반 설계를 적용하여 얻을 수 있다.

먼저 정리 1의 요소 그래프의 정점 수에 대한 조건을 만족하는지 살펴본다.

**보조 정리 8.** 분지수가  $\delta$ 인 재귀원형군  $G(cd^m, d)$ ,  $d \geq 3$ 과  $m$ -차원 토러스  $C_{k_1} \times C_{k_2} \times \dots \times C_{k_m}$ 의 정점 수  $n$ 은  $\delta$ 가 홀수이면  $n \geq 2 \cdot 3^{(\delta-1)/2}$ 이고 짝수이면  $n \geq 3^{\delta/2}$ 이다.

**증명.** 먼저 재귀원형군  $G(cd^m, d)$ 이 정점 수에 대한 조건을 만족함을  $m$ 에 대한 수학적 귀납법으로 증명한다.  $G(cd^0, d)$ 의 정점 수는  $c$ 이다. 그리고  $c=1$ 이면 분지수 0,  $c=2$ 이면 분지수 1,  $c \geq 3$ 이면 분지수가 2가 되므로 성립한다.  $G(cd^m, d)$ 는  $G(cd^{m-1}, d)$ 에 비하여 정점 수가  $d(d \geq 3)$ 배이고 분지수가 둘 늘어나므로 보조 정리가 증명된다.  $m$ -차원 토러스  $C_{k_1} \times C_{k_2} \times \dots \times C_{k_m}$ 에서 모든  $k_i (1 \leq i \leq m)$ 는 3 이상인 정수이고 정점 수가  $k_1 k_2 \dots k_m$ 이며 분지수는  $2m$ 이므로 정점 수의 조건을 만족함이 명백하다.  $\square$

재귀원형군과 토러스의 고장 해밀톤 성질은 아래와 같이 연구되어 있다. 이 고장 해밀톤 성질은 분지수  $\delta=3, 4$ 인 경우에  $f+2k \leq \delta-1$ 을 만족하는 모든  $f, k \geq 1$ 에 대하여  $f$ -고장 쌍형  $k$ -DPC가 존재함을 의미한다. 1-DPC는 바로 해밀톤 경로와 같은 의미이다. 또한 정리 1의 요소 그래프  $G_i$ 가 만족해야 할 조건인  $\delta-2$ -고장 해밀톤 그래프임도 보여준다.

**보조 정리 9.** [13] 재귀원형군  $G(cd^m, d)$ 의 분지수를  $\delta$ 라고 하면, 이분 그래프가 아닌 재귀원형군은  $\delta-3$ -고장 해밀톤 연결되어 있고  $\delta-2$ -고장 해밀톤 그래프이다.

**보조 정리 10.** [14]  $m$ -차원 토러스의 분지수를  $\delta$ 라고 하면, 토러스가 이분 그래프가 아니면  $\delta-3$ -고장 해밀톤 연결되어 있고  $\delta-2$ -고장 해밀톤 그래프이다.

위 보조 정리 8, 9, 10과 정리 1을 이용하면 분지수  $\delta$ 에 대한 수학적 귀납법으로 다음 정리가 증명된다.

**정리 2.** 이분 그래프가 아닌 재귀원형군  $G(cd^m, d)$ ,  $d \geq 3$ 과 토러스는 분지수를  $\delta$ 라고 할 때,  $f+2k \leq \delta-1$ 을 만족하는 임의의  $f, k \geq 1$ 에 대하여  $f$ -고장 쌍형  $k$ -DPC를 가진다.

## 4. 결론

이 논문에서는 요소 그래프  $G_0, G_1, \dots, G_{d-1}$ 의 고장 쌍형 DPC 성질과 고장 해밀톤 성질을 이용하여  $\bigoplus_{0 \leq i < d} G_i$ 에서 고장 쌍형 DPC의 설계를 제시하였다. 이 설계를 재귀원형군  $G(cd^m, d)$ ,  $d \geq 3$ 과 토러스에 적용하여 분지수  $\delta$ 가 3 이상이고 이분 그래프가 아닌 경우,

$f+2k \leq \delta-1$ 인 모든  $f$ ,  $k \geq 1$ 에 대하여  $f$ -고장 쌍형  $k$ -DPC를 설계할 수 있음을 보였다. 그런데 그래프  $G$ 가 임의의 소스, 싱크 사이에  $f$ -고장 쌍형  $k$ -DPC를 가지기 위한 필요조건으로  $f+2k \leq \delta+1$ 임이 알려져 있다[3].  $f+2k \leq \delta$ 나  $f+2k \leq \delta+1$ 로 조건이 완화되더라도 이분 그래프가 아닌 재귀원형군과 토러스가 과연  $f$ -고장 쌍형  $k$ -DPC를 가지겠는가 하는 것은 미해결 연구 과제이다.

## 참고 문헌

- [1] J.A. Bondy and U.S.R Murty, *Graph Theory*, Springer Verlag, 2007.
- [2] D.B. West, *Introduction to Graph Theory (2nd edition)*, Prentice Hall, 2001.
- [3] J.-H. Park, H.-C. Kim, and H.-S. Lim, "Many-to-many disjoint path covers in hypercube-like interconnection networks with faulty elements," *IEEE Trans. Parallel and Distributed Systems* **17(3)**, pp. 227-240, Mar. 2006.
- [4] J.-H. Park, "One-to-one disjoint path covers in recursive circulants," *Journal of KISS* **30(12)**, pp. 691-698, 2003.
- [5] C.-H. Tsai, J.J.M. Tan, and L.-H. Hsu, "The super-connected property of recursive circulant graphs," *Inform. Proc. Lett.* **91(6)**, pp. 293-298, 2004.
- [6] C.-H. Chang, C.-K. Lin, H.-M. Huang, and L.-H. Hsu, "The super laceability of the hypercubes," *Inform. Proc. Lett.* **92(1)**, pp. 15-21, 2004.
- [7] J.-H. Park, "One-to-many disjoint path covers in a graph with faulty elements," in *Proc. of the International Computing and Combinatorics Conference COCOON 2004*, pp. 392-401, Aug. 2004.
- [8] J.-H. Park, "Unpaired many-to-many disjoint path covers in hypercube-like interconnection networks," *Journal of KISS* **33(10)**, pp. 789-796, 2006.
- [9] R. Caha and V. Koubek, "Spanning multi-paths in hypercubes," *Discrete Mathematics* **307(16)**, pp. 2053-2066, 2007.
- [10] P.-L. Lai and H.-C. Hsu, "The two-equal-disjoint path cover problem of matching composition network," *Inform. Proc. Lett.* **107(1)**, pp. 18-23, 2008.
- [11] J.-H. Park, H.-C. Kim, and H.-S. Lim, "Fault-hamiltonicity of hypercube-like interconnection networks," in *Proc. of the IEEE International Parallel & Distributed Processing Symposium IPDPS 2005*, Apr. 2005.
- [12] J.-H. Park and K.Y. Chwa, "Recursive circulants and their embeddings among hypercubes," *Theoretical Computer Science* **244(1-2)**, pp. 35-62, Aug. 2000.
- [13] C.-H. Tsai, J.J.M. Tan, Y.-C. Chuang, L.-H. Hsu, "Fault-free cycles and links in faulty recursive circulant graphs," in *Proc. of the Workshop on Algorithms and Theory of Computation ICS 2000*, pp. 74-77, 2000.
- [14] H.-C. Kim and J.-H. Park, "Fault hamiltonicity of two-dimensional torus networks," *Workshop on Algorithms and Computation WAAC'00*, Tokyo, Japan, pp. 110-117, July 2000.