

피라미드 그래프의 해밀톤 특성

장 정 환[†]

요 약

본 논문에서는 피라미드 그래프에서의 해밀톤 사이클 특성을 분석한다. 사이클 확장 연산을 이용하여 사이클의 크기를 확대시켜 나가는 일련의 과정을 통하여 해밀톤 사이클을 찾을 수 있는 제시된 알고리즘을 적용함으로써 임의의 높이 N 인 피라미드 그래프 내에 길이 $(4^N-1)/3$ 인 해밀톤 사이클이 존재함을 증명한다.

키워드 : 피라미드, 해밀톤 사이클, 경로

A Hamiltonian Property of Pyramid Graphs

Jung-Hwan Chang[†]

ABSTRACT

In this paper, we analyze the Hamiltonian property of Pyramid graphs. We prove that it is always possible to construct a Hamiltonian cycle of length $(4^N-1)/3$ by applying the proposed algorithm to construct series of cycle expansion operations into two adjacent cycles in the Pyramid graph of height N .

Key Words : Pyramid, Hamiltonian Cycle, Path

1. 개 요

대규모 병렬처리시스템에서 다양한 응용분야의 알고리즘들을 효율적으로 수용하여 제 성능을 발휘할 수 있도록 하기 위해서는 시스템을 구성하고 있는 노드들의 연결 구조를 그래프 형태로 표현한 상호연결망(interconnection network) 위상의 역할이 매우 중요한데, 그 이유는 해당 그래프에 대한 이론적인 특성에 해당 병렬처리시스템의 성능이 의존적일 수밖에 없기 때문이다. 일반적으로 상호연결망의 성능에 큰 영향을 미치는 그래프 이론적인 요소로는 지름(diameter), 연결도(connectivity), 고장 허용도(fault tolerance), 라우팅 알고리즘의 간결성 및 효율성 등이 중요하게 평가되고 있는데, 이러한 요소들을 비교 분석하여 보다 좋은 성능의 상호연결망을 설계하려는 시도는 그래프 이론 및 병렬처리 분야에서 많은 관심을 받고 있다. 뿐만 아니라 알고리즘에 내재된 자료구조를 실제 구현된 시스템 구조에 효율적으로 매핑시키거나, 상호연결망 상호간 시뮬레이션을 위한 체계적인 접근으로서 그래프 임베딩(embedding) 문제

가 대두하게 되었다[1, 2].

지난 십 수년간 큰 관심을 받아 온 대표적인 상호연결망으로는 하이퍼큐브(hypercube 또는 (이진) n -큐브)를 꼽을 수 있으며[3], 피라미드(pyramid)와 메쉬(mesh) 또한 병렬처리분야에서 비교적 잘 알려진 상호연결망 구조 중 하나로 널리 이용되고 있는 실정이다[4, 5].

높이 N 인 피라미드(pyramid)의 경우 $(4^N-1)/3$ 개의 정점들을 갖는 계층적 구조의 그래프로써 오늘날 각광을 받고 있는 멀티미디어 분야의 저 레벨에서의 픽셀 단위 영상처리 관련 기능을 효율적으로 처리해 줄 수 있는 구조적 특성으로 인해 관련 분야를 중심으로 널리 이용되어 왔으나 오늘날은 내재된 트리(tree) 및 메쉬(mesh) 구조를 다양하게 활용함으로써 그래프 관련 기본문제들의 해결에도 좋은 성과를 얻고 있는 것으로 알려져 있다[5].

손님(guest) 그래프 G 의 주인(host) 그래프 H 로의 임베딩(embedding)이란 G 의 정점들의 집합 $V(G)$ 로부터 H 의 정점(vertex)들의 집합 $V(H)$ 로의 일대일 대응을 의미하는 함수 F 로 정의된다. 이때 G 의 간선(edge) $e=(v,u)$ 는 $F(v)$ 와 $F(u)$ 를 연결하는 H 의 경로(path) $F(e)$ 에 대응된다.

높이가 N 인 피라미드를 대상으로 한 임베딩 관련 연구로는 주로 2-차원 또는 3-차원 메쉬(mesh)를 주인 그래프로

* 이 논문은 2005년도 부산외국어대학교 학술연구조성비에 의해 연구되었음.

† 정 회 원 : 부산외국어대학교 디지털미디어학부 조교수

논문접수 : 2005년 10월 17일, 심사완료 : 2006년 5월 4일

하는 연구결과들이 비교적 잘 알려져 있다[6-9].

한편 상호연결망 그래프에 내재된 사이클 특성을 분석한 연구들이 갖는 의미는 다양한 응용분야에 적용되고 있는 대부분 알고리즘들의 수행에 필요한 기본 자료구조일 뿐만 아니라 수많은 프로세서들로 구성된 다중컴퓨터 병렬처리시스템에서 사이클 기반의 방송 또는 멀티캐스팅 형태의 메시지 전달 과정을 필요로 하는 응용분야에서 하부구조로 사이클 특성을 활용하고 있는 점을 예로 들 수 있다[10].

본 논문에서는 피라미드 그래프를 대상으로 링을 임베딩 하는 문제와 동일한 의미를 갖는 해밀톤 사이클을 찾는 문제를 다룬다. 피라미드 그래프의 특성을 활용한 사이클 확장 연산을 이용함으로써 모든 정점들이 포함되는 최대 크기의 해밀톤 사이클을 찾을 수 있는 알고리즘을 제시함으로써 해밀톤 사이클의 존재를 확인하게 된다.

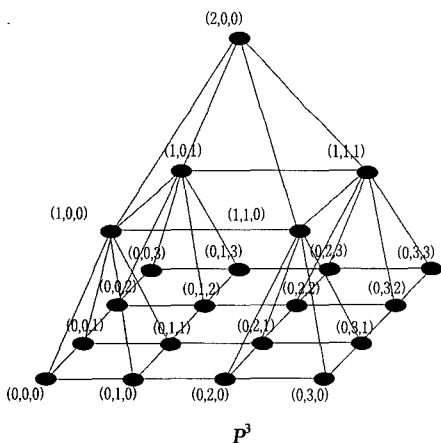
본 논문은 네 개의 절로 구성되며, 제2절에서는 본 논문에서 주로 다루게 될 피라미드 그래프에 대한 정의와 주요 특성들에 대해 살펴보고 제3절에서는 본 논문에서 제안하고 있는 사이클 확장 연산을 이용한 해밀톤 사이클 구성 알고리즘을 제시함으로써 해밀톤 사이클의 존재를 증명하게 되며, 마지막 제4절에서 결론을 맺는다.

2. 그래프 특성 분석

본 절에서는 연구 대상이 되는 피라미드 그래프에 대한 정의 및 그래프 이론적 관점에서의 주요 특성들을 살펴보고자 한다.

[정의 1] 주어진 양의 정수 N 에 대해 높이가 N 인 피라미드 P^N 은 정점(vertex)의 집합 $V(P^N)$ 과 간선(edge)의 집합 $E(P^N)$ 에 의해 다음과 같이 정의한다.

- 1) $V(P^N) = \bigcup_{0 \leq l \leq N-1} \{(l, x, y) | 0 \leq x, y \leq 2^{N-l-1} - 1\}$
- 2) $E(P^N) = \bigcup_{0 \leq l \leq N-2} \{((l, x_1, y_1), (l, x_2, y_2)) | |x_1 - x_2| + |y_1 - y_2| = 1 \ \&\& \ (l, x_1, y_1), (l, x_2, y_2) \in V(P^N)\}$
 $\cup \bigcup_{0 \leq l \leq N-2} \{(l, x, y), (l+1, \lfloor x/2 \rfloor, \lfloor y/2 \rfloor)\} | 0 \leq x, y \leq 2^{N-l-1} - 1\}$



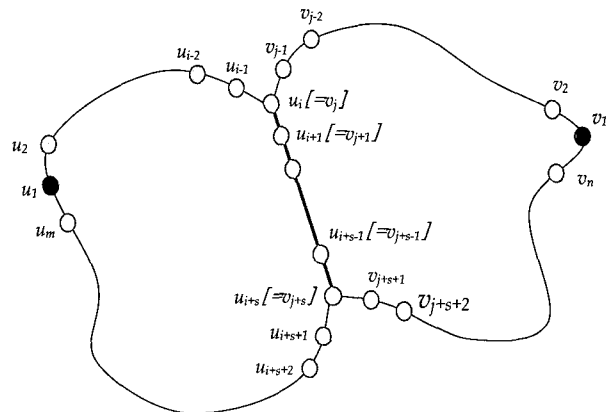
(그림 1) 높이 3인 피라미드 그래프 예

(그림 1)은 높이(height)가 3인 피라미드의 예를 보여주고 있다. 일반적으로 높이가 N 인 피라미드 P^N 의 각 계층 $l(0 \leq l < N)$ 에는 4^{N-l-1} 개의 정점들로 구성된 정방형 2-차원 메쉬 $2^{N-l-1} \times 2^{N-l-1}$ 구조로 이루어져 있어 가장 위 계층에 해당하는 계층 $N-1$ 의 1개의 정점으로 구성된 2-차원 메쉬로부터 시작하여 가장 아래 계층에 해당하는 계층 0 의 4^{N-1} 개로 구성된 2-차원 메쉬까지 다양한 그래프들을 포함하고 있음을 알 수 있다.

[정의 2] 그래프(graph)에서의 「경로(path)」란 시작 정점 v_1 으로부터 연속하여 인접된 간선을 따라 연속적으로 나열된 정점들의 정렬 $(v_1, v_2, v_3, v_4, \dots, v_n)$ 을 의미하며, 특별히 간선의 중복 방문이 허용되지 않도록 각 구성 정점들의 분지수(degree)가 최대 2로 제한된 경로를 「단순 경로(simple path)」라고 일컫는다. 특별히 시작 정점과 마지막 정점 간에 연결 간선을 갖는 닫힌 단순 경로(closed simple path)를 「사이클(cycle)」이라고 한다.

한편 주어진 그래프(또는 부그래프(sub-graph))를 구성하고 있는 모든 정점들을 반드시 한 번씩만 지나가게 되는 해당 그래프 내의 최대 길이 단순 경로를 「해밀톤 경로(Hamiltonian path)」라고 한다.

주어진 경로의 「길이(length)」란 해당 경로를 구성하고 있는 간선들의 개수를 의미한다. 즉 n 개의 정점 $(v_1, v_2, v_3, v_4, \dots, v_n)$ 로 구성된 경로 T 와 해당 경로의 길이를 각각 $T = (v_1, v_2, v_3, v_4, \dots, v_n)$ 및 $|T|$ 로 표기하며, 따라서 $|T| = n-1$ 이다. 반면에 n 개의 정점 $(v_1, v_2, v_3, v_4, \dots, v_n)$ 로 구성된 사이클 C 인 경우에 길이는 $|C|=n$ 이다. 일반적으로 길이가 n 인 경로 T 및 사이클 C 를 각각 T_n 과 C_n 으로 표기한다.



(그림 2) 사이클 결합 연산(\oplus)의 예제

[정의 3] 길이가 각각 m 과 n 인 주어진 두 개의 사이클 $C_m = (u_1, u_2, \dots, u_m)$ 과 $C_n = (v_1, v_2, \dots, v_n)$ 이 각각 $u_i(1 \leq i < m-s)$ 및 $v_j(1 \leq j < n-s)$ 로부터 시작되는 길이 $s(1 \leq s \leq \min(m, n))$ 의 부-경로(sub-path)를 서로 공유하여 다음 식이 성립한다고 가정하자.

$$u_{i+k} = v_{j+k} \quad (0 \leq k \leq s)$$

이와 같이 길이 s 에 해당하는 부-경로를 서로 공유하고 있는 두 개의 사이클 C_m 과 C_n 을 대상으로 하는 「사이클 결합 연산(cycle merge operation)」 \oplus 는 다음과 같이 정의된다.

$$C_m \oplus C_n = (u_1, u_2, \dots, u_{i-1}, u_i [=v_j], v_{j-1}, v_{j-2}, \dots, v_2, v_1, v_n, v_{n-1}, \dots, v_{j+s+2}, v_{j+s+1}, v_{j+s} [=u_{i+s}], u_{i+s+1}, u_{i+s+2}, \dots, u_{m-1}, u_m)$$

(그림 2)에서는 이러한 사이클 관련 결합 연산(\oplus)의 예를 보여주고 있다. 그림에서 제시된 예와 위의 결합 연산 정의에 의해 다음과 같은 기본 성질을 확인할 수 있다.

[보조정리 1] 길이가 각각 $m(\geq 2)$ 과 $n(\geq 2)$ 인 주어진 두 개의 사이클 C_m 과 C_n 내에 길이 $s(1 \leq s \leq \min(m, n))$ 에 해당하는 부-경로(sub-path)를 서로 공유하고 있다면, $C_m \oplus C_n$ 연산의 결과 얻어지는 확장된 사이클의 길이는 다음 식으로 표현된다.

$$|C_m \oplus C_n| = m+n-2s$$

[증명] 사이클 결합 연산의 정의를 참조하여 \oplus 연산 적용 후 변화가 일어나는 부분들을 중심으로 잔존 길이를 다음과 같이 계산 가능하다((그림 2) 참조).

- \oplus 연산 적용 후 C_m 부분 중 남아있게 되는 길이 = $m-s$
- \oplus 연산 적용 후 C_n 부분 중 남아있게 되는 길이 = $n-s$

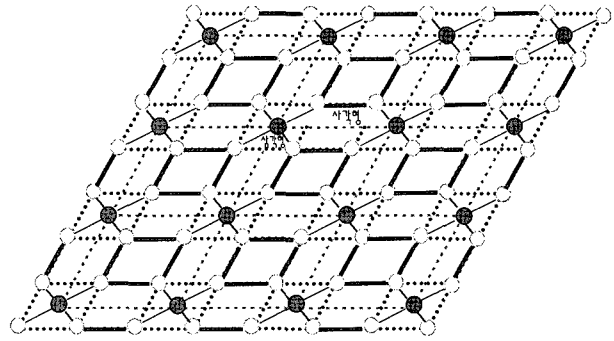
따라서 연산 적용 후 확장된 사이클의 길이는 두 부분을 합한 길이에 해당하므로 $m+n-2s$ 가 된다. □

계층간 연결 간선(inter-level edge)들은 크게 두 가지 부류로 구분이 가능한데, 하나는 해당 간선이 삼각형(triangle)을 구성하는 간선 그룹($E_q(P^N)$)과 사각형(quadrangle)을 구성하는 간선 그룹($E_r(P^N)$)으로 다음과 같이 분류할 수 있다.

$$E_q(P^N) = \bigcup_{0 \leq l \leq N-1} \{((l, 2x, y), (l, 2x+1, y)) \mid 0 \leq x < 2^{N-l-2} \ \&\& \ 0 \leq y < 2^{N-l-1} \ \&\& \ (l, 2x, y), (l, 2x+1, y) \in V(P^N)\} \\ \bigcup_{0 \leq l \leq N-2} \{((l, x, 2y), (l, x, 2y+1)) \mid 0 \leq x < 2^{N-l-1} \ \&\& \ 0 \leq y < 2^{N-l-2} \ \&\& \ (l, x, 2y), (l, x, 2y+1) \in V(P^N)\}$$

$$E_r(P^N) = \bigcup_{0 \leq l \leq N-2} \{((l, 2x-1, y), (l, 2x, y)) \mid 1 \leq x < 2^{N-l-2} \ \&\& \ 0 \leq y < 2^{N-l-1} \ \&\& \ (l, 2x-1, y), (l, 2x, y) \in V(P^N)\} \\ \bigcup_{0 \leq l \leq N-2} \{((l, x, 2y-1), (l, x, 2y)) \mid 0 \leq x < 2^{N-l-1} \ \&\& \ 1 \leq y < 2^{N-l-2} \ \&\& \ (l, x, 2y-1), (l, x, 2y) \in V(P^N)\}$$

간선 그룹 $E_q(P^N)$ 에 속하는 l -계층 정점들의 경우는 상위 계층인 $(l+1)$ -계층의 동일한 정점으로 연결되어 인접된 계층의 정점 상호간에 삼각형을 형성하게 되지만, $E_r(P^N)$ 에 속하는 l -계층 정점들의 경우는 상위 계층인 $(l+1)$ -계층의 동일한 정점으로 연결되는 것이 아니라 인접된(adjacent) 정점으로 각각 연결됨으로써 인접된 계층의 정점 상호간에 사각형을 형성하게 된다((그림 3) 참조).



—	: $E_q(P^3)$ 소속 0-계층 간선들
⋯⋯⋯	: $E_r(P^3)$ 소속 0-계층 간선들
—	: 0- 및 1-계층간 연결 간선들
⋯⋯⋯	: P^3 내 1-계층 소속 간선들
○	: P^3 내 0-계층 소속 정점들
●	: P^3 내 1-계층 소속 정점들

(그림 3) P^3 내 0- 및 1-계층 구성 평면도

[보조정리 2] $0 \leq l < N-1$ 을 만족하는 피라미드 내의 계층을 나타내는 임의의 정수 l 에 대해 계층- l 과 계층- $(l+1)$ 사이를 연결하는 간선들에 의해 형성되는 사각형 및 삼각형은 각각 $(4^{N-l-1} - 2^{N-l})$ 개와 4^{N-l-1} 개가 존재한다.

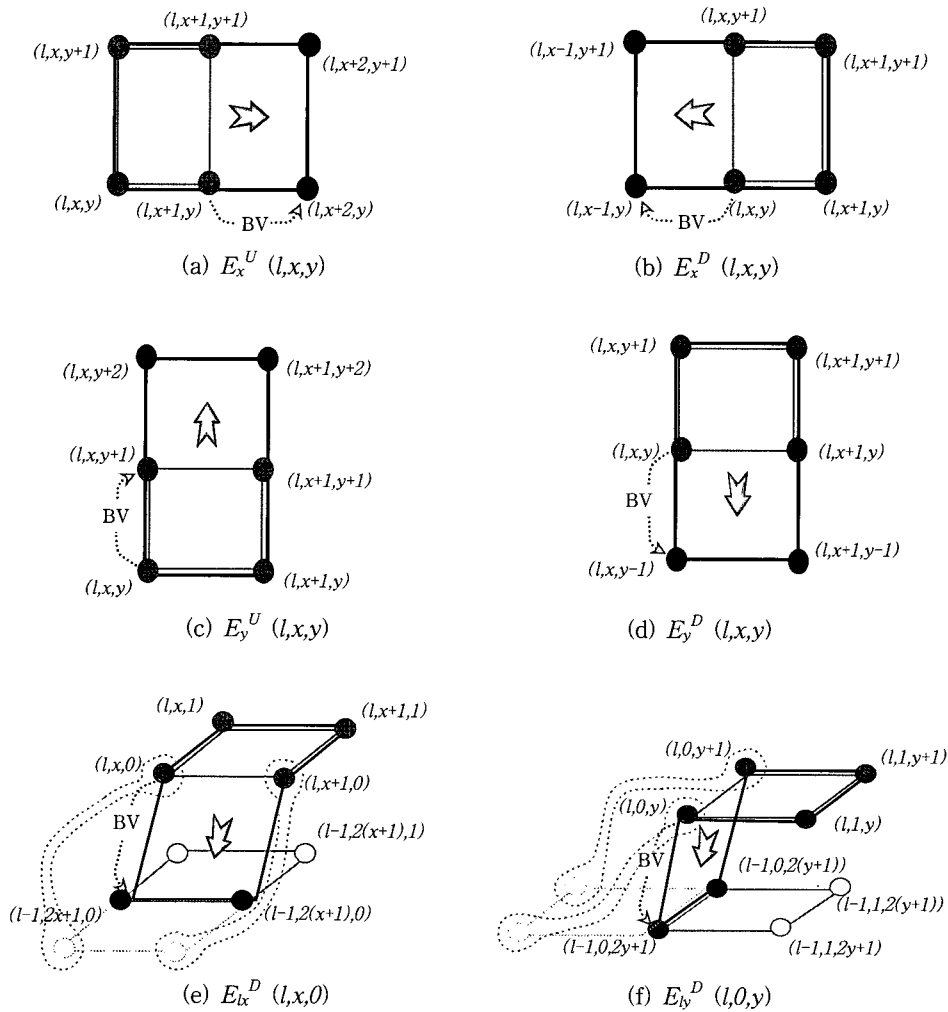
[증명] 계층- l 에 속한 $2^{N-l} \cdot (2^{N-l-1} - 1)$ 개의 전체 간선들 중에서 상위 계층인 $(l+1)$ -계층으로의 연결 간선들과 인접된 두 개의 간선 그룹 $E_q(P^N)$ 과 $E_r(P^N)$ 의 크기는 (그림 4)를 참조하여 일반화시키면 각각 다음과 같은 식으로 나타낼 수 있음을 확인할 수 있다.

$$|E_q(P^N)| = 4^{N-l-1} - 2^{N-l} \\ |E_r(P^N)| = 4^{N-l-1}$$

결국 이러한 간선들마다 각각 하나씩의 사각형 또는 삼각형을 구성하는데 기여하게 되므로 전체적으로 계층- l 및 계층- $(l+1)$ 상호간 연결 간선들에 의해 형성되는 사각형 및 삼각형의 개수는 각각 $(4^{N-l-1} - 2^{N-l})$ 개와 4^{N-l-1} 개가 된다. □

[정의 4] 피라미드 그래프 상의 임의의 계층- l 에 위치한 정점들로 구성된 사각형 $(l, a, b), (l, a+1, b), (l, a+1, b+1)$ 및 $(l, a, b+1)$ 에서 「기본 정점(BV; base vertex)」이란 사각형을 구성하고 있는 네 개의 점들을 대표하는 하나의 정점으로서 x -좌표 및 y -좌표값이 최소에 해당하는 좌하(左下: left-lower)에 위치한 (l, a, b) 로 지정한다. 따라서 기본 정점 $BV = ((l, a, b))$ 에 의해 정의된 사각형에 기반을 두고 각 축을 따라 진행되는 사각형 확장의 유형을 다음과 같이 정의한다.

- (1) $E_x^U(l, x, y)$: 인수로 주어진 기본 정점을 바탕으로 확장을 적용한 후 기본 정점은 $BV = (l, x, y)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 각각 $C = (\dots, (l, x, y+1), (l, x+1, y+1), (l, x+1, y), (l, x, y), \dots)$ 에서 $C' = (\dots, (l, x, y+1), \underline{(l, x+1, y+1)}, (l, x+2, y+1), (l, x+2, y), \underline{(l, x+1, y)}, (l, x, y), \dots)$ 로 확장된다.



(그림 4) 기본 정점에 기반한 사각형의 다양한 확장

(2) $E_x^U(l,x,y)$: 인수로 주어진 기본 정점을 바탕으로 확장을 적용한 후 기본 정점은 $BV'=(l,x,y+1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 각각 $C = (\dots, (l,x,y), (l,x,y+1), (l,x+1,y+1), (l,x+1,y), \dots)$ 에서 $C' = (\dots, (l,x,y), \underline{(l,x,y+1)}, (l,x,y+2), (l,x+1,y+2), \underline{(l,x+1,y+1)}, (l,x+1,y), \dots)$ 로 확장된다.

(3) $E_x^D(l,x,y)$: 인수로 주어진 기본 정점을 바탕으로 확장을 적용한 후 기본 정점은 $BV'=(l,x-1,y)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 각각 $C = (\dots, (l,x+1,y), (l,x,y), (l,x,y+1), (l,x+1,y+1), \dots)$ 에서 $C = (\dots, (l,x+1,y), \underline{(l,x,y)}, (l,x-1,y), \underline{(l,x,y+1)}, (l,x+1,y+1), \dots)$ 로 확장된다.

(4) $E_y^D(l,x,y)$: 인수로 주어진 기본 정점을 바탕으로 확장을 적용한 후 기본 정점은 BV' 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 각각 $C = (\dots, (l,x,y+1), (l,x,y), (l,x+1,y), (l,x+1,y+1), \dots)$ 라고 가정하면, 확장 후 기본 정점 및 부분 사이클(또는 경로)은 각각 $BV=(l,x,y-1)$ 과 $C = (\dots, (l,x,y+1), \underline{(l,x,y)}, (l,x,y-1), (l,x+1,y-1), \underline{(l,x+1,y)}, (l,x+1,y+1), \dots)$ 로 변경 및 추가된다.

(5) $E_x^D(l,x,0)$: 인수로 주어진 기본 정점의 확장 전 x -

좌표값을 나타내는 x 값이 홀수(odd)라고 가정할 때, 기본 정점을 바탕으로 확장을 적용한 후 기본 정점은 $BV'=(l-1,2x+1,-1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 각각 $C = (\dots, (l,x,0), (l,x+1,0), \dots)$ 에서 $C' = (\dots, \underline{(l,x,0)}, (l-1,2x+1,0), (l-1,2(x+1),0), \underline{(l,x+1,0)}, \dots)$ 로 확장된다. 여기서 새로운 기본정점으로 지정된 BV' 은 실제로는 존재하지 않는 가상의 정점으로 표현하였는데, 그 이유는 인접 계층의 가장자리에 위치한 정점을 원래의 기본 계층의 확장으로 해석가능하게 함으로써 연속된 확장 적용시 일관성을 유지시키기 위한 배려이다.

(6) $E_y^D(l,0,y)$: 인수로 주어진 기본 정점의 확장 전 y -좌표값을 나타내는 y 값이 홀수(odd)라고 가정할 때, 기본 정점을 바탕으로 확장을 적용한 후 기본 정점은 $BV'=(l-1,-1,2y+1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 각각 $C = (\dots, (l,0,y), (l,0,y+1), \dots)$ 에서 $C' = (\dots, \underline{(l,0,y)}, (l-1,0,2y+1), (l-1,0,2(y+1)), \underline{(l,0,y+1)}, \dots)$ 로 확장된다. 여기서 위의 (5)에서와 동일한 이유로 인해 새로운 기본정점으로 지정된 BV' 은 실제로는 존재하지 않는 가상의 정점으로 표현된다.

```

Algorithm Hamiltonian Cycle Construction Part 1 of 3
Input:  $N$  /* dimension of Pyramid graph */
Output: Cycle  $C$  with max length  $(4^N-1)/3$  (ordered sequence of vertices in  $V(P^N)$ )

/* Global variables declaration */
integer  $N$ ; /* dimension(height) of Pyramid graph */
integer  $BVl$ ,  $BVx$ ,  $BVy$ ; /* Basic Vertex information(coordinate) */
sequence  $C$  that will maintain ordered elements of vertices in  $V(P^N)$ , initially empty;
integer  $total\_len$ ; /* total length of resulting cycle */

main() {
     $BVl=N-2$ ;  $BVx=0$ ;  $BVy=0$ ; /* BV Initialization */
     $C = \{(N-1,0,0),(N-2,0,0),(N-2,1,0),(N-2,1,1),(N-2,0,1)\}$ ; /* C Initialization */
     $total\_len = 5$ ;
    call  $E_k^D(N-2,0,0)$  routine;
    for ( $l=N-3$ ;  $l>=0$ ;  $l--$ ) {
        if ( $(N-l)\%2 == 0$ ) { /* for even levels */
            call  $even\_level\_exp()$  routine;
            call  $E_k^D(l,2^{N-l-2}-1,0)$  routine; /* Inter-level expansion by  $E_k^D(l,x,0)$  */
        } else { /* for odd levels */
            call  $odd\_level\_exp()$  routine;
            call  $E_y^D(l,0,2^{N-l-2}-1)$  routine; /* Inter-level expansion by  $E_y^D(l,0,y)$  */
        }
    }
}

even_level_exp() { /* processing routine for even levels */
    call  $E_x^U(BVl,BVx,BVy)$  routine;
     $tmp\_l=BVl$ ;  $tmp\_x=BVx$ ;  $tmp\_y=BVy$ ; /* temporary save */
    for ( $i=0$ ;  $i<2^{N-l-3}$ ;  $i++$ ) { /* LEFT-side expansion */
        if ( $i \neq 0$ )
            call  $E_y^U(BVl,BVx,BVy)$  routine;
        call  $E_y^U(BVl,BVx,BVy)$  routine;
        if ( $i\%2 == 0$ ) /* for even loop for index i */
            for ( $j=0$ ;  $j<(2^{N-l-1}-2)$ ;  $j++$ )
                call  $E_x^U(BVl,BVx,BVy)$  routine;
        else /* for odd loop for index i */
            for ( $j=0$ ;  $j<(2^{N-l-1}-2)$ ;  $j++$ )
                call  $E_x^D(BVl,BVx,BVy)$  routine;
    }
     $BVl=tmp\_l$ ;  $BVx=tmp\_x$ ;  $BVy=tmp\_y$ ;
    for ( $i=0$ ;  $i<2^{N-l-3}$ ;  $i++$ ) { /* RIGHT-side expansion */
        if ( $i \neq 0$ )
            call  $E_y^D(BVl,BVx,BVy)$  routine;
        call  $E_y^D(BVl,BVx,BVy)$  routine;
        if ( $i\%2 == 0$ ) /* for even level for index i */
            for ( $j=0$ ;  $j<(2^{N-l-1}-2)$ ;  $j++$ )
                call  $E_x^U(BVl,BVx,BVy)$  routine;
        else /* for odd loop for index i */
            for ( $j=0$ ;  $j<(2^{N-l-1}-2)$ ;  $j++$ )
                call  $E_x^D(BVl,BVx,BVy)$  routine;
    }
}
}

```

3. 해밀톤 사이클 특성

본 절에서는 주어진 피라미드 그래프의 주요 성질들을 이용하여 그래프 내에서 해밀톤 사이클이 존재함을 증명하고 해당 사이클을 찾는 알고리즘을 제시한다. 먼저 앞 절에서 정의된 사이클 결합 연산을 이용하여 인접된 사각형 형태의 사이클들을 확장시키는 일련의 과정을 통해 사이클의 크기를 연속적으로 확대시켜 나감으로써 최종적으로 해밀톤 사

이클 구성이 가능한 알고리즘을 제안한다.

제시된 알고리즘 Hamiltonian Cycle Construction은 앞에서 언급된 사이클 결합 연산의 개념을 이용하여 임의의 높이 N 인 피라미드 그래프에서 모든 정점들을 한 번씩만 방문하게 되는 해밀톤 사이클을 찾아주는 알고리즘이다. 본 알고리즘의 실행 예를 (그림 5)에서 보여주고 있으며, 알고리즘에 대한 정확성(correctness) 검증은 다음의 (정리 1)의

```

Algorithm Hamiltonian_Cycle_Construction Part 2 of 3
odd_level_exp() { /* processing routine for odd levels */
  call  $E_y^U(BVl, BVx, BVy)$  routine;
  tmp_l=BVl; tmp_x=BVx; tmp_y=BVy; /* temporary save */
  for (i=0; i<2N-1-3; i++) { /* LEFT-side expansion */
    if ( i != 0 )
      call  $E_x^D(BVl, BVx, BVy)$  routine;
      call  $E_y^D(BVl, BVx, BVy)$  routine;
    if (i%2 == 0) { /* for even loop for index i */
      for (j=0; j<(2N-1-1-2); j++)
        call  $E_y^U(BVl, BVx, BVy)$  routine;
    } else { /* for odd loop for index i */
      for (j=0; j<(2N-1-1-2); j++)
        call  $E_y^D(BVl, BVx, BVy)$  routine;
    }
  }
  BVl=tmp_l; BVx=tmp_x; BVy=tmp_y;
  for (i=0; i<2N-1-3; i++) { /* RIGHT-side expansion */
    if ( i != 0 )
      call  $E_x^U(BVl, BVx, BVy)$  routine;
      call  $E_y^U(BVl, BVx, BVy)$  routine;
    if (i%2 == 0) { /* for even level for index i */
      for (j=0; j<(2N-1-1-2); j++)
        call  $E_y^U(BVl, BVx, BVy)$  routine;
    } else /* for odd loop for index i */
      for (j=0; j<(2N-1-1-2); j++)
        call  $E_y^D(BVl, BVx, BVy)$  routine;
  }
}

```

증명 과정을 통해 보여주고 있다.

[정리 1] 높이가 N 인 주어진 피라미드 P^N 에는 길이 $(4^N-1)/3$ 인 해밀톤 사이클이 존재한다.

[증명] 피라미드 그래프의 각 계층 내의 정점들은 정방형 2-차원 메쉬(mesh) 구조를 형성하고 있으므로 앞에서 정의한 사이클 결합 연산(또는 사각형 확장)을 이용하여 인접 사각형들을 일정한 순서에 따라 차례대로 확장해 가는 과정을 통해 사이클의 크기를 점진적으로 확장시키게 되며, (보조정리 1)에 의하면 인접 계층인 계층-1 및 계층-(1-1) 사이를 연결하는 간선들에 의해 형성된 사각형의 개수가 개 존재하고 있음을 확인할 수 있으므로 이러한 사각형들 중 하나를 이용하면 인접 계층 상호간에도 사각형 확장이 가능함을 알 수 있으므로 이러한 확장 과정을 통해 최대 크기의 사이클로 확장이 가능함을 확인할 수 있다.

이러한 사각형 확장 과정을 표현하고 있는 주어진 알고리즘 Hamiltonian_Cycle_Construction를 참조하여 이 과정에서 생성되는 최종 사이클의 길이가 해밀톤 사이클에 해당하는지 여부를 확인하기 위해 사각형 확장이 적용되는 횟수를 참고하여 계산하고자 한다.

(1) 초기화 과정에 해당하는 시작 시점에서는 각각 하나 및 네 개의 정점으로 구성된 상위 두 개 계층에 해당하는 계층 $-(N-1)$ 및 계층 $-(N-2)$ 에 해당하는 정점들로 구성된 사이클을 최초 사이클로 지정하여 시작하게 되므로 시작 시점의 초기 사이클 길이는 5로부터 시작한다(그림 5)의 ①부분 참조).

```

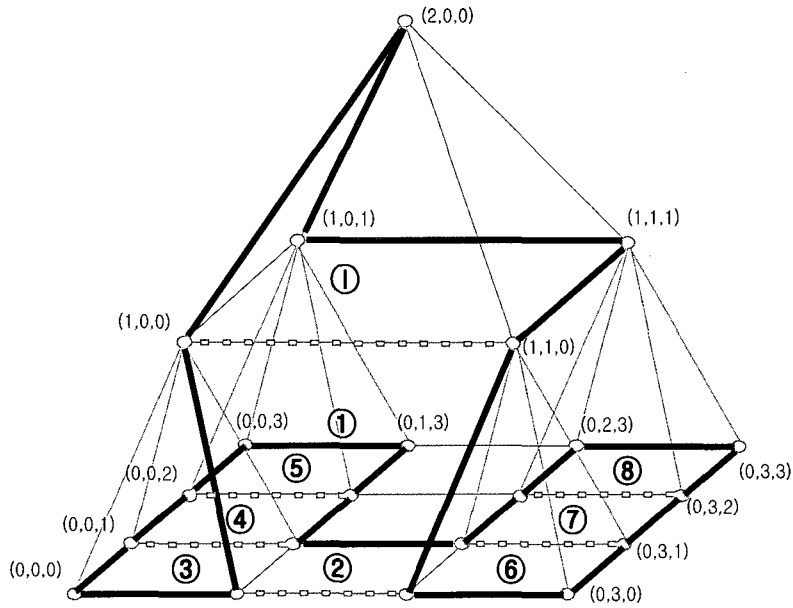
Algorithm Hamiltonian_Cycle_Construction Part 3 of 3
/* each expansion routine( $E_x^U, E_y^U, E_x^D, E_y^D, E_k^D, E_l^D$ ) updates BV
coordinate,
and insert new vertices to sequence C for constructing a larger
size cycle */
 $E_x^U$ (int l, int x, int y) {
  C = C U {(l,x+2,y+1), (l,x+2,y)} between vertices (l,x+1,y+1)
& (l,x+1,y);
  BVx = x+1;
  total_len = total_len+2;
}
 $E_y^U$ (int l, int x, int y) {
  C = C U {(l,x,y+2), (l,x+1,y+2)} between vertices (l,x,y+1) &
(l,x+1,y+1);
  BVy = y+1;
  total_len = total_len+2;
}
 $E_x^D$ (int l, int x, int y) {
  C = C U {(l,x-1,y), (l,x-1,y+1)} between vertices (l,x,y) &
(l,x,y+1);
  BVx = x-1;
  total_len = total_len+2;
}
 $E_y^D$ (int l, int x, int y) {
  C = C U {(l,x,y-1), (l,x+1,y-1)} between vertices (l,x,y) &
(l,x+1,y);
  BVy = y-1;
  total_len = total_len+2;
}
 $E_k^D$ (int l, int x, int y) {
  C = C U {(l-1,2x+1,y), (l-1,2(x+1),y)} between vertices (l,x,y)
& (l,x+1,y);
  BVl = l-1; BVx = 2x+1; BVy = -1;
  total_len = total_len+2;
}
 $E_l^D$ (int l, int x, int y) {
  C = C U {(l-1,x,2y+1), (l-1,x,2(y+1))} between vertices (l,x,y)
& (l,x,y+1);
  BVl = l-1; BVx = -1; BVy = 2y+1;
  total_len = total_len+2;
}

```

(2) 계층간 연결 간선을 이용하는 확장의 경우는 $1 \leq l \leq N-2$ 에 해당하는 각 계층- l 에 대해 각각 한 번씩의 사각형 확장 E_k^D 또는 E_l^D 가 적용되므로 총 $(N-2)$ 회에 걸친 사각형 확장 연산이 발생하였음을 알 수 있다. 한편 (성질 1)에 의하면 각 하나의 간선만을 공유하고 있는 임의의 사이클과 사각형을 사이클 결합 연산 \oplus 에 해당하는 확장(E_k^D, E_l^D) 과정을 거치게 됨에 따라 매 확장 연산마다 새로운 두 개씩의 정점이 추가되게 되므로 결과적으로 생성되는 사이클의 크기에 영향을 미치는 기여도는 2가 된다(그림 5)의 ①부분 참조).

따라서 인접 계층간 확장 과정이 최종 사이클 크기에 기여하는 값은 전체적으로 $2 \cdot (N-2)$ 가 된다.

(3) 마지막으로 피라미드의 각 계층 내에서 이루어지는



[참고] ①는 초기 구성을, 숫자 ①~⑧은 확장 순서를 의미
(그림 5) 알고리즘 실행 예 (N=3인 경우)

사이클 확장($E_x^U, E_y^U, E_x^D, E_y^D$) 과정을 통해 이루어지는 최종 사이클 기여도를 분석하기 위해 먼저 $0 \leq l \leq N-3$ 에 해당하는 각 계층별 사이클 확장의 개수를 집계하면 다음 식으로 환산할 수 있다(그림 5)의 ②~⑧부분 참조.

$$\begin{aligned}
 \text{Total Number of Expansions} &= \sum_{l=0}^{N-3} [1+2 \cdot \{2^{N-l-3} \cdot (2^{N-l-2}) + 2 \cdot 2^{N-l-3} - 1\}] \\
 &= \sum_{l=0}^{N-3} [1+2 \cdot (2^{2N-2l-4}-1)] \\
 &= 2^{2N-3} \cdot \sum_{l=0}^{N-3} [2^{2l}] - (N-2) \\
 &= 2^{2N-3} \cdot \left(\frac{1-2^{-2(N-2)}}{1-2^{-2}} \right) - (N-2) \\
 &= \frac{1}{3} \cdot (2^{2N-1}-8) - (N-2)
 \end{aligned}$$

위의 (2)에서와 같이 (보조정리 1)에 의거 사이클 결합 연산 \oplus 에 해당하는 확장(E_{kx}^D, E_{ky}^D) 과정에서 매 확장 연산마다 최종 사이클의 크기에 영향을 미치게 되는 기여도를 의미하는 2만큼을 곱한 값 $\frac{2}{3} \cdot (2^{2N-1}-8) - 2 \cdot (N-2)$ 가 최종 사이클 기여도가 된다.

종합적으로 위의 (1), (2) 및 (3)에서 분석된 사이클 크기에 영향을 준 각 기여도 성분들을 합하면 다음과 같은 값의 최종 사이클 크기에 해당하는 값을 구할 수 있다.

$$\begin{aligned}
 \text{Total Length of Cycle} &= 5 + 2 \cdot (N-2) + \frac{2}{3} \cdot (2^{2N-1}-8) - \\
 &2 \cdot (N-2) = \frac{4^N-1}{3}
 \end{aligned}$$

따라서 이 값은 높이 N인 피라미드 그래프 P^N 의 전체 정점 개수에 해당하는 값을 의미하므로 해밀톤 사이클이 존재함을 의미한다. □

4. 결론

본 논문에서는 다중 병렬처리시스템에서의 영상처리 및 그래프 문제 해결에 다양하게 응용됨에 따라 관심을 받고 있는 상호연결망 그래프 중의 하나인 피라미드를 대상으로 사이클 특성을 분석하였다.

피라미드 그래프 내에 존재하는 사각형 구조의 사이클들을 이용하는 연속적인 사이클 확장 연산을 통해서 사이클의 크기를 확대시켜 나가는 과정을 통해서 그래프 내 최대 크기의 해밀톤 사이클을 찾을 수 있는 알고리즘을 제시하고 해당 알고리즘을 이용하면 높이가 N인 피라미드 내에서 항상 $(4^N-1)/3$ 의 길이를 갖는 해밀톤 사이클이 존재함을 증명할 수 있었다.

이러한 상호연결망 내에서의 해밀톤 사이클 특성 분석이 갖는 의의는 그래프 이론적인 의미뿐만 아니라 다중 프로세서들로 구성된 병렬처리시스템 상에서 프로세서 상호간 방송 또는 멀티캐스팅 형태의 메시지 전송과정에서 사이클을 하부 구조로 이용하게 되는 응용 등에서 활용이 가능할 것으로 기대된다.

참고 문헌

[1] F. Berman and L. Snyder, "On mapping parallel algorithms into parallel architectures," *J. of Parallel and Distrib.*

Comput., Vol.4, pp.439-458, 1987.

[2] B. Monien and H. Sudborough, "Embedding one inter-connection network in another," *Computing Supplement*, Vol.7, pp.257-282, 1990.

[3] Y. Saad and M. H. Schultz, "Topological properties of hypercubes," *IEEE Trans. on Comput.*, Vol.37, pp.867-872, 1988.

[4] F. T. Leighton, "Introduction to Parallel Algorithms and Architectures : Arrays, Trees, Hypercubes," *Morgan Kaufmann Pub.*, CA., 1992.

[5] R. Miller and Q. F. Stout, "Data Movement Techniques for the Pyramid Computer," *SIAM J. on Comput.*, Vol.16, No.1, pp.38-60, 1987.

[6] Q. F. Stout, "Mapping Vision Algorithms to Parallel Architectures," *Proc. of the IEEE*, pp.982-995, 1988.

[7] D. M. C. Ip, C. K. Y. Ng, L. K. L. Pun, M. Hamdi, and I. Ahmad, "Embedding Pyramids into 3D Meshes," *Proc. of 1993 Int'l Conf. on Paral. and Distrib. Sys.*, pp.348-352, 1993.

[8] K. -L. Chung and Y. -W. Chen, "Mapping Pyramids into 3-D Meshes," *Nordic J. of Computing*, Vol.2, No.3, pp. 326-337, 1995.

[9] 장정환, "피라미드의 3-차원 메쉬로의 신장율 개선 임베딩", 정보처리학회논문지-A, Vol.10-A, No.6, pp.627-634, 2003.

[10] Y. C. Tseng, D. K. Panda and T. H. Lai, "A Trip-based Multicasting Model in Wormhole-routed Networks with Virtual Channels," *IEEE Trans. on Paral. and Distrib. Sys.*, Vol.7, No.2, pp.138-150, 1996.



장 정 환

e-mail : jhchang@pufs.ac.kr

1983년 경북대학교 전자공학과(학사)

1985년 한국과학기술원(KAIST) 전산학과 (석사)

1998년 한국과학기술원(KAIST) 전산학과 (공학박사)

1985년~2000년 (주)케이티(KT) 선임연구원

2000년~현재 부산외국어대학교 디지털미디어학부 조교수

관심분야 : 상호연결망, 그래프 이론 및 응용, 통신망 보안 등