

피라미드 그래프로의 링 임베딩

장정환*

*부산외국어대학교 디지털정보공학부

e-mail: jhchang@pufs.ac.kr

An Embedding of Ring into Pyramid Graphs

Jung-Hwan Chang*

*Division of Digital & Information Eng., Pusan University of Foreign Studies

요 약

본 논문에서는 피라미드 그래프를 대상으로 링을 임베딩하는 문제를 다룬다. 사이클 확장 연산을 이용하여 사이클의 크기를 확대시켜 나가는 일련의 과정을 통하여 최대 크기의 링을 의미하는 헤밀톤 사이클을 찾을 수 있는 알고리즘을 제시함으로써 임의의 높이 N 인 피라미드 그래프 내에 길이 $4^N-1/3$ 인 링을 임베딩 할 수 있음을 증명한다.

1. 서론

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 중요한 의미를 갖는데, 그 이유는 해당 그래프에 대한 이론적인 특성에 해당 병렬처리시스템의 성능이 의존적일 수밖에 없기 때문이다. 알고리즘에 내재된 자료구조를 실제 구현된 시스템 구조에 효율적으로 매핑시키거나, 상호연결망 상호간 시뮬레이션을 위한 체계적인 접근으로서 그래프 임베딩(embedding) 문제가 대두하게 되었다[1,2].

높이 N 인 피라미드(pyramid)의 경우 $(4^N-1)/3$ 개의 정점들을 갖는 계층적 구조의 그래프로써 오늘날 각광을 받고 있는 멀티미디어 분야의 영상처리 관련 기능을 효율적으로 처리해 줄 수 있는 구조적 특성 때문에 관련 분야를 중심으로 널리 이용되어 왔다[3].

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) $e=(v,w)$ 는 $F(v)$ 와 $F(w)$ 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 하는 연구결과들이 비교적 잘 알려져 있다[4,5,6,7].

본 논문에서는 피라미드 그래프를 대상으로 링을 임베딩하는 문제와 동일한 의미를 갖는 헤밀톤 사이클을 찾는 문제를 다룬다. 피라미드 그래프의 특성

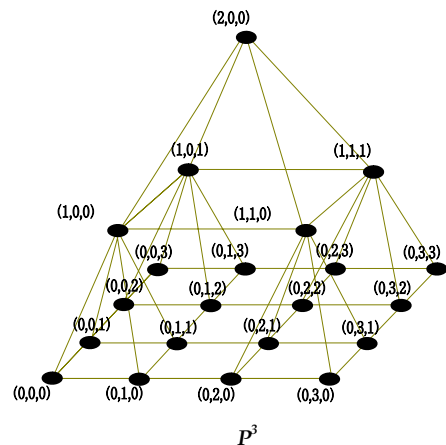
을 활용한 사이클 확장 연산을 이용함으로써 모든 정점들이 포함되는 최대 크기의 헤밀톤 사이클을 찾을 수 있는 알고리즘을 제시함으로써 최대 크기의 링을 임베딩할 수 있음을 보여주게 된다.

2. 피라미드 그래프 특성

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

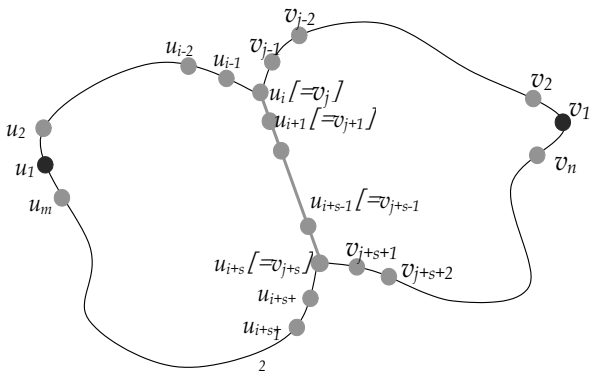


(그림 1) 높이 3인 피라미드 그래프 예

[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.

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

이러한 두 개의 사이클 C_m 과 C_n 을 대상으로 하는 「사이클 결합 연산(cycle merge operation)」 \oplus 는 다음과 같이 정의된다.

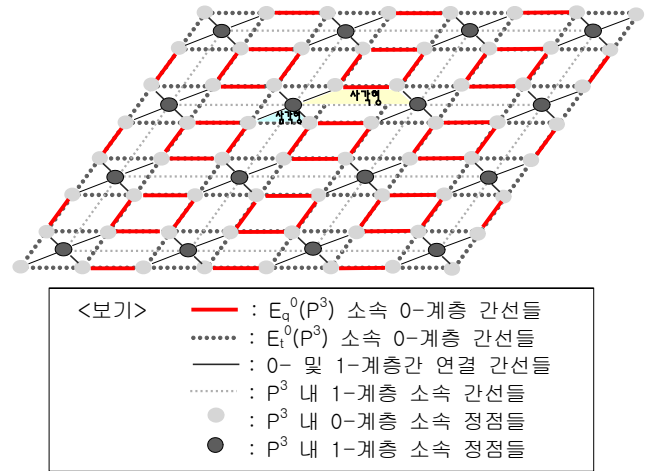
$$C_m \oplus C_n = (u_1, u_2, \dots, u_{i-1}, u_i [=v_j], v_{j-1}, v_{j-2}, \dots, v_2, v_1, v_n, v_{n-1}, \dots, v_{j+s+2}, v_{j+s+1}, v_{j+s} [=u_{i+s}], u_{i+s+1}, u_{i+s+2}, \dots, u_{m-1}, u_m, u_1)$$

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 있는 경우에 (그림 2)를 참조하면 연산의 결과 얻어지는 확장된 사이클의 길이는 $m+n-2s$ 이 된다.

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

다음과 같이 분류할 수 있다((그림 3) 참조).

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



(그림 3) P^3 내 0- 및 1-계층 구성 평면도

이러한 두 부류의 간선 그룹의 크기를 추정해보면 전체적으로 계층- l 및 계층- $(l+1)$ 상호간 연결 간선들에 의해 형성되는 사각형 및 삼각형의 개수는 각각 $(4^{N-1} - 2^{N-1})$ 개와 4^{N-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 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 여섯 가지 형태로 정의할 수 있다.

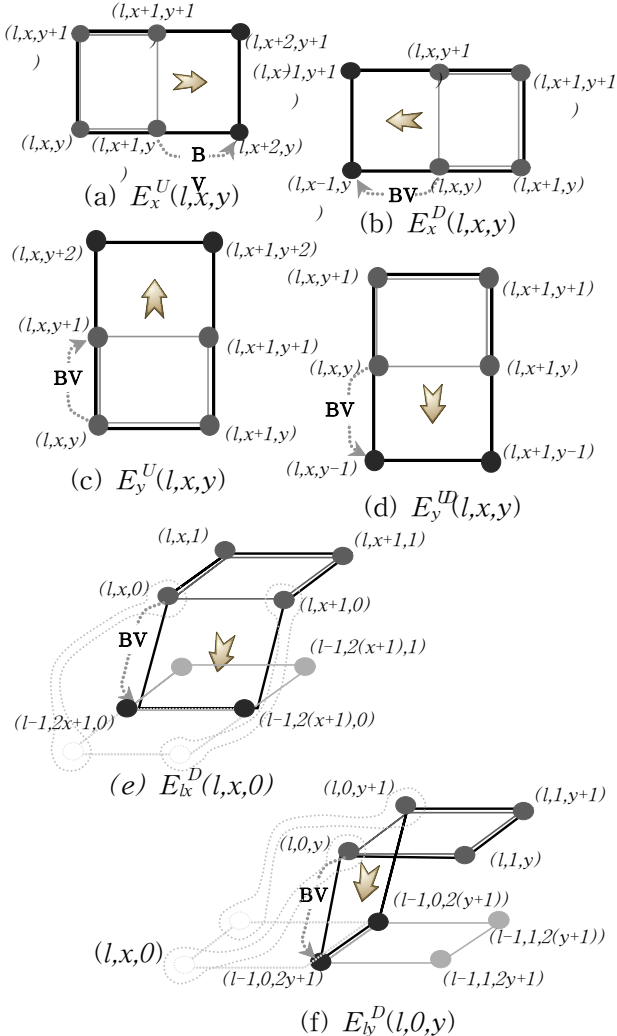
(1) $E_x^U(l, x, y)$: 확장을 통해 기본 정점은 $BV = (l, x+1, y)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 $C = (\dots, (l, x, y+1), (l, x+1, y+1), (l, x+1, y), (l, x, y), \dots)$ 에서 $C' = (\dots, (l, x, y+1), (l, x+1, y+1), (l, x+2, y+1), (l, x+2, y), (l, x+1, y), (l, x, y), \dots)$ 로 변경 및 추가된다.

(2) $E_y^U(l, x, y)$: 확장을 통해 기본 정점은 $BV = (l, x, y+1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 $C = (\dots, (l, x, y), (l, x, y+1), (l, x+1, y+1), (l, x+1, y), \dots)$ 에서 $C' = (\dots, (l, x, y), (l, x, y+1), (l, x, y+2), (l, x+1, y+2), (l, x+1, y+1), (l, x+1, y), \dots)$ 로 변경 및 추가된다.

(3) $E_x^D(l, x, y)$: 확장을 통해 기본 정점은 $BV = (l, x-1, y)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 $C = (\dots, (l, x+1, y), (l, x, y), (l, x, y+1), (l, x+1, y+1), \dots)$ 에서 $C' = (\dots, (l, x+1, y), (l, x, y), (l, x-1, y), (l, x-1, y+1), (l, x, y+1), (l, x+1, y+1), \dots)$

...)로 변경 및 추가된다.

(4) $E_y^D(l,x,y)$: 확장을 통해 기본 정점은 $BV=(l,x,y-1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 $C=(\dots,(l,x,y+1),(l,x,y),(l,x+1,y),(l,x+1,y+1),\dots)$ 에서 $C'=(\dots,(l,x,y+1),(l,x,y),(l,x,y-1),(l,x+1,y-1),(l,x+1,y),(l,x+1,y+1),\dots)$ 로 변경 및 추가된다.



(그림 4) 기본 정점에 기반한 사각형의 확장 형

(5) $E_k^D(l,x,0)$: 인수로 주어진 확장 전 x -좌표값을 나타내는 x 값이 홀수(odd)라고 가정할 때, 확장을 통해 기본 정점은 $BV=(l-1,2x+1,-1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 $C=(\dots,(l,x,0),(l,x+1,0),\dots)$ 에서 $C'=(\dots,(l,x,0),(l-1,2x+1,0),(l-1,2(x+1),0),(l,x+1,0),\dots)$ 로 변경 및 추가된다. 여기서 새로운 VB로 지정된 정점은 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) $E_b^D(l,0,y)$: 인수로 주어진 확장 전 y -좌표값을 나타내는 y 값이 홀수라고 가정할 때, 확장을 통해 기본 정점은 $BV=(l-1,-1,2y+1)$ 로 재설정되고, 해당 부분 사이클(또는 경로)은 확장 전후에 $C=(\dots,(l,0,y),(l,0,y+1),\dots)$ 에서 $C'=(\dots,(l,0,y),(l-1,0,2y+1),(l-1,0,2(y+1)),(l,0,y+1),\dots)$ 로 변경 및 추가된다. 여기서도 위의 (5)에서와 동일하게 VB

로 지정된 정점은 가상의 정점에 해당한다.

3. 피라미드 그래프의 링 임베딩

앞 절에서 정의된 사이클 결합 연산을 이용하여 인접된 사각형 형태의 사이클들을 확장시키는 일련의 과정을 통해 사이클의 크기를 연속적으로 확대시켜 나감으로써 최종적으로 해밀톤 사이클 구성이 가능한 알고리즘을 제안한다.

Algorithm *Max_Ring-Embedding P. 1 of 3*

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

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

```

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

제시된 알고리즘은 앞에서 언급된 사이클 결합 연산의 개념을 이용하여 임의의 높이 N 인 피라미드에서 모든 정점들을 한 번씩만 방문하여 최대 길이인 $(4^N-1)/3$ 에 해당하는 링을 임베딩시킬 수 있게 된다. 본 알고리즘에 대한 정확성 검증은 다음 (정리 1)의 증명 과정을 통해 증명하고자 있다.

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

[증명] 사이클 결합 연산을 이용하여 인접 사각형들을 차례대로 확장해 가는 일련의 과정을 통해 사이클의 크기를 점진적으로 확장시키게 된다.

주어진 알고리즘의 수행을 통해 생성되는 최종 사이클의 길이를 계산하고자 한다.

(1) 초기화 과정에 해당하는 시작 시점에서는 각각 하나 및 네 개의 정점으로 구성된 상위 두 개 계층인 계층- $(N-1)$ 및 계층- $(N-2)$ 에 해당하는 정점들로 구성된 길이 5인 사이클로부터 시작한다.

(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_b^D 가 적용되므로 총 $(N-2)$ 회에 걸친 사각형 확장 연산이 발생하게 되므로 결과적으로 생성되는 사이클의 크기에 대한 기여도는 $2 \cdot (N-2)$ 가 된다.

Algorithm Max Ring-Embedding P. 2 of 3

```

/* processing routine for even levels */
even_level_expansion() {
  call  $E_x^U(BVl, BVx, BVy)$  routine;
  tmp_l=BVl; tmp_x=BVx; tmp_y=BVy;
  /* LEFT expansion */
  for (i=0; i<2N-l-3; i++) {
    if (i!=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 (j=0; j<(2N-l-1-2); j++)
        call  $E_x^U(BVl, BVx, BVy)$  routine;
      } else { /* for odd loop */
      for (j=0; j<(2N-l-1-2); j++)
        call  $E_x^D(BVl, BVx, BVy)$  routine;
      }
  }

  BVl=tmp_l; BVx=tmp_x; BVy=tmp_y;
  /* RIGHT expansion */
  for (i=0; i<2N-l-3; i++) {
    if (i!=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 (j=0; j<(2N-l-1-2); j++)
        call  $E_x^U(BVl, BVx, BVy)$  routine;
      } else { /* for odd loop */
      for (j=0; j<(2N-l-1-2); j++)
        call  $E_x^D(BVl, BVx, BVy)$  routine;
      }
  }
}

```

Algorithm Max Ring-Embedding P. 3 of 3

```

/* processing routine for odd levels */
odd_level_expansion() {
  call  $E_x^U(BVl, BVx, BVy)$  routine;
  tmp_l=BVl; tmp_x=BVx; tmp_y=BVy;
  /* LEFT-side expansion */
  for (i=0; i<2N-l-3; i++) {
    if ( i != 0 )
      call  $E_x^D(BVl, BVx, BVy)$  routine;
      call  $E_x^D(BVl, BVx, BVy)$  routine;
    if (i%2 == 0) { /* for even loop */
      for (j=0; j<(2N-l-1-2); j++)
        call  $E_y^U(BVl, BVx, BVy)$  routine;
      } else { /* for odd loop */
      for (j=0; j<(2N-l-1-2); j++)
        call  $E_y^D(BVl, BVx, BVy)$  routine;
      }
  }

  BVl=tmp_l; BVx=tmp_x; BVy=tmp_y;
  /* RIGHT-side expansion */
  for (i=0; i<2N-l-3; i++) {
    if ( i != 0 )
      call  $E_x^U(BVl, BVx, BVy)$  routine;
      call  $E_x^U(BVl, BVx, BVy)$  routine;
    if (i%2 == 0) { /* for even level */
      for (j=0; j<(2N-l-1-2); j++)
        call  $E_y^U(BVl, BVx, BVy)$  routine;
      } else { /* for odd loop */
      for (j=0; j<(2N-l-1-2); j++)
        call  $E_y^D(BVl, BVx, BVy)$  routine;
      }
  }
}

```

(3) 피라미드의 각 계층 내에서 이루어지는 사이클 확장(E_x^U , E_y^U , E_x^D , E_y^D) 과정을 통해 이루어지는 최종 사이클 기여도 분석을 위해 $0 \leq l \leq N-3$ 에 해당하는 각 계층별 사이클 확장의 개수를 계산하면 다음

식과 같이 나타낼 수 있다.

Total Number of Expansion

$$\begin{aligned}
 &= \sum_{l=0}^{N-3} [1+2 \cdot \{2^{N-l-3} \cdot (2^{N-l-1}-2)+2 \cdot 2^{N-l-3}-1\}] \\
 &= \frac{1}{3} \cdot (2^{2N-1}-8) - (N-2)
 \end{aligned}$$

따라서 최종 사이클 기여도는 $\frac{2}{3} \cdot (2^{2N-1}-8) - 2 \cdot (N-2)$ 이 된다.

따라서 (1), (2) 및 (3)의 결과를 종합하면 최종 사이클 크기는 $(4^N-1)/3$ 이다.

따라서 이 값은 높이 N 인 피라미드의 전체 정점 개수에 해당하는 값을 의미하므로 최대 크기의 링이 임베딩되었음을 의미한다. □

4. 결론

본 논문에서는 다중 병렬처리시스템에서의 영상 처리 및 그래프 문제 해결에 다양하게 응용됨에 따라 관심을 받고 있는 상호연결망 그래프 중의 하나인 피라미드를 대상으로 링을 임베딩시키는 문제를 다루었다.

피라미드 그래프 내에 존재하는 사각형 구조의 사이클들을 이용하는 연속적인 사이클 확장 연산을 통해서 사이클의 크기를 확대시켜 나가는 과정을 통해서 그래프 내에서 헤밀톤 사이클을 찾을 수 있는 알고리즘을 제시함으로써 최대 크기의 링(사이클)이 임베딩 가능함을 증명할 수 있었다.

참고문헌

- [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 interconnection 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] Q. F. Stout, "Mapping Vision Algorithms to Parallel Architectures," *Proc. of the IEEE*, pp.982-995, 1988.
- [5] 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.
- [6] 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.
- [7] 장정환, "피라미드의 3-차원 메쉬로의 신장울 개선 임베딩," *정보처리학회논문지 A*, Vol.10-A, No.6, pp. 627-634, 2003.