

# 팬케익 그래프의 망비용을 개선한 하프팬케익 연결망

김주봉<sup>†</sup>, 서정현<sup>\*\*</sup>, 이형옥<sup>\*\*\*</sup>

## A Half Pancake network that improve the network cost for Pancake graph

JuBong Kim<sup>†</sup>, Jung-Hyun Seo<sup>\*\*</sup>, HyeongOk Lee<sup>\*\*\*</sup>

### ABSTRACT

The pancake graph is node symmetric and is utilized on the data sorting algorithm. We propose a new half pancake graph that improve pancake graph's network cost. The half pancake degree is approximately half of pancakes degree and diameter is  $3n+4$ . The pancake graph's network cost is  $O(1.64n^2)$  and half pancake's is  $O(1.5n^2)$ . Additionally half pancake graph is sub graph of pancake graph. As this result, The several algorithms developed in pancake graph has the advantage of leverage on the pancake by adding constant cost.

**Key words:** Interconnection network, Pancake Graph, Star Graph, Routing Algorithm

### 1. 서 론

지식 정보화 사회에서 재난재해 분석, 유전자 분석, 과학 기술 응용분야의 시뮬레이션 등의 대용량 자료 처리를 위해 고성능 컴퓨터에 대한 연구가 진행되고 있으며 상용화된 시스템이 개발 및 운영되고 있다. 고성능의 컴퓨터를 제작하는 방법으로 여러 대의 컴퓨터를 조합하여 사용하는 병렬처리(parallel processing) 기술이 사용되고 있다. 병렬처리란 다수의 프로세서들이 여러 개의 프로그램들 또는 한 프로그램의 분할된 부분들을 분담하여 동시에 처리하는 기술을 말한다[1].

병렬처리를 위한 MIMD(multiple instruction multiple data)형 컴퓨터는 각 프로세서가 서로 독립

적으로 다양한 프로그램을 수행할 수 있는 컴퓨터로 Cosmic Cube, nCUBE 2, iPSC, Symmetry, FX-8, CM-5 등이 있다. MIMD형 컴퓨터는 크게 공유 메모리를 갖는 다중 프로세서(multi-processor) 시스템과 분산 메모리를 갖는 다중 컴퓨터(multi-computer) 시스템으로 분류할 수 있다[2]. 다중 컴퓨터 시스템의 성능은 각 프로세서의 성능과 프로세서를 연결하는 상호연결망(interconnection network) 구조 및 그에 따라 사용되는 알고리즘에 의해 결정된다. 특히 상호연결망 구조는 하드웨어적인 비용과 관련된 분지수(degree), 소프트웨어적인 비용을 나타내는 지름(diameter), 연결망의 대칭성, 확장성, 고장 허용도 등의 성질에 따라 전체 시스템의 성능에 영향을 미친다[3,4].

\* Corresponding Author : HyeongOk Lee, Address: (540-950) Jungang-ro 255, Suncheon, Chonnam, Republic of Korea, TEL : +82-61-750-3345, FAX : +82-61-750-3340, E-mail : oklee@scnu.ac.kr

Receipt date : Mar. 21, 2014, Revision date : May. 21, 2014  
Approval date : May. 23, 2014

<sup>†</sup> Suncheon High School

(E-mail : vhrxks@hanmail.net)

<sup>\*\*</sup> Dept. of Computer Engineering, Suncheon National University (jhseo@scnu.ac.kr)

<sup>\*\*\*</sup> Dept. of Computer Education, Suncheon National University

\* The first two authors JuBong Kim and Jung-Hyun Seo contributed equally to this work

\* This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2012R1A1A4A01014439)

병렬 컴퓨터를 위한 상호연결망은 각 프로세서들을 노드로, 프로세서들 사이에 통신 채널을 예지로 나타내는 무방향 그래프  $G=(V,E)$ 로 표현 될 수 있다. 임의의 두 프로세서  $v$ 와  $w$  사이에서 양방향 통신 채널이 존재하면 두 프로세서 사이에는 에지  $(v,w)$ 가 있다. 상호연결망의 성능 평가 척도 중 망 비용(network cost)은 분지수 $\times$ 지름이다. 분지수는 하드웨어 설치비용이고 지름은 소프트웨어 처리비용이다. 일반적으로 분지수가 증가하면 지름은 짧아지고 분지수가 줄어들면 지름은 늘어나게 되어 분지수와 지름은 상호 역상관 관계에 있다[2,5].

상호연결망은 노드수를 기준으로 다음 4가지로 나눌 수 있다. 첫째, 메쉬(mesh) 부류[6]는 일반적으로 격자 구조에서  $n\times k$ 개의 격자점에 노드를 나타내고, 둘째, 스타(star) 그래프 부류[7]는  $n$ 개의 정수를 이용하여  $n!$ 개의 노드를 갖고, 셋째, 하이퍼큐브(hypercube) 부류[8]는 이진수를 이용하여 노드 주소를 표현하는 방식으로  $2^n$ 개의 노드를 갖고, 마지막으로 이진수를 이용하여 노드가 조합(combination) 개수를 갖는 하이퍼-스타(hyper-star) 부류[9]로 나눌 수 있다.

팬케익 정렬(pancake sort) 문제는 스택이라는 공간에 쌓여있는 크기가 각각 다른 팬케익을 손님들의 식탁에 놓기 좋도록 가장 큰 팬케익을 스택의 바닥에 두고 차츰 팬케익의 크기가 작아지도록 쌓는 문제이다[10]. [11]에서는  $n$ 차원 팬케익 네트워크를 정의하고 뒤집기 연산에 의한 라우팅과 지름을 분석하였다. 팬케익 네트워크에서 라우팅은 팬케익 정렬문제에서 정렬연산의 프로세스와 같고 팬케익 네트워크의 지름은 팬케익 정렬문제에서 정렬연산 최대 횟수와 같다. 팬케익 그래프에서 정렬을 위해 요구되는 연산의 최소 횟수는  $(15/14)n$ 과  $(18/11)n$  사이로 정리 하였지만 이것이 최솟값임을 증명하진 못하였다[12].

본 연구에서는 스타 그래프 부류인 팬케익 그래프의 분지수를  $1/2$  정도 줄이면서 망 비용을 개선한 새로운 하프팬케익(half pancake) 그래프를 제안한다. 본 논문의 구성은 다음과 같다. 2장에서 하프팬케익 그래프와 비슷한 스타 그래프 부류 상호연결망의 특성에 대해 살펴보고, 3장에서 본 논문에서 제안하는 하프팬케익 그래프를 정의하고 기본성질 및 지름을 분석한다. 추가하여 라우팅 알고리즘을 제시하고 스타 그래프 부류의 연결망과 팬케익 그래프의 망

비용을 비교분석한다. 마지막으로 4장에서 본 논문의 결론을 맺는다.

## 2. 스타 그래프 부류의 상호연결망

스타 그래프 부류는  $n$ 개의 심볼을 이용하여 노드 주소를 표현하므로 노드 개수가  $n!$ 개 이고, 대략  $n$ 정도의 분지수를 갖도록 구성되어 있다. 이러한 스타 그래프 부류로는 스타 그래프[7],  $(n,k)$ -스타 그래프, 팬케익 그래프[13], 불완전 팬케익 그래프[14] 매크로스타 그래프[5], 버블정렬 그래프, 전위 그래프, Rotator 그래프 등이 제안되었다. 스타 그래프  $S_n$ 은  $n$ 개 정수를 이용하여 노드 주소를 표현하고, 노드 주소를 표현한 순열에서 첫 번째 심볼과  $k$ 번째 심볼이 서로 교환된 순열 사이에 에지가 있다( $2\leq k\leq n$ ). 따라서 노드 개수는  $n!$ 개 이고 분지수는  $n-1$ 이다. 스타 그래프는  $n-1$  가지의 방법으로  $n$ 개의 노드 중복 없는  $(n-1)$ -차원 스타 그래프  $S_{n-1}$ 로 분할 가능한 재귀적 구조를 갖고 있다. 스타 그래프  $S_n$ 은 이분 그래프이고 노드 및 에지 대칭이다. 지름은  $\lfloor 3(n-1)/2 \rfloor$  이고 최대고장 허용도 등 여러 가지 유용한 성질이 있음이 알려졌다. 스타 그래프 부류는 하이퍼큐브 부류와 비슷한 노드 개수를 가질 때 상대적으로 적은 분지수와 짧은 지름을 갖는 장점이 있지만, 노드 개수의 증가율이 급격하고 하이퍼큐브 부류와 임베딩이 어려운 단점이 있다.

버블정렬 그래프  $B_n$ 의 노드 개수는  $n!$ 이고, 에지 수는  $n!(n-1)/2$ 개로 구성된다. 버블정렬 그래프  $B_n$ 의 라우팅은  $n$ 개 정수를 오름차순으로 정렬하는 방식으로 수행하므로 지름은  $n(n-1)/2$ 이고, 에지를 중심으로 그래프를 분할 할 수 있는 계층적 연결망이다. 버블정렬 그래프  $B_n$ 은 노드 대칭적이고 에지 대칭적이며, 이분 그래프(bipartite graph)이고 해밀턴 싸이클을 포함하고 있다. 이와 같이 스타 그래프와 비슷한 장점들을 가지고 있지만 같은 노드수를 갖는 스타 그래프와 비교하여 분지수와 지름이 커 망 비용이 나쁜 단점이 있다.

팬케익 그래프  $P_n$ 의 각 노드 주소는 1부터  $n$ 까지의 서로 다른 심볼의 순열로 표현한다. 에지는 다음의 조건을 만족하는 노드  $v$ 와  $w$  사이에 존재한다.  $w$ 는  $v$ 의 순열을 첫 번째 심볼에서부터  $k$ 번째 심볼까지 역순으로 뒤집은 순열이다. 여기에서  $k$ 는 1보다 크고

$n$ 보다 적거나 같다. 따라서 노드 개수는  $n!$ 개이고, 분지수는  $n-1$ 이고, 전체 에지 개수는  $n!(n-1)/2$ 개다. 팬케익 그래프는 노드 대칭적이고 재귀적 구조를 갖고 있다. 팬케익 그래프  $P_n$ 은 해밀턴 싸이클을 포함하지만 4차원 이상의 팬케익 그래프는 길이가 홀수인 싸이클이 존재하기 때문에 이분 그래프가 아님이 알려져 있다. 팬케익 그래프에서 전위 합 문제, 정렬과 합병 알고리즘에 대한 연구 결과가 있다. 또한 지름, 방송, 병렬 라우팅과 정렬, 임베딩, 부하균등 문제 등 다양한 성질들이 발표되었다[13]. Fig. 1은 2차원, 3차원 팬케익 그래프이다. 4차원 팬케익 그래프는 Fig. 2의 4차원 하프팬케익 그래프와 동형이다.

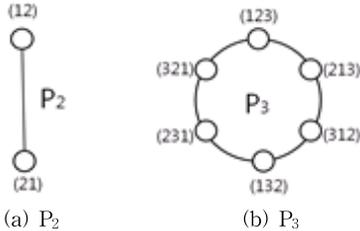


Fig. 1. Pancake Graph  $P_2, P_3$ .

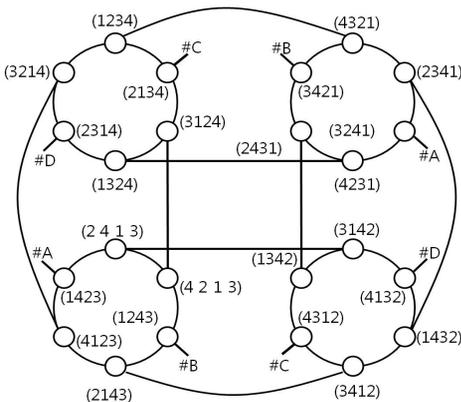


Fig. 2. 4-dimension Half Pancake  $HP_4$ .

### 3. 하프팬케익 그래프 정의와 라우팅 알고리즘

#### 3.1 하프팬케익 그래프 정의와 성질

하프팬케익 그래프  $HP_n$ 의 노드는  $\langle N \rangle = \{1, 2, 3, \dots, n\}$ 의 순열로 표현하고, 노드를 연결하는 에지는  $P_n$ 과  $P_k$ 로 나눌 수 있다( $2 \leq k \leq \lfloor n/2 \rfloor + 1$ ). 노드를 표현하는 순열에서 첫 번째 심볼부터  $n$ 번째 심볼까지

심볼의 순서를 뒤집어 생성된 순열과 원래의 순열을 연결하는 에지를 에지  $P_n$ 이라 하고, 첫 번째 심볼부터  $k$ 번째 심볼까지 순서를 뒤집어 생성된 순열과 원래의 순열을 연결하는 에지를 에지  $P_k, 2 \leq k \leq \lfloor n/2 \rfloor + 1$ 이라 한다. 하프팬케익 그래프  $HP_n$ 의 노드 개수는  $n!$ 개 이고, 분지수는  $\lfloor n/2 \rfloor + 1$ 이다. 본 논문에서는 노드의 주소를 순열로 표현할 수 있으므로 노드 주소와 순열을 동일한 의미로 사용한다. 또한  $HP_n$ 의 임의의 노드  $S$ 와 에지  $P_n$ 에 의해 인접한 노드를  $P_n(S)$ 라 하고, 에지  $P_k$ 에 의해 인접한 노드를  $P_k(S)$ 로 나타낸다. 예를 들어 임의의 노드  $S$ 의 순열을  $s_1 s_2 s_3 \dots s_{\lfloor n/2 \rfloor} s_{\lfloor n/2 \rfloor + 1} \dots s_{n-1} s_n$ 이라 하자. 노드  $S$ 와 에지  $P_n$ 에 인접한 노드  $P_n(S)$ 의 순열  $P_n(S) = s_n s_{n-1} \dots s_{\lfloor n/2 \rfloor + 1} s_{\lfloor n/2 \rfloor} \dots s_3 s_2 s_1$ 이고, 에지  $P_k$ 에 의해 인접한 노드  $P_k(S)$ 의 순열은  $s_k s_{k-1} s_{k-2} \dots s_2 s_1 s_{k+1} s_{k+2} \dots s_{n-1} s_n$ 이다( $2 \leq k \leq \lfloor n/2 \rfloor + 1$ ). Fig. 2는 4차원 하프팬케익 그래프이다.

예를 들어  $HP_5$  그래프에서 노드  $S=12345$ 에 인접한 노드의 순열은 3개의 에지에 인접한 노드  $P_5(S)=54321, P_3(S)=32145, P_2(S)=21345$ 이다. Table 1에서는 하프팬케익 그래프와 팬케익 그래프의 분지수를 비교하였다.  $n$ -차원 하프팬케익 그래프의 분지수는  $n$ -차원 팬케익 그래프의 분지수보다 대략 1/2 줄어든 값이다.

Table 1. Compare degree on Half Pancake and Pancake

$n$	4	5	6	7	8	9	...	$n$
Half Pancake	3	4	5	...	$\lfloor n/2 \rfloor + 1$			
Pancake	3	4	5	6	7	8	...	$n-1$

**성질 1.** 하프팬케익 그래프  $HP_n$ 에는  $(\lfloor n/2 \rfloor + 1)$ -차원 팬케익 그래프  $P_{\lfloor n/2 \rfloor + 1}$ 와 동형인 그래프가  $n \times (n-1) \times (n-2) \times \dots \times (\lfloor n/2 \rfloor + 2)$ 개 존재한다( $n \geq 5$ ).

**증명 :** 하프팬케익 그래프  $HP_n$ 의 임의의 노드  $S$ 의 순열을  $s_1 s_2 s_3 \dots s_{\lfloor n/2 \rfloor} s_{\lfloor n/2 \rfloor + 1} s_{\lfloor n/2 \rfloor + 2} \dots s_{n-1} s_n$ 이라 할 때, 에지는  $P_n$ 과  $P_k$ 로 구성되어 있다( $2 \leq k \leq \lfloor n/2 \rfloor + 1$ ).  $HP_n$ 의 에지  $P_k$ 는  $P_2$ 에서  $P_{\lfloor n/2 \rfloor + 1}$ 까지 구성되어 있으며  $(\lfloor n/2 \rfloor + 1)$ -차원 팬케익 그래프의 에지와 동일하다. 따라서 하프팬케익 그래프  $HP_n$ 에서 에지  $P_k$ 를 모두 포함하면서 생성

가능한 부분 그래프의 경우 수는 노드의 순열에서  $(\lfloor n/2 \rfloor + 2)$ 번째부터  $n$ 번째까지  $n - (\lfloor n/2 \rfloor + 1)$ 개 심볼  $s_{\lfloor n/2 \rfloor + 2}, s_{\lfloor n/2 \rfloor + 3}, s_{\lfloor n/2 \rfloor + 4}, \dots, s_n$ 에 대한 순열 개수이므로  $n \times (n-1) \times (n-2) \times \dots \times (\lfloor n/2 \rfloor + 2)$ 개 존재한다. □

**성질 2.** 하프팬케익 그래프  $HP_n$ 은 팬케익 그래프  $P_n$ 의 서브 그래프이다.

**증명 :** 하프팬케익 그래프와 팬케익 그래프 정의를 통해 하프팬케익 그래프가 팬케익 그래프의 서브 그래프임을 보인다.  $n$ -차원 하프팬케익 그래프와  $n$ -차원 팬케익 그래프의 노드는  $\langle N \rangle = \{1, 2, 3, \dots, n\}$ 에 대한 순열로 표현되므로 전체 노드 개수는  $n!$ 로 동일하다.  $n$ -차원 팬케익 그래프의 분지수는  $n-1$ 이고, 한 순열  $S = s_1 s_2 s_3 \dots s_k \dots s_{\lfloor n/2 \rfloor} \dots s_{n-1} s_n$ 의  $n-1$ 개 각각의 위치에서 순열의 심볼을 뒤집어서 생성된 순열과 원래 순열  $S$  사이에 에지가 있다.  $n$ -차원 하프팬케익 그래프의 분지수는  $\lfloor n/2 \rfloor + 1$ 이고, 에지  $P_k$ 는 한 순열  $s_1 s_2 s_3 \dots s_k \dots s_{\lfloor n/2 \rfloor} \dots s_{n-1} s_n$ 에서 심볼의 위치가  $2 \leq k \leq \lfloor n/2 \rfloor$ 인 위치에서 순열의 심볼 뒤집기를 수행하고, 에지  $P_n$ 는 모든 심볼의 뒤집기를 수행하는 에지이다. 따라서  $n$ -차원 하프팬케익 그래프  $HP_n$ 은  $n$ -차원 팬케익 그래프  $P_n$ 의 서브 그래프이다. □

### 3.2 라우팅 알고리즘과 지름

본 절에서는 심플 라우팅을 먼저 제안하고, 추가로 향상된 라우팅과 그에 따른 지름을 분석한다. 그 다음, 스타 그래프 부류의 상호연결망과 하프팬케익 그래프의 망비용을 비교한다. 라우팅은 메시지를 전송하고자 하는 두 노드 사이의 짧은 경로를 설정하는 것이다. 이 경로는 에지로 연결된 노드들의 순차적인 표시이다. 출발노드  $S = s_1 s_2 s_3 \dots s_i \dots s_{n-1} s_n$ 라고 하고 목적노드  $V = v_1 v_2 v_3 \dots v_i \dots v_{n-1} v_n$ 라고 한다. 노드  $S$ 와  $V$  순열을 이루는 심볼은 모두  $n$ 이하의 자연수로 이루어져 있으므로  $S$  심볼과  $V$  심볼은 원소는 서로 같고 단지 순서만 다를 수 있다. 라우팅은 출발노드  $S$  심볼의 순서를 목적노드  $V$  심볼의 순서와 일치하도록 연결망의 다른 노드들을 경유하는 경로이다. 심볼을 교환하는 방법은 하프팬케익 그래프의 에지 정의에 따른다.

임의의 노드  $W = w_1 w_2 w_3 \dots w_i \dots w_{n-1} w_n$ 의 순열에서 전위영역의 범위는  $1 \leq i < n - (\lfloor n/2 \rfloor + 1)$ 이고, 후위영역의 범위를  $\lfloor n/2 \rfloor + 1 < i \leq n$ 라고 하자. 단 전위영역과 후위영역에 있는 심볼의 개수는 항상 동일하다. 전위와 후위영역이 동일한 심볼 개수를 갖도록 하기 위해  $n$ 이 홀수인 경우 1개의 심볼이,  $n$ 이 짝수인 경우 2개의 심볼이 전위와 후위영역에서 제외된다. 라우팅에서 설명되는 심볼들의 교환은 모두 출발노드에서 이루어진다. 다음은 라우팅에서 사용될 심볼 교환에 대한 보조정리들이다. 임의의 노드  $W$ 에서 에지 시퀀스  $\langle P_4, P_2, P_7 \rangle$ 를 순서대로 경유한 경로는  $P_7(P_2(P_4(W)))$  표현한다. 간단히  $W \rightarrow P_4 \rightarrow P_2 \rightarrow P_7$ 로 나타낼 수 있고, 두 개의 표현은 같은 경로이다.

**[보조정리 1]**  $HP_n$ 의 임의의 노드  $W = w_1 w_2 w_3 \dots w_i \dots w_{n-1} w_n$ 에서 전위에 있는 심볼  $w_i$ 를 전위의 임의의 위치  $j$ 로 이동시키는 방법은 다음 증명과 같고 두 노드 사이의 경로 길이는 2이다.

**증명 :** 전위에 있는 심볼  $w_i$ 를 전위의 임의의 위치  $j$ 로 이동시키는 방법은 다음과 같다.

$$W \rightarrow P_i \rightarrow P_j \quad \square$$

**[보조정리 2]**  $HP_n$ 의 임의의 노드  $W = w_1 w_2 w_3 \dots w_i \dots w_{n-1} w_n$ 에서 전위의 심볼 1개를 후위로 이동시키고 후위의 심볼 1개를 전위로 이동시키는 방법은 다음과 같고 두 노드 사이의 경로 길이는 6이다.

**증명 :**  $HP_n$ 에서  $n$ 이 홀수인 경우와 짝수인 경우로 나눈다.

첫째,  $n =$  홀수일 경우

전위에 있는  $w_i$ 와 후위에 있는  $w_j$ 를 교환하기 위해 다음과 같이 자리바꿈한다.

$$W \rightarrow P_i \rightarrow P_{\lfloor n/2 \rfloor + 1} \rightarrow P_n \rightarrow P_{n-j} \rightarrow P_{\lfloor n/2 \rfloor + 1} \rightarrow P_n$$

둘째,  $n =$  짝수일 경우

전위에 있는  $w_i$ 와 후위에 있는  $w_j$ 를 교환하기 위해 다음과 같이 자리바꿈한다.

$$W \rightarrow P_i \rightarrow P_{\lfloor n/2 \rfloor + 1} \rightarrow P_n \rightarrow P_{n-j} \rightarrow P_{\lfloor n/2 \rfloor - 1} \rightarrow P_n \quad \square$$

[보조정리 3, 4]에서  $w_i = w_{i-1} + 1, w_{i+1} = w_i + 1$  이다.

**[보조정리 3]**  $HP_n$ 의 임의의 노드에서 내림차순으로 정렬된 심볼들  $w_{i+1} w_i w_{i-1}$ 이 맨 왼쪽에서부터 있

고, 심볼  $w_j \mid w_j = w_{j+1} + 1$ 가 전위에 있다면,  $w_j$ 를 포함하여 내림차순으로 정렬된 심볼들을 맨 왼쪽부터 배치하기 위한 경로의 길이는 2이다.

**증명 :** 라우팅 경로는 다음과 같고, 경로 길이는 2이다.

$$\begin{aligned} w_{i+1}w_iw_{i-1} \dots w_3w_2w_1 \dots w_j \dots w_{\lfloor n/2 \rfloor + 1} &\rightarrow \\ \dots w_1w_2w_3 \dots w_{i-1}w_iw_{i+1}w_j \dots w_{\lfloor n/2 \rfloor + 1} &\rightarrow \\ w_jw_{i+1}w_iw_{i-1} \dots w_3w_2w_1 \dots w_{\lfloor n/2 \rfloor + 1} &\square \end{aligned}$$

**[보조정리 4]**  $HP_n$ 의 임의의 노드 전위에서, 오름차순으로 정렬된 심볼들  $w_{i-1}w_iw_{i+1}$ 이 맨 왼쪽에서부터 있고, 심볼  $w_j \mid w_j = w_{j-1} - 1$ 가 전위에 있다면,  $w_j$ 를 포함하여 오름차순으로 정렬된 심볼들을 맨 왼쪽부터 배치하기 위한 경로의 길이는 2다.

**증명 :** 라우팅 경로는 다음과 같고, 경로 길이는 2이다.

$$\begin{aligned} w_{i-1}w_iw_{i+1} \dots w_3w_2w_1 \dots w_j \dots w_{\lfloor n/2 \rfloor + 1} &\rightarrow \\ \dots w_1w_2w_3 \dots w_{i+1}w_iw_{i-1}w_j \dots w_{\lfloor n/2 \rfloor + 1} &\rightarrow \\ w_jw_{i-1}w_iw_{i+1} \dots w_3w_2w_1 \dots w_{\lfloor n/2 \rfloor + 1} &\square \end{aligned}$$

### 3.2.1 심플 라우팅 알고리즘

심플 라우팅 알고리즘은 크게 두 단계로 진행된다. 첫 번째 단계에서는 목적노드의 후위 심볼과 같은 값을 가지는 출발노드의 심볼들을 전위나 후위에 모은다. 이 작업이 완료되면 목적노드의 전위 심볼들 역시 출발노드에서 후위 심볼들의 반대편에 모이게 된다. 이 단계를 그루핑(grouping)이라고 한다. 두 번째 단계에서는 출발노드의 전위에서 목적노드의 심

볼과 같은 순서로 출발노드의 심볼을 일치시킨다.

목적노드의 후위 심볼 집합을 TR(Target Rear), 전위 심볼 집합을 TF(Target front)라고 하고, 출발노드의 후위 심볼 집합을 SR(Source Rear), 전위 심볼 집합을 SF(source Front)라고 한다.

**[결정 1]**  $|TR \cap SR| \leq |TR \cap SF|$ 이면 TR을 전위로 TF를 후위로 그루핑하고  $|TR \cap SR| > |TR \cap SF|$ 라면 TR을 후위로 TF를 전위로 그루핑한다. 그루핑에서 전위에서 후위로 또는 후위에서 전위로 이동해야 할 심볼의 개수  $K = \min(|TR \cap SR|, |TR \cap SF|)$ 이다.

만약  $|TR \cap SR| \leq |TR \cap SF|$ 이면 후위에서 전위로 이동해야 할 심볼 집합  $R \rightarrow F = TR \cap SR$ 이고 전위에서 후위로 이동해야 할 심볼 집합  $F \rightarrow R = TF \cap SF$ 이다. 만약  $|TR \cap SR| > |TR \cap SF|$ 라면 전위에서 후위로 이동해야 할 심볼 집합  $F \rightarrow R = TR \cap SF$ 이고 후위에서 전위로 이동해야 할 심볼 집합  $R \rightarrow F = TF \cap SR$ 이다. 집합  $R \rightarrow F$ 에서  $R \rightarrow F[j]$ 는  $j$ 번째 심볼을 말한다.

**[보조정리 5]**  $HP_n$ 의 출발노드에서 TF와 TR을 전위와 후위로 그루핑하기 위해 교환해야 할 심볼의 수  $K$ 의 상한을  $E$ 라고 하면,  $E = \lceil (n-1)/4 \rceil$ 이다.

**증명 :**  $HP_n$ 의 출발노드에서  $|TF| = |TR| = n - \lfloor n/2 \rfloor + 1$ 이고,  $|TF| + |TR| = 2(n - \lfloor n/2 \rfloor + 1)$ 이다.  $|TF| + |TR| - \lfloor n/2 \rfloor + 1$ 는  $|R \rightarrow F| + |F \rightarrow R|$ 의 하한이고, 이것은  $2n - 3(\lfloor n/2 \rfloor + 1)$ 이다. 다시 말하면 전위

#### Simple Routing Algorithm

```

START
  FOR j = 1 to K
    R→F[j]와 F→R[j]를 [보조정리 2]에 따라 교환
  END-FOR

  IF(|TR ∩ SR| ≤ |TR ∩ SF|) then
    [보조정리 1]에 따라 출발노드의 전위에 있는 심볼들을 목적노드의 후위 심볼의 역순으로 배치 → Pn
    [보조정리 1]에 따라 출발노드의 전위에 있는 심볼들을 목적노드의 전위 심볼의 순으로 배치
  ELSE
    [보조정리 1]에 따라 출발노드의 전위에 있는 심볼들을 목적노드의 전위 심볼의 순으로 배치 → Pn
    [보조정리 1]에 따라 출발노드의 전위에 있는 심볼들을 목적노드의 후위 심볼의 역순으로 배치 → Pn
  END-IF
END
    
```

에 있는 TF와 TR 심볼의 개수의 합은 적어도  $2n - 3(\lfloor n/2 \rfloor + 1)$ 개다.  $K = (n - \lfloor n/2 \rfloor + 1) - \text{MAX}(|F \rightarrow R|, |R \rightarrow F|)$ 에서  $\text{MAX}(|F \rightarrow R|, |R \rightarrow F|)$ 가 적을수록 K는 커진다.  $\text{MAX}(|F \rightarrow R|, |R \rightarrow F|)$ 가 가장 적은 경우는  $|F \rightarrow R|$ 와  $|R \rightarrow F|$ 가 같을 때다.  $|F \rightarrow R| + |R \rightarrow F| =$  짝수일 경우 가장 적은  $\text{MAX}(|F \rightarrow R|, |R \rightarrow F|)$ 의 값은  $(|F \rightarrow R| + |R \rightarrow F|)/2$ 이고 홀수일 경우  $\lceil (|F \rightarrow R| + |R \rightarrow F|)/2 \rceil$ 이다. 따라서  $\text{MAX}(|F \rightarrow R|, |B|)$ 가 가장 적은 경우는  $\lceil (|F \rightarrow R| + |R \rightarrow F|)/2 \rceil$ 이다.

$HP_n$ 의 출발노드에서 TF와 TR을 전위와 후위로 그루핑하기 위해 교환해야 할 심볼의 수 K의 상한  $E = (n - \lfloor n/2 \rfloor + 1) - \lceil (|F \rightarrow R| + |R \rightarrow F|)/2 \rceil$ 이다. 여기에서  $\lceil (|F \rightarrow R| + |R \rightarrow F|)/2 \rceil$ 의 하한은  $|TF| + |TR| - \lfloor n/2 \rfloor + 1$  이므로

$$\begin{aligned} E &= 2n - 3(\lfloor n/2 \rfloor + 1) & (1) \\ &= (n - \lfloor n/2 \rfloor + 1) - \lceil (2n - 3(\lfloor n/2 \rfloor + 1))/2 \rceil \\ &= \lceil (n-1)/4 \rceil, \lceil (n-1)/4 \rceil \leq (n-1)/4 + 1 \\ E &\leq 0.25n + 0.75 \quad \square \end{aligned}$$

### 3.2.2 향상된 라우팅 알고리즘과 지름

향상된 라우팅 알고리즘이 심플 라우팅과 같은 점은 출발노드의 전위와 후위에 각각 TF와 TR을 그루핑을 한다는 것이다. 다른 점은 심플 라우팅은 그루핑 종료 후 그루핑 된 결과를 각각 처리하는데 반해 향상된 라우팅 방법은 그루핑과 동시에 전위와 후위에서 각각 라우팅도 진행한다. 향상된 라우팅 방법은 라우팅을 진행하면서 두 개의 묶음을 점차 키워나간다. 두 묶음의 크기가  $n-1$ 이 되면 라우팅이 완료된다. TF의 심볼들로 이루어진 묶음을 BoFV(bundle of front value)라고 하고 TR의 심볼들로 이루어진 묶음을 BoRV(bundle of rear value)라고 한다. 최초 BoFV는  $v_1$ 이고 나머지  $v_2, v_3, v_4 \dots v_{\lfloor n/2 \rfloor}$ 가 차례대로 추가된다. 최초 BoRV는  $v_n$ 이고 나머지  $v_{n-1}, v_{n-2}, v_{n-3} \dots v_{\lfloor n/2 \rfloor + 2}$ 가 차례대로 추가된다.  $\text{BoFV} = \{v_1, v_2, v_3\}$ 이라면  $\text{NEXT}(\text{BoFV}) = v_4$ 이다. 최초 순열에서 두 묶음이 분리되고 난후, 두 묶음은 항상 다른 위치(전위나 후위)에 있다.

향상된 라우팅 알고리즘의 개요는 다음과 같다. 먼저 [결정 1]에 따라 목적노드의 TR을 출발노드의 전위나 후위 중 어느 곳으로 그루핑할지 결정하고, 그 결정에 따라  $v_1$ 과  $v_n$ 을 전위의 맨 앞과 후위의

맨 뒤로 이동시킨다. 최초 BoFV는  $v_1$ 이고 최초 BoRV는  $v_n$ 이 된다. BoFV가 전위에 있다면  $R \rightarrow F$  심볼 중  $j$ 가 가장 큰  $v_j$ 를  $\lfloor n/2 \rfloor + 1$ 로 이동한다. 그 다음  $\text{NEXT}(\text{BoFV})$ 가 전위에 있는 동안 BoFV에  $\text{NEXT}(\text{BoFV})$ 를 추가한다. 더 이상  $\text{NEXT}(\text{BoFV})$ 가 전위에 없다면 에지  $P_n$ 을 적용한다. 전위에 BoRV가 있다면  $F \rightarrow R$  심볼 중  $j$ 가 가장 적은  $v_j$ 를 순열의  $\lfloor n/2 \rfloor + 1$  위치로 이동한다. 그 다음  $\text{NEXT}(\text{BoRV})$ 가 전위에 있는 동안 BoRV에  $\text{NEXT}(\text{BoRV})$ 를 추가한다. 더 이상  $\text{NEXT}(\text{BoRV})$ 가 전위에 없다면 순열에 에지  $P_n$ 을 적용한다. 정렬이 완료될 때까지 이 과정을 반복한다.

상호연결망의 지름은 임의의 두 노드 사이의 최단 경로들 중 최댓값이다. 여기에서 경로는 라우팅 알고리즘에 따른다. 지름을 계산하기 위해 향상된 라우팅 알고리즘의 몇 가지 특징을 살펴본다.

(특징 1) 출발노드의 전위에 BoFV와  $\text{NEXT}(\text{BoFV})$ 가 둘 다 있다면 [보조정리 3]에 따라 경로길이 2에  $\text{NEXT}(\text{BoFV})$ 가 추가된 새로운 BoFV가 만들어진다. 마찬가지로 BoRV와  $\text{NEXT}(\text{BoRV})$ 도 [보조정리 4]에 따라 경로길이 2에 새로운 BoRV가 만들어진다. 따라서 최악의 경우는 TF와 TR의 연속되는 심볼이 같은 위치(전위 혹은 후위)에 하나도 존재 하지 않는 경우이다.

(특징 2) 서로 교환해야 할 심볼의 개수  $K=0$ 일 경우 전위와 후위의 심볼들 간의 교환은 필요 없다. 단지 전위와 후위의 심볼들을 각자 정렬하면 된다. [보조정리 5]에 따라 최악의 경우는  $K=E$ 일 경우다.

(특징 3)  $K=E$ 일 경우, 전위에서 [보조정리 3]에 따라 경로길이 2에 BoFV에 추가되는 심볼은  $\lfloor n/2 \rfloor + 1 - E$ 개이다. 마찬가지로 [보조정리 4]에 따라 경로길이 2에 BoRV에 추가되는 심볼은  $n - \lfloor n/2 \rfloor + 1 - E$ 개이다.

**정리 1.** 하프팬케익 그래프  $HP_n$ 에서 향상된 라우팅 알고리즘에 의해 임의의 두 노드 사이의 라우팅 경로 길이를 T라고 하면  $T = 4K + 4(\lfloor n/2 \rfloor + 1) - 3$ 이고, 라우팅 경로 길이 T의 상한인 지름은  $3n + 4$  이하이다.

**증명 :** 증명의 편의를 위해 향상된 라우팅 알고리즘을 적용하여 설명한다. 최악의 경우 Block 1, 2는

Improved Routing Algorithm

```

START:
IF(전위에 BoFV가 있고 |BoFV| < ⌊n/2⌋ 이면) {
  IF(BoRV중 하나가 전위에 있다면)                               Section 1
    F→R 심볼 중 j가 가장 작은 vj를 ⌊n/2⌋+1에 이동
  WHILE(NEXT(BoFV)가 전위에 있다면)                             Section 2      Block 1
    NEXT(BoFV)를 BoFV에 추가
  P⌊n/2⌋ → Pn                                                 Section 3
  GO TO START
}
IF(전위에 BoRV가 있고 |BoRV| < (n - ⌊n/2⌋) - 1 이면) {
  IF(BoFV중 하나가 전위에 있다면)                               Section 1
    R→F 심볼 중 j가 가장 큰 vj를 ⌊n/2⌋+1에 이동
  WHILE(NEXT(BoRV)가 전위에 있다면)                             Section 2      Block 2
    NEXT(BoRV)를 BoRV에 추가
  P⌊n/2⌋ → Pn                                                 Section 3
  GO TO START
}
END:
    
```

K번 번갈아 수행되며, Block 1, 2가 수행될 때 Section 1, 3은 각각 1번씩 수행된다. 그러나 Section 2는 ⌊n/2⌋+1-K번 수행된다. 마지막 직전 Block이 수행될 때 Section 2는 수행되지 않는다. Block이 마지막으로 수행될 때 Section 1,3 모두 수행되지 않는다. 마지막의 경우 더 이상 넘겨줄 심볼이 없기 때문이고, 그 전 단계에서 넘어온 심볼은 그 자체로 정렬되어 있다. 마지막에 필요하다면 예지 P<sub>n</sub>이 한번 실행된다.

라우팅 경로 길이의 상한 T =

[K×(Section 1의 경로길이 2 + Section 3의 경로길이 2)+( ⌊n/2⌋+1-K) × Section 2의 경로길이 2] × 2(Block 1,2) - 6+1로 나타낼 수 있다.

$$\begin{aligned}
 T &= [K \times 4 + (\lfloor n/2 \rfloor + 1 - K) \times 2] \times 2 - 5 \\
 &= 8K + 4(\lfloor n/2 \rfloor + 1) - 4K - 3 \quad (2) \\
 &= 4K + 4(\lfloor n/2 \rfloor + 1) - 3
 \end{aligned}$$

최악의 경우 T =

$$\begin{aligned}
 &= 4K + 4(\lfloor n/2 \rfloor + 1) - 3, \text{ 여기서 최악의 경우 } K=E \text{이므로} \\
 &= 4E + 4(\lfloor n/2 \rfloor + 1) - 3, \text{ 여기서 } E \leq 0.25n + 0.75, \lfloor n/2 \rfloor + 1 \leq 0.5n + 1 \text{이므로} \\
 &\leq n + 3 + 2n + 4 - 3 \\
 &\leq 3n + 4 \quad \square
 \end{aligned}$$

Table 2에 스타 그래프 부류의 상호연결망에 대해 망비용을 비교하였다. 비교된 상호연결망들은 모두 노드 수가 n!개이다. 본 연구에서 제안한 하프팬케익

Table 2. Compare Network Cost among Star Graph class

Networks	Degree	Diameter	Network Cost
Star Graph $S_n$	$n-1$	$\lfloor 3(n-1)/2 \rfloor$	$O(1.5n^2)$
Transposition $T_n$	$n(n-1)/2$	$n-1$	$O(0.5n^3)$
Bubble Sort $BS_n$	$n-1$	$n(n-1)/2$	$O(0.5n^3)$
Pancake $P_n$	$n-1$	$(18/11)n$	$O(1.64n^2)$
Incomplete Pancake $P_nK$	$n-1$	$2n-3$	$O(2n^2)$
Half Pancake $HP_n$	$\lfloor n/2 \rfloor + 1$	$3n+4$	$O(1.5n^2)$

그래프는 전위 그래프와 버블정렬 그래프보다는 우수하고 팬케익 그래프의 망비용이 개선됨을 확인하였다.

하프팬케익 그래프의 망비용은 전위 그래프나 버블정렬 그래프의 망비용의 약  $1/n$ 이다. 따라서 망비용 측면에서 하프팬케익 그래프는 전위그래프나 버블정렬 그래프에 비해 약  $n$ 배 우수하다. 마지막으로 하프팬케익 그래프는 팬케익 그래프와 불완전 팬케익 그래프보다 분지수는  $1/2$ 정도 줄였음에도 불구하고 망비용은 각각 9%, 25%개선되었다. 따라서 하프팬케익 그래프는 팬케익 그래프보다 초기설치비용이 적게 들고 소프트웨어 운용비용도 적게 든다.

#### 4. 결 론

본 논문에서는 팬케익 그래프의 망비용을 개선하기 위해 팬케익 그래프의 분지수를 대략  $1/2$  줄인 새로운 하프팬케익 그래프를 제안하였다. 제안한 하프팬케익 그래프는 팬케익 그래프의 서브 그래프로 존재함을 보였으며, 차원이 낮은 팬케익 그래프는 하프팬케익 그래프의 부분 그래프로 존재함을 보였다. 또한 본 연구의 라우팅 알고리즘은 [15]에서 제시한 지름  $3.5n+4$ 를  $3n+4$ 로 15% 개선하였다. Table 2에 보인 것처럼 팬케익 그래프의 망비용은  $O(1.64n^2)$ 이고 하프팬케익 그래프의 망비용은  $O(1.5n^2)$ 으로 팬케익 그래프 보다 9%개선된 결과를 갖는다. 이러한 결과는 본 연구에서 제안한 하프팬케익 그래프가 병렬 처리를 위한 상호연결망으로 망비용 측면에서 스타 그래프 부류의 다른 연결망보다 효율적임을 알 수 있다. 향후에는 하프팬케익 그래프의 고장허용도 및 성질 분석, 방송 알고리즘, 임베딩 등에 관한 연구가 필요하다.

#### REFERENCE

[1] K. Hwang and F.A. Briggs, *Computer Architecture and Parallel Processing*, 4th Printing, McGraw-Hill International Editions, New York, 1988.

[2] A. Varma and C.S. Raghavendra, "Interconnection Networks for Multiprocessors and Multicomputers Theory and Practice," *IEEE*

*Computer Society Press*, pp. 8-18, 1994.

[3] T. Feng, "A Survey of Interconnection Networks," *IEEE Transactions on Computers*, Vol. 14, No. 12, pp. 12-27, 1981.

[4] J.H. Chang, "Cycle Extendability of Torus Sub-Graphs in the Enhanced Pyramid Network," *Journal of Korea Multimedia society*, Vol. 13, No. 8, pp. 1183-1193, 2010.

[5] C.H. Yeh and E.A. Varvarigos, "Macro-Star Networks: Efficient Low-Degree Alternatives to Star Graphs," *IEEE Transactions on Parallel and Distributed Systems*, Vol. 9, No. 10, pp. 987-1003, 1998.

[6] Q. Dong, J. Zhou, Y. Fu, X. Yang, "Embedding a Mesh of Trees in the Crossed Cube," *Information Processing Letters*, Vol. 112, No. 14-15, pp. 599-603, 2012.

[7] S.B. Akers and B. Krishnamurthy, "A group-Theoretic Model for Symmetric Interconnection Network," *IEEE Transactions on Computer*, Vol. 38, No. 4, pp. 555-565, 1989.

[8] Y. Saad and M.H. Schultz, "Topological Properties of Hypercubes," *IEEE Transactions on Computer*, Vol. 37, No. 7, pp. 867-872, 1988.

[9] J.S. Kim, E. Cheng, and H.O. Lee, "Embedding Hypercubes, Rings, and Odd Graphs into Hyper-Stars," *International Journal of Computer Mathematics*, Vol. 86, No. 5, pp. 771-778, 2013.

[10] H. Dweighter, *Amer. Math. Monthly*, Vol. 82, No. 1, pp. 1010, 1975.

[11] M.H. Heydari and I.H. Sudborough, "On Sorting by Prefix Reversals and the Diameter of Pancake Networks," *Proceedings of the First Heinz Nixdorf Symposium on Parallel Architectures and Their Efficient Use*, pp. 218-227, 1993.

[12] M.H. Heydari and I.H. Sudborough, "On the Diameter of the Pancake Network," *Journal of Algorithms*, Vol. 25, No. 1, pp. 67-94, 1997.

[13] J. Kim, *Design and Analysis for Half Pancake*, Master's Thesis of University of Sunchon,

2014.

- [14] E. Konstantinova, "On Some Structural Properties of Star and Pancake Graphs," *LNCS*, Vol. 7777, No. 2013, pp. 472-487, 2013.
- [15] K. Keiichi, "Routing Problems in Incomplete Pancake Graphs," *Proceedings of Seventh IEEE ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel Distributed Computing*, pp. 151-156, 2006.



김 주 봉

1989년 2월 전남대학교 사범대학  
수학교육과 졸업  
2014년 2월 순천대학교 교육대학  
원 컴퓨터교육과 졸업



서 정 현

2008년 8월 순천대학교 컴퓨터과  
학과 박사  
2009년~현재 순천대학교 과학교  
육연구소  
관심분야: 그래프이론, 알고리즘



이 형 옥

1999년 2월 전남대학교 전산학과  
박사  
2002년 3월~현재 순천대학교 컴  
퓨터교육과 교수  
관심분야: 그래프이론, 알고리즘