

멀티컴퓨팅 시스템을 위한 피터슨-토러스(PT) 네트워크

(Petersen-Torus(PT) Network for Multicomputing System)

서 정 현[†] 이 형 옥^{††} 장 문 석^{†††}
(Junghyun Seo) (Hyeongok Lee) (Moonsuk Jang)

요 약 본 논문은 피터슨 그래프를 기본으로 만든 새로운 상호연결망 피터슨-토러스(PT) 네트워크를 제안한다. PT 네트워크는 동일한 노드수를 갖는 honeycomb 토러스보다 짧은 지름을 갖고 망 비용이 개선된 연결망이다. 본 논문에서는 PT 네트워크의 최적 라우팅 알고리즘과 해밀톤 사이클 알고리즘을 제안하고, 지름, 망 비용 그리고 bisection width를 분석하였다.

키워드 : 병렬 컴퓨터, 상호연결망, 피터슨-토러스, 피터슨 그래프, 라우팅

Abstract We propose and analyze a new interconnection network, called petersen-torus(PT) network based on well-known petersen graph. PT network has a smaller diameter and a smaller network cost than honeycomb torus with same number of nodes. In this paper, we propose optimal routing algorithm and hamiltonian cycle algorithm. We derive diameter, network cost and bisection width.

Key words : parallel computer, interconnection network, petersen-torus, petersen graph, routing

1. 서 론

대규모 병렬처리 시스템에서 공학과 과학 분야의 다양한 응용 알고리즘을 효율적으로 수용하고, 제 성능을 발휘하기 위해서는 병렬처리 시스템을 구성하는 프로세서들의 연결 구조를 그래프형태로 표현한 상호연결망(interconnection network)의 역할이 매우 중요하다. 지금까지 병렬처리 시스템을 위한 상호연결망으로 다양한 위상들이 발표되었으며, 대표적인 위상으로는 토러스, 하이퍼큐브, 스타그래프 등이 있다. 상호연결망은 크게

동적연결망과 정적연결망으로 나눈다. 정적연결망은 두 프로세서 사이의 통신 링크가 수동적이므로 다른 프로세서와 다시 연결될 수 없으며, 동적연결망은 스위칭 요소에 의해 통신 링크가 다른 프로세서와 연결될 수 있다. 동적 연결망은 single stage, multi stage, 크로스바 그리고 cellular array로 분류할 수 있다. single stage에 recirculating shuffle-exchange 연결망이 있고, multi stage에 data manipulator, flip, indirect binary n-큐브, omega, clos 그리고 cantor가 있다. cellular array에는 near-neighbor connection이 있다[1]. 정적 연결망은 망을 구성하는 노드와 에지의 구성에 따라 메쉬 부류, 하이퍼큐브 부류 그리고 스타그래프 부류로 나눌 수 있다. 메쉬 부류는 토러스[2], 허니컴브 메쉬(honeycomb mesh)[3], 다이아고날 메쉬(diagonal mesh)[2], 헥사고날 메쉬(hexagonal mesh)[4] 등이 있으며, 하이퍼큐브 부류는 하이퍼큐브[5], folded 하이퍼큐브[6], multiply-twisted-cube[7], 재귀원형군[8] 등이 있으며, 스타그래프 부류는 스타그래프[9], 매크로-스타(macro-star)[10], 전위(transposition) 그래프[11], 행렬-스타(matrix-star) 그래프[12] 등이 있다.

메쉬 구조는 분지수가 상수이므로 연결망의 확장이 쉽고, VLSI 회로설계 분야에서 많이 이용되는 구조로 현재까지 널리 이용되고 있으며 다양한 시스템으로 상용화되었다. 낮은 차원의 메쉬는 설계하기 쉽고 알고리

· 본 연구는 2005학년도 순천대학교 학술연구비 공모 과제로 연구되었음

† 학생회원 : 순천대학교 컴퓨터공학
jhseo@sunchon.ac.kr

†† 정 회원 : 순천대학교 컴퓨터교육 교수
oklee@sunchon.ac.kr
(Corresponding author)

††† 종신회원 : 순천대학교 컴퓨터공학 교수
jang@sunchon.ac.kr

논문접수 : 2007년 11월 16일

심사완료 : 2008년 2월 25일

Copyright©2008 한국정보과학회 : 개인 목적이나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 시스템 및 이론 제35권 제6호(2008.6)

즘 관점에서도 매우 유용하므로 병렬처리 컴퓨터의 연결망으로 많이 쓰인다. 높은 차원의 메쉬 일수록 지름이 작아지고 bisection width가 커지며 여러 가지 병렬 알고리즘을 빨리 수행할 수 있지만 비용이 많이 드는 단점이 있다[13]. 허니컴브 메쉬는 같은 노드수를 갖는 다른 종류의 메쉬 보다 망 비용이 우수하고 이동 전화 기지국 위치문제, 컴퓨터그래픽, 이미지 처리 등의 응용분야를 갖는 연결망으로 제안되었다[3].

상호연결망의 성능을 평가하는 척도는 분지수, 대칭성, 확장성, 연결도, 지름, 망 비용, 평균거리, 고장 지름, 고장 허용 알고리즘 그리고 임베딩 등이 있다. 망 비용은 분지수 × 지름으로 정의되며, 두 요소는 역상관계(trade-off)관계에 있기 때문에 망 비용을 줄이는 것은 쉽지 않다. 그래프에서 분지수 d 와 지름 k 가 주어졌을 때 최대 노드수를 가지는 그래프를 찾는 것이 (d,k) 그래프 문제이다. 분지수 3, 지름 2를 갖는 $(3,2)$ 그래프 문제에서 최대 노드수는 10개이고, 그 그래프는 피터슨(petersen) 그래프임이 알려졌다[14].

본 논문에서는 (d,k) 그래프 문제의 최적으로 알려진 피터슨 그래프를 기본모듈로 하고, 분지수가 상수로 고정된 메쉬부류의 PT 네트워크를 제안한다. PT 네트워크는 기본모듈을 격자구조의 격자점에 배치하고, 기본모듈을 주변 기본모듈과 완전 그래프 형태로 연결한 구조이다. 본 논문의 연구결과로 PT 네트워크와 동일한 노드수를 갖는 허니컴브 메쉬 및 허니컴브 토러스, 다이아고날 메쉬 및, 토러스 그리고 헥사고날 토러스와 비교하여 개선된 망 비용, 지름 그리고 bisection width를 표 2에 보였다. 특히 동일한 노드 수(N)를 가질 때 허니컴브 토러스의 지름은 $0.81\sqrt{N}$, PT 네트워크의 지름은 $0.47\sqrt{N}$ 이고, 허니컴브 토러스의 망 비용은 $2.45\sqrt{N}$, PT 네트워크의 망 비용은 $1.9\sqrt{N}$ 으로, 널리 알려진 허니컴브 토러스보다 PT 네트워크가 지름과 망 비용에서 개선된 값을 가진다. 본 논문의 구성은 2장에서 (d,k) 그래프 문제, 피터슨 그래프의 라우팅 그리고 피터슨 그래프의 해밀톤 성질을 보였다. 3장에서 PT 네트워크를 제안하고, 단순 라우팅 알고리즘, 최적 라우팅 알고리즘, 해밀톤 사이클 그리고 기본적인 몇 가지 위상 속성을 보였다. 또한 분지수가 고정인 비슷한 부류의 연결망과 망 비용 및 bisection width를 비교하였다. 마지막으로 결론을 맺는다.

2. 관련연구

그래프 이론에서 분지수는 임의의 노드와 결합된(incident) 에지 수다. 거리(distance)는 임의의 두 노드 사이의 가장 짧은 경로길이를 말하고, 연결망의 지름(diameter)은 임의의 두 노드 사이의 거리 중 가장 큰

값이다. 주어진 노드수에서 분지수(degree)와 지름을 최소화하는 그래프를 찾는 것은 효과적인 마이크로프로세서 네트워크를 찾는 것과 같다. 역으로 주어진 분지수 d 와 지름 k 에서 최대 노드수를 가지는 무방향(undirected) 그래프를 찾는 것이 (d,k) 그래프 문제이고, 이와 같은 그래프를 (d,k) 그래프라고 한다. (d,k) 그래프 문제는 최초로 Elspas에 의해서 정립되었다[15]. 링 네트워크는 분지수가 2이고 지름이 k 일 때 가질 수 있는 최대 노드수는 $2k+1$ 이다. 하이퍼큐브 네트워크는 분지수와 지름이 같으며 그 값이 k 일 때 가질 수 있는 최대 노드수는 2^k 이다. (d,k) 그래프 문제에서 d, k 가 일정할 때 노드수의 상한을 정리하기 위해 트리를 이용한 무어그래프(moore graph)가 설계되었다. 분지수가 d 일 때, 근 노드는 d 개의 부 트리를 가지고, d 개의 부 트리는 각각 $d-1$ 개의 부 트리를 가진다. 이와 같은 방법으로 $k-1$ 레벨 트리를 완성하면 무어그래프가 된다. 무어그래프의 노드수는 $1+d+d(d-1)+d(d-1)^2+ \dots + d(d-1)^{k-1} = \frac{d(d-1)^k - 2}{d-2}$ ($d>2$)이다. 이것이 무어그래프의 상한이며, 무어바운드(moore bound)라고 한다. d, k 가 일정할 때 무어그래프보다 많은 노드수를 가진 그래프는 없다. 무어그래프를 만드는 것은 매우 어려워서 지름이 2일 때 단지 $(3,2)$ $(7,2)$ $(57,2)$ 에서 만들어졌다[16-18]. 그림 1은 $(3,2)$ 무어그래프이다. 이것은 그림 2의 피터슨 그래프와 동형이며, $d=3, k=2$ 에서 가장 많은 노드수를 가진 그래프이다. (d,k) 그래프 문제에서 무어그래프 방식이 아닌 다른 방법으로 많은 그래프가 설계되었다.

표 1은 주어진 분지수와 지름에서 이미 설계된 (d,k) 그래프가 가지는 가장 많은 노드수를 나타낸다. '*' 표시는 그 수를 초과하는 노드수를 가지는 그래프를 설계하는 것이 불가능하다고 증명된 것들이다[14]. $d=3, k=2$ 일 때 그래프가 가질 수 있는 최대 노드수는 10이다 [14,15,19-22].

피터슨 그래프는 10개의 노드를 가지고 있다. 그림 2의 (a)처럼 바깥쪽 5개의 노드로 구성된 오각형 모양의 사이클이 있고, 안쪽 5개 노드로 구성된 별 모양의 사이클이 있다. 바깥쪽 사이클의 노드들은 안쪽 사이클의 서로 다른 노드로 연결되어 있다. 피터슨 그래프 노드주소

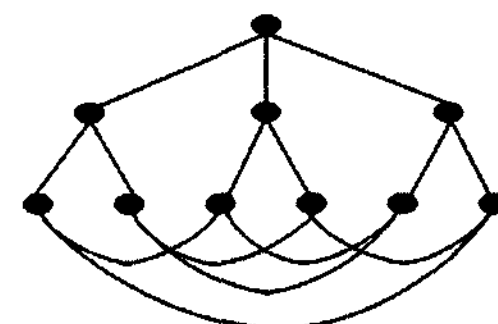


그림 1 (3,2) 무어그래프

표 1 (d,k) 그래프에서 설계된 최대 노드수

지름(k) 분지수(d)	1	2	3	4	5	6	7
2	3*	5*	7*	9*	11*	13*	15*
3	4*	10*	20*	30	56	72	120
4	5*	15*	35	48	80	114	212
5	6*	24*	42	126	130	232	530
6	7*	31	62	108	462	729	2187
7	8*	50*	80	150	378	1716	2144
8	9*	57	114	256	1280	4096	16384

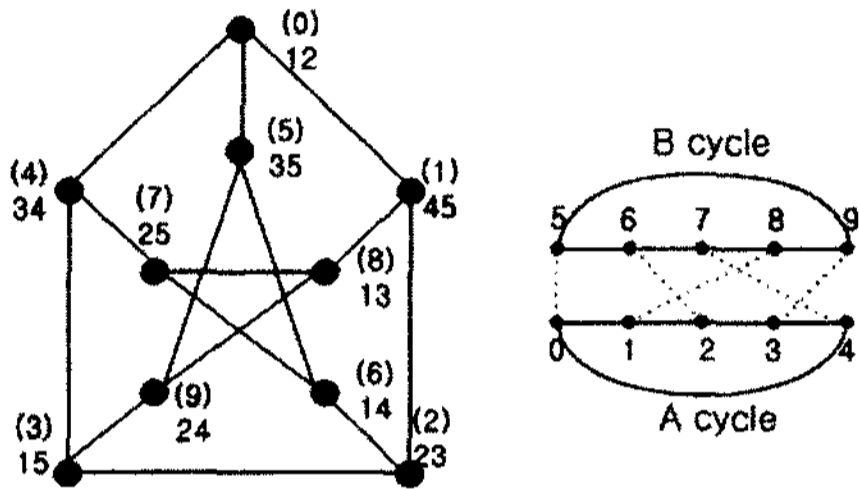


그림 2 (a) 피터슨 그래프 (b) A, B사이클의 피터슨 그래프

할당방법은 여러 가지가 있는데 그림 2의 (a)와 같은 방법은 [1,23]에 있다. 피터슨 그래프는 정규 그래프이고, 노드(에지) 대칭 그래프이다. 분지수 3, 지름 2, 연결도 3 그리고 girth는 5이다.

피터슨 그래프 $P=(V_p, E_p)$ 이다. 노드 $V_p=(xy \mid x,y \in \{1,2,3,4,5\}, x < y)$ 이다. 에지 $E_p=((xy, x'y') \mid x,y \in \{1,2,3,4,5\}, x < y, x'y' \in \{\{1,2,3,4,5\} - \{x,y\}\}, x' < y')$ 이다.

피터슨 그래프에서 임의의 노드 $U=u_1u_2$ 를 출발노드라고 하고, 임의의 노드 $V=v_1v_2$ 를 목적노드라 하자. 심벌 $u_1, u_2, v_1, v_2 \in \{1,2,3,4,5\}$ 이다. U 에서 V 로의 라우팅 알고리즘은 다음과 같다. U 와 V 는 다른 노드라고 가정한다.

보조정리 1

경우 1. $\{u_1, u_2\} \cap \{v_1, v_2\} = \emptyset$ 이면 U 와 V 는 인접(adjacent)한다.

경우 2. $\{u_1, u_2\} \cap \{v_1, v_2\} \neq \emptyset$ 이면 U 로부터 $\{1,2,3,4,5\} - (\{u_1, u_2\} \cup \{v_1, v_2\})$ 로 이루어진 노드를 거쳐 V 에 도달한다. 라우팅에 대한 자세한 설명은 [1,9,21-23]에 있다.

그래프의 해밀턴 경로는 그 그래프의 모든 노드를 오직 한번만 지나는 경로를 말하며, 시작노드와 도착노드가 같은 해밀턴 경로를 해밀턴 사이클이라고 말한다. 해밀턴 사이클을 가지는 그래프를 해밀턴(hamiltonian) 그래프라 한다. 피터슨 그래프는 해밀턴 그래프는 아니지만 해밀턴 경로는 존재한다. 그림 2의 (b)는 노드 0, 1, 2, 3, 4로 구성된 A 사이클과, 노드 5, 6, 7, 8, 9로 구성된 B 사이클이 있다. 임의의 노드에서 출발하여 자신이 속한 사이클의 모든 노드를 순회하고 다른 사이클의 모든 노드를 순회하면 해밀턴 경로가 만들어진다. 노드

0에서 출발하여 A 사이클의 모든 노드를 순회하고 도착할 수 있는 노드는 1과 4이다. 노드 1, 4로부터 B 사이클의 모든 노드를 순회하고 도착할 수 있는 노드는 6, 7, 8, 9이다. 노드 0, 1, 8, 9, 5를 A 사이클로 노드 2, 3, 4, 7, 6을 B 사이클로 다시 그리기를 하면, 노드 0에서 출발하여 A 사이클의 모든 노드를 순회하고 도착할 수 있는 노드는 1과 5이다. 노드 1, 5로부터 B 사이클의 모든 노드를 순회하고 도착할 수 있는 노드는 2, 3, 6, 7이다. 따라서 노드 0으로부터 출발하여 모든 노드를 순회하고 도착할 수 있는 노드는 2, 3, 6, 7, 8, 9이다. 피터슨 그래프는 노드(에지) 대칭이므로 모든 노드에서 노드 자신과 자기노드와 인접한 노드를 제외한 모든 노드로 해밀턴 경로가 있다.

보조정리 2. 피터슨 그래프는 임의의 노드에서 출발하여 출발노드와 출발노드에 인접한 3개의 노드를 제외한 모든 노드로 도착하는 해밀턴 경로가 있다. 이것에 대한 증명은 [19]에 있다.

3. 피터슨-토러스 PT(m,n) 네트워크

3.1 피터슨-토러스 PT(m,n) 네트워크의 정의 및 성질

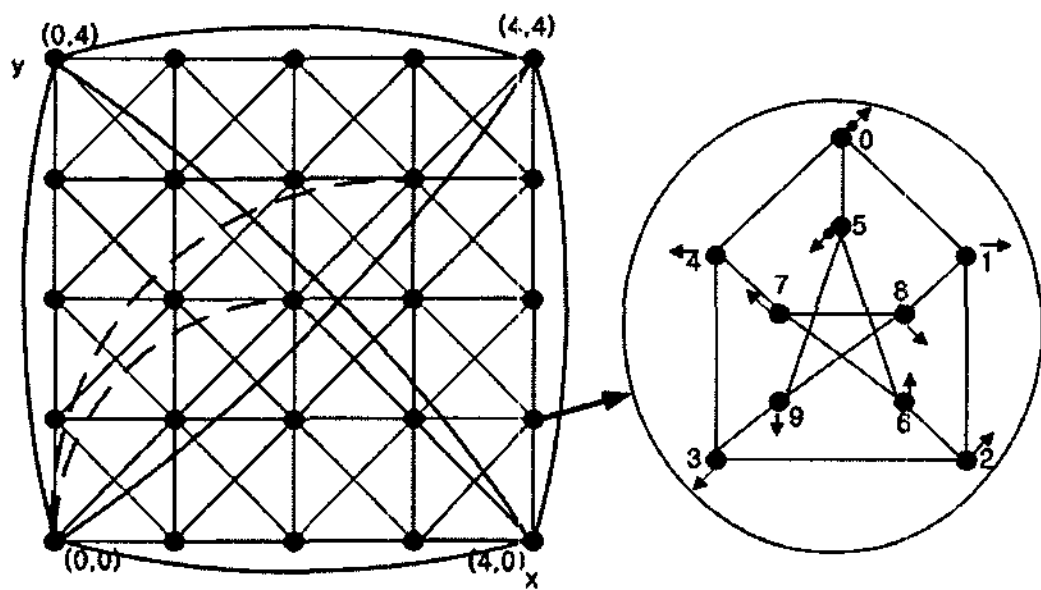
$PT(m,n)(m,n \geq 2)$ 은 피터슨 그래프(그림 3의 (b))를 기본모듈로 하고, 기본모듈 $m(x$ 축) $\times n(y$ 축)개를 격자점에 배치하고 에지 정의에 의해 연결한다. 피터슨-토러스 네트워크 $PT(m,n)=(V_{pt}, E_{pt})$ 이다. 본 논문에서는 $PT(m,n)$ 를 그림 3의 (a)와 같이 2차원 그래프에 사상시켜 설명한다. 단위 피터슨 그래프를 기본모듈이라고 하며, 기본모듈은 x 와 y 의 교차점에 있다. 기본모듈의 주소는 (x,y) 로 나타내고 노드 주소는 (x,y,p) 로 나타낸다. "노드 (x,y,p) 는 기본모듈 (x,y) 에 속한다"라고 말한다. x 는 기본모듈의 x 축의 좌표, y 는 기본모듈의 y 축의 좌표, p 는 기본모듈인 피터슨 그래프에 있는 노드의 주소이다. 피터슨-토러스 $PT(m,n)$ 의 노드정의는 다음과 같다.

$$V_{pt} = \{(x,y,p), 0 \leq x < m, 0 \leq y < n, 0 \leq p \leq 9\}$$

$PT(m,n)$ 의 에지는 다음과 같이 내부에지와 외부에지로 나눈다. 같은 기본모듈에 속한 노드들을 연결하는 에지를 내부에지라고 하고, 내부에지는 피터슨 그래프의 에지를 그대로 사용한다. 서로 다른 기본모듈에 있는 노드들을 연결하는 에지를 외부에지라고 하고 다음과 같이 정의한다. 아래 에지를 나타내는 수식에서 심벌 '/'는 나머지 연산자이다. ① 세로에지는 $((x,y,6), (x,(y+1)/n,9))$ 과 $((x,y,9), (x,(y-1+n)/n,6))$ 이다. ② 가로에지는 $((x,y,1), ((x+1)/m,y,4))$ 과 $((x,y,4), ((x-1+m)/m,y,1))$ 이다. ③ 사선에지는 $((x,y,2), ((x+1)/m,(y+1)/n,3))$ 과 $((x,y,3), ((x-1+m)/m,(y-1+n)/n,2))$ 이다. ④ 역 사선에지는 $((x,y,7), ((x-1+m)/m,(y+1)/n,8))$ 과 $((x,y,8), ((x+1)/m,(y-1+n)/n,7))$ 이

다. ⑤ 지름에지는 $((x,y,0), ((x + \lfloor \frac{m}{2} \rfloor)/m, (y - \lfloor \frac{n}{2} \rfloor)/n, 5))$ 과 $((x,y,0), ((x - \lfloor \frac{m}{2} \rfloor + m)/m, (y - \lfloor \frac{n}{2} \rfloor + n)/n, 5))$ 이다.

그림 3의 (a)는 피터슨-토러스 $PT(5,5)$ 에서 기본모듈을 격자점으로 표현하였다. 기본모듈은 이웃한 기본모듈들과 완전 그래프 형태로 연결되어 있다. 가장자리를 제외한 모든 기본모듈은 지름에지를 제외한 에지들이 그려져 있으며, 기본모듈 (0,0)의 지름에지만 짧은 파선(짧은 선이 일정한 규칙으로 반복되는)으로 그려져 있다. 가장자리의 기본모듈들은 랩어라운드(wraparound)에지가 생략되어 있으나 4개의 꼭지점에 있는 기본모듈은 몇 개의 랩어라운드 에지가 짧은 실선으로 그려져 있다. 피터슨-토러스 $PT(m,n)$ 의 기본모듈인 피터슨 그래프(그림 3의 (b))를 보면, 노드 1, 4는 가로에지, 노드 6, 9는 세로에지, 노드 2, 3은 사선에지, 노드 7, 8은 역 사선에지, 노드 0, 5는 지름에지와 결합된 노드들이다. $PT(m,n)$ 은 노드수 $10mn$, 에지 수 $20mn$, 연결도 4 그리고 분지수(degree) 4를 가지는 정규 그래프(regular graph)이다. 추가로 노드 U 로부터 노드 V 까지 경로가 있다고 가정하면, 그 경로에 사용된 외부에지의 수는 외부경로길이라고 하고, 내부에지의 수는 내부경로길이라고 한다. 이것은 논문 전반에 걸쳐 사용된다.



(a) $PT(5,5)$ (b) 피터슨 그래프
그림 3 피터슨-토러스 $PT(5,5)$

3.2 단순 라우팅(simple routing) 알고리즘

단순 라우팅 알고리즘은 먼저 출발노드가 있는 기본모듈(출발기본모듈)에서 목적노드가 있는 기본모듈(목적기본모듈)로 라우팅 하고, 목적기본모듈 안에서 목적노드로 라우팅 한다. 단순 라우팅 알고리즘에서는 지름에지를 고려하지 않는다. 또한 출발기본모듈과 목적기본모듈의 노드 조합도 고려하지 않는다. 같은 기본모듈에 있는 노드간의 라우팅은 내부 라우팅, 다른 기본모듈에 있는 노드간의 라우팅은 외부 라우팅이라고 한다. 단순 라우팅의 기본전략은 x축과 y축으로 각각 1만큼 떨어져

있는 기본모듈로 라우팅 할 때 가로에지, 세로에지를 이용하는 대신 사선에지를 이용한다. 내부 라우팅은 보조 정리 1에 따른다. 단순 라우팅 알고리즘을 요약하면 다음과 같다.

Step 1. 출발기본모듈의 내부 라우팅

출발노드에서 사선(역 사선)에지에 결합된 노드나 가로(세로)에지에 결합된 노드로 라우팅 한다.

Step 2. 사선(역 사선)에지를 이용한 외부 라우팅

출발기본모듈과 목적기본모듈의 x축 거리와 y축 거리 중에 적은 값만큼 사선(역 사선)에지를 이용하여 라우팅 한다.

Step 3. 가로(세로)에지를 이용한 외부 라우팅

가로(세로)에지를 이용하여 남은 x축(y축) 거리만큼 라우팅 한다.

Step 4. 목적기본모듈의 내부 라우팅

사선(역 사선)에지에 결합된 노드나 가로(세로)에지에 결합된 노드로부터 목적노드로 라우팅 한다.

다음은 단순 라우팅 알고리즘이다. $U(x_1, y_1, p_1)$ 를 출발노드, $V(x_2, y_2, p_2)$ 를 목적노드, $T(x', y', p')$ 를 중간 경유 임의의 노드라고 한다. dx 는 x좌표가 증가하는 방향으로 x축 거리, dy 는 y좌표가 증가하는 방향으로 y축 거리, dm 은 x축과 y축을 동시에 움직이는 거리, dr 은 dm 만큼 이동한 후 x축이나 y축으로 남은 거리다. 내부 라우팅은 \rightarrow , 외부 라우팅은 \Rightarrow 로 표시한다.

$dx = (x_2 - x_1 + m) / m$ 이고 $dy = (y_2 - y_1 + n) / n$ 이다. $dm = \min(dx, dy)$ 이고 $dr = dx - dy$ 이다. 단순 라우팅 알고리즘은 dx 와 dy 에 따라서 아래와 같이 4가지 경우로 나눈다.

경우 1. $dx \leq \lfloor \frac{m}{2} \rfloor, dy \leq \lfloor \frac{n}{2} \rfloor$

Begin

$j := dm, k := dr$

While($j \neq 0$) Begin //사선에지를 이용한 외부 라우팅

if($j = dm$) $(x_1, y_1, p_1) \rightarrow (x_1, y_1, 2) \Rightarrow (x_1 + 1, y_1 + 1, 3)$ [Step 1]

else $(x', y', 3) \rightarrow (x', y', 2) \Rightarrow (x' + 1, y' + 1, 3)$ [Step 2]

$j := j - 1$

End while

While($k > 0$) Begin //가로에지를 이용한 외부 라우팅

if($k = dr$)

if ($j = dm$) $(x_1, y_1, p_1) \rightarrow (x_1, y_1, 1) \Rightarrow (x_1 + 1, y_1, 4)$ [Step 1]

else $(x', y', 3) \rightarrow (x', y', 1) \Rightarrow (x' + 1, y', 4)$ [Step 3]

else $(x', y', 4) \rightarrow (x', y', 1) \Rightarrow (x' + 1, y', 4)$ [Step 3]

$k := k - 1$

End while

While(k<0) Begin //세로에지를 이용한 외부 라우팅
 if(k=dr)
 if (j=dm) $(x_1, y_1, p_1) \rightarrow (x_1, y_1, 6) \Rightarrow (x_1, y_1 + 1, 9)$ [Step 1]
 else $(x', y', 3) \rightarrow (x', y', 6) \Rightarrow (x', y' + 1, 9)$ [Step 3]
 else $(x', y', 9) \rightarrow (x', y', 6) \Rightarrow (x', y' + 1, 9)$ [Step 3]
 k:=k+1
End while

 if(dr=0) $(x_2, y_2, 3) \rightarrow (x_2, y_2, p_2)$ [Step 4]
 if(dr>0) $(x_2, y_2, 4) \rightarrow (x_2, y_2, p_2)$ [Step 4]
 if(dr<0) $(x_2, y_2, 9) \rightarrow (x_2, y_2, p_2)$ [Step 4]
End

경우 2. $dx \leq \lfloor \frac{m}{2} \rfloor, dy > \lfloor \frac{n}{2} \rfloor$ 는 경우 1에서 사선예지 $(x', y', 2) \Rightarrow (x'+1, y'+1, 3)$ 를 역 사선예지 $(x', y', 8) \Rightarrow (x'+1, y'-1, 7)$ 로 변경하고, 세로예지 $(x', y', 6) \Rightarrow (x', y'+1, 9)$ 를 $(x', y', 9) \Rightarrow (x', y'-1, 6)$ 로 변경한다. 경우 3 $dx > \lfloor \frac{m}{2} \rfloor, dy \leq \lfloor \frac{n}{2} \rfloor$ 은 경우 1에서 사선예지 $(x', y', 2) \Rightarrow (x'+1, y'+1, 3)$ 를 역 사선예지 $(x', y', 7) \Rightarrow (x'-1, y'+1, 8)$ 로 변경하고, 가로예지 $(x', y', 1) \Rightarrow (x'+1, y', 4)$ 를 $(x', y', 4) \Rightarrow (x'-1, y', 1)$ 로 변경한다. 경우 4 $dx > \lfloor \frac{m}{2} \rfloor, dy > \lfloor \frac{n}{2} \rfloor$ 는 경우 1에서 사선예지 $(x', y', 2) \Rightarrow (x'+1, y'+1, 3)$ 를 사선예지 $(x', y', 3) \Rightarrow (x'-1, y'-1, 2)$ 로 변경하고, 세로예지 $(x', y', 6) \Rightarrow (x', y'+1, 9)$ 를 $(x', y', 9) \Rightarrow (x', y'-1, 6)$ 로 변경하고, 가로예지 $(x', y', 1) \Rightarrow (x'+1, y', 4)$ 를 $(x', y', 4) \Rightarrow (x'-1, y', 1)$ 로 변경한다. 경우 2, 경우 3, 경우 4에서 사선예지(역 사선예지)와 가로(세로)예지 변화에 따른 출발기본 모듈과 목적기본 모듈에서의 내부 라우팅 변경은 생략하였다. 예를 들어 PT(8,8)에서 $U=(0,0,6), V=(4,3,7)$ 인 경우 라우팅 경로는 다음과 같다. $(0,0,6) \rightarrow (0,0,2) \Rightarrow (1,1,3) \rightarrow (1,1,2) \Rightarrow (2,2,3) \rightarrow (2,2,2) \Rightarrow (3,3,3) \rightarrow (3,3,2) \rightarrow (3,3,1) \Rightarrow (4,3,4) \rightarrow (4,3,7)$

3.3 최적 라우팅(optimal routing) 알고리즘 및 지름

최적 라우팅 알고리즘은 단순 라우팅 알고리즘을 그대로 사용하고, 추가로 라우팅에서 사용되는 외부예지의 수(외부 경로길이)를 줄이기 위해 단순 라우팅 알고리즘에서 고려되지 않은 지름예지를 이용한다. 지름예지를 이용함으로써 단순 라우팅 알고리즘에서 외부 경로길이 가 가장 긴 노드들 사이의 외부 경로길이를 반으로 줄였다. 또한 출발기본 모듈에서 노드와 목적기본 모듈에서 노드 조합을 고려하였다. 이것은 출발기본 모듈과 목적기본 모듈에서 라우팅에 사용되는 내부예지의 수(내부 경로 길이)를 최소화 한다. 최적 라우팅 알고리즘을 요약하면

다음과 같다.

Step 1. 지름예지 라우팅

출발기본 모듈로부터 목적기본 모듈까지의 외부 경로길이 가 출발기본 모듈에서 지름예지로 연결된 기본 모듈로부터 목적기본 모듈로까지의 외부 경로길이보다 1 초과 하면 출발기본 모듈로부터 지름예지로 연결된 기본 모듈로 라우팅 한다.

Step 2. 출발기본 모듈의 내부 라우팅

단순 라우팅 알고리즘의 Step 1과 같다.

Step 3. 외부 라우팅

출발기본 모듈에서 노드와 목적기본 모듈에서 노드 조합에 따라 단순 라우팅 알고리즘의 Step 2와 Step 3의 순서를 바꾸어 라우팅 한다.

Step 4. 목적기본 모듈의 내부 라우팅

단순 라우팅 알고리즘의 Step 4와 같다.

최적 라우팅 알고리즘에서는 Step 1과 Step 3만을 다룬다. 출발노드 U가 그림 4처럼 기본 모듈(0,0,p)에 있다면, 목적노드 V는 $dx \leq \lfloor \frac{m}{2} \rfloor, dy \leq \lfloor \frac{n}{2} \rfloor$ 이면 A 영역, $dx \leq \lfloor \frac{m}{2} \rfloor, dy > \lfloor \frac{n}{2} \rfloor$ 이면 C 영역, $dx > \lfloor \frac{m}{2} \rfloor, dy > \lfloor \frac{n}{2} \rfloor$ 이면 B 영역, $dx > \lfloor \frac{m}{2} \rfloor, dy \leq \lfloor \frac{n}{2} \rfloor$ 이면 D 영역에 있다.

목적노드 V가 B영역에 있다면 xy축에 대칭, C영역에 있다면 x축에 대칭, D영역에 있다면 y축에 대칭 이동시키면 A영역과 동일하다. 즉 A영역의 라우팅 알고리즘에 대해서 C영역은 x축, D영역은 y축, B영역은 xy축에 대칭하여 라우팅하면 된다. 최적 라우팅 알고리즘은 A영역에 대해서만 다룬다. 그림 4의 A영역에서 x축으로 가장 먼 거리는 $\lfloor \frac{m}{2} \rfloor, y$ 축으로 가장 먼 거리는 $\lfloor \frac{n}{2} \rfloor$ 이다. 사선예지를 이용하면 x축과 y축을 동시에 1씩 증감 시키며 이동할 수 있으므로 보조정리 3과 같은 결과를 가질 수 있다.

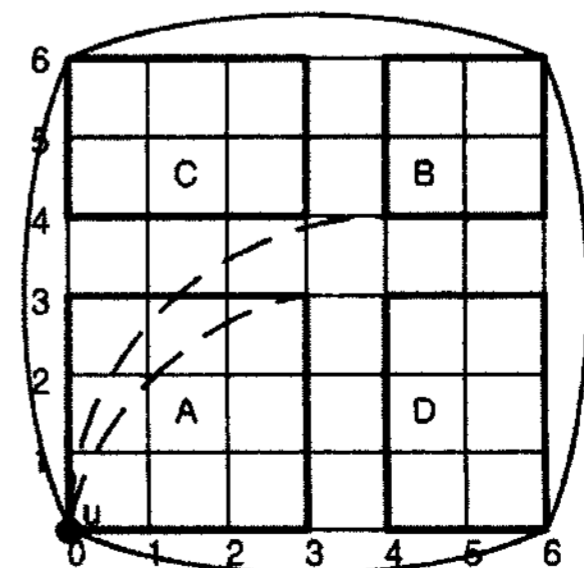


그림 4 PT(7,7) 라우팅 영역

보조정리 3. $PT(m,n)$ 의 외부에지의 최대사용횟수는

$$\text{Max}\left(\left\lfloor \frac{m}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor\right)$$

Step 1. 지름에지 라우팅

그림 5는 $PT(14,10)$ 의 A영역을 나타낸 것이다. 출발노드 U 는 기본모듈 $(0,0,p)$ 에 있고, 목적노드 V 의 좌표를 기준으로 1, 2, 3 영역으로 나누어 분석하였다. 목적노드 V 가 점선의 오른쪽에 위치할 때 지름에지로 라우팅 한다.

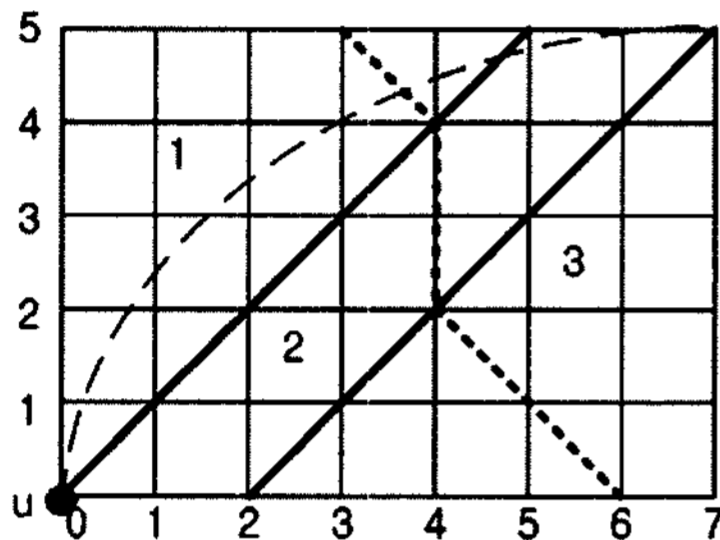


그림 5 사선에지와 지름에지의 선택

경우 1. $x \leq y$ (1 영역)일 경우

$$dx+dy \leq \left\lfloor \frac{m}{2} \right\rfloor + 1 \text{이면 사선에지, } dx+dy > \left\lfloor \frac{m}{2} \right\rfloor$$

+1이면 지름에지

경우 2. $y < x < y + (\left\lfloor \frac{m}{2} \right\rfloor - \left\lfloor \frac{n}{2} \right\rfloor)$ (2 영역)일 경우

$$dx \leq \left\lfloor \frac{\left\lfloor \frac{m}{2} \right\rfloor}{2} \right\rfloor + 1 \text{이면 사선에지, } dx > \left\lfloor \frac{\left\lfloor \frac{m}{2} \right\rfloor}{2} \right\rfloor$$

+1이면 지름에지

경우 3. $x \geq y + (\left\lfloor \frac{m}{2} \right\rfloor - \left\lfloor \frac{n}{2} \right\rfloor)$ (3 영역)일 경우

$$dx+dy \leq \left\lfloor \frac{n}{2} \right\rfloor + 1 \text{이면 사선에지, } dx+dy > \left\lfloor \frac{n}{2} \right\rfloor + 1$$

이면 지름에지

$\left\lfloor \frac{m}{2} \right\rfloor < \left\lfloor \frac{n}{2} \right\rfloor$ 일 경우는 위 $\left\lfloor \frac{m}{2} \right\rfloor \geq \left\lfloor \frac{n}{2} \right\rfloor$ 경우에 x 와 y 그리고 n 과 m 을 서로 맞교환하면 된다.

Step 3. 외부 라우팅

라우팅의 방향과 노드 U, V 의 기본모듈 안에서 노드 p 에 따라 다음 두 가지 경로 중 하나를 선택한다. ㉓ 경로는 사선(역 사선)에지, 가로(세로)에지 순서로 라우팅하고, ㉔ 경로는 가로(세로)에지, 사선(역 사선)에지 순서로 라우팅 한다. 라우팅의 방향은 그림 4의 A, B, C, D 영역과 같고 B, C, D 영역은 각각 A영역에 대칭되므로 A영역에 대해서만 다룬다. dx 와 dy 의 값에 따라 2가지 경우로 나눈다. $dx=dy$ 일 경우는 ㉓ 경로만으로

라우팅 된다.

경우 1. $dx < dy$ 일 경우, 노드 $p=\{1, 2, 3\}$ 이면 ㉓ 경로, 노드 $p=\{5, 6, 7\}$ 이면 ㉔ 경로, 노드 $p=\{0, 4, 8, 9\}$ 이고 노드 $p'=\{0, 1, 5, 6, 7, 8, 9\}$ 이면 ㉓ 경로, 노드 $p=\{0, 4, 8, 9\}$ 이고 노드 $p'=\{2, 3, 4\}$ 이면 ㉔ 경로이다.

경우 2. $dx > dy$ 일 경우, 노드 $p=\{2, 3, 6\}$ 이면 ㉓ 경로, 노드 $p=\{0, 1, 8\}$ 이면 ㉔ 경로, 노드 $p=\{4, 5, 7, 9\}$ 이고 노드 $p'=\{0, 1, 4, 5, 6, 7, 8\}$ 이면 ㉓ 경로, 노드 $p=\{4, 5, 7, 9\}$ 이고 노드 $p'=\{2, 3, 9\}$ 이면 ㉔ 경로이다. 예를 들어 $PT(8,8)$ 에서 $U=(0,0,6), V=(4,3,7)$ 인 경우 라우팅 경로는 다음과 같다. $(0,0,6) \rightarrow (0,0,5) \rightarrow (0,0,0) \Rightarrow (4,4,5) \rightarrow (4,4,9) \Rightarrow (4,3,6) \rightarrow (4,3,7)$

증명. Step 2와 Step 4는 보조정리 1에 따라 라우팅하므로 증명은 필요 없다. Step 1 또한 xy 그래프에서 두 지점사이의 거리에 관한 것으로 명백하다. 다음은 Step 3 외부 라우팅에 대해 증명이다.

$dx=dy$ 일 때, 가로에지-세로에지 순서로 라우팅 하는 경우나 세로에지-가로에지 순서로 라우팅 하는 경우보다 사선에지로 라우팅 하는 경우가 경로길이가 짧음을 증명한다. 가로에지-세로에지 순서로 라우팅 하는 경우나 세로에지-가로에지 순서로 라우팅 하는 경우는 값이 같으므로 가로에지-세로에지 순서로 라우팅 하는 경우만 설명한다.

그림 6에서 출발노드 U 가 있는 기본모듈에서 사선에지와 결합된 노드 2, 가로에지와 결합된 노드 1 그리고 세로에지와 결합된 노드 6은 서로 인접하다. 따라서 출발기본모듈의 내부 경로길이는 사선에지로 라우팅 하는 경우가 가로에지-세로에지 순서로 라우팅 하는 경우보다 내부경로 길이가 최대 1크다. 목적노드 V_1 이 있는 목적기본모듈에서 사선에지와 결합된 노드 3은 가로에지와 결합된 노드 4와 세로에지와 결합된 노드 9와 인접하다. 따라서 목적기본모듈의 내부 경로길이는 사선에지로 라우팅 하는 경우가 가로에지-세로에지 순서로 라우팅 하는 경우보다 내부경로 길이가 최대 1크다.

가로에지-세로에지 순서로 라우팅 하는 경우는 중간경유 기본모듈에서 내부경로 길이가 2이다. 사선에지로 라우팅 하는 경우는 중간경유 기본모듈에서 내부경로 길이가 1이다. 사선에지로 라우팅 하는 경우의 중간경유 기본모듈은 $dx-1$ 개이고, 가로에지-세로에지 순서로 라우팅 하는 경우의 중간경유 기본모듈은 $2dx-1$ 개이다. 따라서 사선에지로 라우팅 하는 경우의 중간경유 기본모듈에서 내부 경로길이는 세로에지-가로에지 순서로 라우팅 하는 경우 중간경유 기본모듈에서 내부 경로길이보다 $3dx-1(=2(2dx-1) - (dx-1))$ 만큼 적다.

사선에지로 라우팅 하는 경우 내부 경로길이는 가로에지-세로에지 순서로 라우팅 하는 경우보다 출발기본

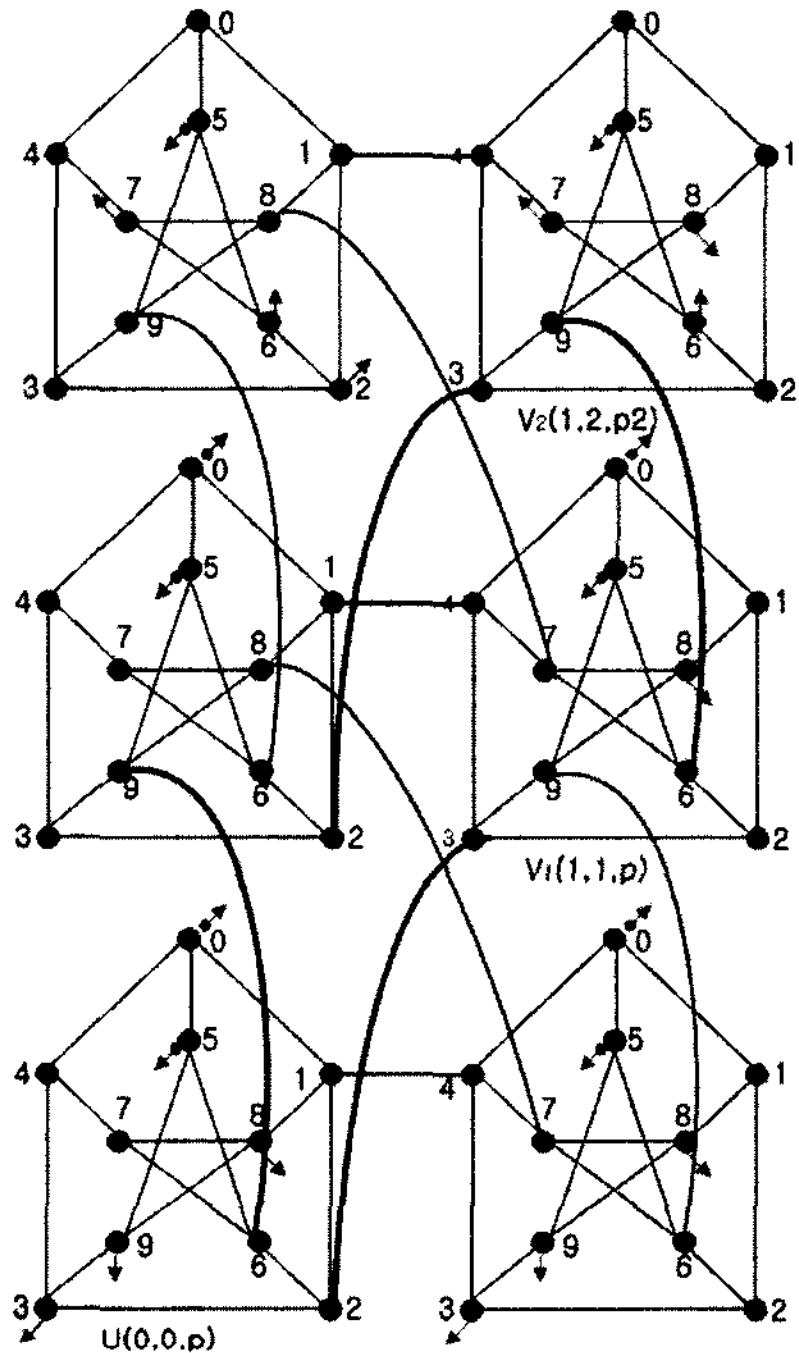


그림 6 내부에지에 따른 라우팅

모듈과 목적기본모듈에서 최대 2크며, 중간경유 기본모듈에서 3dx-1만큼 적다. 사선에지로 라우팅 하는 경우 외부 경로길이는 가로에지-세로에지 순서로 라우팅 하는 경우보다 dx만큼 적다. 따라서 사선에지로 라우팅 하는 경우의 경로 길이는 가로에지-세로에지 순서로 라우팅 하는 경우의 경로길이보다 4dx-3(=3dx-1+dx-2)만큼 적다.

dx≠dy 일 때, ㉑ 경로(사선에지-가로(세로)에지 순서)로 라우팅 하느냐 ㉒ 경로(가로(세로)에지-사선에지 순서)로 라우팅 하느냐 문제는 출발기본모듈과 목적기본모듈에서 내부 경로길이를 최소화하는 방향으로 선택하면 된다. 그림 6의 노드 U와 V₂를 보면 ㉑ 경로(U⇒V₁⇒V₂)와 ㉒ 경로(U⇒(0, 1, p)⇒V₂)가 있다. ㉑ 경로와 ㉒ 경로의 출발노드들은 경로 길이 1의 거리에 있으므로 출발기본모듈에서 어느 경로를 선택하던 경로 길이가 1차이 이하이다. ㉑ 경로와 ㉒ 경로의 목적노드들은 경로 길이 1의 거리에 있으므로 목적기본모듈에서 어느 경로를 선택하던 경로 길이가 1차이 이하이다. 따라서 출발기본모듈이나 목적기본모듈에서 경로 길이를 1줄이는 선택은 최소한 다른 선택보다 경로 길이가 같거나 작다. 최적 라우팅 알고리즘 Step 3은 출발(목적)기본모듈에서 내부 경로길이가 두 가지 경로 중 경로 길이가 1작은 경로를 선택하도록 되어있다. 따라서 최적 라우팅 알고리즘 Step 3은 최적이다. □

지름은 내부경로 길이와 외부경로 길이를 나누어서 분석한다. 보조정리 3에 의해서, 외부경로 길이는 $\text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor)$ 이다. 기본모듈의 내부경로 길이는 단위 피터슨 그래프의 지름은 2이므로 최대 2이다. 출발기본모듈, 목적기본모듈, $\text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) - 1$ 개 중간경유 기본모듈에서 내부경로가 있다. 출발기본모듈과 목적기본모듈의 내부경로 길이는 최대 2이다. 중간경유 기본모듈의 내부경로 길이는 라우팅에서 나타나는 에지의 연속된 순서에 따라 달라진다. 연속된 가로(세로)에지 사이의 중간경유 기본모듈은 내부경로 길이가 2이고, 연속된 사선에지 사이의 중간경유 기본모듈은 내부경로 길이가 1이다. 사선에지와 가로(세로)에지의 교차점에 있는 중간경유 기본모듈의 내부경로 길이도 2이다. 라우팅에서 사용되는 에지는 세로(가로)에지만으로 구성되거나, 사선에지만으로 구성되거나 사선에지-가로(세로)에지로 구성된다. 최악의 경우 가로(세로)에지만으로 라우팅이 될 수 있으므로 중간경유 기본모듈의 내부경로 길이는 $(\text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) - 1) \times 2$ 이다. 외부경로 길이 + 출발기본모듈 내부경로 길이 + 목적기본모듈 내부경로 길이 + 중간경유 기본모듈 내부경로 길이가 지름이다. 따라서 정리 4와 같은 결과를 가질 수 있다.

정리 4. $PT(m,n)$ 의 지름은 $(\text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) - 1 + 2) \times 2 + \text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor)$ 이다.

n=m일 때

지름 $k = (\text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) + 1) \times 2 + \text{Max}(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) = (\lfloor \frac{n}{2} \rfloor + 1) \times 2 + \lfloor \frac{n}{2} \rfloor$ 이다. n이 짝수 일 때와 홀수 일 때 지름이 다른 값을 가지므로 더 큰 값을 갖는 짝수 일 때로 정리하면 지름은 다음과 같다.

$$\leq (\frac{n}{2} + 1) \times 2 + \frac{n}{2}$$

$$\leq 1.5n + 2$$

bisection width는 연결망을 노드수가 같은 두개의 망으로 분리하기 위해 제거해야 할 에지의 최소 개수이다. 분리된 두 개의 망은 노드수가 1차이 보다 같거나 적어야 한다. bisection width가 적으면 보다 적은 에지 고장만으로도 망이 분리되어 제 역할을 할 수 없으므로 bisection width는 클수록 좋다. x축의 값이 j인 노드(j,y,p)들과 x축의 값이 j-1인 노드(j-1,y,p)들의 연결을 제거하기 위해서 필요한 최소 에지수는 가로에지 n개,

사선에지 n 개, 역 사선에지 n 개이다. 두 개의 망을 분리하기 위해서는 x 축의 값이 $(j + \lfloor \frac{m}{2} \rfloor, y, p)$ 인 노드들과 $(j-1 + \lfloor \frac{m}{2} \rfloor, y, p)$ 인 노드들도 분리되어야 한다. 이 노드들이 분리되기 위해 필요한 에지수 또한 $3n$ 개 이다. 추가로 x 축의 값이 $j \sim j-1 + \lfloor \frac{m}{2} \rfloor$ 의 값을 갖는 모든 노드로부터 x 축의 값이 j 보다 적고 $j-1 + \lfloor \frac{m}{2} \rfloor$ 보다 큰 모든 노드로의 지름에지도 제거되어야 한다. 이 에지수는 $2 \times \lfloor \frac{m}{2} \rfloor \times n$ 이다. $n \neq m$ 일 경우, 적은 수의 에지를 제거하는 방향으로 선택되어야 한다. 따라서 정리 5와 같은 결과를 가질 수 있다.

정리 5. $PT(m, n)$ 의 bisection width는 $6 \times \min(m, n) + 2 \times \min(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) \times \max(m, n)$ 이다.

$n=m$ 일 때

bisection width $b = 6 \times \min(m, n) + 2 \times \min(\lfloor \frac{m}{2} \rfloor, \lfloor \frac{n}{2} \rfloor) \times \max(m, n) = 6n + 2n \times \lfloor \frac{n}{2} \rfloor$ 이다. n 이 짝수일 때와 홀수일 때 다른 값을 가지므로 더 적은 값을 갖는 홀수일 때로 정리하면 다음과 같다.

$$\geq n^2 + 5n$$

분지수가 상수인 상호연결망은 망을 확장할 때 통신 링크를 재설치 하지 않아도 된다는 장점이 있고, bisection width가 클수록 고장에 대해 망이 효율적인 특성을 갖는다. 다음은 분지수가 상수인 연결망이 같은 노드 개수를 가질 때 망 비용, 지름 그리고 bisection width를 비교 하였다[2,3,9]. PT 네트워크가 비교된 다른 연결망보다 지름과 망 비용은 적고 bisection width는 크다. 표 2에서 N 은 노드 개수이다.

표 2 연결망들의 망 비용과 bisection width

연결망	망 척도	분지수	지름	망 비용	Bisection width
honeycomb mesh	3	3	$1.63 \sqrt{N}$	$4.90 \sqrt{N}$	$0.82 \sqrt{N}$
diagonal mesh	4	4	$\sqrt{N-1}$	$4 \sqrt{N-1}$	$4 \sqrt{N}$
torus	4	4	\sqrt{N}	$4 \sqrt{N}$	$2 \sqrt{N}$
hexagonal torus	6	6	$0.58 \sqrt{N}$	$3.46 \sqrt{N}$	$4.61 \sqrt{N}$
honeycomb torus	3	3	$0.81 \sqrt{N}$	$2.45 \sqrt{N}$	$2.04 \sqrt{N}$
PT	4	4	$0.47 \sqrt{N}$	$1.9 \sqrt{N}$	$0.1N + 1.58 \sqrt{N}$

3.4 해밀톤 사이클

연결망이 해밀톤 경로나 해밀톤 사이클을 지나면 링

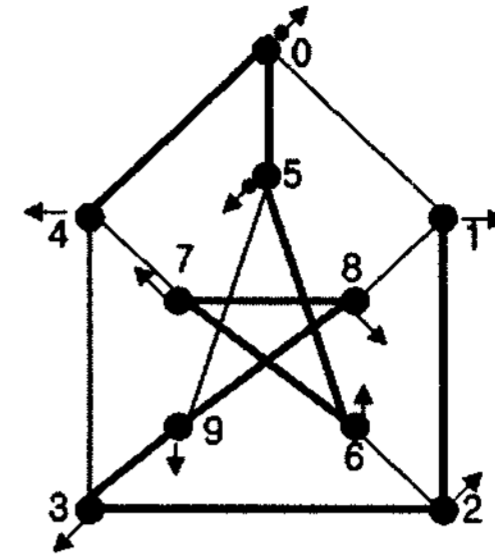


그림 7 해밀톤 경로

이나 선형배열을 쉽게 구현할 수 있어서 병렬처리에 유용하게 쓰이는 파이프라인으로 활용된다. 피터슨 그래프는 보조정리 2에 의해서 서로 인접하지 않는 두 노드 사이에 해밀톤 경로가 있다[19]. 그림 7의 굵은선은 출발노드 4와 목적노드 1사이의 해밀톤 경로를 나타낸다. 해밀톤 사이클 알고리즘의 기본 개념은 다음과 같다. 기본모듈의 임의의 노드로부터 출발하여 그림 7의 경로를 따라 노드 1에 도착한다. 노드 1에서 자신을 제외한 모든 중간경유 기본모듈을 순회하고 출발기본모듈의 노드 4에 도착한다. 노드 4로부터 그림 7의 경로를 따라 출발노드에 도착한다. 출발기본모듈을 제외한 모든 중간경유 기본모듈을 순회 할 때 외부에지는 가로에지와 세로에지만을 이용한다. 가로에지와 세로에지에 결합된 노드들은 보조정리 2에 의해 해밀톤 경로가 있으므로 모든 중간경유 기본모듈에서 해밀톤 경로가 있다. 다음은 해밀톤 사이클을 만드는 알고리즘이다.

Step 1. 출발기본모듈에서 출발 노드로부터 노드 1까지 해밀톤 경로를 따라 이동한다.

Step 2. y 축으로 마지막 행이라면 $n-1$, 그렇지 않다면 $n-2$ 번째 오른쪽 기본모듈로 이동한다.

Step 3. y 축으로 마지막 행이라면 Step 7, 그렇지 않다면 위쪽 기본모듈로 이동한다.

Step 4. y 축으로 마지막 행이라면 $n-1$, 그렇지 않다면 $n-2$ 번째 왼쪽 기본모듈로 이동한다.

Step 5. y 축으로 마지막 행이라면 Step 7, 그렇지 않다면 위쪽 노드로 이동한다.

Step 6. Step 2~Step 5를 반복

Step 7. 아래쪽으로 $m-1$ 번 이동

Step 8. 오른쪽으로 1번 이동하면 출발기본모듈의 노드 4에 도착

Step 9. 출발기본모듈의 노드 4에서 피터슨 그래프의 해밀톤 경로를 따라 출발노드에 도착한다.

그림 8에서, 굵은선은 알고리즘에 따라 해밀톤 사이클을 보인 결과이다. n 이 홀수이면 Step 3에서 Step 7로 진행하고, 짝수이면 Step 5에서 Step 7로 진행한다.

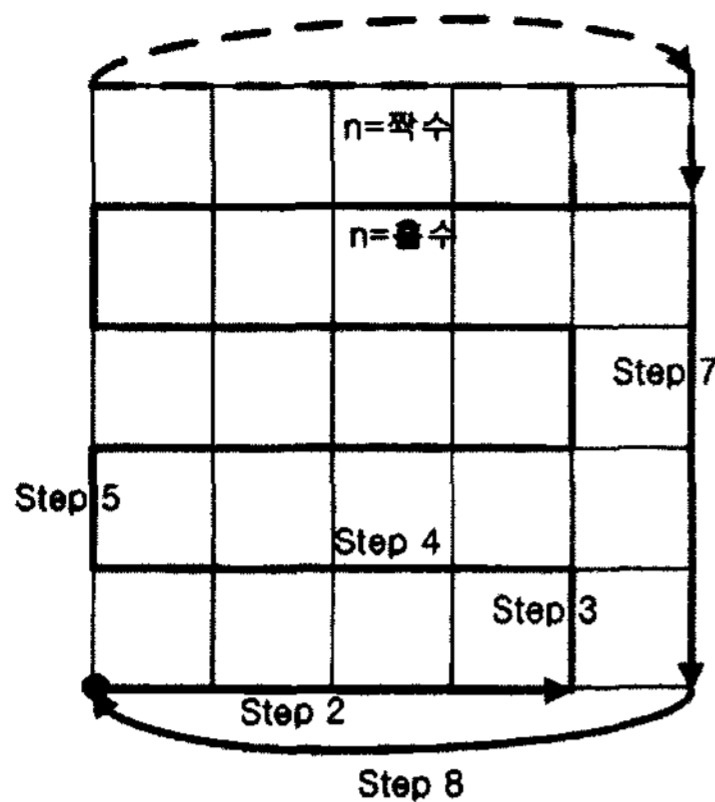


그림 8 해밀턴 사이클

4. 결론

멀티컴퓨팅 시스템의 성능은 각 처리기들의 성능과 처리기들을 연결하는 상호연결망 구조 그리고 이 환경에서 수행되는 병렬 알고리즘에 의해 결정된다. 이러한 요소들 중 특히 처리기를 연결하는 상호연결망의 구조는 전체 시스템의 성능 및 확장성에 영향을 미치고, 효과적인 병렬 알고리즘을 작성하는데 매우 중요한 요소이다.

본 논문에서는 분지수가 상수인 다이아고날 메쉬, 토러스, 허니컴브 메쉬, 허니컴브 토러스, 헥사고날 토러스보다 지름, 망 비용 그리고 bisection width가 개선된 새로운 연결망 PT 네트워크를 제안하였다. PT 네트워크는 동일한 노드수 일 때 다이아고날 메쉬나 토러스에 비해 지름이 대략 1/2 정도이고, 헥사고날 토러스보다 망 비용이 대략 1/2가량 정도이고, 허니컴브 토러스보다 bisection width가 대략 $0.05\sqrt{N}$ (N =노드수)배 만큼 크다. 특히 동일한 노드 수(N)를 가질 때 허니컴브 토러스의 지름은 $0.81\sqrt{N}$, PT 네트워크의 지름은 $0.47\sqrt{N}$ 이고, 허니컴브 토러스의 망 비용은 $2.45\sqrt{N}$, PT 네트워크의 망 비용은 $1.9\sqrt{N}$ 으로 널리 알려진 허니컴브 토러스보다 PT 네트워크가 지름과 망 비용에서 개선된 값을 가진다.

PT 네트워크는 기본모듈을 단일노드로 보면 망 전체가 노드(에지) 대칭이며, 격자구조를 가지고 있어서 병렬 알고리즘을 작성하기 쉽고, 향후 메쉬 부류의 연결망으로 임베딩하는 연구가 이루어질 수 있다. 고장 허용 알고리즘을 위해 두 노드 사이의 에지-disjoint한 병렬 경로는 매우 중요하다. 망 분지수가 4이고 4개의 에지-disjoint한 병렬 경로가 있음을 확인하였다. 최단 경로 길이를 갖는 병렬 경로를 만드는 것은 연구해볼만한 가치가 있다.

참고 문헌

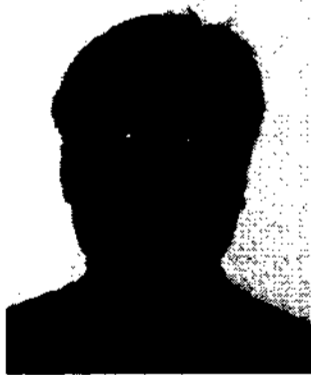
- [1] Ohring and S. K. DAS, "Folded Petersen cube networks: new competitors for the hypercubes," IEEE Transactions on Parallel and Distributed Systems, Vol.7, No.2, pp. 151-168, Feb 1996.
- [2] K. W. Tang and S. A. Padubidri, "Diagonal and toroidal Mesh Networks," IEEE trans. comput., Vol.43, No.7, pp. 815-826, Jul 1994.
- [3] I. Stojmenovic, "Honeycomb Network: Topological Properties and Commnication Algorithms," IEEE trans. on Parallel and Distributed Systems, Vol.8, No.10, pp. 1036-1042, Oct 1997.
- [4] M. S. Chen and K. G. Shin, "Addressing, Routing, and Broadcasting in Hexagonal Mesh Multiprocessors," IEEE trans. comput., Vol.39, No.1, pp. 10-18, Jan 1990.
- [5] Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," IEEE Trans. Comput., Vol.37, No.7, pp. 867-872, July 1988.
- [6] A. EI-Amawy and S. Latifi, "Properties and Performances of Folded Hypercubes," IEEE Trans. Parallel and Distributed System, Vol.2, No.1, pp. 31-42, 1991.
- [7] K. Efe, "A Variation on the Hypercube with Lower Diameter," IEEE Trans. comput., Vol.40, No.11, pp. 1312-1316, 1991.
- [8] J-H. Park, "Circulant Graphs and Their Application to Communication Networks," Ph.D. Thesis, Dept. of Computer Science, KAIST, Taejon Korea, 1992.
- [9] V. E. Mendia and D. Sarkar, "Optimal Broadcasting on the Star Graph," IEEE Trans. Parallel and Distributed Systems, Vol.3, No.4, pp. 389-396, July 1992.
- [10] C-H Yeh and E. Varvarigos, "Macro-Star Networks: Efficient Low-Degree Alternatives to Star Graphs for Large-Scale Parallel Architectures," Frontier'96, Symp. on the Frontiers of Massively Parallel Computation, Oct. 1996.
- [11] S. Latifi and P. K. Srimani, "Transposition Networks as a Class of Fault-Tolerant Robust Networks," IEEE Trans. Comput., Vol.45, No.2, pp. 230-238, Feb. 1996.
- [12] H. O. Lee, J. S Kim, K. W. Park and J. H. Seo, "Matrix Star Graphs: A New Interconnection Network Based on Matrix Operations," Lecture Note in Computer Science : ACSAC 2005, LNCS 3740, pp. 478-487, Oct. 2005.
- [13] V. Bokka, H. Gurla, S. Olariu and J. L. Schwing, "Podality-Based Time-Optimal Computations on Enhance Meshes," IEEE Trans. Parallel and Distributed System, Vol.8, No.10, pp. 1019-1035, Oct 1997.
- [14] G. Memmi and Y. Railrd, "Some New Results About the (d,k) Graph Problem," IEEE trans. comput., Vol.c-31, No.8 pp. 784-791, Aug 1982.

- [15] K. W. Cattermole and J.P. Summer, "Communication networks based on the product graph," Proc. Inst. Elec. Eng., Vol.124, pp. 38-48, Jan 1977.
- [16] -----, "On the impossibility of certain Moore graph," J. combinatorial Theory (B), pp. 253-254, Apr 1966.
- [17] E. Bannai and T. Ito, "On finite Moore graphs," J Fac. Sci., Tokyo Univ., pp. 191-208, 1973.
- [18] R. M. Storwick, "Improved construction technique for (d,k) graphs," IEEE trans. Electron. Comput., pp. 1214-1216, Dec 1970.
- [19] G. Chartrand and R.J. Wilson, "The Petersen graph," in Graphs and Applications, F. Harary and J.S. Maybee, eds., pp. 69-100, 1985.
- [20] D. A. Holton and J. Sheehan, "The Petersen Graph," in The Petersen Graph, Cambridge University Press, eds., pp. 1-45, 1993.
- [21] C. H. Yeh and B. Parhami, "Routing and Embeddings in Cyclic Petersen Networks: An Efficient Extension of the Petersen Graph," 1999 International Conference on Parallel Processing (ICPP'99), pp. 258-265, 1999.
- [22] W. Lei and C. Zhiping, "Research on Petersen Graphs and Hyper-cubes Connected Interconnection Networks," ACASC 2006, LNCS 4186, pp. 495-501, 2006.
- [23] S. K. Das and A. K. Banerjee, "Hyper Petersen network: yet another hypercube-like topology," Fourth Symposium on the Frontiers of Massively Parallel Computation, pp. 270-277, Oct 1992.



장 문 석

1983년 광운대학교 전자계산학과(이학사). 1985년 광운대학교 전자계산학과(이학석사). 1995년 광운대학교 전자계산학과(이학박사). 1988년~현재 순천대학교 공과대학 컴퓨터공학과 교수. 관심분야는 인공지능, GIS, 병렬컴퓨터



서 정 현

1998년 2월 순천대학교 컴퓨터과학과 학사. 2000년 9월~2003년 8월 순천대학교 정보전산원 조교. 2002년 2월 컴퓨터과학과 석사. 2002년 3월~현재 컴퓨터과학과 박사과정. 관심분야는 상호연결망, 인공지능



이 형 옥

1994년 2월 순천대학교 전산학과(학사) 1996년 2월 전남대학교 전산통계학과(석사). 1999년 2월 전남대학교 전산통계학과(박사). 1999년 10월~2002년 2월 한국정보사회진흥원(선임연구원). 2006년 1월~2007년 7월 University of Texas at Dallas(방문교수). 2002년 3월~현재 순천대학교 컴퓨터교육과 교수