효율적인 다중 일치 알고리즘

김 수 진, 류 재 철
충남대 전산학과

An Efficient Multiparty Consensus Algorithm

Soo-Jin Kim , Jea-Cheol Ryou
Chungnam National University

요 약 문

본 논문에서는 시스템내의 모든 site들에게 분산되어 있는 정보들을 수렴하여 일치를 이루고, 그 결과를 모든 site들이 알도록 하는 다중 일치 알고리즘을 위한 효과적인 통신 방법을 제안하고자 한다. 분산 시스템에 참여하는 computer 또는 site들의 수를 N 이라 할때, $O(N^2)$의 message를 필요로 하면서 한 round 안에 일치를 이룰 수 있는 알고리즘은 message의 수가 너무 많다는 것이 단점이다. 이에 본 논문에서는 Finite Projective Planes을 이용하여 message의 수를 줄이면서 두 round 안에 일치를 이룰 수 있는 통신방법을 제안한다. 이때, 각 round 마다 필요한 message의 수는 $O(N\sqrt{N})$이다. 또한, 이 통신 방법에서 이용되는 Finite Projective Planes을 구축하는 알고리즘을 제안하고자 한다.

1. 개 요

본 논문에서 고려하는 분산 시스템은 point-to-point 통신 네트워크에 의해 연결된 독립된 computer (Autonomous computer) 또는 site들로 이루어져 있다. 이 시스템은 global clock이나 global memory을 가지고 있지 않으며, 미리 정해진 중앙 조절자 (central controller) 를 포함하고 있지 않다. 또한 시스템 내의 모든 site들은 동등하며, 서로 message를 통해 통신한다. 이러한 시스템 하에서 여러 site에 분산 되어 있는 정보를 매개 변수로 갖는associative 함수나 명제(predicate)를 계산하여 그 결과를 모든 site들이 알도록 하는 것이 본 논문의 목적이다.

예를 들어, 각 site는 'Yes'(1) 또는 'No'(0)에 대한 값을 하나씩 가지고 있고, 각 site들은 이러한 값들을 수렴하여 전체 의견을 알고 싶다고 하자. 이러한 계산을 하기 위해서는 여러 site에 분산 되어 있는 정보들을 수집해야 만 한다. Global memory나 중앙 조절자가 없으므로 이 계산은 여러 site들의 협조와 조정 아래 이루어져야만 한다.

위와 같은 계산을 위한 가능한 방법으로 각 site가 자신의 정보('Yes' 또는 'No')를 시스템내의 다른 모든 site에게 전달함으로서 전체의견을 수렴하는 방법이 있을수 있다.

이 방법은 한 round[1]의 message 교환을 통해 일치를 이룰수는 있지만, 시스템내의 site의 갯수를 N이라 할때, N*(N-1)개의 message가 필요하다는 것이 단점이다. 이에 본 논문에서는 message의 갯수를 줄이면서 두 round 안에 일치를 이룰수 있는 효과적인 통신 방법을 제안하고자 한다. 이 통신 방법은 Finite Projective Planes을 기초로 하며 각 round 마다 $O(N\sqrt{N})$의 message 만을 사용하여 일치를 이룰수 있다. 이러한 방법은 암호학에서도 효율적인 알고리즘 개발에 널리 이용되리라 기대된다.

다음장에서는, Finite Projective Planes을 개략적으로 소개하고 필요한 정리들을 서술한다. 3장에서는 Finite Projective Planes을 이용한 통신 방법이 제안되며, 4장에서는 Finite Projective Planes을 구축하는 알고리즘을 제안한다. 5장에서는 이러한 통신 방법을 응용한 최대,최소값을 찾는 알고리즘을 다룬다. 6장에서는, 두 round 안에 일치를 이루기 위해 필요한 message 갯수의 lower bound는 $O(N\sqrt{N})$임을 보여 주고, 7장에서 앞으로의 연구방향과 결론을 정리한다.

2. Finite Projective Planes

모든 일치 알고리즘에 있어서, 각 site는 자신과 통신하는 다른 site들의 집합과 연관되어 있다. 앞장에서 보았듯이 이 집합은 자신을 제외한 다른 모든 site들로 이루어질 수도 있고, ring 구조로 연결된 시스템에서는 자신의 이웃 site가 이집합을 구성 할 수도 있다. 이러한 집합의 크기 (cardinality), 구성 원소 (membership), 교차(intersection)등의 특성은 일치 알고리즘의 message 복잡도(complexity), 대칭성(symmetry), message 교환의 round 횟 수에 영향을 미친다.

본 논문에서 제안하는 통신방법은 message의 갯수를 줄이기 위해 각 site는 시스템내의 일부 site들과만 통신할 것을 요구한다. 시스템내의 일부 site들과만 통신하면서도 필요한 모든 message를 받은것과 같은 효과를 얻기 위해서는 각 site들과 연관된 집합을 잘 선택하여야만 한다. 또한 각 site들은 일치를 이루기 위해 서로 협조하고 조정하는데 있어서 같은 양의 역활을 담당해야 한다. 즉 각 site들간에 대칭(symmetry)을 유지해야만 한다. 이러한 대칭을 이루기 위해 다음과 같은 조건을 만족해야 한다.

1) 모든 site들은 각 round 마다 같은 수의 message를 보내야 한다.
2) 각 site와 통신하는 집합의 크기는 서로 같아야 한다.
3) 각 site는 같은 수의 집합에 포함되어야 한다.

본 논문에서 제안하는 통신 방법은 각 round에 기껏해야 $O(N\sqrt{N})$개의 message를 보낼 수 있으므로 각 site와 통신하는 집합의 크기는 $O(\sqrt{N})$ 이어야 함을 알수 있다. 또한 각 site는 두 round안에 다른 site에서 수행되는 계산에 영향을 미칠수 있어야 하므로, 각

[1] 각 site가 message를 전달해 주고 다른 site로 부터 필요한 모든 message를 받을 때 까지를 한 round 이라 한다.

site와 연관된 집합들의 교차 그래프(intersection graph)[2]는 연결 그래프(connected graph)이고 동시에 완전 그래프(complete graph)이어야 함을 알수 있다.

위의 특성을 만족하는 집합을 만드는 한가지 가능한 방법은 Finite Projective Planes을 사용하는 것이다. (Finite Projective Planes의 구축은 4장에서 논한다.)

Finite Projective Planes은 다음의 공리를 만족하는 유한개의 선(lines)과 점(points)들로 이루어 진다.

공리 1 : 서로 다른 두 점은 오직 하나의 공통 선위에 놓여 있다.
공리 2 : 서로 다른 두 선은 오직 하나의 공통 점을 지난다.
공리 3 : 서로 다른 4개의 점들이 있다. 이 점들 중 3개는 같은 선위에 있지 말아야 한다.

공리 3 은 한 선위에 모든 점들이 놓여있는 degenerate finite projective plane을 제외시키기 위한 조건이다. Finite Projective Plane의 각 점을 site로 간주하며, 선들을 각 site와 연관된 집합으로 간주한다. 7개의 점을 가진 Projective Plane이 [그림 1]에 보여 진다.

다음은 본 논문에서 제한한 통신 방법을 위해 필요한 Finite Projective Plane에 관한 정리들이다.

정리 2.1 : Finite Projective Plane에서, 모든 점은 같은 갯 수의 선위에 놓여 있고, 모든 선은 같은 갯 수의 점을 지난다.

정리 2.2 : Finite Projective Plane에서, 각 점을 지나는 선들의 수는 각 선분상의 점들의 갯 수와 같다.

정리 2.3 : 각 선분 상에 m+1 개의 점이 있고, 각 점을 지나는 m+1 개의 선을 가진 Projective Plane은 m^2+m+1 개의 점과 m^2+m+1개의 선을 갖는다.

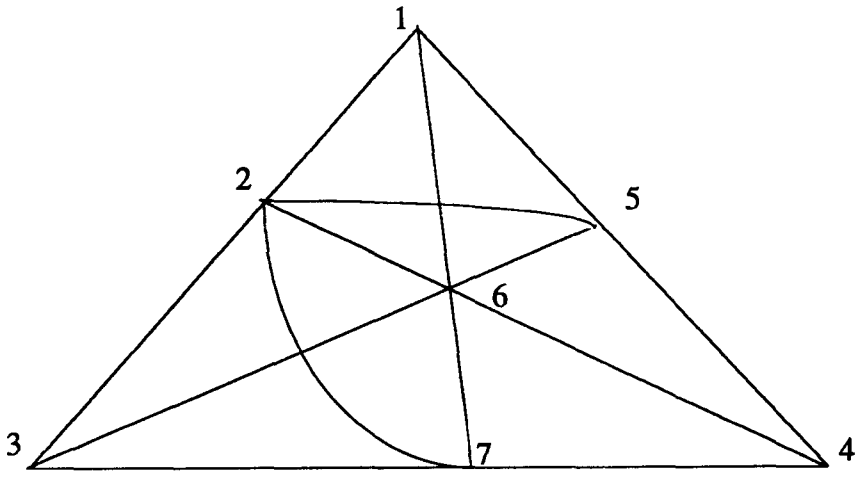
정리 2.4 : 소수 p와 양의 정수 k에 대해 $m = p^k$ 이라면, order m의 Projective Plane이 존재 한다.

정리 2.3에서 m을 Finite Projective Plane의 order라고 한다.

3. 통 신 방 법

이 장에서 논의되는 통신 방법은 Finite Projective Planes에 의해서 얻어진 집합에 따라 각 site들은 시스템 내의 일부 site들과 만 통신하며, 각 site는 Finite Projective Planes의 한 점으로 간주된다. (site와 점은 서로 같은 의미로 번갈아 사용하였다.) 일단 시스템내의 site들의 갯수 N 에 대하여 Finite Projective Planes이 존재한다고 가정하며, 존재하지 않는 경우에 대해서는 이 장의 끝에서 논하기로 한다.

[2] $S = \{ S_1, S_2, ..., S_n \}$ 이라 하고 S_i는 집합이라 하자. S를 절점(Vertex)으로 갖고, $S_i \cap S_j \neq \emptyset$ 이라면 두절점 사이에 철선(edge) (S_i, S_j) 가 존재하도록 하는 그래프를 S의 **교차 그래프(intersection graph)**라 한다.



[그림 1] 7개의 점을 가진 Finite Projective

Projective Plane의 각 점 i는 전단사 함수 F에 의하여 그 점 i와 인접한 (incident) 유일한 선 L_i와 연관된다. 이 선 L_i는 site i와 연관된 집합이 될 것이다. 함수 F의 존재 여부는 연역된 이분 그래프(induced bipartite graph)에서 saturating matching의 존재를 증명함으로서 확인된다[4]. [그림 1]의 Projective Plane에 대한 각 site i와 연관된 집합 L_i는 [그림 2]에서 주어진다.

한 점 i 를 고려해 보자. 점 i는 정리 2.3에 따라 그점에 인접한(incident) 선 L_i와 m개의 다른 선을 가지고 있다. 그 m개의 선은 함수 F의 역함수인 F^{-1}에의해 유일한 점과 사상(mapping)될 수 있다. 이러한 점을 각각 $i_1, i_2, ..., i_m$이라 하자. 본 논문에서 제안하는 알고리즘의 각 단계마다 site i는 $j \in L_i$ 이거나 $i \in L_j$ 이도록 하는 site j ($i \neq j$) 들에게만 message을 보내고 받는다. 물론 $i \in L_j$ 이도록하면서 $i \neq j$ 인 site 들은 F^{-1}에 의해서 결정된 $i_1, i_2, ..., i_m$임을 알수 있다. 그러므로 각 단계마다 한 site는 2m개의 message를 보내면 된다. 점 i가 받는 message의 수는 다음 보조정리(lemma)에 의해서 알 수 있다.

보조정리 3.1 : site i 는 2m개의 site들로 부터 message를 받는다. 즉, $j \in L_i$ 이거나 $i \in L_j$이면서 $i \neq j$인 site j들로부터 message 를 받는다.

$L_1 : \{ 1, 2, 3 \}$
$L_2 : \{ 2, 4, 6 \}$
$L_3 : \{ 3, 5, 6 \}$
$L_4 : \{ 4, 5, 1 \}$
$L_5 : \{ 5, 2, 7 \}$
$L_6 : \{ 6, 7, 1 \}$
$L_7 : \{ 7, 3, 4 \}$

[그림 2] Projective Plane의 각 site와 연관된 집합들

증명 : 각 점은 그 점에 인접한(incident) 모든 선과 연관된 점들에게 message를 보내야 한다. 그러므로, 선 L_i상의 모든 점들은 *i*에게 message를 보낸다. 그리고, 정리2.3에 따라 그런 점들이 m개 있음을 알 수 있다. 또한, 각 점은 그점과 연관된 선 상의 모든 다른 점들에게도 message를 보내야 한다. 점 *i* 에 인접한 선은 L_i를 제외하고 m개가 있으며, 각 선에 연관된 유일한 한 점이 있다. 이런 m개의 점들 각각은 *i* 에게 message를 보내야 한다. 공리 2에 의하여, *i* 에 인접한(incident) m+1개의 선은 *i* 를 제외한 어떠한 점도 공통으로 가질 수 없다는 것을 알 수 있다. 그러므로 *i* 가 받은 2m개의 message는 2m개의 서로 다른 점들로 부터 온 것임을 알 수 있다. ■

Site들과 유도된 통신 경로들(communication paths)은 degree 2m의 regular 그래프를 형성한다. 그래프의 각 절선(edge)은 양방향 통신 경로(bidirectional communication path)를 나타낸다. [그림 1]과 관련된 통신 경로 그래프는 [그림 3]에서 보여 준다.

보조 정리 3.2 : 각 단계마다, 각 site는 $O(\sqrt{N})$개의 message를 보낸다.

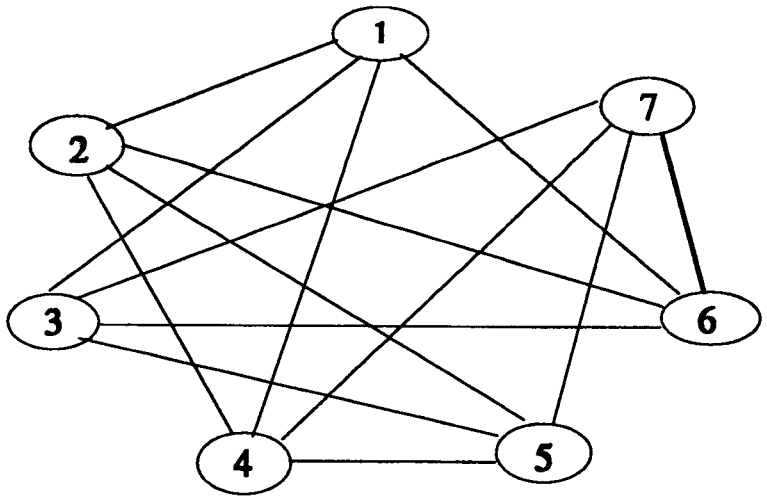
증명 : 고려되어야 할 두가지 경우가 있다. 주어진 N에 대하여,

1)Finite Projective Plane이 존재한다.

각 site은 각 단계마다 2m개의 message를 보낸다. $N = m^2 + m + 1$이므로, $2m = -1 + \sqrt{4N - 3}$ 이다. 그러므로 message의 갯수는 $O(\sqrt{N})$이다.

2)Finite Projective Plane이 존재 하지 않는다.

$m_0 = \lfloor -1 + \sqrt{4N - 3} / 2 \rfloor$ 이라 하고, m_0보다 더 큰 정수 중에서 소수의 멱승(Power)인 가장 작은 정수를 m_1이라 하자. 참고 문헌 [3]에서 어떠한 $n \geq 1$ 에 대하여, $n < p \leq 2n$이도록 하는 소수 p가 적어도 하나 존재함을 보여주었다. 그러므로, $m_0 < m_1 \leq 2m_0$이며, m_1이 $O(\sqrt{N})$임을 알 수 있다. 정리 2.4에 따라, $N^* = m_1^2 + m_1 + 1$을 가진 Finite Projective Plane을 만들 수 있다. $N^* > N$이므로, $N^* - N$개의 가상(virtual) site가 시스템에 첨가되어야 하며, 각 단계마다 $2m_1 = O(\sqrt{N})$개의 message가 한 site에 의해서 보내진다. ■



[그림 3] 통신 경로 그래프

가상 site들은 계산되어질 함수나 명제(Predicate)에 따라 초기값이 다르다는 것을 빼고는, 실제 site들과 같은 알고리즘을 실행한다. 합 계산에 있어서, 가상 site들의 초기값은 0 이 될 것이고, 최대, 최소값 계산에 있어서는, MININT 나 MAXINT가 될 것이다. 가상 site의 참여에도 불구하고, $N^* = O(N)$이므로 각 단계마다 오직 $O(N\sqrt{N})$개의 message가 보내진다.

4. Finite Projective Planes의 구축

4.1 알고리즘

이장에서는 Finte Projective Palnes를 구축하는 알고리즘을 제안한다. Finte Projective Palnes의 각 선들은 직교 라틴 방격(orthogonal latin square)의 완전 표준 집합(complete standardized set)에 의해서 만들어 진다. 라틴 방격(Latin Square)에 관한 사항은 참고논문 [2]를 참고 하기 바란다. 이러한 라틴 방격은 order가 소수의 멱승일 때 존재한다.

알고리즘 : order m 의 Finite Projective Planes의 구축

$K = m + 1$ 이라 하고 K-2개의 표준 직교 라틴 방격들, $S_1, S_2, ..., S_{K2}$이 주어진다고 하자. Finite Projective Planes의 각 선들은 다음과 같은 $N \times K$ 행렬의 각 행에 의해서 표현된다.

	W				
M_1	N_1				
	:				
	N_p				
M_2	N_1^T	V_{11}	V_{12}	....	V_{1q}
M_3	N_1^T	V_{21}	V_{22}	....	V_{2q}
. .	. .	. .	. .	. . .	. .
M_k	N_1^T	V_{p1}	V_{p2}	. . .	V_{pq}

$N \times K$

단, $p = K - 1$ 고 $q = K - 2$

$W = [\ 1\ 2\ \cdots\ K\ ]_{1 \times K}$

$$M_i = \begin{bmatrix} i \\ i \\ : \\ i \end{bmatrix}_{(K-1)\times 1} \qquad 1 \le i \le K$$

$N_i = [K+1 \;\; K+2 \;\; \cdots \;\; K+(K-1)]_{1\times(k-1)} + [r \; r \; \cdots \; r]_{1\times(k-1)}$, $1 \le i \le K-1$ 이고 $r = (K-1)\times(i-1)$ (N_p 의 마지막 원소는 N이어야 한다.) 이다.
N_1^T는 N_1의 행과 열을 교환 함으로서 얻을 수 있는 N_1의 전치 행렬이다. 즉,

$$N_1^T = \begin{bmatrix} K+1 \\ K+2 \\ \vdots \\ K+(K+1) \end{bmatrix}_{(K-1)\times 1}$$

V_{ij} ($i = 1, 2, ..., K-1$이고 $j = 1, 2, ..., K-2$) 는 크기가 (K-1)×(K-1)인 직교 라틴 방격들, $S_1, S_2, ..., S_{K-2}$, 로 이루어진 완전 표준 집합으로 부터 얻어 진다. 라틴 방격들을 다음과 같이 표시하자.

$S_1 = [l_{ij}^{1}]_{(K-1)\times(K-1)}$
$S_2 = [l_{ij}^{2}]_{(K-1)\times(K-1)}$
...
$S_{K-2} = [l_{ij}^{K-2}]_{(K-1)\times(K-1)}$
단, $l_{ij} \in \{1, 2, ..., K-1\}$ 이다.

N_i와 라틴 방격들, $S_1, S_2, ..., S_{K-2}$, 에 대하여, V_{ij} 는 다음과 같이 만들어 진다.

$$V_{ij} = \begin{bmatrix} N_{j+1}^{1} \text{ at } l_{i1}^{j} \text{ th row} \\ N_{j+1}^{2} \text{ at } l_{i2}^{j} \text{ th row} \\ \vdots \\ N_{j+1}^{K-1} \text{ at } l_{i(K-1)}^{j} \text{ th row} \end{bmatrix}_{(K-1)\times 1}$$

단, N_i^j는 N_i의 j번째 열에 있는 원소를 나타낸다.

알고리즘에 따라, V_{ij}는 N_{j+1}의 원소들로 이루어지고, 라틴방격 S_j의 i번째 행이 V_{ij}의 각 원소의 위치를 결정하기 위해 사용되었음을 알 수 있다.

이렇게 만들어진 order m 의 Finte Projective Planes의 각 선들은 3장에서 보았듯이 전단사 함수 F에 의하여 유일한 점과 연관될 수 있다.

4.3 복잡도 (Complexity)

직교 라틴 방격의 완전 표준 집합이 주어진다면, 이 알고리즘의 time complexity는 $O(N \times K)$이다. $K = m + 1$ 이고 $N = m(m+1) + 1$ 이므로, 복잡도는 $O(N^{3/2})$ 또는 $O(K^3)$임을 알수 있다. 또한, order (K-1) 의 직교 라틴 방격의 완전 표준 집합이 $O(K^3)$안에 생산될 수 있다. 그러므로, 이 알고리즘을 사용해서, Finte Projective Planes의 구축은 $O(K^3)$의 time complexity 를 가지고 생성될 수 있다.

5. 최대, 최소값 계산을 위한 분산 알고리즘

분산 시스템내에 N개의 서로 다른 값들이 각 site마다 하나씩 분산되어 있다고 하자. 문제는 그러한 값들 중 최대값 또는 최소값을 시스템내의 모든 site들이 알도록 하는 것이다. 앞 장에서 소개된 통신 방법을 이용한 알고리즘을 이 장에서 설명하였다. 이 알고리즘은 참고문헌[3]에 근거한 것이다.

모든 site는 같은 알고리즘을 수행하고, site i 에서 최대값을 찾기 위해 수행하는 알고리즘은 다음과 같다.

최대값 계산 알고리즘

단계 1. $j \in L_i$ 이거나 $i \in L_j$ 이면서 $i \neq j$ 인 site j에게 자신의 값 V_i를 보낸다.

단계 2. (1) 보조정리 3.1에서 명시된 2m개의 site들로 부터 값을 받을 때까지 기다린다.

(2) 받은 값들과 V_i 사이에 최대값 MAX_i' 을 계산한다.

단계 3. $j \in L_i$ 이거나 $i \in L_j$ 이면서 $i \neq j$ 인 site j에게 MAX_i' 을 보낸다.

단계 4. (1) 보조정리 3.1에서 명시된 2m개의 site들로 부터 값을 받을 때까지 기다린다.

(2) 받은 값들과 MAX_i' 사이에 최대값 MAX_i 을 계산한다.

다음 정리는 이 알고리즘의 타당성을 증명한다.

정리 5.1 : 한 site가 단계 4)의 실행을 끝냈다면, 그 site는 초기에 존재했던 값들 중에 최대값을 소유한다.

증명 : 초기에 존재했던 최대값이 site j 가 가지고 있는 값 V_j라 하자. site i 가 단계 4

의 실행을 끝냈고 V_j가 아닌 다른 값을 최대값으로 가지고 있다고 하자. 또한, i와 연관된 선을 L_i라 할때, 고려되어야 할 두가지 경우는 다음과 같다.

1) $j \in L_i$: 이 경우에, site i가 V_j를 받지 않고 단계 2)의 (1)을 끝낼 수는 없다. 그러므로, 단계 2)의 (2)에서 계산된 MAX_i' 은 V_j이어야만 한다. 단계 4)의 실행을 끝내기 이전에 site i는 MAX_i' 과 받은 값들 중에서 최대값, MAX_i를 계산한다. 이 값은 분명히 V_j일 것이다. 그러므로 위의 가정이 모순임을 알 수 있다. (j = i인 경우도 여기에 준한다.)

2) $j \in L_i$: [공리 1]에 따라, site j는 i에 인접한(incident) m개의 선들 중, 하나에 속해야만 한다. 이 선과 연관된 점이 i_k이라 하자. 첫번째 round에서, site j는 자신의 값 V_j를 i_k에게 보내야 하므로, i_k가 단계 2)에서 계산한 MAX_{i_k}' 은 V_j이어야만 한다. 두번째 round에서 site i_k는 이 값을 i에게 보내야만 한다. 그러므로, site i가 단계 4)에서 계산한 최대값은 V_j이어야만 한다. 그러므로, 위의 가정은 모순이다.

[그림 1]의 Projective Plane에 대한 최대값 계산은 [표 I]에서 보여준다. 각 site i가 가지고 있는 값 V_i를 자신의 site 번호라 가정하였다. 최소값 계산 알고리즘은 각 단계에서 최대값 대신 최소값을 계산한다는 것을 제외하고는 위의 알고리즘과 같다.

6. Lower Bounds

지금까지 논의된 문제들에 있어서, site i가 가진 값 V_i는 일치를 이루기위해 모든 다른 site에서 실행되는 계산에 영향을 미쳐야만 하며, 이것은 두 round 안에 이루어 져야 한다.

표 I

최 대 값 계 산		
Site no.	단 계 2	단 계 4
1	max{1,2,3,4,6} = 6	max{6,6,7,7,7} = 7
2	max{2,1,6,5,4} = 6	max{6,6,7,7,7} = 7
3	max{3,1,7,6,5} = 7	max{7,6,7,7,7} = 7
4	max{4,2,1,7,5} = 7	max{7,6,6,7,7} = 7
5	max{5,4,3,2,7} = 7	max{7,7,7,6,7} = 7
6	max{6,3,2,1,7} = 7	max{7,7,6,6,7} = 7
7	max{7,3,4,5,6} = 7	max{7,7,7,7,7} = 7

각 단계에서 site i 가 통신하는 site들의 최대 갯수를 k라 하자. 두 round의 message 교환이 끝나면 V_i는 기껏해야 (k+1)*k개의 site 에 영향을 미칠 수 있다. 만약 두 round 안에 일치가 이루어 진다면, (k+1)*k = N 이어야 한다. 그것은 k가 $O(\sqrt{N})$임을 의미한다. 만약 각 site마다 동등한 알고리즘을 실행한다면, 모든 알고리즘들에 대한 k의 값은 같다. 그러므로, 일치를 이루기위해 $O(N\sqrt{N})$ message가 필요하며, 본 논문의 알고리즘이 optimal임을 알 수 있다. 그리고 이 통신방법은 c가 round 횟수와 연관된다고 할때, $O(N*N^{1/c})$의 분산일치 알고리즘으로 확장될수 있으리라 기대된다. 이것은 사용된 총 message의 갯수와 round 횟수 사이의 tradeoff의 결과일 것이다.

7. 결 론

본 논문에서 는 시스템 내의 모든 site 들에게 분산되어 있는 정보들을 매개변수로 갖는 함수를 계산하고, 그 계산 결과 가 모든 site 들에게 알려지도록하는 다중 일치 알고리즘을 위한 효과적인 통신방법을 제안했다. 각 site가 시스템내의 다른 모든 site들에게 자신의 정보를 전해주고 다른 모든 site들로부터 정보를 받음으로서 관련된 함수를 계산하는 방법은 한 round의 message 교환에 의해 일치를 이룰수 는 있지만 $O(N^2)$의 message가 필요한 것이 단점이다. 이에 본 논문에서는 Finite Projective Planes를 사용하여 site들이 자신과 관련된 일부 site들에게 만 message을 보냄으로서 두 round의 message 교환과 각 round 마다 $O(N\sqrt{N})$의 message를 사용하여 일치를 이루는 통신방법을 제안했으며 이러한 통신방법에서 필요한 Finte Projective Planes의 각 집합을 구하는 생성 알고리즘을 제안 하였다. 또한, 이 통신방법은 최대, 최소값 계산과 같은 다중 일치 문제에 적용될수 있음을 보였다.

앞으로의 연구방향은 암호학에서 이 통신방법을 적용하여 효율적인 알고리즘을 개발하고자 한다. 다자간의 통신하에서 시스템내의 모든 site로 부터의 정보를 수집해야하는 분야에서 효과적으로 사용될 수 있다고 본다.

* 참 고 문 헌 *

[1] A. A. Albert and R. Sandler, *An Introduction to Finite Projective Planes.* New Tork: Holt, Rinehart and Winston, 1968.

[2] J. Denes and A. D. Keedwell, *Latin Square and Their Applications*, Academic Press, New York, 1974.

[3] T. V. Lakshman and A. K. Agrawala, "Efficient Decentralized Consensus Protocols", IEEE Trans. Software Eng., vol. SE-12, May 1986, pp 600 - 607.

[4] 김 수진, 류재철, "Finite Projective Planes를 이용한 다중일치 알고리즘", 데이타 보호 기반 기술 Workshop, Aug., 1992, pp 219 - 238.