

## 봉쇄가 존재하는 나무형태 대기행렬 네트워크 알고리즘의 이론적 고찰\*

이 효 성\*\*

### Some Theoretical Results on the Algorithm for the Tree-like Queueing Networks with Blocking\*

Hyo-Seong Lee\*\*

#### Abstract

Recently Lee et al.[5] developed an approximation algorithm for the performance evaluation of the open queueing networks with blocking. This algorithm, which solves the exponential queueing networks with general configuration, is developed based on the symmetrical decomposition approach and is reported to have many advantages over the previous algorithms. In addition to being very accurate, this algorithm is reported to be quite simple, pretty fast and solves very general configurations. In this study, we show that if a network has a tree-like configuration, the algorithm developed by Lee et al. always converges to the unique solution. To prove the theoretical results pertaining to the algorithm, some properties associated with symmetrical decomposition approach are exploited. The results obtained in this study such as the proofs of convergence of the algorithm as well as uniqueness of the solution would contribute to the theoretical study for the non-tandem configuration of open queueing networks.

---

\* 본 연구는 95년도 한국과학재단 연구지원(과제번호: 951-1010-012-2)에 의한 결과임.

\*\* 경희대학교 산업공학과

## 1. 서 론

대기행렬 네트워크(queueing network)는 제조 시스템, 컴퓨터 및 통신시스템등 이산형사건 동적시스템(discrete event dynamic system)을 모형화하는 유용한 수학적 도구이다. 만일 대기행렬 네트워크를 구성하는 노우드(node)의 용량이 무한하고 각 노우드에서의 서비스시간이 지수분포를 따른다면, 대기행렬 네트워크는 승법형태(product form)를 취하게 되므로 분석이 간단히 이루어질 수 있다. 그러나 실사회에 존재하는 대부분의 대기행렬 네트워크는 용량이 유한하며 이러한 경우에는 봉쇄(blocking)가 발생하여, 역류가능한 네트워크(reversible network)등 극히 특별한 경우를 제외하고는 승법성질이 보존되지 못한다[8]. 안정상태확률이 승법형태를 취하지 못하므로 용량이 유한한 대기행렬 네트워크의 분석은 극도로 복잡해지며 네트워크를 구성하는 노우드의 개수와 버퍼(buffer)의 크기가 커짐에 따라 정확한 해(exact solution)를 구하기는 현실적으로 불가능해진다. 따라서 용량이 유한한 대기행렬 네트워크를 분석하기 위한 근사적 방법의 개발이 요구되며, 이와 같은 필요성에 의해 많은 연구가 수행되어져 왔다([3],[6],[7],[8]). 개방형 네트워크의 경우 Lee외[5]에 의해 최근에 제안된 알고리즘은 일반적인 형태의 개방형 대기행렬 네트워크를 매우 정확히 분석해 줄 뿐만 아니라 단순성 및 신속성에 있어서도 우수함이 입증되어 향후 대기행렬 네트워크 분석에 유용하게 사용되어 질 가능성이 높을 것으로 보인다. 이와같이 근사적 알고리즘의 개발에 대한 연구는 비교적 활발히 수행되고 있으나 개발된 알고리즘의 수렴성 및 해의 유일성 증명 등 이론적 측면의 연구는 직렬형 네트워크의 경우

[2] 이외에는 매우 미진한 편이다.

본 연구에서는 Lee외[5]에 의해 제안된 알고리즘이 사이클이 존재하지 않고 확정적 경로를 갖는 네트워크(어떤 노우드에서 서비스를 마친 고객의 다음 방문 노우드가 확정적인 네트워크)를 분석할 경우에는 항상 유일한 해에 수렴함을 증명하고자 한다. 확정적 경로를 갖는 네트워크의 형태는 나무모양(tree-like configuration)을 취하게 되며 병합형 네트워크(merge configuration)등 많은 형태의 네트워크가 이에 속한다. 따라서 본 연구는 이미 우수한 것으로 입증된 Lee외에 의해 제안된 알고리즘에 대한 이론적 토대를 제공한다는 점에서 의의가 있을 뿐 만 아니라 직렬형 이외의 알고리즘에 대한 수렴성 및 해의 유일성 증명이라는 점에서 연구의 의의가 있다고 판단된다.

## 2. 모형설명

본 연구에서 분석하고자 하는 모형은 사이클이 존재하지 않는 나무모양의 네트워크로서 총  $M$ 개의 노우드가 있다고 가정한다. 외부와 접해 있는 첫 번째 서어버는 starvation이 발생하지 않는 포화 서어버(saturated server)라고 가정하며 노우드  $j$ 와 연결된 포화 서어버를  $S_0$ 라 표현한다. 포화서어버는 외부로부터 네트워크 내로의 고객의 도착과정을 나타내 주는 역할을 하며 문헌에서는 포화 서어버가 존재하는 모형을 포화 모형(saturated model)이라 부른다.  $j$ 번째 노우드는 비포화 서어버  $S_j$ 와 용량이  $C_j$ (서어버의 위치까지 포함)인 버퍼로 구성된다. 각 노우드에서의 서어버의 수는 한 명이며 서비스시간은 지수분포를 따른다고 가정한다. 서어버  $S_0$ 와  $S_j$ 의 서비

스율은 각각  $\mu_{0j}$  와  $\mu_j$ 로 표시된다. 서버  $S_j$ 에서 서비스를 마친 고객은 확률 1로 다음 서비스를 노우드  $d_j$ 에서 받으며, 네트워크에 사이클이 존재하지 않기 때문에 노우드의 번호는  $j < d_j$ 의 관계가 만족되도록 항상 부여될 수 있다. 마지막 서버  $S_M$ 에서 서비스를 마친 고객은 더 이상 서비스를 받지 않고 시스템을 떠난다. 그림 1은 5개의 포화 서버 및 6개의 비포화 서버로 구성된 대기행렬 네트워크의 예이다. 네트워크 내에 사이클이 존재하지 않고 고객의 경로가 확정적이기 때문에 어떤 노우드와 직접 연결된 downstream node는 두 개 이상 존재할 수 없으나 upstream node는 두 개 이상 존재할 수 있다. 버퍼  $B_j$ 와 직접 연결된 upstream server의 색인집합(index set)을  $U_j$ 라 표현하고 집합  $U_j$ 에 속한 원소의 수를  $N_j$ 라 표현한다. 예를 들면 그림 1에서  $U_2 = \{02\}$ ,  $N_2 = 1$ ,  $U_4 = \{04, 1, 2, 3\}$ ,  $N_4 = 4$ 과 같이 주어진다.

봉쇄가 존재할 경우 사용되는 봉쇄체제(blocking mechanism)는 전달봉쇄(transfer blocking, blocking after service), 서비스봉쇄(service blocking, blocking before service), 기각봉쇄(rejection blocking, repetitive-service blocking)로 분류될 수 있다. 이중 전달 봉쇄는 서비스를 받고 난 뒤 봉쇄가 일어나는 경우로서 고객이 노우드  $i$ 에서 서비스를 받고 난 뒤 목적지인 노우드  $j$ 에 자리가 없으면 자리가 생길 때까지 노우드  $i$ 의 서버를 점유하고 있게 된다. 서비스봉쇄는 서비스를 받기 전 봉쇄가 발생하는 경우로서 서비스를 받기 직전 목적노우드에 자리가 없으면 자리가 생길 때까지 봉쇄당하는 현상이다. 기각봉쇄는 서비스를 받고 난 후 목적 노우드에 자리가 없으면 자리가 생길 때까지 서비스를 반복하여 받는 봉쇄체제이다. 본 연구에서는 봉쇄체제로 전달봉쇄가 사용된다고 가정한다. 그러나 포화 서버에서는 전달봉쇄 대신 기각봉쇄가 사용된다고 가정할 수도 있다. 이와 같이 가정하면 본 연구에서의 모형은 고객이 외부로부터 포아송과정에 의해 들어오며 버퍼가 차 있을 때 도착하는

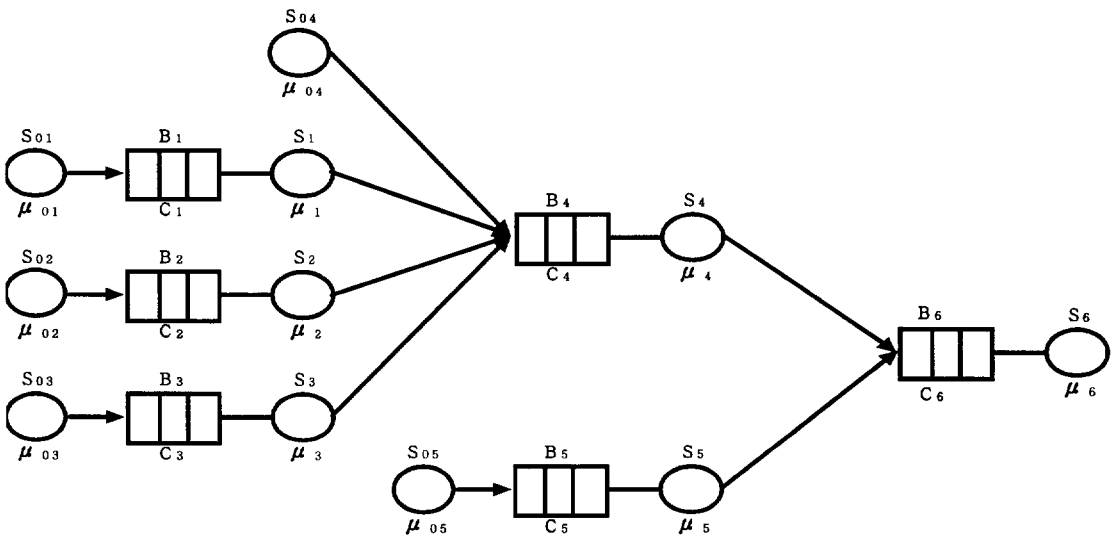


그림 1. 5개의 포화 서버와 6개의 비포화 서버로 구성된 대기행렬 네트워크

고객은 유실되는 기존 연구에서의 모형을 표현할 수 있다.

본 연구에서는 특정 노우드와 직접 연결된 upstream node가 두 개 이상 존재할 수 있으므로 한 노우드에 의해 다수의 노우드가 동시에 봉쇄당하는 다중봉쇄(multiple blocking)가 발생할 수 있다. 다중봉쇄가 발생하는 경우에는 목적지에 자리가 생겼을 때 봉쇄가 해제되는 순서를 정할 필요가 있으며 본 연구에서는 먼저 봉쇄당한 노우드부터 우선적으로 봉쇄가 해제된다고 (first-blocked-first-enter basis)가정한다.

봉쇄가 존재하는 대기행렬 네트워크를 분석하기 위한 근사적 방법은 과거 많은 연구가 수행되어져 왔다([3],[6],[7],[8]). 근사적 방법은 대기행렬 네트워크를 구성하는 노우드들을 각각 독립적인 노우드로 간주하여 분석하는 분해법(decomposition method)에 대부분 기초하고 있다. 분해법은 i)네트워크를 하부시스템으로 분해, ii)미지의 매개변수 값을 결정하기 위한 방정식 수립, iii) 방정식의 해를 구하기 위한 알고리즘 개발의 3 단계 과정으로 구성되어 있다. 본 연구에서 연구 대상으로 하는 Lee외의 알고리즘[5]도 분해법에 기초하고 있다. Lee외의 알고리즘은 사이클이 존재할 수 있는 일반형 대기행렬 네트워크의 분석을 위해 개발되었으나 나무구조를 갖지 않는 일반형 네트워크 분석의 경우에는 알고리즘의 이론적 규명이 쉽지 않은 것으로 보인다. 본 연구의 3장에서는 나무구조를 가진 네트워크를 분석할 경우에 대한 Lee외의 알고리즘을 각 단계별로 설명하며, 본 연구의 4장에서는 알고리즘의 수렴성 및 해의 유일성 등 이론적 특성을 규명하고자 한다.

### 3. 알고리즘 설명

#### 3.1. 주어진 네트워크를 하부시스템으로 분해

Lee외[5]의 알고리즘에서는 주어진 네트워크를 노우드의 갯수 만큼의 하부시스템(subsystem)으로 분해한다. 각각의 하부시스템은 포화 서버인 upstream server와 비포화 서버인 downstream server가 버퍼에 의해 연결되어 있는 형태를 취한다. 이때 포화 서버와 비포화 서버는 서로 쌍대적 역할(dual role)을 하고 있으며 포화 서버는 starvation이 발생하지 않고 비포화 서버는 봉쇄가 발생하지 않는 특징을 가지고 있다.  $j$ 번째 하부시스템  $T(j)$ 는 downstream server  $S_d(j)$ 와  $N_j$ 개의 포화서버  $S_i(j), i \in U_j$ 로 구성되어 있으며  $S_d(j)$ 의 서비스율은  $\mu_d(j)$ ,  $S_i(j)$ 의 서비스율은  $\mu_i(j)$ 로 표시된다. 하부시스템  $T(j)$ 는 원 네트워크의 버퍼  $B_j$ 에 있는 고객의 흐름을 근사화한다. 하부시스템  $T(j)$ 의 버퍼는  $B(j)$ 로 표현되며  $B(j)$ 의 용량은  $B_j$ 와 같은  $C_j$ 로 주어진다. 하부시스템  $T(j)$ 에서  $S_i(j)$ 는 원 네트워크에서의 노우드  $i$ 와 그 upstream 노우드를 나타내며,  $S_d(j)$ 는 원 네트워크에서의 서버  $S_j$ 와 그 downstream 노우드를 나타낸다. 그림 2은 그림 1의 네트워크가 어떻게 하부시스템으로 분해되는지 보여준다.

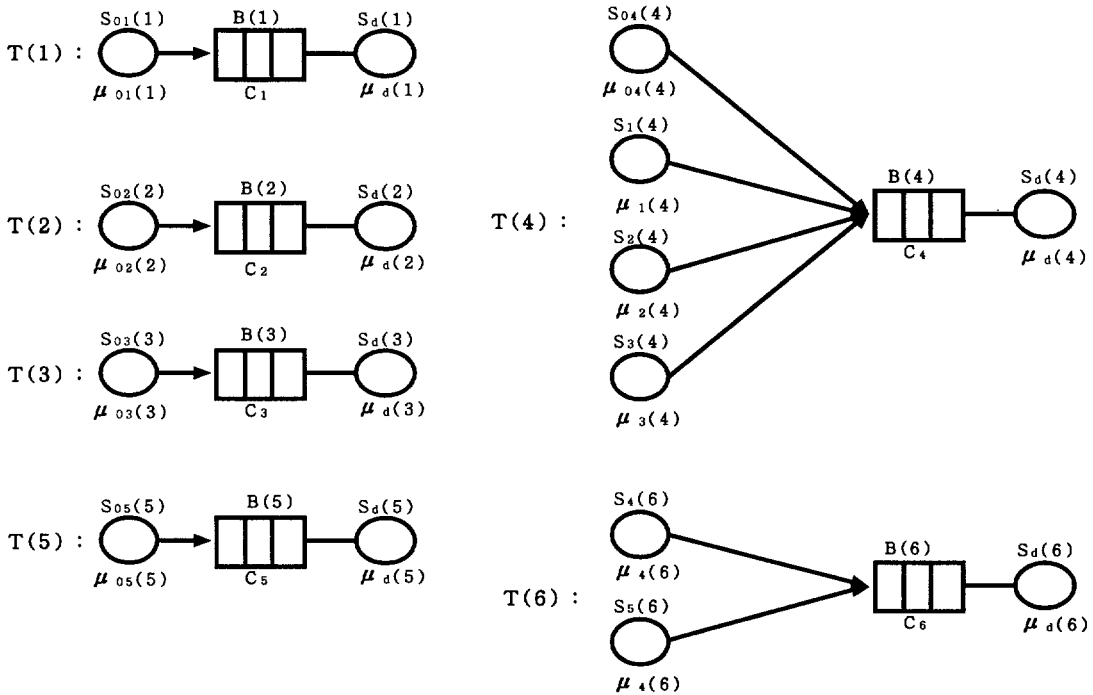


그림 2. 하부시스템으로의 분해

3.2 하부시스템의 분석

Lee 외의 알고리즘에서는 각 하부시스템에서의 upstream server와 downstream server의 서비스 시간을 모두 지수분포로 특성화하고, 각 하부시스템에 대해서는 근사분석이 아닌 정확한 분석 (exact analysis)을 수행한다. 하부시스템  $T(j)$  (그림 3)을 정확히 분석하기 위하여 Lee와 Pollock[4]의 분석기법을 이용한다. 이를 위하여  $T(j)$ 의 상태를  $B(j)$ 에 있는 고객 및  $S_d(j)$ 에 의해 봉쇄당하고 있는 고객의 수로 정의한다. 따라서  $C_j + n$ 은  $n$ 개의 upstream server가  $S_d(j)$ 에 의해 봉쇄당하고 있는 상태를 의미한다.  $T(j)$ 의 분석을 위하여 다음과 같은 기호를 정의한다.

$P(n; j) = T(j)$ 가 상태  $n$ 에 있을 안정상태 확률,  $n = 0, \dots, C_j + N_j$ .

$b_i(n; j) = S_i(j)$ 를 포함하여  $n$ 명의 고객이  $S_d(j)$ 에 의해 봉쇄당하고 있을 확률,  $n = 1, \dots, N_j$ .

$\bar{b}_i(j) = S_i(j)$ 가  $S_d(j)$ 에 의해 봉쇄당하고 있을 확률,

$\bar{b}_i(j) =$ 버퍼  $B(j)$ 는 차고  $S_i(j)$ 는  $S_d(j)$ 에 의해 봉쇄당하고 있지 않을 확률,

$P_s(j) =$ 서비스 종료시점에  $S_d(j)$ 가 starved 될 확률,

$P_{bi}(n; j) = S_i(j)$ 의 서비스 종료시점에  $n$ 명의 서버가  $S_d(j)$ 에 의해 봉쇄당하고 있을 확률,  $n = 0, \dots, N_j - 1$ . ( $n = 0$ 의 경우에는 서비스 종료시점에 버퍼는 차고 봉쇄중인 서버는 존재하지 않을 확률)

위와 같이 기호를 정의하면  $T(j)$ 의 안정상태 확률,  $P(n; j)$ 는 Lee와 Pollock[4]의 결과로부터

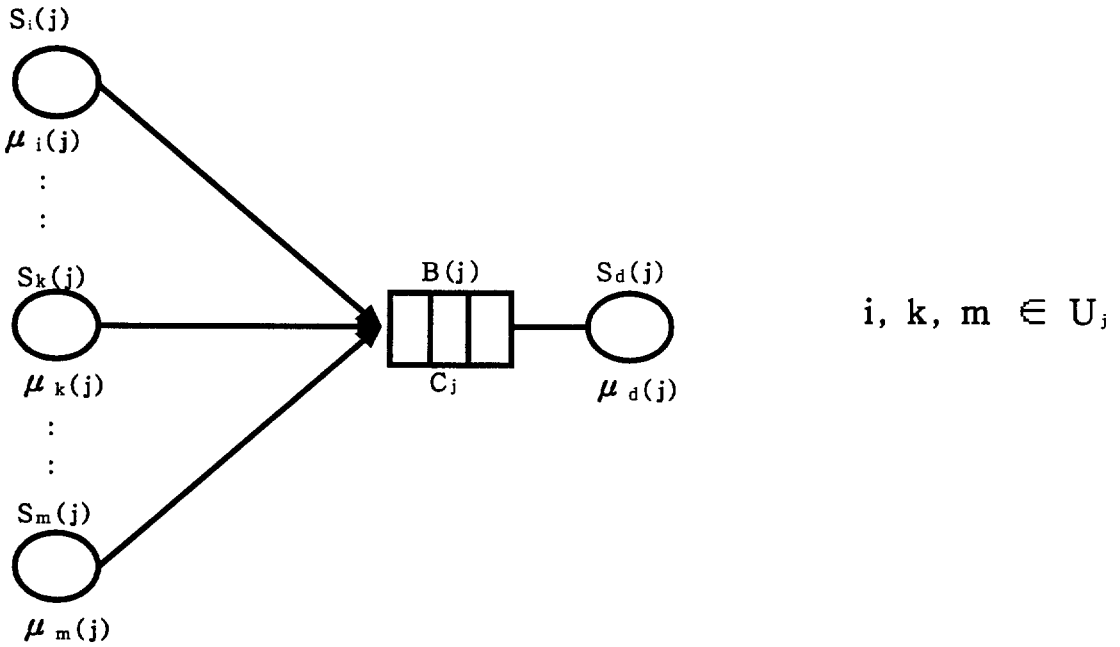
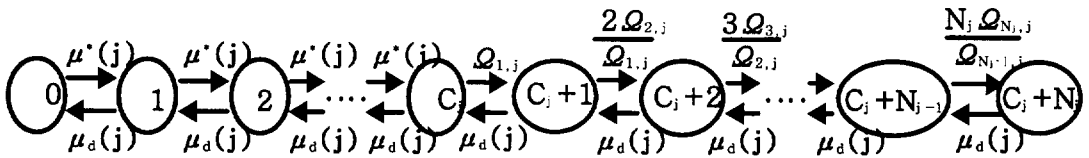


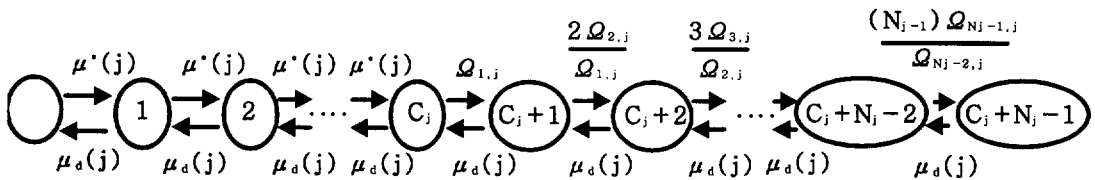
그림 3. 하부시스템 T(j)



여기에서  $\mu^*(j) = \sum_{i \in U_j} \mu_i(j)$ 이고  $\Omega_{i,j}$ 는  $\{\mu_k(j) : k \in U_j\}$ 에 속하는

서로 다른  $i$ 개의  $\mu_k(j)$ 항의 곱의 합이다.

그림 4.  $S_0(j)$ 에서 전달봉쇄가 사용될 경우의 전이율 다이어그램



여기에서  $\mu^*(j) = \sum_{i \in U_j} \mu_i(j)$ 이고  $\Omega_{i,j}$ 는  $\{\mu_k(j) : k \in U_j, k \neq 0\}$ 에 속하는

서로 다른  $i$ 개의  $\mu_k(j)$ 항의 곱의 합이다.

그림 5.  $S_0(j)$ 에서 기각봉쇄가 사용될 경우의 전이율 다이어그램

그림 4나 그림 5의 출생-사멸 과정(birth and death process)의 해를 구함으로써 얻을 수 있다. 그림 4는  $S_{0j}(j)$ 에서 전달봉쇄가 사용될 경우의 전이율 다이어그램이고 그림 5는  $S_{0j}(j)$ 에서 기각봉쇄가 사용될 경우의 전이율 다이어그램을 나타낸다.

예를 들어 그림 1에서  $U_4 = \{04, 1, 2, 3\}$ 이다.

만일  $S_{04}(4)$ 에서 전달봉쇄가 사용될 경우에는

$$\begin{aligned} \Omega_{3,4} &= \mu_{04}(4)\mu_1(4)\mu_2(4) + \mu_{04}(4)\mu_1(4)\mu_3(4) \\ &\quad + \mu_{04}(4)\mu_2(4)\mu_3(4) + \mu_1(4)\mu_2(4)\mu_3(4) \\ \Omega_{2,4} &= \mu_{04}(4)\mu_1(4) + \mu_{04}(4)\mu_2(4) \\ &\quad + \mu_{04}(4)\mu_3(4) + \mu_1(4)\mu_2(4) \\ &\quad + \mu_1(4)\mu_3(4) + \mu_2(4)\mu_3(4), \end{aligned}$$

이고, 만일  $S_{04}(4)$ 에서 기각봉쇄가 사용될 경우에는 다음과 같이 표현된다.

$$\begin{aligned} \Omega_{2,4} &= \mu_1(4)\mu_2(4) + \mu_1(4)\mu_3(4) \\ &\quad + \mu_2(4)\mu_3(4), \\ \Omega_{3,4} &= \mu_1(4)\mu_2(4)\mu_3(4). \end{aligned}$$

일단 안정상태확률이 구해지면 봉쇄가 발생할 확률은 다음과 같이 구하여 질 수 있다.

$$b_i(n; j) = \frac{\mu_i(j)\Omega_{n-1,i}^i}{\Omega_{n,j}} P(C_j + n; j) \quad n=1, 2, \dots, N_j.. \quad (1)$$

여기에서  $\Omega_{n-1,i}^i$ 는  $S_{0j}(j)$ 에서 전달봉쇄가 사용될 경우에는  $\{\mu_k(j) : k \in U_j, k \neq i\}$ 에 속하는 서로 다른  $n-1$ 개의  $\mu_k(j)$ 항의 곱의 합으로 정의되고,  $S_{0j}(j)$ 에서 기각봉쇄가 사용될 경우에

는  $\{\mu_k(j) : k \in U_j, k \neq 0j, k \neq i\}$ 에 속하는 서로 다른  $n-1$ 개의  $\mu_k(j)$ 항의 곱의 합으로 정의된다. 예를 들어 그림 1의  $S_{04}(4)$ 에서 전달봉쇄가 사용되면  $\Omega_{2,4}^3 = \mu_{04}(4)\mu_1(4) + \mu_{04}(4)\mu_2(4) + \mu_1(4)\mu_2(4)$ 과 같이 표현되고,  $S_{04}(4)$ 에서 기각봉쇄가 사용되면  $\Omega_{2,4}^3 = \mu_1(4)\mu_2(4)$ 과 같이 표현된다.

$b_i(n; j)$ 가 구하여 지면 이로부터 서버버  $S_i(j)$ 가 서버버  $S_d(j)$ 에 의해 봉쇄당하고 있을 확률은 다음과 같이 계산되어질 수 있다.

$$b_i(j) = \sum_{n=1}^{N_j} b_i(n; j). \quad (2)$$

버퍼  $B(j)$ 는 차고  $S_i(j)$ 는  $S_d(j)$ 에 의해 봉쇄당하고 있지 않을 확률은  $b_i(j)$ 로부터 다음과 같이 구하여 진다.

$$\bar{b}_i(j) = \sum_{n=0}^{N_j} P(C_j + n; j) - b_i(j) \quad (3)$$

미지의 매개변수 값을 결정하기 위한 방정식을 수립하기 위해서는 서비스 종료시점에서의 starvation 확률과 봉쇄확률이 구하여 저야만 하며 이들 확률은 다음과 같이 구하여 질 수 있다.

$$P_s(j) = \frac{P(1; j)}{1 - P(0; j)}. \quad (4)$$

$$P_{bi}(n; j) = \frac{P(C_j + n; j) - b_i(n; j)}{1 - b_i(j)}, \quad n=0, 1, 2, \dots, N_j - 1. \quad (5)$$

3.3 미지의 매개변수 값을 결정하기 위한 방정식 수립

3.2절에서는 매개변수  $\mu_i(j)$ 와  $\mu_d(j)$ 의 값을 모두 알고 있다는 가정 하에서 분석을 수행하였다. 그러나 하부시스템의 매개변수 값  $\mu_i(j)$ ,  $i \in U_j$ ,  $j=1, 2, \dots, M$ 과  $\mu_d(j)$ ,  $j=1, 2, \dots, M$ 은 결정되어야 할 미지의 값이다. 따라서 모두  $M + \sum_{j=1}^M N_j$ 의 값을 결정하여야 만 하며 이를 위해서는  $M + \sum_{j=1}^M N_j$  개의 방정식이 필요하다. 그러나 포화서버  $S_{0j}$ 는 원 네트워크에서 upstream 노드를 가지고 있지 않으므로 하부시스템 T(j)에서의  $S_{0j}(j)$ 는 원 네트워크의  $S_{0j}$ 를 나타낸다. 또한 downstream node를 가지고 있지 않은  $S_M$ 도 하부시스템에서  $S_d(M)$ 로 대변된다. 따라서 이러한 서버에 대해서는  $\mu_{0j}(j) = \mu_{0j}$ ,  $\mu_d(M) = \mu_M$ 의 식이 성립되므로 우리는 포화서버의 수+1 만큼 매개변수의 값을 미리 알고있다.

이제 나머지 매개변수의 값을 결정하기 위한 방정식 체계를 유도하여 보자. 첫 번째 방정식 체계는 downstream server  $S_d(j)$ 의 서비스 과정과 관련하여 유도되어 질 수 있다. 하부시스템 T(j)에서  $S_d(j)$ 는 원 네트워크에서의 서버  $S_j$ 와  $S_j$ 의 downstream 노드를 나타내므로  $S_d(j)$ 의 서비스 시간은 원 네트워크에서  $S_j$ 가 서비스를 시작한 시점으로부터 downstream 노우드로 고객을 이동시킨 시점까지를 대변한다. 이 기간은  $S_j$ 에서의 서비스 시간과 확률적으로 발생 가능한 봉쇄시간으로 구성된다. 봉쇄는  $S_j$ 에서 서비스가 종료됐을 때 목적 노우드  $d_j$ 의

버퍼가 가득차 있을 경우에 발생한다. 목적 노우드  $d_j$ 를  $m$ 이라 하면 서버  $S_j$ 가 노우드  $m$ 에 의해 봉쇄당하는 현상은 하부시스템 T(m)에서 서버  $S_j(m)$ 이 서버  $S_d(m)$ 에 의해 봉쇄당하는 현상으로 대변된다. 따라서  $S_j$ 에서 서비스를 마친 고객이 서비스가 종료된 시점에서 n명의 다른 고객이 목적 노우드 m에 의해 이미 봉쇄당하고 있음을 불확률은  $P_{bj}(n, m)$ 으로 근사화 될 수 있다. 그러므로  $S_d(j)$ 에서의 평균 서비스시간은 다음과 같이 주어진다.

$$\frac{1}{\mu_d(j)} = \frac{1}{\mu_j} + \sum_{n=0}^{N_j-1} P_{bj}(n, m) (n+1) \frac{1}{\mu_d(m)} \tag{6}$$

위에서 유도된 방정식체계는 M개의 식밖에 포함하고 있지 않으므로 매개변수의 값을 결정하기 위해서는 또 다른 방정식 체계의 유도가 필요하다. 두 번째 방정식 체계는 upstream server  $S_i(j)$ 의 서비스 과정과 관련하여 유도되어 질 수 있다. 하부시스템 T(j)에서  $S_i(j)$ 는 원 네트워크에서의 서버  $S_i$ 와  $S_i$ 의 upstream 노우드를 나타내므로  $S_i(j)$ 의 서비스 시간은 원 네트워크의  $S_i$ 가 고객을 downstream 버퍼  $B_j$ 로 이동시킨 시점으로 부터 다음 고객의 서비스가 완료되는 시점까지로 대변될 수 있다. 따라서  $\mu^*(i) = \sum_{k \in U_i} \mu_k(i)$ 라 놓으면  $S_i(j)$ 의 평균 서비스 시간은 다음과 같이 표현된다.

$$\frac{1}{\mu_i(j)} = \frac{1}{\mu_i} + P_s(i) \frac{1}{\mu^*(i)}. \tag{7}$$



식 (6)과 (7)로 구성된 방정식 체계를 SE1라 하면 SE1에 포함된 방정식의 개수는 미지수의 개수와 일치함을 쉽게 확인할 수 있다. 따라서 미지의 매개변수 값은 SE1를 풀음으로써 구하여 질 수 있다. 또한 식 (6)과 (7)을 만족하는 해는 흐름보존법칙(flow of conservation law)을 만족함을 증명할 수 있다[5].  $X_d(i)$ 를  $S_d(i)$ 의 산출율(throughput),  $X_i(j)$ 를  $S_i(j)$ 의 산출율이라 정의하면 하부시스템 간의 흐름이 보존되기 위한 흐름보존법칙은 다음 식과 같이 작성될 수 있다.

$$X_i(j) = X_d(i), \quad i \in U_j, \quad j = 1, 2, \dots, M. \quad (8)$$

만일  $X_i(j)$ 와  $X_d(i)$ 의 값을 구할 수 있으면  $X_i(j)$ 와  $\mu_i(j)$ , 그리고  $X_d(i)$ 와  $\mu_d(i)$ 의 다음 관계식에 의하여  $\mu_i(j)$ 와  $\mu_d(i)$ 의 값도 구할 수 있게된다.

$$X_i(j) = \frac{1}{\frac{1}{\mu_i(j)} + \sum_{n=0}^{N_i-1} P_{bi}(n, j) (n+1) \frac{1}{\mu_d(j)}}. \quad (9.A)$$

$$X_d(i) = \frac{1}{\frac{1}{\mu_d(i)} + P_s(i) \frac{1}{\mu^*(i)}}. \quad (9.B)$$

따라서 식 (8)도 식 (6), (7)과 더불어 미지의 매개변수의 값을 결정하기 위한 방정식체계가 이용될 수 있다. 그러나 매개변수의 값을 결정하기 위해서는 3개의 방정식체계가 모두 필요하지 않고 3개중 2개만 이용하면 된다. 이중 식 (6)과 (7)로 구성된 방정식체계를 SE1, 식 (6)과 (8)로 구성된 방정식체계를 SE2, 그리고 식 (7)과

(8)로 구성된 방정식 체계를 SE3라하면 매개변수의 값은 SE1, SE2, SE3중 어느 한 방정식체계를 풀음으로써 결정할 수 있으므로 이에 따르게 종류의 알고리즘을 얻을 수 있다. 그러나 SE1, SE2, SE3는 동일한 해를 산출해 줌을 보일 수 있으며[5], 따라서 하부시스템으로의 분해 방법과 하부시스템의 분석방법이 동일하다면 해를 구하는데 소요되는 시간의 차이는 존재하나 세 종류의 알고리즘이 모두 동일한 해를 산출해 준다.

개방형 대기행렬 네트워크의 분석을 위한 기존의 근사적 알고리즘은 대부분 방정식체계 SE2에 기초하여 개발되었다[1][4][9][10]. 그러나 Lee의[5]에 의해 개발된 알고리즘은 일반형 네트워크로는 최초로 SE1에 기초한 알고리즘이며 이들의 연구에 의하면 SE1에 기초한 알고리즘이 SE2에 기초한 알고리즘에 비하여 속도 면에서 우수하다고 보고되어 있다.

### 3.4 방정식의 해를 구하기 위한 알고리즘 개발

3.3절에서 유도된 방정식체계 SE1는 고정점 문제(fixed point problem)형태를 취하고 있다. 따라서 방정식의 해를 구하기 위해서는 반복적 기법(iterative method)의 사용이 불가피하다. SE1의 해를 구하기 위한 Lee의 알고리즘은 매 반복단계가 다시 두 단계로 구성되어 있다. 첫 번째 단계는 전진단계로서  $P_s(j)$ 와  $\mu_j(d_j)$ 의 값을 새로이 계산 해준다. 이를 위하여 가장 최근에 구하여진  $\mu_i(j)$ 값과  $\mu_d(j)$ 값을 이용하여 3.2에 소개된 방법에 따라 하부시스템 T(j)를 분석하고  $P_s(j)$ 값을 구한다.  $P_s(j)$ 값이 구해진 후에는 식 (7)을 이용하여  $\mu_j(d_j)$ 값을 새로이 계산한다. 두 번째 단계는 후진단계로서 가장

최근에 구하여진  $\mu_i(j)$  값과  $\mu_d(j)$  값을 이용하여 하부시스템 T(j)를 분석하여  $P_{bi}(n, j)$  값을 새로이 계산해 준다.  $P_{bi}(n, j)$  값이 새로이 구하여지면 식 (6)을 이용하여  $\mu_d(k)$ ,  $k \in U_j$ 을 새로이 구한다. 매개변수의 초기값으로는 봉쇄나 starvation이 발생하지 않을 경우의 매개변수 값을 이용한다. 각 단계에서 하부시스템 T(M)의 경우에는  $P_s(M)$  값을 계산할 필요가 없으며, 버퍼 B<sub>j</sub>에 연결된 upstream 서어버가 포화서어버 뿐 일 경우에는  $P_{bi}(n, j)$  값을 계산할 필요가 없으므로 이를 이용하면 각 단계에서의 계산량을 줄일 수 있다. 이러한 반복과정을 본 알고리즘에서는 전 단계에서의 매개변수의 값과 이번 단계에서 새로이 계산된 매개변수의 값이 거의 일치할 때까지 반복 수행한다. 이상의 과정은 다음과 같이 정리될 수 있으며 본 연구에서는 나무구조의 네트워크를 분석하기 위한 Lee와의 알고리즘을 알고리즘(L)이라 부르도록 한다.

### 알고리즘(L)

#### 초기단계

매개변수의 초기값을 다음과 같이 놓는다.

$$\begin{aligned} \mu_d(j) &= \mu_j, \quad j=1, \dots, M, \\ \mu_{0j}(j) &= \mu_{0j}, \quad j=1, \dots, M \end{aligned}$$

#### 반복단계

1.  $j=1, 2, \dots, M$ 에 대하여
  - 1.1: 식(4)를 이용하여  $P_s(j)$ 를 계산한다.
  - 1.2: 식(7)를 이용하여  $\mu_j(d_j)$ 를 계산한다.
2.  $j=M, \dots, 2, 1$ 에 대하여
  - 2.1: 식 (5)을 이용하여  $P_{bi}(n, j)$ 를 구한다.  $n=0, 1, \dots, N_j-1, i \in U_j$ .

2.2: 식 (6)을 이용하여  $\mu_d(i)$ ,  $i \in U_j$ 을 구한다

#### 수렴단계

수렴조건이 발생했으면 알고리즘을 중단한다.  
그렇지 않으면 반복단계로 간다.

사이클이 존재하지 않는 네트워크의 분석을 위한 위 알고리즘은 주어진 네트워크를 하부시스템으로 분해하는 과정과 하부시스템을 분석하는 과정이 본질적으로 Lee와 Pollock의 알고리즘[4]과 동일하다. 따라서 포화 서어버에서 각각 봉쇄를 사용할 경우에는 Lee와 Pollock의 알고리즘과 동일한 해를 산출해 준다. 그러나 Lee와 Pollock의 알고리즘은 SE2에 기초한 알고리즘인 반면 위 알고리즘은 SE1에 기초하고 있으므로 해를 구하는 속도에는 차이가 있으며 일반적으로 위 알고리즘은 Lee와 Pollock의 알고리즘에 비하여 속도면에서 15%가량 우수한 것으로 보고되어 있다[5]. 또한 Lee와 Pollock의 알고리즘은 알고리즘의 수렴성 및 유일성 증명에 어려움이 따르나 알고리즘(L)은 4장에서 볼 수 있듯이 알고리즘의 이론적 규명이 가능하다.

## 4. 알고리즘의 이론적 고찰

알고리즘 개발이라는 실제적 측면의 연구도 중요하나 개발된 알고리즘과 관련된 이론적 측면의 연구도 매우 중요하다. 매개변수의 값을 결정하기 위하여 수립된 방정식체계의 해가 존재한다면 그 해가 유일한지의 여부, 그리고 해가 유일하다면 제안된 알고리즘에 의해 항상 해를

얻을 수 있는 지(해의 존재성 및 알고리즘의 수렴성)의 여부를 증명함으로써 알고리즘은 완벽성을 갖추게 된다. 본 장에서는 이에대한 증명이 주어진다. 증명방법은 알고리즘이 반복단계를 거듭함에 따라 매개변수 값들은 단조증가 혹은 단조감소하고 이 값들이 상한 혹은 하한을 가짐을 보임으로써 알고리즘의 수렴성을 증명하게 되고, SE1에 서로 다른 두개 이상의 해가 존재할 경우 모순이 발생함을 보임으로써 해의 유일성을 증명하게 된다.

4.1 하부시스템 T(j)의 특성

알고리즘의 특성을 이론적으로 규명하기 위해서는 하부시스템이 지니고 있는 세부적 특성들이 우선 규명되어야만 한다. 본 절에서는 알고리즘의 이론적 규명에 이용될 하부시스템 T(j) (그림 3 참조)와 관련된 몇가지 성질들이 증명된다. 이들 성질은 S<sub>0</sub>에서의 봉쇄체제(전달봉쇄 혹은 기각봉쇄)에 관계없이 성립하나 본 절에서는 전달봉쇄를 가정하여 증명한다.

성질 1

3.2에서 정의된 Ω<sub>k,j</sub>을 Ω<sub>k</sub>로 약식표기 하면 임의의 양의 정수 n, m(n + m ≤ N<sub>j</sub>)에 대하여 다음이 성립한다.

$$\frac{\Omega_{n+m}}{\Omega_n} \text{은 } \mu_i(j), i \in j \text{에 대하여 증가함수이다.}$$

증명

$\frac{\Omega_{n+1}}{\Omega_n}$  이  $\mu_i(j)$ 에 대하여 증가함수이면  $\frac{\Omega_{n+m}}{\Omega_n}$ 도  $\mu_i(j)$ 에 대하여 증가함수이므로 성질 1을 증명하기 위해서는  $\frac{\Omega_{n+1}}{\Omega_n}$  이  $\mu_i(j)$ 에 대하여 증가함수

임을 증명하면 된다. 이를 위하여  $\mu_i(j)$ 의 값이  $\mu_i(j) + \Delta$ 로 Δ만큼 증가한 반면  $\mu_k(j), k \neq i, k \in U_j$ 의 값은 변함이 없다고 가정하자.  $\mu_i(j)$ 의 값이 Δ만큼 증가함에 따라 Ω<sub>n</sub>과 Ω<sub>n+1</sub>의 값은 각각 ΔΩ<sub>n-1</sub><sup>i</sup>와 ΔΩ<sub>n</sub><sup>i</sup>만큼 증가함을 보일 수 있다. 따라서  $\frac{\Omega_{n+1}}{\Omega_n}$  이  $\mu_i(j)$ 에 대하여 증가함수임을 증명하기 위하여 다음 부등식이 성립함을 보이기만 하면 된다.

$$\frac{\Omega_{n+1} + \Delta\Omega_n^i}{\Omega_n + \Delta\Omega_{n-1}^i} \geq \frac{\Omega_{n+1}}{\Omega_n}. \tag{10}$$

부등식 (10)이 성립하기 위한 필요충분 조건은  $\frac{\Omega_{n+1}}{\Omega_n} \leq \frac{\Omega_n^i}{\Omega_{n-1}^i}$ , 즉 Ω<sub>n+1</sub>Ω<sub>n-1</sub><sup>i</sup> ≤ Ω<sub>n</sub>Ω<sub>n</sub><sup>i</sup> 이다. 그런데 Ω<sub>n</sub>Ω<sub>n</sub><sup>i</sup>에는 Ω<sub>n+1</sub>Ω<sub>n-1</sub><sup>i</sup>에 있는 모든 항이 포함되어 있으며 항의 개수도 Ω<sub>n+1</sub>Ω<sub>n-1</sub><sup>i</sup>에 있는 항의  $\frac{n+1}{n}$  배가 됨을 확인할 수 있다. 이상의 사실과  $\mu_k(j), k \in U_j$ 의 값이 모두 양수라는 사실로부터 Ω<sub>n+1</sub>Ω<sub>n-1</sub><sup>i</sup> ≤ Ω<sub>n</sub>Ω<sub>n</sub><sup>i</sup>이 성립함을 알 수 있으며 따라서  $\frac{\Omega_{n+1}}{\Omega_n}$  이  $\mu_i(j)$ 에 대하여 증가함수임을 알 수 있다. ■

성질 1로 부터 다음에 설명되는 성질 2가 바로 유도될 수 있다.

성질 2

임의의 정수 n, m (n ≥ 0, m > 0)에 대하여 다음이 성립한다.

$\frac{P(n+m:j)}{P(n:j)}$  은  $\mu_i(j)$ 가 증가함에 따라 증가한다.

**성질 3**

$$\sum_{n=k}^{N_j-1} P_{bm}(n:j), \quad m \in U_j, \quad k=0, \dots, N_j-1$$

은  $\mu_i(j), i \in U_j$ 가 증가함에 따라 증가한다.

**증명**

$\overline{b}_m(n:j)$ 을 서버  $S_m(j)$ 을 제외한  $n$ 명의 upstream 서버가  $S_d(j)$ 에 의해 봉쇄당하고 있을 확률이라 정의하자. 그러면 그림 4의 전이율 다이어그램과 식 (1)로부터  $\overline{b}_m(n:j)$ 은 다음과 같이 표현된다.

$$\begin{aligned} \overline{b}_m(n:j) &= P(C_{j+n:j}) - b_m(n:j) \\ &= P(C_{j+n:j}) - \frac{\mu_m(j) \Omega_n^{m-1}}{\Omega_n} P(C_{j+n:j}) \\ &= P(C_{j+n:j}) \left\{ 1 - \frac{\mu_m(j) \Omega_n^{m-1}}{\Omega_n} \right\} \\ &= P(C_{j:j}) \frac{n! \Omega_n}{\mu_d(j)^n} \frac{\Omega_n - \mu_m(j) \Omega_n^{m-1}}{\Omega_n} \\ &= P(C_{j:j}) \frac{n! \Omega_n^m}{\mu_d(j)^n} \end{aligned} \tag{11}$$

마지막 줄의 유도에는  $\Omega_n^m = \Omega_n - \mu_m(j) \Omega_n^{m-1}$ 의 관계가 이용되었다. 이제 식 (5)와 (11)로부터 하단 식을 얻는다.

하단 식의 분모에 속해있는 항을 살펴보면 각 항이  $\mu_i(j)$ 에 대하여 감소 함수임을 성질 1과 성질 2로부터 확인할 수 있다. 따라서 성질 3은 성립한다. ■

**성질 4**

$$\sum_{n=0}^{N_j-1} P_{bm}(n:j)(n+1), \quad m \in U_j, \quad \mu_i(j), i \in U,$$

가 증가함에 따라 증가한다.

**증명**

$k=0, \dots, N_j-1$ 의 모든  $k$ 값에 대하여  $\sum_{n=k}^{N_j-1} P_{bm}(n:j)$ 이  $\mu_i(j)$ 에 대하여 증가함수이면 성질 4는 성립한다. 성질 3으로부터  $\sum_{n=k}^{N_j-1} P_{bm}(n:j)$ 은  $\mu_i(j)$ 에 대하여 증가함수이므로 성질 4는 성립된다. ■

$$\begin{aligned} \sum_{n=k}^{N_j-1} P_{bm}(n:j) &= \frac{\sum_{n=k}^{N_j-1} \overline{b}_m(n:j)}{1 - b_m(j)} \\ &= \frac{\sum_{n=k}^{N_j-1} \overline{b}_m(n:j)}{\sum_{r=0}^{C_j} P(r:j) + \sum_{r=1}^{N_j-1} \overline{b}_m(r:j)} \\ &= \frac{P(C_{j:j}) \sum_{n=k}^{N_j-1} \frac{n! \Omega_n^m}{\mu_d(j)^n}}{\sum_{r=0}^{C_j} P(r:j) + P(C_{j:j}) \sum_{r=1}^{k-1} \frac{r! \Omega_r^m}{\mu_d(j)^r} + P(C_{j:j}) \sum_{r=k}^{N_j-1} \frac{r! \Omega_r^m}{\mu_d(j)^r}} \\ &= \frac{\sum_{r=0}^{C_j} \frac{P(r:j)}{P(C_{j:j}) \sum_{n=k}^{N_j-1} \frac{n! \Omega_n^m}{\mu_d(j)^n}} + \sum_{r=1}^{k-1} \frac{\frac{r! \Omega_r^m}{\mu_d(j)^r}}{\sum_{n=k}^{N_j-1} \frac{n! \Omega_n^m}{\mu_d(j)^n}} + 1 \end{aligned}$$

성질 5

$P(0:j)$ 은  $\mu_i(j)$ ,  $i \in U_j$ 가 증가함에 따라 감소한다.

증명

$P(0:j) + \sum_{n=1}^{C_i+N_j} P(n:j) = 1$ 로 부터 다음식이 성립한다.

$$1 + \sum_{n=1}^{C_i+N_j} \frac{P(n:j)}{P(0:j)} = \frac{1}{P(0:j)}$$

성질 2로 부터  $\frac{P(n:j)}{P(0:j)}$ 은  $\mu_i(j)$ 에 대하여 증가함수이다. 따라서  $P(0:j)$ 은  $\mu_i(j)$ 에 대하여 감소함수가 될 수 밖에 없다. ■

성질 1-5는  $\mu_i(j)$ 값이 증가함에 따라 중요 함수들이 어떻게 변화하는 지를 설명해 준다. 성질 6은  $\mu_d(j)$ 값이 감소함에 따라 이러한 함수들이 어떻게 변화하는 지를 설명해 준다. 성질 6의 증명은 성질 1-5의 증명과 유사하므로 생략한다.

성질 6

(a) 임의의 정수  $n, m$  ( $n \geq 0, m > 0$ ) 대하여 다음이 성립한다.

$\frac{P(n+m:j)}{P(n:j)}$ 은  $\mu_d(j)$ 가 감소함에 따라 증가한다.

(b)  $P(0:j)$ 은  $\mu_d(j)$ 가 감소함에 따라 감소한다.

(c)  $\sum_{n=0}^{N_j-1} P_{bm}(n:j)(n+1)$ ,  $m \in U_j$ 은  $\mu_d(j)$ 가 감소함에 따라 증가한다.

성질 6은  $\mu_d(j)$ 값이 감소하는 경우도  $\mu_i(j)$ 값이 증가하는 경우와 중요 함수들에 대해 동일한 영향을 미침을 말해준다. 따라서  $\mu_i(j)$ 값은 증가하고 동시에  $\mu_d(j)$ 값은 감소하는 경우에도

성질 6은 성립함을 쉽게 보일 수 있다. 앞에서 유도된 성질들로 부터 다음의 보조정리들이 증명될 수 있다.

보조정리 1

$\mu_i(j)$ ,  $i \in U_j$ 가 증가함에 따라

(a)  $P_s(j)$ 는 감소한다.

(b)  $X_d(j)$ 은 증가한다.

(c)  $X_i(j)$ 은 증가하는 반면  $X_m(j)$ ,  $m \neq i$ ,  $m \in U_j$ 는 감소한다.

증명

(a) 성질 2로 부터  $\mu_i(j)$ 가 증가함에 따라

$\frac{P(k:j)}{P(1:j)}$ ,  $k \geq 1$ 는 증가하므로  $\mu_i(j)$ 가 증가함에 따라  $P_s(j)$ 는 감소함을 다음 식으로 부터 알 수 있다.

$$P_s(j) = \frac{P(1:j)}{1 - P(0:j)} = \frac{1}{\sum_{k=1}^{C_i+N_j} \frac{P(k:j)}{P(1:j)}}$$

(b)  $S_d(j)$ 의 산출율(throughput)은  $X_d(j) = \mu_d(j)(1 - P(0:j))$ 로 표현된다. 성질 5로 부터  $P(0:j)$ 은  $\mu_i(j)$ 가 증가함에 따라 감소하므로  $X_d(j)$ 은  $\mu_i(j)$ 이 증가함에 따라 증가한다.

(c)  $S_m(j)$ 의 산출율은 다음과 같이 표현된다.

$$\frac{1}{X_m(j)} = \frac{1}{\mu_m(j)} + \sum_{n=0}^{N_j-1} P_{bm}(n:j)(n+1) \frac{1}{\mu_d(j)}$$

성질 4로 부터  $\sum_{n=0}^{K-1} P_{bm}(n:N_j+1)(n+1) \frac{1}{\mu_d(j)}$ 은  $\mu_i(j)$ 이 증가함에 따라 증가

하며 결과적으로 만일  $m \neq i$ 이면  $X_m(j)$ 은  $\mu_i(j)$ 이 증가함에 따라 감소한다. 그러나 시스

템의 산출을  $X_d(j)$ 은  $\mu_i(j)$ 이 증가함에 따라 증가하며,  $X_d(j) = \sum_{m \in U_j} X_m(j)$ 이므로  $X_i(j)$ 은  $\mu_i(j)$ 이 증가함에 따라 증가한다. ■

**보조정리 2**

$\mu_d(j)$ 가 감소함에 따라

- (a)  $P_s(j)$ 는 감소한다.
- (b)  $X_m(j)$ ,  $m \in U_j$ 는 감소한다.
- (c)  $X_d(j)$ 는 감소한다.

**증명**

(a) 보조정리 1.a와 동일한 방식으로 증명 가능하다.

$$(b) \frac{1}{X_m(j)} = \frac{1}{\mu_m(j)} + \sum_{n=0}^{N_i-1} P_{bm}(n:j)$$

$(n+1) \frac{1}{\mu_d(j)}$ 에서  $\mu_d(j)$ 가 감소하면  $\sum_{n=0}^{N_i-1} P_{bm}(n:j)(n+1)$ 와  $\frac{1}{\mu_d(j)}$ 이 모두 증가하므로  $X_m(j)$ 는 감소한다.

(c)  $X_d(j) = \sum_{m \in U_j} X_m(j)$ 이므로  $\mu_d(j)$ 가 감소하면  $X_d(j)$ 는 감소한다. ■

**보조정리 3**

T(j)과 관련된 다음 두 매개변수 값의 집합이 있다고 가정하자.

$$\bar{\mu}^1(j) = (\mu_i^1(j), \dots, \mu_k^1(j), \mu_d^1(j)), \quad i, k \in U_j,$$

$$\bar{\mu}^2(j) = (\mu_i^2(j), \dots, \mu_k^2(j), \mu_d^2(j)), \quad i, k \in U_j,$$

집합 A, B, C를  $A = \{i \in U_j \mid \mu_i^1(j) < \mu_i^2(j)\}$ ,  $B = \{i \in U_j \mid \mu_i^1(j) > \mu_i^2(j)\}$  그리고  $C = \{i \in U_j \mid \mu_i^1(j) = \mu_i^2(j)\}$ 라 정의하고,  $\mu_d^1(j) = \mu_d^2(j)$

라 가정하자. 그러면 다음 관계식이 성립한다.

$$\sum_{i \in A} X_i^1(j) < \sum_{i \in A} X_i^2(j), \quad \sum_{i \in B} X_i^1(j) > \sum_{i \in B} X_i^2(j).$$

**증명**

T(j)의 upstream 서어버의 서비스율이  $i \in A$ 에 대해  $\mu_i^1(j)$ 에서  $\mu_i^2(j)$ 로 증가하면 보조정리 1으로 부터  $X_i(j)$ ,  $i \in A$ 는 감소하는 반면  $X_d(j)$ 은 증가하며 이는  $\sum_{i \in A} X_i(j)$ 가 증가함을 의미한다. 또한 서비스율이  $i \in B$ 에 대해  $\mu_i^1(j)$ 에서  $\mu_i^2(j)$ 로 감소하면  $\sum_{i \in A} X_i(j)$ 은 증가하는 반면  $\sum_{i \in B} X_i(j)$ 은 감소하며 이러한 사실로부터 보조정리 3는 성립한다. ■

**4.2 알고리즘의 수렴성 및 해의 유일성 증명**

이제 4.1에서 규명된 하부시스템의 특성을 이용하여 알고리즘(L)의 수렴성 및 해의 유일성을 증명해 본다.

**정리 1**

알고리즘(L)은 SE1의 해에 항상 수렴한다.

**증명**

알고리즘(L)은 고정점 형태(fixed point type) 방정식을 풀기 위하여 Gauss-Seidel기법을 적용한 것이다. 알고리즘의 목적은 미지의 매개변수 값  $\mu_i(j)$ 과  $\mu_d(j)$ 을 구하는 것이며 이를 위하여 초기단계와 반복단계로 구성되어 있다. 매 반복단계에서는  $P_s(j)$ ,  $\mu_i(j)$ ,  $P_{bi}(n:j)$ 과  $\mu_d(j)$ 를 새로이 계산한다.  $P_s^n(j)$ ,  $\mu_i^n(j)$ ,  $P_{bi}^n(n:j)$ 과  $\mu_d^n(j)$ 를 n번째 반복단계에서 얻어진 매개변수의 값이라 하자.  $\mu_d^0(j)$ 는  $\mu_d(j)$ 의 초기치를 의

미하며 따라서  $\mu_d^0(j) = \mu_j$ 로 주어진다.  $n$ 번째 반복단계에서는 전진단계와 후진단계가 차례로 수행된다. 전진단계에서는 하부시스템  $T(j)$ 가  $j=1, \dots, M$ 의 순서로 분석되며 전진단계 종료 후 갖게되는 매개변수의 값은  $\mu_i^n(j)$ 과  $\mu_d^{n-1}(j)$ 이다. 후진단계에서는  $j=M, \dots, 1$ 의 순서로 하부시스템  $T(j)$ 가 분석되며 후진단계 종료 후 갖게되는 매개변수의 값은  $\mu_i^n(j)$ 과  $\mu_d^n(j)$ 이다. 노우드  $j$ 와 직접 연결된 포화 서버가 존재할 경우에는  $n$ 과 관계없이  $\mu_{0j}^n(j) = \mu_{0j}$ 이고 하부시스템  $T(M)$ 에 대해서는  $n$ 과 관계없이  $\mu_d^n(M) = \mu_M$ 이 된다. 정리 1을 증명하기 위하여 다음 보조정리를 우선 증명하자.

**보조정리 4**

반복단계  $n$ 과  $n+1$ 의 종료 후 얻게되는 매개변수의 값에는 다음과 같은 관계가 성립한다.

$$\begin{aligned} \mu_i^{n+1}(j) &\geq \mu_i^n(j), \quad i \in U_j, \quad j=1, \dots, M, \\ \mu_d^{n+1}(j) &\leq \mu_d^n(j), \quad j=1, \dots, M. \end{aligned}$$

**증명**

$\mu_d^n(j) \leq \mu_d^{n-1}(j), j=1, \dots, M$  이라 가정하고 반복단계  $n+1$ 의 전진단계를 고려하여 보자. 집합  $I_s = \{j \mid S_{0j}(j) \text{가 } T(j) \text{의 유일한 upstream 서버}\}$ 라 정의하자. 전진단계에서는  $j=1, \dots, M$ 의 순서로 하부시스템들이 분석되며 네트워크의 구조상  $j \in I_s$ 인  $T(j)$ 들이 우선적으로 분석된다.  $j \in I_s$ 인  $T(j)$ 에서는 upstream 서버의 서비스율은  $n$ 과 관계없이  $\mu_{0j}^n(j) = \mu_{0j}$ 가 되며  $\mu_d^n(j) \leq \mu_d^{n-1}(j)$ 라 가정했으므로 보조정리 2로부터

$P_s^{n+1}(j) \leq P_s^n(j)$ 가 성립한다. 이러한 사실과 식 (7)로부터  $\mu_j^{n+1}(d_j) \geq \mu_j^n(d_j)$ 이 성립함을 알 수 있다.  $m=d_j$ 이라 하면 네트워크의 구조상  $T(m)$ 의 분석시에는  $\mu_i(m), i \in U_m$ 의 값이 모두 새롭게 계산된 상태이며, 이들 값들은  $\mu_i^{n+1}(m) \geq \mu_i^n(m)$ 의 관계가 성립한다. 또한  $\mu_d^n(m) \leq \mu_d^{n-1}(m)$ 라 가정하였으므로 보조정리 1, 2와 식 (7)로부터  $P_s^{n+1}(m) \leq P_s^n(m), \mu_m^{n+1}(d_m) \geq \mu_m^n(d_m)$ 이 성립하며, 이와같은 관계는 하부시스템  $T(M)$ 의 분석이 끝날 때 까지 성립한다. 이러한 성질을 이용하여  $n$ 에 대하여 수학적 귀납법을 적용하면 인접하는 반복단계 간에 다음 성질이 존재함을 보일 수 있다.

$$\begin{aligned} \mu_d^n(j) \leq \mu_d^{n-1}(j) \text{이면 } \mu_i^{n+1}(j) &\geq \mu_i^n(j), \\ i \in U_j, \quad j=1, \dots, M. \end{aligned} \tag{12}$$

이제  $\mu_i^{n+1}(j) \geq \mu_i^n(j), i \in U_j, j=1, \dots, M$ 이 성립한다고 가정하고 반복단계  $n+1$ 의 후진단계를 고려하여 보자. 후진단계에서는  $j=M, \dots, 1$ 의 순서로 하부시스템이 분석되며 전진단계에서 구한  $\mu_i^{n+1}(j)$ 의 값을 이용하여  $\mu_d^{n+1}(j)$ 의 값을 새로이 계산한다. 후진단계에서 가장 먼저 분석되는  $T(M)$ 의 경우에는  $\mu_d^{n+1}(M) = \mu_d^n(M) = \mu_M$ 이다. 또한  $\mu_i^{n+1}(M) \geq \mu_i^n(M)$ 라 가정하였으므로 성질4에 의해 모든  $m \in U_M$ 에 대해  $\sum_{k=0}^{N_M-1}$

$$\begin{aligned} P_{bm}^{n+1}(k; M)(k+1) \frac{1}{\mu_d^{n+1}(M)} &\geq \sum_{k=0}^{N_M-1} P_{bm}^n(k; M) \\ (k+1) \frac{1}{\mu_d^n(M)} \end{aligned} \text{ 이 성립하며 이러한 사실과 식}$$

(6)으로 부터  $\mu_d^{n+1}(m) \leq \mu_d^n(m)$ 이 성립한다. 하부시스템 T(m)의 분석시에도  $\mu_i^{n+1}(m) \geq \mu_i^n(m)$ ,  $i \in U_m$ 이고  $\mu_d^{n+1}(m) \leq \mu_d^n(m)$ 이므로 T(M)분석시와 동일한 과정을 거쳐  $i \in U_m$ 에 대해  $\mu_d^{n+1}(i) \leq \mu_d^n(i)$ 가 성립하며 이와같은 관계는 하부시스템 T(1)의 분석이 끝날 때 까지 유지된다. 이러한 성질을 이용하여 n에 대하여 수학적 귀납법을 적용하면 인접하는 반복단계 간에 다음 성질이 존재함을 보일 수 있다.

만일  $\mu_i^{n+1}(j) \geq \mu_i^n(j)$ ,  $i \in U_j$  이면

$$\mu_d^{n+1}(j) \leq \mu_d^n(j). \tag{13}$$

(12)와 (13)으로 부터 다음 관계를 얻을 수 있다.

$$\begin{aligned} \mu_d^n(j) \leq \mu_d^{n-1}(j) &\rightarrow \mu_i^{n+1}(n+1) \geq \mu_i^n(j) \\ \rightarrow \mu_d^{n+1}(j) \leq \mu_d^n(j). \end{aligned} \tag{14}$$

식 (6)으로 부터  $\mu_d^1(j) \leq \mu_d^0(j) = \mu_j, j=1, \dots, M$ 임을 알 수 있으며 이러한 사실과 관계 (14)으로 부터 보조정리 4는 성립한다. ■

이제 보조정리 4를 이용하여 정리 1을 증명하자. 보조정리 4는 급수  $\{\mu_i^n(j), n=1, 2, \dots\}$ ,  $i \in U_j$ 가 단조 증가함을 의미한다. 또한 식 (7)로부터  $\mu_i^n(j) \leq \mu_i$ 가 성립함을 알 수 있으며 급수가 단조 증가하고 상한을 가지므로 급수  $\{\mu_i^n(j), n=1, 2, \dots\}$ ,  $i \in U_j$ 는 수렴함을 알 수 있다. 보조정리 4는 또한 급수  $\{\mu_d^n(j), n=1, 2, \dots\}$ 가 단조 감소함을 의미한다.

$\mu_d^n(j)$ 의 하한을 구하기 위해 식 (6)의 값을 가장 크게 하여주는  $P_{b_j}^n(N_m-1; m) = 1, j \in U_m$ 의 경우를 식 (6)에 대입하면 다음 관계를 얻는다.

$$\mu_d^n(j) \geq \frac{\mu_j \mu_d^n(m)}{\mu_d^n(m) + (N_m - 1) \mu_j}.$$

급수가 단조 감소하고 하한을 가지므로 급수  $\{\mu_d^n(j), n=1, 2, \dots\}$ 는 수렴함을 알 수 있다. 알고리즘(L)은 방정식체계 SE1에 Gauss-Seidel법을 적용한 것이므로  $\mu_d^n(j), \mu_i^n(j), i \in U_j$ 의 극한치는 SE1의 해가 된다. ■

정리 1은 SE1이 최소한 하나의 해를 가지며 이 해는 알고리즘(L)에 의해 구해짐을 의미한다.

다음에 소개되는 정리는 해의 존재가 보장된다면 그 해는 유일한 해임을 말해준다.

**정리 2**

방정식 체계 SE1은 해가 존재한다면 유일하다.

**증명**

SE1이 서로 다른 두개 이상의 해를 갖을 경우 모순이 발생함을 보임으로써 정리 2를 증명하고자 한다. 이를 위하여 SE1이 서로 다른 두 개의 해를 가지고 있다고 가정하자. 첫 번째 해는 위첨자 (1)로 표시하고 두 번째 해는 위첨자 (2)로 표시하도록 하며  $\mu_d^{(1)}(j), \mu_i^{(1)}(j), \mu_d^{(2)}(j), \mu_i^{(2)}(j)$ 가 이들 두 해를 나타낸다고 하자. 정리 1에서 정의된 집합  $I_s$ 에 속하는 j에 대해서는  $\mu_{0_j}^{(1)}(j) = \mu_{0_j}^{(2)}(j) = \mu_{0_j}$ 가 성립하며 T(M)에 대해서는  $\mu_d^{(1)}(M) = \mu_d^{(2)}(M) = \mu_M$ 가 성립한다. 우선  $j \in I_s$ 인 모든 j에 대하여  $\mu_d^{(1)}(j) = \mu_d^{(2)}(j)$ 이 성립하는 경우를 가정하자. T(j)의 매개변수 값에 있어 두 해가 동일하므로  $X_d^{(1)}(j) = X_d^{(2)}(j)$



가 성립하고, 흐름보존법칙에 의해  $X_j^{(1)}(d_j) = X_j^{(2)}(d_j)$ 이 성립한다. 또한 식 (7)에 의해  $\mu_j^{(1)}(d_j) = \mu_j^{(2)}(d_j)$ 이 성립한다.  $j=1, \dots, M$ 의 순서로 하부시스템이 분석되므로 네트워크의 구조상 임의의 하부시스템  $T(m)$ 이 분석될 경우, 모든  $j \in U_m$ 에 대해  $\mu_j^{(1)}(m) = \mu_j^{(2)}(m)$ ,  $X_j^{(1)}(m) = X_j^{(2)}(m)$ 이 성립하게 된다. 하부시스템  $T(m)$ 에서는  $\sum_{j \in U_m} X_j^{(n)}(m) = X_d^{(n)}(m)$ ,  $n=1, 2$ 가 만족되어야 하므로  $X_d^{(1)}(m) = X_d^{(2)}(m)$ 이 성립할 수 밖에 없으며 이는  $\mu_d^{(1)}(m) = \mu_d^{(2)}(m)$ 이 됨을 의미한다. 따라서 하부시스템  $T(m)$ 의 매개변수 값도 두 해가 동일하게 된다.  $T(m)$ 에 대한 두 해의 매개변수 값이 동일하므로 식 (7)과 흐름보존법칙으로부터  $\mu_m^{(1)}(d_m) = \mu_m^{(2)}(d_m)$ ,  $X_m^{(1)}(d_m) = X_m^{(2)}(d_m)$ 이 성립한다. 이와같은 절차를 반복 적용하면 모든  $j$ 에 대해  $\mu_i^{(1)}(j) = \mu_i^{(2)}(j)$ ,  $i \in U_j$ ,  $\mu_d^{(1)}(j) = \mu_d^{(2)}(j)$ 이 성립함을 알 수 있다. 따라서  $j \in I_s$ 인 모든  $j$ 에 대하여  $\mu_d^{(1)}(j) = \mu_d^{(2)}(j)$ 이 성립하면 SE1에는 서로 다른 두 해가 존재할 수 없다.

이제 하부시스템  $T(j)$ ,  $j \in I_s$ 에 대해  $\mu_d^{(1)}(j) \neq \mu_d^{(2)}(j)$ 인 경우를 고려하여 보자. 만일  $\mu_d^{(1)}(j) > \mu_d^{(2)}(j)$ 이면 보조정리 2에 의해  $P_s^{(1)}(j) > P_s^{(2)}(j)$ ,  $X_d^{(1)}(j) > X_d^{(2)}(j)$ 가 성립하며, 이러한 사실과 식 (7)로부터  $m=d_j$ 라면  $\mu_j^{(1)}(m) < \mu_j^{(2)}(m)$ 이 성립한다. 또한 흐름보존법칙이 성립하므로  $X_j^{(1)}(m) > X_j^{(2)}(m)$ 의 관계

가 성립한다. 유사하게  $\mu_d^{(1)}(j) < \mu_d^{(2)}(j)$ 이면  $P_s^{(1)}(j) < P_s^{(2)}(j)$ ,  $X_d^{(1)}(j) < X_d^{(2)}(j)$ 가 성립하며  $\mu_j^{(1)}(m) > \mu_j^{(2)}(m)$ ,  $X_j^{(1)}(m) < X_j^{(2)}(m)$ 의 관계가 성립한다. 일반적으로  $T(m)$ 의 upstream 서어버는 2개 이상일 수 있으며  $\mu_j^{(1)}(m) \neq \mu_j^{(2)}(m)$ 이 되는 upstream 서어버,  $S_j(m)$ ,도 2개 이상 존재할 수 있다. 이 경우 집합  $A(m)$ ,  $B(m)$ ,  $C(m)$ 을  $B(m) = \{j \mid \mu_j^{(1)}(m) > \mu_j^{(2)}(m)\}$ ,  $A(m) = \{j \mid \mu_j^{(1)}(m) < \mu_j^{(2)}(m)\}$ ,  $C(m) = \{j \mid \mu_j^{(1)}(m) = \mu_j^{(2)}(m)\}$ 과 같이 정의하고 다음의 3 경우로 나누어 분석해 보자.

경우 (1):  $A(m)$ ,  $B(m)$ 에 모두 원소가 존재할 경우

이 경우에는 앞에서 유도된 성질로부터 다음 관계를 보일 수 있다.

$$j \in A \text{에 대해서는 } \mu_j^{(1)}(m) < \mu_j^{(2)}(m), X_j^{(1)}(m) > X_j^{(2)}(m), \quad j \in B \text{에 대해서는 } \mu_j^{(1)}(m) > \mu_j^{(2)}(m), X_j^{(1)}(m) < X_j^{(2)}(m). \quad (15)$$

(15)는  $\mu_d^{(1)}(m)$ ,  $\mu_d^{(2)}(m)$ 의 값에 관계없이 모순되는 관계이며 따라서 경우 (1)은 발생할 수 없다.

경우 (2):  $B(m)$ 에 원소가 존재하지 않을 경우 이 경우에는  $j \in A(m)$ 에 대해서  $\mu_j^{(1)}(m) < \mu_j^{(2)}(m)$ ,  $X_j^{(1)}(m) > X_j^{(2)}(m)$ 의 관계가 성립함을 보일 수 있다. 이 관계와  $X_d^{(1)}(m) =$

$\sum_{j \in U_m} X_j^{(1)}(m) \gg \sum_{j \in U_m} X_j^{(2)}(m) = X_d^{(2)}(m)$ 으로 부터  $\mu_d^{(1)}(m) \gg \mu_d^{(2)}(m)$ 이 성립하여야 한다. 모든  $k \in U_m$ 에 대해  $\mu_k^{(1)}(m) \leq \mu_k^{(2)}(m)$ 이 성립하고  $\mu_d^{(1)}(m) \gg \mu_d^{(2)}(m)$ 이므로  $P_s^{(1)}(m) \gg P_s^{(2)}(m)$ 이 성립하고  $\mu^{*(1)}(m) \ll \mu^{*(2)}(m)$ 의 관계와 식 (7)로 부터  $\mu_m^{(1)}(d_m) \ll \mu_m^{(2)}(d_m)$ 이 성립한다. 또한 흐름보존법칙으로 부터  $X_m^{(1)}(d_m) \gg X_m^{(2)}(d_m)$ 이 성립한다.

경우 (3):  $A(m)$ 에 원소가 존재하지 않을 경우 이 경우에는 경우 (2)와 유사한 과정을 거쳐  $\mu_d^{(1)}(m) \ll \mu_d^{(2)}(m)$ ,  $\mu_m^{(1)}(d_m) \gg \mu_m^{(2)}(d_m)$ ,  $X_m^{(1)}(d_m) \ll X_m^{(2)}(d_m)$ 이 성립함을 보일 수 있다.

임의의 하부시스템  $T(n)$ 을 분석할 경우에도  $T(m)$ 의 경우와 유사한 결과를 얻을 수 있다.  $T(n)$ 의 분석시에도  $j \in A(n)$ 의 경우에는  $\mu_j^{(1)}(n) \ll \mu_j^{(2)}(n)$ ,  $X_j^{(1)}(n) \gg X_j^{(2)}(n)$ 의 관계가 성립하고,  $j \in B(n)$ 의 경우에는  $\mu_j^{(1)}(n) \gg \mu_j^{(2)}(n)$ ,  $X_j^{(1)}(n) \ll X_j^{(2)}(n)$ 의 관계가 성립하며,  $A(n)$ ,  $B(n)$ 에의 원소존재 여부에 따라 3 경우로 분류된다. 만일  $A(n)$ ,  $B(n)$ 에 원소가 모두 존재하면 모순이 됨을 보일 수 있고,  $B(n)$ 에 원소가 존재하지 않으면  $\mu_d^{(1)}(n) \gg \mu_d^{(2)}(n)$ ,  $\mu_n^{(1)}(d_n) \ll \mu_n^{(2)}(d_n)$ ,  $X_n^{(1)}(d_n) \gg X_n^{(2)}(d_n)$ 이 성립하고,  $A(n)$ 에 원소가 존재하지 않으면  $\mu_d^{(1)}(n) \ll \mu_d^{(2)}(n)$ ,  $\mu_n^{(1)}(d_n) \gg \mu_n^{(2)}(d_n)$ ,  $X_n^{(1)}(d_n) \ll X_n^{(2)}(d_n)$ 이 성립한다. 이러한 과정을 계속하면 결국 경우(1)의 모순이 발생하거나  $T(M)$ 에 도달하게 된다. 그러나  $\mu_d^{(1)}(M) = \mu_d^{(2)}(M) = \mu_M$ 가 성립하여야

하므로 경우 (2)나 경우 (3)의  $\mu_d^{(1)}(M) \gg \mu_d^{(2)}(M)$  혹은  $\mu_d^{(1)}(M) \ll \mu_d^{(2)}(M)$ 은 모순임을 알 수 있으며 따라서 정리 2은 성립한다. ■

정리 1은 SE1이 최소한 하나의 해를 가지며 이 해는 알고리즘(L)에 의해 구해짐을 의미한다. 정리 2는 SE1이 두 개 이상의 해를 갖지 않음을 보장한다. 따라서 이상의 두 정리로부터 다음의 따름정리가 성립한다.

#### 따름정리 1

방정식체계 SE1의 해는 존재하며 유일하다. 이 해는 또한 알고리즘(L)에 의하여 구하여 진다.

## 5. 결 론

붕괴가 존재하는 개방형 대기행렬 네트워크의 성능평가를 위해 최근 Lee외에 의해 제안된 알고리즘은 정확도, 속도, 안정성 등에서 매우 우수한 것으로 보고되었다[5]. 본 연구에서는 나무형태의 네트워크를 분석할 경우에는 Lee외[5]에 의해 제안된 알고리즘의 수렴성 및 해의 유일성이 증명될 수 있음을 보였다. 증명방법은 알고리즘이 반복단계를 거듭함에 따라 매개변수 값들은 단조증가 혹은 단조감소하고 이 값들이 상한 혹은 하한을 갖음을 보임으로써 알고리즘의 수렴성을 증명하였고, 유도된 방정식체계에 서로 다른 두개 이상의 해가 존재할 경우 모순이 발생함을 보임으로써 해의 유일성을 증명하였다.

본 연구는 이미 우수한 것으로 입증된 Lee외에 의해 제안된 알고리즘에 대한 이론적 토대를 제공하였다는 점에서 의의가 있을 뿐 만 아니라

직렬형 이외의 알고리즘에 대한 수렴성 및 해의 유일성 증명이라는 점에서 의의가 있다고 판단된다. 또한 보다 일반적인 형태를 갖는 네트워크 성능평가 알고리즘의 이론적 규명을 위한 기초 연구로서의 가치도 있다고 판단된다.

## 참 고 문 헌

- [1] T. Altioik and H. G. Perros, "Approximate Analysis of Arbitrary Configurations of Open Queueing Networks with Blocking," *Annals of Opns. Res.*, 9(1987), pp. 481-509.
- [2] Y. Dallery and Y. Frein, "On Decomposition Methods for Tandem Queueing Networks with Blocking," *Opns. Res.*, 41(1993), pp. 386-399.
- [3] Y. Dallery and S. B. Gershwin, "Manufacturing Flow Line Systems : a review of models and analytical results," *Queueing Sys.*, 12(1992), pp. 3-94.
- [4] H. S. Lee and S. M. Pollock, "Approximate Analysis of Open Acyclic Exponential Queueing Networks with Blocking," *Opns. Res.*, 38(1990), pp. 1123-1134.
- [5] H. S. Lee, A. Bouhchouch, Y. Frein and Y. Dallery, "Performance Evaluation of Open Queueing Networks with Arbitrary Configuration and Finite Buffers," to appear in *Annals of Opns. Res.*
- [6] K. O. Onvural, "Survey of Closed Queueing Networks with Blocking," *ACM Computing Surveys*, 22(1990), pp. 83-121.
- [7] H. G. Perros, "Queueing Networks with Blocking: A Bibliography," *Perf. Eval. Rev.*, 12(1986), pp. 8-14.
- [8] H. G. Perros, *Queueing Networks with Blocking*, Oxford University Press, New York, 1994.
- [9] H. G. Perros and P. M. Snyder, "A Computationally Efficient Approximation Algorithm for Analyzing Open Queueing Networks with Blocking," *Performance Evaluation*(1989).
- [10] Y. Takahashi, H. Miyahara, and J. Hasegawa, "An Approximation Method for Open Restricted Queueing Networks," *Opns. Res.*, 28(1980), PP. 594-602.