

불완전한 기계 및 랜덤가공시간을 갖는 폐쇄형 생산시스템의 성능분석에 관한 연구*

김호균 · 백천현 · 조형수

동의대학교 기계 산업시스템 공학부

Performance Analysis for Closed-Loop Production Systems with Unreliable Machines and Random Processing Times

H. G. Kim · C. H. Paik · H. S. Cho

In this paper we propose new approximate methods for the performance analysis of closed-loop production systems with unreliable machines and random processing times. Each approximate method decomposes the production system consisting of K machines into a set of K subsystems, each subsystem consisting of two machines separated by a finite buffer. Then, each subsystem is analyzed by three different analyzing methods in isolation. The population constraint of the closed-loop production system is taken into account by prescribing that the sum of average buffer levels in the subsystems is equal to the number of customers in the closed-loop production system. We establish a set of equations that characterize unknown parameters of the servers in the subsystems. An iterative procedure is then used to determine the unknown parameters. Experimental results show that these methods provide a good estimation of the throughput.

1. 서 론

봉쇄 대기행렬망(queueing network with blocking)은 생산시스템, 컴퓨터시스템 및 통신망 등을 모형화하고 성능을 평가하기 위한 기법으로 널리 사용되고 있다. 봉쇄 대기행렬망은 특수한 경우를 제외하고는 승법형(product form) 해를 갖지 않기 때문에 정확한 해는 수치적 기법에 의해서만 얻어질 수 있다. 그러나 수치적 기법들은 많은 계산시간을 요하게 되어, 비교적 작은 규모의 대기행렬망만을 처리하는데 국한된다. 따라서, 대부분의 연구에서는 근사적 해를 구하기 위한 발견적(heuristic) 해법의 개발에 치중되어 왔다(Bouchouch *et al.*).

최근에는 봉쇄를 갖는 폐쇄형 대기행렬망(closed queueing network)에 대한 발견적 해법에 관심이 고조되고 있으며, 여러 가지 근사해법들이 제안되었다. 그러나 폐쇄형 대기행렬망은 시스템 내에 존재하는 고객의 수가 일정해야 한다는 제약이 부과되어 개방형 대기행렬망(open queueing network)에 비해 많은 연구가 이루어지지 못한 실정이다.

문헌에서 소개된 폐쇄형 봉쇄 대기행렬망(CQN-B : closed queueing network with blocking)에 대한 근사해법들을 살펴보면, Dallery and Frein(1989)와 Kim and Lie(1996)는 CQN-B에 대한 성능분석을 하였지만, 첫 번째 버퍼가 무한하고 서버가 완전하다고 가정하였다. 유한한 버퍼를 갖는 CQN-B에 대하여, Bouchouch *et al.*는 서버가 일반 서비스시간 분포를 갖는 CQN-B에 대한 근사해법을 제시하였지만, 역시 서버는 완전하다고 가정하였다.

반면, 불완전한 서버를 갖는 모형에 대한 연구는 아직 미비한 편이며, 지금까지 이루어진 연구들은 대부분 서비스시간에 대한 제약을 두었다. 먼저, 개방형 봉쇄 대기행렬망에 대한 연구에서 Dallery *et al.*(1988, 1989)는 서버가 확정적(deterministic)인 서비스시간을 갖는다고 가정하였으며, Gershwin(1989)은 서버가 랜덤서비스시간을 갖는 경우에 대해 효율적인 알고리즘을 제시하였다. 폐쇄형 봉쇄 대기행렬망에 대해서 Frein *et al.*(1996)은 매우 효율적인 근사해법을 제시하였지만 역시 서버가 확정적인 서비스시간을 갖는다고 가정하였다. 또한 불완전한 서버를 갖는 모형에 대한 대부분의 연구들은 시스템의 행태를 연

*이 논문은 1997년도 한국학술진흥재단의 공모과제 연구비에 의해서 연구되었음.

속흐름(continuous flow) 모형으로 근사화하여 분석하였다. 따라서 불완전한 서버 및 랜덤서비스시간을 갖는 폐쇄형 봉쇄 대기행렬망(CQN-BF: closed queueing network with blocking and server failure)에 대한 연구는 아직 이루어지지 못하였다.

한편, 완전 및 불완전한 서버를 갖는 폐쇄형 대기행렬망에 대한 대부분의 연구들은 Gershwin(1987)이 제시했던 분해법(decomposition method)을 사용하고 있으며, 효율적인 방법 개발을 위해서 Dallery et al.(1989)가 제시한 반복적 알고리즘에 기초하고 있다.

이에 본 연구에서는 기계가 불완전하고 랜덤가공시간을 갖는 폐쇄형 생산시스템을 CQN-BF로 모형화하여, 폐쇄형 생산시스템의 성능평가를 위한 알고리즘을 제시하고자 한다. 본 연구에서 제시하고자 하는 알고리즘은 기존 문헌에서 효율성이 확인되고 있는 분해법과 반복적 알고리즘에 기초하였으며, 세 가지의 서로 다른 접근법을 사용하여 각각 폐쇄형 생산시스템의 성능척도를 분석한다.

먼저, 다음 절에서 폐쇄형 생산시스템에 대한 모형을 설명한다. 3절에서는 분해법을 이용한 근사해법 개발을 위해 세 가지 접근법에 의한 세 가지 알고리즘을 소개한다. 4절에서는 폐쇄형 생산시스템에서 산출율(throughput)의 상한에 대한 분석을 수행한다. 다음 5절에서는 수립된 세 가지 알고리즘의 효과 분석을 위해서, 다양한 경우의 자료로서 수치실험한다.

2. 모형 설명

폐쇄형 생산시스템은 <그림 1>의 (a)와 같이 원부품이 로딩

스테이션(loading station)에서 팔렛(pallet)에 올려지고 팔렛과 부품이 정하여진 일련의 기계를 방문하면서 가공된다. 언로딩 스테이션(unloading station)에서는 부품이 팔렛으로부터 내려져 완성된 부품은 그 시스템을 떠나고 빈 팔렛은 로딩 스테이션으로 다시 보내져 새로운 원부품이 올려진다. 이때 로딩 스테이션에서 팔렛을 통해 부품이 이동되고 팔렛이 가용될 때만 새로운 부품이 시스템에 유입될 수 있다. 팔렛은 총 N 개가 존재하며, 폐쇄형 생산시스템 내에 항상 N 개의 팔렛만 존재할 수 있다. 따라서 팔렛은 폐쇄형 생산시스템에 동시에 존재할 수 있는 부품의 수를 제한하는 요소가 된다.

한편, 로딩 스테이션 및 언로딩 스테이션을 포함한 각 기계들은 부품 가공중에 있을 때에만 고장이 발생되며, 고장이 발생하면 즉시 수리가 시작된다. 기계에 고장이 발생하는 순간, 가공중이던 재공품은 기계가 수리되는 동안 기계에 계속 남아 있으며 수리완료 즉시 가공이 재개된다. 또한, 각 기계의 고장 발생시 다른 기계에 대한 영향을 줄이기 위해 기계 사이에 재공품의 저장시설, 버퍼(buffer)를 갖춘다.

이러한 폐쇄형 생산시스템을 CQN-BF로 모형화하면 <그림 1>의 (b)와 같다. K 개의 서버(M_1, M_2, \dots, M_K)와 K 개의 버퍼(B_1, B_2, \dots, B_K)로 구성된 CQN-BF는 첫 번째 서버 M_1 에서 시작하여 B_1, M_2 를 거쳐 M_K, B_K 까지 거친 후, 첫 번째 서버 M_1 으로 다시 피드백된다. 각 서버 M_i 에서의 서비스시간은 서비스를 μ_i 를 갖는 지수분포를 따르며($i=1, 2, \dots, K$), 서비스 제공순서 원칙은 선입선출(FIFO) 원칙을 따른다고 가정한다. 서버 M_i 의 수명시간과 수리시간은 각각 고장을 λ_i 와 수리율 γ_i 를 갖는 독립적인 지수분포를 따른다($i=1, 2, \dots, K$).

버퍼 B_i 의 저장용량은 하위(downstream)서버 M_{i+1} 의 작업공

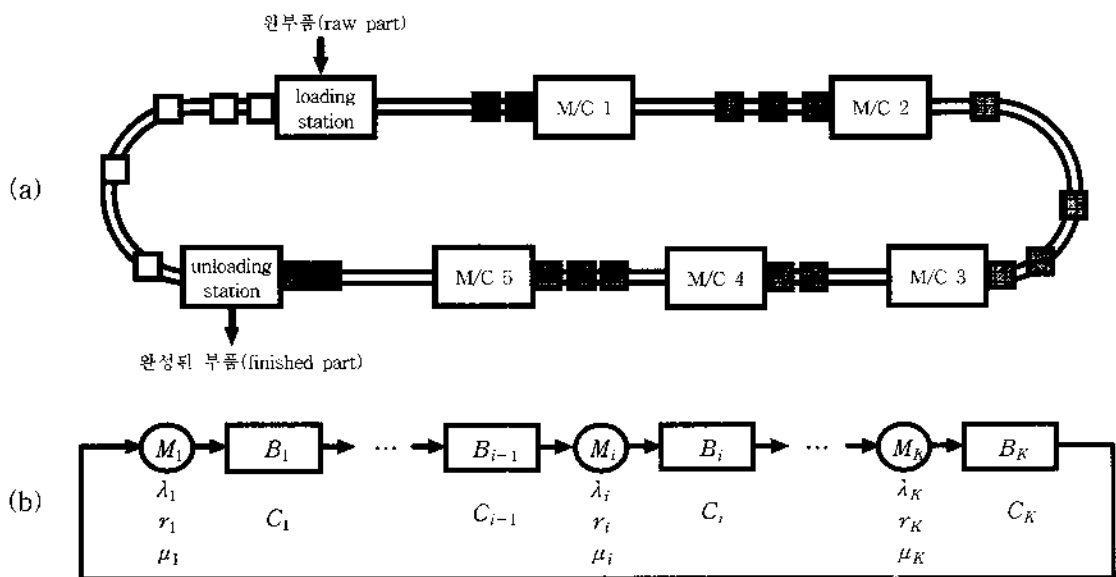


그림 1. 폐쇄형 생산시스템 모형.

간을 포함하여 C_i 이다. 서버 M_i 에서 서비스를 마치고 하부버퍼로 고객을 이전하려고 할 때, 버퍼 B_i 가 가득 찬 경우, 서버 M_i 는 봉쇄된다. 이러한 봉쇄현상을 BAS(Blocking-After-Service) 또는 전달봉쇄 현상이라 한다 (Choong and Gershwin, 1987; Dallery and Frein, 1989; Frein *et al.*, 1996).

시스템의 총 고객수 (N)가 가장 작은 용량을 가진 버퍼 (C_{\min})보다 같거나 작다면 결코 봉쇄는 발생될 수 없다. 또한, N 은 버퍼 저장용량의 합(C)을 초과할 수 없다. 그러므로 본 연구에서는 N 을 C_{\min} 과 C 사이에 있을 때로 한정한다.

3. 분해법을 이용한 근사해법

분해법은 원(original) 시스템을 유한한 버퍼용량을 갖는 독립된 대기행렬 시스템으로 분해하고, 각 시스템의 도착과정, 서비스과정 및 버퍼용량을 특성화하여 원 시스템에서의 행태에 가까운 근사시키는 방법이다.

3.1 분해법

분해법에서는 먼저, 원 시스템 L 을 K 개의 하부시스템 $L(i)$ 로 분해한다 ($i = 1, 2, \dots, K$). 하부시스템 $L(i)$ 는 중간 버퍼 $B(i)$ 를 중심으로 상위서버(upstream server) $M_u(i)$ 와 하위서버(downstream server) $M_d(i)$ 로 구성되는데, 하부시스템 $L(i)$ 는 버퍼 $B(i)$ 에서의 고객 흐름이 원 시스템 L 의 버퍼 B_i 에서의 고객 흐름과 근사하도록 모형화하게 된다. 이때 중간버퍼 $B(i)$ 는 원 시스템 L 의 버퍼 B_i 에 해당되고, 버퍼 $B(i)$ 의 저장용량은 $C(i)$ 로서 원 시스템 L 에서 버퍼 B_i 의 저장능력 C_i 와 동일한 값을 가진다. 버퍼 $B(i)$ 를 중심으로 원 시스템 L 에서 버퍼 B_i 의 상위부분의 행태를 나타내

는 상위서버 $M_u(i)$ 와 원 시스템 L 에서 버퍼 B_i 의 하위부분의 행태를 나타내는 하위서버 $M_d(i)$ 는 각각 원 시스템 L 의 M_i 와 M_{i+1} 에 해당된다. 이때, 만일 $i = K$ 이면 $i+1 = 1$ 이 된다. 하부시스템 $L(i)$ 에서 상위서버 $M_u(i)$ 는 결코 결핍(starvation)되지 않으며, 하위서버 $M_d(i)$ 는 결코 봉쇄되지 않는다. <그림 2>는 분해된 K 개의 하부시스템 $L(i)$ 를 보여주고 있다.

이후 표현법으로 정수첨자(예 : M_u, B, M_d 등)는 원 시스템 L 에서의 서버 및 버퍼, 그리고 매개변수를 지칭하며 괄호 내 정수(예 : $M_u(i), B(i)$ 등)는 하부시스템 $L(i)$ 에서의 서버 및 버퍼, 그리고 매개변수를 지칭하기로 한다.

먼저, 분해된 하부시스템 $L(i)$ 의 분석을 위하여 다음과 같은 기호를 정의한다.

- $1/\mu_u(i)$: 상위서버 $M_u(i)$ 의 평균 서비스시간
- $1/\mu_d(i)$: 하위서버 $M_d(i)$ 의 평균 서비스시간
- $1/D_u(i)$: 상위서버 $M_u(i)$ 의 출발간(interdeparture) 평균 시간
- $1/D_d(i)$: 하위서버 $M_d(i)$ 의 출발간 평균시간
- $P_s(i)$: 하위서버 $M_d(i)$ 가 결핍될 확률
- $P_b(i)$: 상위서버 $M_u(i)$ 가 봉쇄될 확률

이제 각 서버로부터의 출발간 평균시간을 고려하여 본다. 서버 $M_u(i)$ 로부터의 출발간 평균시간은 서버 $M_u(i)$ 의 봉쇄시간과 평균 서비스시간의 합으로 구성된다. 만약 서버 $M_u(i)$ 의 서비스 종료시점에서 버퍼 $B(i)$ 가 가득 차 있다면, 서버 $M_u(i)$ 는 봉쇄되고 봉쇄시간은 서버 $M_u(i)$ 의 서비스 종료시점에서 서버 $M_d(i)$ 의 잔여 서비스시간이 될 것이다. 그러므로 서버 $M_u(i)$ 로부터의 출발간 평균시간 $1/D_u(i)$ 는 다음과 같이 나타낼 수 있다.

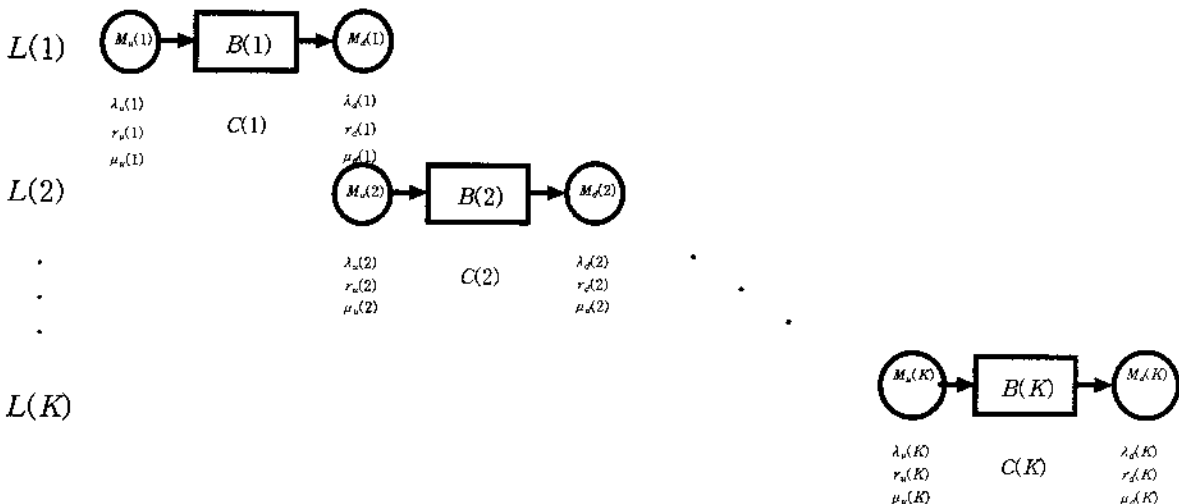


그림 2. 분해된 K 개의 하부시스템들.

$$\frac{1}{D_u(i)} = \frac{1}{\mu_u(i)} + P_b(i) \frac{1}{\mu_d(i)} \quad (1)$$

유사하게, 서버 $M_d(i)$ 로부터의 출발간 평균시간은 서버 $M_u(i)$ 의 결핍시간과 서비스시간의 합으로 구성된다. 만약 서버 $M_d(i)$ 의 서비스 종료시점에서 버퍼 $B(i)$ 가 비어 있다면, 서버 $M_d(i)$ 는 결핍되고, 결핍시간은 서버 $M_u(i)$ 의 서비스 종료시점에서 서버 $M_d(i)$ 의 잔여 서비스시간이 될 것이다. 그러므로 서버 $M_d(i)$ 로부터의 출발간 평균시간 $1/D_d(i)$ 은 다음과 같이 나타낼 수 있다.

$$\frac{1}{D_d(i)} = \frac{1}{\mu_d(i)} + P_s(i) \frac{1}{\mu_u(i)} \quad (2)$$

한편, 각 하부시스템 행태들이 원 시스템의 행태를 표현하기 위해서는, 서버 $M_u(i)$ 와 $M_d(i)$ 의 출발간 평균시간이 흐름보존 원칙에 의하여 다음과 같은 관계를 가져야 한다.

$$D_u(i) = D_d(i) \equiv X(i) \quad (3)$$

또한, 서버 $M_u(i)$ 의 출발간 평균시간과 서버 $M_u(i+1)$ 의 출발간 평균시간은 동일하므로, 식 (3)에 의해서 다음과 같은 관계가 성립한다.

$$\frac{1}{D_d(i)} = \frac{1}{D_u(i+1)} \quad (4)$$

$$X(i) = X(i+1) \quad (5)$$

결국 식 (3)과 식 (5)를 조합하면 다음과 같은 흐름보존방정식을 만족하게 된다 (Bouhchouch *et al.*).

$$X(1) = \dots = X(i) = X(i+1) = \dots = X(K) \quad (6)$$

CQN-BF의 가장 큰 특징으로는 시스템 내에 존재할 수 있는 고객의 수가 제한되어 있다는 것이다. 이는 곧 어떤 한 시점에서 서버 $M(i)$ 와 버퍼 $B(i)$ 로 구성되는 각 스테이션에 존재하는 고객들의 합은 총 고객의 수 N 이 되어야 한다는 것이다. 이를 하부시스템 모형에 반영하기 위해서, 각 하부시스템 $L(i)$ 에 존재하는 평균 고객 수를 $Q(i)$ 라고 하면 $Q(i)$ 들의 합은 원 시스템 L 에 존재하는 총 고객 수 N 과 같아야 한다는 조건이 추가된다 (Bouhchouch; Frein *et al.*, 1996; Lee and Pollock, 1990).

$$\sum_{i=1}^K Q(i) = N \quad (7)$$

본 연구에서는 분해된 하부시스템을 분석하기 위해 세 가지의 서로 다른 방법들을 사용하였으며, 분석방법에 따라 알고

리즘 I, 알고리즘 II 및 알고리즘 III라고 칭하였다. 알고리즘 I은 이중서버(2-server)로 구성된 시스템의 행태를 마코비안 상태로 표현하여 정확한 안정상태확률을 구할 수 있는 확률분포를 도출해낸 Gershwin and Berman (1981)의 해법을 사용하는 것으로서, 하부시스템에 대한 정확한 해를 도출하는 방법이다. 알고리즘 II는 하부시스템 $L(i)$ 를 $M/M/1/C$ 로 근사화하여 분석하는 것으로서, 각 서버의 평균 서비스시간에 서버의 고장 및 수리, 그리고 봉쇄 및 결핍으로 인한 지연시간을 반영하는 방법이다. 마지막으로, 알고리즘 III가 1차 모멘트인 평균만을 사용하는 반면, 알고리즘 III는 각 서버의 서비스시간에 대한 1차 및 2차 모멘트를 이용하여 하부시스템의 행태를 $PH/PH/1/C$ 로 근사화하여 분석하는 방법이다.

3.2 알고리즘 I

먼저, 분해된 하부시스템에서 상위 및 하위서버의 매개변수인 고장을 및 수리율, 그리고 서비스율을 구하기 위한 관계를 유도한다.

하부시스템 $L(i)$ 에서 상위서버 $M_u(i)$ 및 하위서버 $M_d(i)$ 의 고장을 및 수리율은 각각 원 시스템 L 에서 M_i 및 M_{i+1} 의 고장을 및 수리율이다. 다시 말하면, $\lambda_d(i) = \lambda_i$, $\lambda_u(i) = \lambda_{i+1}$ 이고 $\gamma_d(i) = \gamma_i$, $\gamma_u(i) = \gamma_{i+1}$ 이다. 이때 만약 $i=K$ 이면 $i+1=1$ 이 된다.

하부시스템에서 서버 $M_u(i)$ 와 $M_d(i)$ 의 서비스과정은 봉쇄 또는 결핍시 발생하는 지연시간을 반영하게 된다. 먼저, 서버 $M_u(i)$ 는 시스템 L 에서 버퍼 B_i 의 상위부분의 행태를 나타낸다. 따라서 서버 $M_u(i)$ 의 평균 서비스시간 $1/\mu_u(i)$ 는 버퍼 B_i 로 고객이 이전되는 시점으로부터 서버 M_i 가 다음 고객에 대한 서비스를 종료했을 때까지의 시간에 해당된다. 이 시간은 서버 M_i 의 서비스시간과 결핍되는 시간이 포함되어 있다.

서버 M_i 가 서비스를 종료했을 때, 만약 버퍼 B_{i-1} 이 비어있다면 결핍이 발생된다. 이러한 결핍현상은 서버 M_{i-1} 이 하나의 고객 서비스를 종료했을 때에야 해제될 수 있으며, 이는 하부시스템 $L(i-1)$ 에서 서버 $M_d(i-1)$ 이 고객 서비스를 종료했을 때까지의 시간에 해당된다. 그리고 원 시스템 L 에서 서버 M_i 의 결핍은 하부시스템 $L(i-1)$ 에서 서버 $M_u(i-1)$ 의 결핍에 해당되므로, 서버 M_{i-1} 의 결핍될 확률은 서버 $M_d(i-1)$ 이 결핍될 확률로 근사화할 수 있으며 이를 $P(i-1)$ 이라 한다. 따라서 $M_u(i)$ 의 평균 서비스시간 $1/\mu_u(i)$ 는 다음과 같이 계산될 수 있다.

$$\frac{1}{\mu_u(i)} = \frac{1}{\mu_i} + P_s(i-1) \frac{1}{\mu_u(i-1)} \quad (8)$$

유사하게, 서버 $M_d(i)$ 는 원 시스템 L 에서 버퍼 B_i 의 하위부분의 행태를 나타내는 것이다. $M_d(i)$ 의 평균 서비스시간 $1/\mu_d(i)$ 는 서버 M_{i+1} 이 고객에 대해 서비스를 시작하는 시점으로부터 버퍼 B_{i+1} 로 고객을 이전하는 시점까지의 시간에 해당된다. 이 시간은 서버 M_{i+1} 의 서비스시간과 봉쇄되는 시간이 포함되어

있다.

서버 M_{i+1} 이 고객 서비스를 종료했을 때, 만약 버퍼 B_{i+1} 이 가득 차 있다면 봉쇄가 발생하게 된다. 이러한 봉쇄현상은 서버 M_{i+2} 가 하나의 고객 서비스를 종료했을 때 해제될 수 있으며, 이는 하부시스템 $L(i+1)$ 에서 서버 M_{i+1} 이 하나의 고객 서비스를 종료했을 때까지의 시간에 해당된다. 원 시스템 L 에서 서버 M_{i+1} 의 봉쇄는 하부시스템 $L(i+1)$ 에서 서버 M_{i+1} 의 봉쇄에 해당되기 때문에, 서버 M_{i+1} 의 봉쇄될 확률은 서버 M_{i+1} 이 봉쇄될 확률로 근사화할 수 있다. 이를 $P_b(i+1)$ 이라 한다. 따라서 M_{i+1} 의 평균 서비스시간 $1/\mu_{i+1}$ 는 다음과 같이 계산될 수 있다.

$$\frac{1}{\mu_{i+1}} = \frac{1}{\mu_i} + P_b(i+1) \frac{1}{\mu_{i+1}} \quad (9)$$

하부시스템 $L(i)$ 의 상위서버 및 하위서버의 매개변수인 서버 수 및 고장을 그리고 수리율이 주어지면, 봉쇄확률 $P_b(i)$ 와 결핍확률 $P_s(i)$, 그리고 안정상태확률을 구할 수 있다. 먼저 하부시스템의 상태를 (n, α_1, α_2) 와 같이 정의하자. 이때 $n=0, 1, \dots, C(i)+1$ 이고, α_1 과 α_2 는 각각 상위서버 M_{i+1} 와 하위서버 M_i 의 고장유무를 나타내는 것으로, 고장상태에 있으면 1이고 아니면 0이다. 그리고 $n \leq C(i)$ 인 상태에서 n 은 버퍼 내의 고객의 수가 되고, $n = C(i)+1$ 은 버퍼 $B(i)$ 가 가득 차서 상위서버 M_{i+1} 가 봉쇄된 것을 뜻한다. 그러면 하부시스템 $L(i)$ 가 상태 (n, α_1, α_2) 에 있을 안정상태 확률은 $p_i(n, \alpha_1, \alpha_2)$ 와 같이 정의되며, Gershwins and Berman(1981)이 제시하였던 이종서버시스템에 대한 분석법에 의해 안정상태확률이 구해질 수 있다. 그러면 봉쇄확률과 결핍확률, 그리고 평균 고객수 $Q(i)$ 는 다음과 같이 계산된다.

$$P_b(i) = p_i(C(i)+1, 1, 0) + p_i(C(i)+1, 1, 1) \quad (10)$$

$$P_s(i) = p_i(0, 0, 1) + p_i(0, 1, 1) \quad (11)$$

$$Q(i) = \begin{cases} \sum_{n=0}^N \sum_{\alpha_1=0}^1 \sum_{\alpha_2=0}^1 n \cdot p_i(n, \alpha_1, \alpha_2), & C(i) \geq N \\ \sum_{n=0}^{C(i)} \sum_{\alpha_1=0}^1 \sum_{\alpha_2=0}^1 n \cdot p_i(n, \alpha_1, \alpha_2) \\ + \sum_{\alpha_1=0}^1 \sum_{\alpha_2=0}^1 C(i) \cdot p_i(C(i)+1, \alpha_1, \alpha_2), & C(i) < N \end{cases} \quad (12)$$

알고리즘 I 은 두 개의 반복적 루프, 내부 루프와 외부 루프로 구성되어 있다. 내부 루프에서는 각각의 매개변수들의 값을 결정하는 루프이고, 외부 루프는 식 (7)의 조건을 만족하는 μ_{i+1} 값을 결정하는 루프이다.

내부 루프는 다시 두 개의 단계로 구성되는데, 첫 번째 단계에서는 전진방향으로 각 시스템들을 분석하여 상위서버 M_{i+1} 의 평균 서비스시간 $1/\mu_{i+1}$ 을 구하는 단계이다. 두 번째 단

계에서는 후진방향으로 각 시스템들을 분석하여 하위서버 M_i 의 평균 서비스시간 $1/\mu_i$ 을 구하는 단계이다. 이러한 단계들을 각 매개변수들이 수렴될 때까지 반복하게 되며, 내부 루프가 끝나면 외부 루프에서 Q 의 값을 구하여 N 과 비교하게 된다. 이때 만일 $Q > N$ 이면 μ_{i+1} 의 값을 적절히 줄여주고 $Q < N$ 이면 μ_{i+1} 의 값을 적절히 높여주게 된다. 이러한 과정을 Q 의 값이 N 에 만족할 만큼 근접할 때까지 반복 수행하게 된다. 이상의 과정들을 정리한 알고리즘 I 은 다음과 같다.

초기화

매개변수의 초기값을 다음과 같이 설정한다.

$$\mu_{i+1}(1)_{\min} = 0, \mu_{i+1}(1)_{\max} = \mu_i$$

$$\mu_{i+1}(i) = \mu_{i+1}, i = 1, 2, \dots, K$$

$$\lambda_{i+1}(i) = \lambda_{i+1}, \lambda_{i+1}(i) = \lambda_{i+1}, i = 1, 2, \dots, K$$

$$r_{i+1}(i) = r_{i+1}, r_{i+1}(i) = r_{i+1}, i = 1, 2, \dots, K$$

외부 루프

1. 다음을 이용하여 새로운 $\mu_{i+1}(1)$ 을 결정한다.

$$\mu_{i+1}(1) = \frac{\mu_{i+1}(1)_{\min} + \mu_{i+1}(1)_{\max}}{2}$$

2. 내부 루프: 주어진 $\mu_{i+1}(1)$ 에 대해서 각 매개변수의 값을 결정한다.

· 단계 1: $i = 1, 2, \dots, K$ 에 대해서

1.1: 시스템 $L(i)$ 를 분석하여 안정상태확률

$p_i(n, \alpha_1, \alpha_2)$ 을 계산한다.

이때, $n = 0, 1, \dots, C(i)+1, \alpha_1 = 0, 1, \alpha_2 = 0, 1$

1.2: 식(11)을 이용하여 결핍확률 $P_s(i)$ 를 계산한다.

1.3: 식(8)을 이용하여 $\mu_{i+1}(i)$ 을 계산한다.

이때, 만약 $(i+1)$ 이 1이라면 통과한다.

· 단계 2: $i = K, K-1, \dots, 1$ 에 대해서

2.1: 시스템 $L(i)$ 를 분석하여 안정상태확률

$p_i(n, \alpha_1, \alpha_2)$ 을 계산한다,

$n = 0, 1, \dots, C(i)+1, \alpha_1 = \alpha_2 = 0, 1, 2$

2.2: 식(10)을 이용하여 봉쇄확률 $P_b(i)$ 를 계산한다.

2.3: 식(9)을 이용하여 $\mu_{i+1}(i-1)$ 을 계산한다.

각 매개변수들이 수렴할 때까지 단계 1과 단계 2를 반복 수행한다. 각 매개변수들을 참조할 때에는 가장 최근에 생성된 값을 사용한다.

3. 다음 식을 이용하여 평균 고객 수준을 계산한다.

$$Q = \sum_{i=1}^K Q(i)$$

4. Q 가 N 에 만족할 만큼 근접했으면 알고리즘을 중단하고, 그렇지 못한 경우는 다음과 같이 설정하고 외부루프의 1.로 되돌아간다.

$$Q > N \text{이면 } \mu_u(1)_{\max} = \mu_u(1)$$

$$Q < N \text{ 이면 } \mu_u(1)_{\min} = \mu_u(1)$$

3.3 알고리즘 II

알고리즘 II는 전술한 바와 같이 분해된 하부시스템을 $M/M/1/C(i)+1$ 모형으로 근사화하여 분석하는 방법으로서, 각 서버의 서비스과정에 고장 및 수리, 그리고 봉쇄 및 결핍으로 인한 지연시간들을 반영하게 된다.

먼저, 앞선 가정에서 서버의 고장은 서비스 중에만 발생할 수 있다고 하였다. 따라서 고장으로 인하여 발생하는 시간들은 서비스를 지연시키는 시간들로 간주될 수 있다. 이러한 서비스 지연시간에는 고장 발생시의 평균 수리시간($1/\gamma_i$)과 고장나기까지의 평균 서비스시간($1/\mu_i'$)이 포함된다.

서비스 지연시간의 발생형태를 살펴보면, 서버 M_i 가 한 고객에 대한 서비스를 실시하는 도중, 서버는 각각의 발생 확률에 따라 k 번의 고장이 일어날 수 있다 ($k = 0, 1, \dots$). 따라서, <그림 3>에서와 같이 서비스 중 발생하는 고장횟수 (k)에 따라 상응하는 서비스 지연시간들이 추가되는 형태를 갖는다.

이러한 지연시간이 포함되어 있는 서비스시간을 실질(effective)서비스시간이라 하고, 그 평균을 $1/\nu(i)$ 라고 정의한다. 그러면 서버 M_i 의 평균 실질서비스시간은 다음과 같이 서비스시간에 평균 지연시간이 추가되는 형태로 구성된다. 이때, 평균 지연시간은 평균 고장 발생횟수와 서비스 지연시간의 곱으로 이루어진다.

$$\frac{1}{\nu(i)} = \frac{1}{\mu_i} + \frac{\lambda_i}{\mu_i} \left(\frac{1}{\mu_i'} + \frac{1}{\gamma_i} \right), \quad i = 1, 2, \dots, K \quad (13)$$

한편, 고장나기까지의 평균 서비스시간, $1/\mu_i'$ 는 다음과 같은 조건부 기대값으로 계산될 수 있다. 여기에서 F 는 고장나기까지의 시간을 나타내는 확률변수이며, S 는 서비스시간을 나타내는 확률변수이다.

$$\frac{1}{\mu_i'} = E[F | F < S]$$

$$= \frac{1}{\lambda_i} \int_0^\infty \frac{\mu_i e^{-\mu_i t}}{1 - e^{-\lambda_i t}} dt - \int_0^\infty \frac{(1 + \lambda_i t) / (\lambda_i e^{-\lambda_i t})}{1 - e^{-\lambda_i t}} \mu_i e^{-\mu_i t} dt$$

그러면 하부시스템 $L(i)$ 에서 상위 및 하위서버의 평균 서비스시간은 앞서 알고리즘 I에서 언급된 바와 같이 평균 실질서비스시간과 봉쇄 및 결핍시간의 합으로 구성된다.

$$\frac{1}{\mu_u(i)} = \frac{1}{\nu(i)} + P_s(i-1) \frac{1}{\mu_u(i-1)} \quad (14)$$

$$\frac{1}{\mu_d(i)} = \frac{1}{\nu(i+1)} + P_s(i+1) \frac{1}{\mu_d(i+1)} \quad (15)$$

이제, 하부시스템 $L(i)$ 의 상태를 n 이라 하고, $p_n(n)$ 를 시스템 $L(i)$ 가 상태 n 에 있을 안정상태 확률이라 정의하자. $n = 1, 2, \dots, C(i), C(i)+1$ 이며, $n \leq C(i)$ 인 상태에서 n 은 버퍼 내의 고객의 수가 되고, $n = C(i)+1$ 은 버퍼 $B(i)$ 가 가득 차서 상위서버 $M_u(1)$ 가 봉쇄된 것을 뜻한다.

이상으로부터 하부시스템 $L(i)$ 의 상위서버 및 하위서버의 평균 서비스시간이 주어지면, 하부시스템 $L(i)$ 의 도착율과 서버

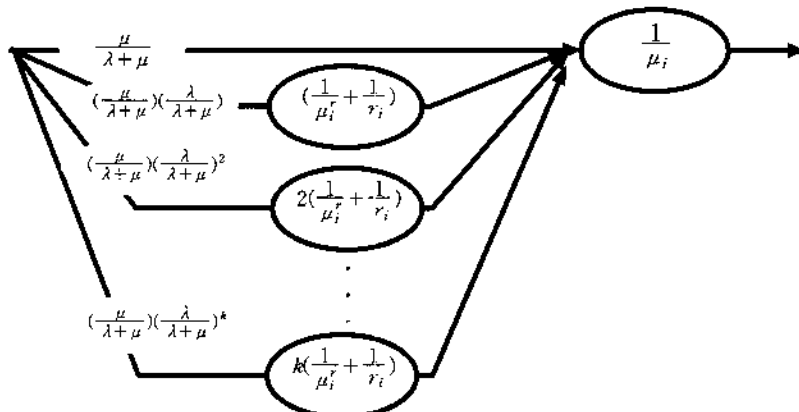


그림 3. 고장발생시의 서비스시간의 지연현상.

스율은 각각 상위서버의 서비스율과 하위서버의 서비스율이 된다. 따라서 하부시스템 $L(i)$ 의 안정상태 확률 $p_i(n)$ 은 $M/M/1/C(i)+1$ 공식을 통해서 쉽게 얻어질 수 있다. 그러면 $P_b(i)$, $P_s(i)$, 그리고 평균 고객 수 $Q(i)$ 는 다음과 같이 구해진다.

$$P_b(i) = \frac{p_i(C(i))}{1 - p_i(C(i)+1)} \tag{16}$$

$$P_s(i) = \frac{p_i(1)}{1 - p_i(0)} \tag{17}$$

$$Q(i) = \begin{cases} \sum_{n=0}^N n \cdot p_i(n), & C(i) \geq N \\ \sum_{n=0}^{C(i)} n \cdot p_i(n) + C(i) \cdot p_i(C(i)+1), & C(i) < N \end{cases} \tag{18}$$

본 절에서 소개되는 알고리즘 II 또한 알고리즘 I 과 전체적인 형태는 동일하다. 그러나 하부시스템 분석에서 약간의 차이가 있으므로 달라진 몇 가지만 따로 소개하기로 한다.

- ① 초기화 단계에서 식 (13)을 이용하여 각 서버의 평균 실질서비스시간 $1/\nu(i)$ 을 구한다.
- ② 내부루프 중 단계 1의 1.2에서 결핍확률 $P_b(i)$ 를 계산함에 있어 식 (17)을 사용한다.
- ③ 내부루프 중 단계 1의 1.3에서 $\mu_d(i+1)$ 을 계산함에 있어 식 (14)를 이용한다.
- ④ 내부루프 중 단계 2의 2.2에서 봉쇄확률 $P_b(i)$ 를 계산함에 있어 식 (16)을 사용한다.
- ⑤ 내부루프 중 단계 2의 2.3에서 $\mu_d(i-1)$ 을 계산함에 있어 식 (15)를 이용한다.

3.4 알고리즘 III

알고리즘 II를 비롯한 폐쇄형 봉쇄 대기행렬망의 분석을 위한 대부분의 연구에서는 서비스시간에 대한 1차 모멘트인 평균만을 사용하여 분석하고 있다. 그러나 최근 연구(Bouchouch; Kim and Lie, 1996)에서 분해된 하부시스템의 서비스과정을 분석함에 있어 2차 모멘트 이상의 사용을 제안했다. 알고리즘 III는 서비스시간에 대한 1차 및 2차 모멘트를 이용하여 하

부시스템 $L(i)$ 의 서비스과정을 2단계를 갖는 PH(phase-type)형태로 변환하고, 이를 $PH/PH/1/C(i+1)$ 로 분석하여 안정상태확률을 구한 방법이다. 먼저, 알고리즘 III에서 사용되는 평균 실질서비스시간 $1/\nu(i)$ 은 알고리즘 II에서와 같이 식 (13)으로 구해질 수 있다. 또한 상위 및 하위서버 $M_b(i)$ 와 $M_d(i)$ 의 평균 서비스시간, $1/\mu_d(i)$ 와 $\mu_d(i)$ 은 식 (14)와 (15)로 구해질 수 있다. 이어서, 각 서버의 평균 실질서비스시간과 상위 및 하위서버의 평균 서비스시간에 대한 분산을 각각 $\sigma_b^2(i)$, $\sigma_d^2(i)$, 그리고 $\sigma_d^2(i)$ 라 하면, 서비스시간에 대한 확률밀도함수의 1차 및 2차 모멘트에 의해 각각 다음과 같이 된다.

$$\sigma_b^2(i) = \frac{1}{\mu_i^2} + \frac{\lambda_i}{\mu_i} \left(\frac{1}{(\mu_i^r)^2} + \frac{1}{r_i^2} \right) + \frac{2\lambda_i(\mu_i + \lambda_i) + \lambda_i \mu_i^2(\mu_i + \lambda_i)}{\mu_i^2} \left(\frac{1}{\mu_i^r} + \frac{1}{r_i} \right)^2 \tag{19}$$

$$\sigma_u^2(i) = \sigma_{v,u}^2(i) + P_s(i-1) \left(\sigma_u^2(i-1) + (1 - P_s(i-1)) \frac{1}{\mu_d(i-1)} \right) \tag{20}$$

$$\sigma_d^2(i) = \sigma_{v,d}^2(i) + P_b(i+1) \left(\sigma_d^2(i+1) + (1 - P_b(i+1)) \frac{1}{\mu_d(i+1)} \right) \tag{21}$$

각 서비스시간에 대한 평균 및 분산이 구해지면, 2단계 (2-phase)를 갖는 PH형태로 변환하여 $PH/PH/1/C(i+1)$ 로서 하부시스템 $L(i)$ 에 대한 분석을 하게 된다. PH형태로의 변환 방법은 Sauer and Reddy(1975)와 Marie(1980)에 의해 사용되었던 방법과 동일한 방법을 사용하였다.

<그림 4>와 같이 2단계를 갖는 PH형태로 변환되어지면, 하부시스템 $L(i)$ 의 상태를 (α_1, α_2, n) 로 정의할 수 있다. 그러면 $n=0, 1, \dots, C(i)+1$ 로서, $n \leq C(i)$ 인 상태에서 n 은 버퍼 내의 고객의 수가 되고, $n = C(i) + 1$ 인 상태는 버퍼 $B(i)$ 가 가득 차서 서버 $M_d(i)$ 가 봉쇄된 것을 뜻한다. α_1 과 α_2 는 각각의 단계에서 서비스 받고 있는 상태를 나타내며, 버퍼 $B(i)$ 가 가득 차 있거나 또는 비어 있어서 각 서버가 대기하고 있는 상태를 0이라 한다. 따라서 $\alpha_1 = \alpha_2 = 0, 1, 2$ 이다. 이때 총 발생

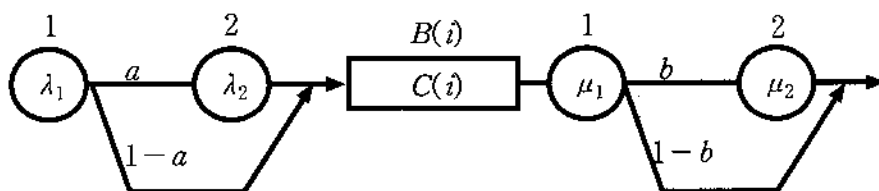


그림 4. 2단계 PH형태로 변환된 하부 시스템 $L(i)$.

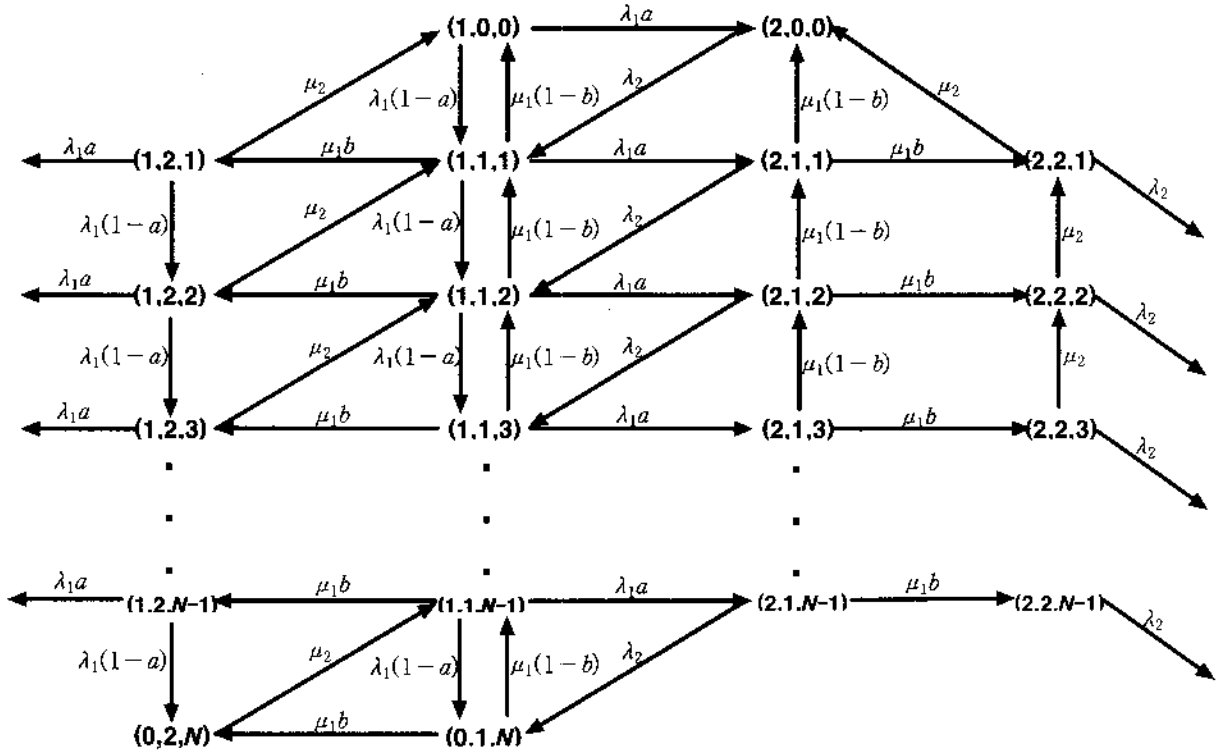


그림 5. PH/PH/1/C(i)+1에 대한 상태 전이도.

가능한 상태의 수는 $4 * (C(i) + 1)$ 개가 된다. 정의된 상태변수를 가지고 시스템 상태간의 전이를 나타내는 상태전이도를 그려보면 <그림 5>와 같다. 이를 이용하여 시스템의 각 상태에 대한 안정상태확률 $p_i(a_1, a_2, n)$ 을 구할 수 있게 된다. 그러면 봉쇄확률($P_b(i)$)과 결핍확률($P_s(i)$), 그리고 평균 고객수준($Q(i)$)은 다음과 같이 구하여 질 수 있다.

$$P_b(i) = p_i(0, 2, C(i) + 1) + p_i(0, 1, C(i) + 1) \quad (22)$$

$$P_s(i) = p_i(1, 0, 0) + p_i(2, 0, 0) \quad (23)$$

$$Q(i) = \begin{cases} \sum_{n=0}^N \sum_{a_1=0}^2 \sum_{a_2=0}^2 n \cdot p_i(a_1, a_2, n), & C(i) \geq N \\ \sum_{n=0}^{C(i)} \sum_{a_1=0}^2 \sum_{a_2=0}^2 n \cdot p_i(a_1, a_2, n) \\ + \sum_{a_1=0}^2 \sum_{a_2=0}^2 C(i) \cdot p_i(a_1, a_2, C(i) + 1), & C(i) < N \end{cases} \quad (24)$$

알고리즘 III 또한 알고리즘 I, II와 동일한 형태로 구성된다. 그러나 알고리즘 II에 비하여 하부시스템의 매개변수인 서비스시간에 대한 분산과 PH형태로의 변환과정이 추가되었

으므로 '초기화' 단계와 외부루프 중 '2. 내부루프' 단계가 다음과 같이 조정된다.

초기화

1. $i=1, 2, \dots, K$ 에 대해서 식 (13)과 (19)를 이용하여 $\nu(i)$ 와 $\sigma_v^2(i)$ 를 계산한다.
2. 매개변수의 초기값을 다음과 같이 설정한다.

$$\mu_u(1)_{\min} = 0, \mu_u(1)_{\max} = \nu(1)$$

$$\sigma_u(1) = \sigma_v^2(1)$$

$$\mu_d(i) = \nu(i+1), i=1, 2, \dots, K$$

$$\sigma_d(i) = \sigma_v^2(i+1), i=1, 2, \dots, K$$

$\mu_u(1)$ 과 $\sigma_u^2(1)$ 을 이용하여 PH형태로 변환한다.

외부 루프

2. 내부 루프 : 주어진 $\mu_u(1)$ 에 대해서 각 매개변수의 값을 결정한다.
 - 단계 1: $i=1, 2, \dots, K$ 에 대해서

1.1 : 시스템 $L(i)$ 를 분석하여 안정상태확률 $p(\alpha_1, \alpha_2, n)$ 을 계산한다,

이때, $n=0, 1, \dots, C(i)+1, \alpha_1=0, 1, 2, \alpha_2=0, 1, 2$

1.2: 식 (23)을 이용하여 결핍확률 $P_d(i)$ 를 계산한다.

1.3: 식 (14)와 (20)을 이용하여 $\mu_u(i+1)$ 와 $\sigma_u^2(i+1)$ 을 계산한다.

이때, 만약 $i+1$ 이 1이라면 $\sigma_u^2(1)$ 만 구한다.

1.4: $\mu_u(i+1)$ 과 $\sigma_u^2(i+1)$ 을 이용하여 PH형태로 변환한다.

· 단계 2: $i=K, K-1, \dots, 1$ 에 대해서

2.1: 시스템 $L(i)$ 를 분석하여 안정상태확률 $p(\alpha_1, \alpha_2, n)$ 을 계산한다.

이때, $n=0, 1, \dots, C(i)+1, \alpha_1=0, 1, 2, \alpha_2=0, 1, 2$

2.2: 식 (22)를 이용하여 봉쇄확률 $P_b(i)$ 를 계산한다.

2.3: 식 (15)와 (21)을 이용하여 $\mu_d(i-1)$ 과 $\sigma_d^2(i-1)$ 을 계산한다.

2.4: $\mu_d(i-1)$ 과 $\sigma_d^2(i-1)$ 을 이용하여 PH형태로 변환한다.

각 매개변수들이 수렴할 때까지 단계 1과 단계 2를 반복 수행한다. 각 매개변수들을 참조할 때에는 가장 최근에 생성된 값을 사용한다.

4. 산출률 상한

본 절에서는 표본경로 추적방법(sample path method)을 이용하여 CQN-BF에 대한 산출률의 상한을 구한다. 먼저 다음과 같은 기호를 정의한다.

$B_{i,k}$: i 번째 서버에서 k 번째 고객에 대한 서비스 시작 시간

$D_{i,k}$: i 번째 서버에서 k 번째 고객의 출발 시간

$S_{i,k}$: i 번째 서버에서 k 번째 고객에 대한 서비스를 시작해서 완료하기까지의 시간

(만약, 서비스 중 서버 고장시 수리시간 포함)

m_i : 초기상태 ($t=0$)에서 i 번째 버퍼에 존재하는 고객의 수

k 번째 고객에 대한 서비스 시작 시간 $B_{i,k}$ 는 두 가지 조건에 의해서 결정된다. 첫째는 서버 M_i 가 가용한지에 관한 것이고, 둘째는 서버 M_i 의 상위버퍼 B_{i-1} 이 비어있지 않은지에 관한 것이다. 첫 번째 조건은 서버 M_i 가 $k-1$ 번째 고객에 대한 서비스를 완료했을 때의 시간 $D_{i,k-1}$ 이 되며, 두 번째 조건은 서버 M_{i-1} 이 $k-m_{i-1}$ 번째 고객에 대한 서비스를 완료했을 때의 시간 $D_{i-1,k-m_{i-1}}$ 이 된다. 그러므로 이러한 조건을 모두 만

족한다면, k 번째 고객에 대한 서비스 시작 시간 $B_{i,k}$ 는 두 가지 시간 중 큰 것이 된다.

서버 M_i 에서 k 번째 고객의 출발 시간 $D_{i,k}$ 또한 두 가지 조건에 의해서 결정된다. 첫 번째 조건은 서버 M_i 가 k 번째 고객에 대해 서비스를 시작해서 완료하기까지의 시간으로서, $B_{i,k}+S_{i,k}$ 이 된다. 두 번째 조건은 서버 M_i 가 봉쇄되어 있는 지에 관한 것으로서, 이는 서버 M_{i+1} 이 $k-(C_i-m_i)$ 번째 고객에 대한 서비스를 완료했을 때의 시간 $D_{i+1,k-(C_i-m_i)}$ 이 된다. 따라서, 이러한 조건을 모두 만족한다면, k 번째 고객의 출발 시간 $D_{i,k}$ 는 두 가지 시간 중 큰 것이 된다. 이상을 정리하면 다음과 같다.

$$B_{i,k} = \max\{D_{i,k-1}, D_{i-1,k-m_{i-1}}\} \quad i=1, \dots, K \quad (25)$$

$$D_{i,k} = \max\{B_{i,k}+S_{i,k}, D_{i+1,k-(C_i-m_i)}\}, \quad i=1, \dots, K, k \geq 1 \quad (26)$$

이때, $k < 1$ 에 대해서 $D_{i,k}=0$ 이다. 한편, $D_{i,k}$ 는 식 (25)와 (26)에서 보여진 바와 같이 서비스시간 $S_{i,k}$ 에 대해서 증가함수이고, 버퍼용량 C_i 에 대해서 감소함수이므로 다음과 같은 정리가 성립된다.

Lemma 1. 버퍼용량이 $C_i=C'_i$ 이고, 초기조건이 $m_i=m'_i$ 인 두 개의 CQN-BF를 고려하자. 만약 서비스시간이 $S_i \leq S'_i$ 이라면, 출발시간은 $D_{i,k} \leq D'_{i,k}$ 이다 ($i=1, \dots, K$).

이때, S_i 는 i 번째 서버에서의 서비스시간 분포를 나타내는 확률변수이며 $S_i \leq S'_i$ 는 $x \in R$ 에 대해서 $P(S_i \geq x) \leq P(S'_i \geq x)$ 임을 나타낸다(Ross, 1983).

Lemma 2. 서비스시간이 $S_i = S'_i$ 이고, 초기조건이 $m_i=m'_i$ 인 두 개의 CQN-BF를 고려하자. 만약, 버퍼용량이 $C_i \leq C'_i$ 이라면, 출발시간은 $D_{i,k} \leq D'_{i,k}$ 이다 ($i=1, \dots, K$).

Lemma 1과 Lemma 2에 대한 자세한 증명은 Paik (1997)을 참조하기 바란다. 또한, CQN-BF를 L 이라 하고, 그 산출률을 $X(L)$ 이라 한다면 다음과 같은 정리 1이 성립된다.

정리 1. 초기조건이 $m_i=m'_i$ 인 두 개의 CQN-BF를 고려하자. 만약, 서비스시간이 $S_i \leq S'_i$ 이거나 또는 버퍼용량이 $C_i \leq C'_i$ 이라면, 그들의 산출률은 $X(L)$

$\leq X(L')$ 이다.

증명 : 산출률은 다음과 같이 정의될 수 있으므로, Lemma 1과 Lemma 2에 의해서 증명은 성립된다.

$$X(L) = \lim_{k \rightarrow \infty} \frac{k}{E(D_{i,k})}, \quad i=1, \dots, K.$$

이제 산출률의 상한에 대한 관계식을 유도해 보기로 한다. 먼저 원 시스템 L 에서 i 번째 버퍼 B_i 를 중심으로 상위서버인 M_i 와 하위서버인 M_{i+1} 를 제외한 나머지 모든 서버 및 버퍼의 서비스시간과 저장용량을 각각 $S_j=I$ 와 $C_j=\infty$ 라 두게 되면, 이는 결과적으로 이종서버 시스템과 같은 형태가 된다(이때, $j \neq i, i+1, I \neq i$). 단, 시스템 내의 총 고객 수 N 은 상위서버인 M_i 가 결핍되는 현상을 없애기 위해서 $N > C_i$ 이어야 하며, I 는 0에서 단위(unit) 단계함수이다. 이러한 이종서버 시스템을 $L(i)$ 라 하면, $L(i)$ 에 대한 정확한 해가 구해질 수 있다. 따라서 $L(i)$ 의 산출률을 $X(L(i))$ 라고 한다면, 원 시스템 L 의 산출률과의 관계는 다음과 같은 정리가 성립한다.

정리 2. CQN-BF인 원 시스템 L 에 대해서, $N > C_i$ 인 조건을 만족하는 i 번째 이종서버 시스템 $L(i)$ 의 산출률, $X(L(i))$ 는 원 시스템 L 의 산출률 상한이 된다. 따라서 다음과 같은 식이 만족된다.

$$X(L) \leq \min\{ X(L(i)) \}, \quad i=1, \dots, K$$

증명 : 임의의 확률변수에 대한 분포함수 F 에 대해서, $F \geq I$ 이기 때문에, $X(L(i))$ 는 정리1에 의해서

$X(L)$ 의 상한이 된다(Stoyan, 1983).

5. 실험 결과

본 연구에서 제시한 알고리즘의 적정성을 검토하고, 그 성능을 평가하기 위하여 다음과 같은 실험을 수행하였다. 실험은 모두 4가지의 그룹(A, B, C, D)으로 나누어 수행되었으며, D 그룹은 다시 3가지의 하위그룹으로 나뉘어 수행되었다.

A그룹은 <표 1>에 나타나 있는 바와 같이 버퍼용량 및 서비스율, 수리율이 같으면서 고장률을 각각 0.01, 0.1, 0.5로 증가시킨 경우이다. <표 2>는 버퍼용량, 고장률, 그리고 수리율이 같으면서 서비스율만을 변화시킨 경우로서 이를 B그룹이라 한다. <표 3>은 고장률 및 수리율, 서비스율이 같으면서 각 스테이션 간 버퍼용량의 변화폭이 다른 경우로서 이를 C그룹이라 하며, 이때 버퍼용량의 변화폭은 모두 같은 경우, 1씩 차이가 나는 경우, 최대 5까지 차이가 나는 경우로 구분되었다. <표 4>는 모든 매개변수들의 값이 서로 다른 경우로서 이를 D그룹이라 한다. D그룹은 다시 3가지의 하위그룹들로 나뉘어 지는데 각 하위그룹들은 서버의 수가 4개, 7개, 그리고 10개인 경우로 구성되며, 각각 2가지의 서로 다른 데이터 군을 갖도록 하였다.

각각의 데이터 군에 대한 실험은 총 고객 수 N 을 변화시키며 수행되었으며, 봉쇄가 발생할 수 있는 N 의 최저개수 ($N \geq C_{\min}$)로부터 버퍼 저장용량의 합인 C 까지 5 또는 10씩 증가시키면서 수행되었다. 시뮬레이션은 주어진 매개변수에

표 1. 그룹 A: 서비스율 및 수리율이 같으면서 고장률이 증가하는 경우

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	N	알고리즘 I	오차	알고리즘 II	오차	알고리즘 III	오차	시뮬레이션	상한
A ₁	C_i	7	7	7	7	7	7	10	0.3617	2.38	0.3867	9.45	0.3509	0.68	0.3533	0.4705
	λ_i	0.01	0.01	0.01	0.01	0.01	0.01	15	0.4260	1.23	0.4461	6.01	0.4114	2.23	0.4208	
	r_i	0.1	0.1	0.1	0.1	0.1	0.1	20	0.4600	6.06	0.4760	7.06	0.4446	2.84	0.4337	
	μ_i	1.0	0.7	0.67	1.0	0.8	0.7	25	0.4718	2.36	0.4859	5.42	0.4640	0.67	0.4609	
								30	0.4707	2.48	0.4857	5.75	0.4634	0.89	0.4593	
								35	0.4538	1.54	0.4739	2.82	0.4438	3.71	0.4609	
								40	0.4056	0.37	0.4336	7.3	0.3959	2.03	0.4041	
A ₂	C_i	7	7	7	7	7	7	10	0.0611	6.81	0.0643	12.41	0.0574	0.35	0.0572	0.0793
	λ_i	0.5	0.5	0.5	0.5	0.5	0.5	15	0.0758	5.87	0.0744	3.91	0.0679	5.17	0.0716	
	r_i	0.1	0.1	0.1	0.1	0.1	0.1	20	0.0775	6.38	0.0797	9.33	0.0749	2.74	0.0729	
	μ_i	1.0	0.7	0.67	1.0	0.8	0.7	25	0.0853	9.5	0.0817	4.88	0.0780	0.13	0.0779	
								30	0.0777	4.09	0.0816	9.24	0.0775	3.75	0.0747	
								35	0.0767	4.02	0.0792	7.32	0.0731	0.95	0.0738	
								40	0.0675	5.83	0.0723	13.32	0.0637	0.16	0.0638	
A ₃	C_i	7	7	7	7	7	7	10	0.1752	2.58	0.1849	8.28	0.1697	0.64	0.1708	0.2330
	λ_i	0.1	0.1	0.1	0.1	0.1	0.1	15	0.2104	1.95	0.2136	4.02	0.1969	4.14	0.2054	
	r_i	0.1	0.1	0.1	0.1	0.1	0.1	20	0.2315	3.03	0.2284	1.65	0.2175	3.2	0.2247	
	μ_i	1.0	0.7	0.67	1.0	0.8	0.7	25	0.2401	2.69	0.2466	5.48	0.2273	2.78	0.2338	
								30	0.2383	2.69	0.2593	5.88	0.2255	7.92	0.2449	
								35	0.2245	3.17	0.2323	6.75	0.2199	1.06	0.2176	
								40	0.1945	3.35	0.1960	4.19	0.1830	2.76	0.1882	
							45	0.1408	0.49	0.1544	9.14	0.1321	6.64	0.1415		

표 2. 그룹 B: 각 매개변수들이 같으면서 버퍼크기의 변화폭이 다른 경우

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	N	알고리즘 I	오차	알고리즘 II	오차	알고리즘 III	오차	시뮬레이션	상한	
C_1	c_i	4	5	4	4	5	4	5	10	0.6335	2.43	0.6399	1.44	0.6480	0.20	0.6493	0.7628
	λ_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	15	0.7223	0.67	0.7383	2.89	0.7050	1.74	0.7175	
	r_i	0.5	0.5	0.5	0.5	0.5	0.5	0.5	20	0.7262	0.08	0.7538	3.88	0.7268	0.16	0.7256	
	μ_i	1.0	2.0	1.5	1.25	2.0	1.0	1.75	25	0.6443	1.6	0.6863	4.80	0.6616	1.04	0.6548	
									30	0.4683	6.09	0.4252	3.67	0.4428	0.32	0.4414	
C_2	c_i	7	7	7	7	7	7	7	10	0.6024	1.52	0.5717	6.53	0.6195	1.27	0.6117	0.8072
	λ_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	15	0.7078	0.87	0.6789	3.12	0.7036	0.27	0.7017	
	r_i	0.5	0.5	0.5	0.5	0.5	0.5	0.5	20	0.7627	8.69	0.7153	4.52	0.7323	2.26	0.7492	
	μ_i	1.0	2.0	1.5	1.25	2.0	1.0	1.75	25	0.7765	1.49	0.7325	4.26	0.7667	0.21	0.7651	
									30	0.7771	3.95	0.7214	3.50	0.7576	1.34	0.7476	
									35	0.7503	3.99	0.6989	4.85	0.7204	1.93	0.7346	
									40	0.7433	2.34	0.6800	6.37	0.7023	3.30	0.7263	
								45	0.6455	1.16	0.6197	2.88	0.6282	1.55	0.6381		
C_3	c_i	8	3	5	9	4	7	3	10	0.6421	0.97	0.6048	4.89	0.6321	0.6	0.6359	0.7289
	λ_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	15	0.7245	1.86	0.6802	4.37	0.7115	0.03	0.7113	
	r_i	0.5	0.5	0.5	0.5	0.5	0.5	0.5	20	0.7437	1.11	0.7216	1.88	0.7245	0.19	0.7355	
	μ_i	1.0	2.0	1.5	1.25	2.0	1.0	1.75	25	0.7321	3.11	0.7156	0.78	0.7036	0.90	0.7100	
									30	0.6653	0.02	0.6725	1.50	0.6566	0.06	0.6652	
								35	0.5850	2.13	0.6090	6.31	0.5701	0.47	0.5728		

표 3. 그룹 C: 고장률 및 수리율이 같으면서 서비스율이 다른 경우

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	N	알고리즘 I	오차	알고리즘 II	오차	알고리즘 III	오차	시뮬레이션	상한	
B_1	c_i	7	7	7	7	7	7	7	10	0.2083	1.56	0.2128	3.75	0.2023	1.36	0.2051	0.2732
	λ_i	0.05	0.05	0.05	0.05	0.05	0.05	0.05	15	0.2485	6.01	0.2461	4.99	0.2400	2.39	0.2344	
	r_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	20	0.2513	7.31	0.2565	5.33	0.2629	7.97	0.2435	
	μ_i	1.0	0.5	0.56	1.0	0.8	0.5	0.57	25	0.2693	6.85	0.2724	7.88	0.2725	7.92	0.2525	
									30	0.2676	7.55	0.2615	5.10	0.2704	8.68	0.2488	
									35	0.2529	6.17	0.2505	5.16	0.2556	7.3	0.2382	
									40	0.2298	0.13	0.2377	3.30	0.2236	2.82	0.2301	
								45	0.1889	8.25	0.1897	8.71	0.1653	5.27	0.1745		
B_2	c_i	7	7	7	7	7	7	7	10	0.4031	5.33	0.4128	7.86	0.3736	1.57	0.3827	0.5910
	λ_i	0.05	0.05	0.05	0.05	0.05	0.05	0.05	15	0.4866	8.04	0.4861	7.92	0.4430	1.64	0.4504	
	r_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	20	0.5340	2.73	0.5665	8.98	0.5042	3.0	0.5198	
	μ_i	1.0	2.0	1.5	1.25	2.0	1.0	1.75	25	0.5519	1.53	0.5724	5.29	0.5429	0.13	0.5436	
									30	0.5468	2.36	0.5715	6.90	0.5404	1.18	0.5342	
									35	0.5141	1.0	0.5405	6.10	0.5107	0.33	0.5090	
								40	0.5068	2.94	0.5177	5.16	0.4992	0.02	0.4923		
								45	0.3768	3.15	0.3697	1.20	0.3696	3.73	0.3653		
B_3	c_i	7	7	7	7	7	7	7	10	0.3222	10.0	0.3137	7.10	0.3017	3.0	0.2929	0.4709
	λ_i	0.05	0.05	0.05	0.05	0.05	0.05	0.05	15	0.3777	6.63	0.3739	5.56	0.3569	0.76	0.3542	
	r_i	0.1	0.1	0.1	0.1	0.1	0.1	0.1	20	0.4254	1.12	0.4363	3.70	0.4042	3.92	0.4207	
	μ_i	1.0	0.9	1.5	1.7	0.8	1.3	0.9	25	0.4401	0.2	0.4684	6.64	0.4194	4.51	0.4392	
									30	0.4358	1.07	0.4563	5.82	0.4140	3.99	0.4312	
									35	0.4099	0.76	0.4196	3.14	0.3877	4.7	0.4068	
								40	0.3542	0.91	0.3780	6.44	0.3332	6.17	0.3551		
								45	0.3283	4.12	0.3393	7.61	0.3269	3.68	0.3153		

표 4. 그룹 D: 모든 매개변수들의 값이 다른 경우

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	N	알고리즘 I	오차	알고리즘 II	오차	알고리즘 III	오차	시뮬레이션	상한
D ₁	c_i	6	6	6	6						10	0.3980	1.14	0.4086	3.84	0.3831	2.46	0.3935	0.4303
	λ_i	0.01	0.01	0.01	0.01						15	0.4100	0.02	0.4213	2.73	0.4113	0.29	0.4101	
	r_i	0.1	0.1	0.1	0.1						20	0.3714	1.78	0.3572	2.11	0.3654	0.14	0.3649	
	μ_i	0.5	1.0	0.57	0.8														
D ₂	c_i	8	7	7	7						10	0.6531	1.54	0.6187	6.86	0.6795	2.44	0.6633	0.7393
	λ_i	0.05	0.04	0.03	0.02						15	0.7079	0.14	0.6664	6.0	0.7095	0.08	0.7089	
	r_i	1.1	1.2	0.9	1.3						20	0.6866	1.04	0.6480	6.6	0.6920	0.26	0.6938	
	μ_i	1.0	0.8	1.1	0.9						25	0.5803	0.71	0.5360	6.98	0.5837	1.3	0.5762	
D ₃	c_i	7	8	7	6	9	7	9			10	0.4129	1.48	0.3714	11.38	0.4263	1.72	0.4191	0.4630
	λ_i	0.1	0.04	0.02	0.3	0.03	0.2	0.01			15	0.4744	7.33	0.4165	6.36	0.4406	0.94	0.4448	
	r_i	0.9	0.85	0.7	1.2	0.9	0.7	1.0			20	0.4981	8.16	0.4296	6.71	0.4601	0.09	0.4605	
	μ_i	0.8	0.9	0.5	0.8	1.67	1.2	1.0			25	0.4997	7.83	0.4309	7.01	0.4735	2.18	0.4634	
											30	0.4984	7.04	0.4310	7.43	0.4614	0.9	0.4656	
											35	0.4980	6.0	0.4309	8.28	0.4742	0.94	0.4698	
											40	0.4857	5.29	0.4294	6.92	0.4689	1.65	0.4613	
											45	0.4438	3.65	0.4155	9.79	0.4577	0.63	0.4606	
											50	0.3968	1.07	0.3784	3.62	0.4065	3.54	0.3926	
D ₄	c_i	8	7	8	7	8	7	8			10	0.5630	2.58	0.5366	7.15	0.5884	1.82	0.5779	0.7381
	λ_i	0.05	0.04	0.01	0.03	0.05	0.02	0.05			15	0.6571	0.59	0.6216	5.96	0.6822	3.21	0.6610	
	r_i	1.0	1.2	1.1	0.9	0.8	1.3	1.1			20	0.7101	2.07	0.6686	3.92	0.7276	4.59	0.6957	
	μ_i	1.0	0.8	1.2	1.1	1.2	0.9	1.0			25	0.7324	0.7	0.6894	5.21	0.7329	0.77	0.7273	
											30	0.7367	0.03	0.6942	5.74	0.7313	0.71	0.7365	
											35	0.7302	0.22	0.6877	5.61	0.7311	0.34	0.7286	
											40	0.6968	0.23	0.6576	5.84	0.7107	1.76	0.6984	
											45	0.6214	3.27	0.5867	8.67	0.6504	1.24	0.6424	
											50	0.4561	5.9	0.4274	7.1	0.4426	2.43	0.4307	
D ₅	c_i	7	7	7	7	7	7	7	7	7	10	0.5319	2.8	0.5258	3.91	0.5459	0.24	0.5472	0.7722
	λ_i	0.01	0.02	0.01	0.05	0.04	0.01	0.01	0.02	0.01	20	0.7118	0.42	0.6908	3.36	0.7118	0.42	0.7148	
	r_i	0.1	1.1	0.9	0.6	0.8	0.7	1.0	1.1	0.8	30	0.7710	0.97	0.7407	7.71	0.7604	0.42	0.7636	
	μ_i	1.0	1.2	1.0	0.9	1.6	1.5	1.7	0.9	1.6	40	0.7738	1.53	0.7443	2.33	0.7618	0.04	0.7621	
											50	0.7653	1.03	0.7422	2.02	0.7555	0.26	0.7575	
											60	0.6865	5.66	0.6192	4.69	0.6498	0.02	0.6497	
											65	0.5803	2.27	0.5271	7.1	0.5536	2.43	0.5674	
D ₆	c_i	8	7	8	7	8	7	8	7	8	10	0.4782	3.55	0.4671	5.79	0.5135	3.57	0.4958	0.7381
	λ_i	0.05	0.04	0.01	0.03	0.05	0.02	0.05	0.01	0.02	20	0.6573	1.16	0.6234	6.25	0.6779	1.94	0.6650	
	r_i	1.0	1.2	1.1	0.9	0.8	1.3	1.1	1.1	1.3	30	0.7306	1.79	0.6899	3.87	0.7364	2.61	0.7177	
	μ_i	1.0	0.8	1.2	1.1	1.2	0.9	1.0	1.2	0.9	40	0.7415	1.67	0.7028	3.63	0.7418	1.71	0.7293	
											50	0.6972	0.5	0.6937	0.0	0.6911	0.37	0.6937	
											60	0.6785	1.35	0.6355	7.6	0.6709	2.46	0.6878	
											70	0.5376	5.04	0.4717	7.83	0.5083	1.56	0.5118	

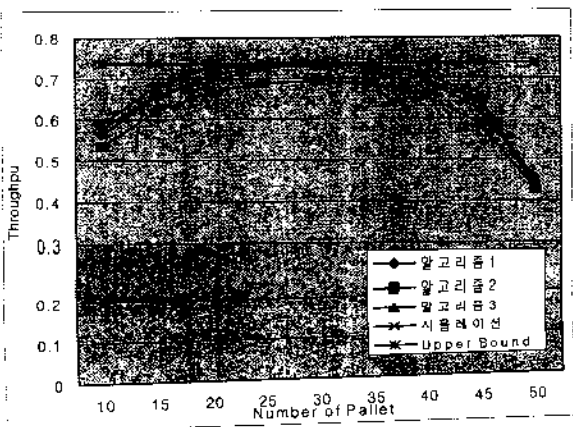


그림 6. N의 증가에 따른 산출물의 변화(D4).

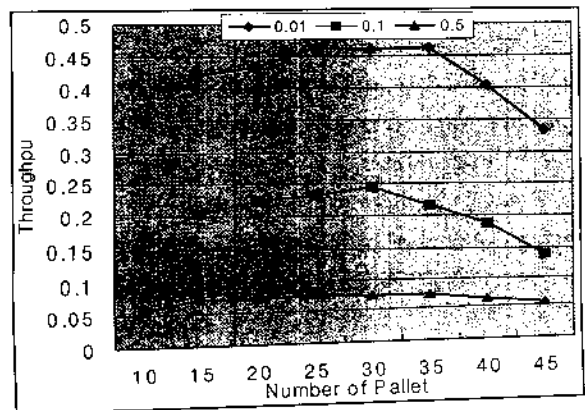


그림 7. 고장률의 증가에 따른 산출물의 변화.

따라 각각 10회씩을 수행하였다. 결과는 알고리즘 I, II, III에 대한 결과와 각각의 상대오차, 그리고 시뮬레이션 결과와 산출률의 상한에 대한 결과를 상호비교 할 수 있도록 배열하였다. 이때, 상대오차는 다음과 같은 식에 의해서 계산되었다.

$$\text{상대오차} = \frac{|\text{근사값} - \text{시뮬레이션}|}{\text{시뮬레이션}} \times 100(\%)$$

실험결과를 살펴보면 알고리즘 I, II, III는 시뮬레이션 결과와 비교하여 각각 평균 상대오차가 3.13%, 5.58%, 2.07%로 관측되었다. 이러한 결과를 볼 때 알고리즘 II는 다소 정확성이 떨어지나, 알고리즘 I과 III는 N에 관계없이 비교적 정확한 결과를 산출해 주고 있다.

구체적으로 살펴보면, 정확도의 측면에서 알고리즘 I은 거의 대부분의 경우에서 시뮬레이션 결과보다 산출률이 높게 측정되었으며, 고장률이 매우 커지게 되면 평균값에 비해 정확성이 다소 떨어지는 결과를 보인다. 반면, 알고리즘 II는 시뮬레이션 결과보다 산출률이 낮게 측정되었으며, 알고리즘 III는 거의 모든 경우에서 비교적 균등하고 정확한 결과를 산출해 주고 있다. 그러나 각 버퍼들의 저장용량이 서로 동일한 경우보다 다른 경우에서 모든 알고리즘의 정확도가 높은 것으로 나타났다.

한편, 계산량의 측면에서 보면 각 알고리즘들이 전체적인 구성 형태는 동일하지만, 하부시스템에 대한 분석방법이 다르므로 계산량에 차이가 생기게 된다. 즉, 각 매개변수들이 구해지면 알고리즘 II는 $M/M/1/C_i+1$ 의 공식에 의하여 쉽게 해를 구하게 되지만, 알고리즘 I은 4개의 식과 4개의 변수가 있는 연립방정식을 풀어야 하며(Gershwin and Bernam(1981) 참조), 알고리즘 III는 2단계에 갖는 PH분포로의 변환과정과 상태수가 $4*(C_i+1)$ 인 전이율행렬을 풀어야 한다. 따라서 알고리즘 III의 계산량이 가장 많음을 알 수 있다.

이상의 실험 결과를 분석하여 CQN-BF의 특성들을 확인하여 보기로 한다. CQN-BF의 산출률은 <그림 6>에서 보는 바와 같이 총 고객액의 수 N에 대한 함수가 된다. 즉, $N=0$ 에서부터 $N=N^*$ 까지는 산출률이 증가하게 되고 $N=N^*$ 에서부터 $N=C$ 까지는 감소하는 형태를 가진다.

이것은 N이 작을 때에는 서버들이 대부분의 시간동안 결핍되어 있고, N이 클 때에는 서버들이 대부분의 시간동안 봉쇄되어 있기 때문이다(Frein *et al.*, 1996). 폐쇄형 봉쇄 대기행렬망에서 봉쇄 구조가 BBS (Blocking-Before-Service)인 경우에는 대칭적으로 $N^* = C/2$ 가 되고, BAS인 경우에는 정확히 대칭적이지는 않지만 유사한 형태(pseudo-symmetry)를 보이는 것으로 알려져 있다(Frein *et al.*, 1996; Ross, 1983). 이러한 성질들은 불완전한 서버가 랜덤서비스시간을 갖는 모형에서도, <그림 6>에서 보는 바와 같이 성립함을 확인할 수 있다.

<그림 7>은 고장률의 증가에 따른 산출률의 변화 추이를 보이는 것으로 고장률의 증가에 따라 산출률은 급격하게 감소

하는 형태를 보이고 있다. 따라서 생산시스템의 성능을 높이기 위해서는 각 서버의 신뢰도, 즉 MTBF(mean time between failure)를 최대화하는 것이 중요하다는 것을 알 수 있다.

<그림 8>은 시스템 중 가장 낮은 서비스율을 갖는 서버에 따른 산출률의 변화 추이를 나타내는 것으로, 시스템 중 가장 낮은 서비스율을 갖는 서버의 서비스율에 따라 전체 시스템의 산출률에 대한 상한이 결정된다는 것을 볼 수 있다. 따라서 전체 시스템의 산출률을 높이기 위해서는 가장 낮은 서비스율을 갖는 서버의 서비스율을 높여서 시스템의 각 서버들 간의 서비스율이 균형되도록 하는 것이 필요하다.

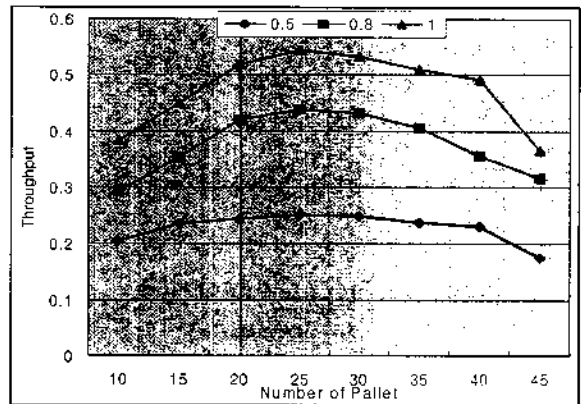


그림 8. 최소 서비스율에 따른 산출률의 상한.

6. 결론

생산시스템의 성능평가를 위한 도구로서 CQN-BF 모형의 중요성이 증가되고 있다. 그러나 분석상의 난점들로 인하여 많은 연구가 이루어지지 못한 실정이다. 더욱이 많은 가정 및 제약들을 두어서 실제 생산시스템과는 차이가 있었다. 이에 본 연구에서는 신뢰할 수 없는 서버가 랜덤서비스시간을 갖는 CQN-BF의 성능평가를 위하여 세 가지의 서로 다른 방법으로 구성되는 근사 알고리즘들을 제시하였다. 각 알고리즘들은 분해법과 반복적 알고리즘에 기초하였으며, 분해된 하부시스템에 대한 분석방법에 따라 알고리즘 I, II, III로 나뉘어 졌다.

제시된 알고리즘에 대한 적정성 및 정확도 검증을 위해 많은 수치실험이 수행되었으며, 시뮬레이션 결과와 비교되었다. 알고리즘 II는 다소 정확도가 떨어졌으나, 알고리즘 I과 III의 정확도는 비교적 우수하다고 판단되었다. 본 연구에서는 CQN-BF의 성능평가와 더불어 실험 결과를 분석·검토하여 CQN-BF에 대한 특성들을 도출함으로써 CQN-BF의 행태에 대한 이해와 통찰력을 제시하였다. 그런 의미에서 본 연구는 CQN-BF의 성능척도들(총 팔렛의 수, 버퍼의 저장능력, 각 서

버의 신뢰성 및 서비스율 등)의 최적화 및 작업통제를 위한 설계문제를 처리하기 위한 기반을 제공하였으며, 보다 더 현실적인 CQN-BF의 근사해법 개발을 위한 단서를 제공하였다고 생각되어진다.

끝으로 본 연구에서는 CQN-BF에서 각 서버들의 서비스시간이 지수분포를 따른다고 가정하였다. 앞으로 각 서버들의 서비스시간이 일반 분포를 따르며, 고장 및 수리 또한 좀더 일반적인 경우로 확장되어져야 할 것 같다. 이는 추후 연구할 수 있는 과제로 남겨두기로 한다.

참고문헌

Altioik, T. (1985), On the phase-type approximations of general distributions, *IIE Transactions*, 17(2), 110-116.

Bouhchouch, A., Frein, Y. and Dallery, Y., Performance evaluation of closed tandem queueing networks with finite buffers, *Technical Report*.

Choong, Y. F. and Gershwin, S. B. (1987), A decomposition method for the approximate evaluation of capacitated transfer lines with unreliable machines and random processing times, *IIE Transactions*, 19(2), 150-159.

Dallery, Y., David, R. and Xie, X. L. (1988), An efficient algorithm for analysis of transfer lines with unreliable machines and finite buffers, *IIE Transactions*, 20, 280-283.

Dallery, Y. and Frein, Y. (1989), A decomposition method for approximate analysis of closed queueing networks with blocking, in *Queueing Networks with Blocking*, Perros, H. G. and Altioik, T.(eds), North-Holland, Amsterdam, 193-215.

Dallery, Y., David, R. and Xie, X. L. (1989), Approximate analysis of transfer lines with unreliable machines and finite buffers, *IEEE Transactions on Automatic Control*, 34(9), 943-953.

Dallery, Y. and Frein, Y. (1993), On decomposition methods for tandem queueing networks with blocking, *Operations Research*, 41(2), 386-399.

Dallery, Y. and Gershwin, S. B. (1992), Manufacturing flow line systems: A review of models and analytical results, *Queueing Systems Theory and Applications*, 12, 3-94.

Frein, Y., Cornnault, C. and Dallery, Y. (1996), Modeling and analysis of closed-loop production lines with unreliable machines and finite buffers, *IIE Transactions*, 28, 545-554.

Gershwin, S. B. and Berman, O. (1981), Analysis of transfer lines consisting of two unreliable machines with random processing times and finite storage buffers, *AIIE Transactions*, 13(1), 2-11.

Gershwin, S. B. (1987), An efficient decomposition method for the approximate evaluation of tandem queues with finite storage space and blocking, *Operations Research*, 35(2), 291-305.

Gershwin, S. B. (1989), An efficient decomposition algorithm for unreliable tandem queueing systems with finite buffers, in *Queueing Networks with Blocking*, Perros, H. G. and Altioik, T. (eds), North-Holland, Amsterdam, 127-146.

Kim, H. G. and Lie, C. H. (1996), Throughput analysis for cyclic queueing networks with production blocking, *대한산업공학회지*, 22(1), 3-15.

Lee, H. S. and Pollock, S. M. (1990), Approximate analysis of open acyclic exponential queueing networks with blocking, *Operations Research*, 38, 1123-1134.

Marie, R. (1980), Calculating equilibrium probabilities for $\lambda(n)/Ck/1/N$ queues, *Proceedings of Performance 80 International Symposium on Computer Performance Modeling*, May, 28-30.

Paik, C. H. (1997), Throughput upper and lower bounds for assembly/disassembly queueing networks with blocking, *대한산업공학회지*, 23(4), 769-778.

Ross, S. M. (1983), *Stochastic Processes*, John Wiley & Sons, New York.

Sauer, C. H. and Reddy, A. R. (1975), *Approximate Analysis of Central Server Modes*, IBM T. J. Watson Research Center, Yorktown Heights, New York.

Stoyan, D. (1983), *Comparison Methods for Queues and Other Stochastic Modes*, John Wiley & Sons.

Zhuang, L., Buzacott, J. A. and Liu, X. G. (1994), Approximation mean value performance analysis of cyclic queueing networks with production blocking, *Queueing Systems*, 16, 139-165.