

주문 - 조립시스템의 설계 및 성능평가*

박찬우** · 이효성***

Design and Performance Evaluation of an Assemble-To-Order System*

Chan-Woo Park** · Hyo-Seong Lee***

■ Abstract ■

We study a multi-component production/inventory system in which individual components are made to meet various demand types. We assume that the demands arrive according to a Poisson process, but there is a fixed probability that a demand requests a particular kit of different components. Each component is produced by a flow line with several stations in which the processing times of each station follow a two-stage Coxian distribution. The production of each component is operated by an independent base-stock policy with blocking. We assume that the time needed to assemble final products follows a general distribution and the capacity of an assembling facility is sufficiently large. The objective of this study is to obtain key performance measures such as the distribution of the number of each orders for each final product and the mean time of fulfilling a customer order. The basic principle of the proposed approximation method is to decompose the original system into a set of subsystems, each subsystem being associated with a flow line. Each subsystem is analyzed in isolation using a Marie's method. An iterative procedure is then used to determine the unknown parameters of each subsystem. Numerical results show that the accuracy of the approximation method is acceptable.

Keyword : Assemble-To-Order System, Multi-Stage Production System, Queueing Networks, Decomposition, Marie's Method, Matrix-Geometric Method, Performance Evaluation.

논문접수일 : 2001년 12월 20일 논문게재확정일 : 2002년 9월 6일

* 이 논문은 2000년도 학술진흥재단의 연구비에 의하여 연구되었음(2000-041-E00124).

** (주)디지털 드림 연구원

*** 경희대학교 기계·산업시스템공학부 교수

1. 서 론

기업이 사용하는 기본적인 제조전략에는 make-to-stock 전략과 make-to-order 전략이 있다. Make-to-stock 전략은 품목을 재고로 보유함으로써 고객의 수요에 즉각적으로 대응할 수 있다는 장점은 있으나, 재고유지 비용이 과다하게 발생하고 적은 수의 일부 표준화된 제품에 국한되기 때문에 고객의 다양한 수요를 충족시키기 어렵다는 단점이 지적된다. 반면에 make-to-order 전략은 고객의 사양에 맞추어 제품을 생산하므로 높은 수준의 고객화를 제공할 수 있다는 장점은 있으나, 제품인도기간이 길어져 고객의 요구를 신속하게 충족시켜줄 수 없다는 단점이 지적된다. 주문-조립(assemble-to-order) 전략은 make-to-stock 전략과 make-to-order 전략의 단점을 보완하고 높은 수준의 고객화와 고객요구에의 신속한 대응이라는 두 가지 목표를 동시에 충족시켜 줄 수 있는 제조전략으로서 최근 많은 관심을 끌고 있다. 주문-조립 시스템은 표준 부품과 반제품만을 재고로 보유하고 있다가 고객의 주문이 도착하면, 관련된 부품과 반제품을 조립하여 고객의 요구를 충족시켜 준다. 소수의 부품과 주요 반제품으로 다양한 제품을 생산하므로 높은 수준의 고객화가 달성 가능하고, 간단한 조립에 의해 제품이 제조되므로 상품인도기간을 단축시킬 수 있어 고객에의 신속한 대응이 가능하다. 따라서 주문-조립 시스템은 높은 수준의 고객화와 고객 요구에의 신속한 대응이라는 두 가지 중요한 경쟁요소에 우위를 점할 수 있게 해주는 매력적인 제조시스템으로 각광을 받고 있다[17, 18].

주문-조립 시스템에서 부품 및 반제품은 외부 공급자로부터 공급받을 수도 있고 자체 생산할 수도 있으나, 어느 경우든 원자재로부터 최종 제품에 이르기까지 순차적으로 가치를 부여하는 과정이라는 점에서 일종의 공급사슬(supply chain)로 파악될 수 있다[7, 12]. 이 경우, 수요의 크기와 종류, 리드타임의 불확실성 및 기계고장 등 공급사슬 안에 존

재하는 불확실성에 대처하기 위하여 가치가 낮은 단계에 있는 부품 및 반제품을 재고로 유지한다. 또한 재고비용이 높고, 다양성 및 유연성 측면에서 높은 가치를 갖는 완제품의 경우는 재고를 두지 않고 고객의 주문에 따라 생산을 하게된다. 주문-조립 시스템에서 관리자는 최종제품에 대한 납품충족률(order fulfillment rate)을 일정수준 이상으로 유지하면서 부품 및 반제품의 재고 유지비용을 줄이기 위해 부품의 재고정책을 적절히 사용하여야 한다.

다양한 제품의 수요에 대한 재고정책 연구는 Hausman[9], Mamer and Smith[13], Smith, Chambers and Shlifer[16]의 모형을 필두로 시작되었으며, 본격적인 주문-조립 시스템에 대한 연구는 1990년 이후 Cheung와 Hausman[5], Song[17, 18], Glassman과 Wang[8] 등의 학자들에 의해 수행되었다. 많은 연구들에서 재고정책으로 기준재고정책(base stock policy)을 사용되었고, 1 단계(single echelon model) 주문-조립 시스템의 모형을 가정하였다. Song[17]은 완제품에 대한 수요는 복합 포아송과정(compound Poisson Process)을 따르고, 주문인도시간은 일정(constant replenishment lead-time)하다는 가정 하에서 1 단계 주문-조립 시스템의 최종 제품에 대한 납품충족률을 정확히 계산하는 방법을 개발하였다. 또한 Glasserman과 Wang[8]은 완제품에 대한 수요가 포아송 과정을 따르고 생산 시간은 지수분포를 따르는 공급사슬 모형에 대해서 재고수준과 최종제품 인도시간 사이에 선형관계가 있음을 보였다. 최근 연구에서 Song[18]은 확률적 인도기간을 가진 주문-조립 시스템에서 최종제품의 기간내 납품충족률을 행렬기하법(matrix geometric method)을 이용하여 계산하는 방법을 제안하였다.

이상에서 살펴본 바와 같이 주문-조립 시스템에 대한 분석적 연구가 최근 활발히 수행되고 있다. 그러나 이들 기존의 연구는 생산/조달 방식을 지나치게 단순화하여 실제 상황을 정확하게 반영하지 못하는 것이 단점으로 지적된다. 그러므로 실

제상황이 보다 잘 고려된 주문-조립 시스템에 대한 연구가 요구된다. 본 연구에서는 기존연구의 확장으로서 부품들이 여러 단계를 거쳐 생산되고, 각 단계에서 부품의 가공시간은 Coxian-2 분포를 따르는 주문-조립시스템을 분석하고자 한다.

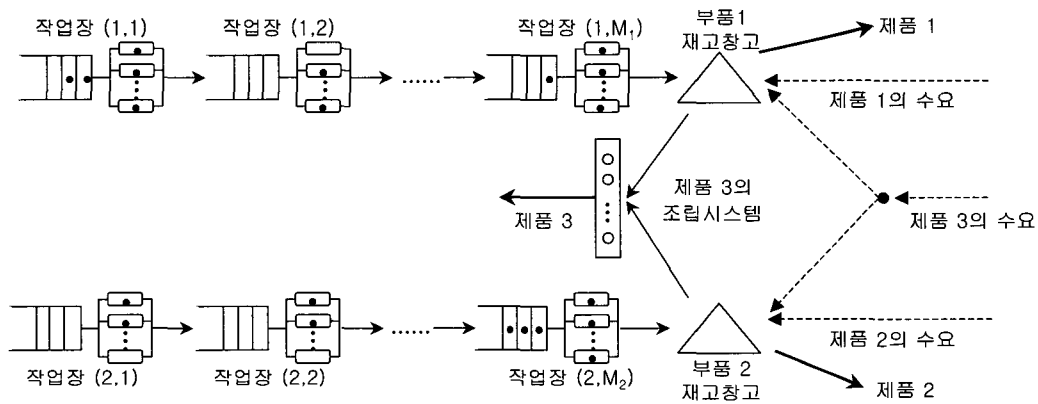
제안된 연구 모형에서 부품들이 여러 단계를 거쳐 생산되므로, 생산시스템 내에는 적지않은 재공품이 존재할 수 있으며, 적절한 생산제어 정책을 사용하여야만 필요한 재고를 유지하면서 재공품의 수를 일정 수준 이하로 줄일 수 있다. 이러한 목적에 부합되는 다단계 생산시스템에서의 제어방식으로 견인 생산제어 방식(pull-type control mechanism)을 들 수 있다[10, 11]. 견인 생산제어 방식은 밀어내기 제어방식(push-type control mechanism)에 비해 재공품 재고가 감소되고 사이클 타임의 변동이 작아지는 등 여러 장점을 지니고 있는 것으로 보고되고 있다[10]. 이에 본 연구에서는 견인 생산제어 방식의 일환으로써 유한한 크기의 버퍼(buffer)를 가지는 기본-재고정책(base-stock policy)을 부품 생산의 제어방식으로 사용하고자 한다[18]. 유한한 크기의 버퍼를 가지는 기본-재고정책은 생산시스템 내의 재공품의 수를 일정 수준 이하로 유지하면서, 목표 재고수준을 별도로 설정할 수 있다는 특성이 있어, 본 연구모형의 부품 생산 제어방식으로 적합하다고 판단된다. 이와 같이 제안된 시스템에 대하여 본 연구에서는 최종제품에 대한 제

품별 납품충족률, 수요가 충족될 때까지의 평균 시간, 각 부품의 품질확률 및 평균 재고 수준 등 시스템의 중요한 성능평가척도를 신속하게 계산하여 주는 근사적 기법을 개발하고자 한다.

2. 연구내용, 범위 및 방법

2.1 연구내용 및 범위

본 연구에서 분석하고자 하는 주문-조립 시스템은 두 개의 부품생산 시스템으로 구성되어 있으며, 각 부품생산 시스템에서 생산된 부품(component)을 이용하여 세 종류의 제품(product)을 고객에게 제공하고 있다([그림 1] 참고). 제품 1과 제품 2는 각각 부품 1과 부품 2에 해당되며, 제품 3은 조립시스템에서 부품 1과 부품 2를 조립하여 생산된다. 조립시스템의 생산시간은 평균 \bar{A} 의 일반분포를 따르고, 조립시스템에서 용량은 충분하여 부품 1과 부품 2가 모두 준비되는 즉시 조립을 시작할 수 있다고 가정한다. 제품의 수요는 도착률 λ 의 포아송(Poisson) 과정에 따라 도착하고, 도착한 수요가 제품 c 을 요구할 확률은 q^c 이며, 각각의 수요의 도착과정은 다른 수요의 도착과정과 독립이라 가정한다($c = 1, 2, 3$). 각 제품의 수요가 도착했을 때 요구되는 부품의 재고가 모두 존재하면, 제품 3의 조립은 시작되고, 제품 1과 제품



[그림 1] 두 개의 부품생산시스템으로 구성된 주문-조립 시스템

2의 수요는 즉시 만족된다. 그러나 요구되는 부품의 재고가 하나라도 존재하지 않는다면, 각 제품의 수요는 생산에 의해 부품이 가용해질 때까지 미충족수요(backorder)로 남아 기다려야 한다. 이때 부품을 기다리는 또 다른 미충족수요가 존재할 수 있으며, 고객은 지나치게 긴 시간을 지체할 수 없으므로, 부품 1(2)의 미충족수요가 B_1 (B_2)을 초과하면, 제품 1(2)의 수요는 즉시 유실(lost sales) 되는 것으로 가정한다. 만약 제품 3의 수요가 도착했을 때 부품 1과 부품 2의 미충족수요가 하나라도 B_1 과 B_2 을 초과하면, 이 수요는 유실된다고 가정한다.

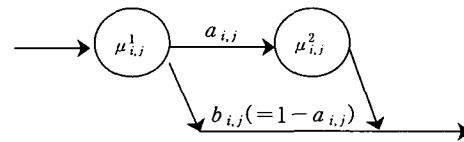
부품 i 의 생산공정은 흐름식 생산라인 시스템(flow line system)으로 목표재고량(target inventory level) S_i 을 가지는 기본재고정책에 의해 통제된다($i = 1, 2$). 작업장 ($i, 1$)에서 최초의 가공을 받은 부품은 후속 가공을 거친 후 작업장 (i, M_i)에서 최후 가공을 받게되며, 작업장 (i, M_i)에서 가공을 마친 부품은 출하되어 부품 i 재고창고에 저장된다. 본 연구에서는 부품 i 을 생산하는 M_i 개의 작업장을 모두 합하여 부품 i 생산시스템이라 정의하고, 두 개의 부품 생산시스템과 양쪽 재고창고를 모두 합하여 주문시스템이라 정의한다. 미충족수요를 포함하여 부품 i 가 수요에 의해 소모되면, 부품 i 의 생산지시가 즉시 작업장 ($i, 1$)에 도착하게 되고, 원자재의 가공이 작업장 ($i, 1$)에서부터 시작되어 밀어내기 방식에 따라 후속 작업장으로 이동한다. 부품 i 의 생산지시를 부품 i 주문(order)으로 정의하고, 부품 i 재고창고의 초기 재고수준을 S_i 로 설정하자. 그러면 부품 i 의 최대 미충족수요의 수가 B_i 이므로, 부품 i 생산시스템에서 존재할 수 있는 최대 주문의 수는 $S_i + B_i$ 이며, 부품 i 재고창고에서 최대 재고수준은 S_i 가 된다. 따라서 유한 크기의 버퍼를 가지는 기본-재고정책은 생산시스템 내의 재공품 수를 일정 수준 이하로 유지하면서 별도의 목표 재고량을 설정할

수 있어, 본 연구모형에 적합한 생산제어 방식으로 판단된다.

본 연구의 모형 수립에서 사용된 이외의 가정과 기호는 다음과 같이 정리되며, 구하고자 하는 중요 성능평가척도는 <표 1>과 같다.

[가 정]

- 1) 작업장 ($i, 1$)에는 충분한 수의 원자재가 항상 존재한다.
- 2) 작업장 (i, j)에는 동일한 성능의 $SE_{i,j}$ 대의 기계가 존재하며, 부품의 가공시간은 [그림 2]와 같은 Coxian-2 분포를 따른다고 가정한다 ($i = 1, 2, j = 1, 2, \dots, M_i$).



[그림 2] 가공시간의 Coxian-2 분포

- 3) 각 작업장에서의 기계고장은 발생하지 않는다.
- 4) 각 작업장 및 재고창고의 서비스 규칙은 선입 선출방식(FIFO)을 따른다.
- 5) 각 작업장의 크기는 충분히 커 봉쇄가 발생하지 않는다.
- 6) 부품의 작업장 간 이동시간은 무시할 수 있을 정도로 짧다.

[기 호]

본 연구에서 사용되는 기호에서는 위 첨자는 제품을, 아래 첨자는 부품을 나타낸다.

- λ^c = 제품 c 의 평균 수요율 = $q^c \lambda$.
- λ_i = 부품 i 의 평균 수요율 = $(q^i + q^3) \lambda$.
- N_i = 부품 i 생산시스템에서 존재할 수 있는 최대 주문 수 = $S_i + B_i$.
- n_i = 부품 i 생산시스템에 존재하는 주문 수,
 $0 \leq n_i \leq N_i$.
- $n_{i,j}$ = 작업장 (i, j)에 존재하는 주문 수,
 $0 \leq n_{i,j} \leq N_i$.

〈표 1〉 시스템의 성능척도

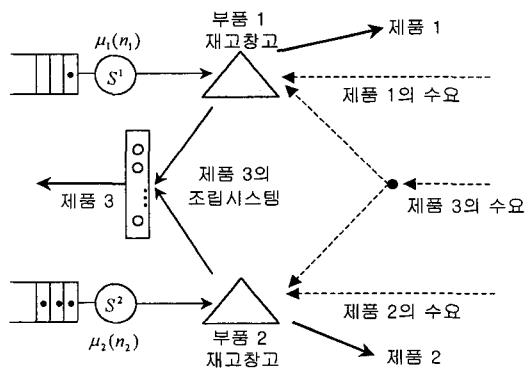
성능척도	평가 수준(level)	제품 c 의 수요 ($c = 1, 2, 3$)	부품 i 의 수요 ($i = 1, 2$)	주문 - 조립시스템
수요가 즉시 만족될 확률(Fill Rate)		F^c	F_i	F
수요가 수용될 확률(Service Level)		SL^c	SL_i	SL
주문시스템에서 수요의 평균 대기시간		\bar{W}^c	\bar{W}_i	\bar{W}
주문 - 조립시스템 내의 평균 대기 고객 수		\bar{BS}^c	\bar{BS}_i	\bar{BS}

본 연구에서 분석하고자 하는 주문 - 조립 시스템은 [그림 1]과 같이 대기 네트워크로 표현될 수 있다. 그러나 각 작업장에서의 부품의 가공시간이 지수분포를 따르지 않고, 하나의 수요가 동시에 여러 부품들을 요구할 수 있으므로 재고창고에 존재하는 각 부품의 수가 상호 종속적인 관계를 갖게 되어 본 연구의 모형은 승법형 네트워크(product-form network)의 요건을 만족하지 못한다. 따라서 대기 네트워크의 분석기법을 이용할 경우 정확한 해(exact solution)를 구하기는 불가능하며, 시스템의 분석을 위해서는 근사적 기법에 의존할 수밖에 없다.

2.2 연구 방법

앞 절에서 설명한 바와 같이 본 연구모형에서 각각의 수요는 Poisson과정을 따르지만, 각 부품의 상태는 상호 종속적인 관계를 갖게 된다. 따라서 시스템을 분석하기 위해서는 부품의 상태를 독립적으로 분석하면 안되고, 각 부품의 상태, 그리고 각 부품의 상태가 주어졌을 경우의 각 부품의 생산율을 모두 고려하여 분석하여야만 한다. 각 부품의 생산은 재고창고의 관점에서는 재고보충에 해당되므로 각 부품의 생산율을 본 연구에서는 재고보충율이라 표현하도록 한다. 만일 각 부품 생산 시스템의 상태가 주어졌을 경우에 재고보충률을 정확히 추정할 수 있다면, 각각의 부품생산시스템을 서비스 시간이 지수분포를 따르고 서비스율이 상태 의존적인 단일 서버로 치환함으로써 본 연구모형을 단순화할 수 있다. 이와 같은 방식으로 본 연

구모형을 단순화하면, 시스템의 효율적 분석이 가능해져 3.1절에서 언급될 방법에 의해 재고창고에서 부품상태의 결합확률분포를 구할 수 있으며, 이로부터 수요가 충족될 확률 등 여러 가지 중요한 성능척도를 구할 수 있다. 따라서 본 연구에서는 부품 i 생산시스템을 서비스 시간이 지수분포를 따르고, n_i 개의 주문이 존재할 때의 서비스율이 $\mu_i(n_i)$ 인 서버 S^i 로 치환하여 표시한다($i = 1, 2$). 그러면 원 네트워크는 [그림 3]과 같은 네트워크로 근사화될 수 있으며, 앞으로 이와 같은 네트워크를 네트워크 - AO(Assemble to Order)로 표기하기로 한다.



〔그림 3〕 네트워크 - AO

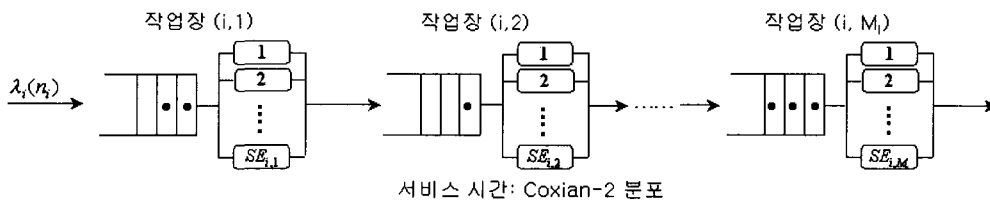
그러나 네트워크 - AO를 분석하기 위해서는 상태종속 서비스율 $\mu_i(n_i)$ 을 추정하는 효과적인 방법을 필요로 하며, 본 연구에서는 $\mu_i(n_i)$ 을 구하기 위하여 다음과 같은 방법을 사용하고자 한다. $\lambda_i(n_i)$ 을 부품 i 생산시스템에 n_i 개의 주문이

있을 때, 부품 i 주문의 상태종속 도착율이라고 정의하자. $\lambda_i(n_i)$ 는 상태종속 서비스율 $\mu_i(n_i)$ 을 알고 있다는 가정 하에 네트워크-AO의 부품상태의 결합확률분포가 구해지면, 3.2절의 식을 이용하여 구할 수 있다. 만약 $\lambda_i(n_i)$ 가 구해지면, 부품 i 생산시스템은 전체 시스템에서 분리되어, [그림 4]와 같은 상태종속 포아송(Poisson) 도착율 $\lambda_i(n_i)$ 을 가지는 개방형 대기네트워크(open queueing network)로 분석될 수 있다. [그림 4]의 개방형 대기네트워크를 분석하여 부품 i 주문 수의 안정상태확률이 구해지면, 부품 i 생산시스템의 상태종속 산출율(state dependent throughput)을 계산할 수 있다. 상태종속 산출율이 구해지면, 이 값은 곧 네트워크-AO의 상태종속 서비스율 $\mu_i(n_i)$ 로 사용된다. 네트워크-AO의 상태종속 서비스율 $\mu_i(n_i)$ 이 구해지면, 네트워크-AO를 다시 분석하여 네트워크-AO의 부품상태의 결합확률분포를 구할 수 있고, 이로부터 상태종속 도착율 $\lambda_i(n_i)$ 을 계산할 수 있다. 요약하면, $\mu_i(n_i)$ 을 알면, $\lambda_i(n_i)$ 을 계산할 수 있고, 역으로 $\lambda_i(n_i)$ 을 알면,

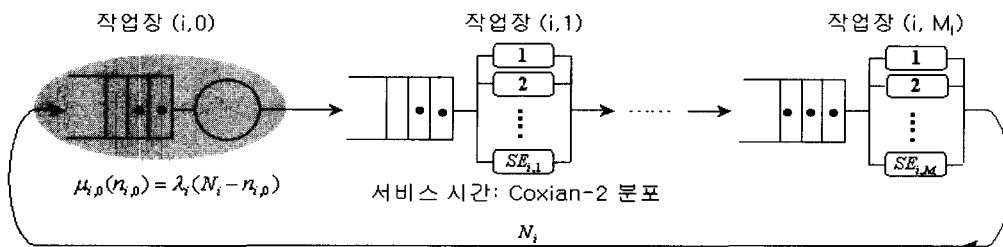
$\mu_i(n_i)$ 을 계산할 수 있다. 그러나 $\mu_i(n_i)$ 와 $\lambda_i(n_i)$ 가 모두 미지수이므로 이를 구하는 문제는 고정점 문제(fixed point problem)가 되며, 이를 구하기 위해서는 반복적 기법을 사용하여야 한다.

각 부품 생산시스템에서 시스템에 존재할 수 있는 최대 주문수가 제한되어 있으므로, [그림 4]에서 표현된 부품 i 생산시스템은 유한한 용량을 가지는 개방형 대기네트워크(OQN-RC, Open Queueing Network with Restricted Capacity)로 모델링 될 수 있으며, 이는 주문의 도착을 나타내는 별도의 스테이션을 추가함으로써 폐쇄형 대기네트워크(closed queueing network)로 변환 가능하다 [2, 6]. 폐쇄형 대기네트워크로 표현된 부품 i 생산시스템을 하부네트워크 R^i 라하고, 추가된 별도의 스테이션을 작업장 $(i, 0)$ 라 정의하자($i = 1, 2$). 그러면 부품 i 주문의 포아송 도착과정을 나타내는 작업장 $(i, 0)$ 의 상태종속 서비스율은 식 (1)과 같으며, 하부네트워크 R^i 는 [그림 5]와 같다.

$$\begin{aligned} \mu_{i,0}(n_{i,0}) &= \lambda_i(N_i - n_{i,0}), \\ n_{i,0} &= 0, 1, \dots, N_i. \end{aligned} \tag{1}$$

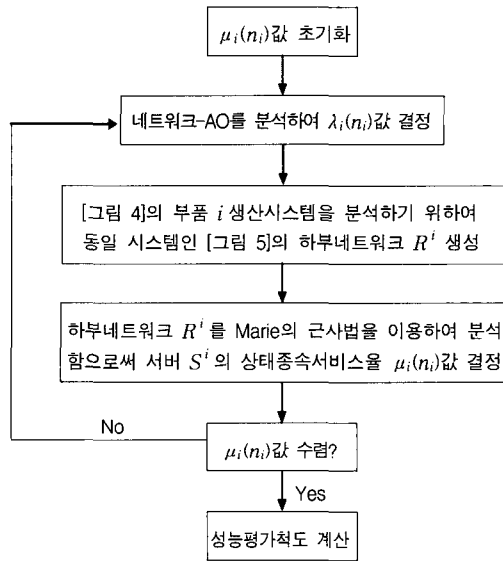


[그림 4] 개방형 대기 네트워크로 표현된 부품 i 생산시스템



[그림 5] 하부네트워크 R^i

여기서 유의할 점은 [그림 5]에서 표현된 하부 네트워크 R^i 은 [그림 4]에서 표현된 부품 i 생산 시스템의 또 다른 표현이며, 같은 시스템을 나타낸다. 따라서 하부네트워크 R^i 를 분석하는 것은 [그림 4]에서 표현된 부품 i 생산시스템을 분석하는 것과 같으며, 본 연구에서는 하부네트워크 R^i 를 분석함으로써 서버 S^i 의 상태종속 서비스를 $\mu_i(n_i)$ 을 구하고자 한다. 그러나 기계의 가공시간이 Coxian-2 분포를 따르므로 하부네트워크 R^i 는 비 승법형 대기네트워크이다. 따라서 하부네트워크 R^i 를 분석하기 위해서는 별도의 근사법을 필요로 하며, 본 연구에서는 Marie[2, 14]의 근사법을 사용하고자 한다. 원 문제를 분석하기 위한 이상의 과정을 요약하여 도식화하면 [그림 6]과 같다.



[그림 6] 원 문제를 해결하기 위한 분석과정 요약

3. 네트워크-AO의 분석

3.1 부품상태의 결합확률분포 구하기

본 절에서는 네트워크-AO에서 서버 S^i 의 상태종속 서비스를 $\mu_i(n_i)$ 을 알고 있다고 가정하

고, 재고창고에서 각 부품상태의 결합확률분포를 구하고자 한다. 만약 서버 S^i 의 상태종속 서비스를 $\mu_i(n_i)$ 을 알고 있다면, 수요과정이 포아송과정에 따라 도착하므로, 주문시스템은 연속시간 마코브체인(continuous time Markov chain)으로 모델링 가능하며, 마코프체인의 분석을 통해 각 부품 상태에 대한 결합확률 분포를 구할 수 있다. 시스템의 상태를 정의하기 위해서, n_i 가 서버 S^i 에 존재하는 주문 수를 나타낸다고 하자 ($i = 1, 2$). 만약 서버 S^i 에 n_i 개의 주문이 있다면, 부품 i 재고창고에는 $(S_i - n_i)$ 개의 부품(음수일 경우, 미충족수요의 수)이 존재하게 된다. 따라서 시스템의 상태는 2차원 상태인 (n_1, n_2) 로 정의될 수 있으며, 상태 (n_1, n_2) 의 상태공간 ST 는 다음과 같이 정의된다.

$$ST = \{(n_1, n_2) \mid 0 \leq n_1 \leq N_1, 0 \leq n_2 \leq N_2\}$$

만약 상태공간 ST 의 모든 상태를 n_1 을 기준으로 사전식 배치(lexicographic order)를 하면, 상태전이율 행렬(transition rate matrix)은 식 (2)와 같이 3종류의 소행렬로 블록(block)화 될수 있으며, $N_2 = 3$ 일 경우에 식 (2)의 소행렬의 예는 [그림 7]과 같다.

$$Q = \begin{pmatrix} A(0) & A^0 & 0 & 0 & \dots & 0 & 0 \\ \mu_1(1)I & A(1) & A^0 & 0 & \dots & 0 & 0 \\ 0 & \mu_1(2)I & A(2) & A^0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & \mu_1(N-1)I & A(N-1) & A^0 \\ 0 & 0 & \dots & \dots & 0 & \mu_1(N)I & A(N) \end{pmatrix} \quad (2)$$

위의 예로부터 식 (2)의 소행렬들은 다음의 식으로 정리된다.

$$A(n_1) = \{a_{i,j}(n_1)\}, n_1 = 0, \dots, N_1, \quad (3)$$

$$i = 1, \dots, N_2 + 1, j = 1, \dots, N_2 + 1$$

$$\begin{array}{c}
\mathbf{A}(n_1) \\
0 \leq n_1 \leq N_1 - 1
\end{array}
= \begin{array}{c}
\begin{array}{cccc}
(n_1,0) & (n_1,1) & (n_1,2) & (n_1,3) \\
\begin{pmatrix}
-\lambda - \mu_1(n_1) & \lambda^2 & & \\
\mu_2(1) & -\lambda - \mu_1(n_1) - \mu_2(1) & \lambda^2 & \\
& \mu_2(2) & -\lambda - \mu_1(n_1) - \mu_2(2) & \lambda^2 \\
& & \mu_2(3) & -\lambda - \mu_1(n_1) - \mu_2(3)
\end{pmatrix}
\end{array} \\
\begin{array}{cccc}
(N_1,0) & (N_1,1) & (N_1,2) & (N_1,3) \\
\begin{pmatrix}
-\lambda^2 - \mu_1(N_1) & \lambda^2 & & \\
\mu_2(1) & -\lambda^2 - \mu_1(N_1) - \mu_2(1) & \lambda^2 & \\
& \mu_2(2) & -\lambda^2 - \mu_1(N_1) - \mu_2(2) & \lambda^2 \\
& & \mu_2(3) & -\lambda^2 - \mu_1(N_1) - \mu_2(3)
\end{pmatrix}
\end{array}
\end{array}
\end{array}
\begin{array}{c}
\mathbf{A}^0 = \begin{array}{cccc}
(n_1+1,0) & (n_1+1,1) & (n_1+1,2) & (n_1+1,3) \\
\begin{pmatrix}
\lambda^1 & \lambda^3 & & \\
& \lambda^1 & \lambda^1 & \\
& & \lambda^1 & \lambda^1 \\
& & & \lambda^1
\end{pmatrix} \\
\mu_1(n_1)\mathbf{I} = \begin{array}{cccc}
(n_1,0) & (n_1-1,0) & (n_1-1,1) & (n_1-1,2) & (n_1-1,3) \\
\begin{pmatrix}
\mu_1(n_1) & & & \\
& \mu_1(n_1) & & \\
& & \mu_1(n_1) & \\
& & & \mu_1(n_1)
\end{pmatrix}
\end{array}
\end{array}
\end{array}$$

[그림 7] $N_2 = 3$ 일 경우 식 (2)의 소행렬의 예

$$a_{i,j}(n_1) = \begin{cases} \lambda^2, & \text{if } 1 \leq i \leq N_2, j = i+1, \\ \mu_2(i-1), & \text{if } 2 \leq i \leq N_2+1, j = i-1, \\ -\lambda^T(n_1, i) - \mu_1(n_1) - \mu_2(i-1), & \text{if } i = j \\ 0, & \text{o/w.} \end{cases}$$

where, $\lambda^T(n_1, i) =$

$$\begin{cases} \lambda, & \text{if } n_1 < N_1, i \leq N_2, \\ \lambda^1, & \text{if } n_1 < N_1, i = N_2+1, \\ \lambda^2, & \text{if } n_1 = N_1, i \leq N_2, \\ 0, & \text{o/w.} \end{cases}$$

$$\mathbf{A}^0 = \{a_{i,j}^0, i = 1, \dots, N_2+1, j = 1, \dots, N_2+1,$$

$$a_{i,j}^0 = \begin{cases} \lambda^1, & \text{if } j = i, \\ \lambda^3, & \text{if } 1 \leq i \leq N_2, j = i+1, \\ 0, & \text{o/w.} \end{cases}$$

$\mathbf{I} =$ 크기 $(N_2+1) \times (N_2+1)$ 의 단위행렬. (4)

식 (2)의 전이율 행렬은 삼각대각(tridiagonal) 형태를 취하고 있으므로, 행렬기하법[4, 15]을 적용하여 분석할 수 있다. 상태 (n_1, n_2) 의 안정상태 확률을 $P_{AO}(n_1, n_2)$ 라 정의했을 때, 소행렬로 표현된 균형방정식은 다음과 같다.

$$\begin{aligned}
\mathbf{P}_{AO}(0) \cdot \mathbf{A}(0) + \mu(1)\mathbf{P}_{AO}(1) &= \mathbf{0}, \\
\mathbf{P}_{AO}(n_1-1) \cdot \mathbf{A}^0 + \mathbf{P}_{AO}(n_1) \cdot \mathbf{A}(n_1) + \mu(n_1+1) \cdot \\
\mathbf{P}_{AO}(n_1+1) &= \mathbf{0}, \quad n_1 = 1, 2, \dots, N_1-1, \\
\mathbf{P}_{AO}(N_1-1) \cdot \mathbf{A}^0 + \mathbf{P}_{AO}(N_1) \cdot \mathbf{A}(N_1) &= \mathbf{0}. \quad (5)
\end{aligned}$$

여기서, $\mathbf{P}_{AO}(n_1) = \{P_{AO}(n_1, 0), P_{AO}(n_1, 1), \dots, P_{AO}(n_1,$

$N_2)\}$

식 (5)의 시스템의 균형방정식은 반복적인 구조를 가지므로, $\mathbf{P}_{AO}(n_1)$ 는 다음과 같은 순환식으로 표현될 수 있다.

$$\mathbf{P}_{AO}(n_1) = \mathbf{P}_{AO}(n_1-1) \cdot \mathbf{R}_{AO}(n_1), \quad n_1 = 1, \dots, N_1. \quad (6)$$

여기서, $\mathbf{R}_{AO}(n_1)$ 은 크기 $(N_2+1) \times (N_2+1)$ 의 행렬로 다음과 같다.

$$\mathbf{R}_{AO}(n_1) = \begin{cases} -\mathbf{A}^0 \cdot [\mathbf{A}(n_1) + \mu(n_1+1) \cdot \mathbf{R}_{AO}(n_1+1)]^{-1}, & \text{if } n_1 = 1, 2, \dots, N_1-1, \\ -\mathbf{A}^0 \cdot \mathbf{A}(n_1)^{-1}, & \text{if } n_1 = N_1. \end{cases} \quad (7)$$

따라서 다음과 같은 연립방정식이 성립되며, 이 연립방정식을 풀면 $\mathbf{P}_{AO}(0)$ 을 구할 수 있다.

$$\begin{aligned}
\mathbf{P}_{AO}(0) \cdot [\mathbf{A}(0) + \mu(1) \cdot \mathbf{R}_{AO}(1)] &= \mathbf{0} \\
\mathbf{P}_{AO}(0) \sum_{k=0}^m \mathbf{R}_{AO}(k) \cdot \mathbf{e}_{AO} &= \mathbf{1} \quad (8)
\end{aligned}$$

여기서, \mathbf{e}_{AO} 는 모든 원소가 1인 크기 (N_2+1) 의 열벡터이며, $\mathbf{R}_{AO}(k)$ 은 크기 $(N_2+1) \times (N_2+1)$ 의 행렬로 다음과 같다.

$$\mathbf{R}_{AO}(k) = \begin{cases} \mathbf{I}, & \text{if } k = 0, \\ \prod_{i=1}^k \mathbf{R}_{AO}(i), & \text{if } k = 1, \dots, N_1. \end{cases}$$

$\mathbf{P}_{AO}(0)$ 가 구해지면, 식 (6)으로부터 다른 상

태의 안정상태확률을 순차적으로 계산할 수 있다.

3.2 부품 생산시스템의 상태종속 도착률 구하기

안정상태확률 $P_{AO}(n_1, n_2)$ 가 구해지면, 이로부터 서버 S^i 에 n_i 개의 주문이 존재할 확률 $P_i(n_i)$ 은 $P_{AO}(n_1, n_2)$ 의 주변 확률 분포이므로 $P_{AO}(n_1, n_2)$ 에서 쉽게 계산할 수 있다. $P_i(n_i)$ 이 구해지면, 서버 S^i 의 상태종속 산출율 $X_i(n_i)$ 은 다음과 같이 계산할 수 있다.

$$X_i(n_i) = \begin{cases} \mu_i(n_i+1) \frac{P_i(n_i+1)}{P_i(n_i)} & \text{for } n_i = 0, 1, \dots, N_i - 1 \\ 0 & \text{for } n_i = N_i. \end{cases} \quad (9)$$

서버 S^i 의 상태종속 산출율 $X_i(n_i)$ 이 계산되면, 부품 i 생산시스템에의 상태종속 도착률 $\lambda_i(n_i)$ 은 다음과 같다.

$$\lambda_i(n_i) = X_i(n_i), \quad n_i = 0, 1, \dots, N_i. \quad (10)$$

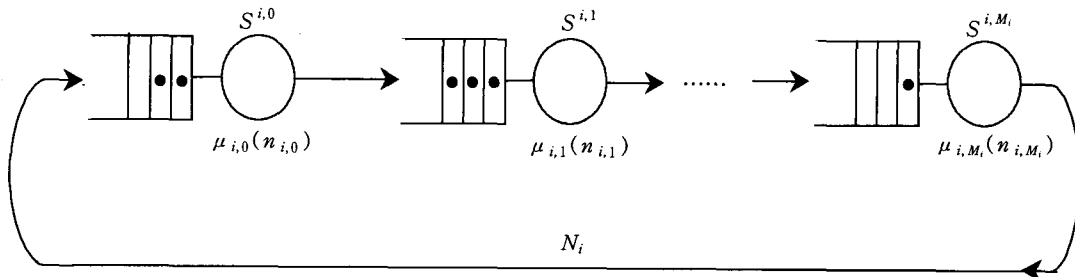
부품 i 생산 시스템에 상태종속 도착률 $\lambda_i(n_i)$ 이 구해지면, 식 (1)을 이용하여 하부네트워크 R^i 의 작업장 $(i, 0)$ 의 상태종속 서비스율을 구할 수 있다. 따라서 문제는 작업장 $(i, 0)$ 을 제외한 하부네트워크 R^i 의 나머지 작업장의 상태종속 서비스율을 구하는 것으로 귀착되며, 다음 절에서 이 문제를 상세히 다루게 된다.

4. 하부 네트워크 분석

4.1 분석 방법

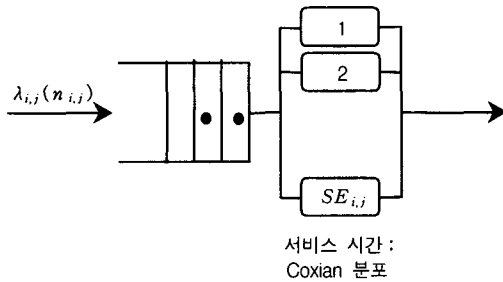
본 연구에서는 [그림 5]에 표현된 하부 네트워크 R^i 를 분석하기 위하여 Marie의 근사법[2, 14]을 사용하고자 한다. Marie의 근사법은 서비스 시간이 일반분포를 따르는 폐쇄형 대기네트워크를 분석하기 위한 근사법으로, 전체 시스템을 몇 개의 하부네트워크로 분해한 후, 각각의 하부네트워크를 서비스 시간이 지수분포를 따르고 서비스율이 상태 의존적인 단일 서버로 치환함으로써 분석하고자 하는 폐쇄형 대기네트워크를 승법형 대기네트워크로 근사화하는 기법이다. 이와 같이 근사화된 승법형 대기네트워크를 동등 승법형 대기네트워크(equivalent product form network)라 부르며, Marie의 근사법은 동등 승법형 대기네트워크를 구성하는 단일 서버의 상태종속 서비스율을 구하기 위하여 각각의 하부네트워크를 상태종속 Poisson 도착과정을 갖는 개방형 대기시스템으로 가정해 분석한다.

본 연구모형에서 하부네트워크 R^i 의 작업장 (i, j) 을 하부네트워크 $R^{i,j}$ 라 했을 때, 하부네트워크 R^i 는 총 $(M_i + 1)$ 개의 하부네트워크로 분해된다. 하부네트워크 $R^{i,j}$ 에 대응하는 동등 승법형 대기네트워크의 서버를 $S^{i,j}$ 라 정의하고, 서버 $S^{i,j}$ 에 $n_{i,j}$ 개의 주문이 있을 때의 상태종속 서비스율을 $\mu_{i,j}(n_{i,j})$ 라 했을 때, 하부네트워크 R^i 의 동등 승법형 대기네트워크는 [그림 8]과 같다.



[그림 8] 하부네트워크 R^i 의 동등 승법형 대기네트워크

서버 $S^{i,j}$ 의 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 은 하부네트워크 $R^{i,j}$ 을 분석함으로써 얻어진다. 하부네트워크 $R^{i,j}$ 는 $n_{i,j}$ 개의 주문이 있을 때 상태종속 도착율이 $\lambda_{i,j}(n_{i,j})$ 이고, 도착간격이 지수분포를 따르는 개방형 대기시스템으로 간주해 분석한다. 우선, 하부네트워크 $R^{i,j}$ 의 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 을 알고 있다고 가정하자. 그러면 하부네트워크 $R^{i,j}$ 는 [그림 9]과 같은 개방형 대기시스템으로 표현될 수 있다.



[그림 9] 하부네트워크 $R^{i,j}$

하부네트워크 $R^{i,j}$ 을 분석하여 안정상태확률이 구해지면, 하부네트워크 $R^{i,j}$ 의 상태종속 산출율을 계산할 수 있다. 상태종속 산출율이 구해지면, 이 값은 곧 동등 승법형 대기네트워크의 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 로 사용된다. 동등 승법형 대기네트워크의 상태종속 서비스율이 구해지면, 동등 승법형 네트워크를 구성하는 각 서버의 상태종속 산출율을 구할 수 있으며, 이 값은 다시 해당 하부네트워크의 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 로 사용된다. 따라서 동등 승법형 대기네트워크의 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 이 구해지면, 이로부터 하부네트워크의 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 을 구할 수 있고, 하부네트워크의 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 이 구해지면, 이로부터 동등 승법형 대기네트워크의 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 을 구할 수 있게 된다. 그러나 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 와 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 가 모두 미지수

이므로, 이를 구하는 문제는 앞에서와 같이 고정점 문제가 되며, 이를 구하기 위해서는 반복적 기법을 사용해야 한다. 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 가 주어졌을 때, 하부네트워크 $R^{i,j}$ 을 분석하는 방법과 하부네트워크 R^i 을 분석하는 과정이 4.2절, 4.3절에서 상세히 소개된다.

4.2 하부네트워크 $R^{i,j}$ 의 분석, $j = 1, \dots, M_i$.

3절에서 우리는 하부네트워크 R^i 의 작업장 $(i, 0)$ 의 상태종속 서비스율을 계산하였다. 만약 하부네트워크 R^i 의 모든 하부네트워크의 상태종속 서비스율을 알고 있다면, 4.4절에서 언급될 방법에 의해서 네트워크-AO의 서버 S^i 의 상태종속 서비스율을 계산할 수 있다. 그러므로 문제는 작업장 $(i, 0)$ 을 제외한 하부네트워크 R^i 의 나머지 하부네트워크의 상태종속 서비스율을 구하는 것으로 귀착되며, 이를 위해 본 연구에서는 작업장 $(i, 0)$ 을 제외한 각각의 하부네트워크를 상태종속 도착율이 $\lambda_{i,j}(n_{i,j})$ 이고, 도착간격이 지수분포를 따르는 개방형 시스템으로 간주해 분석한다. 우선 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 을 알고 있다고 가정하자. 그러면 가공시간이 Coxian-2 분포를 따르므로 하부네트워크 $R^{i,j}$ 는 상태의 수가 유한한 연속시간 마코프체인으로 모델링될 수 있다. 시스템의 상태를 정의하기 위하여 m_1 을 하부네트워크 $R^{i,j}$ 에 존재하는 주문 수, m_2 를 Coxian-2 분포의 1 단계에서 가공 중인 주문 수라 하면, 시스템의 상태는 (m_1, m_2) 로 정의될 수 있다. 따라서 하부네트워크 $R^{i,j}$ 의 상태공간 $\mathbf{ST}_{i,j}$ 는 다음과 같이 표시된다.

$$\mathbf{ST}_{i,j} = \{(m_1, m_2) \mid 0 \leq m_1 \leq N_i, 0 \leq m_2 \leq Mn_0(m_1)\}$$

$$\text{where } Mn_0(m_1) = \text{Min}\{SE_{i,j}, m_1\}.$$

$\mathbf{ST}_{i,j}$ 의 모든 상태들을 m_1 을 기준으로 사전식

$$\begin{aligned}
 \mathbf{W}(0) &= \begin{matrix} (0,0) \\ (0,0) \end{matrix} \left[-\lambda_{i,j}(0) \right] & \mathbf{W}(1) &= \begin{matrix} (1,0) & (1,1) \\ (1,0) & (1,1) \end{matrix} \left[\begin{matrix} -\lambda_{i,j}(1) - \mu_{i,j}^2 & \\ c_{i,j}\mu_{i,j}^1 & -\lambda_{i,j}(1) - \mu_{i,j}^1 \end{matrix} \right] \\
 \mathbf{W}(m_1) &= \begin{matrix} (m_1,0) & (m_1,1) & (m_1,2) \\ (m_1,0) & (m_1,1) & (m_1,2) \end{matrix} \left[\begin{matrix} -\lambda_{i,j}(m_1) - 2\mu_{i,j}^2 & & \\ c_{i,j}\mu_{i,j}^1 & -\lambda_{i,j}(m_1) - \mu_{i,j}^1 - \mu_{i,j}^2 & \\ & 2c_{i,j}\mu_{i,j}^1 & -\lambda_{i,j}(m_1) - 2\mu_{i,j}^1 \end{matrix} \right] & \mathbf{W}(5) &= \begin{matrix} (5,0) & (5,1) & (5,2) \\ (5,0) & (5,1) & (5,2) \\ (5,1) & (5,1) & (5,2) \\ (5,2) & & \end{matrix} \left[\begin{matrix} -2\mu_{i,j}^2 & & \\ c_{i,j}\mu_{i,j}^1 & -\mu_{i,j}^1 - \mu_{i,j}^2 & \\ & 2c_{i,j}\mu_{i,j}^1 & -2\mu_{i,j}^1 \end{matrix} \right] \\
 \mathbf{Z}(0) &= \begin{matrix} (1,0) & (1,1) \\ (0,0) & \end{matrix} \left[\begin{matrix} \lambda_{i,j}(0) \\ \lambda_{i,j}(0) \end{matrix} \right] & \mathbf{Z}(1) &= \begin{matrix} (2,0) & (2,1) & (2,2) \\ (1,0) & (2,1) & (2,2) \\ (1,1) & & \end{matrix} \left[\begin{matrix} \lambda_{i,j}(1) \\ \lambda_{i,j}(1) \\ \lambda_{i,j}(1) \end{matrix} \right] & \mathbf{Z}(m_1) &= \begin{matrix} (m_1+1,0) & (m_1+1,1) & (m_1+1,2) \\ (m_1,0) & (m_1+1,1) & (m_1+1,2) \\ (m_1,1) & & \end{matrix} \left[\begin{matrix} \lambda_{i,j}(m_1) \\ \lambda_{i,j}(m_1) \\ \lambda_{i,j}(m_1) \end{matrix} \right] \\
 \mathbf{H}(1) &= \begin{matrix} (0,0) \\ (1,0) & (1,1) \end{matrix} \left[\begin{matrix} \mu_{i,j}^2 \\ d_{i,j}\mu_{i,j}^1 \end{matrix} \right] & \mathbf{H}(2) &= \begin{matrix} (2,0) & (2,1) & (2,2) \\ (2,0) & (2,1) & (2,2) \\ (2,1) & & \end{matrix} \left[\begin{matrix} 2\mu_{i,j}^2 & & \\ d_{i,j}\mu_{i,j}^1 & \mu_{i,j}^2 & \\ & 2d_{i,j}\mu_{i,j}^1 & \end{matrix} \right] & \mathbf{H}(m_1) &= \begin{matrix} (m_1-1,0) & (m_1-1,1) & (m_1-1,2) \\ (m_1,0) & (m_1-1,1) & (m_1-1,2) \\ (m_1,1) & & \end{matrix} \left[\begin{matrix} 2\mu_{i,j}^2 & & \\ d_{i,j}\mu_{i,j}^1 & \mu_{i,j}^2 & \\ & 2d_{i,j}\mu_{i,j}^1 & \end{matrix} \right]
 \end{aligned}$$

[그림 10] $N_i = 5, SE_{i,j} = 2$ 일 경우의 하부네트워크 $R^{i,j}$ 의 소행렬의 예

배치를 하면, 전이율 행렬은 아래와 같다.

$$\mathbf{Q}_{ij} = \begin{pmatrix} \mathbf{W}(0) & \mathbf{Z}(0) & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{H}(1) & \mathbf{W}(1) & \mathbf{Z}(1) & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}(2) & \mathbf{W}(2) & \mathbf{Z}(2) & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{H}(N_i-1) & \mathbf{W}(N_i-1) & \mathbf{Z}(N_i-1) \\ \mathbf{0} & \mathbf{0} & \cdots & \cdots & \mathbf{0} & \mathbf{H}(N_i) & \mathbf{W}(N_i) \end{pmatrix} \quad (11)$$

식 (11)의 소행렬은 아래의 식으로 표현되며, 각각의 소행렬의 예는 [그림 10]와 같다.

$$\mathbf{W}(m_1) = \{w_{k,l}(m_1)\}, 0 \leq m_1 \leq N_i, \quad (12)$$

$$\begin{aligned}
 1 \leq k \leq Mn_1(m_1), 1 \leq l \leq Mn_1(m_1), \\
 \text{where } Mn_1(m_1) = \text{Min}\{SE_{i,j}, m_1\} + 1,
 \end{aligned}$$

$$w_{k,l}(m_1) = \begin{cases} (k-1)c_{i,j}\mu_{i,j}^1, & \text{if } 2 \leq k \leq Mn_1(m_1), l = k-1, \\ -\lambda_{i,j}(m_1) - (k-1)\mu_{i,j}^1, & \\ -[Mn_1(m_1) - k]\mu_{i,j}^2, & \text{if } i = k, \\ 0, & o = w. \end{cases}$$

$$\mathbf{Z}(m_1) = \{z_{k,l}(m_1)\}, 0 \leq m_1 \leq N_i - 1, \quad (13)$$

$$1 \leq k \leq Mn_1(m_1), 1 \leq l \leq Mn_1(m_1 + 1),$$

$$z_{k,l}(m_1) = \begin{cases} \lambda_{i,j}(m_1), & \text{if } l = k + \rho(m_1), \\ 0, & o/w. \end{cases}$$

$$\text{where, } \rho(m_1) = \begin{cases} 1, & \text{if } m_1 < SE_{i,j}, \\ 0, & o/w. \end{cases}$$

$$\mathbf{H}(m_1) = \{h_{k,l}(m_1)\}, 1 \leq m_1 \leq N_i, \quad (14)$$

$$1 \leq k \leq Mn_1(m_1), 1 \leq l \leq Mn_1(m_1 - 1),$$

$$h_{k,l}(m_1) = \begin{cases} (k-1)d_{i,j}\mu_{i,j}^1, & \\ \text{if } 2 \leq k \leq Mn_1(m_1), & \\ l = k - \rho(m_1 - 1), & \\ (Mn_1(m_1) - k)\mu_{i,j}^2, & \\ \text{if } 1 \leq k \leq Mn_0(m_1), & \\ l = k + [1 - \rho(m_1 - 1)], & \\ 0, & o/w. \end{cases}$$

식 (11)의 전이율 행렬은 삼각대각형태를 취하고 있으므로, 3절에서 언급한 행렬기하법을 적용할 수 있으며, 상태 (m_1, m_2) 의 안정상태확률을 $P_{i,j}(m_1, m_2)$ 라 했을 때, 소행렬로 표현된 균형 방정식은 다음과 같다.

$$\mathbf{P}_{ij}(0) \cdot \mathbf{W}(0) + \mathbf{P}_{ij}(1) \cdot \mathbf{H}(1) = \mathbf{0}, \quad (15)$$

$$\mathbf{P}_{ij}(m_1-1) \cdot \mathbf{Z}(m_1-1) + \mathbf{P}_{ij}(m_1) \cdot \mathbf{W}(m_1) + \mathbf{P}_{ij}(m_1+1) \cdot \mathbf{H}(m_1+1) = \mathbf{0}, m_1 = 1, 2, \dots, N_i - 1,$$

$$\mathbf{P}_{ij}(N_i-1) \cdot \mathbf{Z}(N_i-1) + \mathbf{P}_{ij}(N_i) \cdot \mathbf{W}(N_i) = \mathbf{0}.$$

여기서, $\mathbf{P}_{i,j}(m_1) = \{P_{i,j}(m_1, 0), P_{i,j}(m_1, 1), \dots, P_{i,j}(m_1, Mn_0(m_1))\}$.

식 (15)의 시스템 균형방정식은 반복적인 구조를 가지므로 $\mathbf{P}_{i,j}(m_1)$ 는 다음과 같은 순환식으로 표현될 수 있다.

$$\mathbf{P}_{i,j}(m_1) = \mathbf{P}_{i,j}(m_1-1) \cdot \mathbf{R}(m_1), m_1 = 1, \dots, N_i. \quad (16)$$

여기서, $\mathbf{R}(m_1)$ 은 크기 $Mn_1(m_1-1) \times Mn_1(m_1)$ 의 행렬로 다음과 같다.

$$\mathbf{R}(m_1) = \begin{cases} -\mathbf{Z}(m_1-1) \cdot [\mathbf{W}(m_1) + \mathbf{R}(m_1+1)] \cdot \mathbf{H}(m_1+1)^{-1}, & \text{if } m_1 = 1, 2, \dots, N_i-1, \\ -\mathbf{Z}(m_1-1) \cdot \mathbf{W}(m_1)^{-1}, & \text{if } m_1 = N_i. \end{cases} \quad (17)$$

식 (16)에 의해서, 모든 확률의 합을 1로 하는 정규화 방정식은 다음과 같이 표시된다.

$$\begin{aligned} \sum_{m_1=0}^{N_i} \mathbf{P}_{i,j}(m_1) \cdot \mathbf{e}(m_1) \\ = \mathbf{P}_{i,j}(0) \sum_{m_1=0}^{N_i} \mathbf{R}P(m_1) \cdot \mathbf{e}(m_1) = 1 \end{aligned} \quad (18)$$

여기서, $\mathbf{e}(m_1)$ 는 모든 원소가 1인 크기 $Mn_1(m_1)$ 의 열벡터이며, $\mathbf{R}P(m_1)$ 은 크기 $Mn_1(m_1)$ 의 행벡터로 다음과 같다.

$$\mathbf{R}P(m_1) = \begin{cases} 1, & \text{if } m_1 = 0, \\ \prod_{k=1}^{m_1} \mathbf{R}(k), & \text{if } m_1 = 1, \dots, N_i. \end{cases} \quad (19)$$

따라서 $\mathbf{P}_{i,j}(0)$ 은 식 (18)에 의해서 다음과 같이 구할 수 있다.

$$\mathbf{P}_{i,j}(0) = \mathbf{P}_{i,j}(0,0) = 1 / \sum_{m_1=0}^{N_i} \mathbf{R}P(m_1) \cdot \mathbf{e}(m_1). \quad (20)$$

$\mathbf{P}_{i,j}(0)$ 가 구해지면, 식 (16)으로부터 다른 상태의 확률을 순차적으로 구할 수 있으며, 이로부터 하부네트워크 $R^{i,j}$ 에 m_1 개의 간판이 있을 확률 $P_{i,j}(m_1)$ 는 다음과 같이 표시된다.

$$P_{i,j}(m_1) = \sum_{m_2=0}^{Mn_0(m_1)} P_{i,j}(m_1, m_2), \quad m_1 = 0, 1, \dots, N_i. \quad (21)$$

4.3 승법형 근사법을 이용한 하부네트워크 R^i 의 분석

하부네트워크 $R^{i,j}$ 의 안정상태확률 $P_{i,j}(n_{i,j})$ 이 구해지면, 하부네트워크 $R^{i,j}$ 에 존재하는 주문의 수가 $n_{i,j}$ 일 경우의 상태종속 산출율 $v_{i,j}(n_{i,j})$ 의 값은 다음과 같이 얻을 수 있다.

$$v_{i,j}(n_{i,j}) = \begin{cases} 0, & \text{if } n_{i,j} = 0, \\ \lambda_{i,j}(n_{i,j}-1) \frac{P_{i,j}(n_{i,j}-1)}{P_{i,j}(n_{i,j})}, & \text{if } n_{i,j} = 1, 2, \dots, N_i. \end{cases} \quad (22)$$

하부네트워크 $R^{i,j}$ 의 상태종속 산출율 $v_{i,j}(n_{i,j})$ 의 값이 구해지면, 동등 승법형 대기네트워크를 구성하는 서버 $S^{i,j}$ 의 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 은 다음과 같이 구해진다.

$$\begin{aligned} \mu_{i,j}(n_{i,j}) &= v_{i,j}(n_{i,j}), \\ n_{i,j} &= 0, \dots, N_i, \quad j = 1, \dots, M_i. \end{aligned} \quad (23)$$

따라서 하부네트워크 $R^{i,j}$ 의 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 을 알고 있다면, 하부네트워크 $R^{i,j}$ 의 안정상태확률을 구할 수 있고, 식 (22)과 식 (23)을 이용하여 동등 승법형 대기네트워크의 상태종속 서비스율 $\mu_{i,j}(n_{i,j})$ 을 구할 수 있다. 그러나 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 의 값은 우리가 구하여야만 하는 미지의 값이다. 상태종속 도착율 $\lambda_{i,j}(n_{i,j})$ 을 구하기 위하여, 동등 승법형 대기네트워크의 서비스율 $\mu_{i,j}(n_{i,j})$ 을 알고 있다고 가정하자. 그러면 적합한 알고리즘을 이용하여 서버 $S^{i,j}$ 의 상태종속 산출율 $X_{i,j}(n_{i,j})$ 을 구할 수 있다. 만일 중합(convolution) [알고리즘 3]을 이용한다면, 서버 $S^{i,j}$ 의 상태종속 산출율은 다음 식에 의해 구할 수 있다.

$$X_{i,j}(n_{i,j}) = \begin{cases} e_{i,j} \frac{g_{i,j/(M_i+1)}(N_i - n_{i,j} - 1)}{g_{i,j/(M_i+1)}(N_i - n_{i,j})}, & n_{i,j} = 0, 1, \dots, N_i - 1, \\ 0 & n_{i,j} = N_i. \end{cases} \quad (24)$$

여기서, $e_{i,j}$ 은 서버 $S^{i,j}$ 의 상대적 방문율이고, $g_{i,j/(M_i+1)}(n)$ 은 (M_i+1) 개의 서버 중 서버 $S^{i,j}$ 를 제외하고 계산한 정규화 상수이다.

동등 승법형 대기네트워크를 구성하는 서버 $S^{i,j}$ 의 상태종속 산출율이 구해지면, 하부네트워크 $R^{i,j}$ 의 상태종속 도착율은 다음과 같이 구해진다.

$$\lambda_{i,j}(n_{i,j}) = X_{i,j}(n_{i,j}), \quad n_{i,j} = 0, 1, \dots, N_i, \quad j = 1, \dots, M_i. \quad (25)$$

위와 같은 방법으로 하부네트워크의 안정상태 확률이 구해지면, 하부네트워크의 상태종속 산출율을 계산할 수 있고, 이 값은 곧 동등 승법형 네트워크의 상태종속 서비스율로 사용된다. 또한 동등 승법형 네트워크 상태종속 서비스율이 구해지면, 동등 승법형 네트워크의 상태종속 산출율이 계산될 수 있고, 이 값은 곧 하부 네트워크의 상태종속 도착율로 사용될 수 있다. 승법형 근사법에서는 이와 같은 과정을 모든 하부네트워크의 상태종속 서비스율이 수렴할 때까지 반복 수행되며, 이상의 과정을 정리하면 [알고리즘 I]과 같다.

[알고리즘 I]

- Step 0. 4.3절의 [알고리즘 II]에서 주어지는 $\mu_{i,j}(n_{i,j})$ 의 값으로 [알고리즘 I] 시작.
- Step 1. 동등 승법형 대기네트워크의 분석.
 - : 식 (24)과 식 (25)을 이용하여 도착율 $\lambda_{i,j}(n_{i,j})$ 의 계산, $j = 1, \dots, M_i$.
- Step 2. 하부네트워크 $R^{i,j}$ 의 분석, $j = 1, \dots, M_i$.
 - : $P_{i,j}(n_{i,j})$ 을 계산하고, 식 (22)과 식 (23)을 이용하여 $\mu_{i,j}(n_{i,j})$ 의 계산.

Step 3. $\mu_{i,j}(n_{i,j})$ 의 수렴성 검사, $j = 1, \dots, M_i$.
 : $\mu_{i,j}(n_{i,j})$ 이 수렴하지 않으면, Step 1으로 돌아감.

4.4 $\mu_i(n_i)$ 구하기

하부네트워크 R^i 의 동등 승법형 네트워크는 Gordon-Newell 네트워크[3]이다. 따라서 [알고리즘 I]이 수렴하여, 하부네트워크 R^i 의 동등 승법형 네트워크를 구성하는 모든 서버의 상태종속 서비스율이 구해지면, 서버 $S^{i,0}$ 의 안정상태 확률은 다음과 같이 표현된다.

$$P_{i,0}(n_i, 0) = \frac{g_{i,0/(M_i+1)}(N_i - n_{i,0})}{G_i(N_i)} \cdot \frac{e_{i,0}^{n_{i,0}}}{\prod_{k=1}^{n_{i,0}} \mu_{i,0}(k)}, \quad n_{i,0} = 0, 1, \dots, N_i. \quad (26)$$

여기서, $G_i(N_i)$ 는 하부네트워크 R^i 에 대응하는 동등 승법형 대기네트워크의 정규화 상수이다.

부품 i 생산 시스템에 n_i 개 주문이 있을 때의 안정상태 확률을 $\hat{P}_i(n_i)$ 로 정의했을 때, 서버 $S^{i,0}$ 의 안정상태 확률로부터 $\hat{P}_i(n_i)$ 은 다음과 같이 표시된다.

$$\hat{P}_i(n_i) = P_{i,0}(N_i - n_i), \quad n_i = 0, 1, \dots, N_i. \quad (27)$$

부품 i 생산 시스템의 상태종속 산출률 $\nu_i(n_i)$ 은 다음과 같이 정의된다.

$$\nu_i(n_i) = \begin{cases} 0 & , n_i = 0, \\ \lambda_i(n_i - 1) \frac{\hat{P}_i(n_i - 1)}{\hat{P}_i(n_i)} & , n_i = 1, \dots, N_i. \end{cases} \quad (28)$$

식 (1), 식 (26)과 식 (27)에 의해, 식 (28)의 부품 i 생산 시스템의 상태종속 산출률 $\nu_i(n_i)$ 은 다음

과 같이 정리될 수 있다.

$$v_i(n_i) = \begin{cases} 0 & , n_i = 0, \\ e_{i,0} \frac{g_{i,0/(M_i+1)}(n_i-1)}{g_{i,0/(M_i+1)}(n_i)} & , n_i = 1, \dots, N_i. \end{cases} \quad (29)$$

식 (29)에 의해서 부품 i 생산시스템의 상태종속 산출률이 계산되면, 네트워크 - AO의 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 는 다음과 같이 표현된다.

$$\mu_i(n_i) = v_i(n_i), \quad n_i = 0, \dots, N_i. \quad (30)$$

따라서, 부품 i 생산시스템의 상태종속 도착율 $\lambda_i(n_i)$ 을 알고 있다면, 식 (29)과 식 (30)을 이용하여 네트워크 - AO의 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 을 구할 수 있다. 또한, 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 을 구해지면, 3절에서의 언급된 방법으로 부품상태의 결합확률분포를 구할 수 있고, 식 (9)과 식 (10)을 이용하여 부품 i 생산 시스템의 상태종속 도착율 $\lambda_i(n_i)$ 을 구할 수 있다. 그러나 $\mu_i(n_i)$ 과 $\lambda_i(n_i)$ 가 모두 미지수이므로, 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 을 구하는 문제는 고정점 문제가 되며, 본 연구에는 이와 같은 과정을 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 이 수렴할 때까지 반복 수행한다. 이상의 과정을 정리하면 [알고리즘 II]와 같다.

[알고리즘 II]

Step 0. 초기화 과정.

: $\mu_i(n_i)$ 와 $\mu_{i,j}(n_{i,j})$ 을 초기화, $i = 1, 2,$
 $j = 1, \dots, M_i$

Step 1. 네트워크 - AO의 분석.

Step 1.1. $P_{AO}(n_1, n_2)$ 을 구하고, 식 (9)
과 식 (10)을 이용하여 $\lambda_i(n_i)$
을 계산, $i = 1, 2.$

Step 1.2. 식 (1)을 이용하여 $\mu_{i,0}(n_{i,0})$ 을

계산, $i = 1, 2.$

Step 2. 하부네트워크 R^i 의 분석, $i = 1, 2.$

Step 2.1. [알고리즘 I]을 이용하여 하부네트워크 R^i 을 분석.

Step 2.2. 식 (29)과 식 (30)을 이용하여 $\mu_i(n_i)$ 을 계산.

Step 3. $\mu_i(n_i)$ 의 수렴성 검사.

: $\mu_i(n_i)$ 이 수렴하지 않으면, Step 1로 돌아감.

Step 4. 5절의 식을 이용하여 구하고자 하는 성능평가척도 계산.

5. 시스템의 성능평가척도 계산

[알고리즘 II]가 수렴하면, 네트워크 - AO의 안정 상태확률로부터 제품별 납품 충족률, 제품별 대기 시간, 평균 재고 등 다양한 시스템의 성능평가척도를 계산할 수 있다. 본 절에서는 시스템의 성능평가척도를 i) 제품 수요 수준(level), ii) 부품 수준, iii) 주문 - 조립 시스템의 수준으로 세분화하고자 하며, 이에 대한 내용이 아래의 각 절에서 소개된다.

5.1 제품 c 수요의 시스템 성능척도, $c = 1, 2, 3.$

제품 c 수요가 주문 시스템에 요구한 부품이 즉시 만족될 확률 F^c 은 네트워크 - AO의 안정상태 확률로부터 다음과 같이 계산된다.

$$F^i = \sum_{n_i=0}^{S_i-1} P_i(n_i), \quad i = 1, 2. \quad (31)$$

$$F^3 = \sum_{n_1=0}^{S_1-1} \sum_{n_2=0}^{S_2-1} P_{AO}(n_1, n_2) \quad (32)$$

도착한 제품 c 수요가 수용될 확률 SL^c 은 다음과 같이 표현된다.

$$SL^i = 1 - P_i(N_i), \quad i = 1, 2. \quad (33)$$

$$SL^3 = \sum_{n_1=0}^{M_1-1} \sum_{n_2=0}^{N_2-1} P_{AO}(n_1, n_2) \quad (34)$$

도착한 제품 c 수요가 수용될 확률 SL^c 이 구해지면, 제품별 수요의 대기시간에 관계된 성능평가 척도를 계산할 수 있다. 네트워크-AO에서 서버 S^i 의 서비스율이 상태종속이므로, 주문시스템에서 수요의 대기시간은 수요가 도착한 시점에서의 부품의 재고수준뿐만 아니라 대기시간 동안 도착한 수요의 수에 의존하게 된다. 따라서 주문시스템에서 수요의 대기시간은 각 부품창고에서 수요가 앞으로 기다려야 하는 서비스 단계 수(부품의 생산회수)와 부품창고의 현재 재고수준을 모두 고려해야만 정확히 표현될 수 있으며, 이에 본 연구에서는 아래와 같은 의미를 가지는 상태(t_1, b_1, t_2, b_2)을 정의한다.

t_i = 수요가 부품 i 가 충족될 때까지 부품 i 재고창고에서 기다려야 하는 서비스 단계의 수.
 b_i = 부품 i 재고창고에 존재하는 미충족수요의 수.

주문 시스템에서 제품의 부품 수요가 모두 충족되는 시점을 상태 0으로 정의했을 때, 주문 시스템에서 상태 (t_1, b_1, t_2, b_2)에 있는 수요의 대기시간은 상태 (t_1, b_1, t_2, b_2)에서 상태 0에 도달할 때까지의 시간으로 해석될 수 있다. 따라서 상태 (t_1, b_1, t_2, b_2)에서 상태 0에 도달할 때까지의 시간을 $V(t_1, b_1, t_2, b_2)$ 라하고, 이의 평균을 $\bar{V}(t_1, b_1, t_2, b_2)$ 라 정의했을 때, 주문시스템에서 제품 c 수요의 평균 대기시간 \bar{W}^c 은 다음과 같이 계산될 수 있다.

$$\bar{W}^1 = \sum_{t_1=1}^{B_1} \frac{P_1(t_1 + S_1 - 1)}{SL^1} \bar{V}(t_1, t_1, 0, 0) \quad (35)$$

$$\bar{W}^2 = \sum_{t_2=1}^{B_2} \frac{P_2(t_2 + S_2 - 1)}{SL^2} \bar{V}(0, 0, t_2, t_2) \quad (36)$$

$$\begin{aligned} \bar{W}^3 &= \sum_{t_1=1}^{B_1} \bar{V}(t_1, t_1, 0, 0) \cdot \\ &\quad \sum_{n_2=1}^{S_2-1} \frac{P_{AO}(t_1 + S_1 - 1, n_2)}{SL^3} \\ &+ \sum_{t_2=1}^{B_2} \bar{V}(0, 0, t_2, t_2) \cdot \\ &\quad \sum_{n_1=0}^{S_1-1} \frac{P_{AO}(n_1, t_2 + S_2 - 1)}{SL^3} \\ &+ \sum_{t_1=1}^{B_1} \cdot \\ &\quad \sum_{t_2=1}^{B_2} \frac{P_{AO}(t_1 + S_1 - 1, t_2 + S_2 - 1)}{SL^3} \cdot \\ &\quad \bar{V}(t_1, t_1, t_2, t_2) \end{aligned} \quad (37)$$

같은 방법으로 제품 3 수요가 부품 i 제품을 얻기까지의 평균 대기시간 \bar{W}_i^3 은 다음과 같이 계산될 수 있다.

$$\bar{W}_1^3 = \sum_{t_1=1}^{B_1} \bar{V}(t_1, t_1, 0, 0) \cdot \sum_{n_1=0}^{N_2-1} \frac{P_{AO}(t_1 + S_1 - 1, n_2)}{SL^3} \quad (38)$$

$$\bar{W}_2^3 = \sum_{t_2=1}^{B_2} \bar{V}(0, 0, t_2, t_2) \cdot \sum_{n_1=0}^{N_1-1} \frac{P_{AO}(n_1, t_2 + S_2 - 1)}{SL^3} \quad (39)$$

식 (35)~식 (39)에서 알수 있듯이 $\bar{V}(t_1, b_1, t_2, b_2)$ 의 값을 추정할 수 있다면, 모든 제품 수요의 평균 대기시간을 구할 수 있다. 네트워크-AO에서 서버 S^i 의 서비스 시간과 고객의 도착시간분포가 모두 지수분포를 따르므로, $V(t_1, b_1, t_2, b_2)$ 은 연속시간 마코프체인으로 모형화가 가능하다. 그러나 상태 (t_1, b_1, t_2, b_2)가 4차원으로 정의되어, 이를 분석하는 것은 쉽지 않으며, 마코프체인의 복잡성을 줄이기 위해서 다음의 가정을 추가한다.

(A.1) 상태(t_1, b_1, t_2, b_2)의 정의에 의해 $t_i \leq b_i$ 이 성립한다. 그러나 부품 i 가 충족된 제품 i 수요 또는 부품 i 의 다른 부품을 기다리는 제품 3

수요의 관점에서 부품 i 의 재고수준은 중요하지 않으므로, 만약 $t_i = 0$ 이면, $b_i = 0$ 으로 설정하여 마코프체인의 분석범위를 축소한다.

가정 (A.1)에서 $t_i = 0$ 이면, $b_i = 0$ 으로 설정하여, 마코프체인의 분석범위에서 부품 i 의 재고수준을 제외하였으나, 이에 따른 문제점이 발생한다. 예를 들어, 실제모형에서 $t_1 = 0$, $t_2 > 0$ 그리고 $b_2 > 0$ 의 상태에서 $b_1 = B_1$ 가 되면, 제품 3 수요가 시스템에 도착하지 못하므로, 부품 2의 수요율이 $\lambda^2 + \lambda^3$ 에서 λ^2 로 변화하게 된다. 따라서 실제 수요의 도착율을 정확히 반영하기 위해서는 $t_i = 0$ 일 경우에도 부품 i 의 재고수준을 고려하여 마코프체인을 구성하여야 한다. 그러나 마코프체인의 복잡성을 줄이기 위해서는 가정 (A.1)을 유지할 필요가 있으며, 이를 위해 본 연구에서는 다음과 같은 가정 (A.2)을 추가한다.

(A.2) $t_1 = 0$, $t_2 > 0$, $b_2 > 0$ 의 상태에서 가정 (A.1)에 의해 $b_1 = 0$ 으로 설정된 경우, 본 연구에서는 부품 2 생산시스템에 수요의 도착율로 상태종속 도착율 $\lambda_2(n_2)$ 을 사용하며, 이를 통해 실제 수요의 도착율에 근사화한다. 마찬가지로 $t_1 > 0$, $b_1 > 0$, $t_2 = 0$ 의 상태에서 가정 (A.1)에 의해 $b_2 = 0$ 으로 설정된 경우, 부품 1 생산시스템에 수요의 도착율로 상태종속 도착율 $\lambda_1(n_1)$ 을 고객의 도착율로 사용한다.

가정 (A.1)과 가정 (A.2)에 의해 상태 (t_1, b_1, t_2, b_2) 의 축소된 상태 공간 \mathbf{ST}_v 은 다음과 같이 정의된다.

$$\begin{aligned} \mathbf{ST}_v = \{ & (t_1, b_1, t_2, b_2) \mid 0 \leq t \leq B_1 + B_2, \\ & \text{Max} [0, t - B_2] \leq t_1 \leq \text{Min} [t, B_1], \\ & t_1 \leq b_1 \leq \delta(t_1) \cdot B_1, t_2 = t - t_1, \\ & t_2 \leq b_2 \leq \delta(t_2) \cdot B_2 \} \end{aligned} \quad (40)$$

여기서, t 는 상태 (t_1, b_1, t_2, b_2) 에서 수요가 앞으로 기다려야 하는 총 서비스의 단계 수로, $t = t_1 + t_2$ 로 정의되고, 상태 $(0, 0, 0, 0)$ 는 상태 0을 의미하며, $\delta(x)$ 는 다음과 같이 표현된다.

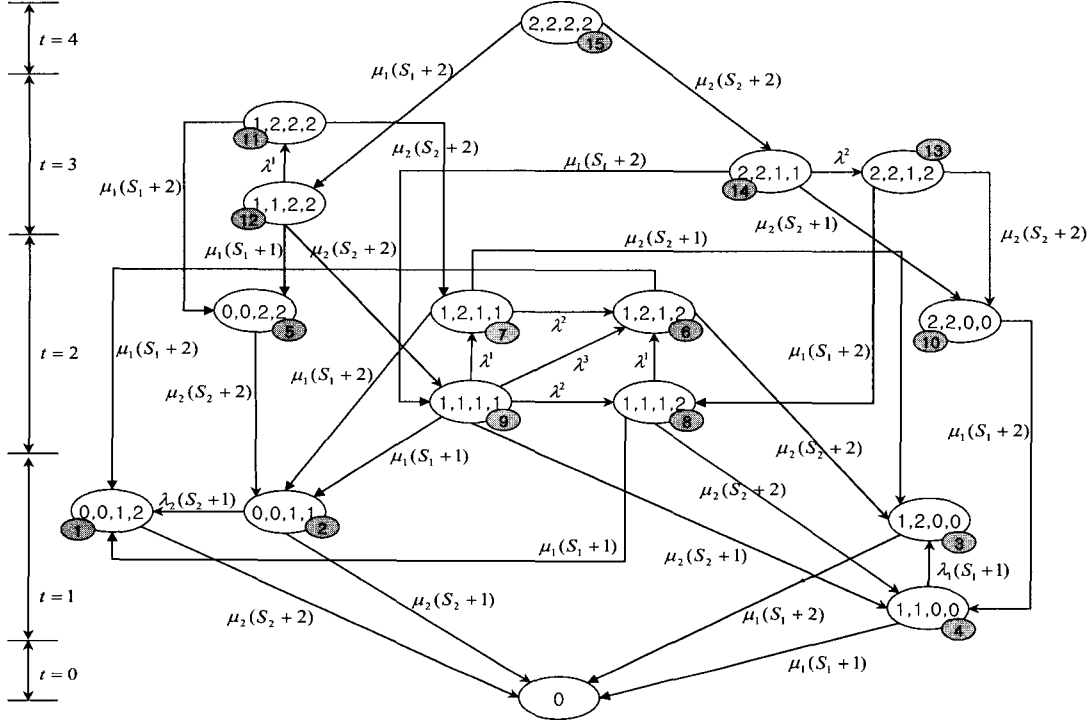
$$\delta(x) = \begin{cases} 1, & \text{if } x > 0, \\ 0, & \text{otherwise} \end{cases}$$

$B_1 = B_2 = 2$ 일 경우에 $V(t_1, b_1, t_2, b_2)$ 의 예는 [그림 11]과 같다.

[그림 11]에서 알 수 있듯이 $V(t_1, b_1, t_2, b_2)$ 은 가정 (A.1)과 가정 (A.2)에 의해서 임의의 상태를 한번 이탈하면, 다시 그 상태를 다시 방문하지 않는 특성을 지니고 있다. [그림 11]에서 작은 원속의 숫자는 각각의 상태를 나타내는 상태번호로서 16개의 상태에 0부터 15까지의 상태번호가 각각 부여되어 있다. [그림 11]로부터 알 수 있듯이 작은 번호가 부여된 상태에서 큰 번호가 부여된 상태로는 방문이 불가능하고 큰 번호가 부여된 상태에서 작은 번호가 부여된 상태로만 방문이 가능하며, 이와 같은 특성을 지니고 있어 마코프체인의 복잡성은 상당히 감소한다. Neut의 이론[1,15]에 의해, $V(t_1, b_1, t_2, b_2)$ 을 초기상태 확률 $P_{A0}(n_1, n_2)$ 을 가지는 단계형 확률변수(PH random variable)로 분석할 수도 있으나, 본 연구에서는 연구의 범위를 $\bar{V}(t_1, b_1, t_2, b_2)$ 의 추정으로 한정하여, 보다 효율적으로 $\bar{V}(t_1, b_1, t_2, b_2)$ 을 구할 수 있는 반복계획(recursive scheme)을 연구하였다. 이를 위해 다음의 기호를 정의하자.

$$\lambda^1(b_1, b_2) = \begin{cases} \lambda^1, & \text{if } 0 < b_1 < B_1 \text{ and } b_2 > 0, \\ \lambda_1(S_1 + b_1), & \text{if } 0 < b_1 < B_1 \text{ and } b_2 = 0, \\ 0, & \text{otherwise.} \end{cases} \quad (41)$$

$$\lambda^2(b_1, b_2) = \begin{cases} \lambda^2, & \text{if } b_1 > 0 \text{ and } 0 < b_2 < B_2, \\ \lambda_2(S_2 + b_2), & \text{if } b_1 = 0 \text{ and } 0 < b_2 < B_2, \\ 0, & \text{otherwise.} \end{cases} \quad (42)$$



[그림 11] $B_1 = B_2 = 2$ 일 경우에 $V(t_1, b_1, t_2, b_2)$ 의 예제

$$\lambda^3(b_1, b_2) = \begin{cases} \lambda^3, & \text{if } 0 < b_1 < B_1 \text{ and } 0 < b_2 < B_2, \\ 0, & \text{otherwise.} \end{cases} \quad (43)$$

[그림 11]으로부터 $\bar{V}(t_1, b_1, t_2, b_2)$ 은 다음과 같은 반복적 수식으로 표현될 수 있다.

$$\bar{V}(t_1, b_1, t_2, b_2) = 1/O(b_1, b_2) + \sum_{i=1}^5 \bar{V}_i(t_1, b_1, t_2, b_2) \quad (44)$$

식 (44)에서 $O(b_1, b_2)$ 은 상태 (t_1, b_1, t_2, b_2) 에서 이탈률로서 다음과 같이 표현된다.

$$O(b_1, b_2) = \lambda^1(b_1, b_2) + \lambda^2(b_1, b_2) + \lambda^3(b_1, b_2) + \delta(b_1) \cdot \mu_1(b_1 + S_1) + \delta(b_2) \cdot \mu_2(b_2 + S_2) \quad (45)$$

식 (44)에서 $\bar{V}_i(t_1, b_1, t_2, b_2)$ 은 식 (45)을 구성하는 각 사건의 발생으로 상태 (t_1, b_1, t_2, b_2) 가 다음 상태로 이동했을 때, 다음 상태에서 상태 0으로 흡수 될 때까지의 평균 시간으로, 다음과 같이 5가지의 식으로 정리된다.

$$\bar{V}_1(t_1, b_1, t_2, b_2) = \begin{cases} \frac{\lambda^1(b_1, b_2)}{O(b_1, b_2)} \bar{V}(t_1, b_1 + 1, t_2, b_2), & \text{if } 0 < b_1 < B_1, \\ 0, & \text{otherwise.} \end{cases} \quad (46)$$

$$\bar{V}_2(t_1, b_1, t_2, b_2) = \begin{cases} \frac{\lambda^2(b_1, b_2)}{O(b_1, b_2)} \bar{V}(t_1, b_1, t_2, b_2 + 1), & \text{if } 0 < b_2 < B_2, \\ 0, & \text{otherwise.} \end{cases} \quad (47)$$

$$\overline{V}_3(t_1, b_1, t_2, b_2) = \begin{cases} \frac{\lambda^3(b_1, b_2)}{O(b_1, b_2)} \overline{V}(t_1, b_1 + 1, t_2, b_2 + 1), \\ \quad \text{if } 0 < b_1 < B_1 \text{ and } 0 < b_2 < B_2, \\ 0, \quad \text{otherwise.} \end{cases} \quad (48)$$

$$\overline{V}_4(t_1, b_1, t_2, b_2) = \begin{cases} \frac{\mu_1(b_1 + S_1)}{O(b_1, b_2)} \overline{V}(t_1 - 1, b_1 - 1, t_2, b_2), \\ \quad \text{if } t_1 > 1, \\ \frac{\mu_1(b_1 + S_1)}{O(b_1, b_2)} \overline{V}(0, 0, t_2, b_2), \\ \quad \text{if } t_1 = 1, \text{ and } t_2 > 0, \\ 0, \quad \text{otherwise.} \end{cases} \quad (49)$$

$$\overline{V}_5(t_1, b_1, t_2, b_2) = \begin{cases} \frac{\mu_2(b_2 + S_2)}{O(b_1, b_2)} \overline{V}(t_1, b_1, t_2 - 1, b_2 - 1), \\ \quad \text{if } t_2 > 1, \\ \frac{\mu_2(b_2 + S_2)}{O(b_1, b_2)} \overline{V}(t_1, b_1, 0, 0), \\ \quad \text{if } t_1 > 0, \text{ and } t_2 = 1, \\ 0, \quad \text{otherwise.} \end{cases} \quad (50)$$

앞에서 언급했듯이 $V(t_1, b_1, t_2, b_2)$ 은 임의의 상태를 한번 이탈하면, 다시 그 상태를 다시 방문하지 않는 특성을 지니고 있다. 따라서 마코프 체인의 이와 같은 특성을 이용하여 식 (44)의 실행 순서를 정리하면, $\overline{V}(t_1, b_1, t_2, b_2)$ 의 값을 순차적으로 계산할 수 있다. 본 연구에서 관찰된 식 (44)의 실행순서는 아래에서 주어진 반복계획으로 정리될 수 있으며, [그림 11]의 예제에 아래의 반복계획을 적용하면 각 상태의 번호 순서대로 식 (44)가 실행되어 $\overline{V}(t_1, b_1, t_2, b_2)$ 을 순차적으로 계산할 수 있다.

[반복 계획]

```

for t = 1 to B1 + B2
  for t1 = Max[0, t - B2] to Min[t, B1]
    t2 = t - t1
    for b1 = δ(t1) · B1 downto t1
      for b2 = δ(t2) · B2 downto t2
         $\overline{V}(t_1, b_1, t_2, b_2) = 1/O(b_1, b_2)$ 
        +  $\sum_{i=1}^5 \overline{V}_i(t_1, b_1, t_2, b_2)$ 
      end
    end
  end
end
  
```

위의 반복계획을 실행하여 $\overline{V}(t_1, b_1, t_2, b_2)$ 의 값을 구해지면, 식 (35)~식 (39)을 이용하여 각 제품 수요의 평균 대기시간을 계산할 수 있다. 각 제품 수요의 평균 대기시간이 구해지면, 주문시스템에서 제품 c 수요의 평균 대기고객 수 \overline{BI}^c 와 부품 i 생산 시스템에서 제품 3 수요의 평균 대기고객 수 \overline{BI}_i^c 는 Little의 법칙을 적용하여 다음과 같이 구해질 수 있다.

$$\overline{BI}^c = SL^c \cdot \lambda^c \cdot \overline{W}^c \quad (51)$$

$$\overline{BI}_i^3 = SL^3 \cdot \lambda^3 \cdot \overline{W}_i^3, \quad i = 1, 2. \quad (52)$$

제품 3 수요의 평균 대기시간과 평균 대기고객 수로부터, 조립시간을 포함한 제품 3 수요의 평균 시스템 체류시간 \overline{SJ}^3 과 체류고객 수 \overline{BS}^3 는 다음과 같이 표현된다.

$$\overline{SJ}^3 = \overline{W}^3 + \overline{A} \quad (53)$$

$$\overline{BS}^3 = SL^3 \cdot \lambda^3 \cdot \overline{SJ}^3 \quad (54)$$

5.2 부품 i 생산시스템의 시스템 성능척도, $i = 1, 2.$

제품수준의 성능평가척도가 계산되면, 이를 이

용하여 각 부품 생산시스템의 성능평가척도를 구할 수 있다. 우선 부품 i 재고창고에서 부품 i 의 수요가 즉시 만족될 확률 F_i 와 부품 i 수요가 수용될 확률 SL_i 은 다음과 같다.

$$F_i = \frac{\lambda^i}{\lambda_i} F^i + \frac{\lambda^3}{\lambda_i} F^3 \quad (55)$$

$$SL_i = \frac{\lambda^i}{\lambda_i} SL^i + \frac{\lambda^3}{\lambda_i} SL^3 \quad (56)$$

다음으로 부품 i 수요의 평균 대기시간 \overline{W}_i 과 부품 i 재고창고에 평균 미충족수요의 수 \overline{BI}_i 는 다음과 같이 계산된다.

$$\overline{W}_i = \frac{\lambda^i}{\lambda_i} \overline{W}^i + \frac{\lambda^3}{\lambda_i} \overline{W}^3 \quad (57)$$

$$\begin{aligned} \overline{BI}_i &= \sum_{n_i=S_i+1}^{N_i} (n_i - S_i) P_i(n_i) \\ &= \overline{BI}^i + \overline{BI}^3 \end{aligned} \quad (58)$$

부품 i 의 재고 창고에 평균 재고수준 \overline{I}_i 과 작업장(i, j)에 평균 주문 수 $\overline{WS}_{i,j}$ 는 다음과 같이 계산된다.

$$\overline{I}_i = \sum_{n_i=0}^{S_i-1} (S_i - n_i) P_i(n_i) \quad (59)$$

$$\overline{WS}_{i,j} = \sum_{n=1}^{N_i} n \cdot P_{i,j}(n), \quad j = 1, \dots, M_i. \quad (60)$$

5.3 주문-조립 시스템의 성능척도

주문-조립 시스템에서 제품의 수요가 즉시 만족될 확률 F 와 수용될 확률 SL 은 다음과 같이 계산된다.

$$F = q^1 F^1 + q^2 F^2 + q^3 F^3 \quad (61)$$

$$SL = q^1 SL^1 + q^2 SL^2 + q^3 SL^3 \quad (62)$$

주문 시스템에서 임의의 제품 수요의 평균 대기시간 \overline{W} 과 대기 중인 평균 수요의 수 \overline{BI} 는 다음과 같이 계산된다.

$$\overline{W} = q^1 \overline{W}^1 + q^2 \overline{W}^2 + q^3 \overline{W}^3 \quad (63)$$

$$\overline{BI} = \overline{BI}^1 + \overline{BI}^2 + \overline{BI}^3 \quad (64)$$

마지막으로 주문-조립 시스템에서 임의의 고객의 체류시간 \overline{SJ} 와 체류 고객 수 \overline{BS} 는 다음과 같이 표시될 수 있다.

$$\overline{SJ} = \overline{W} + q^3 \overline{A} \quad (65)$$

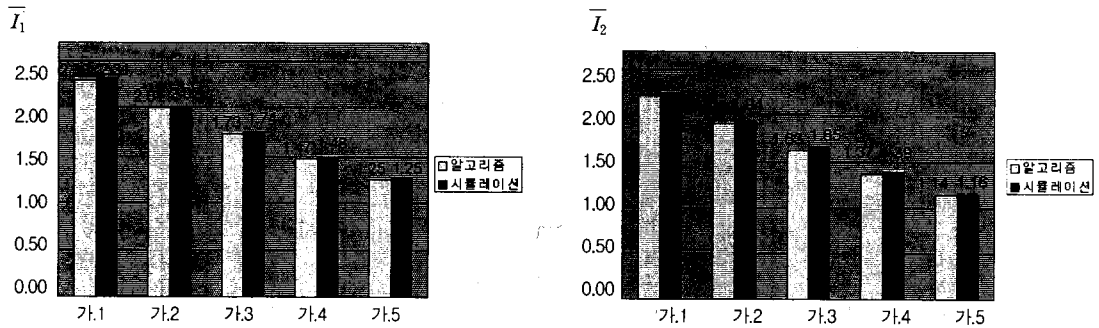
$$\overline{BS} = \overline{BI}^1 + \overline{BI}^2 + \overline{BI}^3 \quad (66)$$

6. 근사적 분석의 정확성 평가

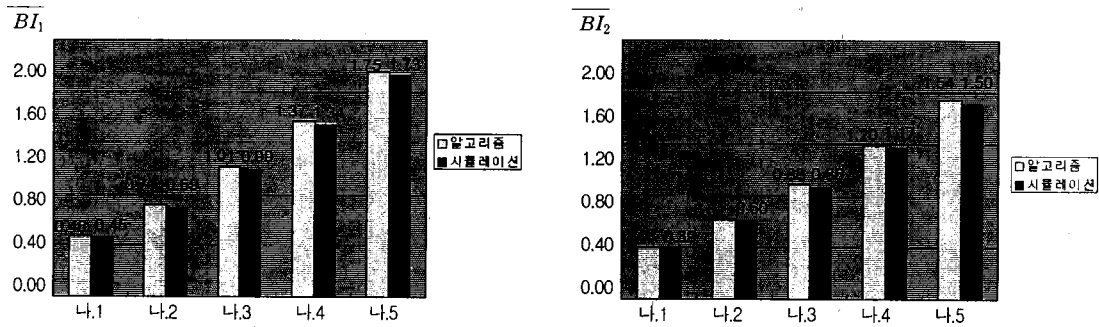
본 연구에서는 시스템의 특성을 관찰하기 위하여 많은 실험을 수행하였으며, 제안된 근사법의 정확성을 평가하기 위하여 근사법에서 얻어진 값과 시뮬레이션에서 얻어진 값을 비교하였다. 시뮬레이션의 종료조건으로 시뮬레이션 시간을 10^6 로 설정하여 10번 반복하였고, 알고리즘의 종료조건은 10^{-6} 이하로 하였다. 본 연구에서 실험한 예제에서 [알고리즘 I]과 [알고리즘 II]의 반복 횟수는 대부분 5번 이내의 반복에서 수렴함을 관측할 수 있었고, 모든 예제에서 수렴하였다. 본 절에서는 파라미터 값이 변화함에 따라 알고리즘의 정확도가 어떻게 변화하는지 알아보기 위해 대표적인 4개의 예제 군을 선정하여 소개하고자 한다. 예제 가 군에서는 도착률 λ 의 값을 변화시켜 보았다. 예제 나 군에서는 q^3 의 비율 증가에 따른 부품들 간의 상호종속관계의 증가가 알고리즘의 정확성에 미치는 영향을 조사해 보았다. 예제 다 군에서는 최고 재고수준과 최대 미충족수요의 수를 변화시켜 보았으며, 예제 라 군에서는 생산단계의 수가 증가함에 따라서 알고리즘의 정확도가 어떻게 변화되는지를 관찰해 보았다. 편의상 모든 예제 군에서 같은 부품 생산시스템에 속하는 모든 작업장에서 가공시간의 분포는 동일하다고 가정하였다. 실험된 데이터는 <표 2>에서 정리되며, 실험결과는 <표 3>~<표 5>와 [그림 12]~[그림 14]에서 요약된다.

〈표 2〉 실험 데이터

실험번호	변수	λ	q^1	q^2	q^3	\bar{A}	S_1	S_2	B_1	B_2	M_1	M_2	$SE_{1,j}$	$\mu_{1,j}^1 = \mu_{1,j}^2$	$c_{1,j}$	$SE_{2,j}$	$\mu_{2,j}^1 = \mu_{2,j}^2$	$c_{2,j}$
가.1	1.1	1/3	1/3	1/3	0.1	5	5	5	5	2	2	2	1.0	0.5	2	1.1	0.7	
가.2	1.2	1/3	1/3	1/3	0.1	5	5	5	5	2	2	2	1.0	0.5	2	1.1	0.7	
가.3	1.3	1/3	1/3	1/3	0.1	5	5	5	5	2	2	2	1.0	0.5	2	1.1	0.7	
가.4	1.4	1/3	1/3	1/3	0.1	5	5	5	5	2	2	2	1.0	0.5	2	1.1	0.7	
가.5	1.5	1/3	1/3	1/3	0.1	5	5	5	5	2	2	2	1.0	0.5	2	1.1	0.7	
나.1	1.7	0.40	0.40	0.20	1.0	8	8	8	8	4	4	3	1.1	0.5	3	1.2	0.6	
나.2	1.7	0.35	0.35	0.30	1.0	8	8	8	8	4	4	3	1.1	0.5	3	1.2	0.6	
나.3	1.7	0.30	0.30	0.40	1.0	8	8	8	8	4	4	3	1.1	0.5	3	1.2	0.6	
나.4	1.7	0.25	0.25	0.50	1.0	8	8	8	8	4	4	3	1.1	0.5	3	1.2	0.6	
나.5	1.7	0.20	0.20	0.60	1.0	8	8	8	8	4	4	3	1.1	0.5	3	1.2	0.6	
다.1	2.0	6/15	4/15	5/15	1.0	7	6	3	3	6	6	4	1.5	0.5	4	1.4	0.5	
다.2	2.0	6/15	4/15	5/15	1.0	8	7	4	4	6	6	4	1.5	0.5	4	1.4	0.5	
다.3	2.0	6/15	4/15	5/15	1.0	9	8	5	5	6	6	4	1.5	0.5	4	1.4	0.5	
다.4	2.0	6/15	4/15	5/15	1.0	10	9	6	6	6	6	4	1.5	0.5	4	1.4	0.5	
다.5	2.0	6/15	4/15	5/15	1.0	11	10	7	7	6	6	4	1.5	0.5	4	1.4	0.5	
라.1	2.2	1/3	1/3	1/3	1.0	10	10	5	5	3	3	3	1.0	0.4	3	1.0	0.5	
라.2	1.7	1/3	1/3	1/3	1.0	10	10	5	5	5	5	3	1.0	0.4	3	1.0	0.5	
라.3	1.5	1/3	1/3	1/3	1.0	10	10	5	5	7	7	3	1.0	0.4	3	1.0	0.5	
라.4	1.2	1/3	1/3	1/3	1.0	10	10	5	5	8	8	3	1.0	0.4	3	1.0	0.5	
라.5	1.0	1/3	1/3	1/3	1.0	10	10	5	5	10	10	3	1.0	0.4	3	1.0	0.5	



[그림 12] 재고창고의 평균 재고수준(예제 가 군)



[그림 13] 재고창고의 평균 미충족수요 수(예제 나 군)

〈표 3〉 제품 수요가 즉시 만족될 확률

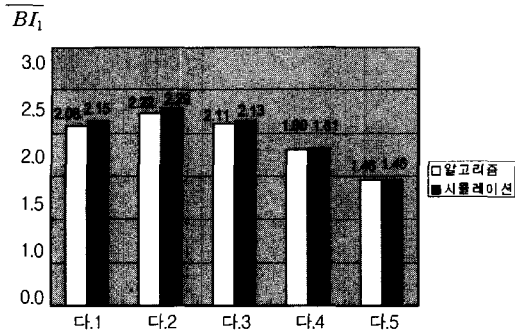
실험 번호	알고리즘(상대오차율 %)						시뮬레이션(+/-신뢰구간)					
	F ¹	F ²	F ³	F ₁	F ₂	F	F ¹	F ²	F ³	F ₁	F ₂	F
가.1	0.7911 (-0.3401)	0.7770 (-0.6775)	0.6374 (-2.1342)	0.7143 (-1.1349)	0.7072 (-1.3393)	0.7352 (-0.9698)	0.7938 (0.0007)	0.7823 (0.0008)	0.6513 (0.0007)	0.7225 (0.0007)	0.7168 (0.0007)	0.7424 (0.0006)
가.2	0.7239 (-0.4401)	0.7052 (-0.9829)	0.5376 (-3.0478)	0.6308 (-1.5605)	0.6214 (-1.8945)	0.6556 (-1.3542)	0.7271 (0.0007)	0.7122 (0.0005)	0.5545 (0.0007)	0.6408 (0.0007)	0.6334 (0.0006)	0.6646 (0.0006)
가.3	0.6519 (-0.5037)	0.6289 (-1.0697)	0.4388 (-3.9194)	0.5454 (-1.8888)	0.5337 (-2.2885)	0.5731 (-1.6137)	0.6552 (0.0009)	0.6357 (0.0009)	0.4567 (0.0010)	0.5559 (0.0009)	0.5462 (0.0009)	0.5825 (0.0009)
가.4	0.5791 (-0.5495)	0.5518 (-1.1996)	0.3474 (-4.8740)	0.4632 (-2.2166)	0.4496 (-2.6418)	0.4927 (-1.8526)	0.5823 (0.0007)	0.5585 (0.0010)	0.3652 (0.0006)	0.4737 (0.0005)	0.4618 (0.0007)	0.5020 (0.0005)
가.5	0.5089 (-0.3915)	0.4785 (-1.0137)	0.2677 (-5.6398)	0.3883 (-2.2653)	0.3731 (-2.7372)	0.4184 (-1.7840)	0.5109 (0.0008)	0.4834 (0.0008)	0.2837 (0.0006)	0.3973 (0.0007)	0.3836 (0.0007)	0.4260 (0.0006)
나.1	0.7150 (-0.2233)	0.7395 (-0.4175)	0.5484 (-2.5240)	0.6595 (-0.9760)	0.6758 (-0.9962)	0.6915 (-0.7464)	0.7166 (0.0005)	0.7426 (0.0005)	0.5626 (0.0006)	0.6660 (0.0005)	0.6826 (0.0005)	0.6967 (0.0004)
나.2	0.6280 (-0.6172)	0.6570 (-0.6502)	0.4450 (-4.7517)	0.5435 (-2.2306)	0.5592 (-2.2036)	0.5832 (-1.6194)	0.6319 (0.0005)	0.6613 (0.0005)	0.4672 (0.0003)	0.5559 (0.0003)	0.5718 (0.0003)	0.5928 (0.0002)
나.3	0.5384 (-0.7923)	0.5711 (-0.9023)	0.3500 (-7.6761)	0.4307 (-4.1184)	0.4447 (-4.0768)	0.4728 (-2.9756)	0.5427 (0.0008)	0.5763 (0.0008)	0.3791 (0.0007)	0.4492 (0.0007)	0.4636 (0.0007)	0.4873 (0.0007)
나.4	0.4519 (-0.7031)	0.4872 (-1.0360)	0.2678 (-11.2069)	0.3292 (-6.6893)	0.3409 (-6.6539)	0.3687 (-4.9007)	0.4551 (0.0008)	0.4923 (0.0006)	0.3016 (0.0005)	0.3528 (0.0006)	0.3652 (0.0005)	0.3877 (0.0005)
나.5	0.3733 (-0.3204)	0.4101 (-0.8702)	0.2005 (-15.1502)	0.2437 (-10.0074)	0.2529 (-9.8717)	0.2770 (-7.4816)	0.3745 (0.0007)	0.4137 (0.0004)	0.2363 (0.0004)	0.2708 (0.0004)	0.2806 (0.0003)	0.2994 (0.0004)
다.1	0.3530 (1.9936)	0.3471 (2.4801)	0.1313 (-10.7410)	0.2522 (-1.3688)	0.2272 (-2.1954)	0.2775 (-0.1080)	0.3461 (0.0004)	0.3387 (0.0003)	0.1471 (0.0002)	0.2557 (0.0003)	0.2323 (0.0002)	0.2778 (0.0002)
다.2	0.4094 (1.0116)	0.4168 (1.2634)	0.1913 (-9.8917)	0.3103 (-2.2985)	0.2915 (-3.1240)	0.3387 (-1.1672)	0.4053 (0.0004)	0.4116 (0.0004)	0.2123 (0.0005)	0.3176 (0.0004)	0.3009 (0.0004)	0.3427 (0.0004)
다.3	0.4970 (0.3027)	0.5165 (0.3887)	0.2874 (-7.8846)	0.4018 (-2.4994)	0.3892 (-3.1841)	0.4323 (-1.6158)	0.4955 (0.0005)	0.5145 (0.0005)	0.3120 (0.0004)	0.4121 (0.0004)	0.4020 (0.0004)	0.4394 (0.0004)
다.4	0.6021 (0.0665)	0.6306 (0.1111)	0.4136 (-5.4196)	0.5164 (-1.9928)	0.5101 (-2.4479)	0.5469 (-1.3528)	0.6017 (0.0006)	0.6299 (0.0006)	0.4373 (0.0005)	0.5269 (0.0005)	0.5229 (0.0004)	0.5544 (0.0004)
다.5	0.7068 (-0.0566)	0.7396 (-0.0541)	0.5521 (-3.5633)	0.6365 (-1.4706)	0.6354 (-1.7929)	0.6640 (-1.0432)	0.7072 (0.0006)	0.7400 (0.0006)	0.5725 (0.0007)	0.6460 (0.0006)	0.6470 (0.0006)	0.6710 (0.0006)
라.1	0.6987 (-0.3281)	0.5983 (-0.6806)	0.4369 (-3.6392)	0.5678 (-1.6286)	0.5176 (-1.9511)	0.5780 (-1.2978)	0.7010 (0.0006)	0.6024 (0.0008)	0.4534 (0.0006)	0.5772 (0.0004)	0.5279 (0.0007)	0.5856 (0.0004)
라.2	0.6262 (0.1760)	0.5320 (-0.2999)	0.3575 (-5.6230)	0.4919 (-2.0120)	0.4448 (-2.4989)	0.5053 (-1.4049)	0.6251 (0.0007)	0.5336 (0.0006)	0.3788 (0.0008)	0.5020 (0.0006)	0.4562 (0.0007)	0.5125 (0.0006)
라.3	0.4709 (0.7057)	0.3767 (0.6681)	0.1969 (-10.0503)	0.3339 (-2.7098)	0.2868 (-3.3041)	0.3482 (-1.5271)	0.4676 (0.0009)	0.3742 (0.0008)	0.2189 (0.0006)	0.3432 (0.0006)	0.2966 (0.0007)	0.3536 (0.0005)
라.4	0.5812 (0.1896)	0.4912 (0.1223)	0.3127 (-7.5672)	0.4469 (-2.6786)	0.4020 (-3.0157)	0.4618 (-1.6819)	0.5801 (0.0007)	0.4906 (0.0012)	0.3383 (0.0008)	0.4592 (0.0006)	0.4145 (0.0010)	0.4697 (0.0007)
라.5	0.5566 (0.3787)	0.4674 (0.4513)	0.2874 (-8.4422)	0.4220 (-2.8098)	0.3774 (-3.1314)	0.4371 (-1.5869)	0.5545 (0.0009)	0.4653 (0.0009)	0.3139 (0.0008)	0.4342 (0.0008)	0.3896 (0.0008)	0.4446 (0.0008)

〈표 4〉 제품 수요가 수용될 확률

실험 번호	알고리즘(상대오차율 %)						시뮬레이션(+/-신뢰구간)					
	SL ¹	SL ²	SL ³	SL ₁	SL ₂	SL	SL ¹	SL ²	SL ³	SL ₁	SL ₂	SL
가.1	0.9938 (-0.0302)	0.9934 (-0.0704)	0.9873 (-0.1012)	0.9905 (-0.0706)	0.9903 (-0.0908)	0.9915 (-0.0706)	0.9941 (0.0000)	0.9941 (0.0000)	0.9883 (0.0000)	0.9912 (0.0001)	0.9912 (0.0001)	0.9922 (0.0001)
가.2	0.9886 (-0.0607)	0.9877 (-0.1314)	0.9765 (-0.2146)	0.9825 (-0.1321)	0.9821 (-0.1829)	0.9843 (-0.1319)	0.9892 (0.0001)	0.9890 (0.0001)	0.9786 (0.0002)	0.9838 (0.0001)	0.9839 (0.0002)	0.9856 (0.0001)
가.3	0.9808 (-0.1324)	0.9792 (-0.2140)	0.9606 (-0.3527)	0.9707 (-0.2364)	0.9699 (-0.2776)	0.9736 (-0.2255)	0.9821 (0.0002)	0.9813 (0.0002)	0.9640 (0.0003)	0.9730 (0.0002)	0.9726 (0.0002)	0.9758 (0.0002)
가.4	0.9704 (-0.1646)	0.9677 (-0.3091)	0.9372 (-0.7098)	0.9548 (-0.3236)	0.9534 (-0.4074)	0.9591 (-0.3222)	0.9720 (0.0001)	0.9707 (0.0002)	0.9439 (0.0002)	0.9579 (0.0001)	0.9573 (0.0002)	0.9622 (0.0001)
가.5	0.9573 (-0.2293)	0.9533 (-0.3866)	0.9126 (-0.6532)	0.9350 (-0.4260)	0.9330 (-0.5118)	0.9411 (-0.4127)	0.9595 (0.0002)	0.9570 (0.0001)	0.9186 (0.0002)	0.9390 (0.0002)	0.9378 (0.0002)	0.9450 (0.0001)
나.1	0.9980 (-0.0100)	0.9986 (-0.0100)	0.9966 (-0.0201)	0.9976 (-0.0100)	0.9979 (-0.0100)	0.9980 (-0.0100)	0.9981 (0.0000)	0.9987 (0.0000)	0.9968 (0.0000)	0.9977 (0.0000)	0.9980 (0.0000)	0.9981 (0.0000)
나.2	0.9957 (-0.0301)	0.9968 (-0.0301)	0.9926 (-0.0604)	0.9943 (-0.0402)	0.9949 (-0.0402)	0.9952 (-0.0301)	0.9960 (0.0000)	0.9971 (0.0000)	0.9932 (0.0000)	0.9947 (0.0000)	0.9953 (0.0000)	0.9955 (0.0000)
나.3	0.9918 (-0.0605)	0.9938 (-0.0704)	0.9857 (-0.1418)	0.9883 (-0.1112)	0.9892 (-0.1010)	0.9900 (-0.0908)	0.9924 (0.0001)	0.9945 (0.0001)	0.9871 (0.0001)	0.9894 (0.0001)	0.9902 (0.0001)	0.9909 (0.0001)
나.4	0.9858 (-0.1216)	0.9892 (-0.1212)	0.9752 (-0.2659)	0.9787 (-0.2141)	0.9799 (-0.2138)	0.9813 (-0.1932)	0.9870 (0.0001)	0.9904 (0.0000)	0.9778 (0.0001)	0.9808 (0.0001)	0.9820 (0.0001)	0.9832 (0.0001)
나.5	0.9776 (-0.1940)	0.9828 (-0.2234)	0.9609 (-0.4558)	0.9651 (-0.3819)	0.9664 (-0.3917)	0.9686 (-0.3600)	0.9795 (0.0001)	0.9850 (0.0001)	0.9653 (0.0001)	0.9688 (0.0001)	0.9702 (0.0001)	0.9721 (0.0001)
다.1	0.8631 (-0.0926)	0.8708 (-0.1491)	0.7497 (-0.8071)	0.8115 (-0.4050)	0.8035 (-0.4954)	0.8273 (-0.3373)	0.8639 (0.0002)	0.8721 (0.0001)	0.7558 (0.0003)	0.8148 (0.0002)	0.8075 (0.0002)	0.8301 (0.0002)
다.2	0.9278 (-0.1077)	0.9377 (-0.1491)	0.8701 (-0.4690)	0.9016 (-0.2655)	0.9001 (-0.3212)	0.9112 (-0.2299)	0.9288 (0.0002)	0.9391 (0.0002)	0.8742 (0.0003)	0.9040 (0.0002)	0.9030 (0.0002)	0.9133 (0.0002)
다.3	0.9675 (-0.0723)	0.9750 (-0.0820)	0.9436 (-0.2115)	0.9566 (-0.1357)	0.9576 (-0.1460)	0.9615 (-0.1246)	0.9682 (0.0001)	0.9758 (0.0001)	0.9456 (0.0002)	0.9579 (0.0002)	0.9590 (0.0002)	0.9627 (0.0001)
다.4	0.9877 (-0.0304)	0.9918 (-0.0202)	0.9797 (-0.0612)	0.9840 (-0.0508)	0.9851 (-0.0406)	0.9861 (-0.0405)	0.9880 (0.0001)	0.9920 (0.0001)	0.9803 (0.0001)	0.9845 (0.0001)	0.9855 (0.0001)	0.9865 (0.0001)
다.5	0.9960 (-0.0100)	0.9978 (-0.0100)	0.9938 (-0.0201)	0.9950 (-0.0201)	0.9956 (-0.0100)	0.9958 (-0.0100)	0.9961 (0.0000)	0.9979 (0.0000)	0.9940 (0.0000)	0.9952 (0.0000)	0.9957 (0.0000)	0.9959 (0.0000)
라.1	0.9755 (-0.1331)	0.9592 (-0.2807)	0.9357 (-0.4363)	0.9556 (-0.2817)	0.9475 (-0.3471)	0.9568 (-0.2814)	0.9768 (0.0002)	0.9619 (0.0003)	0.9398 (0.0003)	0.9583 (0.0002)	0.9508 (0.0003)	0.9595 (0.0002)
라.2	0.9740 (-0.1026)	0.9590 (-0.2081)	0.9342 (-0.3520)	0.9541 (-0.2301)	0.9466 (-0.2844)	0.9557 (-0.2193)	0.9750 (0.0001)	0.9610 (0.0002)	0.9375 (0.0003)	0.9563 (0.0002)	0.9493 (0.0002)	0.9578 (0.0002)
라.3	0.9536 (-0.1780)	0.9312 (-0.2571)	0.8878 (-0.5712)	0.9207 (-0.3679)	0.9095 (-0.4161)	0.9242 (-0.3343)	0.9553 (0.0001)	0.9336 (0.0003)	0.8929 (0.0003)	0.9241 (0.0001)	0.9133 (0.0003)	0.9273 (0.0002)
라.4	0.9742 (-0.1025)	0.9600 (-0.1352)	0.9355 (-0.2984)	0.9548 (-0.2090)	0.9477 (-0.2211)	0.9566 (-0.1774)	0.9752 (0.0001)	0.9613 (0.0002)	0.9383 (0.0002)	0.9568 (0.0001)	0.9498 (0.0002)	0.9583 (0.0001)
라.5	0.9727 (-0.1027)	0.9580 (-0.1043)	0.9320 (-0.2995)	0.9524 (-0.1886)	0.9450 (-0.2007)	0.9542 (-0.1674)	0.9737 (0.0002)	0.9590 (0.0001)	0.9348 (0.0002)	0.9542 (0.0002)	0.9469 (0.0001)	0.9558 (0.0001)

〈표 5〉 제품 수요의 평균 대기시간

실험 번호	알고리즘(상대오차율 %)						시뮬레이션(+/-신뢰구간)					
	W ¹	W ²	W ³	W ₁	W ₂	W	W ¹	W ²	W ³	W ₁	W ₂	W
가.1	0.3497 (2.2515)	0.3745 (3.5675)	0.6544 (5.0401)	0.3485 (2.1395)	0.3733 (3.8387)	0.4595 (3.9123)	0.3420 (0.0020)	0.3616 (0.0031)	0.6230 (0.0025)	0.3412 (0.0018)	0.3595 (0.0029)	0.4422 (0.0018)
가.2	0.4895 (1.9367)	0.5268 (4.1518)	0.9005 (5.1617)	0.4874 (2.0519)	0.5248 (4.1890)	0.6389 (4.0384)	0.4802 (0.0025)	0.5058 (0.0023)	0.8563 (0.0027)	0.4776 (0.0023)	0.5037 (0.0019)	0.6141 (0.0022)
가.3	0.6504 (2.2642)	0.7205 (6.3783)	1.1754 (4.9090)	0.6474 (2.3396)	0.6996 (3.6905)	0.8428 (3.8827)	0.6360 (0.0029)	0.6773 (0.0035)	1.1204 (0.0044)	0.6326 (0.0027)	0.6747 (0.0032)	0.8113 (0.0033)
가.4	0.8252 (1.5881)	0.8934 (2.9737)	1.4650 (4.7476)	0.8213 (1.8983)	0.8897 (3.2494)	1.0612 (3.4106)	0.8123 (0.0029)	0.8676 (0.0039)	1.3986 (0.0033)	0.8060 (0.0028)	0.8617 (0.0034)	1.0262 (0.0021)
가.5	1.0057 (1.3810)	1.0900 (2.2322)	1.7548 (4.1981)	1.0014 (1.5413)	1.0860 (2.5496)	1.2835 (2.8940)	0.9920 (0.0033)	1.0662 (0.0023)	1.6841 (0.0034)	0.9862 (0.0031)	1.0590 (0.0021)	1.2474 (0.0024)
나.1	0.4550 (1.7214)	0.3913 (1.6364)	0.7598 (4.1678)	0.4549 (1.7673)	0.3911 (1.7695)	0.4905 (2.4436)	0.4473 (0.0011)	0.3850 (0.0016)	0.7294 (0.0022)	0.4470 (0.0011)	0.3843 (0.0016)	0.4788 (0.0012)
나.2	0.6441 (1.7536)	0.5579 (2.1982)	1.0368 (5.6558)	0.6433 (1.9008)	0.5570 (2.3897)	0.7317 (3.4936)	0.6330 (0.0015)	0.5459 (0.0011)	0.9813 (0.0019)	0.6313 (0.0012)	0.5440 (0.0011)	0.7070 (0.0010)
나.3	0.8631 (1.9249)	0.7525 (2.4367)	1.3409 (6.9725)	0.8609 (2.1355)	0.7500 (2.7820)	1.0210 (4.6321)	0.8468 (0.0034)	0.7346 (0.0033)	1.2535 (0.0046)	0.8429 (0.0035)	0.7297 (0.0033)	0.9758 (0.0033)
나.4	1.1006 (1.6439)	0.9650 (2.6268)	1.6554 (7.8858)	1.0961 (1.9723)	0.9597 (3.0274)	1.3441 (5.5852)	1.0828 (0.0035)	0.9403 (0.0016)	1.5344 (0.0031)	1.0749 (0.0039)	0.9315 (0.0014)	1.2730 (0.0024)
나.5	1.3431 (1.2209)	1.1826 (2.3010)	1.9639 (8.3232)	1.3355 (1.6981)	1.1738 (3.0011)	1.6835 (6.3621)	1.3269 (0.0034)	1.1560 (0.0030)	1.8130 (0.0025)	1.3132 (0.0031)	1.1396 (0.0023)	1.5828 (0.0022)
다.1	0.7692 (-1.3340)	0.9300 (-1.6081)	1.2415 (-5.7041)	0.7677 (-0.0260)	0.9274 (0.0324)	0.9695 (-3.3207)	0.7796 (0.0010)	0.9452 (0.0013)	1.3166 (0.0012)	0.7679 (0.0007)	0.9271 (0.0009)	1.0028 (0.0008)
다.2	0.7431 (-0.6684)	0.8524 (-0.7914)	1.1787 (-4.1785)	0.7402 (0.1488)	0.8477 (0.3433)	0.9175 (-2.2272)	0.7481 (0.0016)	0.8592 (0.0013)	1.2301 (0.0019)	0.7391 (0.0012)	0.8448 (0.0015)	0.9384 (0.0013)
다.3	0.6341 (-0.0158)	0.6889 (-0.1305)	1.0025 (-3.2709)	0.6318 (0.3973)	0.6849 (0.5579)	0.7715 (-1.4813)	0.6342 (0.0012)	0.6898 (0.0017)	1.0364 (0.0025)	0.6293 (0.0012)	0.6811 (0.0017)	0.7831 (0.0015)
다.4	0.4841 (0.1655)	0.4951 (0.0202)	0.7693 (-2.7188)	0.4829 (0.3533)	0.4930 (0.4278)	0.5821 (-1.1547)	0.4833 (0.0013)	0.4950 (0.0011)	0.7908 (0.0019)	0.4812 (0.0011)	0.4909 (0.0013)	0.5889 (0.0011)
다.5	0.3355 (0.4491)	0.3209 (0.2813)	0.5374 (-1.8985)	0.3351 (0.5401)	0.3202 (0.5969)	0.3989 (-0.6476)	0.3340 (0.0016)	0.3200 (0.0015)	0.5478 (0.0021)	0.3333 (0.0014)	0.3183 (0.0016)	0.4015 (0.0015)
라.1	0.3964 (1.7454)	0.5886 (2.5078)	0.8724 (4.5792)	0.3946 (2.1222)	0.5870 (2.7301)	0.6191 (3.3039)	0.3896 (0.0016)	0.5742 (0.0019)	0.8342 (0.0019)	0.3864 (0.0015)	0.5714 (0.0019)	0.5993 (0.0012)
라.2	0.6087 (0.1975)	0.8489 (1.3975)	1.2541 (5.1127)	0.6054 (0.8832)	0.8461 (1.6825)	0.9039 (2.7977)	0.6075 (0.0026)	0.8372 (0.0027)	1.1931 (0.0023)	0.6001 (0.0021)	0.8321 (0.0019)	0.8793 (0.0020)
라.3	1.1340 (-0.1497)	1.4945 (0.4031)	2.1562 (5.6857)	1.1291 (0.8035)	1.4904 (0.9483)	1.5949 (2.5791)	1.1357 (0.0021)	1.4885 (0.0049)	2.0402 (0.0027)	1.1201 (0.0017)	1.4764 (0.0039)	1.5548 (0.0024)
라.4	0.9383 (0.1815)	1.2707 (0.4983)	1.8655 (6.1572)	0.9330 (1.0944)	1.2663 (0.9326)	1.3582 (2.9407)	0.9366 (0.0030)	1.2644 (0.0034)	1.7573 (0.0046)	0.9229 (0.0028)	1.2546 (0.0036)	1.3194 (0.0024)
라.5	1.2046 (-0.1326)	1.6143 (-0.1793)	2.3599 (6.1918)	1.1978 (0.7401)	1.6087 (0.3744)	1.7263 (2.6399)	1.2062 (0.0049)	1.6172 (0.0044)	2.2223 (0.0053)	1.1890 (0.0046)	1.6027 (0.0034)	1.6819 (0.0037)



[그림 14] 평균 체류고객 수(예제 다 군)

위의 표와 그림에서 볼수 있듯이 본 연구에서 제안된 근사적 기법은 일반적으로 만족할 만한 수준의 정확한 해를 산출해 주며, 특히 수요가 수용될 확률의 정확도는 매우 높다(<표 4> 참고). 그러나 제품 3 수요의 평균 대기시간(\bar{W}^3)의 상대오차는 일반적으로 5%로, 예제 나 군에서 q^3 의 비율이 증가함에 따라 \bar{W}^3 의 정확도가 다른 성능평가척도에 비해 다소 떨어진다는 것을 알 수 있다(<표 5> 참고). 이것은 부품상태의 결합확률 분포를 추정할 때, 부품의 상태를 파악할 수 있는 최소 상태인 2차원 상태를 정의하여 부품의 결합확률 분포를 구하고, 대기시간을 나타내는 4차원 상태의 상태공간을 가정 (A.1)과 가정 (A.2)를 이용하여 축소했기 때문이다. 만약 각 제품별 고객의 수를 파악하기 위해 부품상태를 4차원으로 정의하여 제품별 고객의 수의 안정상태확률을 구하고, Little의 법칙을 사용하여 평균 대기시간을 계산했다면, 알고리즘의 정확성은 보다 증가했을 것이다. 그러나 극히 단순한 경우를 제외하고는, 마코프체인의 복잡성 때문에 실질적으로 이를 분석하는 것은 불가능하다. 지면 관계상 제시하지 못하였으나 이외의 모든 실험에서 \bar{W}^3 의 관측된 상대오차가 10% 미만이라는 점과 시스템 모형화의 초기단계에서 데이터 수집의 오차를 고려한다면, 본 연구에서 제안된 알고리즘은 유용한 수리적 기법으로 판단된다. 또한 <표 3>에서 수요가 즉시 만족될 확률이 0.4 미만일 경우에 알고리즘의 정확성은 감소한다. 이는 재고수

준이 낮을수록 부품들간의 상호종속성이 증가되는데 반하여, 본 연구에서는 각 부품 생산 시스템을 분석할 때, 다른 부품의 상태를 고려하지 못했기 때문에 발생하는 오차로 판단된다. 그러나 실제사회에서 수요가 즉시 만족될 확률은 일반적으로 0.4 이상이므로 이 오차는 크게 중요하지 않을 것으로 생각되며, 이 오차가 중요한 경우라도 제안된 알고리즘은 특별한 대안이 존재하지 않는 상황에서 시스템의 개선을 위한 유용한 수리적 기법이 될수 있을 것으로 생각된다. 따라서 오차의 범위가 지나치게 크지 않는 범위에서 제안된 알고리즘은 시물레이션을 대신할 수 있는 기법이 될 것으로 기대한다.

7. 결 론

본 연구에서는 두 종류의 부품을 생산하여 세 종류의 제품 수요를 충족시키는 주문-조립 시스템을 분석하기 위한 근사적 기법을 개발하였다. 기존 연구가 부품 생산방식을 지나치게 단순하게 가정하여 현실적인 면을 반영하지 못했다면, 본 연구에서는 부품이 여러 단계를 거쳐 생산되고, 부품의 생산방식은 유한한 버퍼를 가지는 기본재고정책에 의해 통제된다고 가정하여 보다 현실적인 면을 고려하였다. 또한 본 연구에서는 제조시스템의 생산 시간분포로 널리 사용되는 Coxian-2 분포를 부품 가공시간의 분포로 사용하여 기존 연구의 가정을 완화하였다. 그러나 이와 같이 완화된 가정과 주문-조립 시스템에 내재된 부품들 간의 종속관계는 시스템 분석에 어려움을 준다. 따라서 본 연구에서는 이를 해결하고자 각 부품 생산 시스템을 서비스율이 상태 의존적인 단일 서버로 근사화하여, 부품상태의 결합확률분포를 효율적으로 계산하였고, 근사화하는 과정에서 Marie의 근사법과 행렬기하법을 이용하였다. 또한 본 연구에서는 부품상태의 결합확률분포가 구해진 후, 마코프체인의 복잡성을 감소시켜 제품별 수요의 대기시간을 효율적으로 계산할 수 있는 근사법을 제안하였다. 본 연구에서 제안된 근사법은 부품 상태의 결합확률 분포, 제품

별 수요의 평균 대기시간, 재고창고의 평균 재고수준 등 시스템의 중요한 성능척도를 비교적 정확하고 신속하게 계산하여 준다. 따라서 본 연구에서 제안된 근사법은 향후 주문-조립 시스템의 경제적인 설계 및 효율적인 운영에 유용하게 이용되어 질 수 있을 것으로 기대된다.

참고 문헌

- [1] 이호우, 「대기행렬이론」, 시그마프레스, 개정판, 1998.
- [2] Baynat, B. and Y. Dallery, "a Unified View of Product-Form Approximation Techniques for General Closed Queueing Networks," *Performance Evaluation*, Vol.18(1993), pp. 205-224.
- [3] Bruell, S.C. and G. Balbo, *Computational Algorithms for Closed Queueing Networks*, North-Holland, Oxford, 1980.
- [4] Buzacott, J.A. and D. Kostelski, "Matrix-geometric and Recursive Algorithm Solution of a two-stage Unreliable Flow Line," *IIE Trans.*, Vol.19(1987), pp.429-438.
- [5] Cheung, K.L. and W.H.Hausman, "Multiple Failures in a Multi-Item Spares Inventory Model," *IIE Transactions*, Vol.27(1995), pp. 171-180.
- [6] Dallery, Y., "Approximate Analysis of General Open Queueing Networks with Restricted Capacity," *Performance Evaluation*, Vol.11(1990), pp.209-222.
- [7] Gattorna, J., *Strategic Supply Chain Alignment*, Gower, Hampshire, England, 1998.
- [8] Glasserman, P. and Y. Wang, "Leadtime-Inventory Trade-Offs in Assemble to Order Systems," *Operations Research*, Vol.46, No.6 (1998), pp.858-871.
- [9] Hausman, W.H., "On Optimal Repair Kits under a Job Completion Criterion," *Management Science*, Vol.28, No.11(1982), pp.1350-1351.
- [10] Hopp, W.J. and M.K., Spearman, *Factory Physics*, Irwin/McGraw-Hill, Boston, 2000.
- [11] Kragewsky, L.J., B.E. King, L.P. Ritzman and D.S. Wong, "Kanban, MRP, and Shaping the Manufacturing Environment," *Management Science*, Vol.33, No.1(1987), pp.39-57.
- [12] Lee, H.L. and C. Billington, "Material Management in Decentralized Supply Chains," *Operations Research*, Vol.41, No.5(1993), pp.835-847.
- [13] Mamer, J.W. and S.A. Smith, "Optimizing Field Repair Kits Based on Job Completion Rate," *Management Science*, Vol.28, No.11, pp.1328-1333.
- [14] Marie, R.A. "an Approximate Analytical Method for General Queueing Networks," *IEEE Trans. Soft. Eng.*, Vol.SE-5, No.5 (1979), pp.530-538.
- [15] Neuts, M.F., *Matrix Geometric Solutions in Stochastic Models-an Algorithmic Approach*, John Hopkins University Process, Baltimore, 1981.
- [16] Smith, S.A., J.C. Chambers and E. Shlifer, "Optimal Inventories Based on Job Completion Rate for Repairs Requiring Multiple Items," *Management Science*, Vol.26, No.8 (1980), pp.849-852.
- [17] Song, J.S., "On the Order Fill Rate in a Multi-item, Base-Stock Inventory System," *Operations Research*, Vol.46, No.6(1998), pp.831-845.
- [18] Song, J.S., S.H. Xu and B. Lin, "Order-Fulfillment Performance Measures in an Assemble-to-Order System with Stochastic Leadtimes," *Operations Research*, Vol. 47, No.1(1999), pp.131-149.