

CONWIP 통제방식에 의해 운영되는 다품목 생산시스템의 성능평가

박찬우¹ · 이효성[†] · 김창곤²

¹경희대학교 기계 · 산업시스템공학부/²한국해양수산 개발원

Performance Evaluation of a Multi-Item Production System Operated by the CONWIP Control Mechanism

Ghan-Woo Park¹ · Hyo-Seong Lee² · Chang-Gon Kim²

¹School of Mechanical and Industrial Systems Engineering, Kyung Hee University, Yongin, 449-701

²Port Information and Information System, Korea Maritime Institute, Seoul, 138-730

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. The production of each component is operated by the CONWIP control mechanism. To analyse this system, we propose an approximation method based on aggregation method. In application of the aggregation method, a product-form approximation technique as well as a matrix-geometric method is used. Comparisons with simulation show that the approximation method provides fairly good results.

Keywords: multi-component production/inventory, CONWIP control mechanism, queueing network, product-form approximation method, performance evaluation

1. 서론

오늘날 기업 환경의 급격한 변화는 고객의 다양한 요구에 기업이 신속하게 반응할 것을 요구하고 있으며, 이러한 요구를 재고정책에 반영할 수 있는 연구가 최근 많은 관심을 끌고 있다. 예를 들어, 고객의 주문은 여러 제품으로 구성되며, 기업은 이에 따라 다양한 제품을 생산하고 재고를 유지한다. 이러한 경우, 시스템의 성능평가척도는 고객의 주문에 대한 납품충족률(order fulfillment rate)이 되며, 이는 주문을 구성하는 제품별 성능평가척도보다 관리자에게 중요한 의사결정의 척도를 제공한다. 그러나 지금까지 많은 재고모형에서 제품들 간의 상호관계가 고려되지 못했고, 한 제품의 수요는 다른 제품의 수

요와 독립이라 가정되어, 제품별 재고정책에 관한 연구가 진행되었다. 따라서 제품들 간의 상호 관계를 고려할 수 있는 새로운 재고모형의 연구가 요구된다(Lee and Billington, 1993; Song, 1998).

다양한 제품의 수요에 대한 재고정책연구는 Smith *et al.* (1980), Hausman(1982), Mamer and Smith(1982) 등의 수리용구 문제(repair kit problem)를 필두로 하여 1990년 이후 Glassman and Wang(1988), Cheung and Hausman(1995), Song(1998, 1999) 등의 학자들에 의해 수행되었다. 많은 연구에서 재고정책으로 기준재고정책(base stock policy)이 사용되었고, 1단계(single echelon model) 생산시스템의 모형을 가정하였다. Song(1998)은 완제품에 대한 수요는 복합 포아송과정(compound Poisson Process)을 따르고, 주문인도시간은 일정(constant replenish-

이 논문은 2000년도 한국과학재단의 연구비에 의하여 연구되었음(2000-1-31500-001-3).

† 연락저자: 이효성 교수, 449-701 경기도 용인시 기흥읍 서천리1 경희대학교 기계 · 산업시스템공학부, Fax : 031-201-4004, e-mail : hslee@khu.ac.kr.

2001년 10월 접수, 2001년 11월 게재 확정.

ment lead-time)하다는 가정하에서 1단계 생산시스템의 최종 제품에 대한 납품충족률을 정확히 계산하는 방법을 개발하였다. 또한 Glasserman and Wang(1988)은 완제품에 대한 수요가 포아송과정을 따르고 생산시간은 지수분포를 따르는 공습사슬모형에 대해서 재고수준과 최종제품 인도기간 사이에 선형 관계가 있음을 보였다. 최근 연구에서 Song(1999)은 확률적인 인도기간을 가진 주문-조립 시스템(assembly-to-order system)에서 최종제품의 기간 내 납품충족률을 행렬기하법(matrix geometric method)을 이용하여 계산하는 방법을 제안하였다.

이상에서 살펴본 바와 같이 다양한 제품 수요에 대한 재고 정책연구가 최근 활발히 수행되고 있다. 그러나 기존 연구들은 제품의 생산/조달 방식을 지나치게 단순하게 가정하여 실제 상황을 정확하게 반영하지 못하는 것이 단점으로 지적된다. 그러므로 실제상황이 보다 잘 고려된 다양한 제품의 수요에 대한 재고정책연구가 요구된다. 본 연구에서는 기존연구의 확장으로서 각 제품이 여러 단계를 거쳐 생산되고, 각 제품의 생산은 CONWIP 제어방식(CONstant Work In Process control mechanism)에 의해 이루어지는 다품목 생산 시스템을 분석하고자 한다.

CONWIP 제어방식은 견인 제어방식(pull control mechanism)과 밀어내기 제어방식(push control mechanism)의 혼합(hybrid) 시스템으로 Spearman *et al.*(1990)에 의해 제안되었다. 이 연구에서 CONWIP 제어방식은 재공품 수를 일정하게 유지하면서, 고객만족도 측면의 성능이 우수한 것으로 증명되었다. 또한 CONWIP 제어방식은 통제방식이 용이하며, 각 공정에서 생산 준비 시간이 비교적 큰 경우에도 적용될 수 있는 부수적인 장점이 있는 것으로 보고되고 있다.

Duenyas *et al.*(1993)은 각 공정에서의 가공시간이 일정(deterministic)하고 기계고장이 발생할 수 있는 직렬형 CONWIP 시스템을 분석하였다. 이 연구에서는 기계의 고장 발생간격과 수리시간이 모두 지수분포를 따르고, 완제품에 대한 수요는 항상 존재(saturated demands)한다는 가정하에서 시스템의 생산률을 추정하기 위한 근사적 방법이 제시되었다. Duenyas and Hopp은 이 연구를 확장하여 CONWIP 조립시스템에 대한 분석을 시도하였다. 이들은 기계의 고장은 발생하지 않는다는 가정하에서 가공시간이 일정한 경우(Duenyas and Hopp, 1992)와 지수분포를 따르는 경우(Duenyas and Hopp, 1993)에 대해 시스템의 생산률을 구하기 위한 근사적 방법을 개발하였다. 완제품에 대한 수요가 포아송과정에 따라 발생하는 시스템의 연구는 Mascolo *et al.*(1996)에 의해 수행되었다. 이 연구에서는 각 작업장에서의 가공시간이 Coxian-2 분포를 따른다는 가정하에서 시스템의 안정상태 확률을 구하기 위한 근사적 기법이 제시되었다. 최근에 Lee and Lee(1998)는 완제품에 대한 수요가 복합 포아송과정에 따라 발생하는 직렬형 CONWIP 시스템의 성능평가를 수행하기 위한 근사적 기법을 제안하였고, 후속 연구로 Lee and Lee(1998)는 각 공정에서의 가공이 로트(lot) 단위로 이루어지는 경우의 직렬형 CONWIP 시스템에 대한 근사적 분석을 시도하였다.

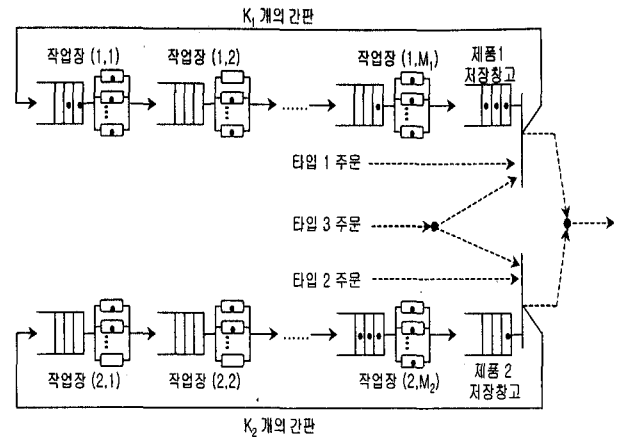


그림 1. CONWIP 제어방식으로 구성된 다품목 생산시스템.

그러나 이들 기존의 연구에서 고객의 수요는 단일 제품에 제한되어 있거나, 수요는 항상 존재한다고 가정되었다. 따라서 본 연구에서는 CONWIP 제어방식의 기존 연구를 다 제품 수요를 갖는 재고모형으로 확장하고자 하며, 제안된 시스템의 중요한 성능평가척도를 신속, 정확하게 구할 수 있는 근사적 알고리즘을 제시하고자 한다.

2. 연구내용, 범위 및 방법

2.1 연구내용 및 범위

본 연구에서는 두 종류의 제품이 생산되며, 각각의 제품 혹은 제품의 조합을 요구하는 세 종류의 수요가 존재하는 생산 시스템을 분석한다(<그림 1> 참고). 타입(type) 1과 타입 2 수요는 각각 제품 1과 제품 2를 요구하는 수요이며, 타입 3 수요는 제품 1과 제품 2 모두를 요구하는 수요라 가정한다. 고객의 수요는 도착률 λ 의 포아송과정에 따라 도착하고, 도착한 고객의 수요가 타입 c 제품을 요구하는 확률은 q^c 이며, 각각의 수요의 도착과정은 다른 수요의 도착과정과 독립이라 가정한다 ($c=1, 2, 3$). 타입 1 수요와 타입 2 수요가 도착했을 때 요구되는 제품이 존재하지 않으면 수요는 유실되는 것으로 가정한다. 만약 타입 3 수요가 도착했을 때, 제품 1과 제품 2가 모두 존재하면, 타입 3 수요는 즉시 만족될 수 있다. 그러나 제품 1과 2 중 하나라도 존재하지 않을 경우, 본 연구에서는 다음의 두 가지 유실(lost sales) 모형을 고려한다.

- i) TOS(Total Order Service) 모형: 제품 1과 제품 2가 타 제품의 부품으로 사용되는 제조 시스템에 적합한 모형으로, 제품 1과 제품 2 중 하나라도 재고가 존재하지 않을 경우, 도착한 수요 전체가 유실된다.
- ii) POS(Partial Order Service) 모형: 유통공정에 적합한 모형으로, 타입 3 수요가 도착했을 때 필요한 제품이 하나라도 존재한다면, 재고가 있는 제품에 한해서 부분적

서비스가 가능하다고 가정한다.

제품 i 의 생산공정은 CONWIP 제어방식으로 통제되는 흐름식 생산라인 시스템(flow line system)이다($i=1, 2$). 작업장 $(i, 1)$ 에서 최초의 가공을 받은 제품은 후속 가공을 거친 후 작업장 (i, M_i) 에서 최후 가공을 받게 되며, 작업장 (i, M_i) 에서 가공을 마친 제품은 출하되어 제품 i 의 저장창고에 저장된다. 제품 i 의 저장창고를 작업장 J_i 라하고, 제품 i 를 생산하는 M_i 개의 작업장을 모두 합하여 제품 i 생산시스템이라 정의하자. 제품 i 생산시스템에서는 총 K_i 개의 간판(kanban)이 사용되며, 생산중에 있는 모든 재공품에는 간판이 부착되고, 최후 가공을 받은 제품의 경우도 간판이 부착된채 저장된다. 수요가 발생하여 제품 i 가 작업장 J_i 에서 인출되면, 제품에 부착되어 있던 간판은 즉시 제품 i 에서 떼어져 작업장 $(i, 1)$ 로 이동해 작업지시서(production authorization card)의 역할을 한다. 간판이 작업장 $(i, 1)$ 에 도착되면 작업장 $(i, 1)$ 에서는 원자재 제품의 가공이 시작되고 가공이 끝난 제품은 밀어내기 방식에 따라 후속 작업장으로 이동된다. 작업장 $(i, 1)$ 에는 충분한 수의 원자재가 항상 존재한다고 가정하며, 이외의 중요 가정은 다음과 같다.

- 1) 작업장 (i, j) 에는 동일한 성능의 $SE_{i,j}$ 대의 기계가 존재하며, 제품의 가공시간은 <그림 2>와 같은 Coxian-2 분포를 따른다고 가정한다($i=1, 2, j=1, 2, \dots, M_i$).

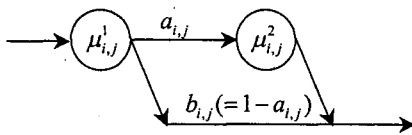


그림 2. 가공시간의 Coxian-2 분포.

- 2) 각 작업장에서의 기계고장은 발생하지 않는다.
- 3) 각 작업장 및 완제품 저장창고의 크기는 간판의 수보다 커 붐배가 발생하지 않는다.
- 4) 제품의 작업장 간 이동시간은 무시할 수 있을 정도로 짧다.

본 연구모형에서 각 타입의 수요는 독립적인 포아송과정을 형성한다. 그러나 하나의 수요가 동시에 여러 제품들을 요구할 수 있으므로 각 제품의 상태가 상호 종속적인 관계를 갖게 되어 시스템 분석에 어려움을 준다. 또한 저장창고에서 간판의 이동이 이루어지기 위해서는 제품이 존재해야 하고 수요가 발생해야 하므로 저장창고는 동기화 스테이션(synchronization station)임을 알 수 있다. 따라서 본 연구의 모형은 승법형 네트워크(product-form network)의 요건을 만족하지 못하며, 시스템의 분석을 위해서는 근사적 기법에 의존해야 한다.

2.2 연구방법

앞 절에서 설명한 바와 같이 본 연구모형에서 각각의 수요는 포아송과정을 따르지만, 각 제품의 상태는 상호 종속적인

관계를 갖게 된다. 따라서 시스템을 분석하기 위해서는 제품의 상태를 독립적으로 분석하면 안 되고, 각 제품의 상태, 그리고 각 제품의 상태가 주어졌을 경우의 재고보충률을 모두 고려하여 분석해야만 한다. 만일 각 제품 생산시스템의 상태가 주어졌을 경우에 재고보충률을 정확히 추정할 수 있다면, 각각의 제품생산시스템을 서비스율이 상태 의존적인 단일서버(server)로 치환함으로써 본 연구모형을 단순화할 수 있으며, 이 서버의 서비스 시간이 지수분포를 따른다고 가정하여 원네트워크를 효율적 분석이 가능한 네트워크로 근사화할 수 있다. 이와 같은 방법은 승법형 근사법(product-form approximation techniques)에서 대기네트워크를 몇 개의 하부네트워크로 분해(decomposition)한 후, 각각의 하부네트워크를 서비스 시간이 지수분포를 따르고 상태의존 서비스율을 갖는 단일서버로 치환함으로써, 분석하고자 하는 네트워크를 승법형 대기네트워크로 근사화하는 것과 유사하며, 승법형 근사법의 정확성은 이미 많은 연구에서 입증된 바 있다(Baynat and Dallery, 1993). 다만, 본 연구에서는 원네트워크를 승법형 네트워크가 아닌 효율적 분석이 가능한 네트워크로 근사화한다는 것이 승법형 근사법과의 차이점이라 말할 수 있다. 따라서 본 연구에서는 제품 i 생산 시스템을 서비스 시간이 지수분포를 따르고, n_i 개의 간판이 존재할 때의 서비스율이 $\mu_i(n_i)$ 인 서버 S^i 로 치환하여 표시한다($i=1, 2$). 그러면 원네트워크는 <그림 3>과 같은 네트워크로 근사화될 수 있으며, 앞으로 이와 같은 네트워크를 네트워크-AO로 표기하기로 한다.

그러나 네트워크-AO를 분석하기 위해서는 상태종속 서비스율 $\mu_i(n_i)$ 을 추정하는 효과적인 방법을 필요로 하며, 본 연구에서는 통합법(aggregation method)을 사용하고자 한다(Harrison and Patel, 1992; Baynat and Dallery, 1993). 통합법은 비승법형 폐쇄형 대기네트워크(non-product form closed queueing network)를 분석하기 위한 근사법으로, 대기네트워크를 몇 개의 하부네트워크로 분해한 후 각각의 하부네트워크를 서비스 시간이 지수분포를 따르고 상태의존 서비스율을 갖는 서버로

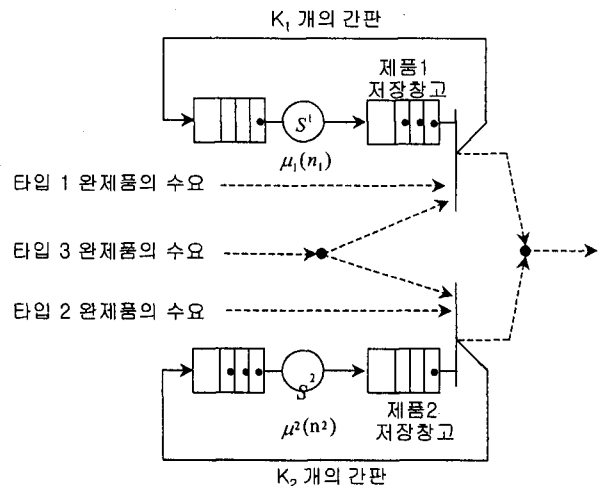


그림 3. 네트워크-AO.

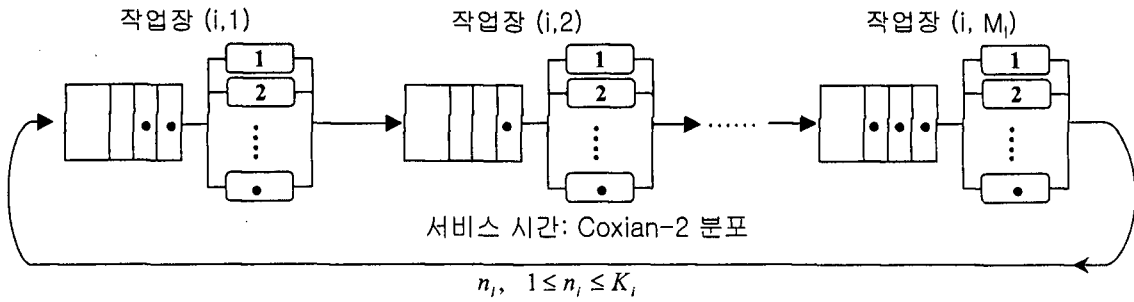


그림 4. 하부네트워크 $R^i(n_i)$, $1 \leq n_i \leq K_i$, $i = 1, 2$.

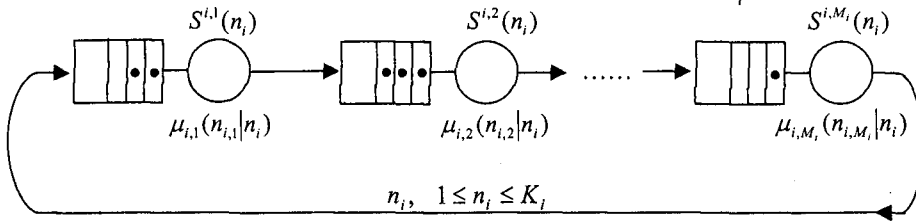


그림 5. 하부네트워크 $R^i(n_i)$ 의 동등 승법형 대기네트워크.

근사화하는 기법이다. 서버의 서비스율을 구하기 위하여 통합법은 각각의 하부네트워크를 떠난 고객은 즉시 해당 하부네트워크로 복귀한다고 가정한다. 따라서 각각의 하부네트워크는 하나의 독립된 폐쇄형 대기네트워크(closed queueing network)로 모델링되며, n 명의 고객이 존재할 때의 하부네트워크의 상태종속 서비스율을 구하기 위하여, 총고객의 수가 n 명인 폐쇄형 네트워크를 분석하여 산출률(throughput)을 구한다. 총고객의 수가 n 명인 폐쇄형 네트워크의 산출률이 구해지면, 이는 곧 n 명의 고객이 존재할 때의 하부네트워크의 상태종속 서비스율로 사용된다. 따라서 원래 총고객수를 N 이라 가정하면, 각 하부네트워크를 나타내는 폐쇄형 네트워크는 $1 \leq n \leq N$ 에 대하여, n 명의 고객을 가지는 폐쇄형 대기네트워크로 분석된다. 이상의 절차를 본 연구에 적용하면, 제품 i 생산시스템은 <그림 4>에서 주어진 폐쇄형 대기네트워크로 표현되며, 이와 같이 n_i 명의 고객을 가지는 폐쇄형 대기네트워크를 앞으로 하부네트워크 $R^i(n_i)$ 로 표기로 한다.

그러나 각 작업장에서 서비스 시간이 Coxian-2 분포를 따르므로 하부네트워크 $R^i(n_i)$ 는 비승법형 네트워크이다. 따라서 하부네트워크 $R^i(n_i)$ 를 분석하기 위해서는 별도의 근사법을 필요로 하며, 본 연구에서는 Marie의 근사법(1979)을 사용하였다. Marie의 근사법은 서비스 시간이 일반분포를 따르는 폐쇄형 대기네트워크를 분석하기 위한 근사법으로, 통합법과 마찬가지로 대기네트워크를 몇 개의 하부네트워크로 분해한 후, 각각의 하부네트워크를 서비스 시간이 지수분포를 따르고 서비스율이 상태 의존적인 단일서버로 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 단일서버의 상태종속 서비스율을 구하기 위하여 각각의 하부네트워크를 상태종속 포아송 도착과정을 갖는 개방형 대기시스템(open queueing system)으로 가정해 분석한다.

본 연구모형에서 하부네트워크 $R^i(n_i)$ 의 작업장 (i, j) 을 하부네트워크 $R^{i,j}(n_i)$ 라 했을 때, 하부네트워크 $R^i(n_i)$ 는 총 M_i 개의 하부네트워크로 분해된다. 하부네트워크 $R^{i,j}(n_i)$ 에 대응하는 동등 승법형 대기네트워크의 서버를 $S^{i,j}(n_i)$ 라 정의하고, 서버 $S^{i,j}(n_i)$ 에 $n_{i,j}$ 개의 간판이 있을 때의 상태종속 서비스율을 $\mu_{i,j}(n_{i,j} | n_i)$ 라 했을 때, 하부네트워크 $R^i(n_i)$ 의 동등 승법형 대기네트워크는 <그림 5>와 같다.

서버 $S^{i,j}(n_i)$ 의 상태종속 서비스율 $\mu_{i,j}(n_{i,j} | n_i)$ 는 하부네트워크 $R^{i,j}(n_i)$ 를 분석함으로써 얻어진다. 하부네트워크 $R^{i,j}(n_i)$ 는 $n_{i,j}$ 개의 간판이 있을 때 상태종속 도착률이 $\lambda_{i,j}(n_{i,j} | n_i)$ 이고, 도착간격이 지수분포를 따르는 개방형 대기시스템으로 간주해 분석한다. 우선, 하부네트워크 $R^{i,j}(n_i)$ 의 상태종속 도착률 $\lambda_{i,j}(n_{i,j} | n_i)$ 를 알고 있다고 가정하자. 그러면 하부네트워크 $R^{i,j}(n_i)$ 는 <그림 6>과 같은 개방형 대기시스템으로 표현될 수 있다.

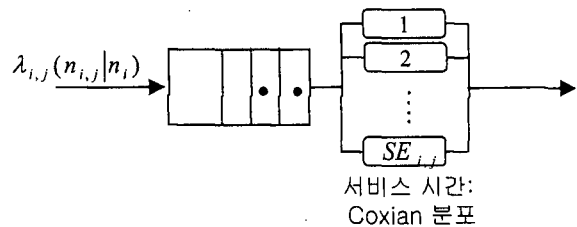


그림 6. 하부네트워크 $R^{i,j}(n_i)$.

하부네트워크 $R^{i,j}(n_i)$ 의 안정상태확률이 구해지면, 하부네트워크의 상태종속 산출률(state dependent throughput)을 계산할 수 있고, 이 값은 곧 동등 승법형 대기네트워크의 상태종속 서비스율 $\mu_{i,j}(n_{i,j} | n_i)$ 로 사용된다. 또한 동등 승법형 대기네트워크의 상태종속 서비스율이 구해지면, 동등 승법형 대기네트워크의 분석을 통해 하부네트워크의 상태종속 도착률 $\lambda_{i,j}(n_{i,j} | n_i)$ 를 계산할 수 있다. 승법형 근사법에서는 이와 같은 과정을 $\mu_{i,j}(n_{i,j} | n_i)$ 가 수렴할 때까지 반복 수행한다. 상태종속 도착률 $\lambda_{i,j}(n_{i,j} | n_i)$ 가 주어졌을 때, 하부네트워크 $R^{i,j}(n_i)$ 를 분석하는 방법과 하부네트워크 $R^{i,j}(n_i)$ 을 분석하는 방법이 다음 절에서 상세히 소개된다.

3. 제품 i 생산 시스템의 분석, $i = 1, 2$

3.1 하부네트워크 $R^{i,j}(n_i)$ 의 분석, $j = 1, \dots, M_i$

본 절에서는 하부네트워크 $R^{i,j}(n_i)$ 의 분석방법을 살펴보고자 한다. 상태종속 도착률 $\lambda_{i,j}(n_{i,j} | n_i)$ 를 알고 있다고 가정하면, 가공시간이 Coxian-2 분포를 따르므로 하부네트워크 $R^{i,j}(n_i)$ 는 상태의 수가 유한한 연속시간 마코프체인(continuous time Markov Chain)으로 모델링될 수 있다. 시스템의 상태를 정의하기 위하여 m_1 을 하부네트워크 $R^{i,j}(n_i)$ 에 존재하는 간판 수, m_2 를 Coxian-2 분포의 1단계에서 서비스 중인 서버 수라 하면, 시스템의 상태는 (m_1, m_2) 로 정의될 수 있다. 따라서 하부네트워크 $R^{i,j}(n_i)$ 의 상태공간 $ST_{i,j}(n_i)$ 는 다음과 같이 표시된다.

$$ST_{i,j}(n_i) = \{(m_1, m_2) | 0 \leq m_1 \leq n_i, 0 \leq m_2 \leq Mn_0(m_1)\},$$

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

$ST_{i,j}(n_i)$ 의 모든 상태들을 m_1 을 기준으로 사전식 배치(lexicographic order)를 하면, 전이율 행렬(transition rate matrix)은 다음과 같이 3종류의 소행렬로 블록(block)화 될 수 있다.

$$Q_{i,j}(n_i) = \begin{pmatrix} W(0) & Z(0) & 0 & 0 & \dots & 0 & 0 \\ H(1) & W(1) & Z(1) & 0 & \dots & 0 & 0 \\ 0 & H(2) & W(2) & Z(2) & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \dots & H(n_i-1) & W(n_i-1) & Z(n_i-1) \\ 0 & 0 & \dots & \dots & 0 & H(n_i) & W(n_i) \end{pmatrix} \quad (1)$$

식 (1)에서 각각의 소행렬은 다음과 같이 표시되며, 각각의 소행렬의 예는 부록을 참고하기 바란다.

$$W(m_1) = \{w_{k,l}(m_1)\}, 0 \leq m_1 \leq n_i, 1 \leq k \leq Mn_1(m_1), 1 \leq l \leq Mn_1(m_1), \text{ where } Mn_1(m_1) = \text{Min}\{SE_{i,j}, m_1\} + 1. \quad (2)$$

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

$$Z(m_1) = \{z_{k,l}(m_1)\}, 0 \leq m_1 \leq n_i - 1, 1 \leq k \leq Mn_1(m_1), 1 \leq l \leq Mn_1(m_1 + 1), \quad (3)$$

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

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

$$H(m_1) = \{h_{k,l}(m_1)\}, 1 \leq m_1 \leq n_i, 1 \leq k \leq Mn_1(m_1), 1 \leq l \leq Mn_1(m_1 - 1), \quad (4)$$

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

식 (1)의 전이율 행렬은 삼각대각(tridiagonal) 형태를 취하고 있으므로 행렬기하법(matrix geometric method)을 적용하여 분석할 수 있다(Neuts, 1981). 상태 (m_1, m_2) 의 안정상태확률을 $\hat{P}_{i,j}(m_1, m_2 | n_i)$ 라 했을 때, 소행렬로 표현된 균형방정식은 다음과 같다.

$$\begin{aligned} \hat{P}_{i,j}(0 | n_i) \cdot W(0) + \hat{P}_{i,j}(1 | n_i) \cdot H(1) &= 0, \\ \hat{P}_{i,j}(m_1 - 1 | n_i) \cdot Z(m_1 - 1) + \hat{P}_{i,j}(m_1 | n_i) \cdot W(m_1) \\ &+ \hat{P}_{i,j}(m_1 + 1 | n_i) \cdot H(m_1 + 1) = 0, m_1 = 1, 2, \dots, n_i - 1, \\ \hat{P}_{i,j}(n_i - 1 | n_i) \cdot Z(n_i - 1) + \hat{P}_{i,j}(n_i | n_i) \cdot W(n_i) &= 0. \end{aligned} \quad (5)$$

$$\text{여기서, } \hat{P}_{i,j}(m_1 | n_i) = \{\hat{P}_{i,j}(m_1, 0 | n_i), \hat{P}_{i,j}(m_1, 1 | n_i), \dots, \hat{P}_{i,j}(m_1, Mn_0(m_1) | n_i)\}$$

식 (5)의 시스템 균형방정식은 반복적인 구조를 가지므로 $\hat{P}_{i,j}(m_1 | n_i)$ 는 다음과 같은 순환식으로 표현될 수 있다.

$$\hat{P}_{i,j}(m_1 | n_i) = \hat{P}_{i,j}(m_1 - 1 | n_i) \cdot R(m_1), m_1 = 1, \dots, n_i. \quad (6)$$

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

$$R(m_1) = \begin{cases} -Z(m_1 - 1) \cdot [W(m_1) + R(m_1 + 1) \cdot H(m_1 + 1)]^{-1} & \text{if } m_1 = 1, 2, \dots, n_i - 1, \\ -Z(m_1 - 1) \cdot W(m_1)^{-1}, & \text{if } m_1 = n_i. \end{cases} \quad (7)$$

식 (6), (7)에 의해서, 모든 확률의 합을 1로 하는 정규화 방정식은 다음과 같이 표시된다.

$$\sum_{m_1=0}^{n_i} \hat{P}_{i,j}(m_1 | n_i) \cdot e(m_1) = \hat{P}_{i,j}(0 | n_i) \sum_{m_1=0}^{n_i} R(m_1 | n_i) \cdot e(m_1) = 1 \quad (8)$$

여기서, $\mathbf{e}(m_i)$ 는 모든 원소가 1인 크기 $Mn_i(m_i)$ 의 열벡터이며, $\mathbf{RP}(m_i)$ 은 크기 $Mn_i(m_i)$ 의 행벡터로 다음과 같다.

$$\mathbf{RP}(m_i) = \begin{cases} 1, & \text{if } m_i = 0, \\ \prod_{k=1}^{m_i} \mathbf{R}(k), & \text{if } m_i = 1, \dots, n_i. \end{cases} \quad (9)$$

따라서 $\hat{P}_{i,j}(0|n_i)$ 은 식 (8)에 의해서 다음과 같이 구할 수 있다.

$$\hat{P}_{i,j}(0|n_i) = \hat{P}_{i,j}(0,0|n_i) = \frac{1}{\sum_{m_i=0}^{n_i} \mathbf{RP}(m_i|n_i) \cdot \mathbf{e}(m_i)} \quad (10)$$

$\hat{P}_{i,j}(0|n_i)$ 가 구해지면, 식 (6)으로부터 다른 상태의 확률을 순차적으로 구할 수 있으며, 이로부터 하부네트워크 $R^{i,j}(n_i)$ 에 m_i 개의 간판이 있을 확률 $P_{i,j}(m_i|n_i)$ 는 다음과 같이 표시된다.

$$P_{i,j}(m_i|n_i) = \sum_{m_2=0}^{Mn_i(m_i)} \hat{P}_{i,j}(m_1, m_2|n_i), \quad m_1 = 0, 1, \dots, n_i. \quad (11)$$

3.2 하부네트워크 $R^i(n_i)$ 의 분석, $i=1,2$

하부네트워크 $R^{i,j}(n_i)$ 의 안정상태확률 $P_{i,j}(n_{i,j}|n_i)$ 가 구해지면, 하부네트워크 $R^{i,j}(n_i)$ 에 존재하는 간판의 수가 $n_{i,j}$ 일 경우의 상태종속 산출률 $\nu_{i,j}(n_{i,j}|n_i)$ 의 값은 다음과 같이 얻을 수 있다.

$$\nu_{i,j}(n_{i,j}|n_i) = \begin{cases} 0, & \text{if } n_{i,j} = 0, \\ \lambda_{i,j}(n_{i,j}-1|n_i) \frac{P_{i,j}(n_{i,j}-1|n_i)}{P_{i,j}(n_{i,j}|n_i)}, & \text{if } n_{i,j} = 1, 2, \dots, n_i \end{cases} \quad (12)$$

하부네트워크 $R^{i,j}(n_i)$ 의 상태종속 산출률 $\nu_{i,j}(n_{i,j}|n_i)$ 의 값이 구해지면, 동등 승법형 대기네트워크를 구성하는 서버 $S^{i,j}(n_i)$ 의 상태종속 서비스율 $\mu_{i,j}(n_{i,j}|n_i)$ 는 다음과 같이 구해진다.

$$\mu_{i,j}(n_{i,j}|n_i) = \nu_{i,j}(n_{i,j}|n_i), \quad n_{i,j} = 0, \dots, n_i, \quad j = 1, \dots, M_i \quad (13)$$

따라서 하부네트워크 $R^{i,j}(n_i)$ 의 상태종속 도착률 $\lambda_{i,j}(n_{i,j}|n_i)$ 를 알고 있다면, 하부네트워크 $R^{i,j}(n_i)$ 의 안정상태확률을 구할 수 있고, 식 (12)와 식 (13)을 이용하여 동등 승법형 대기네트워크의 상태종속 서비스율 $\mu_{i,j}(n_{i,j}|n_i)$ 를 구할 수 있다. 그러나 상태종속 도착률 $\lambda_{i,j}(n_{i,j}|n_i)$ 의 값은 우리가 구해야만 하는 미지의 값이다. 상태종속 도착률 $\lambda_{i,j}(n_{i,j}|n_i)$ 를 구하기 위하여, 동등 승법형 대기네트워크의 서비스율 $\mu_{i,j}(n_{i,j}|n_i)$ 를 알고 있다고 가정하자. 그러면 임의의 알고리즘을 이용하여 서버 $S^{i,j}(n_i)$ 의 상태종속 산출률 $X_{i,j}(n_{i,j}|n_i)$ 를 구할 수 있다. 만일 중첩(convolution) 알고리즘을 이용한다

면, 서버 $S^{i,j}(n_i)$ 의 상태종속 산출률은 다음 식에 의해 구할 수 있다.

$$X_{i,j}(n_{i,j}|n_i) = \begin{cases} e_{i,j} \frac{G_{i,j|M_i}(n_i - n_{i,j} - 1)}{G_{i,j|M_i}(n_i - n_{i,j})}, & n_{i,j} = 0, 1, \dots, n_i - 1, \\ 0, & n_{i,j} = n_i. \end{cases} \quad (14)$$

여기서, $e_{i,j}$ 는 서버 $S^{i,j}(n_i)$ 의 상대적 방문율이고, $G_{i,j|M_i}(n)$ 은 M_i 개의 서버 중 서버 $S^{i,j}(n_i)$ 를 제외하고 계산한 정규화 상수이다.

동등 승법형 대기네트워크를 구성하는 서버 $S^{i,j}(n_i)$ 의 상태종속 산출률이 구해지면, 하부네트워크 $R^{i,j}(n_i)$ 의 상태종속 도착률은 다음과 같이 구해진다.

$$\lambda_{i,j}(n_{i,j}|n_i) = X_{i,j}(n_{i,j}|n_i), \quad n_{i,j} = 0, 1, \dots, n_i, \quad j = 1, \dots, M_i \quad (15)$$

위와 같은 방법으로 하부네트워크의 안정상태확률이 구해지면, 하부네트워크의 상태종속 산출률을 계산할 수 있고, 이 값은 곧 동등 승법형 대기네트워크의 상태종속 서비스율로 사용된다. 또한 동등 승법형 대기네트워크 상태종속 서비스율이 구해지면, 동등 대기네트워크의 상태종속 산출률이 계산될 수 있고, 이 값은 곧 하부네트워크의 상태종속 도착률로 사용될 수 있다. 승법형 근사법에서는 이와 같은 과정을 모든 하부네트워크의 상태종속 서비스율이 수렴할 때까지 반복 수행되며, 이상의 과정을 정리하면 알고리즘 1과 같다.

알고리즘 1

Step 0. 초기화과정

: $\mu_{i,j}(n_{i,j}|n_i)$ 을 초기화

Step 1. 반복과정

Step 1.1. 동등 승법형 대기네트워크의 분석.

: 식 (14)와 식 (15)를 이용하여 도착률 $\lambda_{i,j}(n_{i,j}|n_i)$ 의 계산.

Step 1.2. 하부네트워크 $R^{i,j}(n_i)$ 의 분석.

: $P_{i,j}(n_{i,j}|n_i)$ 를 계산하고 식 (12)와 식 (13)을 이용하여 $\mu_{i,j}(n_{i,j}|n_i)$ 의 계산.

Step 1.3. $\mu_{i,j}(n_{i,j}|n_i)$ 의 수렴성 검사.

: $\mu_{i,j}(n_{i,j}|n_i)$ 가 수렴할 때까지 반복.

3.3 $\mu_i(n_i)$ 구하기

알고리즘 1이 수렴하여 하부네트워크 $R^i(n_i)$ 의 각 하부네트워크의 상태종속 서비스율이 구해지면, 하부네트워크 $R^i(n_i)$ 의 평균 산출률, $X_i(n_i)$ 는 다음과 같이 계산된다.

$$X_i(n_i) = \frac{G_i(n_i-1)}{G_i(n_i)} \quad (16)$$

여기서, $G_i(n_i)$ 는 하부네트워크 $R^i(n_i)$ 의 동등 승법형 대기네트워크의 정규화 상수이다.

하부네트워크 $R^i(n_i)$ 의 평균 산출률, $X_i(n_i)$ 가 구해지면, 제품 i 생산시스템의 상태종속 산출률 $\nu_i(n_i)$ 는 다음과 같이 계산된다.

$$\nu_i(n_i) = X_i(n_i) \quad (17)$$

제품 i 생산시스템의 상태종속 산출률 $\nu_i(n_i)$ 가 계산되면, 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 는 다음과 같이 구해진다.

$$\mu_i(n_i) = \nu_i(n_i) \quad (18)$$

이상의 과정을 정리하면 알고리즘 2와 같다.

알고리즘 2

Step 1. for $i=1, 2$, for $n_i=1, \dots, K_i$

- (a) 하부네트워크 $R^i(n_i)$ 를 알고리즘 1을 이용하여 분석.
- (b) 식(16)을 이용하여 하부네트워크 $R^i(n_i)$ 의 평균 산출률, $X_i(n_i)$ 의 계산.

Step 2. for $i=1, 2$, for $n_i=1, \dots, K_i$

- : 식 (17)과 식 (18)을 이용하여 서버 S^i 의 상태종속 서비스율, $\mu_i(n_i)$ 를 계산.

4. 네트워크-AO의 분석

4.1 TOS 모형

알고리즘 2에 의해서 서버 S^i 의 상태종속 서비스율 $\mu_i(n_i)$ 이 구해지면, 수요과정이 포아송과정에 따라 도착하므로 네트워크-AO는 연속시간 마코브 체인으로 모델링이 가능하며, 마

코브체인의 분석을 통해 각 제품 상태에 대한 결합확률 분포를 구할 수 있다. 시스템의 상태를 정의하기 위해서, 앞에서와 같이 n_i 가 서버 S^i 에 존재하는 간판 수를 나타낸다고 하자 ($i=1, 2$). 만약 서버 S^i 에 n_i 개의 간판이 있다면, 작업장 J_i 에는 $(K_i - n_i)$ 개의 간판이 존재하게 된다. 따라서 시스템의 상태는 2차원 상태인 (n_1, n_2) 로 정의될 수 있으며, 네트워크-AO의 상태공간 ST 는 다음과 같이 정의된다.

$$ST = \{(n_1, n_2) \mid 0 \leq n_1 \leq K_1, 0 \leq n_2 \leq K_2\}$$

만약 상태공간 ST 의 모든 상태를 n_1 을 기준으로 사전식 배치를 하면, 전이율 행렬은 다음과 같은 소행렬로 블록(block)화될 수 있다.

$$Q = \begin{pmatrix} A(0) & A^0 & 0 & 0 & \dots & 0 & 0 \\ \mu_1(1)I & A(1) & A^0 & 0 & \dots & 0 & 0 \\ 0 & \mu_1(2)I & A(2) & A^0 & \dots & 0 & 0 \\ M & M & M & M & O & M & M \\ 0 & 0 & \dots & \dots & \mu_1(K_1-1)I & A(K_1-1) & A^0 \\ 0 & 0 & \dots & \dots & 0 & \mu_1(K_1)I & A(K_1) \end{pmatrix} \quad (19)$$

타입 c 수요의 도착률을 $\lambda^c = q^c \lambda$ 라 정의하자($c=1, 2, 3$). 그러면 $K_2=3$ 일 경우에 식 (19)의 소행렬의 예는 <그림 7>과 같이 표시된다.

<그림 7>의 예로부터 식 (19)의 소행렬들은 다음의 식으로 정리된다.

$$A(n_i) = \{a_{i,j}(n_i)\}, n_i = 0, K, K_1, i = 1, \dots, K_2 + 1, j = 1, \dots, K_2 + 1, \quad (20)$$

$$a_{i,j}(n_i) = \begin{cases} \lambda^2, & \text{if } 1 \leq i \leq K_2, j = i + 1, \\ \mu_2(i-1), & \text{if } 2 \leq i \leq K_2 + 1, j = i - 1, \\ -\lambda^j(n_i, i) - \mu_1(n_i) - \mu_2(i-1), & \text{if } i = j, \\ 0, & \text{o/w.} \end{cases}$$

$$\text{where, } \lambda^i(n_i, i) = \begin{cases} \lambda, & \text{if } n_i < K_1, i \leq K_2, \\ \lambda^1, & \text{if } n_i < K_1, i = K_2 + 1, \\ \lambda^2, & \text{if } n_i = K_1, i \leq K_2, \\ 0, & \text{o/w.} \end{cases}$$

$$A(n_i) = \begin{matrix} & \begin{matrix} (n_i,0) & (n_i,1) & (n_i,2) & (n_i,3) \end{matrix} \\ \begin{matrix} (n_i,0) \\ (n_i,1) \\ (n_i,2) \\ (n_i,3) \end{matrix} & \begin{bmatrix} -\lambda - \mu_1(n_i) & \lambda^2 & & \\ \mu_2(1) & -\lambda - \mu_1(n_i) - \mu_2(1) & \lambda^2 & \\ & \mu_2(2) & -\lambda - \mu_1(n_i) - \mu_2(2) & \lambda^2 \\ & & \mu_2(3) & -\lambda - \mu_1(n_i) - \mu_2(3) \end{bmatrix} \end{matrix}$$

$$A^0 = \begin{matrix} & \begin{matrix} (n_i+1,0) & (n_i+1,1) & (n_i+1,2) & (n_i+1,3) \end{matrix} \\ \begin{matrix} (n_i,0) \\ (n_i,1) \\ (n_i,2) \\ (n_i,3) \end{matrix} & \begin{bmatrix} \lambda^1 & \lambda^2 & & \\ & \lambda^1 & \lambda^2 & \\ & & \lambda^1 & \lambda^2 \\ & & & \lambda^1 \end{bmatrix} \end{matrix}$$

$$A(K_i) = \begin{matrix} & \begin{matrix} (K_i,0) & (K_i,1) & (K_i,2) & (K_i,3) \end{matrix} \\ \begin{matrix} (K_i,0) \\ (K_i,1) \\ (K_i,2) \\ (K_i,3) \end{matrix} & \begin{bmatrix} -\lambda^2 - \mu(K_i) & \lambda^2 & & \\ \mu^2(1) & -\lambda^2 - \mu(K_i) - \mu^2(1) & \lambda^2 & \\ & \mu^2(2) & -\lambda^2 - \mu(K_i) - \mu^2(2) & \lambda^2 \\ & & \mu^2(3) & -\lambda^2 - \mu(K_i) - \mu^2(3) \end{bmatrix} \end{matrix}$$

$$\mu^2(m)I = \begin{matrix} & \begin{matrix} (m-1,0) & (m-1,1) & (m-1,2) & (m-1,3) \end{matrix} \\ \begin{matrix} (m,0) \\ (m,1) \\ (m,2) \\ (m,3) \end{matrix} & \begin{bmatrix} \mu^2(m) & & & \\ & \mu^2(m) & & \\ & & \mu^2(m) & \\ & & & \mu^2(m) \end{bmatrix} \end{matrix}$$

그림 7. $K_2=3$ 일 경우 TOS 모형의 소행렬의 예.

$$\mathbf{A}^0 = \{a^0_{i,j}\}, i=1, \dots, K_2+1, j=1, \dots, K_2+1, \quad (21)$$

$$a^0_{i,j} = \begin{cases} \lambda^1, & \text{if } j = i, \\ \lambda^3, & \text{if } 1 \leq i \leq K_2, j = i+1, \\ 0, & \text{o/w.} \end{cases}$$

$\mathbf{I} =$ 크기 $(K_2+1) \times (K_2+1)$ 의 단위행렬.

식 (19)의 전이율 행렬은 삼각대각 형태를 취하고 있으므로, 앞에서와 같이 행렬기하법(matrix geometric method)을 적용하여 분석할 수 있다. 상태 (n_1, n_2) 의 안정상태확률을 $P_{AO}(n_1, n_2)$ 라 정의했을 때, 소행렬로 표현된 균형방정식은 다음과 같다.

$$\begin{aligned} \mathbf{P}_{AO}(0) \cdot \mathbf{A}(0) + \mu(1)\mathbf{P}_{AO}(1) &= \mathbf{0}, \\ \mathbf{P}_{AO}(n_1-1) \cdot \mathbf{A}^0 + \mathbf{P}_{AO}(n_1) \cdot \mathbf{A}(n_1) + \\ \mu(n_1+1)\mathbf{P}_{AO}(n_1+1) &= \mathbf{0}, n_1=1, 2, \dots, K_1-1, \\ \mathbf{P}_{AO}(K_1-1) \cdot \mathbf{A}^0 + \mathbf{P}_{AO}(K_1) \cdot \mathbf{A}(K_1) &= \mathbf{0}. \end{aligned} \quad (22)$$

여기서, $\mathbf{P}_{AO}(n_1) = \{P_{AO}(n_1, 0), P_{AO}(n_1, 1), \dots, P_{AO}(n_1, K_2)\}$

식 (22)의 시스템의 균형방정식은 반복적인 구조를 가지므로, $\mathbf{P}_{AO}(n_1)$ 는 다음과 같은 순환식으로 표현될 수 있다.

$$\mathbf{P}_{AO}(n_1) = \mathbf{P}_{AO}(n_1-1) \cdot \mathbf{R}_{AO}(n_1), n_1=1, \dots, K_1 \quad (23)$$

여기서, $\mathbf{R}_{AO}(n_1)$ 은 크기 $(K_2+1) \times (K_2+1)$ 의 행렬로 다음과 같다.

$$\mathbf{R}_{AO}(n_1) = \begin{cases} -\mathbf{A}^0 \cdot [\mathbf{A}(n_1) + \mu(n_1+1) \cdot \mathbf{R}_{AO}(n_1+1)]^{-1} \\ \text{if } n_1 = 1, 2, \dots, K_1-1, \\ -\mathbf{A}^0 \cdot \mathbf{A}(n_1)^{-1}, & \text{if } n_1 = K_1 \end{cases} \quad (24)$$

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

$$\begin{aligned} \mathbf{P}_{AO}(0) \cdot [\mathbf{A}(0) + \mu(1)\mathbf{R}_{AO}(1)] &= \mathbf{0} \\ \mathbf{P}_{AO}(0) \sum_{k=0}^{K_1} \mathbf{R}_{AO}(k) \cdot \mathbf{e}_{AO} &= \mathbf{1} \end{aligned} \quad (25)$$

여기서, \mathbf{e}_{AO} 는 모든 원소가 1인 크기 (K_2+1) 의 열벡터이며, $\mathbf{R}_{AO}(n_1)$ 은 크기 $(K_2+1) \times (K_2+1)$ 의 행렬로 다음과 같다.

$$\begin{aligned} \mathbf{A}^{(m)} &= \begin{pmatrix} (m,0) & (m,1) & (m,2) & (m,3) \\ (m,0) & -\lambda - \mu^{(m)} & \lambda & \\ (m,1) & \mu^2(1) & -\lambda - \mu^{(m)} - \mu^2(1) & \lambda^2 \\ (m,2) & & \mu^2(2) & -\lambda - \mu^{(m)} - \mu^2(2) & \lambda^2 \\ (m,3) & & & \mu^2(3) & -\lambda - \mu^{(m)} - \mu^2(3) \end{pmatrix} \\ \mathbf{A}^{(K^1)} &= \begin{pmatrix} (K^1,0) & (K^1,1) & (K^1,2) & (K^1,3) \\ (K^1,0) & -\lambda^2 - \mu^{(K^1)} & \lambda^2 & \\ (K^1,1) & \mu^2(1) & -\lambda^2 - \mu^{(K^1)} - \mu^2(1) & \lambda^2 \\ (K^1,2) & & \mu^2(2) & -\lambda^2 - \mu^{(K^1)} - \mu^2(2) & \lambda^2 \\ (K^1,3) & & & \mu^2(3) & -\mu^{(K^1)} - \mu^2(3) \end{pmatrix} \end{aligned}$$

그림 8. $K_2=3$ 일 경우 POS 모형의 소행렬의 예.

$$\mathbf{RP}_{AO}(k) = \begin{cases} \mathbf{I}, & \text{if } k = 0, \\ \prod_{i=1}^k \mathbf{R}_{AO}(i), & \text{if } k = 1, \dots, K_1 \end{cases}$$

$\mathbf{P}_{AO}(0)$ 가 구해지면 식 (23)으로부터 다른 상태의 안정상태 확률을 구할 수 있다.

4.2 POS 모형

POS 모형은 앞에서 TOS 모형과 모든 가정이 거의 동일하다. 그러나 POS 모형에서는 타입 3 수요가 도착했을 때 제품 1과 제품 2 중 어느 하나의 제품이 존재하지 않을 경우, 전체 수요가 유실되는 것이 아니라 충족될 수 있는 제품은 고객에게 주어진다 가정된다. 그러므로 POS 모형에서는 식 (19)의 전이율 행렬을 구성하는 소행렬이 변화된다. 제품 i 생산시스템에 도착하는 수요율 $\lambda_i = \lambda^i + \lambda^3$ 을 정의하자($i=1, 2$). 그러면 POS 모형에서 TOS 모형의 식 (20)과 식 (21)은 다음과 같이 변화되어 표현된다.

$$\mathbf{A}(n_1) = \{a_{i,j}(n_1)\}, n_1 = 0, \dots, K_1, i = 1, \dots, K_2+1, j = 1, \dots, K_2+1, \quad (26)$$

$$a_{i,j}(n_1) = \begin{cases} \lambda^2 & \text{if } n_1 < K_1, 1 \leq i \leq K_2, j = i+1, \\ \lambda_2 & \text{if } n_1 = K_1, 1 \leq i \leq K_2, j = i+1, \\ \mu_2(i-1), & \text{if } 2 \leq i \leq K_2+1, j = i-1, \\ -\lambda^p(n_1, i) - \mu_1(n_1) - \mu_2(i-1), & \text{if } i = j, \\ 0, & \text{o/w.} \end{cases}$$

$$\text{where, } \lambda^p(n_1, i) = \begin{cases} \lambda, & \text{if } n_1 < K_1, i \leq K_2, \\ \lambda_1, & \text{if } n_1 < K_1, i = K_2+1, \\ \lambda_2, & \text{if } n_1 = K_1, i \leq K_2, \\ 0, & \text{o/w.} \end{cases}$$

$$\mathbf{A}^0 = \{a^0_{i,j}\}, i=1, \dots, K_2+1, j=1, \dots, K_2+1, \quad (27)$$

$$a^0_{i,j} = \begin{cases} \lambda^1, & \text{if } 1 \leq i \leq K_2, j = i, \\ \lambda_1, & \text{if } i = j = K_2+1, \\ \lambda^3, & \text{if } 1 \leq i \leq K_2, j = i+1, \\ 0, & \text{o/w.} \end{cases}$$

$K_2=3$ 일 경우의 예를 들면 식 (26)과 식 (27)의 소행렬은 다음과 같이 표시되며, 안정상태확률 $P_{AO}(n_1, n_2)$ 을 구하는 방

법은 TOS 모형과 동일하다.

$$F = q^1 F^1 + q^2 F^2 + q^3 F^3 \quad (33)$$

각 저장창고의 평균 재고수준 \bar{I}_i , 각 작업장의 평균 재공품 수 $\bar{K}_{i,j}$ 및 평균 가공상태의 기계 수 $\bar{S}_{i,j}$ 는 다음과 같이 구할 수 있다.

4.3 시스템 성능최도의 계산

네트워크-AO의 안정상태확률 $P_{AO}(n_1, n_2)$ 이 계산되면, 제품 i 생산시스템에 존재하는 간판 수의 안정상태확률 $P_i(n_i)$ 는 $P_{AO}(n_1, n_2)$ 의 주변 확률분포이므로 $P_{AO}(n_1, n_2)$ 에서 쉽게 계산할 수 있다. $P_i(n_i)$ 가 계산되면, 작업장 (i, j) 에서 간판 수의 안정상태확률 $P_{i,j}(n_{i,j})$ 는 다음과 같이 계산된다.

$$\bar{I}_i = \sum_{n_i=0}^{K_i-1} (K_i - n_i) P_i(n_i), \quad i = 1, 2 \quad (34)$$

$$\bar{K}_{i,j} = \sum_{n_{i,j}=1}^{K_i} n_{i,j} P_{i,j}(n_{i,j}), \quad j = 1, K, M_i, i = 1, 2 \quad (35)$$

$$P_{i,j}(n_{i,j}) = \begin{cases} P_i(0) + \sum_{n_i=1}^{K_i} P_{i,j}(n_{i,j} | n_i) P_i(n_i), & \text{for } n_{i,j} = 0, \\ \sum_{n_i=n_{i,j}}^{K_i} P_{i,j}(n_{i,j} | n_i) P_i(n_i) & \text{for } 1 \leq n_i \leq K_i \end{cases} \quad (28)$$

$$\bar{S}_{i,j} = \sum_{n_{i,j}=1}^{K_i} M n_0(n_{i,j}) P_{i,j}(n_{i,j}), \quad j = 1, K, M_i, i = 1, 2 \quad (36)$$

참고로 POS 모형의 경우, 생산시스템 관점에서 한 제품의 재고수준은 다른 제품의 재고수준과 독립이다. 따라서 제품 i 생산시스템은 도착률 λ_i 와 상태종속 서비스율 $\mu_i(n_i)$ 을 가지는 $M/M/1/K_i$ 시스템으로 모델링될 수 있으며, 이로부터 제품 i 생산시스템에 존재하는 안정상태확률 $P_i(n_i)$ 를 계산할 수 있다.

5. 근사적 분석의 정확도 평가

각 타입의 수요가 도착한 순간 수요가 충족될 확률 F^i 는 다음과 같이 계산된다.

본 연구에서는 시스템 특성의 관찰을 위하여 많은 실험을 수행하였으며, 제안된 근사법의 정확도를 평가하기 위하여 근사법에서 얻어진 값과 시뮬레이션에서 얻어진 값을 비교하였다. 시뮬레이션의 종료조건으로 시뮬레이션 시간을 10^6 으로 설정하여 10번 반복하였고, 알고리즘의 종료조건은 10^{-5} 이하로 하였다. 본 연구에서 실험한 예제에서 알고리즘 1의 반복 횟수는 간판 수에 따른 변동은 있으나, 대부분 10번 이내의 반복에서 수렴함을 관측할 수 있었고, 모든 예제에서 수렴하였다. 이 절에서는 파라미터 값이 변화함에 따라 시스템의 성능최도가 어떻게 변화하는지 알아보기 위해 대표적인 4개의 예제 군을 선정하여 소개한다. 첫 번째 예제 군에서는 도착률 λ 의 값을 변화시켜 보았다. 두 번째 예제 군에서는 q^3 의 비율을 증가시켜 제품들 간의 종속관계를 증가시켜 보았으며, 세 번째 예제 군에서는 q^3 을 고정시키고, q^1 와 q^2 을 변화시켜 두 제품 생산 시스템 간의 비대칭(asymmetric)의 증가가 시스템 성능치에 미치는 영향을 조사해 보았다. 마지막으로 네 번째 예제 군에서는 제품 생산시스템의 간판 수를 변화시켜 보았다. 편의상 모든 예제 군에서 같은 제품 생산 시스템에 속하는 모든 작업장에서 가공시간의 분포는 동일하다고 가정하였다. 실험데이터와 실험결과는 다음의 표와 그림으로 요약된다.

$$F^i = \sum_{n_i=0}^{K_i-1} P_i(n_i), \quad i = 1, 2 \quad (29)$$

$$F^3 = \sum_{n_1=0}^{K_1-1} \sum_{n_2=0}^{K_2-1} P_{AO}(n_1, n_2) \quad (30)$$

F^i 가 계산되면, 제품 i 생산시스템에서의 수요가 충족될 확률 F^i 와 임의의 수요가 즉시 수용될 확률 F 는 다음과 같다.

(i) TOS 모형: $F_i = \frac{q^i}{q^i + q^3} F^i + \frac{q^3}{q^i + q^3} F^3, \quad i = 1, 2 \quad (31)$

(ii) POS 모형: $F_i = F^i, \quad i = 1, 2 \quad (32)$

표 1. 실험 데이터

예제 군	λ	q^1	q^2	q^3	K_1	K_2	M_1	M_2	$SE_{i,j}$	$\mu_{i,j}^1$	$\mu_{i,j}^2$	$c_{1,j}$	$c_{2,j}$
1	*	1/3	1/3	1/3	15	15	10	10	2	1	1	0.5	0.3
2	0.5	*	*	*	10	10	5	5	3	0.5	0.5	0.3	0.5
3	4	*	*	1/3	20	20	3	3	2	1.5	1.5	0.2	0.2
4	1.2	1/3	1/3	1/3	*	*	7	7	2	1	2	0.7	0.3

(**"은 해당 예제 군의 표에서 명시 됨.)

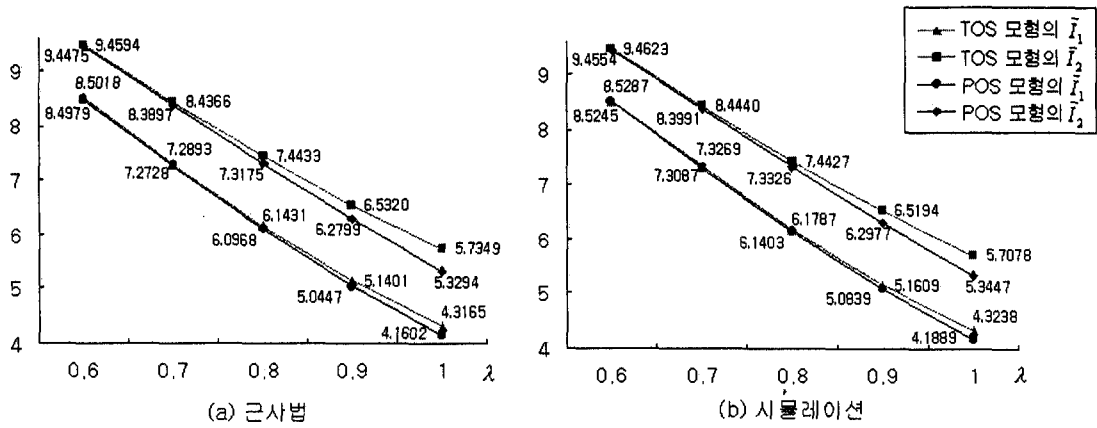


그림 9. λ의 변화에 따른 재고수준의 변화(예제 군 1).

표 2. 예제 군 1의 실험결과

	TOS Model						POS Model				
	F^1	F^2	F^3	F_1	F_2	F	$F^1 = F_1$	$F^2 = F_2$	F^3	F	
$\lambda = 0.6$	근사법	0.9963	0.9990	0.9953	0.9958	0.9971	0.9968	0.9963	0.9989	0.9952	0.9968
	시뮬레이션	0.9968	0.9990	0.9959	0.9963	0.9975	0.9972	0.9967	0.9990	0.9958	0.9971
	Rel. Error (%)	-0.0502	0.0000	-0.0602	-0.0502	-0.0401	-0.0401	-0.0401	-0.0100	-0.0603	-0.0301
$\lambda = 0.7$	근사법	0.9876	0.9960	0.9837	0.9857	0.9899	0.9891	0.9874	0.9957	0.9833	0.9888
	시뮬레이션	0.9889	0.9964	0.9855	0.9872	0.9909	0.9903	0.9886	0.9959	0.9849	0.9898
	Rel. Error (%)	-0.1315	-0.0401	-0.1826	-0.1519	-0.1009	-0.1212	-0.1214	-0.0201	-0.1625	-0.1010
$\lambda = 0.8$	근사법	0.9703	0.9893	0.9600	0.9652	0.9747	0.9732	0.9691	0.9876	0.9580	0.9716
	시뮬레이션	0.9732	0.9902	0.9640	0.9686	0.9771	0.9758	0.9718	0.9883	0.9617	0.9739
	Rel. Error (%)	-0.2980	-0.0909	-0.4149	-0.3510	-0.2456	-0.2664	-0.2778	-0.0708	-0.3847	-0.2362
$\lambda = 0.9$	근사법	0.9441	0.9776	0.9230	0.9336	0.9503	0.9482	0.9404	0.9724	0.9169	0.9433
	시뮬레이션	0.9484	0.9791	0.9294	0.9389	0.9543	0.9523	0.9447	0.9735	0.9231	0.9471
	Rel. Error (%)	-0.4534	-0.1532	-0.6886	-0.5645	-0.4192	-0.4305	-0.4552	-0.1130	-0.6716	-0.4012
$\lambda = 1.0$	근사법	0.9115	0.9611	0.8757	0.8936	0.9184	0.9161	0.9034	0.9492	0.8623	0.9049
	시뮬레이션	0.9172	0.9633	0.8845	0.9009	0.9239	0.9217	0.9089	0.9508	0.8708	0.9102
	Rel. Error (%)	-0.6215	-0.2284	-0.9949	-0.8103	-0.5953	-0.6076	-0.6051	-0.1683	-0.9761	-0.5823

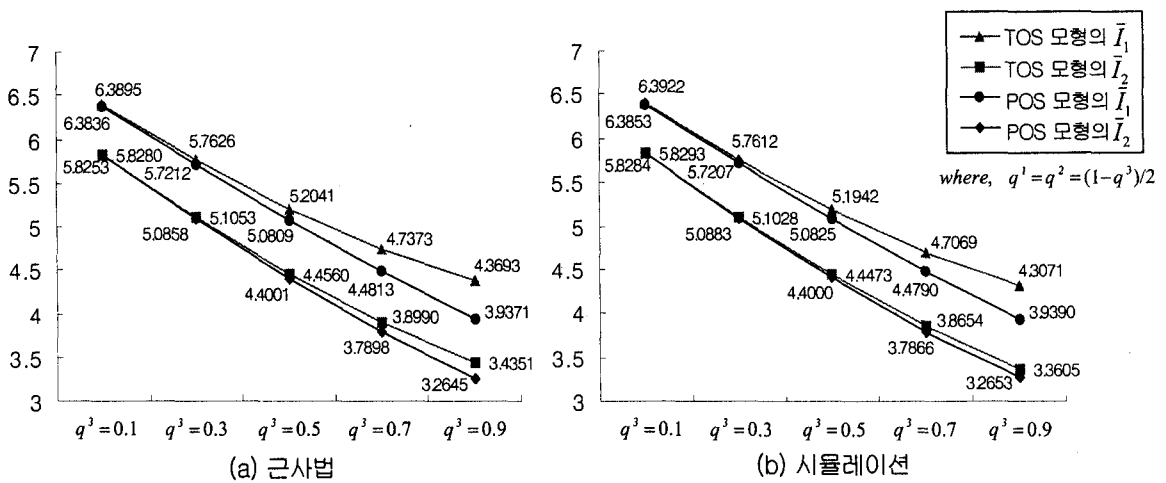
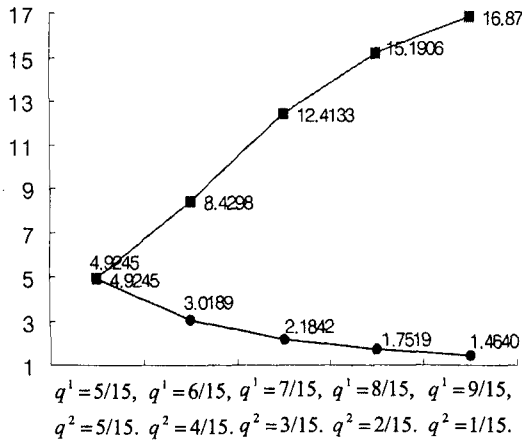


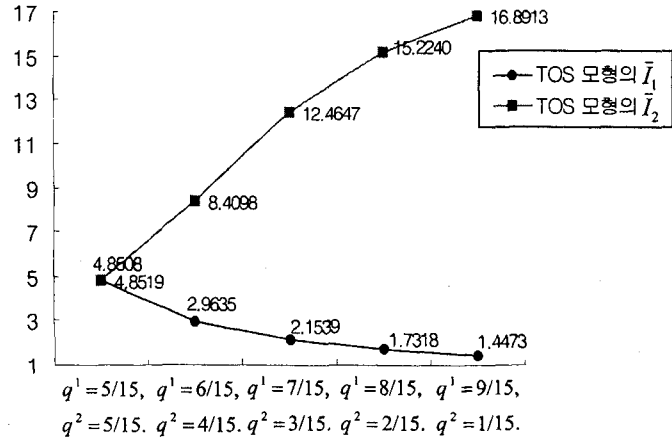
그림 10. q^3 의 변화에 따른 재고수준의 변화(예제 군 2).

표 3. 예제 군 2의 실험결과

		TOS Model						POS Model			
		F^1	F^2	F^3	F_1	F_2	F	$F^1 = F_1$	$F^2 = F_2$	F^3	F
$q_1=0.45$	근사법	0.9963	0.9912	0.9875	0.9947	0.9905	0.9931	0.9962	0.9911	0.9874	0.9931
$q_2=0.45$	시뮬레이션	0.9963	0.9914	0.9874	0.9948	0.9908	0.9933	0.9963	0.9914	0.9878	0.9933
$q_3=0.10$	Rel.Error (%)	0.0000	-0.0202	0.0101	-0.0101	-0.0303	-0.0201	-0.0100	-0.0303	-0.0405	-0.0201
$q_1=0.35$	근사법	0.9905	0.9785	0.9694	0.9807	0.9743	0.9800	0.9897	0.9779	0.9684	0.9792
$q_2=0.35$	시뮬레이션	0.9909	0.9792	0.9706	0.9815	0.9752	0.9807	0.9899	0.9785	0.9695	0.9899
$q_3=0.30$	Rel.Error (%)	-0.0404	-0.0715	-0.1236	-0.0815	-0.0923	-0.0714	-0.0202	-0.0613	-0.1135	-1.0809
$q_1=0.25$	근사법	0.9815	0.9594	0.9421	0.9552	0.9478	0.9542	0.9776	0.9564	0.9376	0.9523
$q_2=0.25$	시뮬레이션	0.9827	0.9607	0.9454	0.9578	0.9505	0.9586	0.9779	0.9574	0.9404	0.9540
$q_3=0.50$	Rel.Error (%)	-0.1221	-0.1353	-0.3491	-0.2715	-0.2841	-0.4590	-0.0307	-0.1044	-0.2977	-0.1782
$q_1=0.15$	근사법	0.9704	0.9350	0.9080	0.9190	0.9128	0.9214	0.9593	0.9272	0.8964	0.9105
$q_2=0.15$	시뮬레이션	0.9737	0.9372	0.9152	0.9255	0.9191	0.9273	0.9595	0.9285	0.9019	0.9145
$q_3=0.70$	Rel.Error (%)	-0.3389	-0.2347	-0.7867	-0.7023	-0.6855	-0.6363	-0.0208	-0.1400	-0.6098	-0.4374
$q_1=0.05$	근사법	0.9587	0.9075	0.8705	0.8752	0.8725	0.8768	0.9350	0.8923	0.8479	0.8545
$q_2=0.05$	시뮬레이션	0.9659	0.9089	0.8823	0.8867	0.8837	0.8878	0.9356	0.8942	0.8574	0.8631
$q_3=0.90$	Rel.Error (%)	-0.7454	-0.1540	-1.3374	-1.2969	-1.2674	-1.2390	-0.0641	-0.2125	-1.1080	-0.9964



(a) 근사법



(b) 시뮬레이션

그림 11. 비대칭성의 증가에 따른 재고수준의 변화(예제 군 3).

표 4. 예제 군 3의 실험결과

		TOS Model						POS Model			
		F^1	F^2	F^3	F_1	F_2	F	$F^1 = F_1$	$F^2 = F_2$	F^3	F
$q_1=5/15$	근사법	0.8672	0.8672	0.7452	0.8063	0.8063	0.8266	0.8204	0.8204	0.6864	0.7758
$q_2=5/15$	시뮬레이션	0.8686	0.8687	0.7488	0.8087	0.8088	0.8287	0.8216	0.8217	0.6894	0.7776
	Rel.Error (%)	-0.1612	-0.1727	-0.4808	-0.2968	-0.3091	-0.2534	-0.1461	-0.1582	-0.4352	-0.2315
$q_1=6/15$	근사법	0.7702	0.9555	0.7323	0.7530	0.8315	0.8070	0.7550	0.8889	0.6823	0.7665
$q_2=4/15$	시뮬레이션	0.7707	0.9578	0.7352	0.7546	0.8341	0.8088	0.7560	0.8905	0.6851	0.7683
	Rel.Error (%)	-0.0649	-0.2401	-0.3945	-0.2120	-0.3117	-0.2226	-0.1323	-0.1797	-0.4087	-0.2343
$q_1=7/15$	근사법	0.6984	0.9938	0.6936	0.6964	0.8061	0.7559	0.6965	0.9490	0.6673	0.7373
$q_2=3/15$	시뮬레이션	0.6990	0.9948	0.6950	0.6974	0.8074	0.7568	0.6975	0.9503	0.6693	0.7386
	Rel.Error (%)	-0.0858	-0.1005	-0.2014	-0.1434	-0.1610	-0.1189	-0.1434	-0.1368	-0.2988	-0.1760
$q_1=8/15$	근사법	0.6454	0.9997	0.6452	0.6454	0.7465	0.6926	0.6454	0.9855	0.6381	0.6883
$q_2=2/15$	시뮬레이션	0.6463	0.9998	0.6462	0.6463	0.7472	0.6934	0.6462	0.9863	0.6394	0.6893
	Rel.Error (%)	-0.1393	-0.0100	-0.1548	-0.1393	-0.0937	-0.1154	-0.1238	-0.0811	-0.2033	-0.1451
$q_1=9/15$	근사법	0.6007	1.0000	0.6007	0.6007	0.6672	0.6273	0.6007	0.9979	0.5997	0.6278
$q_2=1/15$	시뮬레이션	0.6015	1.0000	0.6014	0.6014	0.6679	0.6280	0.6016	0.9980	0.6007	0.6277
	Rel.Error (%)	-0.1330	0.0000	-0.1164	-0.1164	-0.1048	-0.1115	-0.1496	-0.0100	-0.1665	0.0159

표 5. 예제 군 4의 실험결과

		TOS Model						POS Model			
		F^1	F^2	F^3	F_1	F_2	F	$F^1 = F_1$	$F^2 = F_2$	F^3	F
$K_1=K_2=11$	근사법	0.8807	0.9410	0.8281	0.8544	0.8845	0.8833	0.8679	0.9226	0.8086	0.8664
	시뮬레이션	0.8898	0.9440	0.8414	0.8656	0.8927	0.8917	0.8772	0.9256	0.8223	0.8750
	Rel. Error(%)	-1.0227	-0.3178	-1.5807	-1.2939	-0.9186	-0.9420	-1.0602	-0.3241	-1.6661	-0.9829
$K_1=K_2=12$	근사법	0.9058	0.9574	0.8670	0.8864	0.9122	0.9101	0.8973	0.9452	0.8534	0.8986
	시뮬레이션	0.9149	0.9600	0.8795	0.8972	0.9198	0.9181	0.9065	0.9478	0.8663	0.9069
	Rel. Error(%)	-0.9946	-0.2708	-1.4213	-1.2037	-0.8263	-0.8714	-1.0149	-0.2743	-1.4891	-0.9152
$K_1=K_2=13$	근사법	0.9266	0.9698	0.8987	0.9127	0.9342	0.9317	0.9211	0.9619	0.8895	0.9242
	시뮬레이션	0.9354	0.9722	0.9103	0.9228	0.9412	0.9393	0.9299	0.9643	0.9013	0.9318
	Rel. Error(%)	-0.9408	-0.2469	-1.2743	-1.0945	-0.7437	-0.8091	-0.9463	-0.2489	-1.3092	-0.8156
$K_1=K_2=14$	근사법	0.9436	0.9760	0.9239	0.9338	0.9514	0.9488	0.9401	0.9741	0.9180	0.9441
	시뮬레이션	0.9518	0.9809	0.9342	0.9430	0.9576	0.9556	0.9483	0.9760	0.9285	0.9510
	Rel. Error(%)	-0.8615	-0.4995	-1.1025	-0.9756	-0.6475	-0.7116	-0.8647	-0.1947	-1.1309	-0.7256
$K_1=K_2=15$	근사법	0.9573	0.9857	0.9437	0.9505	0.9647	0.9622	0.9552	0.9827	0.9400	0.9593
	시뮬레이션	0.9646	0.9872	0.9527	0.9586	0.9699	0.9682	0.9626	0.9842	0.9492	0.9653
	Rel. Error(%)	-0.7568	-0.1519	-0.9447	-0.8450	-0.5361	-0.6197	-0.7688	-0.1524	-0.9692	-0.6216

제시된 결과에서 볼 수 있듯이 본 연구에서 제안된 근사법은 일반적으로 만족할 만한 수준의 정확한 해를 산출해 준다. 대부분의 경우 상대오차가 1% 미만이고, 관측된 최대 상대오차는 1.6%이다. 관측된 시스템의 중요한 특성을 정리하면 다음과 같다.

위의 실험결과에서 제품 생산시스템 관점의 F_i 는 TOS 모형과 비교하여 POS 모형에서 더 좋은 결과를 나타내는 현상을 볼 수 있는데, 이러한 현상은 결국 POS 모형에서 제품 생산시스템의 주문 밀집도(congestion)를 TOS 모형보다 증가시키게 된다. 따라서 위의 표와 그림에서 확인되었듯이 고객관점의 F^i 와 시스템 관점의 F 는 POS 모형에서 감소하게 된다. 다음으로 <표 2>에서 주목할 만한 현상은 F^1 과 F^2 가 높은 경우, $\min\{F_1, F_2\}$ 은 F^3 의 비교적 좋은 근사해를 준다. 그러나 F^1 과 F^2 가 낮아질수록, 두 제품 간의 종속적인 관계가 증가하여, $\min\{F_1, F_2\}$ 는 F^3 의 좋은 근사값을 주지 못함을 알 수 있다. <표 4>에서는 두 제품생산 시스템 간의 비대칭성의 예를 보여 준다. 비대칭성이 증가함에 따라 두 시스템간의 작업량(work load)의 불균형이 증가하여, F^3 의 결과치가 상당히 감소함을 알 수 있다. 따라서 각 제품 생산 시스템 간에 작업량의 균형이 주문-조립시스템의 설계에서 중요한 요소임을 알 수 있다. 비록 본 논문에서는 지면 관계상 4개의 예제의 결과만을 제시했으나, 다른 예제에 대해서도 위에서 얻어진 결과와 비슷한 양상을 보였음을 밝힌다.

6. 결론

본 연구에서는 두 종류의 제품을 생산하여 세 종류의 수요를 충족시키는 다품목 생산시스템에서 수요가 유실된다는 가정 하에 TOS 모형과 POS 모형을 분석하기 위한 근사법을 개발하였다. 기존 연구가 제품의 생산/조달방식을 지나치게 단순하

게 가정하여 현실적인 면을 반영하지 못했다면, 본 연구에서는 제품이 여러 단계를 거쳐 생산되고, 제품의 생산방식은 CONWIP 제어방식에 의해 이루어진다고 가정하여 보다 현실적인 면을 고려하였다. 또한 본 연구에서는 제조 시스템의 생산시간분포로 널리 사용되는 Coxian-2 분포를 제품 가공시간의 분포로 사용하여 기존 연구의 가정을 완화하였다. 그러나 이와 같이 완화된 가정과 다 제품수요를 가지는 시스템에 내재된 제품들 간의 종속관계는 시스템 분석의 어려움을 준다. 따라서 본 연구에서는 이를 해결하고자 각 제품 생산시스템을 서비스율이 상태 의존적인 단일서버로 치환함으로써, 시스템을 단순화하여 제품 상태의 결합확률분포를 효율적으로 계산하였다. 또한 제품 생산시스템의 서비스율을 추정하기 위하여 통합법을 사용하였으며, 근사화 과정에서 Marie의 승법형 근사법과 행렬기하법을 이용하였다. 본 연구에서 제안된 근사법은 제품 상태의 결합 확률분포, 재고 창고의 평균 재고수준, 각 작업장에서 제품 수의 안정상태확률 등 시스템의 중요한 성능 척도를 비교적 정확하고 신속하게 계산하여 준다. 따라서 본 연구에서 제안된 근사법은 향후 다양한 수요를 갖는 다품목 생산시스템의 경제적인 설계 및 효율적인 운영에 유용하게 이용되어 질 수 있을 것으로 기대된다.

부록

$n_i=5, SE_{i,j}=2$ 일 경우에 식 (1~4)의 전이율 행렬과 소행렬은 다음과 같다.

$$Q_{i,j}(5) = \begin{pmatrix} W(0) & Z(0) & 0 & 0 & 0 & 0 \\ H(1) & W(1) & Z(1) & 0 & 0 & 0 \\ 0 & H(2) & W(2) & Z(2) & 0 & 0 \\ 0 & 0 & H(3) & W(3) & Z(3) & 0 \\ 0 & 0 & 0 & H(4) & W(4) & Z(4) \\ 0 & 0 & 0 & 0 & H(5) & W(5) \end{pmatrix}$$

$$\begin{aligned}
 W(0) &= \begin{matrix} (0,0) \\ (0,0) \end{matrix} \left[-\lambda_{i,j}(0|5) \right] & W(1) &= \begin{matrix} (1,0) & (1,1) \\ (1,1) \end{matrix} \left[\begin{matrix} -\lambda_{i,j}(1|5) - \mu_{i,j}^2 & \\ c_{i,j}\mu_{i,j}^1 & -\lambda_{i,j}(1|5) - \mu_{i,j}^1 \end{matrix} \right] \\
 W(m_1) &= \begin{matrix} (m_1,0) & (m_1,1) & (m_1,2) \\ 2 \leq m_1 \leq 4 & (m_1,1) & (m_1,2) \\ (m_1,2) \end{matrix} \left[\begin{matrix} -\lambda_{i,j}(m_1|5) - 2\mu_{i,j}^2 & & \\ c_{i,j}\mu_{i,j}^1 & -\lambda_{i,j}(m_1|5) - \mu_{i,j}^1 - \mu_{i,j}^2 & \\ & 2c_{i,j}\mu_{i,j}^1 & -\lambda_{i,j}(m_1|5) - 2\mu_{i,j}^1 \end{matrix} \right] & W(5) &= \begin{matrix} (5,0) & (5,1) & (5,2) \\ (5,0) & (5,1) & (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] \\
 Z(0) &= \begin{matrix} (1,0) & (1,1) \\ (0,0) \end{matrix} \left[\begin{matrix} \lambda_{i,j}(0|5) \end{matrix} \right] & Z(1) &= \begin{matrix} (2,0) & (2,1) & (2,2) \\ (1,0) & (1,1) \end{matrix} \left[\begin{matrix} \lambda_{i,j}(1|5) & & \\ & \lambda_{i,j}(1|5) & \end{matrix} \right] & Z(m_1) &= \begin{matrix} (m_1+1,0) & (m_1+1,1) & (m_1+1,2) \\ 2 \leq m_1 \leq 4 & (m_1,1) & (m_1,2) \\ (m_1,2) \end{matrix} \left[\begin{matrix} \lambda_{i,j}(m_1|5) & & \\ & \lambda_{i,j}(m_1|5) & \\ & & \lambda_{i,j}(m_1|5) \end{matrix} \right] \\
 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] & H(2) &= \begin{matrix} (1,0) & (1,1) \\ (2,0) & (2,1) & (2,2) \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] & H(m_1) &= \begin{matrix} (m_1-1,0) & (m_1-1,1) & (m_1-1,2) \\ 3 \leq m_1 \leq 5 & (m_1,1) & (m_1,2) \\ (m_1,2) \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}$$

참고문헌

Baynat, B. and Dallery, Y. (1993), a Unified View of Product-Form Approximation Techniques for General Closed Queueing Networks, *Performance Evaluation*, **18**, 205-224.

Cheung, K. L. and Hausman, W. H. (1995), Multiple Failures in a Multi-Item Spares Inventory Model, *IIE Transactions*, **27**, 171-180.

Duenyas, I. and Hopp, W. J. (1992), CONWIP Assembly System with Deterministic Processing and Random Outages, *IIE Transactions*, **24**(4), 97-111.

Duenyas, I. and Hopp, W. J. (1993), Estimating the Throughput of an Exponential CONWIP Assembly System, *Queueing System*, **14**, 135-157.

Duenyas, I., Hopp, W. J. and Spearman, M. L. (1993), Characterizing the Output Process of a CONWIP Line with Deterministic Processing and Random Outages, *Management Science*, **39**(8), 975-988.

Gattorna, J. (1998), Strategic Supply Chain Alignment, Gower, Hampshire, England.

Glasserman, P. and Wang, Y. (1998), Leadtime-Inventory Trade-Offs in Assemble to Order Systems, *Operations Research*, **46**(6), 858-871.

Harrison, P. G. and Patel, N. M. (1992), Performance Modeling of Communication Networks and Computer Architectures, Addison-Wesley, Seoul, Korea.

Hausman, W. H. (1982), On Optimal Repair Kits under a Job Completion Criterion, *Management Science*, **28**(11), 1350-1351.

Lee, H. L. and Billington, C. (1993), Material Management in Decentralized Supply Chains, *Operations Research*, **41**(5), 835-847.

Lee, H. S. and Lee, J. E. (1998), Approximate Analysis of a CONWIP system with a Lot Production, *Journal of the Korean Institute of industrial Engineers*, **11**(3), 53-64.

Lee, J. E. and Lee, H. S. (1998), Approximate Analysis of a CONWIP system with Compound Poisson Demands, *Journal of the Korean Operations Research and Management Science*, **23**(3), 153-168.

Mamer, J. W. and Smith, S. A. (1982), Optimizing Field Repair Kits Based on Job Completion Rate, *Management Science*, **28**(11), 1328-1333.

Mascolo, M. D., Frein, Y. and Dallery, Y. (1996), an Analytical Method for Performance Evaluation of Kanban Controlled Production Systems, *Operations Research*, **44**(1), 50-64.

Marie, R. A. (1979), an Approximate Analytical Method for General Queueing Networks, *IEEE Trans. Soft. Eng.*, **SE-5**(5), 530-538.

Neuts, M. F. (1981), Matrix Geometric Solutions in Stochastic Models-an Algorithmic Approach, John Hopkins University Process, Baltimore.

Smith, S. A., Chambers, J. C. and Shlifer, E. (1980), Optimal Inventories Based on Job Completion Rate for Repairs Requiring Multiple Items, *Management Science*, **26**(8), 849-852.

Song, J. S. (1998), On the Order Fill Rate in a Multi-item, Base-Stock Inventory System, *Operations Research*, **46**(6), 831-845.

Song, J. S., Xu, S. H. and Lin, B. (1999), Order-Fulfillment Performance Measures in an Assemble-to-Order System with Stochastic Leadtimes, *Operations Research*, **47**(1), 131-149.

Spearman, M. L., Woodruff, D. L. and Hopp, W. J. (1990), CONWIP: a Pull Alternative to Kanban, *Int. J. Prod. Res.*, **28**(5), 879-894.