

로트 단위로 가공되는 CONWIP 시스템의 근사적 분석*

이효성** · 이정은**

Approximate Analysis of a CONWIP System with a Lot Production

Hyo-Seong Lee · Jeong Eun Lee

〈Abstract〉

In this study we consider a CONWIP system in which the processing times at each station follow an exponential distribution and the demands for the finished products arrive according to a compound Poisson process. The demands that are not satisfied instantaneously are assumed to be lost. We assume that the lot size at each station is greater than one. For this system we develop an approximation method to obtain the performance measures such as steady state probabilities of the number of parts at each station, average number of parts at each station and the proportion of lost demands. For the analysis of the proposed CONWIP system, we model the CONWIP system as a closed queueing network with a synchronization station and analyze the closed queueing network using a product form approximation method. A recursive technique is used to solve the subnetwork in the application of the product-form approximation method. To test the accuracy of the approximation method, the results obtained from the approximation method were compared with those obtained by simulation. Comparisons with simulation have shown that the accuracy of the approximate method is acceptable.

1. 서 론

칸반(Kanban) 시스템은 적시생산시스템(Just-In-Time)의 하부시스템으로 칸반을 통해 각 작업장에서의 생산 통제가 이루어지는 일종의 견인 생산시스템(pull control system)이다. 칸반 시스템이 성공을 거둔 이후 견인 생산시스템에 대한 연구가 활발히 진행되어져 왔다[7,8]. 견인 생산시스템은 밀어내기식 생산시스템(push control system)에 비하여 여러 측면에서 장점을 지니고 있다고 보고되어 왔으나[9] 두 시스템은 각기 다른 특성을 지니고 있는 별개의 생산 시스템이므로 모든 생산환경에서 어느 한 시스템이 항상 우월하다고 단정지을 수는 없다. 이러한 이유에서 두 시스템의 관리방식을 혼합하여 두

시스템이 지니고 있는 장점이 모두 구현될 수 있는 혼합 시스템(hybrid system)에 대한 연구가 최근에 활발히 진행되어왔다 [4,5,7]. 그 중에서도 CONWIP 시스템(Constant Work-In-Process System)은 운영방식이 단순하고 적용범위가 넓어 특별히 많은 관심을 끌고 있다[3,9].

CONWIP 시스템에서는 시스템내의 재공품의 수(완제품 포함)가 일정하게 유지되며, 수요가 발생하여 완제품 재고가 인출 될 경우에만 이를 보충하기 위하여 생산지시가 이루어진다. 수요발생에 의해 생산이 견인되므로 이러한 측면에서 보면 CONWIP 시스템은 견인 생산시스템의 특성을 지니고 있다고 할 수 있다. 그러나 완제품의 수요발생이 칸반 시스템에서와 같이 바로 전 공정의 작업을 견인하는 것이 아니고 첫

* 본 연구는 97년도 한국과학재단 연구지원(과제번호 971-1013-078-1)에 의한 결과임.

** 경희대학교 산업공학과

공정의 작업을 견인하며, 원자재가 첫 공정에서 가공을 마치게 되면 완제품이 될 때까지 밀어내기 방식에 의해 특별한 통제 없이 후속 공정에서의 가공이 이루어진다. 따라서 이러한 측면에서는 CONWIP 시스템은 밀어내기식 생산시스템의 특성을 지니고 있다고 볼 수 있다.

Spearman[14]은 stochastic coupling 방식을 사용하여 CONWIP 시스템과 칸반 시스템의 성능을 비교하였다. 그 결과 고객 만족도 측면에서 CONWIP 시스템이 칸반시스템에 비해 우수함을 증명하였다. 고객만족도 측면 이외에도 CONWIP 시스템은 통제방식이 용이하고, 각 공정에서의 생산준비 시간이 비교적 큰 경우에도 적용될 수 있다는 점 등 부수적인 장점이 있는 것으로 보고되고 있다. Duenyas, Hopp, Spearman[5]은 각 공정에서의 가공시간이 일정하고(deterministic) 기계고장이 발생할 수 있는 CONWIP 생산라인 시스템(flow line system)을 분석하였다. 이 연구에서는 기계의 고장발생 간격과 수리시간이 모두 지수분포를 따르고, 완제품에 대한 수요는 항상 존재(saturated demands)한다는 가정 하에서 시스템의 생산율을 추정하기 위한 근사적 방법이 제시되었다. Duenyas와 Hopp은 이 연구를 확장하여 CONWIP assembly 시스템에 대한 분석을 시도하였다. 이들은 각 작업장에서의 가공시간이 일정한 경우[3]와 지수분포를 따르는 경우[4]에 대해 각각 시스템의 생산율을 구하기 위한 근사적 방법을 개발하였다. 완제품에 대한 수요가 Poisson 과정에 따라 발생하는 시스템의 연구는 Mascolo, Frein, Dallery [12]에 의해 수행되었다. 이 연구에서는 각 작업장에서의 가공시간이 2단계 Coxian 분포를 따른다는 가정 하에서 시스템의 안정상태 확률을 구하기 위한 근사적 기법이 제시되었다.

이상에서 살펴본 바와 같이 CONWIP 시스템에 대한 분석적 연구가 최근 활발히 수행되어져 왔다. 그러나 이들 기존의 연구는 각 작업장에서의 가공이 로트 단위로 이루어지지 않고 개별적으로 이루어진다는 가정 하에서 수행되었다[3,4,8,12]. 그러나 생산 준비비용이 많이 소요되는 생산시스템에서는 일반적으로 각 작업장에서의 가공이 로트 단위로 이루어진다. 이와 같은 상황을 반영하기 위하여 본 연구에서는 각 작업장에서의 가공이 로트 단위로 이루어지는 CONWIP 시스템을 분석하기 위한 근사적기법을 제시하고자 한다.

2. 모형의 설명

본 연구에서 분석하고자 하는 CONWIP 시스템은 M개의 작

업장과 완제품 재고창고로 구성된 흐름식 생산라인 시스템(flow line system)으로 각 작업장에는 1대의 기계가 존재한다. 작업장 1에서 최초의 가공을 받은 부품은 작업장 2, 3,...에서 후속 가공을 거친 후 작업장 M에서 최후 가공을 받게되며, 작업장 M에서 가공을 마친 부품은 완제품으로 출하되어 완제품 재고창고에 저장된다. CONWIP 시스템에서도 칸반 시스템에서와 같이 부품은 컨테이너에 적재되어 작업장 간을 이동하며 완제품의 경우도 컨테이너에 적재된 채 저장된다고 가정한다. CONWIP 시스템에서는 시스템내의 재공품재고를 일정한 수준으로 유지하기 위하여 사용되는 컨테이너의 개수를 제한하며, 본 연구에서는 K개의 컨테이너가 사용된다고 가정한다.

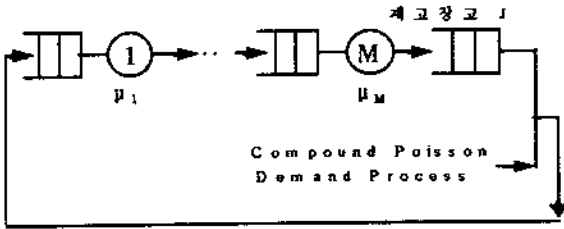
각 작업장에서의 로트의 크기가 n일 경우에는 컨테이너에 적재되는 부품의 개수는 n이 된다. 컨테이너에 적재되어 있는 n개의 완제품이 수요에 의해 모두 인출되어 컨테이너가 비게 되면 컨테이너는 즉시 작업장 1으로 이동된다. 작업장 1에는 충분한 수의 원자재가 확보되어 있다고 가정하며, 작업장 1에서는 컨테이너가 도착하면 n개의 부품을 가공한 다음 밀어내기 방식에 따라 후속 작업장으로 컨테이너를 이동시킨다. 작업장 i에서 1개의 부품을 가공하는데 소요되는 시간은 평균 $\frac{1}{\mu_i}$ 인 지수분포를 따른다고 가정한다. 따라서 한 개의 컨테이너에 적재되어 있는 부품을 모두 가공하는데 소요되는 시간은 n-단계 일량분포가 된다. 완제품에 대한 수요는 도착률 λ_j 인 Poisson 과정에 따라 발생하며, 일회 수요의 크기는 확률변수 G로 표시된다. 수요의 최대 크기 m은 로트의 크기 n보다 작다고 가정하며($m \leq n$) 수요의 크기가 j일 확률은 $g_j(j=1, \dots, m)$ 로 표시된다. 완제품에 대한 수요가 발생하였을 때 충분한 수의 완제품 재고가 존재하면 모든 수요는 즉시 충족되며, 충분한 수의 완제품재고가 존재하지 않으면 일부 미충족 수요는 유실된다. 이외에 본 연구에서는 다음을 가정한다.

- 각 작업장에서의 기계고장은 발생하지 않는다.
- 각 작업장에서의 서비스 규칙은 선입선출방식(FIFO:First-In-First-Out)을 따른다.
- 각 작업장 및 완제품 저장창고의 크기는 충분히 커 붕괴가 발생하지 않는다.
- 컨테이너의 작업장간 이동시간은 무시할 수 있을 정도로 짧다고 가정한다.

본 연구에서는 이와 같은 CONWIP 시스템에서 각 작업장에 존재하는 부품수 및 컨테이너 수의 안정상태 확률을 구하

는 근사적 기법을 개발하고, 이를 이용하여 유실되는 수요의 비율, 각 작업장에서의 평균 재공품 수 등 시스템의 중요한 성능평가지수를 구하고자 한다.

본 연구에서 분석하고자 하는 CONWIP 시스템에서는 시스템 내에 존재하는 컨테이너 수가 K 로 고정되어 있으므로 컨테이너를 고객으로 간주하면 폐쇄형 대기네트워크로 모델링할 수 있다. 완제품 재고창고를 스테이션 J 라 하면, 폐쇄형 대기네트워크는 $M+1$ 개의 스테이션으로 구성된다(그림 1). 스테이션 J 에서 컨테이너의 이동이 이루어지기 위해서는 완제품이 적재된 컨테이너가 존재하고 컨테이너에 적재된 완제품의 수 이상의 규모를 가진 수요가 발생해야 하므로, 스테이션 J 는 두 가지 조건이 동시에 충족되어야만 컨테이너의 이동이 이루어질 수 있는 synchronization 스테이션임을 알 수 있다. J 가 synchronization 스테이션이므로 각 작업장에서의 로트 크기가 1인 단순한 경우에도 대기 네트워크는 승법형 조건을 만족하지 못한다. 따라서 CONWIP 시스템의 분석은 근사적 기법에 의존할 수밖에 없음을 알 수 있다.



〈그림 1〉 CONWIP 시스템

3. CONWIP 시스템 분석

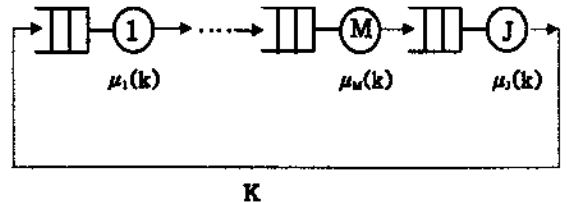
3.1 승법형근사법 개요

본 연구에서는 CONWIP 시스템의 분석을 위하여 Baynat와 Dallery[1]의 승법형근사법(product-form approximation method)을 사용하고자 한다. Baynat와 Dallery의 승법형근사법은 서비스 시간이 일반분포를 따르는 폐쇄형 대기네트워크를 분석하기 위한 기법으로 Marie의 기법[10]에 기초하고 있다. 승법형 근사법은 Mascolo의[12]에 의해 수요가 단순 Poisson 과정을 따르고 로트의 크기가 1인 CONWIP 시스템에 성공적으로 적용된 바 있다. 승법형근사법은 대기네트워크를 몇 개의 하부 네트워크로 분해한 후, 각각의 하부 네트워크를 서비스시간이

지수분포를 따르고 서비스율이 상태의존적인 단일서버로 치환함으로써 분석하고자 하는 폐쇄형 대기네트워크를 승법형 대기네트워크로 근사화 시키는 기법이다. 승법형근사법에서는 근사화된 승법형 대기네트워크를 동등 승법형 대기네트워크(equivalent product form network)라 부르며, 동등 승법형 대기네트워크를 구성하는 단일 서버의 상태종속 서비스율을 구하기 위하여 각각의 하부 네트워크를 상태종속 Poisson 도착 과정을 갖는 개방형 대기시스템으로 가정해 분석한다. 개방형 대기시스템의 상태종속 산출율(state dependent throughput)이 구해지면 이 값이 곧 해당 단일 서버의 상태종속 서비스율로 사용된다. 또한 승법형 대기네트워크의 서비스율이 구해지면 승법형 네트워크를 구성하는 각 서버의 상태종속 산출율(state dependent throughput)을 구할 수 있으며, 이 값은 다시 해당 하부 네트워크의 상태종속 도착율로 사용된다. 따라서 승법형 대기네트워크의 상태종속 서비스율이 구해지면 이로부터 하부 네트워크의 상태종속 도착율을 구할 수 있고, 하부 네트워크의 상태종속 도착율이 구해지면 이로부터 승법형 대기네트워크의 상태종속 서비스율을 구할 수 있게 된다. 그러나 승법형 대기네트워크의 상태종속 서비스율과 하부 네트워크의 상태종속 도착율은 모두 미지의 값이므로 이를 구하는 문제는 고정점 문제(fixed point problem)가 되며, 이를 구하기 위하여는 반복적기법을 사용하여야만 한다.

3.2 하부 네트워크로의 분해

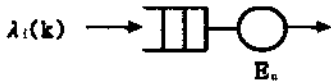
본 연구에서는 각 작업장을 각각 독립적인 하부 네트워크로 간주해 분해한다. 따라서 원 대기네트워크는 $M+1$ 개의 하부 대기네트워크로 분해된다. 이와같은 방식으로 분해가 이루어질 경우 얻게되는 동등 대기네트워크는 서버 i 의 상태종속 서비스율을 $\mu_i(k)$ 이라 할 때 〈그림 2〉와 같다.



〈그림 2〉 상태종속 서비스율을 갖는 지수분포 서버로 구성된 동등 대기네트워크

3.3 하부 네트워크의 분석

하부 네트워크 i 는 상태종속 도착율이 $\lambda_i(k)$ 이고 도착간격이 지수분포를 따르는 개방형 대기 네트워크로 간주해 분석한다. 그러면 하부 네트워크 J 를 제외한 각 하부 대기네트워크는 <그림 3>과 같은 개방형 대기시스템으로 표현될 수 있다.



$i = 1, 2, \dots, M, k = 0, 1, \dots, K.$

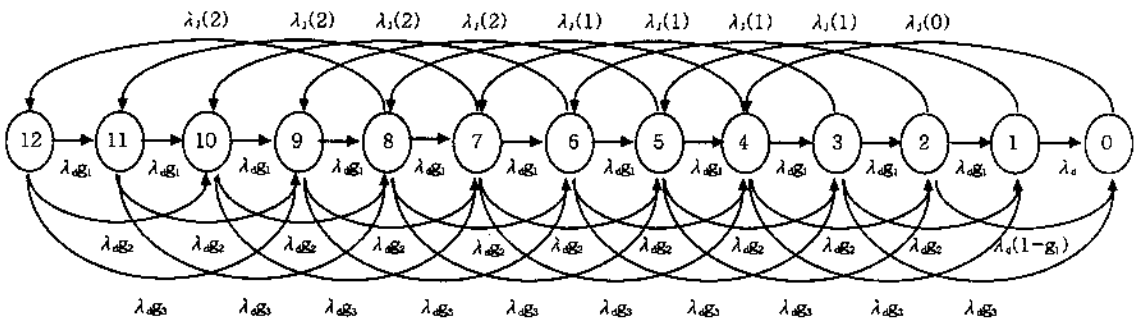
<그림 3> 하부 네트워크 $i(i \neq J)$

하부 네트워크 i 를 분석하기 위하여 우선 $\lambda_i(k)$ 를 알고 있다고 가정하자. 그러면 하부 네트워크 J 를 제외한 각 하부 네트워크는 시스템 내에 존재하는 고객의 수에 따라 도착률이 다른 M/E₁/1/K 대기행렬 시스템으로 분석이 가능하다. 본 연구에서는 상태종속 도착률을 갖는 M/E₁/1/K 대기행렬 시스템의 분석을 위해 Marie에 의해 제안된 기법[11]을 이용하였다. Marie의 기법에 의해 하부 네트워크 $i(i \neq J)$ 에 j 개의 컨테이너가 존재할 안정상태확률 $P_i(j)$ 가 계산되면, 하부 네트워크 i 에 존재하는 부품 수는 하부 네트워크 i 에 존재하는 컨테이너의 수에 n 을 곱한 값이므로 하부 네트워크 i 에 j_n 개의 부품이 존재할 확률은 $P_i(j)$ 가 된다.

하부 네트워크 J 는 synchronization 스테이션이므로 분석이 복잡하며 상태의 수가 유한한 연속시간형 마코브체인(Continuous Time Markov Chain)으로 모델링 할 수가 있다. 컨테이너의 수가 3이고 로트의 크기는 4이며, 일회 수주의 최대크기는 3인 CONWIP 시스템의 상태전이율 다이어그램은 <그림 4>

와 같다.

하부 네트워크 J 의 안정상태확률은 상태전이율 다이어그램으로부터 얻어지는 시스템 균형방정식(balance equation)을 풀면 구할 수 있다. 그러나 컨테이너의 수와 로트의 크기가 증가하면 상태의 수가 급속히 증가하여 연립 방정식을 푸는데 소요되는 노력도 급속히 증가한다. 예를 들어 컨테이너의 수가 3이고 로트의 크기가 4일 경우에는 연립방정식을 구성하는 식의 개수가 13개에 불과하나 컨테이너의 수가 30, 로트의 크기가 10으로 증가하면 연립방정식을 구성하는 식의 개수는 301개로 증가한다. 따라서 마코브체인의 특수한 구조를 이용하여 대규모 연립방정식을 효율적으로 풀기 위한 기법이 요구된다. 본 연구에서는 순환적기법(recursive algorithm)[2]을 이용하여 연립방정식을 효율적으로 풀고자 한다. 순환적기법은 모든 상태의 안정상태확률이 일부 경계상태(boundary state)의 안정상태확률의 1차결함으로 표시될 수 있을 때 적용 가능한 기법이다. 이 경우 경계상태의 안정상태확률만 구하면 모든 상태의 안정상태확률을 구할 수 있기 때문에 연립방정식을 구성하는 식의 개수는 경계상태의 수의 크기로 축소된다. 따라서 임의의 안정상태확률을 일차결함으로 표시해 줄 수 있는 경계상태가 존재하는 경우에 순환적기법의 적용이 가능하며, 경계상태의 수가 작을수록 순환적기법의 효율성은 증가한다. 하부 네트워크 J 는 마코브체인의 특수한 구조로부터 상태 $nK, nK-1, \dots, n(K-1)+1$ 등 n 개의 상태를 경계상태로 설정하면 모든 상태의 안정상태확률이 경계상태의 일차결함으로 표시 가능하다. 따라서 순환적 기법을 이용하면 연립방정식을 구성하는 방정식의 수가 $nK+1$ 에서 n 개로 줄어든다. 상태 j 의 안정상태확률을 경계상태의 안정상태확률의 선형결함으로 식(1)과 같이 표현하면 선형결합계수 $C_k(j)$ 는 $p_k=1$ 이라 놓고 나머지 경계상태의 안정상태확률은 0으로 놓은 후 이를 상태 $(n+j)$ 로부터 얻어지



<그림 4> 하부 네트워크 J 의 연속시간형 마코브 체인으로의 모델링 ($K=3, n=4, m=3$ 인 경우)

는 균형방정식에 대입함으로써 구할 수 있으며, 이러한 과정을 $j=n(K-1), n(K-1)-1, \dots, 1, 0$ 에 대해 순차적으로 수행한다.

$$p_j = C_{nK}(j)p_{nK} + C_{nK-1}(j)p_{nK-1} + \dots + C_{n(K-1)+1}(j)p_{n(K-1)+1}, \quad j=0, 1, 2, \dots, n(K-1) \quad (1)$$

$C_k(j)$ 가 구해지면 상태 $0, 1, \dots, n-2$ 으로부터 얻어지는 균형방정식과 정규화 방정식으로 구성된 연립방정식을 풀음으로써 경계상태의 안정상태확률을 구할 수 있으며, 경계상태의 안정상태확률이 구해지면 식 (1)으로부터 모든 상태의 안정상태확률을 구할 수 있다. 순환적기법에 대한 상세한 설명은 [2]에 주어져 있다.

Synchronization 스테이션에 존재하는 완제품 재고에 대한 안정상태확률 p_j 가 구하여 지면 synchronization 스테이션에 존재하는 컨테이너 수의 안정상태확률 $P_j(j)$ 는 다음 식에 의해 구하여 질 수 있다.

$$P_j(j) = \begin{cases} p_0, & j=0 \\ \sum_{i=(j-1)n+1}^j p_i, & j=1, 2, \dots, K \end{cases} \quad (2)$$

3.4 승법형근사법을 이용한 알고리즘

하부 네트워크 1, 2, 3, ..., M, J의 안정상태확률이 구해지면 하부 네트워크 i에 존재하는 컨테이너의 수가 k일 경우의 상태종속 산출율 $v_i(k)$ 의 값은 다음과 같이 표현된다[1].

$$v_i(k) = \begin{cases} 0 & k=0 \\ \lambda_i(k-1) \frac{P_i(k-1)}{P_i(k)} & k=1, 2, \dots, K \end{cases} \quad (3)$$

하부 네트워크 i의 상태종속 산출율 $v_i(k)$ 의 값이 구해지면 동등 승법형 대기네트워크를 구성하는 서버 i의 상태종속 서비스율의 값, $\mu_i(k)$, 은 다음과 같이 구해진다.

$$\mu_i(k) = v_i(k), \quad k=0, 1, \dots, K, \quad i=1, 2, \dots, M, J. \quad (4)$$

동등 승법형 대기네트워크의 서비스율이 구해지면 승법형 대기네트워크의 분석에 사용될 수 있는 임의의 알고리즘을 이용하여 서버 i의 상태종속 산출율, $X_i(k)$,을 구할 수 있다. Convolution 알고리즘을 이용하면 서버 i의 상태종속 산출율은 다음식에 의해 구할 수 있다.

$$X_i(k) = \begin{cases} \frac{e_i G_{M_i}(K-k)}{G_{M_i}(K-k)} & k=0, 1, \dots, K-1 \\ 0 & k=K \end{cases} \quad (5)$$

여기서 e_i 는 서버 i의 상대적 방문율이고, G_{M_i} 는 M+1개의 스테이션중 스테이션 i를 제외하고 계산한 정규화 상수이다.

동등 승법형 대기네트워크를 구성하는 서버 i의 상태종속 산출율이 구해지면 하부 네트워크 i의 상태종속 도착율은 다음과 같이 구해진다.

$$\lambda_i(k) = X_i(k), \quad k=0, 1, \dots, K, \quad i=1, 2, \dots, M, J. \quad (6)$$

이상과 같이 하부 네트워크의 안정상태확률이 구하여 지면, 하부 네트워크의 상태종속 산출율을 식(3)으로부터 계산할 수 있고, 이 값은 곧 동등 대기네트워크의 상태종속 서비스율로 사용된다. 또한 동등 대기네트워크의 상태종속 서비스율이 구해지면 식 (5)로부터 동등 대기네트워크의 상태종속 산출율이 계산되어 질 수 있고, 이 값은 곧 하부 네트워크의 상태종속 도착율로 사용될 수 있다. 승법형근사법에서는 이와 같은 과정을 해가 수렴할 때까지 반복 수행한다. 이상의 과정을 정리하면 다음과 같다.

승법형근사법을 이용한 알고리즘

step 0(초기화과정)

$$\mu_i(k) = \frac{\mu_i}{n}, \quad i=1, \dots, M, \quad \mu_j(k) = \lambda_j, \quad k=1, 2, \dots, K$$

step 1(반복과정)

i) $\lambda_i(k)$ 의 계산 : 식(5), (6) 이용, $i=1, 2, \dots, M, J$, $k=0, 1, 2, \dots, K$.

$P_i(k)$ 을 계산 : 3.3절의 분석기법 이용

$i=1, 2, \dots, M$ 이면, M/E/1/K 분석기법 이용

$i=J$ 이면, 식 (1),(2)의 순환적기법 이용

ii) $\mu_i(k)$ 의 계산 : 식(3), (4) 이용, $i=1, 2, \dots, M, J$, $k=0, 1, 2, \dots, K$.

수렴이 이루어 졌으면 step 2로 가고, 수렴이 이루어지지 않았으면 step 1로 돌아감.

step 2(성능평가지수 계산과정)

구하고자 하는 성능평가지수를 3.5절의 식을 이용하여 계산

3.5 시스템의 성능평가지수 계산

알고리즘의 수렴이 이루어지면 알고리즘에 의해 가장 최근 에 계산된 각 하부 네트워크의 안정상태확률을 이용하여 각 작업장에 존재하는 부품 수의 안정상태확률, 콘테이너 수의 안정상태확률 등이 구해 질 수 있으며, 이로부터 각 작업장에 존재하는 평균 재공품 수의 안정상태확률을 구할 수 있다. 또한 작업장 i 의 기계의 이용률 U_i 는 다음과 같이 구하여 질 수 있다.

$$U_i = 1 - P_i(0) \quad (7)$$

수요가 도착한 순간 충분한 재고가 없어 수요가 유실될 확률은 CONWIP 시스템의 생산률, X 의 식으로 다음과 같이 표현될 수 있다.

$$P_L = 1 - \frac{Xn}{\lambda_d E(G)} \quad (8)$$

여기서 G 는 1회 수요의 크기를 나타내고, $X = e^{-\lambda} \frac{G_M(K-1)}{G_M(K)}$.

3.6 포화 수요를 갖는 CONWIP 시스템의 분석

경우에 따라서는 완제품에 대한 수요가 항상 존재하여 (saturated demands) 완제품이 생산되는 즉시 인출될 수가 있다. 이와같은 경우에는 컨테이너가 synchronization 스테이션을 거치지 않고 작업장 M 에서 작업장 1으로 직접 이동되므로 그림 1에서 synchronization 스테이션이 제외된 형태의 폐쇄형 대기 네트워크로 모델링이 가능하다. 이 경우에는 시스템의 가장 중요한 성능평가지수는 생산률 X 가 되며, 3.4절에 설명된 알고리즘을 이용하여 생산률 X 와 기타 성능평가지수를 구할 수 있다.

4. 근사적 분석의 정확도 평가

본 연구에서 제안한 알고리즘의 성능을 평가하기 위하여 예제를 통하여 많은 실험을 수행하였다. 실험에는 IBM PC 586이 이용되었으며, 수치분석에는 Matlab 4.2가 이용되었다. 알

고리즘의 정확도를 평가하기 위하여 알고리즘에서 얻어진 값과 시뮬레이션에서 얻어진 값을 비교하였으며, 시뮬레이션은 시스템에 도착한 수요의 수가 300,000에 도달될 때까지 수행하였다. 알고리즘의 종료조건은 10^{-5} 이하로 하였으며, 실험 결과 모든 예제에서 알고리즘이 수렴함을 관측할 수 있었다. 본 절에서는 많은 실험예제 중 6개의 대표적인 예제를 선정하여 소개하도록 한다.

첫 번째 예제는 작업장의 수는 3이고 콘테이너의 수는 5이며, 각 작업장에서의 가공시간은 평균 2인 지수분포를 따르는 경우이다. 각 작업장에서의 로트의 크기는 3이고, 수요의 크기는 1에서 3까지의 값을 취할 수 있는 일양분포를 따른다고 가정하였다. 또한 수요의 도착율을 0.1에서 1.0까지 0.1씩 증가시키며 10 가지 서로 다른 경우에 대하여 실험하였다(표 1). 두 번째 예제는 작업장의 수가 5이고 콘테이너의 수는 10이며, 각 작업장에서의 가공시간은 평균 2인 지수분포를 따르는 경우이다. 각 작업장에서의 로트의 크기 및 수요의 크기는 예제 1에서와 동일하다. 예제 2에서는 수요의 도착율을 0.2에서 2.0까지 0.2씩 증가시키며 10 가지 서로 다른 경우에 대하여 실험하였다(표 2). 세 번째 예제와 네 번째 예제는 작업장의 수가 큰 대규모 CONWIP 시스템의 경우로 작업장의 수가 10이라고 가정하였다. 세 번째 예제에서는 로트의 크기를 2라 가정하였고 네 번째 예제에서는 로트의 크기를 4라 가정하였다. 예제 3과 4에서도 수요의 도착율을 0.2에서 2.0까지 0.2씩 증가시키며 10 가지 서로 다른 경우에 대하여 실험하였다(표 3, 표 4). 각 표에는 완제품창고에 존재하는 평균 완제품의 수 (\overline{FP}), 완제품창고에 존재하는 평균 콘테이너의 수(\overline{K}), 작업장 M 에 존재하는 평균 콘테이너의 수(\overline{K}_M) 및 유실되는 수요의 비율(P_L) 등의 성능평가지수에 대해 근사적기법에 의해 구해진 값과 시뮬레이션에 의해 구해진 값이 정리되어 있다. 다섯 번째 예제와 여섯 번째 예제는 완제품에 대한 수요가 항상 존재하는 포화 수요의 경우로서 로트의 크기를 1에서 10까지 1씩 증가시키며 10가지 서로 다른 경우에 대해 근사적기법에 의해 얻어진 생산률의 값과 시뮬레이션에 의해 얻어진 생산률의 값을 비교하였다(표 5).

표에서 볼 수 있듯이 본 연구에서 제안된 근사적기법은 각 작업장에 존재하는 평균 콘테이너의 수와 평균 부품의 수에 대해서는 일반적으로 만족할 만한 수준의 정확한 해를 산출해 준다. 그러나 근사적기법에 의해 얻어진 수요가 유실될 확률의 추정치는 정확도가 떨어지는 것으로 관측되었다. 또한 수요가 유실될 확률은 근사적기법에 의해 얻어지는 값이 시뮬레

〈표 1〉 예제 1의 결과

	λ_d	Simulation				Algorithm			
		\overline{FP}	\overline{K}_J	P_L	\overline{K}_s	\overline{FP}	\overline{K}_J	P_L	\overline{K}_s
M=3 K=5 n=3 $g_1=g_2=g_3=1/3$ $\mu=(2,2,2,2,2)$	0.1	13.0533	4.6865	0.0000	0.1039	13.0419	4.6806	0.0001	0.1065
	0.2	12.0012	4.3340	0.0008	0.2197	11.9394	4.3124	0.0014	0.2292
	0.3	10.8419	3.9449	0.0061	0.3462	10.6773	3.8884	0.0089	0.3705
	0.4	9.5637	3.5135	0.0201	0.4866	9.3121	3.4249	0.0291	0.5250
	0.5	8.2925	3.0792	0.0459	0.6241	7.9640	2.9611	0.0648	0.6796
	0.6	7.1040	2.6682	0.0825	0.7563	6.7464	2.5365	0.1131	0.8211
	0.7	5.9902	2.2757	0.1286	0.8854	5.7159	2.1724	0.1680	0.9425
	0.8	5.0782	1.9526	0.1830	0.9842	4.8754	1.8716	0.2241	1.0428
	0.9	4.3258	1.6845	0.2342	1.0792	4.2002	1.6271	0.2780	1.1243
	1.0	3.7567	1.4768	0.2804	1.1440	3.6590	1.4291	0.3279	1.1903

〈표 2〉 예제 2의 결과

	λ_d	Simulation				Algorithm			
		\overline{FP}	\overline{K}_J	P_L	\overline{K}_s	\overline{FP}	\overline{K}_J	P_L	\overline{K}_s
M=5 K=10 n=3 $g_1=g_2=g_3=1/3$ $\mu=(2,2,2,2,2)$	0.2	25.6919	8.8976	0.0000	0.2177	25.5393	8.8464	0.0000	0.2307
	0.4	21.3902	7.4630	0.0006	0.5076	20.5254	7.1744	0.0017	0.5651
	0.6	15.4032	5.4607	0.0181	0.9095	13.6270	4.8595	0.0402	1.0281
	0.8	9.3557	3.4137	0.0992	1.3117	8.1432	2.9887	0.1531	1.4023
	1.0	5.5217	2.0892	0.2176	1.5721	5.2264	1.9709	0.2780	1.6058
	1.2	3.6206	1.4147	0.3269	1.6988	3.6984	1.4253	0.3812	1.7149
	1.4	2.6771	1.0689	0.4138	1.7628	2.8175	1.1041	0.4617	1.7792
	1.6	2.0960	0.8526	0.4828	1.7985	2.2601	0.8972	0.5249	1.8206
	1.8	1.7302	0.7146	0.5370	1.8743	1.8807	0.7542	0.5752	1.8492
	2.0	1.4605	0.6077	0.5832	1.8615	1.6076	0.6499	0.6162	1.8700

〈표 3〉 예제 3의 결과

	λ_d	Simulation				Algorithm			
		\overline{FP}	\overline{K}_J	P_L	\overline{K}_{10}	\overline{FP}	\overline{K}_J	P_L	\overline{K}_{10}
M=10 K=10 n=2 $g_1=g_2=1/2$ $\mu=(3,3,3,3,3,3,3,3,3,3)$	0.2	17.3815	8.9403	0.0000	0.1059	17.3419	8.9210	0.0000	0.1079
	0.4	14.9891	7.7449	0.0002	0.2242	14.7792	7.6394	0.0003	0.2361
	0.6	12.2232	6.3607	0.0047	0.3598	11.7452	6.1199	0.0080	0.3880
	0.8	9.2544	4.8667	0.0282	0.5133	8.6380	4.5544	0.0449	0.5446
	1.0	6.7507	3.5982	0.0819	0.6401	6.1596	3.2933	0.1166	0.6707
	1.2	4.8782	2.6386	0.1607	0.7301	4.4866	2.4320	0.2020	0.7568
	1.4	3.6298	1.9928	0.2373	0.8021	3.4061	1.8691	0.2837	0.8131
	1.6	2.8072	1.5606	0.3115	0.8388	2.6942	1.4938	0.3553	0.8506
	1.8	2.2626	1.2712	0.3708	0.8639	2.2060	1.2337	0.4163	0.8766
	2.0	1.8798	1.0660	0.4277	0.8887	1.8569	1.0459	0.4679	0.8954

〈표 4〉 예제 4의 결과

	λ_d	Simulation				Algorithm			
		\overline{FP}	\overline{K}_j	P_L	\overline{K}_{10}	\overline{FP}	\overline{K}_j	P_L	\overline{K}_{10}
M=10 K=10 n=4 g=g ₀ =1/2 $\mu=(3,3,3,3,3,3,3,3,3,3)$	0.2	34.4650	8.9909	0.0000	0.1010	34.2896	8.9474	0.0000	0.1053
	0.4	30.1902	7.9218	0.0000	0.2088	29.4314	7.7329	0.0000	0.2267
	0.6	25.5727	6.7692	0.0001	0.3247	23.6741	6.2929	0.0015	0.3707
	0.8	20.2754	5.4413	0.0021	0.4567	17.2807	4.6872	0.0204	0.5313
	1.0	14.8725	4.0863	0.0190	0.5879	11.8400	3.3043	0.0791	0.6696
	1.2	10.1490	2.8868	0.0653	0.7106	8.2427	2.3728	0.1630	0.7627
	1.4	6.9690	2.0636	0.1383	0.7894	6.0422	1.7903	0.2479	0.8210
	1.6	5.0854	1.5603	0.2185	0.8423	4.6640	1.4171	0.3234	0.8583
	1.8	3.9011	1.2406	0.2857	0.8640	3.7547	1.1654	0.3879	0.8835
	2.0	3.1565	1.0310	0.3447	0.8817	3.1230	0.9869	0.4425	0.9013

〈표 5〉 예제 5, 예제 6의 결과(포화수요를 갖는 CONWIP 시스템의 생산율 비교)

M=3, K=5, $\mu=(2,2,2)$			M=10, K=10, $\mu=(3,3,3,3,3,3,3,3,3,3)$		
n	Throughput (Simulation)	Throughput (Algorithm)	n	Throughput (Simulation)	Throughput (Algorithm)
1	1.4284	1.4286	1	1.5798	1.5789
2	0.7984	0.7706	2	0.9225	0.8487
3	0.5615	0.5293	3	0.6641	0.5819
4	0.4360	0.4034	4	0.5226	0.4431
5	0.3559	0.3260	5	0.4327	0.3578
6	0.3007	0.2735	6	0.3696	0.3001
7	0.2605	0.2356	7	0.3233	0.2584
8	0.2305	0.2070	8	0.2878	0.2269
9	0.2064	0.1845	9	0.2594	0.2023
10	0.1866	0.1665	10	0.2362	0.1825

이전에 의해 얻어진 값에 비하여 항상 큰 것으로 관측되었다. 표 1-5로부터 알 수 있듯이 일반적으로 근사적기법의 정확도는 작업장의 수가 많아지고 로트의 크기가 커짐에 따라 저하되는 것으로 관측되었다. 따라서 본 연구에서 제안된 기법은 작업장의 수가 5미만으로 비교적 작고 로트의 크기가 3미만으로 작을 경우에 특히 유효할 것으로 판단된다. 그러나 작업장의 수가 커지거나 로트의 크기가 큰 경우에도 정교한 성능평가 지수를 요구하지 않는 설계의 초기 단계에서는 근사적기법의 사용이 부분적으로 가능하리라 판단된다. 특히 본 연구에서 제안한 분석기법은 수요가 compound Poisson 과정에 따라 발생하고 각 작업장에서의 가공이 로트 단위로 이루어지는 CONWIP 시스템을 분석할 수 있는 특별한 대안이 존재하지 않는 상황에서 하나의 유효한 수리적 기법이 될 수 있으므로

오차의 범위가 지나치게 크지 않는 한 시뮬레이션을 대신할 수 있는 유용한 기법이 될 것으로 기대한다.

5. 결 론

본 연구에서는 완제품에 대한 수요가 compound Poisson 과정을 따르고 각 작업장에서의 가공이 로트단위로 이루어지는 경우의 CONWIP 시스템에 대한 근사적분석을 수행하였다. 제안된 CONWIP 시스템을 분석하기 위하여 본 연구에서는 각각의 CONWIP 시스템을 synchronization 스테이션이 존재하는 폐쇄형 대기행렬 네트워크로 모델링하고 이를 승법형근사법을 이용하여 분석하였다. 허부 네트워크의 분석에는 Marie가 제안한 M/E₁/1/K 대기행렬 시스템의 분석기법과 순환적기법을

이용하였다. 본 연구에서 제안된 근사적기법은 각 작업장에 존재하는 부품수의 안정상태확률, 평균 재공품 수, 기계의 이용률 및 유실되는 수요의 비율 등 CONWIP 시스템의 중요한 성능평가지수를 비교적 정확하고 신속하게 계산하여 준다.

[참 고 문 헌]

[1] Baynat, B. and Y. Dallery, "A Unified View of Product-form Approximation Techniques for General Closed Queueing Networks," Performance Evaluation Vol.18, pp.205-224, 1993.

[2] Buzacott, J. A. and D. Kostelski, "Matrix-geometric and Recursive Algorithm Solution of a Two-stage Unreliable Flow Line," IIE Transactions, Vol 19, pp.429-438, 1987.

[3] Duenyao, I. and W. J. Hopp, "CONWIP Assembly System with Deterministic Processing and Random Outages," IIE Transactions, Vol. 24. No.4, pp.97-111, 1992.

[4] Duenyas, I. and W.J.Hopp, "Estimating the Throughput of an Exponential CONWIP Assembly System," Queueing Systems, Vol.14, pp.135-157, 1993.

[5] Duenyas, I., W.J.Hopp, and M.L. Spearman, "Characterizing the Output Process of a CONWIP Line with Deterministic Processing and Random Outages," Mgmt. Sci., Vol.39, pp.975-988, 1993.

[6] Hall, K.W. "Synchro MRP: Combined Kanban with an MRP, the Yamaha PTMAC System-production Planning and Control in Japan," American Production and Inventory Control Society, pp.43-56, 1986.

[7] Karnmarkar, U.S. "Push, Pull and Hybrid Control Schemes," Working Paper QM8614, Graduate school of Management, Univ. of Rochester, 1986.

[8] Kimura, O. and H. Terada, "Design and Analysis of Pull Systems; a Method of Multistage Production Control," Int. J.Prod.Res. Vol.19, pp 241-253, 1981.

[9] Kragewsky, L.J. and L.P. King, "Kanban, MRP, and Shaping the Manufacturing Environment," Mgmt. Sci. Vol. 33, pp.39-57,1987.

[10] Marie, R.A. "An Approximate Analytical Method for General Queueing Networks," IEEE Trans. Soft. Eng. SE-5, pp.530-538, 1979.

[11] Marie, R.A. "Calculating Equilibrium Probabilities for $\lambda(n)C_k/1/N$ Queues", Performance Evaluation Review, Vol. 9 pp.117-125, 1980,

[12] Mascolo, M.D, Y. Frein, and Y. Dallery, "An Analytical Method for Performance Evaluation of kanban Controlled Production Systems," Opns. Res., Vol.44, pp.50-64, 1996.

[13] Spearman, M.L. "Push and Pull Production System: Issues and Comparisons," Opns. Res. Vol.40, pp.521-532, 1992.

[14] Spearman, M.L. "Customer Service in Pull Production System," Opns. Res. Vol.40, pp.948-958, 1992.

[15] Spearman, M.L., D.L. Woodruff, and W. J. Hopp, "CONWIP : A Pull Alternative to Kanban," Int. J. Prod. Res. Vol. 28, pp.878-894, 1990.



이호성
1978년 서울대학교 산업공학과 학사
1980년 한국과학기술원 산업공학과 석사
1988년 미국 University of Michigan 산업공학 박사
현재 경희대학교 산업공학과 교수
관심분야 대기행렬이론 및 응용, 추계적모형, 신뢰도공학 등



이정은
1995년 경희대학교 산업공학과 학사
현재 경희대학교 산업공학과 석사과정 재학
관심분야 Optimization, Simulation, 인공지능.

98년 5월 최초접수, 98년 7월 최종수정