

Linearizer 解法을 이용한 柔軟生産시스템의 分析**

(An Analysis of Flexible Manufacturing Systems Using the Linearizer Algorithm)

朴 用 集*
金 聖 植*

ABSTRACT

In the design and operation of Flexible Manufacturing Systems (FMSs), it is useful to have a tool with which one can evaluate system's performances under various conditions.

As one of such tools, this study chooses the Linearizer algorithm which uses mean value analysis (MVA). Linearizer algorithm is known to be the most accurate one among the MVA based heuristic algorithms yet developed.

Though, up until now, the method is solely used to evaluate computer system's performances, this study shows that the Linearizer algorithm can be adapted for FMSs and still produces accurate solutions.

Numerical examples for various classes of FMSs are presented and compared with the results from other methods and simulations.

1. 序 論

유연생산시스템(Flexible Manufacturing System : FMS)에서의 자재의 흐름을 분석하기 위한 수학적 모형에 관한 많은 연구가 이루어지고 있다. 이들중 많은 연구들이 시스템의 성능평가를 위하여 폐쇄형 대기행렬네트워크(Closed Network of Queues) 이론을 사용하고 있다. FMS를 CNQ로 가정할 수 있는 이유는 시스템내에는 가용한 파렛트(Pallet)의 수가 한정되어 있기 때문에 모든 작업을 마친 작업물이 적재/하역장(loading / unloading station)으로 운반되어 파렛트로부터 하역되면 바로 새로운 작업물이 그 파렛트에 적재되어 시스템내로 들어오게 되므로 시스템내에는 항상 일정한 수의

파렛트를 유지하게 된다.

위와 같이 시스템을 CNQ로 볼 수 있는 시스템으로는 컴퓨터시스템이나 통신망들을 들 수 있고 FMS의 작업물을 컴퓨터시스템의 Job으로 보게 되면 두 시스템의 유사성은 명확해진다.

CNQ의 실제 시스템에서의 응용은 컴퓨터시스템의 성능평가에서 시작되었으며 현재도 이에 대한 연구들이 활발히 진행되고 있다.

Gordon & Newell이 CNQ의 이론을 소개한 이래 이에 바탕을 둔 유용한 해법들이 많이 개발되고 있다. CNQ에서의 시스템의 성능을 평가할 수 있는 방법중 대표적인 것으로 Convolution Method와 평균치분석(Mean Value Analysis : MVA)을 들 수 있다.

*高麗大學校 工科大學 産業工學科

** 본연구는 일부 과학재단의 지원을 받아 수행되었음.

Convolution Method는 승법형해 (product-form solution)로부터 정상화계수 $G(N)$ 을 구하여 성능척도들을 구하는 방법인데 단일종류의 작업물의 경우 계산에 있어서는 효율적이지만 여러 종류의 작업물 (multi-class customer)에 있어서는 계산량이 급격히 증가하여 계산이 용이하지 않다. Solberg[16]는 Convolution Method를 FMS에 적용시켜 CAN-Q라는 프로그램을 만들었으나 위에서 언급한 문제점을 내포하고 있다.

MVA (Mean Value Analysis)는 기존의 해법들에 비해 여러 종류의 작업물을 다루기가 용이하고 규모가 큰 시스템의 성능척도를 정확히 구할 수 있다. 이러한 MVA의 기본식은 시스템에 도착한 고객이 보는 대기행렬의 분포는 관찰자 관점에서 도착고객을 제외하고 본 대기행렬의 분포와 같다는 도착정리 (arrival theorem)로부터 구해진다.

MVA의 정확한 해법 (exact solution)은 시스템의 규모가 커지면 기억용량 등의 제약으로 인하여 계산이 곤란하다. 이러한 이유로 발견적해법 (heuristic algorithm)이 제시되었고 Bard [1, 2], Schweitzer [15], Chandy 등 [8]에 의해 개발되어 컴퓨터시스템의 분석에 이용되고 있다. 이와 같은 MVA의 발견적 해법들은 FMS의 분석에도 적용되어 Cavaille 등 [5], Hildebrandt [9], Suri 등 [10]의 연구가 발표되었고 Schweitzer 등 [14]은 작업물의 우선 순위가 고려될 FMS에 이 방법을 사용하였다.

본 연구에서는 컴퓨터시스템의 분석에 이용된 Chandy 등 [8]의 Linearizer 해법을 FMS의 분석에 적용한다. Linearizer 해법은 Bard-Schweitzer의 발견적 해법을 변형하여 정확한 MVA의 해법과 비교할 때 오차가 가장 작은 방법이다.

본 연구에서는 Linearizer 해법을 각 작업장마다 서버가 하나인 경우에 작업시간의 분포가 지수분포와 상수로 주어질 때에 적용하고 또한 작업장이 다중서버 (parallel server)를 갖고 다수의 운송시스템이 있는 시스템에 적용하여 기존의 결과에 비해 보다 정확한 FMS의 성능

을 평가하려 한다.

2. MVA-Linearizer 해법

N_r 을 시스템내에 있는 r 종류 고객의 수라고 모집단 벡터 $N = (N_1, N_2, \dots, N_r, \dots, N_n)$ 과 같이 정의한다. S_{ir} 은 서비스센터 i 에서 r 종류 고객이 받는 서비스시간, V_{ir} 은 고객 r 이 센터 i 를 방문하는 평균횟수, $X_r(N)$ 은 고객 r 의 시스템 출력률 (throughput), $L_{ir}(N)$ 은 센터 i 에서 대기하고 있는 r 종류 고객들의 평균수를 나타낸다. $(N - I_r)$ 은 벡터 N 으로 표시되는 고객을 갖는 시스템의 상태보다 r 종류 고객이 하나 적은 시스템의 상태를 표시하는데 I_r 은 r 번째 요소만 1이고 나머지는 0인 R 차원 방향벡터이다.

센터 i 가 단일서버를 갖는 경우 고객 r 의 평균대기시간 $W_{ir}(N)$ 은,

$$W_{ir}(N) = S_{ir} \left(1 + \sum_{j=1}^R L_{ij}(N - I_r) \right) \quad (1)$$

로 표시되고 시스템의 출력률과 평균대기 길이는 Little's law를 이용하여,

$$X_r(N) = N_r / \sum_{i=1}^R V_{ir} W_{ir}(N) \quad (2)$$

$$L_{ir}(N) = X_r(N) \cdot V_{ir} \cdot W_{ir}(N) \quad (3)$$

으로 나타낸다. 정확한 해법의 MVA는 $N = (0, 0, \dots, 0)$ 인 상태에서부터 출발하여 (1)식에서 $W_{ir}(N)$ 을 구한 후 (2)식에서 출력률을 계산하고 (3)식에서 각 작업장에서의 평균 대기길이를 구하여 다시 (1)식에 대입하여 반복하는 계산절차를 N 이 목표로 하는 시스템의 고객의 수를 표시하는 상태벡터가 될 때까지 계속한다. 위와 같은 과정을 통하여 우리는 정확한 시스템의 성능척도를 구할 수 있으나 앞에서 언급했듯이 시스템의 규모가 커지면 컴퓨터의 기억용량으로 인한 제약을 받는다. 이러한 문제점을 보완하기 위해 Reiser [13]에 의하여 발견적 해법이 제시되었고 이 해법은 FMS 분석에 적용되기도 하였다. [5]

Bard [1]와 Schweitzer [15]는 Reiser 와는 다른 방법으로 발견적 해법을 제시하였다. 이 해법에서는 $(N - I_r)$ 의 값들을 N 상태에서의 정보

를 이용하여 추정하고 이 추정값들을 기초로 다시 N 상태의 값들을 계산하게 된다. 이러한 반복과정(N→N-I_r→N)을 통하여 결과의 정확성을 향상시켜 주어진 기준에 도달하게 되면 계산은 종료된다. 이들의 해법에서는 L_v(N-I_r) 값을 구하기 위해,

$$L_v(N-I_r) = \begin{cases} L_v(N) & j \neq r \text{ 일 때} \\ \frac{N_j-1}{N_j} L_v(N) & j = r \text{ 일 때} \end{cases} \quad (4)$$

의 관계식을 이용한다.

Chandy 등[8]의 Linearizer 해법은 앞의 Bard-Schweitzer 해법을 변형하여 사용하게 되는데 Bard-Schweitzer 해법을 과정의 일부로 사용하기 때문에 계산량은 다소 많아지게 되지만, 정확도가 증가하여 MVA의 정확한 해법과 매우 근사한 결과를 가져온다. 이 해법에서는 L_v(N-I_r) 값을 추정하기 위해 다음과 같은 양들을 정의하여 사용한다.

$$f_v(N) = L_v(N) / N_r \quad (5)$$

$$d_{ir,j}(N) = f_v(N-I_r) - f_v(N) \quad (6)$$

이 두 식으로부터,

$$L_v(N-I_r) = (N-I_r)_j (f_v(N) + d_{ir,j}(N) + d_{ir,j}(N)) \quad (7)$$

여기서, $(N-I_r)_j = \begin{cases} N_j - 1 & r = j \text{ 일 때} \\ N_j & r \neq j \text{ 일 때} \end{cases}$

(7)식에서 d_{ir,j}(N)을 0으로 놓게 되면, (4)식이 되어 Bard-Schweitzer 해법의 식과 같게 된다. "Linearizer"란 의미는 이 해법이 d_{ir,j}(N-I_r) = d_{ir,j}(N)을 가정하는데서 연유한다. 즉 (5)식의 f_v(N)이 N의 선형함수임을 뜻하는 것이다. Linearizer 해법의 절차는 다음과 같다. 먼저 L_v(N)과 L_v(N-I_r)의 초기값을 각 작업장에서의 대기행렬의 길이는 같다는 가정에서 구한다. d_{ir,j}(N) 값을 0으로 놓고 (4)식대신 (7)식을 사용하는 Bard-Schweitzer 해법의 절차를 거쳐 N 상태의 성능척도들을 구하고, (N-I_r)에 대해서 고객의 수 R만큼 Bard-Schweitzer 해법을 적용하여 성능척도를 구해준다. 이 때 입력값으로 사용되는 d_{ir,j}(N-I_r) 값을 위의 Linearizer의 가정에서 d_{ir,j}(N)과 같다고 하였으므로 d_{ir,j}(N) 값을 사용한다. 다음에 (5)식으로부터 구한

f_v(N)과 f_v(N-I_r) 값으로부터 (6)식을 이용하여 d_{ir,j}(N) 값을 구해준다. 이러한 전체 계산과정을 2회 반복후 계산을 종료한다.

Chandy 등 [8]은 Linearizer 해법을 단일서버를 갖는 CNQ에 적용할 경우 Bard-Schweitzer의 해법에 비해 1.54(R+1)배의 비용이 들게 되지만, 오차의 한계 (error tolerance)는 향상됨을 보였다.

3. 대상 FMS모델의 구성

본 연구에서는 가능한 한 일반적인 형태의 FMS를 연구모델의 대상으로 삼기 위하여 다음과 같은 모델로 가정한다.

생산시스템은 M개(≥1)의 작업장과 운송시스템 그리고 작업장 앞에 재공품을 위한 저장소(local buffer)로 구성되어 있다. 저장소는 blocking 현상을 막기에 충분히 크다고 가정한다.

각 작업장은 하나 또는 다중의 서버를 가질 수 있고 가공될 작업물의 종류는 R(≥1)가지이고 각 종류마다 할당된 파렛트의 수는 한정되어 있다. 각 작업물들은 적재/하역장에서 파렛트/치구에 부착되어 공정순서대로 작업장을 방문하게 되고 작업물의 가공시간은 작업장에 따라 주어진 분포를 따르게 된다.

운송시스템은 컨베이어나 복수의 AGV를 사용하게 되는데 운송시스템은 작업물들을 작업이 끝날때마다, 다음 작업장으로 운반하여 주기 때문에 컴퓨터시스템의 CPU와 같은 중앙서버(central server)의 역할을 하게 된다. 각 작업장간의 평균이송시간(mean transport time)은 작업장간의 거리에 따라 다르게 되지만 분석의 편의를 위하여 전체거리의 평균을 운송시스템의 속도로 나누어 동일한 값을 갖는다고 가정한다. 운송시스템은 작업장과 마찬가지로 (M+1)번째 작업장으로 생각할 수 있는데 Schweitzer 등 [14]은 운송시스템의 출력률을 작업장에서의 대기시간에 운송시스템의 대기시간을 더하여 구하였지만, 과정이 복잡하므로 본 연구에서는 모든 작업물들은 운송시스템을 거쳐야

한다는 가정하에 출력률을 다음과 같이 나타낸다.

$$X_{M+1,r} = \sum_{i=1}^M X_{i,r} = \sum_{i=1}^M V_{i,r} \cdot X_r$$

$$V_{M+1,r} = \sum_{i=1}^M V_{i,r} \text{ 이라 하면,}$$

$$X_{M+1,r} = V_{M+1,r} X_r \quad (8)$$

로 표시할 수 있다. 컨베이어의 출력률도 마찬가지로 (8)식을 사용하는데 컨베이어는 충분한 서버(infinite server)를 갖는다고 가정할 수 있으므로 대기시간의 식은 다음과 같이 표시한다.

$$W_{M+1,r} = S_{M+1,r} \quad (9)$$

즉, 작업으로 인한 지연(delay)만 생기고 대기행렬로 인해 기다리는 시간은 발생하지 않는다. 그림 1은 지금까지 설명한 시스템의 모델을 나타낸다.

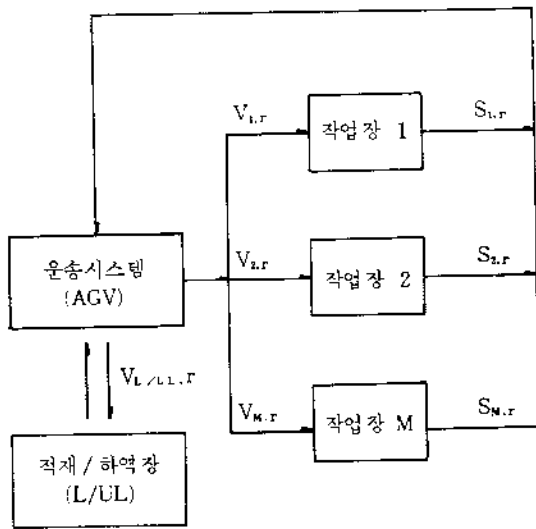


그림 1. FMS의 개략적 모델

4. Linearizer 해법의 확장

본 연구에서는 이러한 FMS에서 각 작업장이 단일서버인 경우와 다중서버와 다수의 운송 시스템을 갖는 경우 Linearizer 해법을 적용한다.

4.1. 작업장이 단일서버를 갖는 경우

앞의 2절에서는 모든 작업물들의 대기행렬의 길이($L_{i,r}(N-I_r)$)는 종류에 관계없이 작업시간이 $S_{i,r}$ 로 같다고 가정하였으나 실제 시스템에서는 작업물의 대기행렬은 종류에 따라 서로 다른(class-dependent) 작업시간에 의해 가공된다. 또한 FMS와 같은 생산시스템에서는 작업시간의 분포가 확률적(stochastic)이 아닌 확정적(deterministic)인 상수값을 갖는 경우가 많게 된다.

작업물 r 이 작업장 i 에 도착하였을 때 기다리게 되는 시간은 다음과 같은 3가지 부분의 합으로 나타낸다.

1) 도착한 작업물의 평균 작업시간: $S_{i,r}$ (10)

2) 작업물이 도착하였을 때 미리 와서 기다리고 있던 작업물들의 평균 대기시간(작업중인 작업물은 제외)

$$\sum_{i=1}^R S_{i,r} (L_{i,r}(N-I_r) - U_{i,r}(N-I_r)) \quad (11)$$

3) 작업물이 도착했을 때 작업중인 작업물

$$\text{의 남은 평균 작업시간}$$

$$\sum_{i=1}^R U_{i,r}(N-I_r) S_{i,r} \frac{1+C_{i,r}^2}{2} \quad (12)$$

$C_{i,r}^2$ 은 변동계수(coefficients of variations)로 $C_{i,r}^2 = \frac{\text{Var}(x)}{[E(x)]^2}$ 로 표시한다. 작업물의 평균 대기시간은 (10)~(12)식을 합한 값이 된다.

$$W_{i,r}(N) = S_{i,r} + \sum_{i=1}^R S_{i,r} (L_{i,r}(N-I_r) + U_{i,r}(N-I_r) \frac{C_{i,r}^2 - 1}{2}) \quad (13)$$

작업시간이 지수분포를 할 경우(13식에서 $C_{i,r}^2 = 1$ 이 되어,

$$W_{i,r}(N) = S_{i,r} + \sum_{i=1}^R S_{i,r} L_{i,r}(N-I_r) \quad (14)$$

로 표시되고,

작업시간이 상수인 경우에는 $C_{i,r}^2 = 0$ 가 되어

$$W_{i,r}(N) = S_{i,r} + \sum_{i=1}^R S_{i,r} (L_{i,r}(N-I_r) - \frac{U_{i,r}(N-I_r)}{2}) \quad (15)$$

로 표시된다.

작업시간이 지수분포를 하는 경우는 (14)식으로 Linearizer해법의 평균 대기시간을 대치하여 계산을 할 수 있으나 상수인 경우에는 (15)식에서 $U_{i,r}(N-I_r)$ 값을 구해야 하기 때문에 계산이 복잡해진다. Cavaille등(5)은 Bard-Schweitzer해법으로는 $U_{i,r}(N-I_r)$ 값을 구할 수 없다고

하였으나 본 연구에서는 $L_{ir}(N-I_r)$ 과 마찬가지로 $X_j(N-I_r)$ 에 수정항 (correction term) 을 이용하여 값을 구한 결과 Cavaille 등 [5] 의 결과보다 정확한 값을 얻을 수 있었다.

출력률 $X_j(N-I_r)$ 은 다음과 같이 구한다.

$$X_j(N-I_r) = \begin{cases} X_j(N) & j \neq r \text{ 일 때} \\ \frac{N_j-1}{N_j} X_j(N) & j=r \text{ 일 때} \end{cases} \quad (16)$$

위의 결과를 Linearizer 해법에 적용할 경우 원래의 절차와 다른 부분만을 설명한다.

$X_j(N)$ 과 $X_j(N-I_r)$ 의 초기값을 구해 주고 $U_{ir}(N-I_r)$ 값을 구한다.

$$V_{ir}(N) - V_{ir}(N-I_r) = S_{ir} [X_j(N) - X_j(N-I_r)] = S_{ir} \delta_r \text{ 이 되고}$$

$V_{ir}(N-I_r) = V_{ir}(N) - S_{ir} \delta_r$ 가 된다.

$$\text{이때 } \delta_r = \begin{cases} 0 & j \neq r \text{ 일 때} \\ \frac{X_r(N)}{N_r} & j=r \text{ 일 때} \end{cases}$$

평균 대기시간 $W_{ir}(N)$ 을 (15)식에 의해 구한다.

4.2. 작업장이 다중서버를 갖는 경우

다중서버모델의 경우 작업장 i 에서 작업물 r 이 기다리는 시간의 식은 다음과 같이 표시된다. [14]

$$W_{ir}(N) = S_{ir} + W_{ir}(N) + \frac{1}{NS(i)} \sum_{j=1}^R S_{ij} \{L_{ir}(N-I_r) - V_{ir}(N-I_r)\} \quad (17)$$

이 식의 의미는 앞의 단일서버의 대기시간의 식과 같다.

돌짜항의 $W_{ir}(N)$ 은 작업물이 도착했을 때 모든 서버가 작업중인 경우 작업중인 작업물의 하나를 내보낼 때까지 걸리는 시간이고 세째항은 작업중인 작업물을 제외한 대기행렬의 평균 대기 시간을 뜻하는데 서버의 수 $NS(i)$ 만큼 대기 시간이 짧아진다.

$W_{ir}(N)$ 의 계산은 $NS(i) = 1$ 인 경우와 $NS(i) > 1$ 인 경우로 나누어 계산하는데, $NS(i) = 1$ 인 경우 단일서버를 뜻하므로,

$$W_{ir}(N) = S_{ir} + \sum_{j=1}^R S_{ij} V_{ir}(N-I_r) + \sum_{j=1}^R S_{ij}$$

$$\{L_{ir}(N-I_r) - V_{ir}(N-I_r)\} = S_{ir} + \sum_{j=1}^R S_{ij} L_{ir}(N-I_r) \quad (18)$$

$NS(i) > 1$ 인 경우 $W_{ir}(N) \cdot D_{ir}(N)$ 으로 표시되며 여기서 $B_{ir}(N)$ 은 작업물 r 이 도착했을 때 작업장 i 에서의 모든 서버가 작업중일 확률이고 $D_{ir}(N)$ 은 모든 서버가 작업중일 때 작업물이 작업을 마치고 떠날때까지의 시간이다.

$$D_{ir}(N) = \frac{\sum_{j=1}^R V_{ir}(N-I_r)}{NS(i)! \sum_{j=1}^R X_{ir}(N-I_r)} \quad (19)$$

로 표시되고,

$$B_{ir}(N) = \begin{cases} \frac{U_{ir}^{NS(i)} (1 - F_{ir}^{L_{ir} - NS(i)})}{NS_{ir}! (1 - F_{ir}) \sum_{t=0}^{L_{ir}-1} \frac{V_{ir}^t}{t!} +} \\ \frac{1 - F_{ir}^{L_{ir} - NS(i)}}{NS_{ir}! (1 - F_{ir})} & NS_{ir} \leq L_{ir} \text{ 일때} \\ 0, NS_{ir} > L_{ir} \text{ 일때} \end{cases} \quad (20)$$

여기서 $U_{ir} = \sum_{j=1}^R V_{ir}(N-I_r)$ 이고 $F_{ir} = \frac{V_{ir}}{NS_{ir}}$ 이다.

위 식에서 $L(i)$ 는 시스템을 (M/M/s/N)의 대기행렬로 보았을 때의 N 과 같이 작업장 i 에 도착한 작업물이 그 곳에서 찾아볼 수 있는 최대 작업물의 수를 표시한다.

5. 적용예제

5.1. 작업장이 單一서버를 갖는 경우

본 모델에서는 Cavaille 등 [5]이 사용한 예제에 Linearizer 해법을 적용하여 기존의 결과와 비교한다. 본 모델은 운송시스템은 고려하지 않았다.

表 1. 작업물의 작업장별 가공시간

작업물 \ 작업장	1	2	3	4	5	6	7	8	파트수
1	2.0	-	3.7	-	1.7	0.5	1.0	1.5	6
2	3.9	-	-	-	3.7	3.2	1.0	1.5	6
3	0.95	2.0	2.2	2.0	3.1	4.3	1.0	1.5	6
4	1.1	1.2	-	-	-	1.9	1.0	1.5	6
5	0.7	-	6.4	1.0	1.3	1.6	1.0	1.2	6
6	1.4	1.7	-	1.0	-	0.4	1.0	1.2	6

1) 작업시간이 指数分布를 따를때

본 연구의 Linearizer 해법을 이용한 결과, 기존해법의 결과를 기존 연구 [5]에서 제시한 시뮬레이션 결과를 기준으로 삼아 비교한다.

- A 1 : 기존연구의 결과
- B 1 : Linearizer 해법 결과
- C 1 : 시뮬레이션 결과

表 2. 각 작업장의 이용률과 대기행렬의 길이

작업장		1	2	3	4	5	6	7	8
이용률	A 1	0.88	0.44	0.86	0.39	0.70	0.97	0.51	0.68
	B 1	0.91	0.44	0.87	0.32	0.72	1.00	0.52	0.71
	C 1	0.91	0.44	0.86	0.31	0.71	0.97	0.52	0.70
대기행렬 길이	A 1	7.4	0.8	4.5	0.5	2.4	17.3	.0	2.1
	B 1	9.0	0.8	4.4	0.47	2.4	15.4	1.2	2.4
	C 1	8.9	0.9	4.3	0.45	2.3	15	1.4	2.6

表 3. 작업시간각 작업물의 출력률

작업물		1	2	3	4	5	6
출력률	A 1	0.072	0.092	0.067	0.107	0.070	0.104
	B 1	0.074	0.096	0.067	0.110	0.072	0.105
	C 1	0.073	0.096	0.067	0.108	0.071	0.104

2) 작업시간이 常數로 주어질 때

본 연구의 Linearizer 해법을 이용하여 기존 해법의 결과의 시뮬레이션 값과 함께 비교한다.

- A 2 : 기존 연구 결과
- B 2 : Linearizer 해법의 결과
- C 2 : 시뮬레이션 결과

表 1. 각 작업장의 이용률과 대기행렬의 길이

작업장		1	2	3	4	5	6	7	8
이용률	A 2	0.91	0.45	0.91	0.32	0.73	1.00	0.53	0.71
	B 2	0.90	0.44	0.92	0.32	0.73	1.00	0.52	0.70
	C 2	0.86	0.43	0.96	0.33	0.74	1.00	0.51	0.69
대기행렬 길이	A 2	5.3	0.6	4.5	0.4	1.7	21	0.8	1.6
	B 2	4.9	0.6	4.1	0.4	1.6	22	0.8	1.4
	C 2	2.5	0.5	2.3	0.4	1.4	27	0.6	0.9

表 5. 각 작업물의 출력률

작업물		1	2	3	4	5	6
출력률	A 2	0.075	0.096	0.071	0.11	0.074	0.10
	B 2	0.077	0.093	0.071	0.10	0.074	0.10
	C 2	0.079	0.089	0.078	0.094	0.079	0.093

지금까지 單一서버 모델에서 본 연구의 해법들을 이용해 기존연구의 결과와 비교해 보았다. 기존 연구결과와 함께 해법의 타당성 평가를 위해 시뮬레이션 결과를 기준으로 비교해 보았을 때 본 연구의 해법이 좀 더 정확함을 알 수 있었다. 여기서는 나타내지 않았지만 작업물에 해당된 파렛트의 수가 커질수록 두 연구결과의 차이가 커짐을 보였다. 특히 대기행렬의 길이에 있어서는 본 연구의 결과가 시뮬레이션 값에 훨씬 가까운 결과를 나타낸다.

5.2. 多重서버 모델의 경우

多重서버 모델은 Schweitzer 등 [14]의 기존 예제를 적용한다. 각 작업장은 多重서버를 갖고 있고 2대의 AGV를 운송시스템으로 사용한다. 시스템내에는 檢査場을 두어 각 작업물 별로 10%의 품질검사를 실시한다. 이 예제에서도 본 연구의 결과, 기존의 연구결과와 시뮬레이션을 함께 비교한다.

表 6. 入力資料

작업물 \ 작업장	1	2	3	4	5	파렛트수
	1	20	15	-	35	
2	30	15	-	45	8	7
3	25	18	25	35	7	10

- A : 기존연구의 결과
- B : 본 연구의 결과
- C : 시뮬레이션 결과

表 7. 각 작업물의 출력률

작업물		1	2	3
출력률	A	0.0373	0.0351	0.0324
	B	0.0372	0.0351	0.0334
	C	0.0372	0.0352	0.0336

表 8. 각 작업물이 AGV에 머무는 시간

작업물		1	2	3
AGV에 머무는 시간	A	81.45	18.49	18.58
	B	17.62	18.10	81.13
	C	17.6	17.6	17.8

表 9. 작업장과 운송시스템의 이용률과 대기행렬의 길이

작업장		1	2	3	4	5	AGV
이용률	A	0.870	0.825	0.811	0.402	0.769	0.893
	B	0.877	0.842	0.835	0.405	0.775	0.902
	C	0.871	0.848	0.830	0.394	0.782	0.898
대기행렬 길이	A	3.59	2.96	2.19	0.25	2.14	4.83
	B	3.62	3.15	1.85	0.27	2.27	4.68
	C	3.52	3.23	1.88	0.24	2.41	4.62

시스템이 多重서버모형을 갖는 경우 본 연구의 Linearizer 해법은 앞의 單-서버 모델에의 적용보다 더욱 정확성을 보였는데 表7에서 알 수 있듯이 작업물 1과 2의 出力率은 시물레이션과 거의 일치했고 특히 작업물 3의 出力率은 기존결과에 비해 보다 시물레이션에 가까운 값을 보여준다. 또한 그외의 수행척도들도 매우 정확한 결과를 보여준다.

6. 결론

본 연구에서는 FMS를 폐쇄형 대기행렬네트워크(CNQ)로 가정하여 컴퓨터의 성능 평가에 쓰이는 발견적 MVA중 정확한 해법의 MVA의 값과 오차한계가 가장 적은 Linearizer 해법을 이용하여 FMS의 성능평가를 위한 성능척도들을 구하였다. 본 연구에서는 작업장에서 작업물의 종류에 따라 대기행렬의 작업시간이 다를 때 (class-dependent), 작업시간이 指數分布와

常數를 갖는 경우 Linearizer 解法을 적용하여 기존의 결과보다 定確한 값을 얻을 수가 있었고 또한 작업장이 多重서버를 갖고 AGV와 같은 운송시스템이 여러 대가 있는 경우 본 연구의 Linearizer 解法을 擴張하여 좋은 결과를 얻을 수가 있었다.

Linearizer해법의 실행시간에 있어서는 본 연구의 적용대상이 된 시스템의 경우 미니컴퓨터인 PDP-11/24를 사용하였을 때 20초를 넘지 않았다. 또한 시스템의 규모를 크게 해도 계산시간은 규모에 비례하지 않고 상식적으로 허용할 수 있는 시간 내에서 계산이 가능하였다.

다중서버의 경우에 있어서는 작업시간이 지수분포를 한다는 가정하에서 평균 대기시간을 구하였는데 단일서버의 경우와 마찬가지로 FMS에서의 현실적인 작업시간은 상수로 많이 나타나므로 이 경우의 평균 대기시간을 구하는 문제 등이 앞으로의 과제로 남아 있다.

References

1. Bard, Y., "Some extensions to multiclass queueing network analysis," *Performance of Computer Systems*, North-Holland pp. 51-62 (1979).
2. , "The VM/370 performance predictor," *Computing Survey*, Vol 10, No 8, pp. 333-342 (1978).
3. Buzen, J. P., "Computational Algorithms for Closed Queueing Networks with Exponential Servers," *Comm. ACM*, pp. 527-531 (1973).
4. Buzen, J. P., and Denning, P. J., "The operational analysis of queueing network model," *Computing Survey* Vol 10, No, 3, pp 225-262 (1978).

5. Cavaille J. B. and Dubois, D., "Heuristic Methods based on MVA for Flexible Manufacturing Systems Performance Evaluation," *Proceeding of IEEE Conference on Decision & Control*, pp. 1061-1065 (1982).
6. _____, "Théorie des réseaux de files d'attente et algorithmes pour l'évaluation des performances dun atelier de production," *Repport DERA n° 117 CERT*, (1982).
7. Chandy, K. M. and Sauer, C. H., *Computer Systems Performance Modeling*, Prentice Hall Inc. (1981).
8. Chandy, K. M. and Neuse, D., "Linearizer: A heuristic algorithm for queueing network models of computer systems," *Comm. ACM*, pp. 126-134 (1982).
9. Hildebrant, R. R., "Scheduling flexible machining systems using MVA," *Proceedings of IEEE Conf. on Decision & Control*, pp. 701-765 (1980).
10. Hildebrant, R. R., and Suri, R., "Modelling Flexible Manufacturing Systems using Mean Value Analysis," *Journal of Manufacturing Systems*, Vol. 1, No. 3, pp. 27-36 (1984).
11. Lazowska, E. D., Zahorjan, J., Graham, G. S., and Sevick, K. C., *Quantitative System Performance*, Prentice-Hall, Inc. (1984).
12. Reiser, M. and Lavenberg, S. S., "Mean Value Analysis of Closed Multichain Queueing Networks." *Journal of ACM*, Vol. 27, pp. 313-322 (1980).
13. Reiser, M., "A queueing network analysis of computer communication network with window flow control," *IEEE Trancions on communications*, Vol. COM-27, No. 8, pp. 1199-1209 (1979).
14. Schweitzer, P. J. and Shalev-Oren, S. and Seidmann, A., "Analysis of Flexible Manufacturing Systems with Priority Scheduling: PMVA," *Annals of O.R.*, pp. 115-139 (1985).
15. Schweitzer, P. J., "Approximate Analysis of Multiclass Closed Network of Queues", *Presented at the International Conf. on Stochastic Control and Optimization*, Amsterdam, The Netherlands. (1979).
16. Solberg, J., "Capacity planning with a stochastic work flow model," *AIIE Trancion*, Vol. 13, pp. 116-122 (1981).