

안정상태 시뮬레이션 출력 데이터의 온라인 제거 시점 결정 방법*

Methods for On-Line Determination of Truncation Point in Steady-State Simulation Outputs

이영해,** 경규형,*** 박세준**

Young Hae Lee, Kyu Hyung Kyung, Se Joon Park

Abstract

Simulation output is generally stochastic and autocorrelated, and includes the initial condition bias. To exclude the bias, the determination of truncation point has been one of important issues for the steady-state simulation output analysis. In this paper, two methods are presented for detection of truncation point in order to estimate efficiently the steady-state measure of simulation output. They are based on the Euclidean distance equation, and the backpropagation algorithm in Neural Networks. The experimental results obtained by M/M/1 and M/M/2 show that the proposed methods are very promising with respect to coverage and relative bias. The methods could be used for the on-line analysis of simulation outputs.

* 이 논문은 1997학년도 한양대학교 공학기술연구소 연구비에 의하여 연구되었음.

** 한양대학교 산업공학과

*** 현대자동차(주)

1. 서론

출력분석의 관점에서 시뮬레이션은 종료 시뮬레이션과 안정상태 시뮬레이션의 두 가지 형태로 나눌 수 있는데, 안정상태 시뮬레이션 모델의 출력을 분석할 때에는 종료 시뮬레이션에서는 존재하지 않는 문제인 초기 조건에 의한 편의(initial bias), 표본간의 자동상관(autocorrelation), 시뮬레이션 수행 길이(run length) 등의 문제가 발생한다. 또한, 안정상태 시뮬레이션 출력분석과 관련하여 가장 중요하면서도 어려운 문제는 시뮬레이션 모델의 안정상태에서의 평균과 이 평균에 대한 신뢰구간을 정확하게 구하는 것이다. 그러나, 시뮬레이션으로부터 얻어진 출력데이터가 일반적으로 확률적이고 자동상관되어 있기 때문에, 전통적인 통계적 기법을 그대로 이용할 수 없다[7].

그리고, 시뮬레이션 수행 회수, 수행 길이, 데이터 수집방법 등과 관련하여 시뮬레이션 출력분석 방법이 자동화 되지 않으면, 시뮬레이션에 필요한 시간과 인적 자원이 상당히 필요하게 된다. 따라서, 시뮬레이션 출력분석 과정의 온라인화 및 자동화는 시뮬레이션 기법을 이용하여 시스템 분석을 행하는 연구의 생산성 향상에 매우 중요하다. 특히, 컴퓨터와 정보통신 시스템에 대한 시뮬레이션 분석의 경우와 같이 수많은 모델에 대한 시뮬레이션을 수행하게 되는 경우, 온라인화 및 자동화된 시뮬레이션 출력분석 방법이 필히 요구된다[5].

시뮬레이션 출력분석의 자동화를 위해서는 시뮬레이션을 수행하는 경우에 항상 발생하는 초기 편의를 제거하는 문제가 선결되어야 한다. 시뮬레이션 출력분석에 사용되는 데이터들이 초기 편의를 반영하지 않는 안정상태에서 수집된 것이어야만 실제 시스템에 대한 올바른 해석이 가능하다.

본 논문에서는, 안정상태 시뮬레이션에서 초기 편의를 제거하기 위해서 필요한 출력데이터의 제거시점(truncation point)을 찾는 새로운 기법으로, 온라인으로 단일 시뮬레이션 수행(single run)이 가능한, 유클리드 거리(Euclidean Distance)를 이용한 방법과 역전파 신경망(Backpropagation Neural Networks)을 이용하는 방법을 제시한다.

2 장에서는 제거시점 결정에 관한 기존연구에 대하여 살펴보고, 3, 4 장에서는 본 논문에서 제안하는 유클리드 거리를 이용한 방법(ED방법)과 신경망을 이용한 방법(NN 방법)에 대해 각각 설명하고, 5 장에서는 수치 예제를 통한 실험과 결과를 분석한다. 마지막으로 6 장에서 결론 및 추후 연구과제를 제시한다.

2. 기존 연구

안정상태 시뮬레이션 출력분석 분야에서 제거시점 결정 방법과 관련된 기존연구는 다음과 같다. 콘웨이방법(Conway Rule: CR)[4]은 현재의 데이터가 그 이후 데이터의 최대값이나 최소값이 아니면 이 데이터를 제거시점으로 결정하는데, 알고리즘 구조상 온라인으로 제거시점 결정이 불가능하다. 콘웨이 방법이 알고리즘의 성격상 온라인이 불가능한 반면, 수정 콘웨이 방법(Modified Conway Rule: MCR)[4]은 현재의 데이터가 이전 데이터와 비교했을 때 최대값이나 최소값이 아닌 경우 현재의 데이터를 제거시점으로 결정하기 때문에 온라인이 가능하다.

평균교차방법(Crossings-of-the-Mean Rule: CMR)[5]은 누적평균을 이용하면서 이 평균을 중심으로 관측치가 위에서 아래로, 또는 아래서 위로 교차하는 회수로 결정한다. 이 기법을 사용하려면 교차 회수를 결정해야 하는데, 일반적으로 결정된 교차 회수가 시스템에 상관없이 일반적으로 적용이 가능하지 않다.

누적평균방법(Cumulative-Mean Rule: CMR2)[4]은 여러 번의 시험수행을 통해서 얻어진 출력데이터에 대한 총누적평균(grand cumulative mean)을 그래프로 그린 다음, 안정상태인 점을 육안으로 결정한다. 이 방법은 여러 번의 시뮬레이션을 수행에서 얻어진 데이터들의 평균들에 대한 누적평균을 사용하기 때문에 온라인 제거시점 결정이 불가능하며, 작업자가 그래프를 보고 임의로 결정해야 하는 단점이 있다.

Law-Kelton방법(Law-Kelton's Method: LKM)[6]은 회귀이론에 기초하는데, 시뮬레이션이 종료된 후 누적평균 데이터에 대해서 회귀직선을 적합시

킨다. 회귀직선의 기울기가 0 이라는 귀무가설이 채택되면 그 시점을 제거시점으로 결정한다. 일단 시뮬레이션이 종료된 다음, 데이터가 모아진 순서의 반대 순서로 데이터를 이용하기 때문에 온라인이 불가능하다.

Welch 방법(Welch's Procedure: WP)[10]는 5 회이상의 시뮬레이션수행을 통해 수집한 데이터의 이동평균을 이용해서 시각적으로 제거시점을 결정해야 하며, 반복제거방법을 사용해야 하기 때문에 온라인 제거시점의 결정이 불가능하다. 또한, 한번에 이동할 데이터의 크기(window size)를 결정해야 한다.

또한, 앞에서 설명한 방법 이외에도 관련 연구로서 초기 편의의 존재 유무를 테스트하는 기법[8,9]도 제시되었다.

위에서 설명한 기존의 방법들은 시뮬레이션의 단일 수행 중의 온라인 제거시점 결정의 관점에서는 미약한 면이 있다. 또한, 현재의 시뮬레이션 상용 소프트웨어는 작업자로 하여금 제거시점을 임의로 결정하도록 하기 때문에, 실험 중인 시스템에 대해서 정확하고도 정량적으로 제거시점을 결정할 수 없게 되어 있다. 사용자가 임의로 제거시점을 결정하게 되면, 초기편의 문제를 효과적으로 해결하기 어려울 뿐만 아니라, 필요 이상으로 너무 많은 양을 제거하거나 초기 편의를 해결하지 못할 만큼 너무 적은 양을 제거할 가능성이 커지게 된다[7].

3. 유클리드 거리를 이용한 방법

벡터 X 를 시뮬레이션에서 발생한 출력데이터를 원소로 하는 n 차원 벡터라 하고

$X = (x_1, x_2, \dots, x_n)$ 로 표시하자. 이때, 이 벡터와 원점 사이의 유클리드 거리(Euclidean distance: ED)는 다음의 식으로 나타내어 진다.

$$ED = \sqrt{x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2}$$

이때, 원래의 시뮬레이션 출력데이터를 [0,1] 사이의 값으로 정규화(normalization) 하려면, 벡터 X

의 각 원소들에 다음의 정규화 인수(normalization factor) $f = 1/ED$ 를 곱하면 된다. 그 결과, 정규화된 벡터 X' 의 원소들 값의 범위는 [0,1] 이며, 정규화된 벡터 X' 은 크기가 1인 단위 벡터가 된다.

< 그림 1 > 에는 M/M/1 대기행렬에서 $n = 10$ (즉, 10 개의 연속된 출력데이터로 하나의 벡터를 구성)의 경우, 이용률(utilization factor)에 따른 시뮬레이션 출력데이터의 정규화 전과 후의 그래프가 나타나 있다. 이용률에 상관없이 안정상태로 접근함에 따라 X' 의 원소들 값은 약 0.3162277이 된다.

여기서, 0.3162277 값은,

벡터 $x_i = x, i = 1, 2, \dots, 10$ 인 경우,

방정식 $\sqrt{x_1^2 + x_2^2 + x_3^2 + \dots + x_{10}^2} = 1$ 의 해로

$x_i = x = 0.3162277, i = 1, 2, \dots, 10$ 에서 나왔음을 알 수 있다. 안정상태에서 10 개 데이터 그룹으로 정규화된 데이터는 약 0.3162277을 평균으로 유한 크기의 분산을 가지고 상하로 움직인다고 해석할 수 있다.

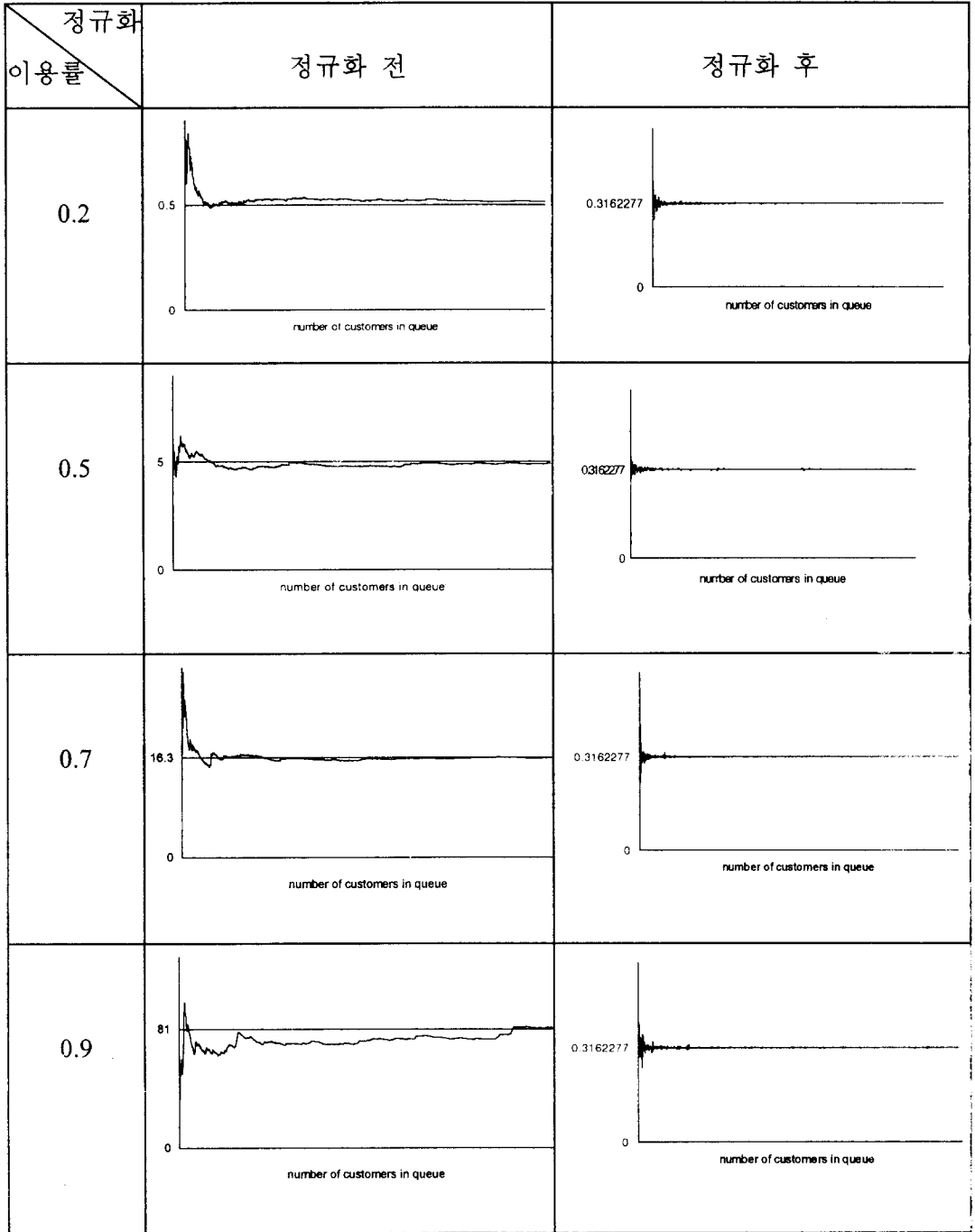
ED 방법에 의한 절차는 다음과 같다.

-**단계 1:** 시뮬레이션을 수행하면서 얻어지는 n 개의 출력데이터를 한 개의 그룹으로 하여, 정규화를 행함.

-**단계 2:** 정규화된 n 개 출력데이터 그룹들의 평균의 차이가 연속 k 번 미리 정해진 아주 작은 값(ED 기준 이라 부름, 거의 0 의 값)내에서 움직이면, 단계 3 으로 가고, 아니면, 단계 1로 감.

-**단계 3:** 지금부터 안정상태에 진입(제거 시점)한 것으로 간주하고, 필요한 통계량을 구하기 위한 데이터를 수집하고, 시뮬레이션 종료함.

시뮬레이션의 출력데이터가 연속으로 k 번 ED 기준에 만족되는 시점 이후를 안정상태라고 정한 이유는 일시적으로 안정상태에 들어간 것을 안정상태로 판단하는 오류를 범하지 않기 위함이다.



< 그림 1 > ED 방법에 의한 출력데이터 정규화의 결과

4. 신경망을 이용한 방법

신경망은 어떤 패턴이 들어왔을 때 여러 가지 패턴들을 저장하였다가 즉석에서 그것을 분류할 수 있는 실시간 패턴인식에 있어서 뛰어난 능력을 가지고 있다고 알려져 있다. 다층 퍼셉트론은 단층 퍼셉트론과 유사한 구조를 가지고 있지만 중간층과 각 유닛의 입출력 특성을 비선형으로 함으로써 네트워크의 능력을 향상시켜 단층 퍼셉트론의 여러가지 단점을 극복했다[3].

신경망의 학습규칙 중에 가장 많이 사용되어지는 역전파(backpropagation) 훈련 알고리즘은 최소경사법(gradient descent law)을 다층 퍼셉트론(multi-layer perceptron)에 확장 적용한 것으로서 비선형 패턴분류에 탁월한 성능을 발휘하는 일종의 지도 학습 규칙이다. 역전파 훈련 알고리즘은 다층 전향 퍼셉트론의 실제 출력과 기대되는 출력 간의 평균제곱 오차를 최소화하도록 설계된 알고리즘으로 연속적이며 미분가능한 비선형 함수를 필요로 한다[3].

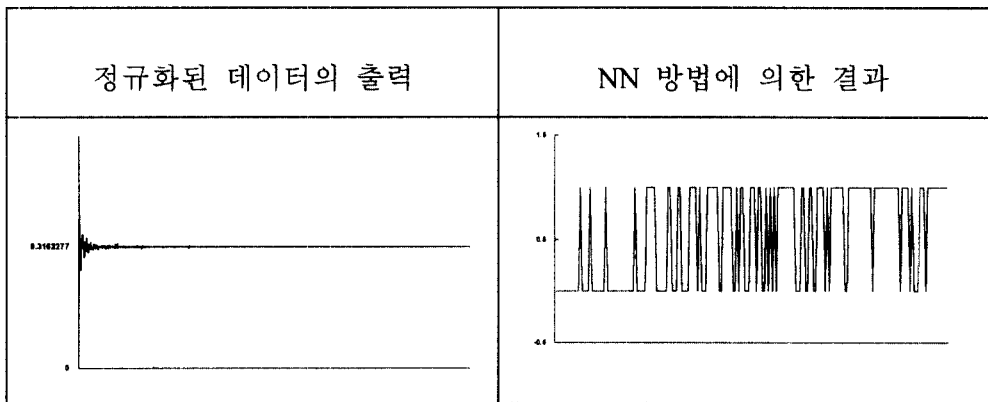
본 논문에서 제시하는 신경망을 이용한 방법(NN방법)은 유클리드 거리를 이용해서 정규화된 데이터를 사용했다는 점에서는 ED 방법과 마찬가지로, 정규화된 데이터의 평균을 이용하지 않고, n개의 데이터가 가지는 특성을 패턴으로 인식하도록 신경망을 학습시킨 다음, 훈련 패턴과 유사한 형태의 패턴이 들어오면 원하는 출력값을 출력하도록 했

다는 점에서 다르다. 즉, 수치적으로 나타내기 어려운 패턴의 성질들을 패턴의 유사성에 근거하여 출력값을 나타내도록 한 것이다.

신경망의 출력 노드값 1의 의미는 훈련 데이터상의 안정상태인 데이터의 패턴과 현재의 패턴이 일정한 오차범위 내에서 일치한다는 의미다. 신경망의 출력노드값이 1인 경우를 NN 기준 이라고 하고, NN 기준이 k 번 연속으로 만족되는 시점 이후를 안정상태로 보았다. 그 이유는 ED 방법과 마찬가지로, 일시 안정상태에 들어간 것을 영구히 안정상태로 판단하는 오류를 범하지 않기 위해서 이다. <그림 2>는 시스템이 안정상태에 들어가면 신경망이 1의 결과 값을 나타내는 경우가 빈번해지는 것을 보여 주고 있다.

5. 실험 및 평가

알고리즘의 효율성을 알아보기 위해 앞서 제시한 여러가지 수행도를 기준으로 기존 방법들과 본 논문에서 제시한 알고리즘을 비교했다. 실험에 사용한 시스템은 안정상태에서 이론적인 평균대기시간을 구할 수 있는 M/M/1과 M/M/2 대기행렬 시스템을 사용했으며, 이때 사용한 이용률은 **0.2, 0.5, 0.7, 0.9**의 네 가지 수준이다. 안정상태를 결정하기 전까지 본 논문의 알고리즘과 기존 알고리즘이 사용하는 데이터의 형태는 시뮬레이션에서 발생하는 출력데이터(각 고객의 대기시간)에 대한 누적평균 데이터



< 그림 2 > 정규화된 데이터의 출력과 NN 방법의 결과

이며, 제거시점이 결정된 다음에는 실제의 시물레이션에서 발생하는 출력데이터를 사용했다.

시물레이션을 수행함에 있어서 seed number를 다르게 지정했으며, 25번의 시물레이션에 대한 각각의 알고리즘이 결정한 제거시점을 기준으로 여기에서 사용된 구간평균방법(Barch Means Method)을 위해 Chien의 논문[2]을 참조하여, 배치크기는 100, 배치수는 50과 100의 두 수준으로 데이터를 수집했다. M/M/1, M/M/2 대기행렬 시스템의 초기 조건은 시스템이 'idle and empty'한 상태이며, 모델링에 사용한 언어는 SLAMSYSTEM이다. M/M/1, M/M/2 대기행렬 시스템의 평균대기시간의 이론치($1/\lambda=10$ 인 경우)는 < 표 1 >과 같다.

<표 1> 평균 대기시간의 이론치 ($1/\lambda=10$)

이용률	M/M/1	M/M/2
0.2	0.5	0.17
0.5	5.0	3.33
0.7	16.3	13.45
0.9	81.0	76.74

유클리드 거리를 이용한 ED 방법을 위해서는 $n = 10$, ED 기준 = $0.3162278 - 0.3162276 = 0.0000002$, $k = 5$ 의 값을사용하였다.

신경망을 이용한 NN 방법을 위해서는 알고리즘에 사용한 신경망의 구조는 입력층 노드 수 10 개, 출력층 노드 수 1개, 그리고, 2개의 은닉층이며, 은닉층의 노드수는 20 이다. 신경망의 구조를 정하는 데에 있어서 입력노드 수는 10개에서 30개까지, 은닉노드 수는 10개에서 50개까지, 은닉층 수는 1개에서 2개 사이에서 값을 변경하면서 사용하였다. 은닉층수가 1개일 때는 훈련오차가 수렴되지 않았으며, 입력노드수가 20개, 30개인 경우는 입력노드수가 10개인 경우와 비교해서 훈련에 걸리는 시간이 상대적으로 많은 반면, 수행도에서 차이가 없었다.

신경망의 훈련에 있어서 훈련정지조건은 오차값이 0.0000001이하이거나, 훈련반복수가 100,000번이 된 경우이다. 사용된 학습률은 0.3 이고 모델값은 0.0, 노이즈는 0.2 이다. 지도 학습을 위해 앞서 ED

방법에서와 같은 근거로, 정규화된 10개의 입력 데이터와 이들 데이터의 안정상태 유무를 0과 1로 표현한 출력 데이터를 포함하여 신경망의 훈련 데이터로 사용했다. 이때 사용된 데이터는 이용률이 0.2인 M/M/1 대기행렬에 대한 시물레이션의 출력데이터이다.

지도 학습의 결과 정해진 가중치를 적용하여 임의의 시스템에서 구한 10개의 데이터를 정규화하여 입력 데이터로 사용한다. 그 결과로 신경망은 0 또는 1의 출력결과를 나타낸다. NN기준이 $k = 5$ 번 연속으로 만족되는 시점 이후를 안정상태로 보았다.

다음의 4가지의 수행도를 기준으로 기존의 방법과 본 논문에서 제시한 방법의 효율성을 비교했다.

- 1) 평균 제거량(TP): 알고리즘이 결정한 제거시점 이전까지의 데이터 양으로, 너무 많은 양을 필요 없이 제거하거나 초기의 편의를 없애지 못할 정도로 너무 적은 양을 제거하면 안됨.
- 2) Coverage: 모수가 추정된 신뢰구간에 포함되는 확률을 말하는데, 본 논문에서 90% 신뢰구간을 사용하므로, 0.9의 확률을 갖는 알고리즘이 좋음.
- 3) 평균상대편의(Relative Bias: RB): 모수인 평균과 추정치간의 상대적 편이의 평균치를 나타내는데 크기가 작을 수록 좋으며 다음의 식을 사용.

$$RB = \frac{1}{r} \sum_{i=1}^r \frac{|\hat{\mu}_i - \mu|}{\mu}$$

여기서, r : 시물레이션 수행회수,
 μ : 추정하고자 하는 안정상태의 평균
 $\hat{\mu}_i$: μ 의 추정치

- 4) 상대 구간 추정폭(Estimated Relative Half-Width: ERHW): 평균치에 대한 신뢰구간의 비율로 크기가 작을 수록 좋음.

비교대상 알고리즘을 정하는데 있어서, 온라인 제거시점 결정이 가능하며 시험수행이 필요 없는 알고리즘만을 선택했는데, 수정콘웨이방법(MCR)과 평균교차방법(CMR)이 기준을 만족했다. 본 논문에서 제시한 알고리즘과 기존 알고리즘의 특성은 <표 2 >와 같다.

<표 2> 알고리즘들의 특성치 비교

특성치	방법	기존방법						제한된 방법	
		CR	MCR	CMR	CMR2	LKM	WP	ED	NN
온라인 가능		N	Y	Y	N	N	N	Y	Y
시험수행 필요		Y	N	N	Y	Y	N	N	N
계산량		낮음	낮음	중간	중간	높음	중간	중간	높음

CMR에서 사용한 교차회수는 20이다. 실험 결과는 다음의 <표 3> 에서 <표 10>까지 나타나 있다. 표에서 TP는 데이터의 평균 제거량, Coverage는 이론 평균치가 추정된 신뢰구간에 포함되는 확률, RB는 이론 평균치와 추정치 간의 상대적 편이의 평균, ERHW는 상대 구간 추정 폭으로 평균치에 대한 신뢰구간의 비율을 각각 나타낸다. 그리고, 실험에서 n = 10 을 사용하였으므로 각 표에서의 실제의 데이터의 평균 제거량은 TP 값*10으로 환산되어야 한다. M/M/1 대기행렬 시스템의 실험을 수행한 결과, 기존의 알고리즘들의 효율성이 매우 나빠서 M/M/2 대기행렬 시스템의 실험에서는 제외시켰다.

CMR은 10만개의 개체가 빠져나가는 동안에도 제거시점을 발견하지 못하는 경우가 많기 때문에 실제적으로 사용하기가 힘들다. 반면에 MCR, ED 및 NN방법은 10만개의 개체가 빠져나가기 전에 제거시점을 항상 발견하였다.

<표 3> M/M/1 대기행렬 시스템의 실험 결과치($\rho = 0.2$)

METHOD/BATCH수	TP	COVERAGE	RB	ERHW
MCR/50	4.42857	0.714286	0.065286	0.092187
CMR/50	65.63636	0.909091*	0.044727*	0.082618*
ED/50	98.87500	0.875000*	0.044083*	0.104915
NN/50	89.62500	0.791667	0.065083	0.087353*
MCR/100	4.42857	0.642857	0.042857	0.061695
CMR/100	65.63636	0.818182*	0.035818*	0.059656*
ED/100	98.87500	0.791667*	0.040250*	0.059341*
NN/100	89.62500	0.666667	0.041833	0.061446

<표 4> M/M/1 대기행렬 시스템의 실험 결과치($\rho = 0.5$)

METHOD/BATCH수	TP	COVERAGE	RB	ERHW
MCR/50	4.0666	0.800000*	0.050387*	0.091058*
CMR/50	138.2222	0.777778*	0.051844*	0.093096
ED/50	100.4583	0.750000	0.066183	0.094150
NN/50	92.7500	0.750000	0.06465	0.091184*
MCR/100	4.0666	0.800000	0.039387*	0.062997*
CMR/100	138.2222	0.888889*	0.030644*	0.065553
ED/100	100.4583	0.916667*	0.039525	0.066203
NN/100	92.7500	0.791667	0.042875	0.064172*

<표 5> M/M/1 대기행렬 시스템의 실험 결과치($\rho = 0.7$)

METHOD/BATCH수	TP	COVERAGE	RB	ERHW
MCR/50	5.8666	0.666667	0.077656	0.117060*
CMR/50	110.5000	0.800000*	0.090497	0.133639
ED/50	119.7500	0.916667*	0.065580*	0.120165
NN/50	114.2083	0.791667*	0.074749*	0.118341*
MCR/100	5.8666	0.866667	0.049894	0.084807*
CMR/100	110.5000	1.000000*	0.047178*	0.107276
ED/100	119.7500	0.750000	0.068824	0.087946
NN/100	114.2083	0.875000*	0.047756*	0.084900*

<표 6> M/M/1 대기행렬 시스템의 실험 결과치($\rho = 0.9$)

METHOD/BATCH수	TP	COVERAGE	RB	ERHW
MCR/50	5.3333	0.466667	0.186116	0.156408*
CMR/50	129.6667	0.333333	0.266317	0.143774*
ED/50	159.4500	0.850000*	0.109325*	0.159623
NN/50	142.0500	0.700000*	0.144031*	0.164026
MCR/100	5.3333	0.400000	0.162679	0.119584
CMR/100	129.6667	0.333333	0.175588	0.100938*
ED/100	159.4500	0.700000*	0.089274*	0.115691*
NN/100	142.0500	0.650000*	0.106483*	0.120666

<표 7> M/M/2 대기행렬 시스템의 실험 결과치($\rho = 0.2$)

METHOD/ BATCH수	TP	COVERAGE	RB	ERHW
ED/50	118.9583	0.891000	0.072510	0.122888
NN/50	127.8333	0.833333	0.081306	0.121051
ED/100	118.4167	0.875000	0.045345	0.083730
NN/100	131.1667	0.833333	0.051026	0.082005

<표 9> M/M/2 대기행렬 시스템의 실험 결과치($\rho = 0.7$)

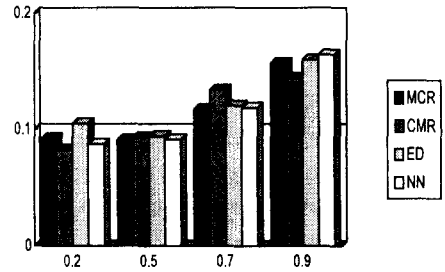
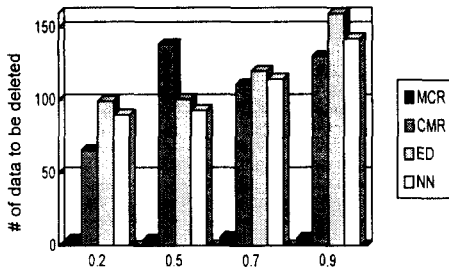
METHOD/ BATCH수	TP	COVERAGE	RB	ERHW
ED/50	127.8750	0.863000	0.281989	0.438738
NN/50	140.0833	0.850000	0.079870	0.194891
ED/100	127.8750	0.911000	0.099512	0.141143
NN/100	139.2500	0.875000	0.051016	0.100161

<표 8> M/M/2 대기행렬 시스템의 실험 결과치($\rho = 0.5$)

METHOD/ BATCH수	TP	COVERAGE	RB	ERHW
ED/50	156.0833	0.865000	0.077696	0.166743
NN/50	158.2917	0.932000	0.047059	0.169232
ED/100	156.7083	0.833333	0.052941	0.117828
NN/100	158.2917	0.877000	0.037500	0.114058

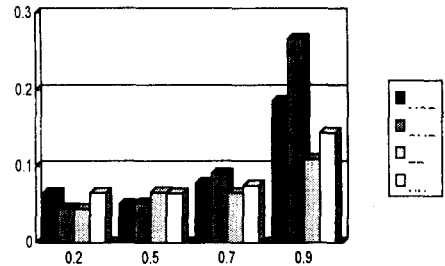
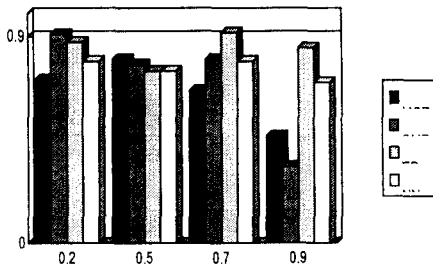
<표 10> M/M/2 대기행렬 시스템의 실험 결과치($\rho = 0.9$)

METHOD/ BATCH수	TP	COVERAGE	RB	ERHW
ED/50	151.1000	0.850000	0.171896	0.174127
NN/50	147.2500	0.765000	0.190188	0.158438
ED/100	168.1000	0.900000	0.100000	0.130399
NN/100	173.0500	0.865000	0.123392	0.126580



<그림 3> 초기 편의 데이터의 제거량

<그림 4> ERHW



<그림 5> Coverage

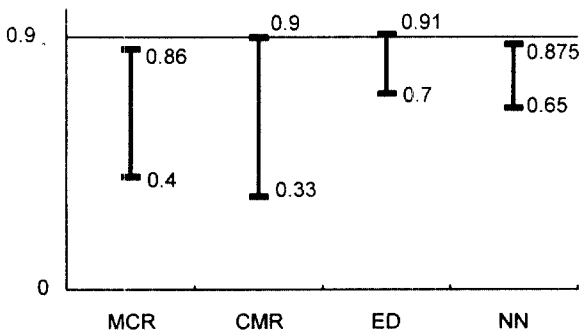
<그림 6> RB

안정상태를 위한 데이터의 제거량은, 필요 이상으로 너무 많은 양을 제거하면 경제적이지 못하게 되고, 또 너무 적은 양을 제거하게 되면, 초기 편의 문제를 효과적으로 해결하지 못할 가능성이 커지게 된다. 즉, 바람직한 제거량은 초기 편의를 해소하는 범위 내에서 가장 적은 것이 좋은 것이다. 따라서, 실험 결과는 RB 값과 동시에 TP값을 비교하면서 종합적으로 판단되어야 한다. 그런 관점에서 제안된 방법이 기존 방법에 비하여 대체적으로 우수함을 보였다.

M/M/1시스템은 이용률이 증가하면 일반적으로 변이기간(transient period)이 길어지게 되는데, 본 논문이 제시한 알고리즘은 이러한 경향을 반영해서 이용률의 증가에 따라 제거량이 증가하는 추세를 정확하게 반영하였다.

<표 3, 4, 5, 6 >의 결과에서 보이는 것과 같이 이용률 0.2, 배치수 50인 경우는 CMR이 다른 알고리즘보다 우수하며, 이용률 0.2, 배치수 100인 경우는 CMR이나 ED방법이 수행도상 다른 알고리즘보다 우수하다. 이용률 0.5인 경우는 MCR이 수행도상 다른 알고리즘보다 우수하다. 이용률이 0.7인 경우는 NN방법이 좋다.

특히, 이용률이 큰 경우(즉, 출력데이터의 자동상관이 큼)의 결과가 중요한데, <표 6, 10 >의 이용률 0.9의 실험 결과에서 볼 수 있듯이, coverage와 RB의 경우는 ED나 NN방법이 기존 방법보다 수행도가 우수하다. 여기서 주목할 것은 ED나 NN방법은 이용률에 상관없이 항상 비슷한 coverage를 보이는 반면, 기존의 알고리즘들은 이용률에 따라 coverage값의 변화가 많다.



<그림 7> 방법별 coverage의 범위

M/M/1 실험결과에 대한 요약 그래프는 <그림 3, 4, 5, 6 >에 나타나 있다.

<그림 7 >은 M/M/1 실험에서 나타난 방법별 coverage의 최고, 최저 값의 범위를 나타내고 있는데, ED나 NN방법이 기존 방법보다 coverage의 범위가 좁음을 알 수 있다.

6. 결론

본 논문에서 제시한 알고리즘은 수행한 실험 시스템에 대해서 기존 알고리즘보다 안정적으로 제거 시점을 발견하였다. 또한, 본 논문에서 제시한 방법이 알고리즘보다 제거량이 많을 경우도 있지만, 시스템의 변화에 따른 알고리즘의 효율성에 변화가 거의 없으나, 기존의 방법들은 수행도의 수준에 있어서 매우 심한 변화를 보였다.

M/M/1, M/M/2 대기행렬 시스템에 대한 실험의 결과를 보면, 본 논문에서 제시한 방법은 시뮬레이션 실험의 자동화의 일부분인 안정상태 시점의 발견 기법으로 사용이 가능하리라 생각되는데, 더욱 다양한 시스템에 대한 실험을 통해서 알고리즘의 일반성을 확인해 보아야 할 것이다.

ED 방법에서 k의 크기는 일반적으로 출력데이터의 분산에 많이 좌우되고, 주어진 시스템의 형태나 환경(시뮬레이션 출력의 correlation 크기 포함)에 따라 많이 달라질 것으로 생각되며, 이 부분에 대해서는 추후 과제로 심도 있게 연구, 실험이 필요하다고 생각된다.

본 연구에서는 안정상태에서의 모수 추정의 방법으로 구간평균방법을 사용하였으나, 기타 다른 방법[7]의 사용도 가능하다. 또한, 편의상 알고리즘의 효율성을 비교하기 위해, 구간평균방법을 사용할 때, 고려해야 할 파라미터인 배치 크기(batch size)와 배치 수(number of batches)를 고정시켰는데, 실제로 시뮬레이션을 이용한 실험의 완전 자동화를 위해서는 시뮬레이션을 하는 동안에 위의 두 파라미터를 온라인으로 결정해 주는 알고리즘이 필요하다. 이러한 방법 중의 하나로 Law와 Kelton이 제시한 순차적 방법(sequential procedure)[7]의 응용이 가능하다. ED나 NN방법과 함께 순차적 방법을 병행해서

사용하면, 실험의 완전 자동화에 필요한 온라인상의 제거시점 발견 문제와 시뮬레이션의 수행길이 결정 문제가 해결될 수 있다.

신경망 훈련문제도 온라인 학습을 고려해서 시뮬레이션 수행시 발생하는 새로운 데이터를 기존의 기증치에 반영하는 문제를 생각해 볼 수 있다[1]. 또한, 시간공유(time sharing) 컴퓨터 시스템의 경우에서 처럼, CPU 이용률과 평균 반응 지연시간 등의 다중 수행도(multiple measure of performance)를 이용한 제거시점 결정에 대한 연구도 필요하다고 생각된다[7].

참고문헌

- [1] Angulo, V.R. and C. Torras, "On-Line Learning with Minimal Degradation in Feedforward Networks", *IEEE Transaction on Neural Networks*, Vol. 6, No. 3, 1995.
- [2] Chien, C., "Batch Size Selection for the Batch Means Method", *Proceedings of the 1994 Winter Simulation Conference*, pp. 345-352, 1994.
- [3] Fausett, L.V., *Fundamentals of Neural Networks: Architectures, Algorithms, and Applications*, Prentice Hall, 1994.
- [4] Gafarian, C. J., Jr., and T. Morisaku, "Evaluation of Commonly Used Rules for Detecting Steady State in Computer Simulation", *Naval Research Logistics Quarterly*, pp. 511-529, 1978.
- [5] Heidelberger, P. and P. D. Welch, "Simulation Run Length Control in the Presence of an Initial Transient", *Operations Research*, Vol. 31, pp. 1109-1144, 1983.
- [6] Kelton, W. D. and A. M. Law, "A New Approach for Dealing with the Startup Problem in Discrete Event Simulation", *Naval Research Logistics Quarterly*, Vol. 30, pp. 641-658, 1983.
- [7] Law, A. M. and W. D. Kelton, *Simulation Modeling & Analysis, 2nd*, McGraw-Hill, 1991.
- [8] Schruben, L. W., "Control of Initialization Bias in Multivariate Simulation Response", *Communications of the ACM*, Vol. 24, No. 4, pp. 246-252, 1981.
- [9] Schruben, L. W., "Detecting Initialization Bias in Simulation Output", *Operations Research*, Vol. 30, No. 3, pp. 569-590, 1982.
- [10] Welch, P. D., The Statistical Analysis of Simulation results, in *Computer Performance Modeling Handbook*, S. S. Lavenberg, Academic Press, New York, pp. 268-328, 1983.

● 저자소개 ●



이영해

1977년

고려대학교 산업공학, 학사

1983년

Univ. of Illinois, 산업공학, 석사

1986년

Univ. of Illinois, 산업공학, 박사

경 력

한국시뮬레이션학회, 대한산업공학회, 한국경영과학회 이사

Osaka Univ., Purdue Univ. 객원교수, 대우중공업 근무

현 재

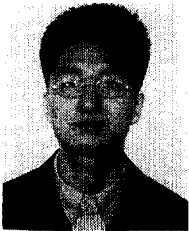
한양대학교 산업공학과 교수, 한국시뮬레이션학회 부회장,

대한설비관리학회, 대한산업공학회 이사

관심분야

시뮬레이션 출력분석, Simulation Optimization,

Web-based Simulation



경규형

1994년

한양대학교 산업공학 학사

1996년

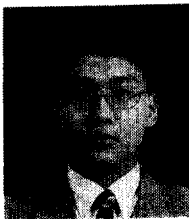
한양대학교 산업공학 석사

1996~현재

현대자동차 근무

관심분야

시뮬레이션 출력분석



박세준

1997년

한양대학교 산업공학 학사

현 재

한양대학교 산업공학과 석사과정

관심분야

시뮬레이션 출력분석, 설비배치 및 물류관리