

시뮬레이션 출력 데이터 분석 기법

한양대학교 산업공학과

이 영해 교수

출력 데이터의 통계적 분석

1 서 론

지금까지 많은 시뮬레이션 연구에서, 모델개발이나 프로그래밍에는 많은 양의 비용과 시간이 소모되었다. 그러나 시뮬레이션 결과의 분석에 대해서는 연구가 별로 없었다. 사실, 시뮬레이션 운용(operation)의 가장 일반적인 형태는, 어떤 임의의 시간 동안 시뮬레이션을 한번 행하여 거기서 산출되는 시뮬레이션 추정치(simulation estimates)를 주어진 시스템에 대한 참값(true answers)으로 취급하는 것이다. 이들 추정치들은 큰 분산을 갖는 확률변수이기 때문에 어떤 특정 시뮬레이션 과정에서 이 추정치들은 참값들과 크게 다를 수가 있다. 이런 경우 연구중인 시스템에 대해 잘못된 추론을 할 가능성이 농후하다.

이제까지 명확한 결과 분석이 부족했던 이유들 중의 하나는 시뮬레이션 결과가 독립적(independent)이지 못하기 때문이다. 그래서, IID(independent identically distributed) 관측치에 기초한 고전적 통계분석이 직접적으로 적용될 수가 없다. 현재까지 여러 가지 결과분석문제가 남아 있는데, 그 이유는 완전하게 수용할 수 있는 방법이 없고, 있다고 해도 적용하기가 어렵기 때문

이다. 모델의 참 모수치나 특성치 (true parameters or characteristics)의 정확한 추정치를 구하는 데 대한 또 하나의 방해 요인은 필요한 만큼의 시뮬레이션 결과를 모으는 데 대한 컴퓨터 비용이다. 사실, 적절한 통계적 과정이 유용할 상황이 있는데, 그 상황이 요구하는 만큼의 데이터를 모으는 비용이 매우 비싸다. 그러나, 이 문제는 컴퓨터에 대한 비용이 하락하면서 점차 해결되어져 나갈 것이다.

이 장에서의 목적은 시뮬레이션 결과의 통계적 분석에 대한 여러 기법들을 소개하고 통계학의 기초지식을 가지고 있는 독자들에게 쉽게 이해될 수 있는 실질적인 방법들을 제공하는 것이다.

2 시뮬레이션의 형태

결과 분석에 따르는 시뮬레이션의 형태는 종료 시뮬레이션 (terminated simulation)과 안정상태 시뮬레이션 (steady-state simulation)으로 나누어진다. 종료 시뮬레이션이란 시뮬레이션이 행해진 한정된 시간 $[0, T_E]$ 사이에 적절한 시스템의 수행 결과치 (measures of system performance)가 얻어지는 시뮬레이션이다. 여기서 T_E 는 시뮬레이션에서 사건(event) E 가 발생한 시간이고, 확률변수일 수도 있다. 사건 E 는 시뮬레이션 시행 이전에 정의된다. 종료 시뮬레이션의 수행 결과치는 시스템의 초기상태에 크게 의존하므로 초기상태를 결정하는 데 유의해야 한다.

[예제 9.1] 은행은 매일 저녁에 영업을 중단한다. 만약 은행이 오전 9시부터 오후 5시까지 영업을 한다면 시뮬레이션의 목적은 오전 9시부터 오후 5시에 영업을 중단할 때까지 도착한 고객에 대한 서비스 질(quality)의 결과치를 추정하는 것이다. 이 경우 사건 $E = \{\text{적어도 시뮬레이션이 8시간 경과되고, 그 시스템은 비어 있다}\}$ 이고 시뮬레이션에 대한 바람직한 초기조건은 시간 0에서 고객이 없어야 한다.

안정상태 시뮬레이션이란 실험 결과치가 시뮬레이션이 행해진 시간이 무한대로 갈 때 얻어짐으로써 정의되는 시뮬레이션이다. 이 경우, 시뮬레이션을 끝내기 위한 자연적인 사건 E (natural event)가 없기 때문에 한 시뮬레이션

의 길이(시뮬레이션 실행시간)는 바람직한 추정치를 얻기에 충분하면 된다. 시뮬레이션의 길이가 비용을 감안하여 결정되어질 수도 있지만, 이것은 타당한 결과를 얻지 못할 것이다.

[예제 9.2] 한 화학 제조업자는 제한된 화학공정운영의 시뮬레이션 모형을 만든다. 공정운영은 임의로 발생하는 고장에 지배를 받고 공정에 대한 원료의 투입물과 통제가능한 변수들 — 예를 들면 온도와 압력 — 이 시간의 경과에 따라 안정된다고 하자. 그러면 회사는 초기조건이 얻은 결과에 대하여 어떠한 영향을 미치지 않을 만큼 충분한 시간이 지난 뒤에 생산율을 추정하려 할 것이다.

위의 내용으로부터 모든 시스템에 대해서 종료 시뮬레이션이나 안정상태 시뮬레이션 중 어느 하나의 유일한 시뮬레이션 형태가 존재한다고 생각할 수 있다. 그러나 분석자가 그 시스템에 대해서 무엇을 알고 싶은가에 따라서 적절한 시뮬레이션 형태가 정해지기 때문에 어떤 시스템은 두 시뮬레이션 형태가 모두 존재한다. 예를 들어, 종업원이 시간의 흐름에 따라 일을 배워가는 조립라인의 시뮬레이션 모형에 대해 생각해 보자. 이 시스템의 안정상태 시뮬레이션은 조립라인을 충분히 운영하여 종업원이 일을 배운 후에 제조율을 추정하도록 설계되어야 한다. 한편으로, 6개월 동안의 종료 시뮬레이션은 종업원이 일을 배우고 있는 중인 첫 6개월 동안의 제조율을 추정하도록 설계되어야 한다.

3 시스템의 수행 결과치

3.1 수행 결과치의 비교

여기에서는 예제를 통하여 종료 시뮬레이션과 안정상태 시뮬레이션의 실험 결과치를 비교해 보자.

[예제 9.3] $M/M/1$ 대기행렬에서, D_i 가 i 번째 고객이 대기행렬에서 기다리는 시간인 출력 프로세스(output process) $\{D_i, i \geq 1\}$ 에 대하여 생각해 보자. 이 대기행렬은 이

비스 시간이 평균 $1/\omega$ 의 IID 지수분포를 따르고, 도착시간이 평균 $1/\lambda$ 의 IID 지수분포를 따르며, FIFO의 순서로 서비스를 받으며 $\rho = \lambda/\omega < 1$ 로 가정한다. 종료 시뮬레이션의 목적은 처음 m 명의 고객의 기대평균대기시간을 추정하는 것이다. 종료사건은 $E = \{m\text{명의 고객이 대기를 끝낸다}\}$ 이다. 초기조건, 즉 시간 0에서 시스템의 고객의 수는 $L(0)=0$ 이다. 기대평균대기시간을 $d(m|L(0)=0)$ 이라 하면

$$d(m|L(0)=0) = E\left[\frac{\sum_{i=1}^m D_i}{m} \mid L(0)=0\right] \quad (9.1)$$

로 주어진다. 즉, 확률변수 $X = \sum_{i=1}^m D_i/m$ 가 처음 m 명의 고객이 평균적으로 기다리는 시간이고 $L(0)=0$ 인 $E(X)$ 를 추정하는 것이 종료 시뮬레이션의 목적이 된다. 여기서 $E(X)$ 를 각각의 길이 m 인 매우 많은 수의 독립된 시뮬레이션 과정에서 얻어진 X 들의 평균으로 간주할 때, $E(X)$ 의 좋은 추정치를 얻기 위해서는 길이 m 의 독립된 실험을 몇 번 실행해야 하는가 하는 문제가 제기된다.

종료 시뮬레이션의 실험 결과치는 시스템의 초기조건에 따라서 달라진다. 예를 들어, $l_1 \neq l_2$ 에 대해 $d(m|L(0)=l_1) \neq d(m|L(0)=l_2)$ 이다. 안정상태 시뮬레이션의 목적은 안정상태의 기대평균대기시간 d 를 구하는 것인데

$$d = \lim_{m \rightarrow \infty} d(m|L(0)=l) , \quad l=0,1,\dots \quad (9.2)$$

로 주어진다.

$\rho > 1$ 인 경우, 길이 m 이 무한대로 갈 때, 대기행렬의 길이가 무한대로 길어지므로 d 는 무한대로 발산한다. 따라서 $\rho < 1$ 인 경우에 한하여 d 가 존재한다. 식 (9.2)에서 알 수 있듯이 d 는 초기조건에 무관하다.

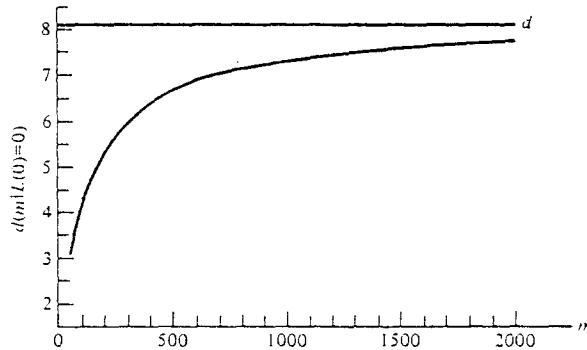


그림 9.1 $\rho = 0.9$ 인 M/M/1 대기행렬에 대한 m 의 함수로서의 $d(m|L(0)=0)$

그림 9.1은 $d(m|L(0)=0)$ 를 m 의 함수로 나타낸 것이다. ($\lambda=1, \omega=10/9, \rho=0.9$). $d(m|L(0)=0)$ 가 접근하는 수평선은 $d=8.1$ 이다. $L(0)$ 를 인위적으로 0의 값을 취했기

때문에 작은 m 값에 대해서 $d(m|L(0)=0)$ 의 값도 작다.

3.2 안정상태의 의미

예제 9.3에서 d 를 처음 m 명의 고객에 대한 기대평균대기시간의 극한으로 정의하였다. 이 정의에서의 어려운 점은 d 가 각각 무한한 길이를 갖는 무한한 시뮬레이션 과정에서 얻어지는 평균 대기시간이라는 점이다. 그러므로, d 를 좀더 실용적으로 정의하면

$$d = \lim_{m \rightarrow \infty} \frac{\sum_{i=1}^m D_i}{m} \quad \text{w.p. 1, } L(0) = \text{모든 } l \text{에 대해서} \quad (9.3)$$

로 된다. (w.p. 1 = with probability one)

이제부터는 '기대'란 용어를 사용하지 않고 d 를 대기행렬에서의 안정상태 평균대기시간이라 하겠다. 식 (9.3)이 의미하는 것은 다음과 같다. 만약, 무한한 시뮬레이션 과정을 실행하고 각각의 실행에서 충분히 큰 m 에 대하여 $\bar{D}(m) = \sum_{i=1}^m D_i/m$ 을 얻었다면, $\bar{D}(m)$ 은 확률 1로 모든 시뮬레이션 과정에 대하여 d 에 접근한다. 따라서, d 를 충분히 긴 한 번의 시뮬레이션 과정에서 얻은 평균대기시간이라 할 수 있다.

[예제 9.4] 앞의 예제 9.3에 대해 m 의 함수로서의 $\bar{D}(m)$ 과 d 를 그림 9.2에 나타내었다. m 이 무한대로 감에 따라 $\bar{D}(m)$ 이 d 로 수렴하는 경향은 그림 9.1과 같이 단조롭지 않다. m 이 5000만큼 커질 때까지도 $\bar{D}(m)$ 이 임의로 변동함을 알 수 있다.

이제, 안정상태의 올바른 의미에 대해서 알아 보자. 예제 9.3의 $M/M/1$ 대기행렬에 대해 $F_{i,i}(x)$ 를

$$F_{i,i}(x) = P\{D_i \leq x | L(0) = i\}$$

로 표시하자. 여기서 $F_{i,i}(x)$ 는 $L(0) = i$ 일 때에 시간 i 에서의 대기의 순간분포(transient distribution)를 나타낸다. 이 때 '순간'이란 단어에서 각각의 시간 i 에 대해 다른 분포가 존재함을 알 수 있다.

어떠한 $x \geq 0$ 에 대해서도 대기의 안정상태분포(steady-state distribution)

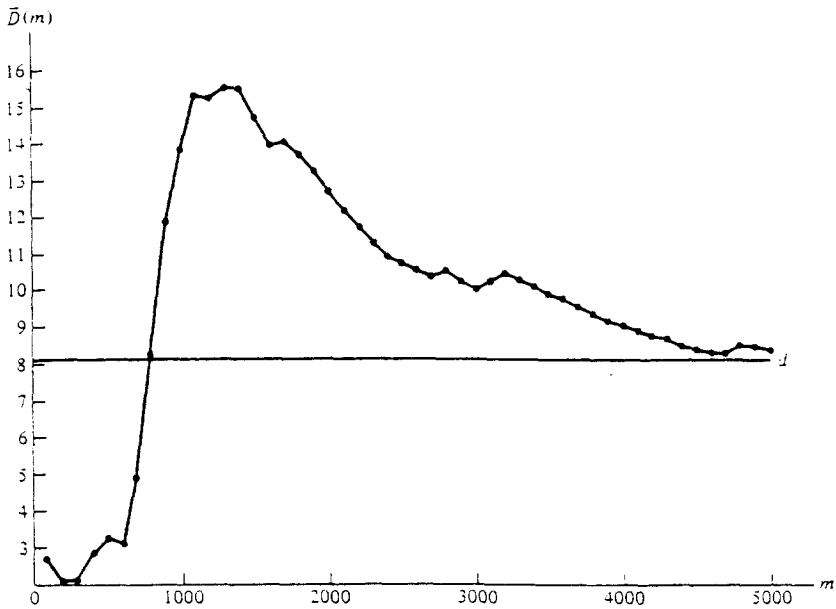


그림 9.2 $\rho = 0.9$ 인 $M/M/1$ 대기행렬 m 의 함수로서 $\bar{D}(m)$

$$F(x) = \lim_{i \rightarrow \infty} F_{i,i}(x), \quad L(0) = \text{모든 } i \text{에 대해서} \quad (9.4)$$

가 존재한다. $F(x)$ 가 i 에 대해서 본질적으로 더이상 변하지 않는 시점에서, 우리는 직관적으로 프로세스 $\{D_i, i \geq 1\}$ 가 "안정상태"에 있다고 말한다. 따라서, 안정상태란 시뮬레이션의 하나의 단일과정에서 실제대기시간이 어떤 시점 이후로 일정해지는 것을 나타내는 것이 아니고 대기시간의 분포가 변하지 않음을 나타낸다. D 를 프로세스 $\{D_i, i \geq 1\}$ 가 안정상태가 된 후에 도착한 고객의 대기시간이라 하면 $E(D) = d$ 임을 알 수 있다. 따라서, 앞에서 정의된 세 가지 d 중에서 식 (9.3)에서 정의된 d 를 안정상태의 수행결과치로 사용한다.

3.3 평균이 아닌 수행결과치

몇 개의 시스템을 비교하는 일반적인 기준은 평균시스템반응(average

system response)이다. 하나의 예를 들어 위의 기준이 잘못된 결과를 나타낼 수 있다는 것을 보여 주겠다.

[예제 9.5] 5명의 출납계원 (teller)이 있는 예제 9.1의 은행에 대해 고려해 보자. 고객의 도착은 분당 1명의 비율로 포아송 프로세스를 따르고, 서비스 시간은 평균 4분인 IID 지수확률변수이다. 따라서 $\rho = \lambda / 5\omega = 0.8$ 이다. 대기행렬에서의 기대평균대기시간과 기대시간-평균고객수를 기준으로 하여 각 창구마다 하나씩의 대기행렬이 형성되는 정책과 모든 창구에 대해 하나의 대기행렬이 형성되는 정책을 비교하려고 한다. 이때의 수행결과치들은

$$E\left(\frac{\sum_{i=1}^N D_i}{N}\right), \quad E\left[\frac{\int_0^T Q(t) dt}{T}\right]$$

로 정의된다. 여기서 T =은행의 영업시간($T \geq 8$), N =시간 $[0, T]$ 사이에서 서비스를 받은 고객의 수, $Q(t)$ =시간 t 에서 대기행렬 내의 고객의 수를 나타낸다.

표 9.1은 한번의 시뮬레이션 과정으로 얻은 각 정책에 대한 결과를 제시한다. 이 시뮬레이션 과정은 i 번째 고객의 ($i=1, 2, \dots, N$) 도착시간과 서비스를 하기 위한 시간이 두 정책에 대해 같도록 실행되었다. '평균시스템반응'에 준하여 두 정책은 같게 나타나 있다. 그러나, 이것은 명백하게 잘못된 것이다. 왜냐하면, 다대기행렬 (multiqueue)에서는 고객이 도착순대로 꼭 서비스를 받는 것이 아니기 때문이다. 표 9.2는 두 정책에 대한 시간구간 $[0, 5]$, $[5, 10]$, ..., $[40, 45]$ 에 대하여 대기를 행한 고객의 수와 전체 고객수와와의 비율을 나타내고 있다. 이것으로부터 단일대기행렬보다 다대기행렬에서 고객의 기다리는 시간이 길기 쉬움을 알 수 있다.

표 9.1 두 정책에 대한 시뮬레이션 결과 : 평균

수행 척도	추정치	
	대기열이 5인 경우	대기열이 1인 경우
기대 영업시간(시간)	8.140	8.140
기대 평균 대기시간 분	5.567	5.567
대기열에서의 기대평균 고객수	5.517	5.517

표 9.2 두 정책에 대한 시뮬레이션 결과 : 비율

구 간	기대비율의 추정치	
	대기열이 5인 경우	대기열이 1인 경우
$[0, 5]$	0.626	0.597
$[5, 10]$	0.182	0.188
$[10, 15]$	0.076	0.107

[15,20)	0.047	0.095
[20,25)	0.031	0.013
[25,30)	0.020	0
[30,35)	0.015	0
[35,40)	0.003	0
[40,45)	0	0

4 신뢰구간의 필요성

$\rho=0.9$ 인 $M/M/1$ 대기행렬에 대하여 $d(25|L(0)=0)$ 을 추정하려 한다고 하자. $L(0)=0$ 을 조건으로 서로 다른 난수를 사용하여 확률변수 $\sum_{i=1}^{25} D_i/25$ 에 대한 10개의 독립적인 실현치를 다음과 같이 구하였다.

1.051 6.438 2.646 0.805 1.505
0.546 2.281 2.822 0.414 1.307

추정량의 범위는 최소 0.414에서 최대 6.438까지로, 대부분의 추정량이 참값 2.124와는 상당한 차이가 있다. 따라서, 하나의 실현치로 만족스러운 수행 결과치를 얻기는 힘들고, 추정량이 참값에 얼마나 접근해 있는가를 확인하는 방법이 필요하다. 추정량의 정확도를 확인하는 일반적인 방법은 실제값에 대한 신뢰구간을 설정하는 것이다.

이상은 종료 시뮬레이션에 대해서만 논의한 것이지만 안정상태 시뮬레이션에 대해서도 마찬가지이다.

5 종료 시뮬레이션의 신뢰구간

특정사건 E 에 의해 각 반복실험의 길이가 정해지고, 각 반복실험의 초기조건이 같은 n 번의 종료 시뮬레이션을 독립적으로 행한다고 가정하자. 반복실험

험의 독립성은 매 반복실험마다 서로 다른 난수를 사용함으로써 지켜진다. 문제를 간단히 하기 위하여 구하는 실험 결과치가 하나라고 가정하자. 만약 X_j 가 j 번째 반복실험에서 얻은 실험 결과치의 추정량이라면, X_j 는 IID 확률변수이고 $\mu = E(X)$ 에 대한 신뢰구간을 구하는데 고전적 통계분석을 사용할 수 있는 것이다.

5.1 표본크기 방법(Fixed-sample-size-Procedures)

μ 에 대한 신뢰구간을 구하는 일반적인 접근방법은 고정된 표본크기 $n(n \geq 2)$ 번 만큼 반복실험을 실시하는 것이다. X_1, X_2, \dots, X_n 을 정규확률변수라 하면 μ 에 대한 $100(1-\alpha)\%$ 신뢰구간은

$$\bar{X}(n) \pm t_{n-1, 1-\alpha/2} \sqrt{\frac{s^2(n)}{n}} \quad (9.5)$$

으로 주어진다. 식 (9.5)는 고전적 통계학에서 모평균에 대한 신뢰구간을 구하는 식과 동일하다.

[예제 9.6] $\rho = 0.9$ 인 $M/M/1$ 대기행렬에서 $d(25|L(0)=0)$ 의 90% 신뢰구간을 구하려고 한다. 9.4절에 있는 10개의 실험치로부터

$$\bar{X}(10) = 1.982, \quad s^2(10) = 3.172$$

를 얻는다. 따라서 $d(25|L(0)=0)$ 의 90% 신뢰구간은

$$\bar{X}(10) \pm t_{9, 0.95} \sqrt{\frac{s^2(10)}{10}} = 1.982 \pm 1.032$$

이다. 즉, $d(25|L(0)=0)$ 는 대략 90%의 신뢰로 구간 $[0.950, 3.014]$ 에 포함된다고 할 수 있다.

식 (9.5)에 의해 주어진 신뢰구간의 정확성은 X_j 가 정규확률변수라는 가정에 의존한다. 이러한 이유로 예제 9.6에서 신뢰구간을 대략 90% 신뢰구간으로 하였다. 그런데 이 가정이 실제 문제에 있어서는 정확하게 만족되어지지 않으므로 정규성에서 벗어나는 신뢰구간을 실험적으로 알아보겠다. 우리의 특적인 시뮬레이션에 종사하는 사람들에게 실제 문제에 있어서 신뢰구간이 얼마나 정확한가에 대한 정보를 제공하는 것이다.

$\rho=0.9$ 인 $M/M/1$ 대기행렬에 대하여 $n=5,10,20,40$ 으로 500번의 독립된 시뮬레이션을 행하였다. 각 n 에 대해 식(9.5)를 사용하여 $d(25|L(0)=0)=2.124$ 의 90% 신뢰구간을 설정하였다. 500개의 신뢰구간 중 $d(25|L(0)=0)=2.124$ 를 포함하는 비율 \hat{p} 와 실제 포함 비율(true coverage) p 의 90% 신뢰구간, 그리고 신뢰구간의 $\frac{1}{2}$ 길이를 점추정치 $\bar{X}(n)$ 으로 나눈 비율, 즉 상대 정확도(relative precision) $[t_{n-1,1-\alpha/2}\sqrt{s^2(n)/n}/\bar{X}(n)]$ 가 표 9.3에 나타나 있다. 여기서 모비율 p 에 대한 90% 신뢰구간은

$$\hat{p} \pm z_{0.95} \sqrt{\frac{\hat{p}(1-\hat{p})}{500}}$$

를 사용하여 계산한 것인데, 이것은 $(\hat{p}-p)/\sqrt{\hat{p}(1-\hat{p})/500}$ 이 대략 표준정규 확률분포를 한다는 사실에 기초를 두고 있다.

표 9.3에서, $n=10$ 인 경우에 대하여 설명하면 다음과 같다.

500개의 신뢰구간 중 $d(25|L(0)=0)=2.124$ 를 포함하는 비율이 86.4%임을 나타내고 대략 90%의 신뢰도는 비율이 83.9%~88.9%임을 나타낸다. 시뮬레이션 모델이 실제의 시스템을 대략적으로 표현한다고 생각할 때, 표 9.3에 있는 추정된 포함비율(estimated coverages)이 바람직한 값인 0.9로서 사용하기에 충분히 가깝다고 할 수 있다. n 이 증가함으로써 상대적 정확도가 감소하는데, 그 이유는 식 (9.5)에서 신뢰구간의 $\frac{1}{2}$ 길이에 대한 식의 분모가 \sqrt{n} 이기 때문이다.

표 9.3 $\rho=0.9$ 인 $M/M/1$ 대기행렬의 500회 실험에 의한 $d(25|L(0)=0)$ 의 결과

n	추정된 포함비율	(신뢰구간 $\frac{1}{2}$ 길이)/ $\bar{X}(n)$ 의 평균
5	0.880±0.024	0.672
10	0.864±0.025	0.436
20	0.886±0.023	0.301
40	0.914±0.021	0.212

식 9.5로부터 구한 신뢰구간이 항상 $1-\alpha$ 에 접근하지 않는다는 것을 다음의 예를 통해 설명하겠다. 3개의 요소로 이루어진 신뢰도 모델(reliability model)은 요소 1이 가동하고 요소 2,3 중의 하나가 가동할 때 기능을 발휘한다. G 가 모든 시스템이 고장나는 시각이라 하고, G_i 는 요소 $i(i=1,2,3)$ 가

고장나는 시각이라 하면 $G = \min\{G_1, \max\{G_2, G_3\}\}$ 가 된다. G_i 가 독립확률변수이고 형태 모수(shape parameter) 0.5와 규모 모수(scale parameter) 1을 가지는 와이불(Weibull) 분포에 따른다고 가정하자. 이 와이불 분포는 극히 정규적이지 못하고 비뚤어진 형태를 하고 있다. 다시 한번 더 $n=5, 10, 20, 40$ 으로 500번의 독립된 시뮬레이션 실험을 행하여 식 (9.5)를 사용하여 $E(G|\text{모든 요소가 새로운 것})=0.778$ 에 대한 90% 신뢰구간을 설정하였다. 이 실험의 결과가 표 9.4에 나타나 있다. n 값이 작을 때에는 추정된 범위가 확연히 낮은 값을 갖고 n 값이 클 때는 추정된 범위가 중심극한정리에 의해서 0.9에 접근함을 알 수 있다.

표 9.4 신뢰도 모델에서 500회 실험에 의한 $E(G|\text{모든 요소가 새로운 것})=0.778$ 의 결과

n	추정된 포함비율	(신뢰구간 $\frac{1}{2}$ 길이)/ $\bar{X}(n)$ 의 평균
5	0.708±0.033	1.163
10	0.750±0.032	0.820
20	0.800±0.029	0.600
40	0.840±0.027	0.444

표 9.3과 표 9.4로부터 식 (9.5)로 구한 신뢰구간이 표본크기 n 에 의존한다는 것을 알 수 있다.

5.2 지정된 정확도를 가지는 신뢰구간

신뢰구간을 설정하는 데 있어서 표본의 크기가 고정됨으로써 불리한 점은 시뮬레이터(simulator)가 신뢰구간의 $\frac{1}{2}$ 길이를 통계할 수 없다는 점이다. 고정된 n 에 대한 신뢰구간의 $\frac{1}{2}$ 길이는 X_j 의 도분산 $\sigma^2(X)$ 에 의존한다. 예제 9.6에서 신뢰구간의 $\frac{1}{2}$ 길이가 1.032이었는데, 이것은 $d(25|L(0)=0)$ 의 참값을 알기에는 너무 큰 값이다. 이 장에서는 정해진 정확도를 가지는 신뢰구간을 알기 위해 필요한 반복실험의 수를 결정하는 방법에 대해 논하겠다.

신뢰구간의 정확도를 추정하는 두 가지 방법에 대해 다음과 같이 정의한다.

절대 정확도 (absolute precision)란 신뢰구간의 $\frac{1}{2}$ 길이의 실제값을 말하고 상대 정확도 (relative precision)란 신뢰구간의 $\frac{1}{2}$ 길이를 점추정량 $\bar{X}(n)$ 으로 나타낸 비율을 말한다.

고정된 반복실험의 수 n 에 의해 설정된 신뢰구간이 너무 커서 실질적인 사용가치가 없다고 가정하자. 모분산의 추정치 $s^2(n)$ 이 반복횟수가 증가할 때 크게 변화하지 않는다면, 신뢰구간의 절대 정확도를 바람직한 값 $\beta (\beta > 0)$ 로 줄이는 데 필요한 총반복실험횟수 $n_a^*(\beta)$ 는

$$n_a^*(\beta) = \min \left\{ i \geq n : t_{i-1, 1-\alpha/2} \sqrt{s^2(n)/i} \leq \beta \right\} \quad (9.6)$$

로 주어진다.

$n_a^*(\beta)$ 는 $t_{i-1, 1-\alpha/2} \sqrt{s^2(n)/i} \leq \beta$ 가 되도록 i 값을 1씩 반복적으로 증가시키면서 구할 수 있다. 또는 $i \geq s^2(n) [z_{1-\alpha/2}/\beta]^2$ 을 만족시키는 최소정수 i 와 같다. $n_a^*(\beta) - n$ 만큼의 추가 반복 실험을 실행한 다음의 절대 정확도는 대략 β 의 값을 가질 것이다. 같은 방법으로, 반복실험횟수가 증가할 때 모분산과 모평균의 추정치가 크게 변화하지 않는다면, 신뢰구간의 상대 정확도를 바람직한 값 $\gamma (0 < \gamma < 1)$ 만큼 줄이는 데 필요한 총반복실험횟수 $n_r^*(\gamma)$ 는 다음과 같이 주어진다.

$$n_r^*(\gamma) = \min \left\{ i \geq n : \frac{t_{i-1, 1-\alpha/2} \sqrt{s^2(n)/i}}{|\bar{X}(n)|} \leq \gamma \right\} \quad (9.7)$$

$n_r^*(\gamma)$ 는 $i \geq s^2(n) [z_{1-\alpha/2}/\gamma \bar{X}(n)]^2$ 을 만족시키는 최소정수 i 와 같다.

[예제 9.7] 예제 9.6에서 설정된 90% 신뢰구간에 대한 절대 정확도는 1.032이고, 상대 정확도는 $1.032/1.982=0.521$ 이다. 절대 정확도를 $\beta=0.5$ 가 되도록 $d(25|L(0)=0)$ 를 추정하려면, 총반복실험횟수는 다음과 같다.

$$n_a^*(0.5) = \min \left\{ i \geq 10 : t_{i-1, 0.95} \sqrt{\frac{3.172}{i}} \leq 0.5 \right\} = 37$$

따라서, 반복실험을 37번 하면 90% 신뢰도 $d(25|L(0)=0)=0.5$ 이내에 있는 $d(25|L(0)=0)$ 를 추정할 수 있다. 만약, 상대 정확도 $\gamma=0.15$ 가 되도록 $d(25|L(0)=0)$ 을 추정한다면 총반복실험횟수는 다음과 같다.

$$n_r^*(0.15) = \min \left\{ i \geq 10 : \frac{t_{i-1, 0.95} \sqrt{3.172/i}}{1.982} \leq 0.15 \right\} = 99$$

따라서, 반복실험을 99번 하면 90% 신뢰도 $d(25|L(0)=0)$ 에서 15% 이상 벗어나지 않는 $d(25|L(0)=0)$ 의 추정치를 구할 수 있다.

작은 상대 정확도 γ 를 가지는 신뢰구간을 얻기 위해 식 (9.7)을 직접 사용할 때의 어려운 점은, $s^2(n)$ 이 모분산 $\sigma^2(X)$ 보다 크게 추정되었을 때, 반복 실험횟수 $n^*(\gamma)$ 가 실제로 필요한 횟수보다 아주 크게 계산될 수 있다는 점이다. 실제로 시뮬레이션을 행할 때 반복실험으로 인하여 발생하는 비용이 매우 크므로, 직접 식 (9.5)를 사용하는 것은 불필요한 비용을 발생시킬 수 있는 것이다. 지정된 상대 정확도를 가지는 신뢰구간을 설정하기 위해 실제로 필요한 횟수만큼의 반복실험을 행하는 연속적 방법(sequential procedure)에 대해 설명하겠다. X_1, X_2, \dots 가 IID 확률변수이고 정규분포가 아니어도 된다고 가정하자. 이 실험의 목적은 상대 정확도가 $\gamma(0 < \gamma < 1)$ 보다 작거나 같아지도록, μ 에 대한 $100(1-\alpha)\%$ 신뢰구간을 설정하는 것이다. 먼저 처음의 반복실험횟수를 $n_0 \geq 2$ 가 되도록 한다. 그리고 신뢰구간의 $\frac{1}{2}$ 길이를

$$\delta(n, \alpha) = t_{n-1, 1-\alpha/2} \sqrt{\frac{s^2(n)}{n}}$$

으로 나타낸다. 그러면 연속적인 실험의 절차는 다음의 단계를 따른다.

0. n_0 만큼의 반복실험을 행하고, $n = n_0$ 라고 놓는다.
1. X_1, X_2, \dots, X_n 으로부터 $\bar{X}(n)$ 과 $\delta(n, \alpha)$ 를 계산한다.
2. 만약 $\delta(n, \alpha) / |\bar{X}(n)| \leq \gamma$ 이 만족되면, μ 에 대한 $100(1-\alpha)\%$ 신뢰구간을

$$I(\alpha, \gamma) = [\bar{X}(n) - \delta(n, \alpha), \bar{X}(n) + \delta(n, \alpha)]$$

로 결정하고 실험을 끝낸다. 그렇지 않으면, n 을 $n+1$ 로 바꾸고 추가적인 반복실험을 행한 후, 단계 1로 간다.

이 방법에서는 매 반복실험 뒤에 $\sigma^2(X)$ 의 추정치를 새로이 계산해야 하고 또, 필요로 하는 총실험횟수는 확률변수가 된다는 것을 알 수 있다.

[예제 9.8] $\mu = 0.9$ 인 $M/M/1$ 대기행렬에 대하여, 상대 정확도가 0.15 보다 작거나 같

도록 $d(25;L(0)=0)$ 의 신뢰구간을 설정하려고 한다. $n_0=10$ 으로 하여, 다음의 결과를 얻는다.

$$n=68 \text{ (총반복실험횟수)}, \quad \bar{X}(68)=1.872$$

$$s^2(68)=1.865, \quad I(0.10, 0.15)=[1.592, 2.148]$$

이 결과를 예제 9.7과 비교해 보면 총반복실험횟수가 99에서 68로 줄어들었음을 알 수 있다.

5.3 실험방법의 선택

종로 시뮬레이션에 대해서, 고정 표본 크기 방법과 연속적 방법 중 어느 것을 사용할 것인가에 대하여 논하겠다.

신뢰구간의 정확도가 그렇게 중요치 않은 탐구적인 실험(exploratory experiment)에는 표본크기가 고정된 실험을 행할 것을 추천한다. 그러나 만약, X_j 들이 거의 정규분포가 아니고 반복실험의 수 n 이 작으면 표본크기가 고정된 실험에 의해 설정된 신뢰구간의 범위는 원하는 것보다 낮을 수 있다. n 번의 반복을 행하는 탐구적인 실험으로부터 반복실험당 비용과 X_j 들의 모분산을 추정할 수 있다. 그리고 식 (9.6)을 사용하여 $n_0^*(\beta)$ 를 계산할 수 있고, $n_0^*(\gamma)$ 는 식 (9.7)을 사용하여 계산할 수 있다. 때때로, β 와 γ 의 선정은 반복실험횟수에 관련된 비용에 의해 영향을 받는다. 만약 상대 정확도가 낮은 신뢰구간을 설정하려 한다면, $\gamma \leq 0.15$, $n_0 \geq 10$ 의 연속적인 실험을 행할 것을 추천한다. 그러나, 단일 상대 정확도 γ 가 0.15보다 높은 신뢰구간을 설정하려 한다면, 표본의 크기가 고정된 실험을 몇 회 연속적으로 행할 것을 추천한다. 또, 절대 정확도 β 가 낮은 신뢰구간을 설정하려 할 때는, 표본크기가 고정된 실험을 몇 회 연속적으로 행할 것을 추천한다.

5.4 초기조건의 선정

앞에서 언급했던 것과 같이 종로 시뮬레이션의 실험결과치는 시스템의 초기 상태에 크게 영향을 받는다. 따라서, 초기조건을 선정할 때에는 주의를 기울여야 한다. 예를 들어, 은행에서 가장 바쁜 시간인 12시에서 1시 사이에 고객

이 도착하여 대기할 때까지의 평균 대기시간을 추정하려 한다고 하자. 일반적으로 12시에는 은행이 불비기 때문에, 초기조건을 고객이 없다고 가정하고 12시부터 시뮬레이션을 시작한다면 평균 대기시간의 추정치는 정확하지 않을 것이다. 따라서, 여기에서는 이 문제에 대한 두 가지 발견적인 방법(heuristic approaches)을 소개하는 데, 일반적으로 첫번째 방법이 많이 사용된다.

첫번째 접근방법은 은행이 9시에 영업을 시작한다는 가정하에서 시뮬레이션을 9시부터 4시간 동안 실행한다. 그리고, 원하는 평균 대기시간을 추정할 때에는 12시부터 1시 사이의 자료만을 이용한다. 이 방법의 단점은 12시에서의 초기조건을 구하기 위해서 시뮬레이션된 9시부터 12시까지의 3시간이 낭비된다는 점이다.

두 번째 방법은, 며칠 동안 12시에 은행에 있는 고객수를 조사한다. 그리고 \hat{p}_i 를 12시에 $i(i=0,1,\dots)$ 명의 고객이 있는 비율이라 하자. 그리고 분포 $\{\hat{p}_i\}$ 로부터 임의로 초기조건을 뽑아서 12시부터 1시까지 시뮬레이션을 실행한다. 이때 12시에 서비스를 받고 있는 고객은 서비스를 시작한 것으로 가정한다. 1회 이상의 시뮬레이션을 행할 때에는 새로운 초기조건을 분포 $\{\hat{p}_i\}$ 로부터 뽑는다.

6 안정상태 시뮬레이션의 신뢰구간

Y_1, Y_2, \dots 를 1회의 시뮬레이션 과정에서 얻은 결과라고 하자. 그러면 $\{Y_i, i \geq 1\}$ 에 대한 안정상태의 평균반응(steady-state average response) ν 는

$$\nu = \lim_{m \rightarrow \infty} \frac{\sum_{i=1}^m Y_i}{m} \quad \text{w.p. 1}$$

로 정의된다. 그리고 극한치 ν 는 시뮬레이션의 초기상태에 대해 독립이라고 가정한다. ν 에 대한 신뢰구간을 설정하는 데에는 일반적으로 다음과 같은 두 가지 접근방법이 있다.

1. 고정 표본 크기 방법 fixed-sample-size procedures: 임의적으로 고

정된 길이의 시뮬레이션 과정을 실행한 후, 얻어진 자료로부터 신뢰구간을 설정한다.

2. 연속적 방법(sequential procedures) : 바람직한 신뢰구간이 설정될 때까지 시뮬레이션의 길이를 연속적으로 증가시킨다. 시뮬레이션을 언제 그칠 것인가에 대해서는 몇 가지 기법이 있다.

우리는 안정상태의 평균반응 ν 를 추정하는 데 있어서 산출과정 Y_1, Y_2, \dots 의 분석에 종료 시뮬레이션을 왜 사용할 수 없는가 하는 의문을 갖게 된다. 동일한 초기조건으로 행하여지는 독립적인 반복실험의 위험성과 안정상태의 종료 결과치의 추정에 종료 시뮬레이션 분석을 행함으로써 야기되는 위험성을 설명하기 위해서, $\rho=0.9$ 인 $M/M/1$ 대기행렬을 생각해 보자. 안정상태의 평균 대기시간 $d=8.1$ 을 추정한다고 가정하고 초기조건 $L(0)=0$ 에 대해서 길이 $m=320$ 인 n 번의 독립된 실험을 행한다. 그림 9.1에서 $E(X_j)=d(320|L(0)=0)=6.01$, $E[\bar{X}(n)]=6.01$ 이므로 $\bar{X}(n)$ 은 아무리 많은 반복실험을 행하더라도 d 에 대해 편중된 추정량이다. 더구나, n 이 커짐에 따라 식 (9.5)와 (9.8)에 의해 설정된 신뢰구간의 길이는 더욱 작아져서 신뢰구간의 범위가 결국은 0에 접근할 것이다. 따라서, 실질적으로는 d 가 아닌 $d(320|L(0)=0)$ 에 대한 신뢰구간을 추정하는 것이 되는 것이다.

그림 9.1로부터 왜 위의 종료 시뮬레이션 분석이 안정상태의 경우에 잘 적용이 안되는가 하는 것을 알 수 있다. 시스템의 안정상태가동을 나타내는 상태에서는 시뮬레이션을 시간 0에서 출발할 수 없기 때문에, 시뮬레이션의 초기 출력데이터는 안정상태의 평균반응 ν 에 대한 좋은 추정치가 아니다. 이러한 난점을 시뮬레이션 관련서적들은 출발문제(start-up problem) 또는 초기추이문제(problem of the initial transient)라고 한다. 이 문제를 해결하기 위해서 초기 데이터를 수집하기 이전에 처음 k 개의 데이터를 제거해 버리는 반복-제거 방법(replication-deletion approach)이 있다.

6.1 교정표본크기 방법

기본적으로 이 방법에 대하여 5가지 정도의 기법들이 연구되어 있으나, 비

교적 실제문제에 적용이 가능하고, 수행력이 높은 다음과 같은 2가지에 대하여 설명을 한다. 양기법 모두 출력 데이터 Y_1, Y_2, \dots 를 ν 에 대한 신뢰구간을 고전적 공식을 써서 구할 수 있도록 IID 관측치로 간주한다.

i) Batch Means 방법

$\{Y_i : i \geq 1\}$ 을 $E(Y_i) = \nu$ (모든 i 에 대하여)인 covariance stationary process라고 가정한다. 시뮬레이션 런(run)을 통해 얻은 m 개의 데이터 Y_1, Y_2, \dots, Y_m 을 크기가 l 인 n 개의 배치(batch)로 나눈다($m = n \cdot l$). 따라서 배치 1은 Y_1, Y_2, \dots, Y_l 로 구성이 되고, 배치 2는 $Y_{l+1}, Y_{l+2}, \dots, Y_{2l}$ 이 된다. $\bar{Y}_j(l)$ ($j = 1, 2, \dots, n$)을 l 개의 관측치로 구성이 되는 j 번째 배치의 배치평균(batch mean)이라 하고, $\bar{Y}(n, l) = \sum_{j=1}^n \bar{Y}_j(l) / n = \sum_{i=1}^m Y_i / m$ 을 총 표본평균(grand sample mean)이라고 하자. $\bar{Y}(n, l)$ 이 ν 의 점추정치로 사용된다.

그리고 $\bar{Y}_j(l)$ 들이 9.5.1항의 고정표본크기 방법에서 신뢰구간을 구하기 위해 사용된 X_j 들에 해당이 된다.

만약 배치 크기 l 을 충분히 크게 하면 $\bar{Y}_j(l)$ 들은 대략적으로 무상관(uncorrelated) 관계를 가진다. 또한 l 을 적절하게 취하여 $\bar{Y}_j(l)$ 들이 대략적인 정규분포를 한다고 가정하자. 그러면 다음의 공식을 이용하여 평균치 ν 에 대한 $100(1-\alpha)\%$ 신뢰구간을 구할 수 있다.

$$\bar{Y}(n, l) \pm t_{n-1, 1-\alpha/2} \sqrt{S_{\bar{Y}_j(l)}^2(n) / n} \tag{9.8}$$

여기에서 $S_{\bar{Y}_j(l)}^2(n) = \sum_{j=1}^n [\bar{Y}_j(l) - \bar{Y}(n, l)]^2 / (n-1)$

이다. 위 식 (9.8)은 앞의 식 (9.5)와 유사하다.

위 식(9.8)을 사용하는 데 발생가능한 다음의 3가지 오류가 있다.

- 1) 실제로 Y_1, Y_2, \dots 는 드물게 covariance stationary하다.
그러나 만약에 ν 가 존재한다면 아주 큰 k 에 대하여 Y_{k+1}, Y_{k+2}, \dots 는 대략적으로 covariance stationary하다.
- 2) 만약에 l 이 충분히 크지 않으면, $\bar{Y}_j(l)$ 들은 대략적인 정규분포를 하지 않을 수도 있다.
- 3) 만약에 l 이 충분히 크지 않다면, $\bar{Y}_j(l)$ 들은 높은 상관관계를 나타낼 수

도 있으며, $S_{\bar{Y}_j(t)}^2(n)$ 은 아주 편의(bias)된 $\sigma^2[\bar{Y}(n, t)]$ 의 추정량일 수도 있다. 실제로 Y_i 들이 양상관관계를 나타낸다면 $\bar{Y}_j(t)$ 들도 그러하여 낮게 편의된 분산 추정량을 구하게 되어 아주 좁은 신뢰구간을 얻게 된다.

Law의 연구에 의하면 $M/M/1$ 과 같이 단순한 시스템에서도 위의 3가지 가능한 오류들 중 3)이 제일 심각한 오류이며, 2)의 $\bar{Y}_j(t)$ 들의 비정규성을 배치 수, n 을 크게(40 이상) 함으로 해서 해결될 수 있다.

ii) 재 시작(Regenerative) 방법

이 방법의 기본 아이디어는, 시뮬레이션이 수행되는 확률 시스템(stochastic system)은 시스템이 수행되는 프로세스중에 임의의 점들에서 그 프로세스들이 확률적으로 새로이 시작된다(regeneration)는 사실에 기초를 두고 있다. 이 점들을 재시작점(예를 들면, 대기행렬 시스템에서는 idle, empty한 상태에 도착한 고객)라고 하며 이 점들 사이의 프로세스들을 독립적으로 생각하여 독립적인 확률변수의 값들을 구하여 앞에서 사용한 고전적인 공식을 사용하여 신뢰구간을 구한다.

출력 프로세스 $\{Y_i, i \geq 1\}$ 에 대하여 일련의 재시작점, $1 \leq B_1 < B_2 < \dots$ 들이 존재하며 각 B_j 에서 시작되는 프로세스는 서로 독립이라고 하자. 두 개의 인접한 재시작점 사이에 있는 프로세스의 부분을 재시작 사이클이라고 부른다. 각 사이클들이 서로 독립된 IID 반복들을 나타낸다.

$Z_i = i$ 번째 사이클의 해당되는 값(예를 들면, i 번째 사이클에 있는 고객들의 대기시간들의 합)

$L_i = i$ 번째 사이클의 길이(예를 들면, 고객의 수 또는 사이클에 해당되는 시간의 길이)

$X_{i,k} = i$ 번째 사이클의 k 번째 관측치

$n =$ 수행된 총 사이클 수

$m =$ 시뮬레이션을 통해 얻은 관측치의 총 수

라고 두면

$$Z_i = \sum_{k=1}^{L_i} X_{ik}$$

$$\sum_{i=1}^n L_i = m$$

이 된다. 시뮬레이션을 통해 얻은 관측치의 총 평균 $\bar{Y}(n)$ 은

$$\bar{Y}(n) = \frac{\sum_{i=1}^n \sum_{k=1}^{L_i} X_{ik}}{m} \quad (9.9)$$

이 되며, 다음과 같이도 나타낼 수 있다.

$$\bar{Y}(n) = \frac{\sum_{i=1}^n \sum_{k=1}^{L_i} X_{ik}}{m} = \frac{\sum_{i=1}^n Z_i}{\sum_{i=1}^n L_i} = \bar{Z}/\bar{L} \quad (9.10)$$

각 사이클에 포함되어 있는 관측치의 수는 확률변수이므로 총 사이클수와 총 관측치의 수 모두를 구체적으로 나타낼 수 없으므로 $\bar{Y}(n)$ 의 분산에 대한 추정치는 다음의 비율 추정량(ratio estimator)을 사용하여 구할 수 있다.

$$S_{\bar{Y}}^2 = S^2 / (\bar{L}\sqrt{n})^2 \quad (9.11)$$

여기서 $S^2 = S_Z^2 - 2\bar{Y}(n)S_{ZL} + (\bar{Y}(n))^2 S_L^2$

$$S_Z^2 = \frac{1}{n-1} \sum_{i=1}^n (Z_i - \bar{Z})^2$$

$$S_L^2 = \frac{1}{n-1} \sum_{i=1}^n (L_i - \bar{L})^2$$

$$S_{ZL} = \frac{1}{n-1} \sum_{i=1}^n (Z_i - \bar{Z})(L_i - \bar{L})$$

이다.

위의 식(9.9), (9.10), (9.11)의 추정량들은 편의된 추정량(biased estimator)이므로 다음의 제크 나이프(Jackknife) 추정량 공식을 이용하기도 한다. \bar{J} 를 표본 평균의 제크나이프 추정량이라고 하면 \bar{J} 는 다음 식(9.12)를 이용하여 구한다.

$$\bar{J} = \frac{1}{n} \sum_{i=1}^n J_i \quad (9.12)$$

여기에서
$$J_i = n \cdot \bar{Y}(n) - (n-1) \sum_{j=1}^n Z_j / \sum_{j=1}^n L_j$$

이다. 위에서 구한 J_i 들은 IID 분포를 하며, \bar{J} 와 S_J/\sqrt{n} 및 다음 공식을 사용하여 신뢰구간을 구할 수 있다.

$$\bar{J} \pm t_{n-1, 1-\alpha/2} \cdot S_J/\sqrt{n} \quad (9.13)$$

위에서 기술한 재시작(regenerative) 방법이 사용되어지기 위해서는, 주어진 시스템에 대한 적절한 재시작점을 발견해야 하고 시뮬레이션을 통해 얻을 수 있는 총 사이클 수인 n 의 값이 너무 작아서는 안된다. 이러한 점들이 위의 방법이 일반적으로 모든 시스템에 적용되지 못하는 이유들이다.

안정상태 시뮬레이션의 신뢰구간 작성을 위한 연속적 방법에 대하여서도 이론상의 어려움과 실제 사용상의 힘든 점으로 인하여 설명을 생략하고 다음 항에서는 일반적으로 많이 사용하는 반복-제거 방법(replication-deletion approach)에 대하여 기술한다.

6.2 반복-제거 방법(replication-deletion approach)

각 m 개의 관측치들을 가지는 총 n 번의 반복(replication) 또는 런(run)을 수행하여 다음과 같이 관측치를 구했다고 하자.

$$\begin{array}{c} Y_{11}, Y_{12}, \dots, Y_{1m} \\ \dots \dots \dots \dots \\ Y_{n1}, Y_{n2}, \dots, Y_{nm} \end{array}$$

여기에서 Y_{ij} 는 j 번째 반복에서의 i 번째 관측치이다.

각 반복에서 안정상태를 나타내지 않는다고 생각되는 앞 k 개의 관측치($0 < k < m$)를 제거하고 반복 및 총 평균치는 다음과 같이 각각 정의하자.

$$\bar{Y}_j(m, k) = \sum_{i=k+1}^m Y_{ji} / (m-k) \quad (9.14)$$

$$\bar{Y}(n, m, k) = \sum_{j=1}^n \bar{Y}_j(m, k) / n \quad (9.15)$$

위 식(9.14), (9.15)를 사용하여 평균 ν 에 대한 신뢰구간을 구할 수 있다.

위 방법을 실제로 적용하기 위해서는 어떻게 적절한 k 값을 구하느냐는 데 있다. 일반적으로 통용되는 방법으로는 많은 반복을 시행하기전 1번의 시험 런(pilot run)을 행하여 그래프를 그려본 후 적당한 k 값을 정한 후 나머지 반복시행에도 같은 값의 k 를 적용하여 분석을 행하며, 신뢰구간을 구한다.

7 결 론

시뮬레이션 결과 분석에 관한 대부분의 논문과 서적은 안정상태의 경우에 대해서만 다루고 있다. 그래서 독자는 안정상태 시뮬레이션만이 중요하다고 생각할 것이다. 그러나, 이것은 대부분의 논문과 서적이 일반적으로 안정상태 분석만이 가능한 수학적 대기행렬이론을 취급함으로써 생긴 결과일 것이다. 따라서 종료 시뮬레이션도 역시 중요하다. 많은 시뮬레이션 실무자들은 현실의 시뮬레이션들은 거의 모두가 종료 시뮬레이션에 속한다고 주장한다. 왜 안정상태분석이 적절하지 못한가 하는 이유는 다음과 같다.

1. 고려하고 있는 시스템은 물리적으로 종료한다. 이 경우, 시뮬레이션의 길이를 임의로 길게 하는 것은 의미가 없다.
2. 시스템에 대한 분포들은 시간에 따라 변한다. 이 경우, 안정상태의 실험 결과치는 존재하지 않을 것이다.
3. 안정상태의 실험 결과치가 존재한다고 해도 우리는 오히려 그 시스템의 일시적인 행동을 연구해야 할 경우가 많다.

종료 시뮬레이션의 결과를 분석할 때 가장 중요한 사항은 반복실험 j 로부터 얻은 X_j 가 우리들이 추정하고자 하는 그 무엇을 대표하고 있어야 한다. 종료 시뮬레이션분석은 각 반복실험에서 얻은 관측치가 IID라는 장점이 있고, 따라

서 신뢰구간을 설정하는 절차가 상대적으로 간단하다.

안정상태 시뮬레이션의 경우, 신뢰구간을 설정하는 절차는 상당히 복잡하다. 안정상태의 실험결과치를 추정하기 위해 긴 시뮬레이션 과정을 실행하게 되면 인위적 초기조건이 단 한 번밖에 실험되지 않기 때문에 출발문제는 그렇게 중요치 않다. 종료분석이나 안정분석 중 어느 방법을 적용해야 좋을지 분명하지 않을 경우가 있는데, 이는 처한 상황에 따라 어떠한 방법을 택해도 알맞은 것이라 생각한다.