

## 시뮬레이션 출력의 효율적인 분석을 위한 프로크루스테스 기법의 응용\*

### Application of Procrustes Analysis Method for Efficient Analysis of Simulation Outputs

이영해\*\*, 박경종\*\*, 문기석\*\*\*

Young-Hae Lee\*\*, Kyoung-Jong Park\*\*, Kee S. Moon\*\*\*

#### Abstract

Output analysis is one of the most important fields of simulation to achieve the accurate simulation results. This study shows how to analyze simulation output data in the steady state using Procrustes analysis technique which has not been used in the field of simulation yet. In this paper Procrustes analysis method is used to perform the analysis of simulation output efficiently and effectively by applying the improved version of the method. The experiments are conducted using M/M/1 queueing simulation model. The results obtained by Procrustes analysis method show better estimates for average waiting times and average queue lengths which are closer to true values and narrower confidence intervals than when replication-deletion method is used. Also it requires the smaller number of simulation runs.

#### 1. 서론

과학 문명과 통신, 교통 등의 발달로 시스템은 국가적인 범위를 벗어나 세계화를 지향

하게 됨으로써, 실제로 인간이 다루게 되는 문제는 보다 광범위하고, 복잡성을 띠게 된다. 이 같은 문제들을 분석하고 해결하기 위한 가장 좋은 방법이 모델링과 시뮬레이션에

\* 본 논문은 1994년도 한양대 산업과학연구소 지원 과제임

\*\* 한양대학교 산업공학과

\*\*\* Dept. of Mechanical Engr. and Engr. Mechanics, Michigan Technological University

의한 분석 방법이다. 따라서 오늘날과 같은 산업사회나, 이미 다가온 정보화 사회에서 중요한 시스템의 개발이나 정책결정을 함에 있어서 컴퓨터에 의한 모델링과 시뮬레이션은 필수적인 접근 방법이 되고 있다. 특히, 각종 공학분야는 물론 시뮬레이션에 의한 실험을 해야 하는 자연과학 분야에서 부터, 의사결정 내지는 정책결정이 요구되는 경영학 및 경제학을 비롯한 사회과학 분야 전반에 걸쳐서 모델링과 시뮬레이션에 의한 분석 방법은 점차 그 활용도가 높아지고 다양해지고 있다 [1, 7].

지금까지 시뮬레이션 연구에서, 모델 개발이나 프로그래밍에는 많은 양의 비용과 시간이 소모되었으나 시뮬레이션 결과의 분석에 대해서는 연구가 상대적으로 많지 않았다. 사실 많은 경우의 시뮬레이션 실행의 일반적인 형태는, 어떤 임의의 시간 동안 시뮬레이션을 한 번 행하여 거기서 산출되는 시뮬레이션 추정치를 연구 대상으로 주어진 시스템에 대한 참값으로 취급하는 것이다. 이들 추정치들은 큰 분산을 갖는 확률변수이기 때문에 어떤 특정 시뮬레이션 과정에서 이 추정치들은 참값들과 크게 다를 수가 있다. 이런 경우 연구중인 시스템에 대한 잘못된 추론을 할 가능성이 농후하다.

안정상태(steady state) 시뮬레이션의 출력 데이터는 변이 상태(transient state)의 존재, 데이터의 종속성(dependency), 정규성(normality)의 부재 등으로 인하여 점 추정치의 편위(bias)와 구간 추정치(분산)의 변동으로 정확한 분석이 힘들다. 따라서, IID(independent identically distributed) 관측치에 기초한 고전적 통계분석이 직접적으로 적용될 수 없고,

있다고 해도 적용하기가 어렵다[2, 5, 7, 10].

본 논문에서는 시뮬레이션 출력 분석에 있어서 지금까지 알려진 방법과는 다른 프로크루스테스 분석 기법을 사용하여 시뮬레이션 출력 데이터의 안정상태 모수의 추정치를 작은 편위 및 분산, 적은 수의 반복 수행 횟수를 필요로 하며, warm-up 기간의 데이터를 소거할 필요도 없고, 1가지 이상의 모수를 동시에 추정할 수 있는 방법을 제시하며, M/M/1 대기 모형을 대상으로 시뮬레이션을 수행하여 결과를 비교하였다.

## 2. 기존 방법

시뮬레이션 대상 시스템의 안정상태 추정치에 대한 통계적 출력 분석 방법은 반복-제거 방법(replication-deletion method), 배치 평균(batch means), 자동 회귀(autoregressive), 스펙트럼 분석(spectrum analysis), 재발생(regenerative) 방법과 표준 시계열(standardized time series) 방법으로 나누어 진다[7]. 이러한 방법들 중에서 반복-제거 방법이 적용하기가 쉬워서 가장 많이 사용되고 있으므로, 본 논문에서 제시하는 프로크루스테스 분석 기법과의 비교대상으로 삼는다.

반복-제거 방법을 자세히 설명하면, 각  $n$  개의 관측치들을 갖는 총  $m$  번의 반복(replication) 또는 런(run)을 수행하여, 각 반복에서 안정상태를 나타내지 않는다고 생각되는 warm-up 기간의  $l$  개 관측치( $0 \leq l < n$ )를 제거하고, 각 반복의 평균치를 구하여 이 평균치를 사용하여 모평균에 대한 신뢰구간을 구한다. 본 방법을 실제로 적용하기 위해서는 어떻게 적절한  $l$  값을 구하는가가 매우

중요하다. 일반적으로 통용되는 방법으로는 많은 반복을 시행하기 전에 1번의 시험 런 (pilot run)을 행하여 그래프를 그려서 적당한  $l$  값을 정한 후 나머지 반복 시행에도 같은  $l$  값을 적용하여 신뢰구간을 구한다.

### 3. 프로크루스테스 분석 기법

프로크루스테스 분석 기법의 이름은 여행자들을 잡아다가 침대보다 크면 몸을 잘라서 침대에 맞추고, 침대보다 작으면 팔 다리를 늘여서 침대에 맞추는 그리스 신화의 여관집 주인인 프로크루스테스에서 유래되었다. 이 신화에서와 같이 프로크루스테스 분석 기법은 의견일치에 도달하기 위하여 수집된 데이터 집합을 어떻게 조합하고 조작하는가를 어느 정도 설명해 준다. 프로크루스테스 분석 기법은 원래 인문과학 분야에서 사물을 보고 의견이 각자에 따라 달리 나타나는 경우 의견을 종합하고 일치하게 하기 위한 기법으로 개발되었으나[3, 6], 최근에는 제조 공정에 있어서의 센서 합성(sensor integration)[9]과 이미지 프로세싱 분야[8]에 이용하기 위하여 적용되었다. 이미지 프로세싱 분야에서 비전 카메라로 어떤 물체의 형상을 인식할 때 카메라 자체의 결합과 조명의 밝기, 카메라의 각도, 공기중의 이물질, 인식하고자 하는 물체의 빛에 의한 반사 등으로 인하여 물체의 형상을 제대로 인식할 수 없을 때, 여러대의 카메라를 각각 다른 각도에서 촬영하여 각 카메라로부터 왜곡된 데이터를 합성하여 원래 물체의 형상을 유추해내는 데 프로크루스테스 기법이 이용되었다.

시뮬레이션 출력 데이터도 다수의 반복에

의한 랜덤성을 포함하고 있고 추정하고자 하는 모수의 참값 방향으로의 데이터의 변환은 점 추정치의 편의를 줄이고 구간 추정치의 크기를 감소할 수 있으리라는 생각에 프로크루스테스 분석 기법을 시뮬레이션 출력 분석에 이용하게 되었다.

프로크루스테스 분석 기법을 시뮬레이션 출력 분석에 이용하기 위하여 예를 들어 기법의 기본 개념을 설명한다.  $m$ 번 반복 시뮬레이션에서 각각의 시뮬레이션 반복에 대해서 상태변수  $p=2$ (예를 들어, 대기행렬시스템에서 대기열에서 고객의 평균 대기 시간과 평균 대기 길이)인 데이터를  $n$ 개씩 수집한다고 가정하자. Fig. 3.1은  $m=3$ (3번의 시뮬레이션),  $n=5$ (5개의 관측치),  $p=2$ (2개의 상태변수)에 대한 경우를 나타내고 있다. 행렬  $X_i$ 는  $i$ 번째 시뮬레이션에서의 상태변수 데이터를 포함하며, Configuration이라 한다.  $X_i$ 에서 행의 수는  $n$ 과 같고, 열의 수는  $p$ 와 같다.  $X_i$ 에서 각각의 행은  $p$ 좌표축에서의 점으로 정의될 수 있다.

프로크루스테스 분석은 여러 개의 Configuration 데이터로부터 추정하고자 하는 모수의 값에 가까운 값을 얻기 위하여 Translation, Rotation, Scaling Transformation을 행한다. ( $n \times p$ )행렬인  $X_i$ 의 Rotation은 ( $p \times p$ )인 Orthogonal Rotation matrix  $H_i$ 를 곱해서 얻을 수 있다. Uniform Scaling은 스칼라( $\rho_i$ )를  $X_i$ 에 곱해서 얻을 수 있다. Translation은  $X_i$ 의 각각의 행에 대해서 같은 ( $1 \times p$ )행 벡터를 더하거나,  $X_i$ 에 동일한 행 벡터를 포함하는 행렬  $T_i$ 를 더해서 얻을 수 있다. Translation, Rotation과 Scaling Transformation이  $X_i$ 에 적용된 결과인  $X_i^*$ 는 식 (3.1)을 사용하여 구할

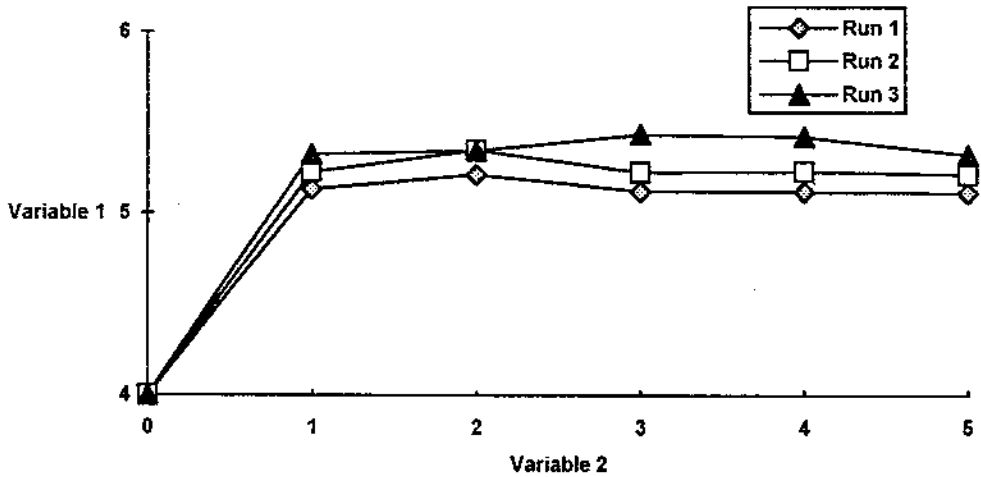


Fig. 3.1 Illustration of individual configurations(m=3, n=5, p=2)

수 있다[3, 8, 9].

$$X_i^* = \rho_i X_i H_i + T_i \tag{3.1}$$

프로크루스테스 분석 기법의 어려운 점은 식 (3.1)의  $\rho_i$ ,  $H_i$ 와  $T_i$ 를 선택하는 문제이다. 식 (3.2)로 표시되는 Residual Sum of Squares 는 최소화되어야 한다.

$$S = \sum_{i,j}^m tr\{[(\rho_i X_i H_i + T_i) - (\rho_j X_j H_j + T_j)] \cdot [(\rho_i X_i H_i + T_i) - (\rho_j X_j H_j + T_j)]\} \tag{3.2}$$

S를 최소화하기 위해 Scaling  $\rho_i$ 값이 0이 되 게 선택한다면, S가 0으로 최소화되는 문제 점이 있다. 즉, 우리가 추정하고자 하는 목표 치에 도달하지 못하고 항상 같은 값으로 고정된다. 이러한 문제점을 피하기 위해 Gower [3]는 식 (3.3)의 제한식을 만들었다.

$$\sum_{i=1}^m \rho_i^2 tr(X_i X_i') = \sum_{i=1}^m tr(X_i X_i') \tag{3.3}$$

이 제한식은 원점(origin)에 대해서 데이터 의 Sum of Squares 값이 바뀌지 않도록 해준다. 또한 데이터들이 다른 값을 갖도록 하기 위해서  $X_i$ 는 식 (3.4)와 같이 초기화 된다.

$$\sum_{i=1}^m tr(X_i X_i') = m \tag{3.4}$$

Translation matrix  $T_i$ 에 대해서, 식 (3.2)를 미분한 후 이 식이 0이 되도록 하는 것은 모든 m개의 Configuration이 같은 중심점으로 이동하도록 해준다. 이러한 과정을 만족시키기 위해 n개의 동일한 행벡터로 구성된  $T_i$ 가 형성될 수 있다.

Rotation에 대해서는  $H_i$ 가 타당한 Rotation matrix가 되어야 한다. 따라서 이것을 확신시키기 위해 다음 식(3.5)의 제한식을 우리가 최소화시키고자 하는 식 (3.2) 에 대입한다. 그리고 Residual Sum of Squares를 만족하는  $H_i$ 와  $\rho_i$ 를 찾는다.

$$H_i H_i' = I \tag{3.5}$$

본 연구에서 사용되는 용어를 다시 정리하면 다음과 같다.

$m$  : 시뮬레이션 반복 횟수

$n$  : 각각의 시뮬레이션 반복에서 얻어진 관측치의 갯수

$p$  : 추정 대상 상태변수의 갯수

$X_i$  :  $i$ 번째 시뮬레이션의 상태변수의 데이터를 포함하는 행렬

$\rho_i$  :  $i$ 번째 시뮬레이션의 Scaling factor

$T_i$  :  $i$ 번째 시뮬레이션의 Translation matrix

$H_i$  :  $i$ 번째 시뮬레이션의 Orthogonal Rotation matrix

$Y$  :  $(n \times p)$ 인 합성 행렬(consensus matrix)

$S$  : Residual Sum of Squares

$E_i$  :  $(X_i'YY'X_i)^{-1}$ 의 고유치(eigen values)를 포함하는 Diagonal matrix

$V_i$  :  $(X_i'YY'X_i)^{-1}$ 의 고유벡터(eigen vectors)를 포함하는 행렬

프로크루스테스 분석 기법의 알고리즘은 다음의 단계 1로 부터 단계 10까지로 구성된다.

단계 1: 각각의  $X_i$  ( $i=1, 2, 3, 4, \dots, m$ )를 원점에 대해서 이동(translate)시킨다. 즉, 열 요소의 평균을 구해서 각각의 행에서 뺀 새로운  $X_i$ 를 구한다.

단계 2:  $\sum_{i=1}^m \rho_i^2 tr(X_i X_i') = m$ 를 만족시키는 초기  $\rho_i$ 를 구하기 위해 식 (3.6)를 이용한다. 이때 구한  $\lambda$ 를  $\rho_i$ 로 사용한다.

$$\sum_{i=1}^m \lambda^2 tr(X_i X_i') = m \quad (3.6)$$

$X_i = \rho_i X_i$ 를 구하여 “초기” Configurations으로 한다.

단계 3: 다음 식을 사용하여  $X_i$  요소들의 평균인 “초기”  $Y$ 를 구한다.

$$Y = \sum_{i=1}^m X_i / m$$

초기 Scaling coefficient  $\rho_i$ 를 1로 정하고, Rotation matrix  $H_i = I$  (identity matrix)로 하며, 식 (3.7)을 이용하여 Residual Sum of Squares를 계산한다.

$$S_R = \sum_{i=1}^m tr[(Y - \rho_i X_i H_i)(Y - \rho_i X_i H_i)'] \quad (3.7)$$

단계 3이 완료되면, 알고리즘에 의해서 필요한 초기화 절차는 끝난다.

단계 4: 앞 단계에서 구한 Scaling factor  $\rho_i$ , Rotation matrix  $H_i$ 와 초기 Configuration  $X_i$ 에 대한 식 (3.8)를 사용하여 현재의 Configuration  $X_i^*$ 를 구한다.

$$X_i^* = \rho_i X_i H_i \quad (3.8)$$

식  $H_i = V_i E_i^{1/2} V_i' X_i' Y$ 을 사용하여 현재의 Configuration  $X_i^*$ 에 대해서 최상의 Rotation matrix  $H_i^*$ 를 구한다. 따라서 새로운 최상의 Configuration은  $X_i^* H_i^*$ 가 된다. 또한 새로운 Rotation matrix는  $H_i = H_i H_i^*$ 가 된다.

단계 5: 현재의  $X_i$ 들의 평균으로 새로운 합성치  $Y$ 를 구한다.

단계 6: 식 (3.9)를 이용하여 새로운 최상의 Scaling coefficient  $\rho_i^*$ 를 구한다.

$$\rho_i^* = \left[ \frac{\text{tr}(\rho_i X_i H_i Y)}{\sum_{i=1}^m \text{tr}(X_i X_i')} \right] \left\{ \frac{m \cdot \text{tr}(Y Y')}{\text{tr}(X_i X_i')} \right\}^{1/2} \quad (3.9)$$

새로운 Configuration  $X_i^*$ 는  $\rho_i^* X_i H_i$ 가 되고, Scaling factor는  $\rho_i = \rho_i^*$ 로 갱신된다.

단계 7: 현재의  $X_i^*$ 들의 평균으로 새로운 합성치  $Y$ 를 구한다. 식 (3.7)을 사용하여 새로운 Residual Sum of Squares  $S_R^*$ 를 구한다.

단계 8:  $S_R$ 과  $S_R^*$ 의 차이가 허용되는 수렴 (convergence) 값보다 작으면 프로크루스테스 알고리즘을 종료한다. 그렇지 않으면,  $S_R = S_R^*$ 로 정하고 단계 4로 가서 다시 알고리즘을 수행한다.

단계 9: 단계 1에서 빼 준 각각의 열 평균을 다시 데이터에 더한  $X_i$ 를 구한다.

단계 10: 새로운 합성치  $Y$ 를 구한다. 이때 구해진  $Y$ 값이 프로크루스테스 분석을 수행한 후의 임의의 시뮬레이션 반복내에서 평균 관측치들이 된다. 얻어진  $Y$ 의 값으로 추정 대상 모수에 대한 평균, 분산과 신뢰구간을 구한다.

#### 4. 실험 및 평가

본 연구에서 대상으로 하는 모델은 M/M/1 대기모형이며, 이 모델을 가지고 일반적으로 사용하기가 쉽고, 많이 사용하는 반복-제거 방법을 사용하여 안정상태에서 얻은 데이터와 프로크루스테스 분석 기법을 이용하여 얻은 출력 데이터를 비교하였다. 반복-제거 방법과 프로크루스테스 분석 방법의 시뮬레이션 종료 시간은 같고, 반복-제거 방법은 안정상태에 시스템이 도달한 이후 데이터를 수

집하고, 프로크루스테스 분석 기법은 warm-up 기간 동안의 데이터를 제거하지 않고 모든 데이터를 수집하였다. 위와 같은 모형을 설정하여 서버 이용률( $\rho$ )을 바꿔가면서 시뮬레이션을 수행하였는데, 이용률은 0.2, 0.3, 0.4, 0.5, 0.67, 0.75, 0.9가 되도록 도착시간 간격( $1/\lambda$ )을 30분으로 하고 서어비스 시간 ( $1/\mu$ )을 적절히 선택한다. 또한 시뮬레이션을 15,000개의 개체가 시스템을 빠져나갈 때까지 수행하여 transient 기간으로 충분하다고 판단되는 10,000개의 개체가 시스템에 도착한 후에는 시스템이 안정상태에 도달하는 것으로 가정하였다. 각각의 시뮬레이션 실험에서 사용된 반복 실행 횟수는 프로크루스테스 분석 기법이 3회이며, 반복-제거 방법은 10회를 수행하였다.

M/M/1 대기모형의 이론치를 계산하기 위해, 대기열의 평균대기시간  $W_q$ 와 평균대기길이  $L_q$ 는 다음의 식 (4.1)과 (4.2)를 사용할 수 있다.

$$W_q = \rho / (1 - \rho) \mu \quad (4.1)$$

$$L_q = \rho^2 / (1 - \rho) \quad (4.2)$$

식 (4.1)을 사용하여 평균대기시간을 구해보면,  $\rho=0.5$ (도착시간 간격이 30분, 서비스 시간이 15분)인 경우  $W_q$ 는 15분이 된다. 식 (4.2)를 이용하여 평균대기길이를 구하면,  $L_q$ 는 0.5가 얻어진다.

반복-제거 방법과 프로크루스테스 분석 기법을 비교하기 위해, 먼저  $\rho=0.5$ 인 M/M/1 모델을 시뮬레이션하였다. Table 4.1은 반복-제거 방법을 사용하여 10회의 런을 수행하여

얻은 평균치이며, Table 4.2, 4.3, 4.4는 프로크루스테스 분석을 수행하기 위해서 수집한 각각의 런에 대한 초기 데이터이다. 초기 데이터를 수집하는 방법은 시뮬레이션이 종료할 때까지의 구간을 동일한 크기의 k개로 나누어 각 구간에 대한 평균치를 얻는다. 구간의 수 k를 결정하는데는 여러가지 방법이 있으나 Fishman[10]은 구간 데이터들의 종속성과 정규성을 고려하여 k가 8이상인 경우 권장하였으며, 본 연구에서는 이를 고려하여 구간을 10개로 나누었다. 이 구간에서 얻어진 평균치를 각 구간의 대표치로 간주하여 m=3개의 반복 실험 데이터에 대하여 프로크루스테스 분석 기법을 수행하였다. 각 구간에서의 평균치 데이터가 독립적으로 m개가 얻어지므로 구간 데이터들간에 종속성이 어느 정도 보정될 수 있으며 데이터의 전체적인 정규성도 k=10으로 어느 정도 보장된다고 생각된다. Table 4.5는 Table 4.2, 4.3, 4.4의 데이터를 이용하여 프로크루스테스 분석 기법의 알고리즘을 적용하여 얻은 변형된 시뮬레이션 출력 데이터이다.

실험을 통해서 나온 결과를 살펴보면, 반복-제거 방법을 사용해서 나온 결과의 평균은 대기열의 평균대기시간이 14.8691이고, 평균대기길이는 0.4967이다. 따라서, 이론적인 평균에서 벗어난 정도는 평균대기시간이  $((14.8691-15)/15) \times 100 = -0.87\%$ 이고, 평균대기길이는  $((0.4967-0.5)/0.5) \times 100 = -0.66\%$ 이다. 90%의 신뢰구간은 평균대기시간이 [14.4628, 15.2753]이고, 평균대기 길이는 [0.4809, 0.5125]이 된다.

프로크루스테스 분석기법을 이용한 알고리즘을 사용할 경우에는 대기열의 평균대기시간

**Table 4.1 Output data for replication-deletion method**

RUN NUMBER	AVERAGE WAITING TIME	AVERAGE QUEUE LENGTH
1	14.639	0.482
2	14.724	0.496
3	14.297	0.472
4	16.022	0.540
5	14.144	0.456
6	14.892	0.495
7	14.126	0.485
8	15.442	0.521
9	15.931	0.535
10	14.474	0.485

**Table 4.2 Initial data for Procrustes analysis from first run**

DATA	AVERAGE WAITING TIME	AVERAGE QUEUE LENGTH
1	14.813	0.497
2	14.952	0.502
3	14.819	0.497
4	14.846	0.499
5	14.633	0.490
6	14.836	0.495
7	14.866	0.496
8	15.023	0.501
9	14.965	0.499
10	15.183	0.507

이 14.9695이고, 평균대기길이는 0.5005의 평균을 갖는다. 따라서 이론적인 평균에서 벗어난 정도는 평균대기시간이  $((14.9695-15)/15) \times 100 = -0.2\%$ 이고, 평균대기길이는  $((0.5005-0.5)/0.5) \times 100 = +0.1\%$ 이다. 90%의 신뢰구간은 대기열의 평균대기시간이 [14.8336, 15.1054]이고, 평균대기길이는 [0.4989, 0.5021]이 된다.

**Table 4.3 Initial data for Procrustes analysis from second run**

DATA	AVERAGE WAITING TIME	AVERAGE QUEUE LENGTH
1	15.488	0.519
2	15.571	0.522
3	15.512	0.518
4	15.287	0.511
5	15.317	0.512
6	15.332	0.512
7	15.423	0.516
8	15.323	0.513
9	15.450	0.519
10	15.281	0.51

**Table 4.5 Resulting data by Procrustes analysis**

RUN NUMBER	AVERAGE WAITING TIME	AVERAGE QUEUE LENGTH
1	15.584	0.518
2	15.338	0.510
3	15.257	0.510
4	15.165	0.507
5	14.934	0.499
6	14.762	0.492
7	14.643	0.489
8	14.830	0.496
9	14.745	0.493
10	14.844	0.498

**Table 4.4 Initial data for Procrustes analysis from third run**

DATA	AVERAGE WAITING TIME	AVERAGE QUEUE LENGTH
1	14.636	0.489
2	14.723	0.492
3	14.778	0.494
4	14.862	0.498
5	14.644	0.490
6	14.498	0.484
7	14.505	0.483
8	14.664	0.489
9	14.431	0.479
10	14.424	0.479

**Table 4.6 Results by replication-deletion method and Procrustes analysis method( $\rho=0.5$ )**

	VARIABLE	AVERAGE	VARIANCE	CONFIDENCE INTERVAL	AVERAGE DIFFERENCE(%)
RM	AWT	14.869	0.4812	[14.463,15.275]	- 0.87
	AQL	0.4967	0.0074	[0.4809,0.5125]	- 0.66
PA	AWT	14.970	0.0550	[14.834,15.105]	- 0.20
	AQL	0.5005	0.0001	[0.4989,0.5021]	+ 0.10

RM : Replication-deletion method, PA: Procrustes analysis method  
 AWT: Average waiting time, AQL: Average queue length

위와 같은 결과를 정리하면 Table 4.6과 같다.

Table 4.6을 살펴보면, PA방법이 RM방법보다 평균대기시간의 이론치인 15와 평균대기길이의 이론치인 0.5에 더 가깝게 나타나며 분산을 비교할 경우에는 PA방법이 RM방

법보다 월등하게 좋아짐을 알 수 있다.  $\rho$  값의 변화에 대해 제안된 기법의 효과를 보기 위하여, 각 경우에 대하여 10번의 실험을 실행하였다. 평균대기 시간에 대해서, 반복-제거 방법과 프로크루스테스 분석 방법을 적용한 결과를 이론치로 부터의 차이를  $\rho$ 가 0.2, 0.5, 0.75와 0.9인 경우를 그림으로 그려 보면 다음의 Fig. 4.1, 4.2, 4.3과 4.4로 나타난다.

Fig. 4.1에서 이용률이 0.2인 경우를 살펴보



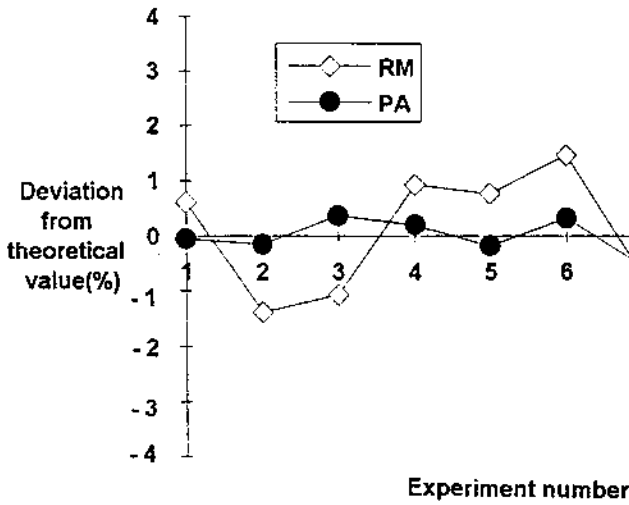


Fig 4.1 Experimental results for average waiting time when  $\rho=0.2$

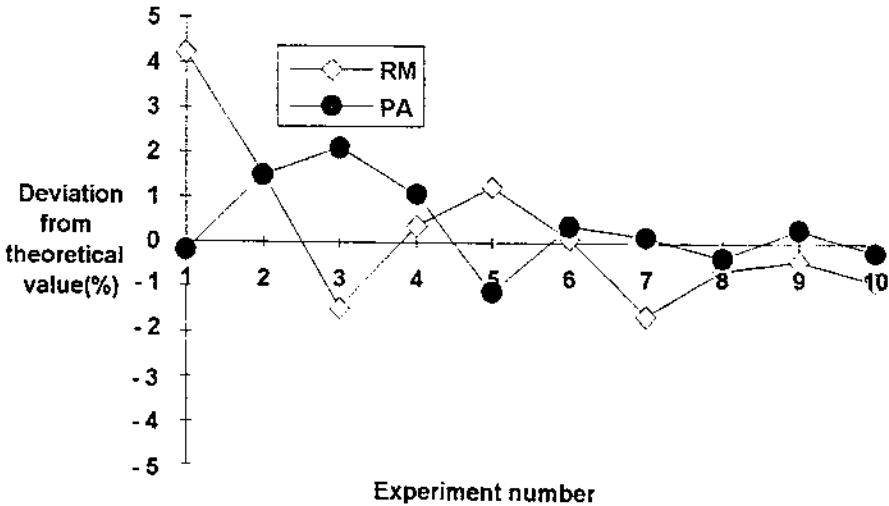


Fig 4.2 Experimental results for average waiting time when  $\rho=0.5$

면, 프로크루스테스 분석기법이 8번째 실행을 제외하고는 이론치에 더 가까우며 이용률이 0.5인 Fig. 4.2를 살펴보면 3번째 실행을 제외하고는 프로크루스테스 분석 기법이 이론치에 더 가까움을 알 수 있다. 또한 이용률이 0.75와 0.9인 경우에도 프로크루스테스

분석기법이 더 우수함을 알 수 있다. 따라서 두 기법의 결과치를 가지고 서로 비교해 보면, 본 논문에서 제안한 새로운 방법이 기존 방법인 반복-제거 방법보다 평균치의 비교에서는 훨씬 우월함을 알 수 있다. 구한 추정치들의 이론치로부터 절대 편차(%)들의 평

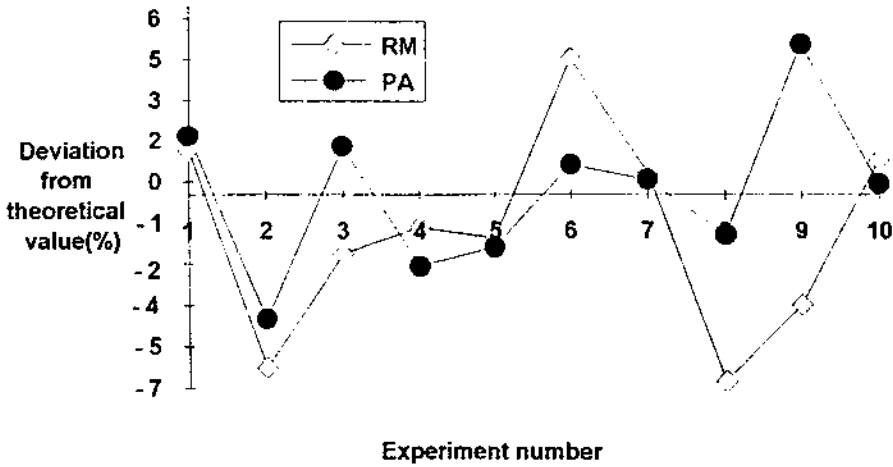


Fig. 4.3 Experimental results for average waiting time when  $\rho=0.75$

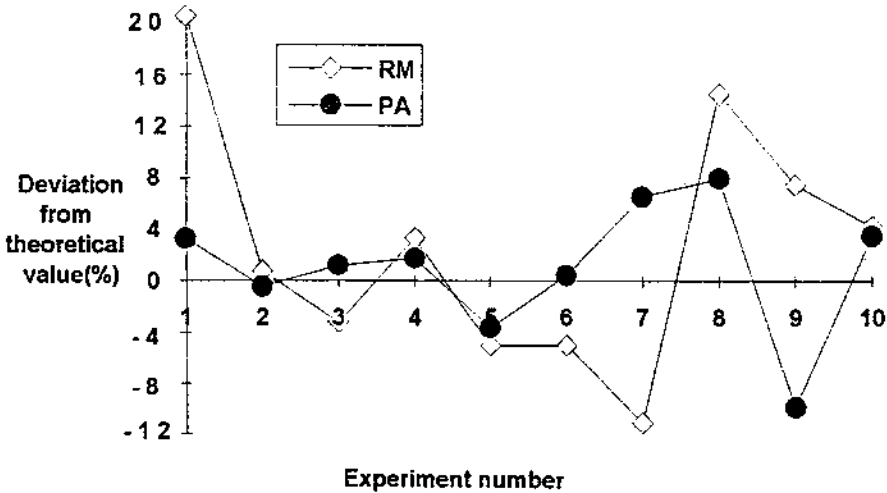


Fig. 4.4 Experimental results for average waiting time when  $\rho=0.9$

균을 비교한 결과가 Table 4.7(평균대기시간) 및 4.8(평균대기길이)에서 보여진다.

Table 4.7의 평균대기시간에 대한 결과를 살펴보면, 이용률이 0.2에서의 PA 방법의 절대 편차는 0.512이고, RM 방법의 절대 편차가 1.331로 프로크루스테스 분석 기법이 반

복-제거 방법보다 편차가 훨씬 작음을 알 수 있다. 이용률이 0.3 이상인 경우를 비교해도 프로크루스테스 분석 기법의 절대 편차가 반 복-제거 방법의 절대 편차보다 항상 작음을 알 수 있다. Table 4.8에서 평균대기길이에 대한 두 방법의 결과치를 비교해 보면, 이용률

**Table 4.7 Average deviation from true value for AWT**

Utilization( $\rho$ )	P A	R M
0.2	0.512	1.331
0.3	0.388	1.565
0.4	1.009	1.673
0.5	0.735	1.245
0.67	1.840	2.415
0.75	2.199	3.069
0.9	3.860	7.522

**Table 4.8 Average deviation from true value for AQL**

Utilization( $\rho$ )	P A	R M
0.2	0.601	1.620
0.3	1.164	1.200
0.4	1.931	2.133
0.5	0.888	1.228
0.67	2.038	2.420
0.75	2.138	3.135
0.9	4.043	7.860

이 0.2에서 0.9까지의 모든 경우에 본 논문에서 제안한 프로크루스테스 분석 기법의 편차가 비교 대상인 반복-제거 방법의 편차보다 항상 양호함을 알 수 있다. 일반적으로, 시뮬레이션 출력분석에 있어서 이용률이 증가할수록 자동상관정도(autocorrelation level)가 높아지는데, 이용률이 가장 큰 0.9일때를 살펴보면 본 논문에서 주장한 방법이 기존방법보다 훨씬 안정되어 있음을 Table 4.7과 4.8로부터 알 수 있다.

다음의 Table 4.9와 4.10은 평균대기시간과 평균대기길의 신뢰구간에 대한 반구간(half length)를 나타낸다. AWT와 AQL에 대한 결

과를 보면 이용률이 0.2와 0.3인 경우에는 PA가 RA보다 반구간이 좋지 않지만 이용률이 0.4이상이면 PA가 RA보다 반구간이 훨씬 좋음을 알 수 있다. 또한 PA는 그 정밀도가 어느 정도 균일함을 알 수 있다.

**Table 4.9 Average half length of confidence interval for AWT**

Utilization( $\rho$ )	P A	R M
0.2	0.1588	0.0534
0.3	0.1589	0.1109
0.4	0.1469	0.2784
0.5	0.1445	0.6366
0.67	0.1592	2.2368
0.75	0.1548	3.4041
0.9	0.1098	13.2838

**Table 4.10 Average half length of confidence interval for AQL**

Utilization( $\rho$ )	P A	R M
0.2	0.0058	0.0021
0.3	0.0044	0.0072
0.4	0.0049	0.0101
0.5	0.0038	0.0277
0.67	0.0045	0.0822
0.75	0.0052	0.1572
0.9	0.0143	1.3533

## 5. 결론

본 연구에서는 시뮬레이션의 안정상태 분석에 있어서 현재 일반적으로 사용되고 있는 반복-제거 방법과 새롭게 제안한 프로크루스테스 분석 기법을 이용한 결과를 상호 비교 평가 하였다.

본 논문에서 제안한 새로운 방법은 기존에

사용되던 방법보다 시뮬레이션의 반복 실행 횟수를 줄여도 이론적인 평균에 더 가까워짐을 보였으며, 새로운 방법이 반복-제거 방법보다 모수의 점 추정치는 모든 이용률에 대하여 평균적으로 좋은 결과를 나타내었으며, 구간 추정치는 이용률의 값이 어느 정도 커지는 경우(0.4이상)에는 평균적으로 더 좋아짐을 알 수 있다.

동시에 1개 이상의 모수의 파라미터를 추정할 수 있어서, 복수개의 파라미터를 1개의 모수를 추정하는 것과 같은 방법으로 구할 수 있는 장점이 있다. 또한 본 연구에서 제시한 프로크루스테스 분석 방법은 시뮬레이션 분야에서 전혀 사용되지 않았던 기법이였다. 그러므로 새로운 방법을 시뮬레이션 분야에 적용했다는 것에 상당한 의의가 있다.

#### 참 고 문 헌

- [1] 김 재련, 컴퓨터 시뮬레이션, 박영사, 1991.
- [2] 이 영해, 백 두권, 시스템 시뮬레이션, 경문사, 1991.
- [3] Gower, J. C., "Generalized Procrustes Analysis", Psychometrika, Vol. 40, No. 1, pp. 30-50, 1975.
- [4] Hadley, G., Linear Algebra, Addison Wesley, pp. 239-242, 1984.
- [5] Hoover, S. V. and Perry, R. F., Simulation, Addison Wesley, 1990.
- [6] Jos, M. F., "Orthogonal Procrustes Rotation for Two or More Matrices", Psychometrika, Vol. 42, No. 2, pp. 267-276, 1977.
- [7] Law, A. M. and Kelton, W. D., Simulation Modeling & Analysis, McGraw-Hill, 1991.
- [8] Moon, K. S. and Sutherland, J. W., "Improved Image Information via Procrustes Analysis", Quality Assurance Through Integration of Manufacturing Process and Systems ASME 1992, PED-Vol. 56, pp. 139-150, 1992.
- [9] Sutherland, J. W. and Moon, K. S., "Procrustes Analysis and Its Application to Sensor Integration", Trans. of NAMRI/SME, Vol. XX, pp. 347-354, 1992.
- [10] Fishman, George S., Principles of Discrete Event Simulation, John Wiley & Sons, 1978.