待ち行列現象のシミュレーション分析

逆瀬川　浩孝

待ち行列現象を理解するために実際に起きている事象を仮想体験するシミュレーション分析が有効である．簡単に試すことができるいくつかの例を「R」のプログラムとともに説明する．定常状態の評価指標をシミュレーションで推定する場合のやり方と結果のまとめ方，結果の持つ意味と限界について説明し，シミュレーションの正しい使い方についての注意をまとめる．

キーワード：待ち行列モデル，シミュレーション，非定常，信頼区間，R

1. はじめに

待ち行列理論は，不特定多数の客が特定のサービス施設を利用する際に生じる混雑現象を分析するために考えられた確率（過程）論をベースにした数理モデルです．その起源は電話交換器システムの設計問題で，マルコフ性と定常性という単純な仮定をおいて，回線容量と通話要求が拒否される確率の関係を導くことに成功したことに遡ります．それ以来，生産システム，コンピュータシステムやサービスシステムなどの様々な混雑現象を分析するためのモデルが考案され，確率論を駆使した分析がなされてきました．将来の不確実性を分析する数理モデルということでオペレーションズリサーチの中で重要な役割を果たしますが，確率論の壁は厚く，初学者には難解のようです．

しかし，学問の世界を離れると，多くの人は，無意識のうちに確率過程の最適意思決定問題を楽々とこなしています．パズドラや碁・将棋などがそれです．確率論などを知らなくても，数多くいろいろな場面を経験することにより，最適な意思決定を身につけていく，という問題解決のアプローチは参考になります．待ち行列現象に対してもこのようなアプローチは有効と思われます．客の到着，サービスのようなランダム現象を仮想的に多数実現させて，何が起こるのかいろいろ経験を積むことによってシステムの動きの特徴を理解し，確率分布としての特性を推定する，というような分析方法はシミュレーションと呼ばれ，理論解析を補完する分析法として待ち行列理論では盛んに利用されています．

シミュレーションを実施するには，シミュレーションモデルを作りそれを実行するというモデル化・計算技術とともに，その結果を正しく理解するための数理統計学の知識が必要になります．これらのことを実例を交えながらわかりやすく解説します．

さかせがわ　ひろたか
早稲田大学理工学術院
〒 169–8555 東京都新宿区大久保 3–4–1

2. シミュレーションの実施

待ち行列のシミュレーションは客の到着とサービス開始，終了などの（離散）確率事象によって，システムの状態（待ち行列の長さなど）が離散的に変化する，いわゆる離散事象シミュレーションの代表的な例です．離散事象シミュレーションを実行するには，システムの状態とその状態推移を促す（ランダムな）離散事象を定義し，事象が起きた場合のシステム状態の変化の規則を定め，ランダムな事象の確率規則（到着間隔，サービス時間，推移規則など）を決める必要があります．単一窓口モデルであれば，系内客数をシステムの状態としたとき，客の到着によって状態が 1 増え，サービス終了によって状態が 1 減るという動きを繰り返し，到着間隔としては，例えば，平均 2 の指数分布に従い，サービス時間は 1 以上 2 以下の一様分布に従う，というように決めます．

そうした前提の元で，乱数を使って架空の確率事象（到着間隔，サービス時間）を生成し，その生成時刻の順番に従って，事象が起きた場合のシステム状態を更新する，ということを繰り返します．将来のランダム事象とその生成時刻のペアを表にしたものを事象時刻表といいますが，この事象時刻表に従って事象が起きるたびにシステムの状態を更新し，必要に応じて統計データを記録します．あらかじめ決められた条件が満たされたところで実験を停止し，それまでに集めた統計データから必要な指標を推定する，というのがシミュレーション実験の全体の流れです．

システムが複雑になるにつれて，事象時刻表や，シス

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

テム状態の更新規則がややこしくなってきますが，専用のシミュレータを利用すれば作業を簡略化することができます．しかし，得られた結果を正しく解釈するには，それ相応の統計の知識が必要になります．

3. 待ち行列シミュレーションの例

3.1 非定常ポワソン過程

気まぐれな客の到着パターンをサービス施設側から見れば，「次の客がいつ来るかわからない」「時間帯によってだいたいの到着頻度が決まっている」という二つの特徴を持っています．時刻 t の到着頻度を表す指標として強度関数 $\lambda(t)$ を定義します．$\lambda(t)\Delta t$ は微少時間 $[t, t+\Delta t]$ の平均到着数を表すものとします．その到着パターンをシミュレーションで仮想体験するための確率モデルが非定常ポワソン過程です．

到着頻度がいつも一定とした定常ポワソン過程は，到着間隔が指数分布に従うことを利用すれば，指数乱数を使って仮想体験することができます．時間帯によって到着頻度が異なる場合は到着間隔を利用することができません．そこで，「削ぎ落とし法 (thinning algorithm)」という巧妙なアルゴリズムを使います．$\lambda^* = \max_t \lambda(t)$ として，率 λ^* の定常ポワソン過程を実現させ，到着頻度の少ないところはランダムに間引くことで，指定された到着率を実現しようというやり方です．具体的には，率 λ^* の定常ポアソン過程に従う標本 $\{t_n\}$ と，それとは別の一様乱数 $\{u_n\}$ を生成し，$\lambda(t_i) < u_i\lambda^*$ となる t_i をとり除いた $\{t_n\}$ が強度関数 $\lambda(t)$ を持つ非定常ポアソン過程に従う事象のサンプルになる，というものです．

シミュレーションを理解するには実際にサンプルパスを作って見るのが早道です．計算道具としてはExcelでもよいのですが，標本調査の場合は多数の繰り返しが必要となるので，操作性を考えるとプログラミングできた方がよいでしょう．ここでは統計データ解析用に作られた「R」を使うことにします．R はフリーソフトです．インターネットで「R インストール」を検索しトップに表示されるページの指示に従えば，すぐに利用できるようになるのでぜひ試してください（詳しくは [1] 参照のこと）．

非定常ポワソン過程のサンプルを削ぎ落とし法で計算するプログラムは次のようにします．1 行目が強度関数 $\lambda(t)$ の定義（$0 < t < 1$，最大値は $\lambda(1/4) = 1$），3 行目が定常ポワソン過程のサンプルの生成，4 行目が削ぎ落としです．2 行目から 6 行目までで非定常ポワソン過程のサンプルを生成する関数を定義しています．詳細説明はしませんが，シミュレーションに興味のある読者ならば解読（推理）可能でしょう．

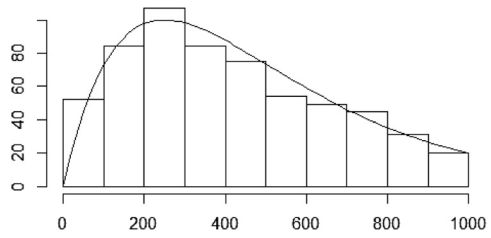

図 1 非定常ポワソン過程到着

＜R のプログラム例　その 1＞

```
lmd=function(t,pk=1/4) (t/pk)*exp(1-t/pk)
ihPP = function(r=1, T=1000, f=lmd) {
  hp = cumsum(rexp(r*T*1.2,r))
  ip = hp[which(f(hp/T)>runif(r*T*1.2))]
  return(ip[which(ip<T)])
}
hist (ihPP())
```

各時間帯ごとの到着頻度をヒストグラムにするのが最後の行です．ヒストグラムに想定した強度関数を重ねて描いたのが図 1 です．ここで，$\lambda(t)$ の定義を変えれば様々なパターンを生成することができます．

これを使っていろいろな待ち行列モデルのシミュレーションを実験してみましょう．

3.2 非定常な単一窓口モデル

非定常なポワソン過程にしたがって到着する客を処理するサービス窓口の混雑をシミュレーションで調べてみましょう．n 番目の客の到着時刻を t_n，退去時刻を d_n とすると，窓口が一つしかない場合，到着したときに前の客がいなければ $(t_n > d_{n-1})$ 待たずにサービスされ，さもなければ $d_{n-1} - t_n$ だけ待たされることがわかるでしょう．n 番目の客の待ち時間を w_n とすると，

$$w_n = \max\{d_{n-1} - t_n, 0\}$$

となりますが，サービス時間を s_n とすると，$d_n = t_n + w_n + s_n$ なので，

$$w_n = \max\{w_{n-1} + s_{n-1} - (t_n - t_{n-1}), 0\}$$

が導かれます．したがって，到着時刻列と架空のサービス時間が与えられれば待ち時間を「計算」することができます．そのプログラムの例を載せておきます．

＜R のプログラム例　その 2＞

```
at = ihPP(1.1,500)  # 非定常ポワソン過程
m = length(at)-1
arv = at[-1] - at[-(m+1)]  # 到着間隔
```


Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

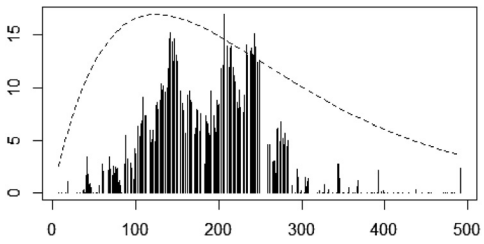

図 2　非定常待ち行列システムの待ち時間推移

```
svc = rexp(m, 1)  # サービス時間
wait = numeric(m)
for(i in 2:m) wait[i] =
    max(wait[i-1]+svc[i-1]-arv[i-1],0)
plot(at[1:m], wait, type="h")
curve(lmd(x/500)*max(wait), lty=2,
   add=TRUE)

end
```

サービス時間分布を指数分布として計算したものの一例が図 2 です．この場合は，ピーク時のトラフィックがサービス処理能力の 1.1 倍として計算しました．棒グラフが待ち時間，点線が到着の強度関数です．到着のピークに向けて待ち行列が発生し，その処理を終えた後はほとんど待ち行列がなくなる，という全く違うパターンを描くことが観察できます．といっても，乱数を変えて同じような図を描いてみると，常にこのようなグラフができるわけではなく，ピーク時の到着量の違いで，行列がほとんどできなかったり，混雑がもっと激しくなるということもしばしば観察されるので，非定常の場合の特徴を一言で表すのは難しいことが実感できます．ピーク時の到着率がもっと大きくなったとき，あるいは，強度関数の形状が変わったときどうなるのか，などいろいろ条件を変えてシミュレーションしてみると，いろいろな知見を得ることができるでしょう．

3.3　高速道路の混雑

次の例は，道路をサービス施設と考えて道路の混雑状況の時間変化を調べた例です．車線数を s，ある特定の場所を通過することをサービス時間と考えると，一定サービス時間の s 窓口モデルになりますので，上のプログラムを少し書き換えるだけでその混雑具合をシミュレーションで調べることができます．

図 3 は 4 車線で，ピーク時に処理能力の 1.5 倍のトラフィックが到着した場合を想定して実験したものです．2 本の単調増加する太い実線は累積到着台数と累積通過台数を表したもの，それ以外の 2 本の点線は単位時間あたりの到着台数と，各時間帯ごとの渋滞長を表したものです．山が左にあるのが到着台数の変化で，到着のピークからかなり遅れて渋滞のピークが訪れることがわかります．

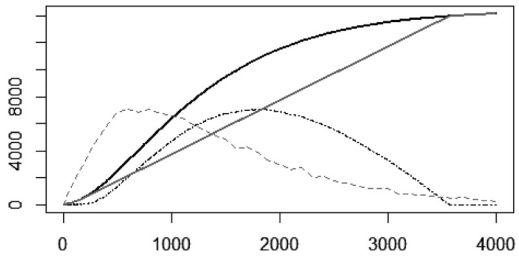

図 3　到着のピークと混雑のピーク

3.4　テーマパークの人出

テーマパークのような巨大施設の混雑現象も興味の対象です．テーマパークに入ってから出るまでをサービスと考えると，入場制限がない場合，窓口が無数にある待ち行列モデルと見なすことができます．通常の意味で「待ち」はありませんが，サービス終了まで「待たされる」と考えれば，これも立派な待ち行列問題になります．待ち時間はサービス時間そのものなので，この場合は系内客数の動向に興味があります．テーマパークはいつ混んでいるのか，ある時間帯に滞在する客数はどれくらいか，などです．

標準的な待ち行列モデルに当てはめると，無限大窓口モデル $\mathrm{M/G/\infty}$ がそれに近く，この場合は理論解析が可能で，系内客数はポアソン分布に従うことが知られています [3]．しかし，これを実際の問題に当てはめようとすると，「定常性」の壁に阻まれて，うまくいきません．そこで登場するのがシミュレーションです．

時間帯毎に違う平均値を持つ分布を適当に定め，n 番目の到着客の到着時刻 t_n と滞在時間 s_n が決まれば，退去時刻は $d_n = t_n + s_n$ によって計算することができます．任意の時刻 t で $t_n < t < d_n$ を満たす n の個数を数えれば，滞在客数 $N(t)$ がわかります．「R」のプログラム例を載せておきます．

＜ R のプログラム例　その 3 ＞

```
a = 1000; T = 10; stay = 3
arv = ihPP(a,T) # 到着時点列
m = length(arv) # 到着数合計
yt = rexp(m,1/stay) # 滞在時間
dpt = arv + sojourn(arv)*yt # 退去時刻
dt = 0.1
time = seq(0,T,dt)
nt = length(time)
que = numeric(nt) # 滞在客数の計算
```

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

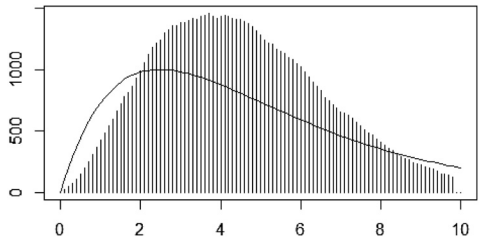

図 4　テーマパークの滞在客数の時間推移

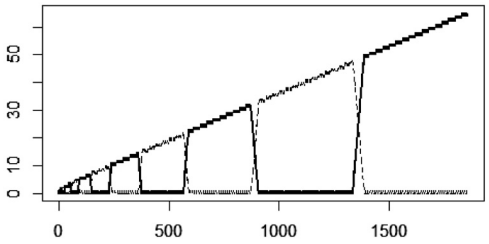

図 5　リエントラント待ち行列モデルの滞在客数

```
for(k in 1:nt) que[k] =
  length(which(arv < k*dt & k*dt < dpt))
plot(time,que,type="h") # 滞在客数表示
curve(a*lmd(x/T), add=TRUE) # 強度関数
```

図 1 のような非定常ポワソン過程に従って到着し，時間が経つにつれて徐々に平均が短くなるような指数分布に従う滞在時間を仮定したとき，上のような手順で求めた滞在客数が図 4 の棒状のグラフです．到着客数も折れ線で重ねて描いてあります．これにより，到着のピークとテーマパーク内の混雑のピークがずれていることがわかります．到着の強度関数 $\lambda(t)$ と，時間帯毎のサービス時間の分布がわかればこのような図が簡単に描けるので，何らかの方法でその情報が得られれば，事前にどのような混雑状況になるか予測が可能になります．あるいは，ピークの人出をある数以下に抑えるために，例えば入場制限をするとすればどのような規制を掛ければよいかを事前に検討することができるようになります．

3.5　リエントラント待ち行列

シミュレーションの役割として，きちんと分析する前に，システムの動きがどうなっているのか掴んでおく，という教育効果もあります．例えば，窓口が二つ（A，B とします）あって，2 種類の客（P，Q とします）がこの窓口システムを利用するものとします．タイプ P の客は A の処理を受けた後 B の処理を受けて退去し，タイプ Q の客は逆に B から A へ移動するものとします．最初のサービスを前処理と呼ぶと，両方の窓口では前処理を受けた客を優先的にサービスするものとします．このとき，各窓口の待ち行列の長さはどのように変化するか，ということが問題です．

簡単のために，到着間隔も処理時間も確率変動しないとしましょう．もし，窓口が分業しないで，窓口 A はタイプ P の客，窓口 B はタイプ Q の客だけを受け付けて，前処理と本処理の両方を行うとすると，到着間隔が処理時間の総和よりも短ければ待ち行列は発生しません．分業してもその状況はそれほど変わらないとすると，待ち行列はあまり長くならないのではないかと考えられます．その予想はシミュレーションによって（間違いであることを）確かめることができます [2].

例えば，どちらのタイプの客も，10 単位時間間隔で到着し，前処理に 1 単位時間，本処理に 6 単位時間かかるものとしましょう．もし，最初にタイプ P の客が到着し，その 5 単位時間後にタイプ Q の客が到着したとして，その後の窓口 A，B の待ち行列長の時間変化を計算したのが図 5 です．太い線が窓口 A の待ち行列（P，Q を合わせたもの），細い線が窓口 B の待ち行列を表しています．待ち行列の長さはどんどん長くなりますが，待ち行列ができるのはどちらか一方だけ，ということがわかります．これは本処理が始まるとその窓口で処理するはずの前処理がストップして，もう一方の窓口で本処理する客がいなくなってしまう可能性があるからです．本処理の時間を短くするとこのような奇妙な動きは見られなくなります．

処理能力には十分余裕がありながら，ある制約を超えたとたんに全く異なる動きをするということは，直感的にはなかなか理解できません．いろいろな条件を設定してシステムの振る舞いを試すことができることもシミュレーションの効用といえましょう．

4.　推定問題

これまでのシミュレーションは，客の気まぐれ行動でどのような混雑状況が発生しうるか，ということをいろいろ体験してみることに重点をおいて説明してきましたが，例えば，テーマパークのピーク時の人出がある数以上になる可能性はどれくらいあるか，というように，数量的な評価に興味がある場合は，それなりの計算が必要になります．実際，乱数を変えてシミュレーションを再実行させると，ピーク時の人出はその都度変化します．シミュレーションを再実行するということは，別の日の人出を仮想体験するということなので，ピーク時の人出が全く同じ数になるという可能性は少ないでしょう．シミュレーションを一回実行し

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

て得られた結果というのは，例えて言えば，ある 1 日の調査結果のようなもので，その一つの数値だけから結論を引き出すのは適切でないことがわかります．

この問題に適切に対処するためには数理統計学の標本調査論の知識が必要になります．未知の母平均を推定するためには，独立標本を抽出して，信頼区間を計算する，というものです．今の場合，「ピーク時の人出がある数を超える確率」が母平均に対応します．シミュレーション実験では，乱数を変えて再計算し直すと独立なサンプルパスが得られると期待できるので，シミュレーションを n 回繰り返し，そのうち m 回でピーク時の人出がある数を超えたとしたら，$m/n(\equiv \hat{p})$ によってその確率を推定することができます．このような推定を点推定といいます．

話はここで終わらせるわけにはいきません．直感的に n を大きくすれば推定値は真の値に近いものが得られそうだということは何となくわかりますが，m/n も（日数は増えたものの）1 回の調査結果に過ぎず，その結果が真の確率と同じという保証はどこにもありません．そこで必要なのが信頼区間を使った区間推定という考え方です．今の場合でいうと，真の確率は $m/n \pm \varepsilon$ の間にあるという主張は 95% 正しい，といえるような誤差の幅 ε を付けて推定するという方法です．この場合はだいたい $\varepsilon = 2\sqrt{\hat{p}(1-\hat{p})/n}$ とすればよいことがわかっています（例えば [1] 参照）．$2\sqrt{\hat{p}(1-\hat{p})}$ はどんなに大きくても 1 止まりですから，誤差は $1/\sqrt{n}$ 程度と見積もることができます．

誤差がシミュレーション回数 n の平方根に反比例する，というこの性質は平方根則とも呼ばれ，シミュレーション実験を行う際の指針を与えています．n は実行時間に比例すると考えると，シミュレーションの推定誤差を半分にしようとすると，それまでの実験時間の 4 倍が必要になる，というように，推定に必要な時間が急速に増大することが制約になります．

5. 定常待ち行列モデルのシミュレーション

待ち行列理論の多くは，待ち行列システムの定常的な振る舞いを分析することに費やされています．定常とは，システム状態の確率分布が時間的に動かない，という性質なので，普通は状態確率に関する方程式（平衡方程式）を解くことが必要になります．その方程式を作るために，あるいは，作った方程式を解くために，様々な高度な確率論が駆使され，待ち行列理論の難解さの源になっているようです．

平衡方程式を解く代わりに，今まで述べてきたようなシミュレーションを使って分析することを考えてみましょう．例えば，単一窓口モデルの定常状態の平均待ち時間を求めたい，という問題を考えます．3 節で示した待ち時間とサービス時間，到着間隔との関係は，文字を確率変数と思えば，そのまま成り立つ式です．文字を確率変数らしく大文字にして再掲します．

$$W_n = \max\{W_{n-1} + S_{n-1} - A_n, 0\}, \quad n = 2, 3, \ldots$$

ただし，到着間隔 $t_n - t_{n-1}$ を A_n と置き換えました．このとき，定常状態の平均待ち時間は，$n \to \infty$ としたときの $E(W_n \mid W_1 = w)$ によって求めることができるので，前節の推定問題で説明したように，たくさんの独立標本を作ってその平均値を使えば平均待ち時間を推定することができます．

理屈ではそうなりますが，実際に計算する場合に使える時間は有限ですから，「$n \to \infty$ としたときの」標本は計算では求めることができません．そこで，n が十分に大きければ $n \to \infty$ としたときと同じような状況を作り出すことができる，と信じて近似計算をせざるをえません．このとき問題になるのが，n はどれくらい大きく取ればよいのか，ということです．この問題は立ち上げ問題 startup problem と呼ばれ，様々な研究がありますが，どんなシミュレーションに対しても有効な指針があるわけではありません．システムの状態から定まる指標（待ち行列長のようなもの）の動きを観察しながら「ある程度」長い時間シミュレーションを実行して，一方的に増加とか減少の傾向が無くなったところで $n \to \infty$ と同じような状態（定常状態）に到達したと見なす，というやりかたで対処するしかありません．

結局，定常状態の平均待ち時間を推定するには，十分大きな n_0 と m を決め，

$$Z = \frac{1}{m}(W_{n_0+1} + \cdots + W_{n_0+m}) \tag{1}$$

を標本値とするシミュレーションを N 回繰り返して $Z_1, \ldots, Z_N$ を求め，それらの平均値を平均待ち時間の推定値とする，という手順になります．

システムが複雑になって状態数が多く，その間の推移が複雑になるほど初期状態の影響を引きずりやすいということはわかるので，そのような複雑なシステムのシミュレーションでは一つの標本を得るために（上の n_0 に当たるスタートアップのための）相当むだな計算が必要になります．そのむだを回避するための工夫がバッチ平均法と呼ばれる方法です．

1 回のシミュレーションを実行して定常状態に達したとすれば，そこから先はずっと定常状態と見なしてよ

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

いのですから，標本を採ったらそれでおしまいにして，また新たに一からやり直し，ではもったいない，というわけで，1 回のシミュレーションを長い間続け，長いサンプルパスを計算し，そこから必要な標本数を採取すればよい，と考えるのです．すなわち，十分大きな n_0 と m，N を決め，1 回のシミュレーションを実行して $\{W_n, n = 1, \ldots, n_0 + Nm\}$ を計算し，

$$Z_i = \frac{1}{m}(W_{n_0+m(i-1)+1} + \cdots + W_{n_0+mi}) \quad (2)$$

によって $Z_1, \ldots, Z_N$ を求め，それらを独立標本と見なして，前節の方法で平均待ち時間を推定することにします．

このやり方の問題は，1 本のサンプルパスからとった別々の標本は厳密に言えば独立でないということですが，これも上の問題と同じように「ある程度」間を置けば独立と見なしてもよい，と考えて独立標本の理論を適用します．このようなシミュレーションのやり方は一標本法，あるいはバッチ平均法 (batch means)，と呼ばれ，それに対して，スタートアップを繰り返す方法は独立標本法と呼ばれます．

m が十分に大きければ，中心極限定理により Z_i は正規分布に従うと考えてよいので，$Z_1, \ldots, Z_N$ が独立標本と考えてよいのであれば，その標本平均は真の平均待ち時間の不偏推定量で，その信頼区間は t 分布を使って

$$\left[\bar{Z} - t_{N-1,\alpha}\frac{s}{\sqrt{N}}, \bar{Z} + t_{N-1,\alpha}\frac{s}{\sqrt{N}}\right] \quad (3)$$

と表すことができます．ただし，s は不偏分散の正の平方根，$t_{n,\alpha}$ は自由度 n の t 分布の両側 $100\alpha\%$ 点を表すものとします．数理統計の理論から，$\bar{Z}$ と s からこの区間を計算すると，この区間のどこかに未知の真の平均があるという主張は $100(1-\alpha)\%$ 正しいということが保証されます．逆に言えば，$100\alpha\%$ は正しくない主張をしたことになります．しかし，標本がばらつく以上，意味のある推論をするためには，それだけの誤差は許容せざるをえないのです．繰り返しになりますが，この理論が適用できるのは，m が十分に大きいことと，$\{Z_i\}$ が互いに独立と見なしてもよい，という条件が必要です．

このような近似を使わないで定常状態の分析を可能にするために，更新過程 (regenerative process) を利用したシミュレーションを考えることもできますが，それについては例えば [4] を参照してください．

6. 実験結果の意味，信頼性

例えばサービス処理方法を変えた二つのサービス施設のどちらが平均待ち時間が短くなるか，というように，二つの待ち行列システムの性能比較をシミュレーションで実行する場合を考えてみましょう．この場合，それぞれのシステム（A，B としましょう）に対してシミュレーションを実行して，それぞれの平均待ち時間 w_A, w_B が推定できたとします．それらの推定値をそれぞれ $\bar{w}_A, \bar{w}_B$ としたとき，それ以外の情報がなければ $\bar{w}_A < \bar{w}_B$ のときシステム A の方が優れている $(w_A < w_B)$ という判断をしがちですが，今まで説明してきたことを正確に理解すれば，そのような判断が間違いであることがわかるでしょう．

$\bar{w}_A, \bar{w}_B$ は 1 回の調査結果と同じなので，たまたま観察したときの結果だけから判断してはいけないということなのです．したがって，シミュレーション実験では「$\bar{w}_A, \bar{w}_B$ 以外の情報がない」ような報告書を書いてはいけません．必ず信頼区間，あるいは誤差の大きさを添える必要があります．もし，w_A, w_B の信頼区間が重なっている場合は，$\bar{w}_A$ と $\bar{w}_B$ の見かけ上の大小関係はたまたま使用した乱数によってもたらされた偶然による誤差で，意味のある差ではないと考えるべきなのです．言い換えれば，$w_A = w_B$ という帰無仮説が棄却できない，ということを意味します．

理屈ではわかっていても，世の中には「優劣を付けろと言われれば $\bar{w}_A, \bar{w}_B$ の小さい方が優れている」という主張をする人が多いようです．もし信頼区間の重なりが小さければ，その主張に賛同する人は増えると思われます．いずれにせよ，信頼区間が重なっている場合「優劣は付けられません」と答えなければいけません．

優劣を付けるためには誤差を小さくする必要があります．誤差，すなわち信頼区間の幅の半分の大きさは，式 (3) を見てわかるように，三つの要因が関係しています．信頼度を表す指標の $t_{N-1,\alpha}$，標本の不偏分散 s^2，それに独立標本の個数 N です．したがって，誤差を小さくするためには分子の $t_{N-1,\alpha}$，あるいは s を小さくするか，分母の N を大きくするかのどちらかが必要です．N を変えずに $t_{N-1,\alpha}$ を小さくすることは α を大きくすることと同じで，それは区間推定の信頼度を下げることを意味します．区間幅を小さくして見かけ上の誤差を小さくしたとしても，その推定が信用できなければ元も子もありません．N を大きくすることは計算時間だけの問題ですから，時間を掛けられるならば

Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.

それも選択肢の一つです．誤差の大きさ ε が指定されれば，$t_{N-1,\alpha}s/\sqrt{N} < \varepsilon$ を満たす N を計算して，シミュレーション回数を決めることができます．しかし，一つの標本を得るのにそれなりの時間がかかるとすると，実行が危ぶまれます．3 番目の選択肢は s を小さくすることです．そのような工夫は s^2（分散）を小さくするという意味で，分散減少法と呼ばれます．二つのシステムの差を計りたい場合は共通乱数法という方法が有効です．

7. おわりに

待ち行列を作る事例は日常生活の至る所に見られ，ちょっとしたきっかけで長くなったり短くなったりする不思議な現象には誰でも興味を引かれると思いますが，そのランダムな事象をきちんと解析しようとすると，とたんに確率論の壁にぶつかって跳ね返されてしまった経験を持つ人も多いのではないかと想像されます．そのような場合にシミュレーション分析は強力なパートナーになってくれるでしょう．乱数を有効利用して，ゲーム感覚で待ち行列現象を架空体験すれば，そのメカニズムを理解し，制御するための知恵が生まれてくるかもしれません．この小論が，そのようなきっかけになってくれれば，これに過ぐる喜びはありません．

参考文献

[1] 伏見正則，逆瀬川浩孝，「R で学ぶ統計解析」，朝倉書店，2012.

[2] P. R. Kumar, Re-entrant lines, *Queueing Systems*, **13**, 87–110, 1993.

[3] 増山博之，滝根哲哉，大規模施設の混雑現象，オペレーションズ・リサーチ，**49**–7, 422–425, 2004.

[4] 森戸晋，逆瀬川浩孝，「システムシミュレーション」，朝倉書店，2000.

 Copyright © by ORSJ. Unauthorized reproduction of this article is prohibited.