数理解析研究所講究録
第452巻 1982年 93-106

ブロッキングを考慮した直列型待ち行列モデルの数値解析

筑波大学 社会工学系
逆瀬川浩孝

1. はじめに

いくつかの窓口が直列型に配置され、客はこれらの総ての窓口を順に通過し、退去するという、いわゆる直列型待ち行列モデルについて考える。このモデルを解析する為には、一つの窓口からの客の退去過程を知る必要がある。良く知られているように、M/M/s あるいは M/G/∞ の退去過程は、到着と同じポアソン過程になるので、このような窓口のみの直列型モデルならば、本質的には単段モデルと同様に取り扱うことができる。また、それらのモデルの拡張として、いわゆる Kelly システム、あるいは、BCMP 型窓口モデルを個々の窓口としても同様なことが言える。これらのモデルは、見かけ上は直列型でありながら、窓口相互が互いに独立に機能しているので、単段モデルの拡張にはなっていない、ということが言える。

直列型モデルを特徴付けるのは、窓口相互の干渉の存在である。例えば、各窓口での待合室の大きさを有限の値に制限すると、あるサービスを終了した客が、次の待合室に入ろうとする時、そこが他の客でみたされていれば先に進むことができないので、窓口の次の客へのサービスを妨害することになる。これを窓口のブロッキングと呼ぶが、このようなブロッキングを伴う待ち行列モデルについては、ポアソン到着・指数サービスの極く基本的なモデルについてさえ、解析的に解くことはできない。二段モデルについて、母関数、あるいは差分方程式にもとづく数値的な解法が得られているが、それより大きなモデルに対しては、マルコフ連鎖モデルに帰着させて、平衡方程式を解くしかないようである。ここでは、与えられたモデルに対して、その平衡方程式を構成し、それをもとに平衡確率を求め、諸特性量を計算する為の計算機プログラムの試作例を報告する。マルコフ・モデル化する為には、アーラン分布を相分解するように、補助変量を導入する必要があり、平衡方程式の次元は、モデルの大きさと共に、指数的に増大するので、大きなモデルに対しては、この方法は余り有効ではなく、何らかの近似法を考えねばならない。しかし、近似による前に、どの程度のモデルならば近似を使わなくても解けるのか、ということを知っておく必要があり

、また、近似法の基礎として、できるだけ多くのモデルに対して厳密解を用意しておく必要もある。更に、マルコフ・モデル化することの利点は、その平衡解を反復法で解くことにより、その過程で、一時解をシミュレートすることができるという点にある。このようないくつかの点で、試作プログラムは役に立つと思われる。

2. モデルの記述

K段からなる直列型待ち行列モデルを考える。各段は c_k 個の窓口と、大きさ $N_k - c_k$ の待合室からなり、そのサービス時間分布は、l_k-アーラン型、平均サービス時間を μ_k^{-1} とする。客の到着は第一段のみで、到着間隔は l_0-アーラン分布に従い、その平均は λ^{-1} である。$k(<K)$ 段で、ある客のサービスが終了した時点で、$k+1$ 段に N_{k+1} 人の客がいる時、その客は先へ進めず、k 段の窓口を占拠したまま、移動できる状態になるまで待つ（k 段窓口をブロックするという）。$k(>1)$ 段である客のサービスが終了してその客がその段を退去した時点で、$k-1$ 段にブロックされている窓口があれば、そのうちの一つを占拠している客を k 段に進め、その窓口のブロックを解除する。ブロックされている窓口が2つ以上ある場合、もし、系内滞在時間の分布を問題にするのであれ

ば、どの窓口のブロックを先に解除するか、という規律が問題となるが、ここでは系内客数について調べるだけなので、ランダムに選んでよい。ある窓口のブロックが解除されることによってその直前の段の窓口のブロックが解除される場合も同様に扱う。第一段にN_1人の客がいた時に到着した客は拒否され、失われる。

3. 解析

モデルのマルコフ表現は、次のような確率変数を導入することによって可能となる。R_0を到着過程の残りフェーズ数とする。Q_kをk段の系内客数、$R_{k1},\dots,R_{kc_k}$をk段の窓口のサービスの残りフェーズ数を大きさの順に並べたものとする。客がいないか、ブロックされている窓口の残りフェーズ数は0である。

$$(R_0(t),\ (Q_k(t), R_{k1}(t), \dots, R_{kc_k}(t))_{k=1,2,\dots,K})$$

は連続時マルコフ連鎖である。その状態集合を$\mathcal{S}_K$，生成行列を$A=(a(s,t))$とすると、$a(s,t)$は次の式で与えられる。ただし、

$$s = (r_0, (q_k, r_{k1}, \dots, r_{k l(k)})_{k=1,\dots,K}) \equiv (r_0, (\underline{n}_k)_{k=1,\dots,K})$$

$$l(k) = \min(q_k, c_k)$$

$$\xi_0(s) = s\,(r_0 \to r_0 - 1) \quad (r_0 > 1)$$

ここで $s(a\to b)$ はベクトル s の a 成分だけを b に変えるだけで、他は不変のベクトルを表すものとする

$$\xi_0(s) = s(r_0 \to l_0) \qquad (r_0 = 1)$$

$$\xi_k(s) = s(q_k \to q_k + 1)$$

$$\eta_k(s) = \begin{cases} \xi_k(s) & (c_k \le q_k \le N_k) \\ s(\underline{n}_k \to (q_k+1, l_k, r_{k1}, \cdots, r_{k\,l(k)})) & (q_k < c_k) \end{cases}$$

$$\zeta_k(s) = \begin{cases} s(\underline{n}_k \to (q_k - 1, r_{k1}, \cdots, r_{k\,l(k)-1})) & (q_k \le c_k) \\ s(\underline{n}_k \to (q_k - 1, l_k, r_{k1}, \cdots, r_{k\,l(k)-1})) & (c_k < q_k \le N_k) \end{cases}$$

$$\bar{\zeta}_k(s) = \xi_k \zeta_k(s)$$

$$\xi_{kj}(s) = s(r_{kj} \to r_{kj} - 1)$$

$$\alpha_j(\underline{n}_k) = \begin{cases} \#\{i \,;\, r_{ki} = r_{kj} \neq 0, i \le j\} & (r_{kj} \neq r_{k\,j+1}) \\ 0 & (\text{それいがい}) \end{cases}$$

と定める。

$$a(s, \xi_0(s)) = l_0 \lambda \qquad (r_0 > 1 \text{ 又は } r_0 = 1 \text{ かつ } q_1 = N_1)$$

$$a(s, \xi_1 \xi_0(s)) = l_0 \lambda \qquad (r_0 = 1 \text{ かつ } c_1 \le q_1 < N_1)$$

$$a(s, \eta_1 \xi_0(s)) = l_0 \lambda \qquad (r_0 = 1 \text{ かつ } q_1 < c_1)$$

$$a(s, \xi_{kj}(s)) = \alpha_j(\underline{n}_k) l_k \mu_k \qquad \left(\begin{array}{l} r_{kj} > 1 \text{ 又は } j \neq l(k) \text{ 又は} \\ j = l(k) \text{ かつ } q_{k+1} = N_{k+1} \end{array} \right)$$

$$a(s, \zeta_k \eta_{k+1}(s)) = \alpha_{l(k)}(\underline{n}_k) l_k \mu_k \qquad (r_{k\,l(k)} = 1 \text{ かつ } r_{k-1\,l(k-1)} > 0)$$

$$a(s, \zeta_{j-1} \bar{\zeta}_j \cdots \bar{\zeta}_k \eta_{k+1}(s)) = \alpha_{l(k)}(\underline{n}_k) l_k \mu_k$$

$$(q_i > 0,\ r_{i\,l(i)} = 0\ (i = j, \cdots, k) \text{ かつ } r_{j-1, l(j-1)} > 0,\ r_{k, l(k)} = 1)$$

状態集合が確定し、生成行列が与えられゝば、原理的には平衡確率を計算することはできるが、与えられたパラメータごとに平衡方程式を構成することは大変なので、その為の計算機プログラムを作ることを考える。それにはいくつかの補助的な道具だてが必要である。

(1) 状態集合 S_K の大きさ、あるいは平衡方程式の次元、を求める漸化式。

$\underline{n}_k=(q_k, r_{k1},\cdots, r_{k\,l(k)})$ は次の条件をみたしているような数ベクトルである。

$$l_k \geqq r_{k1} \geqq \cdots \geqq r_{k\,l(k)} \qquad , \quad 0 \leqq q_k \leqq N_k$$

$$r_{k\,l(k)} \geqq 1 \qquad (q_{k+1} < N_{k+1} \text{ 又は } k=K)$$

$$r_{k\,l(k)} \geqq 0 \qquad (q_{k+1} = N_{k+1} \text{ かつ } k<K)$$

$q_k=i$, $r_{k\,l(k)} \geqq \delta$ $(\delta=0,1)$ の場合、このような組合せの数は

$$\binom{l_k+i-\delta}{i} \qquad (q_k=i<c_k)$$

$$\binom{l_k+c_k-\delta}{c_k} \qquad (q_k=i \geqslant c_k)$$

によって与えられる。そこで

$$\beta_k(\delta) = \sum_{i=0}^{c_k-1} \binom{l_k+i-\delta}{i} + (N_k-c_k)\binom{l_k+c_k-\delta}{c_k}$$

とおくと、次の漸化式によって S_K の大きさを計算することができる。

$$f(0,0) = f(0,1) = l_0$$

$$f(k,\delta) = \beta_k(\delta)\, f(k-1,1) + \binom{l_k+c_k-\delta}{c_k} f(k-1,0) \qquad (k=1,2,\cdots)$$

ここで $f(K,1) = |\mathcal{S}_K|$ である。

(2) $\mathcal{S}_K$ の要素を整列させるアルゴリズム、すなわち、$\mathcal{S}_K$ から $\{0,1,\cdots,f(K,1)-1\}$ への写像 g_K

$s = (r_0, (q_k, r_{k1}, \cdots, r_{k a(k)})_k)$ が与えられたものとする。

1. $g_K \leftarrow 0,\ \delta \leftarrow 1,\ k \leftarrow K$
2. $q_k \leqq 0$ ならば $\delta \leftarrow 1$ として 9 へ.
3. $g_K \leftarrow g_K + f(k-1,1)\left\{\sum_{j=1}^{q_k-1} \binom{l_k-\delta+\min(j,c_k)}{l_k-\delta}\right\}$, $m \leftarrow 0$, $j \leftarrow \delta$
4. もし $\min(q_k, c_k) \leqq 1$ ならば 8 へ
5. $i \leftarrow \min(q_k, c_k)$
6. $r_{ki} \leqq j$ ならば 8 へ
7. $m \leftarrow m + \binom{l_k+i-1-j}{i-1}$, $j \leftarrow j+1$, $i \leftarrow i-1$. もし $i \geqq 2$ ならば 6 へ
8. $m \leftarrow m + r_{k1} - j$. もし $q_k = N_k$ ならば $\delta = 0$. さもなければ $\delta \leftarrow 1$。$g_K \leftarrow g_K + m \cdot f(k-1,\delta)$
9. $k \leftarrow k-1$. もし $k \geqq 1$ ならば 2 へ. さもなければ $g_K \leftarrow g_K + r_0 - 1$

(3) m番目の状態ベクトルを生成するアルゴリズム、すなわち、g_K の逆写像.

0. $k \leftarrow K,\ \delta \leftarrow 1$
1. $l \leftarrow \left[\frac{m}{f(k-1,1)}\right]$, $u_k \leftarrow \sum_{j=0}^{c_k-1} \binom{l_k+j-\delta}{j}$, $a_k \leftarrow u_k + (N_k - c_k)\binom{l_k+c_k-\delta}{c_k}$

2. もし $l \geq a_k$ ならば $q_k \leftarrow N_k$, $m \leftarrow m - a_k f(k-1,1)$, $l \leftarrow \left[\frac{m}{f(k-1,0)}\right]$
   $j \leftarrow c_k$ として 6 へ
3. $m \leftarrow \mathrm{mod}(m, f(k-1,1))$.
   もし $l \geq l_k$ ならば $q_k \leftarrow c_k + \left[\frac{l-l_k}{\binom{l_k+c_k-\delta}{c_k}}\right]$, $l \leftarrow \mathrm{mod}\left(l-l_k, \binom{l_k+c_k-\delta}{c_k}\right)$
   $j \leftarrow c_k$ として 6 へ
4. もし $l \leq 0$ ならば $\underline{n}_k \leftarrow (0,\cdots,0)$ として 11へ
   さもなければ $j \leftarrow 1$, $l \leftarrow l-1$ として 5 へ
5. $l < \binom{l_k+j-\delta}{j}$ ならば $q_k \leftarrow j$ として 6 へ
   さもなければ $l \leftarrow l - \binom{l_k+j-\delta}{j}$ として 5 へ
6. $r_{k,j+1} = \cdots r_{kc_k} \leftarrow 0$, $i \leftarrow j$
7. $l < \binom{l_k+i-1-\delta}{i-1}$ ならば $r_{ki} \leftarrow \delta$, $i \leftarrow i-1$ として 7 へ.
   さもなければ $h \leftarrow 0$ として 8 へ
8. $l \leftarrow l - \binom{l_k-i-1-h-\delta}{i-1}$, $h \leftarrow h+1$
9. もし $l \geq \binom{l_k-i-1-h-\delta}{i-1}$ ならば 8 へ. さもなければ
   $r_{ki} \leftarrow h+\delta$, $i \leftarrow i-1$ とし. $i>1$ の時. 9 へ.
10. $r_{k1} \leftarrow l+h+\delta$
11. $k \leftarrow k-1$. もし $k \geq 1$ ならば 1 へ.
    さもなければ $r_0 \leftarrow m+1$

次に. 与えられた平衡方程式から. 平衡状態確率 $\pi = (\pi(0), \ldots, \pi(f(K,1)-1))$ を計算するアルゴリズムを考えよう。q_K によって整列された状態の並び方による生成行列を. 改めて

$$A = (\ a(j,k)\ ) \quad , \quad a(j,j) = -\sum_{k \neq j} a(j,k)$$

とおけば

$$\begin{cases} \underline{x} A = 0 & \underline{x} = (x_0, \cdots, x_{f(K,1)-1}) \\ \sum_j x_j = 1 & x_j \geq 0 \quad (\forall j) \end{cases}$$

は唯一つの解 π を持つから、この連立方程式を解けば π が求まる。連立方程式の数値解法として有名な消去法を、この問題に対して適用することは得策ではない。何故なら、ここで考えている方程式の次元は数千のオーダになるので、消去法の基礎である積和演算の積み重ねに危険が伴うのと、演算時間が次元の増大に伴って指数的に増大するからである。

さて、総ての c について、

$$\pi\ (cA + I) = \pi$$

が成り立つことから、π は行列 $cA+I$ の固有値1に対する固有ベクトルになる。特に $cA+I$ が確率行列になっていれば、良く知られているように、この固有値1は絶対値最大で、しかもここで扱うモデルの場合には、単根になっている。このような条件をみたす固有ベクトルは反復法によって求めることができ、その収束は、第二固有根の絶対値をパラメータとして、幾何的であることが言える。第二固有根の大きさは c の与え方によって異なるが、

$$c = (\max |a(j,j)|\ )^{-1}$$

とするのが良い。反復法を使った場合の計算時間は，行列の次元の自乗に比例するが，ここで扱う行列 $cA+I$ はスパースなので，行列の次元に比例すると考えてよい。数値実験の例では，数千の次元の固有ベクトルを 10^{-8} の精度で求めるのに要する計算時間は，総てを倍精度で計算しても1分以内，反復回数は数百回のオーダであったから，単純反復法でも十分に実用的であることが確かめられた。次元が大きくなった時の問題は，計算を多数回繰り返す時の累積するであろう計算誤差である。これについては，まだ今後の課題として残されている。

直列型待ち行列モデルの効率の尺度として，最大流出率，maximum throughput rate が良く知られている。これを計算する為には，到着過程の代りに，第一段に最初から長さ無限大の待ち行列が存在するような，いわゆる倉庫付きモデルを考え，その出力率を計算すれば良い。このモデルの平衡確率は上で述べた考え方と全く同様に求めることができる。到着過程がなく，倉庫を除く第一段には常に c_1 人の客がいる，ということによる変更箇所は，(1) の $f(0,1)=0$, $f(0,0)=1$ と，(2), (3) で $q_1 \equiv c_1$ としなければならないところだけである。

4．数値例

いずれも，最大流出量を評価基準にしていくつかのモデルの比較を行ったものである。

4.1　二段モデル

第一段・第二段とも，窓口の構成・サービス能力を同等にする。各段のサービス能力，すなわち窓口数とサービス率を掛け合せたもの，を一定にしたまま，窓口数を増減した場合と，第二段の待合室の大きさを増減した場合について，最大流出率の変化を調べる。$c_1=c_2=c$，$l_1=l_2=l$，$c\mu_1=c\mu_2=1$　$c=1(1)5$，$N_2-c=0(1)4$，$l=1(1)3$　について調べたのが，表.1である。バイパスを作る為にサービス効率を平均的に低下させるより，バッファを置く効果の方が，スループットの向上に寄与し，その傾向は，サービス分布の変動係数の減少と共に顕著になることがわかる。なお，$c=5$，$N_2=9$の時，方程式の次元数は　$l=2$で360　$l=3$にすると　3675　になる。

4.2　三段モデル

ジョブショップ型のモデルで，最大流出率が最大になるようにサービス窓口を順序付ける問題を考える。各段とも窓口数が2，サービス能力が1，待合室の大きさが2（第一段を除く）であり，サービス分布だけが異なり，E_2，E_3，E_4の3

種類がある場合、そのサービス時間分布の並べ方によって最大流出率がどのように変化するかを示したのが表2である。どう並べても、結果にそれ程の差はみられない、というのがOR的結論であるが、その若干の違いを調べると、変動係数の大きいものを中間に置くと最も流れが良くなる、という、単一窓口直列型モデルの場合の結果と一部逆転していることが注目される。その理由についてはまだ解明されていない。なお、この表で判るように、単一窓口直列型モデルで成立していた容量の可逆性が、この場合には成立していない。

5. おわりに

ブロッキングが起こりうる複数窓口直列型待ち行列モデルの解析が殆んど手付かずの状態であるのに対して、アーラン型の分布を持つモデルを数値的に解くプログラムを作り、数値解法の可能性を考えた。このプログラムでは、方程式の次元をa、生成行列の非零要素の数をbとすると、主要なテーブル用の領域が$20a+8b$バイト必要である。a, bの比率はモデルの段数、サーバ数に依存する行列のスパースの度合を表わす尺度であるが、平均的に$b \approx 7a$と考えると、1MBのメモリを持つ計算機を使うと、次元が一万ぐらい迄のモデルならば解けることになる。その範囲のモデルとしては、う

ンダムに並べると、表3のようなものがある。方程式の次元を下げる為には、モデル全体を同時に解く代りに、全体をいくつかのサブシステムに分けて計算を積み上げていくなどの工夫が考えられるが、それについては今後の課題である。

Table.1 窓口数と待合室設置の効果

(1) M サービス

SERVER	BUFFER 0	1	2	3	4
1	0.6667	0.7500	0.8000	0.8333	0.8571
2	0.7500	0.8000	0.8333	0.8571	0.8750
3	0.7907	0.8269	0.8525	0.8714	0.8861
4	0.8161	0.8447	0.8655	0.8815	0.8940
5	0.8339	0.8576	0.8753	0.8891	0.9002

(2) E_2 サービス

SERVER	BUFFER 0	1	2	3	4
1	0.7273	0.8214	0.8682	0.8957	0.9137
2	0.7889	0.8487	0.8834	0.9054	0.9204
3	0.8217	0.8657	0.8935	0.9121	0.9252
4	0.8427	0.8775	0.9009	0.9171	0.9288
5	0.8576	0.8865	0.9067	0.9212	0.9318

(3) E_3 サービス

SERVER	BUFFER 0	1	2	3	4
1	0.7619	0.8587	0.9006	0.9235	0.9378
2	0.8106	0.8750	0.9086	0.9282	0.9409
3	0.8389	0.8866	0.9146	0.9319	0.9435
4	0.8574	0.8954	0.9194		

Table.2 窓口の最適配置

(l_1, l_2, l_3)	C=2	$\hat{C}$=2
(3,2,4)	0.74209	0.67627
(4,2,3)	0.74208	0.67627
(2,4,3)	0.74191	0.67695
(2,3,4)	0.74190	0.67658
(3,4,2)	0.74186	0.67695
(4,3,2)	0.74183	0.67658

Table.3 平衡方程式の次元の大きさ

($l_0 = 1$; $l_j = l$, $N_j = N$, $c_j = c$ for all j)

			K					
l	c	N	2	3	4	5	6	7
1	1	1	5	13	34	89	233	610
〃	〃	2	11	41	153	571	2131	7953
〃	2	2	12	51	219	942	4053	17439
2	1	2	29	169	985	5741	33461	
〃	〃	3	55	433	3409	26839		
〃	2	2	48	396	3276	27108		
〃	2	3	102	1170	13428			
3	2	2	130	1720	22780			

M/M/1(5) E /4(5) M/1(5) M/1(5) の場合 7555