混合型待ち行列網の計算アルゴリズム

紀 一誠
（日本電気 C&C システム研究所）

1. はじめに

本稿では，客のサービスセンタ間の移動経路を確率的に指定する複数個の閉鎖型部分連鎖と複数個の開放型部分連鎖が存在し，複数窓口のタイプ1のセンタを複数個含む4つのタイプのサービスセンタと複数個のサービスクラスが存在する一般的な混合型のBCMP型待ち行列網に関するたたみこみ型の計算方法について示す．

積形式解を持つ待ち行列網は，モデルと実システムとの対応が分り易い，積形式解を持つため取り扱い易い，数値計算で解を得るためモンテカルロ型のシミュレーションに較べて計算機実行時間が極めて短かくてすむ，等の理由により計算機システムや通信網の性能評価のためのモデルとして広く利用されている．

初期のものとしてはBuzen[1]により示されたGordon-Newell型[2]閉鎖型待ち行列網を基礎とするセントラルサーバモデルがあげられる．

1975年，Baskett et al.[3]により積形式解を持つ待ち行列網の範囲は大きく拡大され，それに伴い計算機システムの性能評価モデルとしての応用範囲も大きく広がった．（この待ち行列網は著者4名の頭文字をとってBCMP型の待ち行列網といわれる．） BCMP型待ち行列網の応用上最も重要な拡張点は，Gordon-Newell型の待ち行列網では客の網内の移動経路を確率的に指定する推移確率行列（マルコフ連鎖）は1個のみしか許されないのに対して，BCMP型の網ではこれらが複数個存在する事が許されるようになった点である． これらのマルコフ連鎖のことを部分連鎖（subchain）といい，各々は閉鎖型であっても開放型であってもよい．

異なるマルチプログラミング多重度を持ち，確率的に異なるふるまいをするサブシステムにそれぞれ部分連鎖を対応付けることにより，バッチ処理，トランザクション処理，TSS 等の性格の異なる複数個のサブシステムが一つのシステム内に混在する多次元処理システムのモデル化が可能になった．

BCMP型待ち行列網モデルを基礎とする性能評価向けのソフトウェア・パッケージとしてQNET4[4]，BEST/1[5]，PNET[6]，QSEC[7]，QM-X[8]等の開発が知られている．

現在のところ計算機システムのモデルとして用いられるものは総ての部分連鎖が閉鎖型である閉鎖型待ち行列網が殆んどであるが，開放型の部分連鎖が加わった混合型の待ち行列網モデルではモデル化がさらに一層柔軟に行えるようになる．

混合型待ち行列網に関する計算アルゴリズムは，Reiser and Kobayashi[9]，Chandy and Sauer[10]，Zahorjan[11]により扱かわれているがいずれも許される部分連鎖の数やサービスセンタの種類に制限があり，本稿で扱かう一般的な網に関する方法は知られていない． 本稿の方法はReiser and Kobayashi[9]，Reiser[12]により示された閉鎖型待ち行列網に関するたたみこみ型の計算方法を混合型へと拡張したものである．

2. モデル

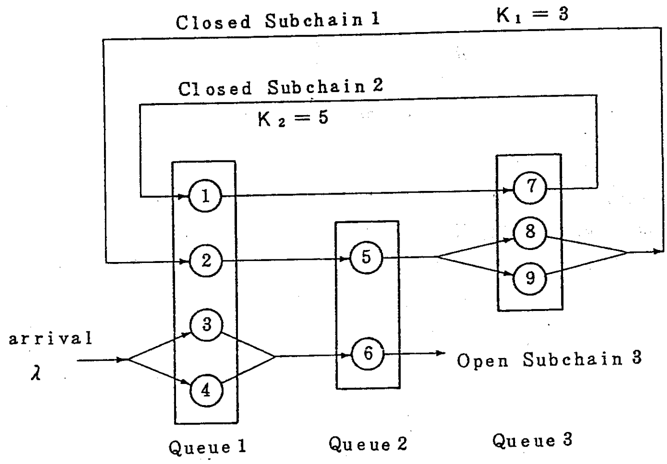
本稿で扱かう混合型待ち行列網モデルを以下に示す．

N：待ち行列中のサービスセンタの数， R：待ち行列網中のサービスクラスの集合，
R_i：サービスセンタ i に属するサービスクラスの集合（但し，$R=\bigcup_{i=1}^{N} R_i$ ，$R_i \cap R_j=\phi$，$\forall i \neq j$ とする．） S：Rの要素数，即ち待ち行列網中のサービスクラスの総数．

p_{ij} : サービスクラス i の客がサービス終了後サービスクラス j に移る確率.

$\mathbb{P}=\{p_{ij}\}$: p_{ij} を要素とする客の移動経路を表現する行列.

本稿では, $\mathbb{P}$ はM個のエルゴート的な部分連鎖 $\mathbb{P}_1, \mathbb{P}_2, \cdots, \mathbb{P}_M$ に分解可能(decomposable)であるとし, 1〜Lまでの部分連鎖は閉鎖型, L+1〜Mまでの部分連鎖は開放型であると仮定する.



(図 1) 混合型 待ち行列網の例

閉鎖型部分連鎖 ℓ に従って網内を移動する客数を K_ℓ とし, 閉鎖型部分連鎖に関する客数ベクトルを $\mathbb{K}=(K_1, K_2, \cdots, K_L)$ とする. 開放型部分連鎖に従がう客の網外からの到着はパラメータ λ_ℓ(定数)のポアソン到着とし, 到着客は確率 g_r でクラス r の客になるものとする.

C_ℓ を部分連鎖 ℓ に従がう客により訪問を受けるサービスクラスの集合とする. 但し, $R=\bigcup_{\ell=1}^{M} C_\ell$ であり, $C_\ell \cap C_m = \phi, \forall \ell \neq m$ とする.

N個のサービスセンタは(表1)に示す4種類のいずれかのタイプのものとする. 客はクラス毎に異なるサービス時間要求分布(Coxian)に従がうサービス時間を要求するものとする. 但し, タイプ1のセンタ i では客はそのクラスに無関係に同一のパラメータ C_i 持つ指数分布に従がうサービス時間を要求するものとする.

センタ i のサービス能力(人/時間)はそのセンタに存在する客の数 j の関数として $\hat{\mu}_i(j)$ と定義される. 客が1人だけセンタ i に存在する時のサービス能力を C_i とし, $\hat{\mu}_i(j)=\mu_i(j)C_i$ とする. サービスセンタのタイプ別のサービス能力 $\mu_i(j)$ の形を(表2)に示す. タイプ1のセンタの固定サービス率は単一窓口, 半可変サービス率を持つ場合は複数窓口(m)のモデルに対応している.

(表 2) センタ・タイプ別のサービス能力

センタ・タイプ		$\mu_i(j)$
1	固定サービス率	$\mu_i(j)=1,\quad 1 \le j$
	半可変サービス率	$\mu_i(j)=\begin{cases} j, & 1 \le j \le m_i \\ m_i, & m_i \le j \end{cases}$
	可変サービス率	$\mu_i(j)=\begin{cases} 1, & j=1 \\ \text{任意}, & 2 \le j \end{cases}$
2, 4		$\mu_i(j)=1,\quad 1 \le j$
3		$\mu_i(j)=j,\quad 1 \le j$

さらに次の記号を準備しておく.

$S_{i\ell}=R_i \cap C_\ell$: センタ i に於て部分連鎖 ℓ に従がう移動経路を持つ客によって訪

(表 1) サービスセンタのタイプ

タイプ	サービス規律	サービス要求時間分布
1	先着順(FCFS)	指数分布
2	プロセッサシェアリング(PS)	Coxian
3	扱い者数無限(IS)	〃
4	後着順・割込型・中断点再開(LCFS-PR)	〃

内される総てのサービスクラスの集合．　μ_r：$\mu_r=\hat{\mu}_r c_i$，$r\in R_i$，但し$\hat{\mu}_r$はクラスrの客の平均サービス要求時間$1/\hat{\mu}_r$の逆数とする．

$S_{i0}=\bigcup_{\ell=L+1}^{M} S_{i\ell}$：センタ$i$に於て開放型部分連鎖に従う客により訪内される総てのサービスクラスの集合．

このモデルの平衡状態におけるクラスrへの客の平均訪内回数θ_rは次の連立方程式の解として定まる．

$$\theta_r = q_r + \sum_{j=1}^{S} p_{jr}\theta_j\ ,\quad r=1,2,\cdots,S\ . \qquad \cdots(1)$$

上式は各部分連鎖に対応するM組の連立方程式に分解される．　この内，閉鎖型部分連鎖に関するものは同次方程式となりその解$\{\theta_r\}$は定数倍を除いて一意に定まり，平均訪内回数の相対値を示す．　開放型の場合には$\{\theta_r\}$は一意に定まり，$\lambda_\ell\theta_r\ (r\in C_\ell)$がクラス$r$への平均訪内回数の絶対値を示す．

次にクラスrへの負荷量e_rを下の如くに定義する．

$$e_r=\begin{cases}\theta_r/\mu_r\ , & r\in C_\ell\ ,\ \ell=1,2,\cdots,L\ ,\\ \lambda_\ell\theta_r/\mu_r\ , & r\in C_\ell\ ,\ \ell=L+1,\cdots,M\ .\end{cases} \qquad \cdots(2)$$

サービスセンタiにかかる負荷ベクトルを$\mathbf{e}_i=(e_r)$，$r\in R_i$とする．　また，平衡状態におけるクラスrの状態の客数をk_rとし，サービスセンタiの状態を表現するベクトルを$\mathbf{k}_i=(k_r)$，$r\in R_i$とする．　待ち行列網全体の状態はベクトル$\mathbf{k}$により，$\mathbf{k}=(\mathbf{k}_1,\mathbf{k}_2,\cdots,\mathbf{k}_N)$と表わすものとする．　混合型待ち行列網の例を(図1)に示す．　例に於ては，$N=3$，$S=9$，$M=3$，$L=2$，$\mathbf{K}=(K_1,K_2)=(3,5)$，$R_1=\{①,②,③,④\}$，$R_2=\{⑤,⑥\}$，$R_3=\{⑦,⑧,⑨\}$，$C_1=\{②,⑤,⑧,⑨\}$，$C_2=\{①,⑦\}$，$C_3=\{③,④,⑥\}$，$S_{11}=\{②\}$，$S_{12}=\{①\}$，$S_{13}=\{③,④\}$，$S_{21}=\{⑤\}$，---　等である．

3. 積形式解

以下を通じ，記法の簡略化のためベクトルに関する次のような表記法を定義する．　$\mathbf{x}$を非負の整数値を要素とするL次元のベクトル，$\mathbf{a}$を実数値を要素とするL次元のベクトルとし，x_i，a_iを各々の要素を示すものとする．

$$|\mathbf{a}|=a_1+a_2+\cdots+a_L\ ,\qquad \mathbf{x}!=x_1!\,x_2!\cdots x_L!\ ,$$
$$\mathbf{a}^{\mathbf{x}}=a_1^{x_1}a_2^{x_2}\cdots a_L^{x_L}\ .$$

さらに，$\sum_{C(\mathbf{x})}(\cdot)$は条件$C(\mathbf{x})$を満たす総ての可能な$\mathbf{x}$に関する和をとるものとする．　また，ベクトルの次元を$\dim(\cdot)$で表わす．　即ち，$L=\dim(\mathbf{x})=\dim(\mathbf{a})$.

先に示した待ち行列網の定常状態確率$P(\mathbf{k})$は次のような積形式で与えられることが知られている[3]．

$$P(\mathbf{k})=\prod_{i=1}^{N} f_i(\mathbf{k}_i)/g(\mathbf{K})\ . \qquad \cdots(3)$$

ここで$g(\mathbf{K})$は正規化定数であり，$f_i(\mathbf{k}_i)$はサービスセンタiが状態$\mathbf{k}_i$である正規化されない状態確率であり，次の形をしている．

$$f_i(\mathbf{k}_i)=a_i(|\mathbf{k}_i|)\frac{|\mathbf{k}_i|!}{\mathbf{k}_i!}\,\mathbf{e}_i^{\mathbf{k}_i}\ . \qquad \cdots(4)$$

(4)式の係数$a_i(\cdot)$はサービスセンタiに関する容量係数といわれるもので次式．

$$a_i(0)=1\ ;\quad a_i(k)=\{\prod_{j=1}^{k}\mu_i(j)\}^{-1}\ ,\quad k=1,2,\cdots \qquad \cdots(5)$$

4. 正規化定数の計算

状態確率(3)は積形式という美しい形をしているが実際の計算に際しては正規化定数$g(\mathbf{K})$の計算という壁を突破しなければならない．

状態ベクトル$\mathbf{k}$のとり得る総ての可能な状態集合をFとすれば，正規化定数$g(\mathbf{K})$は$P(\mathbf{k})$のF上の和が1という条件から次の如くに表現される．

$$g(\mathbf{K}) = \sum_{\mathbf{k} \in F} \prod_{i=1}^{N} f_i(\mathbf{k}) \quad . \qquad \text{--- (6)}$$

(6)式の計算法について考える. このためいくつかの記号と記法を準備する.

$\mathbf{k}_{io} = (k_r)$, $\mathbf{e}_{io} = (e_r)$, $r \in S_{io}$: センタ i に於る開放型部分連鎖に従がう客の状態ベクトルと負荷ベクトル.

$\mathbf{k}_{il} = (k_r)$, $\mathbf{e}_{il} = (e_r)$, $r \in S_{il}$, $l = 1, 2, \cdots, L$: センタ i に於て閉鎖型部分連鎖 l に従がう客の状態ベクトルと負荷ベクトル.

$\mathbf{k}_i = (\mathbf{k}_{io}, \mathbf{k}_{i1}, \cdots, \mathbf{k}_{iL})$: センタ i の状態ベクトル.

$D_l = \sum_{i=1}^{N} \dim(\mathbf{k}_{il})$, $l = 1, 2, \cdots, L$, $D = D_1 + D_2 + \cdots + D_L$: 閉鎖型部分連鎖 l に従がう客が訪問する総てのクラスの数, 及びその総和.

$\mathbf{K}_l$: 要素が総て K_l に等しく, $\dim(\mathbf{K}_l) = D_l$ である定数ベクトル.

$\hat{\mathbf{k}}_{il}$: 要素が非負整数値をとり, $\dim(\hat{\mathbf{k}}_{il}) = D_l$ である変数ベクトル.

$\hat{\mathbf{e}}_{il}$: 要素が次の如くに定義され, $\dim(\hat{\mathbf{e}}_{il}) = D_l$ である拡張された負荷ベクトル. $\hat{\mathbf{e}}_{il} = (\hat{e}_r)$, $\hat{e}_r = \{e_r, r \in S_{il} ; 0 \ , r \notin S_{il}\}$.

$\hat{\mathbf{e}}_i$: $\hat{\mathbf{e}}_i = (\hat{\mathbf{e}}_{i1}, \hat{\mathbf{e}}_{i2}, \cdots, \hat{\mathbf{e}}_{iL})$, $\hat{\mathbf{k}}_i$: $\hat{\mathbf{k}}_i = (\hat{\mathbf{k}}_{i1}, \hat{\mathbf{k}}_{i2}, \cdots, \hat{\mathbf{k}}_{iL})$.

ここに, 変数ベクトル $\hat{\mathbf{k}}_{il}$ は $\mathbf{0} \leq \hat{\mathbf{k}}_{il} \leq \mathbf{K}_l$ なる範囲 (D_l次元の超立方体) を動くものとする. 従って $\hat{\mathbf{k}}_i$ は総ての i について D 次元の超立方体内を動く.

* 閉鎖型待ち行列網 : 以上の準備のもとに, 総ての部分連鎖が閉鎖型である場合には App.(1) に示すたたみこみ演算を用いて (6) 式に示される正規化定数が計算できることを以下に示す. まず, 次の関数を定義する.

$$\hat{f}_i(\hat{\mathbf{k}}_i) = a_i(|\hat{\mathbf{k}}_i|) \frac{|\hat{\mathbf{k}}_i|!}{\hat{\mathbf{k}}_i!} \hat{\mathbf{e}}_i^{\hat{\mathbf{k}}_i} \quad . \qquad \text{--- (7)}$$

(7)式は $i = 1, 2, \cdots, N$ に対して等しく D 次元の超立方体上で定義された D 変数の関数である. $\hat{\mathbf{k}}_{il} = (\hat{k}_r)$, $\hat{k}_r = \{k_r, r \in S_{il} ; 0 \ , r \notin S_{il}\}$ とすれば, $|\hat{\mathbf{k}}_i| = |\mathbf{k}_i|$, $\hat{\mathbf{k}}_i! = \mathbf{k}_i!$, $\hat{\mathbf{e}}_i^{\hat{\mathbf{k}}_i} = \mathbf{e}_i^{\mathbf{k}_i}$ であるから次が成り立つ. 但し, $0^0 = 1$.

$$\hat{f}_i(\hat{\mathbf{k}}_i) = \begin{cases} f_i(\mathbf{k}_i) & , \ \hat{k}_r = \{k_r, r \in S_{il} ; 0 \ , r \notin S_{il}\}, \\ 0 & , \ \hat{k}_r = \{\text{上記以外}\}. \end{cases} \qquad \text{--- (8)}$$

(8)式より, 関数 $\hat{f}_i(\hat{\mathbf{k}}_i)$ に対してたたみこみ演算が実行でき, $i = 1$ から N までのたたみこみ結果から正規化定数 $g(\mathbf{K})$ は次のように計算される.

$$g(\mathbf{K}) = \sum_{\mathbf{z} \in G} (\hat{f}_1 * \hat{f}_2 * \cdots * \hat{f}_N)(\mathbf{z}) \qquad \text{--- (9)}$$

但し, $\mathbf{z}$ は $\mathbf{z}_l$ を非負整数値を要素とし $\dim(\mathbf{z}_l) = D_l$ なるベクトルとし, $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \cdots, \mathbf{z}_L)$ なる変数ベクトルとする. G は次で定義される集合とする.

$$G = \{\mathbf{z} \mid |\mathbf{z}_1| = K_1, |\mathbf{z}_2| = K_2, \cdots, |\mathbf{z}_L| = K_L\}.$$

$\mathbf{z}_l = (z_{lj})$, $j = 0 \sim D_l - 1$ とすれば $\sum_{\mathbf{z} \in G}$ は次のように展開される.

$$\sum_{\mathbf{z} \in G} = \sum_{z_{10}=0}^{K_1} \sum_{z_{11}=0}^{K_1 - z_{10}} \cdots \sum_{z_{1D_1-1}=0}^{K_1-(z_{10}+\cdots+z_{1D_1-2})} \cdots\cdots \sum_{z_{L1}=0}^{K_L} \sum_{z_{L2}=0}^{K_L - z_{L1}} \cdots \sum_{z_{LD_L-1}=0}^{K_L-(z_{L1}+\cdots+z_{LD_L-2})} .$$

従がって, $\hat{f}_i$ に関するたたみこみ結果 $(\hat{f}_1 * \hat{f}_2 * \cdots * \hat{f}_L)(\cdot)$ が得られればそれから直ちに正規化定数を計算することができる. 但し, (9) 式の表現は閉鎖型待ち行列網に関するたたみこみ型の計算法の原理を示すだけのもので, 実際に計算をする場合にはもっと効率の良い方法に依らねばならない.

以上は閉鎖型待ち行列網の場合の正規化定数の計算方法であった.
開放型部分連鎖が混在する混合型待ち行列網の場合には以上の方法をそのまま用いることはできない.

閉鎖型の場合には各部分連鎖に従がう網内客数 K_l は有限であり, 従って各セ

ンタでの状態を表現する関数 $f_i(\mathbf{k}_i)$ 又は $\hat{f}_i(\hat{\mathbf{k}}_i)$ の変数の定義域は有界である．

しかし，開放型部分連鎖が存在すると $dim(\mathbf{k}_{i0}) \to \infty$ となるため定義域は有界ではなくなりたたみこみ演算が実行できなくなる．　たたみこみ演算の定義を少々変更すれば形式的には (9) 式と同じような形を導くことは可能であるが，無限回の演算と無限の記憶容量を必要とするのであまり意味は無い．

本稿ではこの問題点を解決するため，センタ i の状態確率 $f_i(\mathbf{k}_i)$ に関して，そのセンタに存在する閉鎖型部分連鎖に従がう客のみに関する周辺分布を導くことにより有界な定義域上でのたたみこみ演算により混合型待ち行列網の正規化定数の計算を可能にした．

5. 周辺分布

センタ i に閉鎖型部分連鎖 l に従がう客が $n_l{}^{i}$ $(l=1,2,\cdots,L)$ 存在する周辺分布を導く．　次の記号と記法を用意する．

$[k]_n = k(k+1)\cdots(k+n-1)$ ，$[k]_0 = 1$ ，　k, n は非負整数．

$n_{il} = |\mathbf{k}_{il}|$ ，$\rho_{il} = |\mathbf{e}_{il}|$ ，$l = 0, 1, \cdots, L$ ：センタ i に於て，部分連鎖 l に従がう客の数及び負荷量，

$\mathbf{n}_i = (n_{i0}, n_{i1}, \cdots, n_{iL})$ ，$\mathbf{r}_i = (\rho_{i0}, \rho_{i1}, \cdots, \rho_{iL})$ ：センタ i に於る開放型部分連鎖に従がう客を含めた客数ベクトルと負荷ベクトル．

$\mathbf{x}_i = (n_{i1}, n_{i2}, \cdots, n_{iL})$ ，$\boldsymbol{\rho}_i = (\rho_{i1}, \rho_{i2}, \cdots, \rho_{iL})$ ：センタ i に於る閉鎖型部分連鎖に従がう客のみからなる客数ベクトルと負荷ベクトル．

$\mathbf{n} = \mathbf{n}_1 + \mathbf{n}_2 + \cdots + \mathbf{n}_N$ ，　$\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_N$ ．

混合型待ち行列網の状態が $\mathbf{x}$ である確率を $P(\mathbf{x})$ とする．　また，センタ i が状態 $\mathbf{x}_i$ である正規化されない状態確率を $g_i(\mathbf{x}_i)$ とする．

この時次の定理が成り立つ．

〔定理 1〕：　$P(\mathbf{x}) = \prod_{i=1}^{N} g_i(\mathbf{x}_i) / g(\mathbf{K})$,

$$g_i(\mathbf{x}_i) = v_i(|\mathbf{x}_i|, \rho_{i0})\,|\mathbf{x}_i|!\, \boldsymbol{\rho}_i^{\mathbf{x}_i} / \mathbf{x}_i! \quad , \qquad \cdots(10)$$

ここに，$v_i(k, \rho_{i0}) = \sum_{l=0}^{\infty} a_i(k+l)[k+1]_l\, \rho_{i0}^{l} / l!$ ．

（証明）：　$f_i(\mathbf{k}_i)$ の $\mathbf{x}_i$ に関する周辺分布が $g_i(\mathbf{x}_i)$ である事を示せば十分．

以下記法の簡略化のため，$f_i(\cdot)$, $\mathbf{k}_i$, $\mathbf{k}_{il}$, $a_i(\cdot)$, $\mathbf{e}_i$, $\mathbf{e}_{il}$, $\boldsymbol{\rho}_i$, $\mathbf{r}_i$, $\mathbf{n}_i$, $\mathbf{x}_i$, n_{il}, ρ_{il}, S_{il}, $v_i(\cdot)$　等の添字 i を省略する．　$\mathbf{k}_l = (k_{r_l})$, $r_l \in S_l$, $l = 0 \sim L$ ．

定義より，　$g(\mathbf{x}) = \sum_{n_0=0}^{\infty} \sum_{r_0 \in S_0} \sum_{r_1 \in S_1} \cdots \sum_{r_L \in S_L} f(\mathbf{k}_0, \mathbf{k}_1, \cdots, \mathbf{k}_L)$ ．

まず，センタ i が状態 $\mathbf{n}_i$ である正規化されない確率 $p(\mathbf{n}_i) = \sum_{r \in S_0} \cdots \sum_{r \in S_L} f(\cdot)$ を求める．　$|\mathbf{k}_i| = |\mathbf{n}_i|$ および多項係数に関する公式 $|\mathbf{e}_l|^{|\mathbf{k}_l|} = \sum_{r \in S_l} |\mathbf{k}_l|!\, \mathbf{e}_l^{\mathbf{k}_l} / \mathbf{k}_l!$ に注意して以下の関係を得る．

$$\begin{aligned} p(\mathbf{n}) &= \sum_{r_0 \in S_0} \sum_{r_1 \in S_1} \cdots \sum_{r_L \in S_L} a(|\mathbf{k}|)\,|\mathbf{k}|!\, \mathbf{e}_0^{\mathbf{k}_0} \mathbf{e}_1^{\mathbf{k}_1} \cdots \mathbf{e}_L^{\mathbf{k}_L} / \mathbf{k}_0!\,\mathbf{k}_1! \cdots \mathbf{k}_L! \\ &= a(|\mathbf{n}|)\,|\mathbf{n}|!\, |\mathbf{e}_0|^{|\mathbf{k}_0|} |\mathbf{e}_1|^{|\mathbf{k}_1|} \cdots |\mathbf{e}_L|^{|\mathbf{k}_L|} / |\mathbf{k}_0|!\,|\mathbf{k}_1|! \cdots |\mathbf{k}_L|! \\ &= a(|\mathbf{n}|)\,|\mathbf{n}|!\, \boldsymbol{\rho}^{\mathbf{n}} / \mathbf{n}! \quad . \end{aligned}$$

次に $g(\mathbf{x}) = \sum_{n_0=0}^{\infty} p(\mathbf{n})$ を求める．

$$\begin{aligned} g(\mathbf{x}) &= \sum_{n_0=0}^{\infty} a(n_0 + |\mathbf{x}|)(n_0 + |\mathbf{x}|)!\, \rho_0^{n_0} \boldsymbol{\rho}^{\mathbf{x}} / n_0!\,\mathbf{x}! \\ &= \sum_{n_0=0}^{\infty} a(n_0 + |\mathbf{x}|)[|\mathbf{x}| + 1]_{n_0} (\rho_0^{n_0} / n_0!)\,|\mathbf{x}|!\, \boldsymbol{\rho}^{\mathbf{x}} / \mathbf{x}! \\ &= v(|\mathbf{x}|, \rho_0)\,|\mathbf{x}|!\, \boldsymbol{\rho}^{\mathbf{x}} / \mathbf{x}! \quad . \end{aligned}$$

（以上）

〔系 1〕：　系が定常解を持つためには，$i = 1, 2, \cdots, N$ に対して次が必要．

$$\mu_i(\infty) < \rho_{i0} \qquad \cdots(11)$$

（証明）：　$v(k, \rho_0)$ の収束条件より得られる．

サービスセンタのタイプによりv(k, ρ₀)の形は異なっている。

センタ・タイプ別の$v(k,\rho_0)$の形を（表３）に示す。

（表　３）　センタ・タイプ別の$v(k,\rho_0)$の形

サービスセンタタイプ			$v(k,\rho_0)$
1	固定サービス率		$1/(1-\rho_0)^{k+1}$
	半可変サービス率	$k \le m-1$	$m^m/(m-1)!\,(m-\rho_0)^{k+1} + \sum_{\ell=0}^{m-k-1} \left\{ \frac{1}{k!} - \frac{[k+1]_\ell}{m!} \left(\frac{1}{m}\right)^{\ell+k-m} \right\} \frac{\rho_0^\ell}{\ell!}$
		$m \le k$	$m^m/(m-1)!\,(m-\rho_0)^{k+1}$
2, 4			$1/(1-\rho_0)^{k+1}$
3			$e^{\rho_0}/k!$

6. 計算アルゴリズム

定理１より，正規化定数$g(\mathbf{K})$は次のように表現することができる。

$$g(\mathbf{K}) = \sum_{\mathbf{x}=\mathbf{K}} \prod_{i=1}^{N} g_i(\mathbf{x}_i)\ .$$

$g_i(\mathbf{x}_i)$は$i=1,2,\cdots,N$について等しく有界な定義域$\mathbf{0} \le \mathbf{x}_i \le \mathbf{K}$を持つ関数であり，従ってApp.(1)で定義されたたたみこみ演算が実行でき，正規化定数は次の如くに計算される。

$$g(\mathbf{K}) = (g_1 * g_2 * \cdots * g_N)(\mathbf{K})\ . \qquad \cdots (12)$$

(12)式は正規化定数が各センタの正規化されない状態確率のたたみこみ列として得られることを示している。　以下(12)式の計算手順について考える。

今，gおよびgをそれぞれℓ次元方向に，$0,1,2,\cdots,k_\ell$とインデックスされたL次元の記憶領域とする。　即ち，プログラム風に定義すれば（図２）のDim.の如くになる。　(12)式は正規化定数が次の２つの手順を$i=1$からNまで繰返して得られる事を示している。

① 記憶領域g上に状態確率$g_i(\mathbf{x}_i)$を作り出す。

② $i-1$までのセンタに関するたたみこみ結果が格納されているgとgをたたみこみ，その結果を新たなgとする。

以上の手順をプログラム風に表現したものを（図２）に示す。　但し，記憶領域$g(x_1,x_2,\cdots,x_L)$は$g(\mathbf{x})$とする等の表現をするものとし，For $\mathbf{x}$: C の表現は条件Cを満たす総ての$\mathbf{x}$に関する実行を指示するものとする。

（図２）に於て，A_1が①を実行する部分，A_2が②のたたみこみを実行する部分である。　[アルゴリズムA]の完了時点で，記憶領域gのコーナ$g(\mathbf{K})$の部分に求める正規化定数$g(\mathbf{K})$が得られている。

さらに，①の部分の効率的な計算方法について考える。

$$m(k,\rho_0) = v(k,\rho_0)/v(k-1,\rho_0)$$

なる関数を定義する。

Dimension $g(k_1,k_2,\cdots,k_L)$,
$g(k_1,k_2,\cdots,k_L)$

$$g(\mathbf{x}) \leftarrow \begin{cases} 1, & \mathbf{x}=\mathbf{0} \\ 0, & \mathbf{x} \ne \mathbf{0} \end{cases} \qquad \text{(初期条件)}$$

For $i = 1$ to N
- A_1: For $\mathbf{x}$: $\mathbf{0} \le \mathbf{x} \le \mathbf{K}$
  - $g(\mathbf{x}) \leftarrow g_i(\mathbf{x})$
  - Next $\mathbf{x}$
- A_2: For $\mathbf{x}$: $\mathbf{0} \le \mathbf{x} \le \mathbf{K}$
  - $g(\mathbf{x}) \leftarrow \sum_{\mathbf{0} \le \mathbf{y} \le \mathbf{x}} g(\mathbf{y})\,g(\mathbf{x}-\mathbf{y})$
  - Next $\mathbf{x}$

Next i

（図２）　アルゴリズムA

(10) 式はこの $m(k,\rho_o)$ を用いて次のように変形することができる。

$$g(\mathbf{x}) = m(|\mathbf{x}|,\rho_o)\sum_{\ell=1}^{L}\rho_\ell\, g(\mathbf{x}-\mathbf{e}_\ell) \quad \cdots (13)$$

ここで $\mathbf{e}_\ell$ は ℓ 方向の単位ベクトルとする。 $m(k,\rho_o)$ の形はセンタのタイプにより異なる。 これを（表4）に示す。

(13) 式を利用すると ① の部分の計算を効率的に行なうことができる。（図3）にこの手順をプログラム風に表現したものを示す。 但し，記憶領域の未定義部分はすべて 0 が格納されているものとする。 （図2）の［アルゴリズム A］の中の A_1 の部分を（図3）に示す［アルゴリズム B］で置き代えてやればよい。

(表 4) センタ・タイプ別の $m(k,\rho_o)$

サービスセンタ・タイプ			$m(k,\rho_o)$
1	固定サービス率		$1/(1-\rho_o)$
	半可変サービス率	$k \le m-1$	$v(k,\rho_o)/v(k-1,\rho_o)$
		$m \le k$	$1/(m-\rho_o)$
2, 4			$1/(1-\rho_o)$
3			$1/k$

(初期設定)

$$g(0) \leftarrow \begin{cases} e^{\rho_o} & \text{，タイプ3のセンタ} \\ 1 & \text{，上記以外の場合} \end{cases}$$

```
For k = 0 to |K|
    For x : |x| = k
        g(x) ← m(k, ρo) Σ_{ℓ=1}^{L} ρℓ g(x − eℓ)
    Next x
Next k
```

(図 3) アルゴリズム B

7. 例題

（図1）に示したモデルを例にとり計算例を示す。 （表5）にサービスセンタに関する特性を示し，（表6）に部分連鎖に関する条件を示す。

$\mathbb{P}_1,\mathbb{P}_2,\mathbb{P}_3$ について (1) 式より θ_r を求めると次のようになる。 $\theta_1=\theta_2\,(=1)$; $\theta_2=\theta_5\,(=1)$, $\theta_8=0.7$, $\theta_9=0.3$; $\theta_3=0.4$, $\theta_4=0.6$, $\theta_6=1.0$. 以上より負荷ベクトルは次。 $\mathbf{e}_{10}=(\lambda\theta_3/\mu_3,\ \lambda\theta_4/\mu_4)=(0.06,\ 0.12)$, $\mathbf{e}_{11}=(\theta_1/\mu_1)=(0.1)$, $\cdots$, $\mathbf{e}_{20}=(\lambda\theta_6/c_2)=(0.5)$, $\mathbf{e}_{21}=(\theta_5/c_2)=(1.0)$, $\cdots$, 等。 従って，$\mathbf{r}_1=(0.18,\ 0.1,\ 0.2)$, $\mathbf{r}_2=(0.5,\ 0.5)$, $\mathbf{r}_3=(0,\ 1.66,\ 1.4)$; $\rho_1=(0.1,\ 0.2)$, $\rho_2=(0.5,\ 0)$, $\rho_3=(1.66,\ 1.4)$. 以上の条件のもとに計算すると，$g(\mathbf{K})=g(3,5)=1.3461$ となる。

(表 5) サービスセンタに関するパラメータ

センタ 1	センタ 2	センタ 3
タイプ 4 (LCFS-PR)	タイプ 1 (FCFS) 窓口数 $m_2=2$	タイプ 3 (無限サーバ)
$\tilde{\mu}_1=1/0.1$ $\tilde{\mu}_2=1/0.2$ $\tilde{\mu}_3=1/0.3$ $\tilde{\mu}_4=1/0.4$	$\tilde{\mu}_5=$ — $\tilde{\mu}_6=$ —	$\tilde{\mu}_7=1/0.7$ $\tilde{\mu}_8=1/0.8$ $\tilde{\mu}_9=1/0.9$
$C_1=1$ $(\mu_r=\tilde{\mu})$	$C_2=1$	$C_3=0.5$ $(\mu_r=0.5\tilde{\mu}_r)$

(表 6) 部分連鎖に関する条件

部分連鎖 1	部分連鎖 2	部分連鎖 3
閉鎖型	閉鎖型	開放型
網内客数 K_1 $K_1=3$	網内客数 K_2 $K_2=5$	到着率 λ $\lambda=0.5$
$\mathbb{P}_1=$ $\begin{array}{c\|cc} & 1 & 7 \\ \hline 1 & 0 & 1 \\ 7 & 1 & 0 \end{array}$	$\mathbb{P}_2=$ $\begin{array}{c\|cccc} & 2 & 5 & 8 & 9 \\ \hline 2 & 0 & 1 & 0 & 0 \\ 5 & 0 & 0 & 0.7 & 0.3 \\ 8 & 1 & 0 & 0 & 0 \\ 9 & 1 & 0 & 0 & 0 \end{array}$	$\mathbb{P}_3=$ $\begin{array}{c\|cccc} & 0 & 3 & 4 & 6 \\ \hline 0 & 0 & 0.4 & 0.6 & 0 \\ 3 & 0 & 0 & 0 & 1 \\ 4 & 0 & 0 & 0 & 1 \\ 6 & 1 & 0 & 0 & 0 \end{array}$

さらに，センタ1の使用率は78.1％，センタ2の1つの窓口当りの使用率は70.5％，センタ3の呼量は5.12アーラン，等の諸量も計算できる．

8. まとめ

混合型待ち行列網に関するたたみこみ型の計算方法について述べた．本稿で示した方法は，性能評価用ソフトウェア・パッケージQM-Xの計算エンジンとして実現され実用に供されている．また，本稿では省略したがサービスセンタでの滞在時間についても同様のたたみこみ演算により計算することができる．

待ち行列網に関する計算方法は大きく分けるとたたみこみ系統に属するものとMVA (Mean Value Analysis) の系統に属するものとに分けることができる．たたみこみ系統のものは指数部のあふれを発生しやすい，MVA系統のものは桁落を発生しやすいという数値計算上の不安定さを持っており，それを克服するための工夫がいろいろと研究されている．算法はそれぞれ一長一短ではあるが，小規模で固定サービス率のタイプ1センタしか含まないような網をプログラマブル電卓レベルでも計算したい場合にはMVA系統が優れており，種々のタイプのセンタを含む中あるいは大規模な網を扱かう場合にはその汎用性から見てたたみこみ法が向いていると思われる．本稿で扱かった一般的な混合型待ち行列網はMVA系統の算法では現在のところ扱えない．

待ち行列網の応用に際しては効率的な算法の開発が欠かせない．数値計算上の安定化の研究及び積形式解を持たぬ網の数値的近似解法は今後の課題である．

謝辞： 日頃御指導いただく当社C&Cシステム研究所，三上徹所長代理，同応用システム研究部，竹谷誠研究課長に感謝いたします．

Appendix (1) たたみこみ演算

$\mathbf{K}$をその要素が非負整数値であるL次元ベクトル，$\mathbf{x}$もその要素が非負整数値をとるL次元の変数ベクトルとする．$\mathbf{x}$は $\mathbf{0} \leq \mathbf{x} \leq \mathbf{K}$ の範囲を動くものとする．$a_1(\mathbf{x})$および$a_2(\mathbf{x})$は$\mathbf{x}$を変数とする実数値をとる関数とする．次の演算により得られる関数$a(\mathbf{x})$をa_1とa_2をたたみ込んで得られる関数といい，次のように表わす．$a(\mathbf{x}) = (a_1 * a_2)(\mathbf{x})$，$\mathbf{0} \leq \mathbf{x} \leq \mathbf{K}$．$\mathbf{y}$を非負整数値をその要素とするインデックス・ベクトルとする．

$$a(\mathbf{x}) = \sum_{\mathbf{0} \leq \mathbf{y} \leq \mathbf{x}} a_1(\mathbf{y}) \cdot a_2(\mathbf{x} - \mathbf{y}), \quad \mathbf{0} \leq \mathbf{x} \leq \mathbf{K}$$

$$= \sum_{y_1=0}^{x_1} \sum_{y_2=0}^{x_2} \cdots \sum_{y_L=0}^{x_L} a_1(y_1, y_2, \cdots, y_L) \cdot a_2(x_1 - y_1, x_2 - y_2, \cdots, x_L - y_L)$$

$a(\mathbf{x})$も $\mathbf{0} \leq \mathbf{x} \leq \mathbf{K}$ で定義された実数値をとる関数となる．

定義より，$(a_1 * a_2)(\mathbf{x}) = (a_2 * a_1)(\mathbf{x})$ は明らか．また，多重のたたみこみはたたみこみの繰り返しとして定義されるものとする．即ち次．

$$a_1 * a_2 * \cdots * a_N = a_1 * (a_2 * (a_3 * \cdots * (a_{N-1} * a_N)) \cdots).$$

【参考文献】

1) Buzen, J. P. : CACM, **16**, 9 (1977). 2) Gordon and Newell : Oper. Res., **15**, 2 (1967). 3) Baskett et al. : JACM, **22**, 2 (1975). 4) Reiser, M. : IBM Syst. J., **15**, 4 (1976). 5) Buzen et al. : Proc. AFIP NCC (1978). 6) Bruell and Balbo : The Computer Sic. Lib. **7**, North Halland (1980). 7) 池条 他：信学全大 (1978). 8) 紀 他：情処全大24回，4M-1 (1982). 9) Reiser and Kobayashi : IBM Res. Dev. **19** (1975). 10) Chandy and Sauer : CACM, **23**, 10 (1980). 11) Zahorjan, J. : Proc. SIGMETRICS, **10**, 3 (1981). 12) Reiser, M. : TIMS Studies in Manage. Sci. **7** (1977).