

非積形式型 FIFO ノードをもつ待ち行列網の 近似解法†

紀 一 誠††

非積形式型の先着順制御ノードをもつ待ち行列網を分解近似法 (decomposition approximate method) を用いて解く際には、先着順制御ノードにおける客の到着順序を情報にもつように系の状態を設定しなければならない。そのため、網の規模が大きくなるにつれ系を記述する状態数が飛躍的に増加し、解くべき連立方程式の次元数がたちまち実行可能な範囲を越えてしまうという問題点が顕在化してくる。本稿ではこの問題点を解決するため、系を表現する状態数を縮約 (lump) し解くべき方程式の規模を小規模なものとする方法について述べる。状態を縮約するために本稿ではランダム選択モデルを用いる。また、本近似法による近似解と厳密解との比較を示す数値実験例も示される。

1. ま え が き

本稿では、非積形式型の FIFO ノードをもつ待ち行列網を分解近似法を用いて解く場合に現れる大規模な連立方程式に関して、その状態数を削減し解くべき方程式の規模を縮退させる方法について示す。

積形式解をもつ待ち行列網は、Baskett et al.¹⁾によってその範囲を拡大されて以来、計算機システムおよび通信システムの性能を定量的に評価するための数理的モデルとしてその有効性を認められ、現在に至るまで数多くの研究がなされ、またその実用化をめざす性能評価用ソフトウェアパッケージの開発も数多く報告されている。積形式解は美しく扱いやすい形をしておりその応用範囲も広いが、現実のシステムのモデル化に際しては必ずしも常に積形式解をもつ網のみが現れるとは限らないため、積形式解をもたない待ち行列網に関する近似解法の研究も応用上欠かせない。

本稿では、複数の閉鎖型部分連鎖をもつ待ち行列網において、各部分連鎖 (クラス) ごとに異なるパラメータをもつ指数分布に従うサービス要求時間をもつ客を先着順にサービスを行うようなノードが存在する場合について考察する。そのようなノードは非積形式型ノードであり、従ってこの待ち行列網は積形式解をもっていない。

この種の待ち行列に関する近似解法は文献 3) および文献 5) に考察されている。文献 3) では非積形式ノードをクラスに対応する pseudo サーバ群に分割しそのサービス率を非線型方程式を解くことによって定

める方法が提案されている。文献 5) では非積形式ノードをフィードバックループをもつ積形式型の FIFO ノードに置き換え、全体を積形式解をもつ網として解く近似法が示されている。

本稿では分解近似法を基礎とする近似法について述べる。分解近似法は、Chandy et al.²⁾により提案され、Parametric Analysis, Flow Equivalent Method, Decomposition-Approximate Method, Norton (または Thévenin) の定理、等の数多くの名称でよばれており、訳語も定まっていないが本稿ではこの名称を用いることにする。

分解近似法は、積形式解をもつ待ち行列網に関する基本算法であるたたみこみ法との親和性が良いという利点をもつ一方、網の規模が大きくなるにつれ次の二つの問題点が発生してくる。

- 1) 積形式型ノード群をひとまとめでした合成ノードと非積形式型ノードから構成される 2 ノード網を解くため、大規模な連立方程式を解く必要が生ずる。
- 2) 非積形式型ノードに関する近似解が得られた後、それに基づき積形式型ノードに関する諸量を計算する際に、たたみこみ演算の繰り返しによる膨大な演算量が発生する。

本稿に示す方法は上記 1) の問題点の解決をねらいとしたものである。解くべき方程式の次元数を削減するために、2 ノード網の状態を表現する状態集合を縮約 (lump) することを考える。この際厳密な意味では縮約可能条件 (lumpability) は成立しないが、適当なモデルを仮定することによって縮約した状態に関する推移率を近似的に求めることができる。本稿ではこのモデルとしてランダム選択モデルを仮定した。

次章に本稿で扱うモデルを示す。3章に分解近似法

† An Approximate Method for Queueing Networks with a Non-Product-Form FIFO Node by ISSEI KINO (C&C Systems Research Laboratories, NEC Corporation).

†† 日本電気(株) C&C システム研究所

による解法と状態集合の縮約に関する考察を示す。

4章にランダム選択ノードを用いた近似法について述べ、5章に数値実験例による精度の検証を示す。6章にまとめを行う。

2. モデル

Q_0, Q_1, \dots, Q_N なる $N+1$ 個のノードから構成される待ち行列網を Q とする。網内には L 個の閉鎖型部分連鎖に従いノード間を移動する L 種類の客が存在するものとし、部分連鎖 j に従う客をクラス j の客とよぶことにする。

客数ベクトルを、 $\mathbf{K} = (K_1, K_2, \dots, K_L)$ とする。 Q_1 から Q_N までのノードは任意の積形式型ノードとする。 Q_0 はサーバ数 m で先着順サービス規律をもつものとし、クラス j の客の Q_0 におけるサービス要求時間はパラメータ μ_j をもつ指数分布に従うものとする。

クラス j の客のノード Q_i への相対訪問回数を θ_{ij} とする。これらは、客のノード間推移を表現する推移確率行列を係数とする斉次な連立方程式を解くことにより、定数倍を除き一意に定めることができる。本稿では Q_0 への訪問回数を基準にとることにする。すなわち、 $\theta_{0j} = 1, j = 1, 2, \dots, L$ とする。

次に、待ち行列網 Q から他のパラメータはそのままに保ち、ノード Q_i のみを取り除く（サービス時間をゼロとする）ことにより構成される待ち行列網を元の網 Q に関する i -complement な網といい、 $Q_{[i]}$ と記すことにする。

本稿のモデルでは、 Q_0 が非積形式型のノードであるので網 Q は積形式解をもたないが、0-complement な網 $Q_{[0]}$ は積形式解をもつ待ち行列網となる。

3. 分解近似法と状態の縮約

3.1 分解近似法の考え方

分解近似法の基本的な考え方は次のごとくである。網 Q の網内客数ベクトルを $\mathbf{k} = (k_1, k_2, \dots, k_L)$ 、 $0 \leq k \leq K$ とし、 Q_0 のサービス時間をゼロとしたときの Q_0 におけるクラス j の客に関するスループット $\tau_j(\mathbf{k})$ は次に与えられる。

$$\tau_j(\mathbf{k}) = G_{[0]}(\mathbf{k} - \mathbf{e}_j) / G_{[0]}(\mathbf{k}), \quad j = 1, 2, \dots, L. \quad (3.1)$$

ただし、 $G_{[0]}(\mathbf{k})$ は網 $Q_{[0]}$ に関する正規化定数、 \mathbf{e}_j を L 次元ベクトル空間の j 方向の単位ベクトルとする。(3.1) に示されるスループットは、とりもなおさ

ず $Q_{[0]}$ のスループットに等しいことに注意。

この事実を利用して $Q_{[0]}$ とスループット等価となる合成ノード Q_c を構成する。すなわち、 Q_c はその状態が \mathbf{k} であるときにクラス j の客へのサービス時間分布が (3.1) に与えられるパラメータ $\tau_j(\mathbf{k})$ をもつ指数分布に従うようなノードとして構成される。この合成ノード Q_c と非積形式型ノード Q_0 から成る 2 ノード網 (Q_0, Q_c) を構成し、網 (Q_0, Q_c) を解くことにより得られる Q_0 に関する状態確率を、元の網 Q を解いて得られる Q_0 に関する状態確率の近似値とみなすことが分解近似法の基本的な考え方である。

客の網 $Q_{[0]}$ における滞在時間分布は一般に指数分布とはならないが、これを指数分布とみなす点が近似法である理由である。なお、 Q_0 が積形式型ノードの場合には分解近似法により得られる解は厳密解に一致する。

分解近似法は合成ノード Q_c の構成に際して正規化定数が陽に現れるため、積形式解をもつ待ち行列網に関する基本的な計算法であるたたみこみ法⁹⁾になじみやすいという利点をもっている。

一方、解くべき網の規模が大きくなるに従って、分解近似法においては次の二つの問題点が顕在化してくる。1) 合成ノード Q_c と非積形式型ノード Q_0 から成る 2 ノード網を解くために大規模な連立方程式を解く必要が生ずる。2) $Q_{[0]}$ を構成する積形式型のノードに関する諸量を計算する際に、たたみこみ演算の繰り返しによる膨大な演算量が発生する。

分解近似法の実用化に際しては、問題の大規模化への対策、すなわち上記の二つの問題に対する対策を欠かすことができない。前章に示したモデルも分解近似法を直接的に使用して解くことが原理的には可能であるが、実用規模の網の計算に際しては、直接的適用は計算機の能力をはるかに越えてしまう。

本稿では、2章に示されるモデルについて前記の問題点 1) の解決法に関して考察する。なお、問題点 2) に関する一つの解決法は文献 4) に示される。

3.2 2 ノード網 (Q_0, Q_c) の推移行列

本節では網 (Q_0, Q_c) に関する推移行列（無限小生成作用素：infinitesimal generator） A を構成する。ノード Q_0 に関する平衡状態確率ベクトルを \mathbf{x} とすれば、 \mathbf{x} は次の連立方程式の解として定めることができる。ただし、 $|\mathbf{x}|$ は \mathbf{x} の要素和を表すものとする。

$$\mathbf{x}A = \mathbf{0}, \quad |\mathbf{x}| = 1. \quad (3.2)$$

ノード Q_0 は先着順サービス規律をもつので、 $(Q_0,$

Q_i の動きをマルコフプロセスとして記述するために客の到着順序を情報にもつように状態を設定しなければならない。本稿では次のごとくに系の状態を設定する。

ノード Q_0 の状態を、 $\mathbf{c}=(c_1, c_2, \dots, c_{|\mathbf{K}|})$ とする。ただし、 c_i は Q_0 において到着の早い順に客に 1 から $|\mathbf{K}|$ まで番号付けをしたときに i 番号に来る客のクラス番号とする。 i 番目の位置に客が存在しない場合は、 $c_i=0$ とする。状態 \mathbf{c} の実現可能な集合 (feasible set) を F とすると次で与えられる。

$$F = \{ \mathbf{c} | 0 \leq c_j \leq L, i=1, 2, \dots, |\mathbf{K}|; 0 \leq r_j(|\mathbf{K}|) \leq K_j, j=1, 2, \dots, L \}. \quad (3.5)$$

ここで、 $r_j(k)$ は次に定義される Indicator 関数を用いて (3.5) に定義される関数であり、先頭から k 番目までの内でクラス j に属する客の数を表している。

$$I(C) = \{1, C \text{ is true}; 0, C \text{ is false}\}. \quad (3.4)$$

$$r_j(k) = \sum_{i=1}^k I(c_i = j), \quad j=1, 2, \dots, L. \quad (3.5)$$

$\mathbf{r}(k) = (r_1(k), r_2(k), \dots, r_L(k))$ とし、 $\mathbf{c}_{(i)}$ を次に定義されるベクトルとする。

$$\mathbf{c}_{(i)} = (c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_{|\mathbf{K}|}, 0). \quad (3.6)$$

網 (Q_0, Q_i) 内での客の移動は Q_0 からの退去と Q_0 への到着の 2 種類に分けることができる。状態 \mathbf{c} である Q_0 からの退去は i 番目の客 ($i=1, 2, \dots, m$) が退去率 μ_{c_i} をもって退去することにより発生し、退去後の状態は $\mathbf{c}_{(i)}$ に変化する。ただし、 $\mu_0=0$ とする。一方、状態 \mathbf{c} である Q_0 へのクラス j の客の到着は (3.1) より到着率 $\tau_j(\mathbf{K}-\mathbf{r}(|\mathbf{K}|))$ で発生し、発生後の状態は、 $\mathbf{c} + j\mathbf{e}_{|\mathbf{r}(\mathbf{K})|+1}$ に変化する。ただし、 $\mathbf{e}_{|\mathbf{K}|}$ は $|\mathbf{K}|$ 次元空間における $|\mathbf{k}|$ 方向の単位ベクトルとする。また、到着の発生し得る状態 \mathbf{c} においては常に $c_{|\mathbf{K}|}=0$ であることに注意。以上をまとめて、推移行列 A は次のように構成できる。状態 \mathbf{c} から \mathbf{c}' への推移を表す要素を $a(\mathbf{c}, \mathbf{c}')$ とする。 $A = \{a(\mathbf{c}, \mathbf{c}')\}, \forall \mathbf{c} \in F$.

$$\left. \begin{aligned} a(\mathbf{c}, \mathbf{c}_{(i)}) &= \mu_{c_i}, \quad i=1, 2, \dots, m \\ a(\mathbf{c}, \mathbf{c} + j\mathbf{e}_{|\mathbf{r}(\mathbf{K})|+1}) &= \tau_j(\mathbf{K}-\mathbf{r}(|\mathbf{K}|)), \\ & \quad j=1, 2, \dots, L \\ a(\mathbf{c}, \mathbf{e}) &= -\{ \sum_{i=1}^m \mu_{c_i} + \sum_{j=1}^L \tau_j(\mathbf{K}-\mathbf{r}(|\mathbf{K}|)) \}. \end{aligned} \right\} \quad (3.7)$$

3.3 状態数の増加

前節 (3.7) により与えられる推移行列 A を用いて連立方程式 (3.2) を解くことにより求むる平衡状態確率 \mathbf{x} を得ることができる。しかし、先に述べたように、網の規模が大きくなるにつれ実現可能な状態数は増加し、解くべき連立方程式の次元数も増加する。本節で

は簡単な例をとり、状態数の増加の様子を調べておく。

Q_0 がシングルサーバ、すなわち $m=1$ であり、 $L=2$ の場合を考える。客数ベクトルを $\mathbf{K}=(K, K)$ とした場合の実現可能な状態数 S は次のごとくに与えられる。

$$S = \sum_{n=0}^{2K} 2^n - 2 \sum_{n=0}^{K-1} \sum_{u=0}^n \binom{K+n+1}{u}. \quad (3.8)$$

上式により状態数 S を試算してみると、次のようになる。 $K=2, S=19; K=3, S=69; K=5, S=923; K=10, S=705431$ 。以上に見られるように、例にとりあげた簡単な場合でも網の規模の増加につれ状態数が飛躍的に増加して行くことがわかる。

3.4 状態の縮約

実用規模の問題に分解近似法を用いる場合には何らかの方法で状態数の削減をはかる必要がある。本節では本稿のモデルに関して状態を縮約 (lump) する方法について考察する。まず若干の準備をし、推移行列 A を扱いやすい形に整理しておく。

$F(\mathbf{k})$ を次に定義される F の部分集合とする。

$$F(\mathbf{k}) = \{ \mathbf{c} | r_j(|\mathbf{K}|) = k_j, j=1, 2, \dots, L \}. \quad (3.9)$$

F は $F(\mathbf{k})$ によって類別されることは明らかである。すなわち次が成り立っている。

$$F = \bigcup_{0 \leq \mathbf{k} \leq \mathbf{K}} F(\mathbf{k}), \quad F(\mathbf{k}) \cap F(\mathbf{k}') = \emptyset \quad \forall \mathbf{k} \neq \mathbf{k}'. \quad (3.10)$$

次にベクトル \mathbf{k} に関して辞書順に一連番号を付ける。すなわち、ベクトル \mathbf{k} に次の規則に従い番号 $n(\mathbf{k})$ を付ける。

$$n(\mathbf{k}) = k_L + \sum_{i=1}^{L-1} k_i \prod_{j=i+1}^L (K_j + 1), \quad 0 \leq \mathbf{k} \leq \mathbf{K}. \quad (3.11)$$

この番号付けにより、どのクラスに関してもその客数の昇降順序を保ちながらベクトル \mathbf{k} が番号付けられる。すなわち、 $n(\mathbf{k}) < n(\mathbf{k} + \mathbf{e}_i), i=1, 2, \dots, L, \forall \mathbf{k}$ である。

今、状態集合 F を $F(\mathbf{k})$ で類別し、推移行列 A を次のようにブロック小行列 B_{ij} に分割する。すなわち、 $A = \{B_{ij}\}, i=0, 1, \dots, n(\mathbf{K}), j=0, 1, \dots, n(\mathbf{K})$ とし、 $\mathbf{c} \in F(\mathbf{k}), \mathbf{c}' \in F(\mathbf{k}')$ なる \mathbf{c}, \mathbf{c}' について、 $a(\mathbf{c}, \mathbf{c}') \in B_{n(\mathbf{k}), n(\mathbf{k}')}$ となるように小行列 B_{ij} を構成する。

このとき、 $\mathbf{c} \in F(\mathbf{k})$ ならば、 $\mathbf{c}_{(i)} \in F(\mathbf{k} - \mathbf{e}_i), \mathbf{c} + j\mathbf{e}_{|\mathbf{r}(\mathbf{K})|+1} \in F(\mathbf{k} + \mathbf{e}_j)$ であり、 $n(\mathbf{k}) > n(\mathbf{k} - \mathbf{e}_i), n(\mathbf{k}) < n(\mathbf{k} + \mathbf{e}_j)$ であるから (3.7) より $A = \{B_{ij}\}$ の構造は

次のごとくである。

Q_0 からの客の退去を表現する要素は小行列 B_{ij} , $i > j$ の中に現れ, Q への客の到着を表現する要素は小行列 B_{ij} , $i < j$ の中に現れ, 対角小行列 B_{ii} は対角行列となる。すなわち, Q_0 からの客の退去に関する性質は A の下三角行列として表現され, Q_0 への客の到着に関する性質は A の上三角行列として表現されるように推移行列 A を構成することができる。

このように構成された A について, 仮にすべての k について状態集合 $F(k)$ を状態 k に縮約することができると思えば, 小行列 B_{ij} は要素 b_{ij} に代えることができ, 推移行列 A は $A^* = \{b_{ij}\}$ に縮約することができる, 解くべき連立方程式の次元数を大幅に削減することが可能になる。3.3 節に示した例に則してみれば, $K=2$ のとき, 19 次元 \rightarrow 4 次元, $K=3$ の場合, 69 次元 \rightarrow 9 次元, $K=5$ の場合, 923 次元 \rightarrow 25 次元, $K=10$ の場合には 705431 次元 \rightarrow 100 次元の程度となる。

このような状態の縮約が可能となるための必要十分条件は次のごとくであることが知られている⁷⁾。すなわち, A を構成するすべての小行列 B_{ij} に関して, B_{ij} の各々の行和をとったときにその値がすべてある値 b_{ij} に等しくなる場合には状態空間 $\{0, 1, \dots, n(K)\}$ をもつプロセスもやはりマルコフプロセスとなり, その推移行列は $A^* = \{b_{ij}\}$ となる。

本稿のモデルに関してこの縮約可能 (lumpability) 条件を調べてみると次のことがわかる。 B_{uv} , $u < v$ なる小行列についてみると, $u = n(k)$, $v = n(k - e_j)$ としたとき $c \in F(k)$ なら, $r_j(|K|) = k_j$, $j = 1, 2, \dots, L$ であること, すなわち $r(|K|) = k \forall c \in F(k)$ が成り立つので, 任意の $c \in F(k)$ に関する小行列 B_{uv} の行和は (3.7) より $b_{uv} = r_j(K - k)$ なる一定値となり, 縮約可能条件を満足する。一方, B_{uv} , $u > v$ なる小行列についてみると, $u = n(k)$, $v = n(k - e_j)$ としたとき, $c \in F(k)$ から $c' \in F(k - e_j)$ への推移は $c_i = j$, $1 \leq i \leq m$ の場合に推移率 μ_j で発生することから, B_{uv} の c に対応する行に関する行和を $l_{uv}(c)$ とすると, $l_{uv}(c) = r_j(m)\mu_j$ となる。 $r_j(m)$ は一般に $c \in F(k)$ により異なる値をとるため, 行和 $l_{uv}(c)$ は一定値にはならず縮約可能条件は満たされていない。

以上より, 本稿のモデルに関しては, 推移行列 A の上三角行列を構成する小行列に関しては状態の縮約可能条件を満足しているが, 下三角行列を構成する小行列に関してはこの条件を満足しておらず, したがって全体として条件を満たさず状態集合 $F(k)$ を状態 k に

縮約することはできないことがわかる。

しかし, 厳密な意味では上に見たように $F(k)$ を k に縮約することはできないが, 何らかの方法で, 縮約可能条件の成り立たない下三角行列を構成する各小行列 B_{uv} について, その性質を近似的に表現するような要素 (推移率) b_{uv} を作り出すことができるなら, それを用いて近似的に縮約された推移行列 A^* を作り出すことができ近似解を得ることができる。

本稿では, この下三角行列を構成する各小行列を近似的に一つの推移率で表現をするための手段として次章に述べるランダム選択モデルを用いる。

4. ランダム選択モデルを用いた近似法

4.1 ランダム選択モデルの考え方

前章に見たように, Q_0 から Q_0 への客の推移に関する性質を表現する上三角行列の部分については $F(k) \rightarrow k$ なる状態の縮約可能条件が成り立つが, Q_0 から Q_0 への客の推移に関する性質を表現する下三角行列の部分に関してはこの条件が成り立たない。そこで, 本章では Q_0 から Q_0 への客の推移すなわち Q_0 からの客の退去過程に関する性質を状態 k に関する情報のみを利用して近似的に表現できるようなモデルを構成し, それを用いて近似解を得る方法について考察する。

Q_0 からの退去過程を近似的に表現するモデルとは, Q_0 の状態 k が与えられたとき, この状態からクラス j なる客が退去する退去率 $\mu_j(k)$ を与えるモデルである。

このモデルには, Q_0 のサーバ数が m であること, クラス j の客のサービス要求時間がパラメータ μ_j なる指数分布に従うこと等の性質が反映されていなければならない。しかし, 状態 k に関する情報しか使用することができないため, 元の状態集合 $F(k)$ に含まれていた客の並び方に関する情報はこのモデルには反映されないことになる。したがって, モデルを作る際には客のサービス順序に関する情報を補って考えなければならない。

本稿では, このサービス規律にランダム選択を仮定しモデルを構築する。以上の方針で構築されるモデルを本稿ではランダム選択モデルとよぶこととし, $Q_0^*(k)$ と表すことにする。

以上より, 本稿の方法は本来解きたいモデル (Q_0 , $Q_{(0)}$) に比べて, 次の二つの点で近似の概念が入ることになる。1) $Q_{(0)}$ を合成ノード Q_0 で置き換えるこ

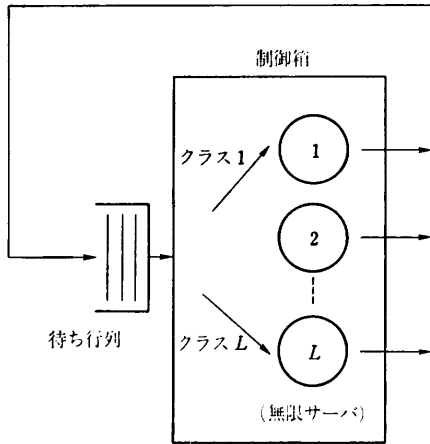


図 1 ランダム選択モデル
Fig. 1 Random choice model.

と. すなわち, $Q_{(0)}$ に入ってから出るまでの滞在時間分布を指数分布と仮定すること, 2) 2ノード網 (Q_0, Q_e) を計算する際に, 縮約可能条件を満足しない状態集合を縮約してしまうこと, である.

4.2 ランダム選択モデルの構成

ランダム選択モデル $Q_0^*(\mathbf{k})$ は, 客数ベクトル $\mathbf{k} = (k_1, k_2, \dots, k_L)$ をもつ L 個の閉鎖型部分連鎖の存在する閉鎖型待ち行列網であり, L 個の無限サーバノードを含む 1 個の制御箱および制御箱の前に作られる 1 本の待ち行列から構成される. 図 1 に構成を示す.

部分連鎖 j に属するクラス j の客は必ず j 番目の無限サーバノードでサービスを受けるものとし, サービス要求時間はパラメータ μ_j をもつ指数分布に従うものとする.

従って, ノード j に m_j 人の客が存在するときの退去率を $\nu_j(m_j)$ とすれば次である.

$$\nu_j(m_j) = m_j \mu_j, \quad j=1, 2, \dots, L. \quad (4.1)$$

ノード j でのサービスを終了した客は, ノード j を退去すると同時に制御箱からも退去し, 直ちに制御箱の前に形成される待ち行列に着くものとする.

制御箱は, L 個の無限サーバノードでサービス中の客数の総和を m (Q_0 のサーバ数) 以下に制限する目的をもっている.

すなわち, ノード j の滞在客数を m_j とし, $\mathbf{m} = (m_1, m_2, \dots, m_L)$ とすれば, 制御箱内は常に $|\mathbf{m}| \leq m$ となるよう制御される.

サービスの終了により箱内客数が減った場合には, 直ちに制御箱の前の待ち行列中の客が 1 人選ばれ箱内に入るものとする. 客の選択はランダムに行われるものとする. すなわち, 客の退去直前における箱内の状

態が \mathbf{m} であるとき, 退去直後にクラス j の客が選択される確率を $\nu_j(\mathbf{m})$ とすると次である. $m < |\mathbf{k}|$ とする.

$$\nu_j(\mathbf{m}) = (k_j - m_j) / |\mathbf{k} - \mathbf{m}|, \quad j=1, 2, \dots, L. \quad (4.2)$$

4.3 状態方程式

系の状態を \mathbf{m} とし, $Q_0^*(\mathbf{k})$ に関する状態方程式を導く.

$m < |\mathbf{k}|$ を仮定しておく. 実現可能な状態 \mathbf{m} の集合 M は次である.

$$M = \{\mathbf{m} \mid 0 \leq m_j, |\mathbf{m}| = m\}. \quad (4.3)$$

また, 状態 \mathbf{m} から, クラス j の客が退去しクラス i の客が入る事象の発生率 $\varphi_{i,j}(\mathbf{m})$ は次に与えられる.

$$\varphi_{i,j}(\mathbf{m}) = \nu_j(m_j) \nu_i(\mathbf{m}), \quad i, j=1, 2, \dots, L. \quad (4.4)$$

また, $\nu(\mathbf{m}) = \nu_1(m_1) + \nu_2(m_2) + \dots + \nu_L(m_L)$ とする.

系の状態が \mathbf{m} である平衡状態確率を $q(\mathbf{m})$ とすると平衡状態方程式は次に与えられる.

$$\nu(\mathbf{m}) q(\mathbf{m}) = \sum_{i=1}^L \sum_{j=1}^L \varphi_{i,j}(\mathbf{m} - \mathbf{e}_i + \mathbf{e}_j) q(\mathbf{m} - \mathbf{e}_i + \mathbf{e}_j), \quad \forall \mathbf{m} \in M. \quad (4.5)$$

4.4 近似法

$Q_0^*(\mathbf{k})$ におけるクラス j の客の退去率 $\mu_j(\mathbf{k})$ は $q(\mathbf{m})$ が与えられると次のごとくに求まる.

$$\mu_j(\mathbf{k}) = \sum_{\mathbf{m} \in M} \nu_j(m_j) q(\mathbf{m}), \quad j=1, 2, \dots, L \quad (4.6)$$

上式に得られる $\mu_j(\mathbf{k})$ を, 状態を $F(\mathbf{k})$ から \mathbf{k} に縮約した場合に, 縮約された推移行列 A^* の $n(\mathbf{k})$ 行 $n(\mathbf{k} - \mathbf{e}_j)$ 列の要素とみなし, $A^* = \{t(i, j)\}$ を次のごとくに構成する. ただし, $|\mathbf{k}| \leq m$ の場合には $\mu_j(\mathbf{k})$ は次に定義されるものとする.

$$\mu_j(\mathbf{k}) = \nu_j(k_j), \quad j=1, 2, \dots, L. \quad (4.7)$$

$0 \leq k_j \leq K, j=1, 2, \dots, L$ について $t(i, j)$ は次.

$$\left. \begin{aligned} t(n(\mathbf{k}), n(\mathbf{k} + \mathbf{e}_j)) &= \tau_j(\mathbf{K} - \mathbf{k}) \\ t(n(\mathbf{k}), n(\mathbf{k})) &= - \sum_{j=1}^L \{ \mu_j(\mathbf{k}) + \tau_j(\mathbf{K} - \mathbf{k}) \} \\ t(n(\mathbf{k}), n(\mathbf{k} - \mathbf{e}_j)) &= \mu_j(\mathbf{k}). \end{aligned} \right\} \quad (4.8)$$

また, \mathbf{z} を次に定義される確率ベクトルとする.

$$\mathbf{z} = (z_0, z_1, \dots, z_{n(\mathbf{k})}), \quad z_i = P(Q_0 \text{ の状態が } \mathbf{k}), \quad i = n(\mathbf{k}). \quad (4.9)$$

このとき, 本近似法では, 次の方程式の解として定まる確率ベクトル \mathbf{z}^* を求める確率ベクトル \mathbf{z} の近似値として採用する, という考え方をとっている.

$$\mathbf{z}^* A^* = \mathbf{0}, \quad |\mathbf{z}^*| = 1. \quad (4.10)$$

以上より近似解を得ることは A^* を定める, すなわ

ち状態方程式 (4.5) を解き (4.6) および (4.7) により $\mu_j(\mathbf{k})$ を定めることに帰着される。

4.5 状態方程式の解

状態方程式 (4.5) の一般解を得るのは困難である。本稿では容易に解ける $L=2$ の場合についての解を示す。

u および l を次とすると解は (4.12) 式のごとくなる。

$$u = \max(0, m - k_2), \quad l = \min(k_1, m) \quad (4.11)$$

$$q(m_1, m_2) = \{ \prod_{n=u}^{m_1-1} \varphi_{12}(n, m-u) / \prod_{n=u+1}^{m_1} \varphi_{21}(n, m-n) \} \times q(u, m-u), \quad u+1 \leq m_1 \leq l,$$

$$q(u, m-u) = \{ 1 + \sum_{i=u+1}^l q(u, m-i) \}^{-1} \quad (4.12)$$

以上により、 $L=2$ の場合については近似解を得ることができる。

4.6 計算手順

以上の近似解法を整理すると次のような計算手順にまとめることができる。

S1: 合成ノード Q_c の作成。

$0 \leq k \leq K$ なるすべての k について、たまたみこみ計算によって、網 $Q_{(0)}$ に関する正規化定数 $G_{(0)}(\mathbf{k})$ を計算する。 $j=1, 2, \dots, L$ について、 $\tau_j(\mathbf{K}-\mathbf{k}) = G_{(0)}(\mathbf{K}-\mathbf{k}-\mathbf{e}_j) / G_{(0)}(\mathbf{K}-\mathbf{k})$ を計算し、行列 A^* の $n(\mathbf{k})$ 行 $n(\mathbf{k}+\mathbf{e}_j)$ 列の位置にその値を格納する。

S2: ランダム選択モデル $Q_0^*(\mathbf{k})$ の作成。

$0 \leq k \leq K$ かつ $|k| \leq m$ なるすべての k および $j=1, 2, \dots, L$ について、 $\mu_j(\mathbf{k}) = \nu_j(k_j)$ を作成。さらに、 $0 \leq k \leq K$ かつ $|k| > m$ なるすべての k について、 $Q_0^*(\mathbf{k})$ に関する状態方程式を解き $q(\mathbf{m})$ を求めそれを用いて、 $j=1, 2, \dots, L$ について $\mu_j(\mathbf{k}) = \sum_{m \in M} \nu_j(m_j) q(\mathbf{m})$ により $\mu_j(\mathbf{k})$ を計算する。 $0 \leq k \leq K$ なるすべての k および $j=1, 2, \dots, L$ につい

て、以上に計算された $\mu_j(\mathbf{k})$ の値を行列 A^* の $n(\mathbf{k})$ 行 $n(\mathbf{k}-\mathbf{e}_j)$ 列の位置に格納する。さらに、行列 A^* の $n(\mathbf{k})$ 行 $n(\mathbf{k})$ 列である対角要素の位置に、 $-\sum_{j=1}^L \{ \mu_j(\mathbf{k}) + \tau_j(\mathbf{K}-\mathbf{k}) \}$ を格納する。

S3: 2ノード網 (Q_0, Q_c) の計算。

連立方程式 $\mathbf{z}^* A^* = \mathbf{0}, |\mathbf{z}^*| = 1$ を解き、 Q_c に関する平衡状態確率の近似解 \mathbf{z}^* を得る。

5. 数値実験例

本章では、4章に示した近似法の精度を調べるため数値実験について述べる。数値実験に用いた待ち行列網モデルを図2に示す。文献5)に提案されている近似法とも精度を比較するため、モデルおよびパラメータの値はすべて文献5)に示されるものに準じた。網は二つのノードから構成され、客数ベクトル $\mathbf{K} = (K_1, K_2)$ とする二つの閉鎖型部分連鎖をもつ閉鎖型待ち行列網とする。ノード1はサービス能力1 (人/単位時間)をもつシングルサーバノードとし、クラス1および2の客はそれぞれパラメータ μ_1, μ_2 をもつ指数分布に従ってサービス時間を要求するものとする。サービス規律は先着順とする。ノード2はプロセッサシェアリング (PS) ノードとし、クラス1およびクラス2の

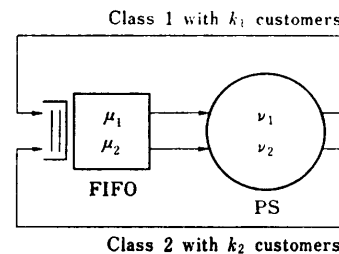


図2 数値実験用ネットワークモデル
Fig. 2 Network model for numerical examples.

表1 サイクルタイムの比較 (1)
 $\mathbf{K} = (1, 2), \nu_1 = \nu_2 = 1.0, \mu_1 = 2.0$

Table 1 Numerical comparisons between exact and approximate cycle time of customers.

μ_2	クラス1の客のサイクルタイム					クラス2の客のサイクルタイム				
	厳密解	ランダム選択近似法		積形式近似法		厳密解	ランダム選択近似法		積形式近似法	
		近似解	相対誤差	近似解	相対誤差		近似解	相対誤差	近似解	相対誤差
20.0	3.43	3.45	0.6%	3.47	1.2%	2.89	2.85	1.4%	2.81	2.8%
4.0	3.26	3.30	1.2	3.36	3.1	3.01	2.99	0.7	2.94	2.3
2.0	3.21	3.22	0.3	3.21	0.0	3.21	3.25	1.2	3.21	0.0
1.3	3.31	3.26	1.5	3.09	6.6	3.48	3.57	2.6	3.58	2.9
1.0	3.50	3.38	3.4	3.00	16.7	3.81	3.94	3.4	4.00	5.0
0.8	3.74	3.56	4.8	2.93	21.7	4.16	4.33	4.1	4.46	7.2

表 2 ノード使用率の比較
 $K=(1, 2), \nu_1=\nu_2=1.0, \mu_1=2.0$

Table 2 Numerical comparisons between exact and approximate node utilizations.

μ_2	FIFO ノード使用率					PS ノード使用率				
	厳密解	ランダム選択近似法		積形式近似法		厳密解	ランダム選択近似法		積形式近似法	
		近似解	相対誤差	近似解	相対誤差		近似解	相対誤差	近似解	相対誤差
20.0	.180	.175	2.8%	.180	0.0%	.984	.994	1.0%	.999	1.5%
4.0	.320	.305	4.7	.319	0.3	.972	.978	0.6	.979	0.7
2.0	.467	.434	7.1	.467	0.0	.933	.943	1.1	.933	0.0
1.3	.581	.530	8.8	.581	0.0	.876	.899	2.6	.882	0.7
1.0	.668	.602	9.9	.667	0.1	.811	.850	4.8	.833	2.7
0.8	.734	.656	10.6	.731	0.4	.747	.802	7.4	.790	5.8

表 3 サイクルタイムの比較 (2)
 $K=(1, 2), \nu_1=\nu_2=1.0$

Table 3 Numerical comparisons between exact and approximate cycle time of customers.

μ_1	μ_2	クラス1の客のサイクルタイム					クラス2の客のサイクルタイム				
		厳密解	ランダム選択近似法		積形式近似法		厳密解	ランダム選択近似法		積形式近似法	
			近似解	相対誤差	近似解	相対誤差		近似解	相対誤差	近似解	相対誤差
5.0	0.5	4.25	3.75	11.8%	2.06	51.5%	5.27	5.49	4.2%	5.76	9.3%
1.0	0.1	17.40	17.37	0.2	4.00	77.0	19.98	21.64	8.3	26.91	34.7
0.5	0.05	38.96	37.23	4.4	6.95	82.2	42.21	42.56	0.8	56.25	33.3

客はそれぞれパラメータ ν_1, ν_2 をもつ指数分布に従いサービス時間を要求するものとする。この網は積形式解をもたないが、平衡方程式を直接解くことにより厳密解を計算することができる。

表1から表3に、本近似法による近似解と厳密解の比較、および文献5) に示される近似法による近似解との比較を示す。いずれも、 $K=(1, 2)$ とした。比較項目には各クラスごとのサイクルタイムおよび各ノードの使用率をとりあげた。サイクルタイムとは客が二つのノードを一巡する時間とする。本稿で示した近似法をランダム選択近似法、文献5) に示される近似法を積形式近似法とよぶことにする。表1および表2にサイクルタイムおよびノード使用率の数値比較例を示す。

表に示される比較例では、 $\mu_1=2.0, \nu_1=\nu_2=1.0$ とし、 μ_2 を20から0.8まで変化させている。積形式近似法は $\mu_1=\mu_2$ のとき原理的に厳密解に一致するのでこの場合を例外とすると、サイクルタイムに関してはクラス1およびクラス2の客ともにランダム選択近似法の方が積形式近似法よりも良い精度を示している。

しかし、使用率に関してはこの関係は成り立たず、積形式近似法の方が一般に良い精度を示している。サイクルタイムに関する近似解と厳密解の比較を図3に示

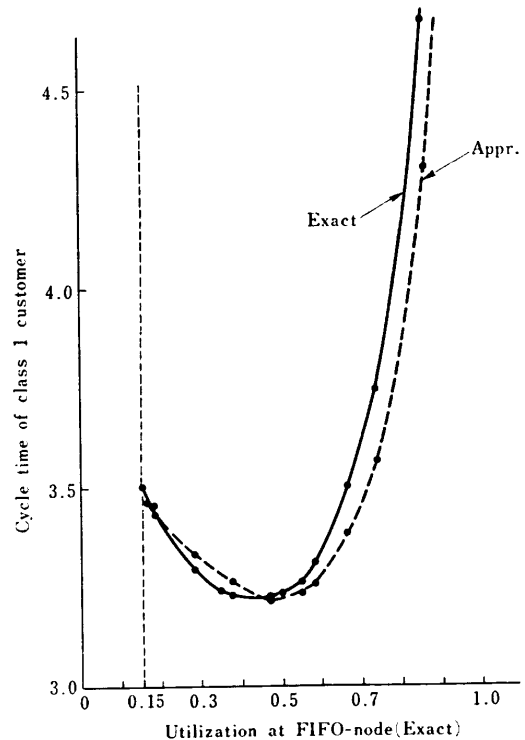


図 3 厳密解と近似解の比較、クラス1の客のサイクルタイム

Fig. 3 Numerical comparison for cycle time of class 1 customer.

す。横軸に先着順ノードの厳密解による使用率をとり、クラス1の客のサイクルタイムを示した。 $\mu_2 = \infty$ のときクラス1の客のサイクルタイムは3.5となることに注意。

さらに、サイクルタイムに関する近似精度の改善度を見るために、 μ_1 および μ_2 の値が大きく異なる場合についての比較例を表3に示す。積形式近似法に比べて著しく精度が改善されていることがわかる。

6. む す び

非積形式型 FIFO ノードをもつ待ち行列網を分解近似法を用いて解く際に現れる連立方程式の次元数を削減する方法について述べ、簡単な待ち行列網を例にとり本稿の方法による近似解と厳密解との比較を行った。

本稿に示される近似解法によると、文献5)に示される近似法に比べて計算メカニズムは複雑になるがサイクルタイムに関する精度は向上することがわかる。

部分連鎖数が3以上の網に関してランダム選択モデルの解を得ることは今後の課題として残されている。

さらにまた、部分連鎖数が3以上の網については上記の方法のほかに、部分連鎖を縮約して2連鎖網に帰着させる近似法も考えられるが詳細な検討は今後の課題として残されている。

謝辞 本研究の基礎となる待ち行列網理論に関して数々の有益なご指導をいただいた、防衛大学校川島武先生、筑波大学逆瀬川浩孝先生、工学院大学山崎源治先生に感謝いたします。また、本研究を開始する機会を与えていただき、途上数々の援助をいただいた当社

C&C システム研究所三上徹所長代理、同研究所応用システム研究部難波田愈部長に感謝いたします。

参 考 文 献

- 1) Baskett, F., Chandy, K. M., Muntz, R. R. and Palacios, F. G.: Open, Closed, and Mixed Networks of Queues with Different Class of Customers, *J. ACM*, Vol. 22, No. 2, pp. 248-260 (1975).
- 2) Chandy, K. M., Herzog, U. and Woo, L.: Parametric Analysis of Queueing Networks, *IBM J. Res. Dev.*, Vol. 19, No. 1, pp. 36-42 (1975).
- 3) 山本 彰, 西垣 通: FCFS スケジューリングを行うサーバを有する複数クラス待ち行列網モデルの近似解析, 情報処理学会論文誌, Vol. 25, No. 3, pp. 488-494 (1984).
- 4) 紀 一誠: 擬似ノード接続法による待ち行列網の近似解法, 情報処理学会論文誌, Vol. 26, No. 6, pp. 1129-1135 (1985).
- 5) 紀 一誠: クラスごとに異なるサービス要求率をもつ FIFO ノードを含む待ち行列網の近似解法, 情報処理学会論文誌, Vol. 26, No. 2, pp. 312-317 (1985).
- 6) Reiser, M. and Kobayashi, H.: Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms, *IBM J. Res. Dev.*, Vol. 19, No. 3, pp. 283-294 (1975).
- 7) Kemeny, J. G. and Snell, T. L.: *Finite Markov Chains*, D. Van Nostrand Company, New York (1960).

(昭和59年10月31日受付)

(昭和60年3月20日採録)