

擬似ノード接続法による待ち行列網の近似解法†

紀 一 誠††

積形式解をもたない待ち行列網に関する近似解法である擬似ノード接続法について示す。本近似法は演算量削減による分解近似法の改良をねらいとしている。積形式解の成立を妨げる非積形式型ノード Q_0 に代えて積形式型の擬似ノード Q_0 を網に接続する。 Q_0 はフィードバックループをもつ FIFO ノードで構成され、 Q_0 とスループット等価となるようにループ回数を調整することにより近似解を得る。この過程は非線型連立方程式を解くことにより実行されるが、解法に用いる Newton-Raphson 法の収束性の良さにより、従来の方法より演算量の削減がはかれる。本稿では、演算量の比較評価にふれ、さらに、 Q_0 が優先権制御ノードである場合を例にとり数値計算例を示す。

1. ま え が き

本稿では、積形式解¹⁾をもたない待ち行列網に関する近似解法である一般化された擬似ノード接続法について示す。本稿の方法は分解近似法 (Decomposition Approximate Method) を基礎とし、これに擬似ノード接続法を組み合わせることにより、合成ノードを構成する個々の積形式型ノードの諸量を計算する際に発生するたたみこみ演算の量を削減し、効率良く近似解を得ることをねらいとしている。

積形式解をもたない網に関する分解近似法は、Chandy et al.²⁾ により提案され、Parametric Analysis, Flow-Equivalent Method, Norton (または Thévenin) の定理、等の数多くの名称をもっており、訳語も定まっていなかったが本稿ではこの名称を用いることにする。分解近似法では、待ち行列網を積形式型のノード群と非積形式型のノード群に分解し、積形式ノード群をひとまとめとしこれとスループット等価となる1個の合成ノード Q_c (Composite Node) を構成する。 Q_c と非積形式ノード群から成る網を数値的に解くことにより近似解を得るものである。

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³⁾ になじみやす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顕在化してくる。1) 合成ノード Q_c と非積形式ノード群から成る網を解くために、大規模な連立方程式を解く必要が生ずる。2) 積形式型ノードに関する諸量を計算するために、たたみこみ計算の繰り返しによる膨大な演算量が発生する。

本稿に示す擬似ノード接続法は問題点 2) の解決を目的としたものである。

次章に本稿で扱うモデルおよび擬似ノード接続法の考え方を示す。3章に分解近似法による解法を述べ、4章に擬似ノード接続法を示す。5章に全体を通しての計算アルゴリズムのまとめを行い、6章に演算量の比較評価を行う。 Q_0 が優先権制御ノードである場合を例にとり、数値計算例を7章に示す。さらに、4章に示される最適擬似ノードに関する Conjecture についての数値実験結果も7章に示される。

2. モデルと擬似ノード接続法の考え方

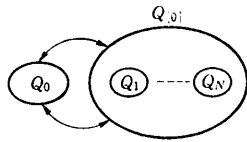
2.1 モデル

$N+1$ 個のノードから構成される待ち行列網を Q ノード番号 i のノードを Q_i とする。 Q_0 を非積形式型のノード、 Q_1, \dots, Q_N を積形式型のノードとする。網内には L 個の閉鎖型部分連鎖に従いノード間を推移する 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 の客とする。客数ベクトルを $\mathbf{K} = (K_1, K_2, \dots, K_L)$ とする。

以下、記述を簡略化するために非積形式ノード Q_0 を次のような条件をもつノードと考えることにする。 Q_0 はサーバ数 m の複数窓口とし、 Q_0 におけるクラス j の客のサービス要求時間はパラメータ μ_j なる指数分布に従うものとする。 $\mathbf{k} = (k_1, k_2, \dots, k_L)$ を状態ベクトル、 $0 \leq k \leq K$ とし、 Q_0 の状態が \mathbf{k} のとき、 Q_0 におけるクラス j の客に関するサービス率を $\mu_j(\mathbf{k})$ とする。例として Q_0 が、割込優先・中断点再開 (pre-emptive-resume) 方式の優先権制御ノードの場合を考える。若番クラスほど高い優先順位をもち、同一順位内は先着順制御とする。割込・中断は常に、存在する最下位順位の客に対して行われるものとする。このと

† Generalized Shadow Node Method for Non-Product-Form Queueing Networks by ISSEI KINO (C&C Systems Research Laboratories, NEC Corporation).

†† 日本電気(株)C&C システム研究所



Q_0 : non-product-form-node
 $Q_{[0]}$: a group of product-form-nodes

図1 非積形式解をもつ待ち行列網
 Fig. 1 Non-product-form-network model.

き, $\mu_j(\mathbf{k})$ は次のごとくに定義される. ただし, $[x]^+ = (x, 0 \leq x; 0, x < 0)$.

$$\mu_j(\mathbf{k}) = \begin{cases} \mu_1 \cdot \min(m, k_1), & j=1 \\ \mu_j \cdot \min([m - (k_1 + \dots + k_{j-1})]^+, k_j), & j=2, \dots, L. \end{cases} \quad (2.1)$$

クラス j の客の $Q_i (i=0, \dots, N)$ への相対訪問回数 を θ_{ij} とする. これらはノード間推移を表現する推移確率行列を係数とする連立方程式を解くことにより定数倍を除いて定めることができる. 本稿では, Q_0 を基準とする. すなわち, $\theta_{0j}=1, j=1, \dots, L$, とする. $\rho_{0j} = \theta_{0j} / \mu_j$ とし, Q_0 の負荷ベクトルを $\rho_0 = (\rho_{01}, \rho_{02}, \dots, \rho_{0L})$ とする. さらに, ベクトルに関する次の表記法を定義しておく. \mathbf{x}, \mathbf{a} を L 次元ベクトルとする.

$$|\mathbf{x}| = x_1 + x_2 + \dots + x_L, \quad \mathbf{a}^* = a_1^* a_2^* \dots a_L^*, \\ \mathbf{x}! = x_1! x_2! \dots x_L!$$

次に, 待ち行列網 Q から他のパラメータはそのままにして, Q_i を取り除いてできる待ち行列網を元の網 Q の i -complement といい, $Q_{[i]}$ と記すことにする. 本稿のモデルでは, 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になる. $Q_{[0]}$ に関する正規化定数を, $G_{[0]}(\mathbf{k}), 0 \leq \mathbf{k} \leq \mathbf{K}$ と記す. 図1に本稿で扱うネットワークモデルを示す.

2.2 擬似ノード接続法の考え方

積形式解の成立を妨げている Q_0 に代えて, Q_0 と似たようなふるまいをする積形式型のノードを構成して網に接続することにより, 全体を積形式解をもつ網におき換えこれを解いて近似解を得ようとするのが擬似ノード接続法の基本的な考え方である. 本稿では, Q_0 に代え網に接続する積形式型の擬似ノード Q_c とし, フィードバックループをもち, 状態依存型のサービス率をもつ先着順制御ノードを設定する. 解法過程は次のごとくとなる.

S1: $Q_{[0]}$ とスループット等価となる合成ノード Q_c を構成する.

S2: 2ノード網 (Q_0, Q_c) を数値的に解いて, Q_0 における平衡状態確率およびスループットを算出する.

S3: 擬似ノード Q_c を $Q_{[0]}$ に接続し, $N+1$ 個のノードから成る積形式解をもつ網, ($Q_c, Q_{[0]}$), を構成する.

S4: Q_c のフィードバックループ回数 (等価的に負荷ベクトル ρ_0) を調節することにより, 網 ($Q_c, Q_{[0]}$) における Q_c のスループットをS2で計算された Q_0 におけるスループットに等しくなるようにする.

以上の過程を終了した時点で得られる網 ($Q_c, Q_{[0]}$)* から算出される $Q_{[0]}$ に関する諸量を, 元の網 ($Q_0, Q_{[0]}$) から定まる $Q_{[0]}$ に関する諸量の近似値とするものである.

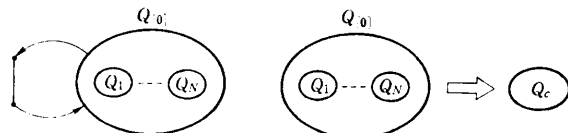
解法過程 S1, S2 は従来の分解近似法と全く同じである. 分解近似法ではこの後, $Q_i (\in Q_{[0]})$ に関する諸量を計算するために, まず状態 \mathbf{k} を条件とする条件付確率を計算し, 次にこれらをS2で得られる状態確率で重味付けをして算出する. 本稿の方法はこの部分をS3, S4で置き換えたものである. 解法過程S4は非線型の連立方程式を解くことにより実行されるが, 解法に用いる Newton-Raphson 法の2次収束性の良さのため, 演算量は従来の方法に比較して大きく削減できる.

3. 分解近似法

本章では, Q_0 に関する近似解を計算する S1, S2 の部分について述べ, さらに分解近似法を用いて $Q_{[0]}$ に関する近似解を構成するアルゴリズムを示す.

3.1 Q_0 に関する近似解

図2に示すように, 分解近似法ではまず, Q_0 に代えてサービス時間がゼロである仮想的な積形式ノードを $Q_{[0]}$ に接続することにより, Q_0 におけるクラス j の客に関するスループットの近似値を次のごとくに求める. ただし, \mathbf{e}_j は j 方向の単位ベクトル, $j=1, 2, \dots$,



(a) Network Q_0 shorted (b) Composite node

図2 分解近似法による合成ノードの構成
 Fig. 2 Reduction of the original network to the equivalent one.

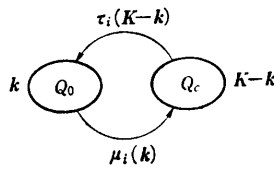


図3 2ノード網 (Q₀, Q_c)
Fig. 3 2-Nodes Network (Q₀, Q_c).

Lとする。

$$\tau_j(\mathbf{k}) = G_{[0]}(\mathbf{k} - \mathbf{e}_j) / G_{[0]}(\mathbf{k}), \quad 0 \leq \mathbf{k} \leq \mathbf{K}. \quad (3.1)$$

この過程は Q₀ をあたかもショートしたかのように見えるため、電気回路理論の Norton (または Thévenin) の定理になぞらえられている。次に、Q_[0] とスループット等価な合成ノード Q_c を構成する。Q_c は、その状態が \mathbf{k} であるときにクラス j の客に関するサービス率が (3.1) に示される状態依存型サービス率 $\tau_j(\mathbf{k})$ をもつノードとして構成される。図2参照。

以上により、網 Q は図3に示されるごとくに、2ノード網 (Q₀, Q_c) に縮約される。この待ち行列網 (Q₀, Q_c) は積形式解をもつ網とはならないが、高々2個のノードしか存在しないので、状態 \mathbf{k} , $0 \leq \mathbf{k} \leq \mathbf{K}$, に関する状態方程式を作りこれを数値的に解くことにより Q₀ に関する状態確率を計算することができる。例えば $\mu_j(\mathbf{k})$ が (2.1) の場合には、この部分は複合構造をもつブロック3重対角行列を係数とする連立方程式を解くことにより実行される。各クラスの客について、Q_[0] に入ってから出て来るまでの網内滞在時間を指数分布と仮定して (Q₀, Q_c) を解いている点が分解近似法が近似法である理由である。2ノード網 (Q₀, Q_c) を解いて得られる Q₀ に関する平衡状態確率を $p_0(\mathbf{k})$ とする。Q₀ におけるクラス j の客に関するスループット φ_j および、スループットベクトル φ は次式に与えられる。

$$\varphi_j = \sum_{0 \leq \mathbf{k} \leq \mathbf{K}} \mu_j(\mathbf{k}) p_0(\mathbf{k}), \quad j=1, 2, \dots, L. \quad (3.2)$$

$$\varphi = (\varphi_1, \varphi_2, \dots, \varphi_L). \quad (3.3)$$

3.2 Q_[0] に関する近似解

Q_i (∈ Q_[0]) におけるクラス j の客に関する待ち行列長 (サービス中を含む) $L_{ij}(\mathbf{K})$ を例にとり、分解近似法を用いた場合の近似解の計算法を示す。

$M_{ij}(\mathbf{k})$ を、Q_[0] における網内容客ベクトルが \mathbf{k} であるという条件のもとでの Q_i におけるクラス j の客の待ち行列長とする。Q_c の状態が \mathbf{k} であるとき Q₀ の状態は $\mathbf{K} - \mathbf{k}$ であることに注意して、 $L_{ij}(\mathbf{K})$ は $M_{ij}(\mathbf{k})$ の重味付けとして次式により得られる。

$$L_{ij}(\mathbf{K}) = \sum_{0 \leq \mathbf{k} \leq \mathbf{K}} M_{ij}(\mathbf{k}) p_0(\mathbf{K} - \mathbf{k}). \quad (3.4)$$

上式に基づいて Q_[0] に関する近似解の計算手順をアルゴリズムFとし、プログラム風に表現すると以下のごとくなる。ただし、 $q_i(\mathbf{x})$ は Q_i の状態が \mathbf{x} である正規化されない状態確率、 $G_{[0, i]}$ を 0 and i -complement な網に関する正規化定数とする。

アルゴリズムF

```

For i=1 to N
  F2: compute G[0, i]( $\mathbf{k}$ ), for 0 ≤  $\mathbf{k}$  ≤  $\mathbf{K}$ 
  For j=1 to L
    F4: For  $\mathbf{k}=0$  to  $\mathbf{K}$ 
      M( $\mathbf{k}$ ) ← ∑0 ≤  $\mathbf{x}$  ≤  $\mathbf{k}$  xj qi( $\mathbf{x}$ )
      G[0, i]( $\mathbf{k} - \mathbf{x}$ ) / G[0, i]( $\mathbf{k}$ )
    Next  $\mathbf{k}$ 
    F5: Lij( $\mathbf{K}$ ) ← ∑0 ≤  $\mathbf{k}$  ≤  $\mathbf{K}$  M( $\mathbf{k}$ ) p0( $\mathbf{K} - \mathbf{k}$ )
  Next j
Next i
    
```

4. 擬似ノード接続法

4.1 擬似ノード Q_c の構成法

Q_[0] に関する近似解を得るため、Q₀ の動きを模倣する積形式型の擬似ノード Q_c を構成する。Q₀ に代え Q_c を網 Q_[0] に接続することにより、積形式解をもつ網 (Q_c, Q_[0]) を作り、これを計算することにより近似解を得る。

Q_c は各クラスごとにフィードバックループをもち、状態依存型のサービス率をもつ先着順制御ノードとする。すなわち、Q_c におけるサービス要求時間は各クラスとも共通のパラメータ μ_c をもつ指数分布に従うものとし、サービス規律は全クラスを通じての先着順制御、ノード内容数が k 人のときのサービス率を $\mu_c(k)$ とする。さらに、クラス j の客は Q_c でのサービス終了後確率 r_j で網 Q_[0] に進み、 $1 - r_j$ なる確率で再び Q_c に戻るものとする。このとき、フィードバックループ回数を含めたクラス j の客の Q_c における平均訪問回数 θ_{ij} は、 $\theta_{0j} = 1$ に注意して、 $\theta_{ij} = 1 / (1 - r_j)$ となる。また、クラス j の客が Q_c に到着してから離れるまでの間に受ける延べサービス時間の平均値を ρ_{ij} とすれば次の関係がある。

$$\rho_{ij} = \theta_{ij} / \mu_c = 1 / \mu_c (1 - r_j), \quad j=1, 2, \dots, L. \quad (4.1)$$

Q_c は Q₀ とスループット等価となるように構成されるものとする。すなわち、Q_c におけるスループットが (3.3) に等しくなるように構成されるものとする。

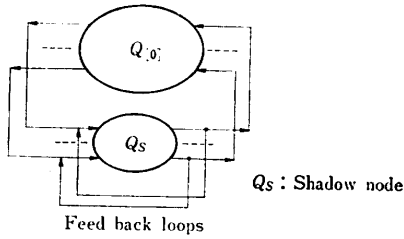


図4 フィードバックループをもつ先着順制御ノードによる擬似ノード \$Q_s\$ の構成

Fig. 4 Shadow node \$Q_s\$ with feed back loops.

このとき、調節されるパラメータはループ回数あるいは \$r_j\$ であるが、(4.1)によりこれらは \$\rho_{ij}\$ に変換できるので、以降はすべて \$\rho_{ij}\$ を変化させ \$Q_s\$ を \$Q_0\$ とスループット等価にするものとする。図4にフィードバックループをもつ \$Q_s\$ の概要を示す。

\$Q_s\$ の負荷ベクトルを \$\rho_s = (\rho_{s1}, \rho_{s2}, \dots, \rho_{sL})\$ とし、\$Q_s(k)\$ を次に定義される \$Q_s\$ の容量係数とする。

$$a_s(k) = \left\{ \prod_{j=1}^k \mu_s(j) \right\}^{-1} \quad 1 \leq k, \quad a_s(0) = 1. \quad (4.2)$$

このとき、\$Q_s\$ の状態が \$k\$ である正規化されない状態確率 \$q_s(k)\$ は次のように表現することができる。

$$q_s(k) = a_s(|k|) \frac{|k|!}{k!} \rho_s^k. \quad (4.3)$$

4.2 最適擬似ノード

\$Q_0\$ におけるスループット(3.3)と等価なスループットを実現するという規範のみを利用して \$Q_s\$ を一意に定めることはできない。\$Q_s\$ を一意に定めるためにはさらに最適性に関する条件を付加して考えなければならない。本稿ではこの最適条件として、\$Q_0\$ における滞在時間に最も近い滞在時間を実現する \$Q_s\$ を最適ノードとする、という条件を採用することにする。すなわち、\$(Q_0, Q_s)\$ を解いて得られる \$Q_0\$ における各クラスの客の滞在時間ベクトルを \$\mathbf{S}\$、\$Q_0\$ とスループット等価となるよう調整された \$Q_s\$ と \$Q_{(0)}\$ からなる網 \$(Q_s, Q_{(0)})\$ から計算される各クラスの客の \$Q_s\$ における滞在時間ベクトルを \$\tilde{\mathbf{S}}\$ としたとき、\$\|\mathbf{S} - \tilde{\mathbf{S}}\|\$ を最適な \$Q_s\$ ノードとする。ただし、\$\|\mathbf{x}\| = |\mathbf{x}|\$ とする。

最適擬似ノードを \$Q_s^*\$ とすると次の Conjecture が成り立つ。

Conjecture: \$Q_s\$ を次のサービス率をもつノードとする。

$$\mu_s(k) = \{k\mu_s, \quad k < M; \quad M\mu_s, \quad M \leq k\}. \quad (4.4)$$

このとき、\$Q_s\$ は \$M=m\$ の場合に \$Q_s^*\$ となる。

以上。

上記の予想は7章に示す数値実験例によって確認することができる。

4.3 スループット方程式

\$Q_s\$ は積形式型のノードとして設定されているので、待ち行列網 \$(Q_s, Q_{(0)})\$ は積形式解をもつ網として解くことができる。この網に関する正規化定数 \$G(\mathbf{k})\$ は次式に与えられる。

$$G(\mathbf{k}) = \sum_{0 \leq \mathbf{x} \leq \mathbf{k}} q_s(\mathbf{x}) G_{(0)}(\mathbf{k} - \mathbf{x}), \quad 0 \leq \mathbf{k} \leq \mathbf{K}. \quad (4.5)$$

\$Q_s\$ におけるクラス \$j\$ の客に関するスループットを \$\nu_j\$、スループットベクトルを、\$\boldsymbol{\nu} = (\nu_1, \nu_2, \dots, \nu_L)\$、とする。

\$\nu_j\$ は正規化定数を用いて次のように得られる。

$$\nu_j = \theta_{ij} G(\mathbf{K} - \mathbf{e}_j) / G(\mathbf{K}), \quad j = 1, 2, \dots, L. \quad (4.6)$$

上式右辺は \$\rho_s\$ の関数であるから、スループットベクトルは \$\rho_s\$ の関数となっている。これを \$\boldsymbol{\nu}(\rho_s)\$ と表す。

\$Q_0\$ とスループット等価な \$Q_s\$ を構成するということは \$Q_0\$ のスループット \$\boldsymbol{\varphi}\$ が与えられたとき、次のスループット方程式を解き、解ベクトル \$\rho_s\$ を定めることにほかならない。

$$\boldsymbol{\varphi} = \boldsymbol{\nu}(\rho_s). \quad (4.7)$$

上式は \$\rho_s\$ を変数とする非線型連立方程式を表している。

4.4 スループット方程式の解法

方程式(4.7)の解法を示す。基本的には、Newton-Raphson 法を用いて解くことができるが、解の存在領域として、\$0 \leq \rho_s\$ なる制約条件が付いており、この制約条件を取り除くための若干の工夫が必要とされる。

以下簡略化のため \$\rho_s\$ の添字 \$s\$ を省略することにする。

\$n\$ 回目の繰り返しの際に得られる解を \$\rho^{(n)}\$ とすれば、Newton-Raphson 法による計算スキームは次のようになる。

$$\rho^{(n+1)} = \rho^{(n)} - \{ \boldsymbol{\nu}(\rho^{(n)}) - \boldsymbol{\varphi} \} [\partial \boldsymbol{\nu} / \partial \boldsymbol{\rho}]_{\rho = \rho^{(n)}}^{-1}. \quad (4.8)$$

ここで、\$[\partial \boldsymbol{\nu} / \partial \boldsymbol{\rho}]\$ は次に定義される行列とする。

$$\left[\frac{\partial \boldsymbol{\nu}}{\partial \boldsymbol{\rho}} \right] = \begin{bmatrix} \frac{\partial \nu_1}{\partial \rho_1} & \frac{\partial \nu_2}{\partial \rho_1} & \dots & \frac{\partial \nu_L}{\partial \rho_1} \\ \dots & \dots & \dots & \dots \\ \frac{\partial \nu_1}{\partial \rho_L} & \frac{\partial \nu_2}{\partial \rho_L} & \dots & \frac{\partial \nu_L}{\partial \rho_L} \end{bmatrix}. \quad (4.9)$$

制約条件 \$0 \leq \rho\$ を外すために \$\boldsymbol{\nu}(\rho)\$ を、\$-\infty < \rho < +\infty\$、で定義される関数 \$\tilde{\boldsymbol{\nu}}(\rho)\$ に次のように拡張する。

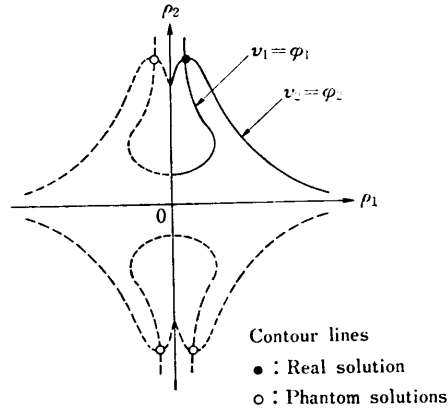


図 5 真の解とファントム解
Fig. 5 Real solution and phantom solutions.

$$\tilde{v}(\rho_1, \rho_2, \dots, \rho_L) = v(|\rho_1|, |\rho_2|, \dots, \rho_L). \quad (4.10)$$

拡張された関数 \tilde{v} を用いて (4.8) に示される計算スキームを適用してスループット方程式を解く。この結果得られる解は必ずしも制約条件を満足する解とは限らない。制約条件を満足しない解をファントム解とよぶことにする。もし、方程式 (4.7) の解が $0 \leq \rho$ なる範囲に一意に存在し、かつ \tilde{v} に関する (4.8) が収束するならば、(4.10) より明らかな関数の対称性から、ファントム解の符号のみを正とすることによって容易に真の解を得ることができる。図 5 に $L=2$ の場合の真の解とファントム解との関係を示す例を示す。図 5 には、 $v_1 = \phi_1$, $v_2 = \phi_2$, となる等高線とそれを (4.10) により全平面に拡張したものが示される。等高線の交点のうち、第 1 象限に現れるもののみが真の解となり、その他はファントム解となる。

計算スキーム (4.8) を実行するためには (4.9) に示される行列の要素を計算可能な形に導いておく必要がある。

以下これを導いておく。(4.2) から (4.6) より次

$$\frac{\partial v_i}{\partial \rho_j} = \frac{\theta_{ji}}{G(\mathbf{K})^2} \left\{ G(\mathbf{K}) \frac{\partial}{\partial \rho_j} G(\mathbf{K} - \mathbf{e}_i) - G(\mathbf{K} - \mathbf{e}_j) \frac{\partial}{\partial \rho_j} G(\mathbf{K}) \right\}. \quad (4.11)$$

また、 $\partial G(\mathbf{K}) / \partial \rho_j$ は次のごとくになる。

$$\begin{aligned} \frac{\partial}{\partial \rho_j} G(\mathbf{K}) &= \sum_{0 \leq k \leq K} \frac{\partial}{\partial \rho_j} a_i(|\mathbf{k}|) \frac{|\mathbf{k}|!}{\mathbf{k}!} \rho^{\mathbf{k}} G_{[0]}(\mathbf{K} - \mathbf{k}) \\ &= \sum_{0 \leq k \leq K} k_j q_i(\mathbf{k}) G_{[0]}(\mathbf{K} - \mathbf{k}) / \rho_j. \end{aligned} \quad (4.12)$$

以上をまとめて次が得られる。

$$\frac{\partial v_i}{\partial \rho_j} = \left\{ \theta_{ji} \sum_{0 \leq k \leq K} k_j q_i(\mathbf{k}) G_{[0]}(\mathbf{K} - \mathbf{e}_i - \mathbf{k}) \right.$$

$$\left. - v_i \sum_{0 \leq k \leq K} k_j q_i(\mathbf{k}) G_{[0]}(\mathbf{K} - \mathbf{k}) \right\} / \rho_j G(\mathbf{K}). \quad (4.13)$$

4.5 擬似ノード構成アルゴリズム

本章の解法をアルゴリズム S として以下にまとめておく。ただし、 $\Delta \rho^{(n+1)} = \rho^{(n+1)} - \rho^{(n)}$ とする。

アルゴリズム S

- S1: 初期設定, $\rho \leftarrow \rho^{(0)}$
- S2: 正規化定数 $G(\mathbf{k})$, $0 \leq k \leq K$ を計算
- S3: スループットベクトル v の作成
 $v_j \leftarrow \theta_{ij} G(\mathbf{K} - \mathbf{e}_j) / G(\mathbf{K})$, $j=1, 2, \dots, L$.
- S4: 行列の作成, $[\partial v / \partial \rho] \leftarrow \{\partial v_i / \partial \rho_j\}$,
 $i, j=1, 2, \dots, L$.
- S5: 連立方程式を解き, $\rho^{(n+1)}$ を求める。
 $v - \phi = -\Delta \rho^{(n+1)} [\partial v / \partial \rho]$.
- S6: 収束判定, $\max_j |v_j - \phi_j| / \phi_j < \epsilon$.

満足したら終了, しない場合は S2 にもどる。

5. 擬似ノード接続法の計算手順

本章には、3 章および 4 章を通して、擬似ノード接続法全体の計算手順のまとめを行う。以下の手順のうち、A1~A4 が分解近似法、A5~A7 が擬似ノード接続法に対応する部分となっている。

- A1: 初期設定, $G_{[0]}(\mathbf{k})$, $0 \leq k \leq K$ を作成。
- A2: 合成ノード Q_0 および網 (Q_0, Q_c) の構成。
 $\tau_j(\mathbf{K} - \mathbf{k})$ の計算, および $\mu_j(\mathbf{k})$, $\tau_j(\mathbf{K} - \mathbf{k})$,
 $0 \leq k \leq K$, より係数行列 A を作成。
- A3: 網 (Q_0, Q_c) の計算, $\mathbf{x}A = \mathbf{0}$, $|\mathbf{x}| = 1$ を解く。
- A4: Q_0 に関する諸量の計算, ϕ, S はか。
- A5: 最適擬似ノード Q^* の構成。 $M=m$ とし, 方程式 $\phi = v(\rho)$ をアルゴリズム S により解く。
- A6: 網 $(Q_1, Q_{[0]})$ に関する正規化定数 $G(\mathbf{k})$, $0 \leq k \leq K$ の計算。
- A7: Q_i , $Q_i \in Q_{[0]}$ に関する諸量の計算。

6. 演算量の比較評価

本章では、分解近似法によるアルゴリズム F と、擬似ノード接続法による A5~A7 の部分の演算量の評価を行う。A1~A4 部分に関しては共通手順なので除外する。

本例においては、いずれも乗算回数と加算回数はほぼ等しいので、これを演算回数とよび評価指標とする。

アルゴリズム F の演算回数を I_F とすれば、3.2 節

表 1 演算量の比較評価

$L=2, N=5$ の場合, $n^*: I_F=I_S$ となるループ回数,
 n : 実測ループ回数, $E=n/n^*$: 演算量の削減比
 Table 1 Iteration loop count n^* that satisfies $I_F=I_S$, observed iteration loop count in numerical experiments and the ratio $E=n/n^*$.

客数ベクトル K	n^* (回)	n (回)	$E=n/n^*$
$K=(3, 3)$	9.6	5	52%
$K=(5, 5)$	16.9	6	36
$K=(8, 8)$	35.2	8	23
$K=(10, 10)$	50.9	10	20

に示した手続きを吟味することにより, 次に与えられる.

$$I_F = NR \left[1 + L \left\{ \prod_{j=1}^L (K_j + 2) / 2^L + 1 \right\} \right]. \quad (6.1)$$

ただし, R は $0 \leq k \leq K$ なる定義域上で行われるたみこみ演算に要する演算回数で, $R \equiv \prod_{j=1}^L (K_j + 1)$ である.

一方, 同じ待ち行列長の計算を A5~A7 の手順で行う場合の演算量 I_s は次のごとくに推定される. A6 に計算される正規化定数はアルゴリズム S の終了時点で自動的に得られていることに注意すると, I_s はアルゴリズム S に必要な演算回数と A7 に必要な回数とに分けられる. 手順を吟味することにより, 前者はおよそ, $R(L^2 + L + 1) + 2L + L^3/2$ 回, 後者は, $NR(L + 1)$ 回となる. ただし連立方程式は Gauss-Jordan 法を使用するとした. アルゴリズム S における収束までの繰り返し回数を n 回とすると, 全体の演算回数は次となる.

$$I_s = n \{ R(L^2 + L + 1) + 2L + L^3/2 \} + NR(L + 1). \quad (6.2)$$

ループ回数 n の大小により効率の良さが左右される. $L=2, N=5$ の場合を例にとり, 両者の演算回数が等しくなる, すなわち $I_F=I_S$ となるようなループ回数 n^* を表 1 に示す. また, $\epsilon=0.001$ として実際に計算を実行した場合に観測された収束までのループ回数 n も合わせて表 1 に示す. 表 1 に見られるように, $n < n^*$ であり, 網内容数の増加につれ n^* が増加するのに対して収束ループ回数はあまり増加せず, 本稿の方法は網の規模が大きくなるほど相対的に効率的な算法となっていることがうかがえる.

7. 数値実験例

7.1 優先権制御ノードを含む網

Q_0 が 2.1 節の例に示したような割込優先・中断点

再開方式の優先権制御を行うノードである場合⁴⁾ を例にとり, 擬似ノード接続法を用いて解いた数値計算例を示す.

図 6 に数値計算に用いたモデルを示す. $L=2, N=5$ とし, Q_1, \dots, Q_N はすべてシングルサーバで先着順サービス規律とし, サービス時間はすべて, $\mu_i=0.5$ なるパラメータをもつ指数分布に従うものとする. 相対訪問回数 θ_{ij} および負荷 ρ_{ij} は表 2 のごとくに設定される.

Q_0 はシングルサーバ, すなわち $m=1$ とする. 表 3 に, $K=(3, 3), \rho_0=(4.0, 1.0)$ とした場合の, 各クラスごとの客によるノードの使用率および各ノードでの滞在時間についての数値計算例を示す. さらに, ρ_0 をそのままにして, Q_0 が先着順制御をすると仮定して計算された値も表 3 に付記されている. ノード滞在時間の合計値が, 客が網内を一巡するサイクルタイムの平均値となる.

7.2 最適擬似ノード

4.2 節に示した Conjecture を確認するための数値実験例を表 4 に示す. モデルおよびパラメータは前節に準ずるものとする. ただし, 本例では $\rho_0=(2.5,$

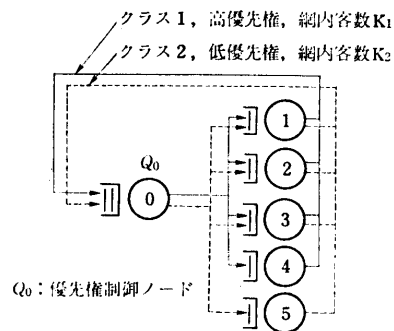


図 6 数値計算例のための非積形式解型待ち行列網モデル

Fig. 6 Network model for numerical examples.

表 2 数値実験例に用いたパラメータ

Table 2 Parameters for numerical examples.

Node No. (i)	θ_{ij}		$\rho_{ij} = \theta_{ij} / \mu_i$	
	1	2	1	2
0	1	1	—	—
1	2	3	1.0	1.5
2	3	2	1.5	1.0
3	4	4	2.0	2.0
4	3	0	1.5	0
5	0	3	0	1.5

$\mu_i=0.5 (i=1, 2, \dots, 5)$

表 3 数値計算例

$K=(3, 3), \rho_0=(4.0, 1.0)$

Table 3 Numerical examples with $K=(3, 3)$ and $\rho_0=(4.0, 1.0)$.

ノード番号	ノード使用率 (%)			ノード滞在時間	
	クラス1	クラス2	計	クラス1	クラス2
0	70.2 (58.1)	11.9 (22.2)	82.1 (80.4)	7.69 (9.53)	15.37 (2.88)
1	17.5	17.9	35.4	1.47	2.22
2	26.3	11.9	38.2	2.19	1.58
3	35.1	23.8	58.9	3.90	4.21
4	26.3	0	26.3	1.83	—
5	0	17.9	17.9	—	1.73
計	175.4	83.4	258.8 (300.7)	17.08 (20.62)	25.13 (13.47)

注: () 内は Q_0 で先着順制御を行う場合.

表 4 最適擬似ノードに関する Conjecture の検証

Table 4 Numerical examples testing the Conjecture for an optimal shadow node.

Q_0	$m=1$		$m=3$	
	$E_{01}=3.9$	$E_{02}=13.4$	$E_{01}=2.50$	$E_{02}=2.64$
Q_0	E_{11}	E_{12}	E_{21}	E_{22}
$M=1$	4.0	13.9	2.27	2.42
$M=3$	4.3	14.4	2.49	2.65
$M=\infty$	4.4	14.7	2.53	2.69

2.5) とし, Q_0 のサーバ数 m が $m=1$ および $m=3$ の場合について, 擬似ノード Q_0 のサーバ数 M を, $M=1, M=3, M=\infty$ と変化させた場合のノード滞在時間を比較した. E_{11}, E_{12} は擬似ノード Q_0 におけるクラス1, 2の客の滞在時間, E_{01}, E_{02} は網 (Q_0, Q_c) を解いて得られるクラス1およびクラス2の客の Q_0 における滞在時間とする. E_{01}, E_{02} に最も近い E_{11}, E_{12} を実現する M をもつ擬似ノードが最適擬似ノード Q^* となる. 表4に見られるように, 本例ではいずれの場合も $M=m$ の場合に最適擬似ノードが実現されており, 4.2節の Conjecture の成立を確認することができる.

本例以外にもいくつかの数値実験を試みたが, いずれの場合も成立を確認することができた. Q_0 と同数のサーバ数を擬似ノード Q_0 にもたせるという構造的な類似性が客のノード滞在時間の近似を良くしていると推定できるが厳密な証明は困難なようである.

8. むすび

積形式解をもたない待ち行列網に関する近似解法で

ある擬似ノード接続法について述べた. 本稿の方法は分解近似法を実用規模の網に適用する際に発生する問題点の一つを解決することをねらいとしたものであった.

分解近似法を実用化する際に残されるもう一つの問題点, 大規模連立方程式への対応, に関しては今後の課題として残される.

積形式解をもつ待ち行列網は, コンピュータシステムおよび通信システムの有効な性能評価モデルとして広く実用に供されているが, 実用上要請されるモデルは必ずしも積形式解をもつ待ち行列網の範囲に閉じているわけではなく, 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は欠かせない.

本稿では非積形式ノードが1個のみ存在する網を扱ったが, これ複数個の非積形式ノードが存在する網にも適用できるように拡張すること, また本稿で扱った閉鎖型部分連鎖に加えて, 開放型部分連鎖の加わった混合型待ち行列網へとその応用範囲を拡張すること等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, J.: Open, Closed, and Mixed Networks 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) 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).
- 4) 紀 一誠: 優先権付き待ち行列網の近似解法, 待ち行列理論とその周辺, 京都大学数理解析研究所講究録, Vol. 519, pp. 137-155 (1984).

(昭和59年9月3日受付)

(昭和60年6月20日採録)