数理解析研究所講究録 1457 巻 2005 年 54-61

閉鎖型待ち行列ネットワークに対する高速近似解法

北海道大学・経済学研究科　木村 俊一 (Toshikazu Kimura)
Graduate School of Economics and Business Administration
Hokkaido University
北海道大学・経済学部　森岡 和行 (Kazuyuki Morioka)
Faculty of Economics, Hokkaido University

概 要

本論文では，閉鎖型待ち行列ネットワークに対する指数化法の改良を提案する．まず，従来の指数化法について説明し，サービス時間分布を Coxian 分布で近似することの問題点を指摘する．次に，拡散近似を適用した $M(n)/G/s/n$ 待ち行列に対する新しい近似式を紹介し，この近似式を指数化法に適用することで指数化法の計算負荷を軽減する方法を提案する．最後に，数値例を通して本論文で提案する改良指数化法が実用上問題のない近似精度をもつことを示す．

1 序論

閉鎖型待ち行列ネットワークは情報通信システムや生産システム等の性能評価において欠かせないモデルである．待ち行列ネットワークに関する研究の歴史は古く，積形式解を持つ様々なネットワークが Jackson [2] や Baskett *et.al.* [1] などによって研究された．しかし，積形式解を持つネットワークは厳しい仮定に依存しているため，現実問題に応用する際には，積形式解を持たないような一般的な待ち行列ネットワークに対する近似解法が必要になる．

閉鎖型待ち行列ネットワークに対する近似解法の中で有用なものに指数化法がある．指数化法は，元のネットワークをそれと等価な積形式解を持つ指数ネットワークで近似する手法である．指数化法は，Marie [3] によって提案され，Yao & Buzacott [6] によって拡張された．指数化法は，閉鎖型待ち行列ネットワークに対する近似手法の中で最も精度のよいものとして知られている．しかし，Marie [3]，Yao & Buzacott [6] の方法では，サービス時間分布に Coxian 分布を使用しているため実装が困難であると同時に，計算負荷が大きくなるという欠点があった．Stewart & Marie [5], Willits & Dietz [4] は，Coxian 分布を使った計算アルゴリズムの改良を行った．しかし，依然としてその計算量はノード数や客数が大きくなるに従って膨大なものとなる．本論文では，指数化法に拡散近似を適用することで指数法の改良を行う．

2 閉鎖型待ち行列ネットワークに対する指数化法

本論文では，ノード数 M，客数 N の閉鎖型待ち行列ネットワークを考える．ノード i には s_i (≥ 1) 個の並列サーバと容量 r_i (≥ 0) のバッファがあるものとする．また，ノード i におけるサービス時間は独立で同一の一般分布に従う．F_i をその累積分布関数とし，μ_i^{-1} を平均，c_i を変動係数とする．また，客は到着順 (FIFO) にサービスを受けるものとする．ノード i からノード j への推移確率を p_{ij} とし，推移確率行列 $\boldsymbol{P} = (p_{ij})$ は固定されている

ものとする．さらに，ノードiに存在する客数をk_iとし，状態確率を$p_i(n) \equiv P(k_i = n)$で表す．ノードiに存在することのできる最大客数を$n_i \equiv \min(s_i + r_i, N)$と表す．

ここでは，一般的な指数化法のアルゴリズムをまとめる．指数化法では，元のネットワークにおいて各ノードを到着率が$\lambda_i(n)$の$M(n)/C_k/s_i/n_i$待ち行列として考え，その定常状態確率を求めることで元のネットワークの状態確率の近似値とする．まず，到着率$\lambda_i(n)$の適切な値を求めるために元のネットワークと等価な指数ネットワークを考え，その指数ネットワークにおけるサービス率を元のネットワークのサービス率を使って初期化する．

$$\mu_i(n) := \min(n, s_i)\mu_i$$

指数ネットワークに積形式解が存在すれば，サービス率が与えられるとその状態確率$p_i^*(n)$を計算することができる．次に，以下の関係式を用いて到着率$\lambda_i(n)$の値を計算する．

$$\lambda_i(n) := \begin{cases} 0, & n = n_i \\ \dfrac{p_i^*(n+1)}{p_i^*(n)}\mu_i(n+1), & 0 \le n \le n_i - 1 \end{cases} \tag{1}$$

次のステップでは，この$\lambda_i(n)$の値を用いて，$M(n)/C_k/s_i/n_i$待ち行列の定常状態確率$\pi_i(n)$を計算する．また，次の関係式を用いてサービス率$\nu_i(n)$を求める．

$$\nu_i(n) := \begin{cases} 0, & n = 0 \\ \dfrac{\pi_i(n-1)}{\pi_i(n)}\lambda_i(n-1), & 1 \le n \le n_i \end{cases} \tag{2}$$

ここで，収束条件$\max_{i,n} |\mu_i(n) - \nu_i(n)| < \varepsilon$を満たすならば，$\pi_i(n)$を近似値とし，満たさないならば更新されたサービス率の値を用いて指数ネットワークを解きなおし，以下同じことを繰り返すという手順をとる．(1)，(2)式は，状態が均衡している様子を表している．指数化法のアルゴリズムは次のようになる．

指数化法のアルゴリズム

Step 0 サービス率を初期化する．$\mu_i(n) := \min(n, s_i)\mu_i$.

Step 1 指数ネットワークを解き$p_i^*(n)$を求め，(1)式より$\lambda_i(n)$を計算する．

Step 2 $M(n)/C_k/s_i/n_i$待ち行列を解き$\pi_i(n)$を求め，(2)式より$\nu_i(n)$を計算する．

Step 3 収束条件$\max_{i,n} |\mu_i(n) - \nu_i(n)| < \varepsilon$の判定．
満たすならば，$p_i(n) := \pi_i(n)$として終了する．
満たさないならば，$\mu_i(n) := \nu_i(n)$としてStep 1へ戻る．

指数化法は，指数ネットワークが解析解を持つことを利用した非常に単純な近似手法であるが，閉鎖型待ち行列ネットワークに対する近似解法の中で最も精度のよいものとして知られている．Marie [3]，Yao & Buzacott [6]は，Step 2において，サービス時間分布にCoxian分布を仮定して$\pi_i(n)$の値を計算した．Coxian分布を使って状態確率を求めるためには大規模な数値演算を実行する必要があり，このことが指数化法の計算負荷を大きくする原因となっていた．その後，Stewart & Marie [5], Willits & Dietz [4]らによってアルゴリズムの改良が行われたが依然としてCoxian分布を使用しているため計算負荷は大きいままである．本論文では，Coxian分布を使わずに，次節で取り上げる拡散近似によって状態確率$\pi_i(n)$を求めることで指数化法の計算負荷を軽減する手法を提案する．

3 改良指数化法

本節では，拡散近似を用いて $M(n)/G/s/n$ 待ち行列における状態確率 π_k を求める．まず，基本復帰境界を持つ $[0, N]$ 上の拡散過程 X を考える．基本復帰境界を持つ拡散過程においては，一度境界に到達するとランダムな時間その境界上にとどまり，その後，境界から内部へジャンプし，拡散運動を再開する．

このような基本復帰境界を持つ拡散過程において，$M(n)/G/s/n$ 待ち行列の定常状態確率に対する近似式を求めると次のようになる．

$$\pi_i(n) = \begin{cases} \pi_i(0)\dfrac{\alpha_i}{b_i(1)}\xi_i(n), & 1 \le n \le n_i - 1 \\ \pi_i(0)\dfrac{\alpha_i}{\beta_i}\dfrac{b_i(n_i)}{b_i(1)}\dfrac{\xi_i(n_i)}{\gamma_i(n_i)-1}, & n = n_i \end{cases} \tag{1}$$

ここで，$i = 1, \ldots, M$ ， $n = 1, \ldots, n_i$ について，

$$\begin{aligned} a_i(n) &= \lambda_i(n-1) + \min(n, s_i)\mu_i d_i^2(n) \\ b_i(n) &= \lambda_i(n-1) - \min(n, s_i)\mu_i \\ d_i^2(n) &= 1 + \mathbf{1}_{\{n \ge s_i\}}(n)(1 - p_i^*(0))(c_i^2 - 1) \\ \gamma_i(n) &= \exp\left\{\frac{2b_i(n)}{a_i(n)}\right\} \\ \xi_i(n) &= (\gamma_i(1) - 1)\prod_{k=2}^{n}\gamma_i(k) \\ \alpha_i &= \lambda_i(0) \quad , \quad \beta_i = s_i\mu_i \end{aligned}$$

とおいた．また，等価サービス率は，(1) 式を (2) 式に代入することで，

$$\nu_i(n) = \begin{cases} 0, & n = 0 \\ \dfrac{b_i(1)}{\gamma_i(1)-1}, & n = 1 \\ \dfrac{\lambda_i(n-1)}{\gamma_i(n)}, & 2 \le n \le n_i - 1 \\ \dfrac{\beta_i}{b_i(n_i)}\dfrac{\gamma_i(n_i)-1}{\gamma_i(n_i)}\lambda_i(n_i-1), & n = n_i \end{cases} \tag{2}$$

となる．(1), (2) 式を指数化法の Step 2 に適用することで，指数化法の計算負荷を軽減できる．改良指数化法のアルゴリズムをまとめると次のようになる．

改良指数化法のアルゴリズム

Step 0 サービス率を初期化する．$\mu_i(n) := \min(n, s_i)\mu_i$.

Step 1 指数ネットワークを解き $p_i^*(n)$ を求め，(1) 式より $\lambda_i(n)$ を計算する．

Step 2 (1) 式より $\pi_i(n)$，(2) 式より $\nu_i(n)$ を計算する．

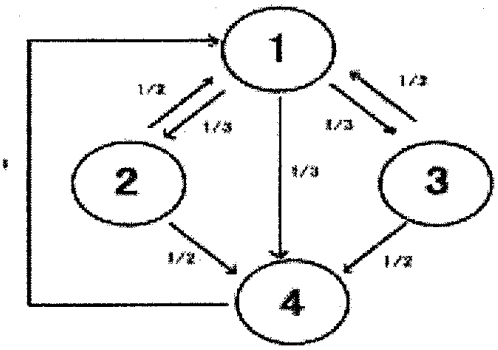

図 1: 単一サーバ・ネットワークの例

ノード	1	2	3	4
サービス時間分布	H_2	E_1	E_2	H_2
平均 (μ_i^{-1})	1	2	1	2
scv (c_i^2)	1.3	1	0.5	1.5
客数 (N)	8			
精度 (ϵ)	10^{-4}			
推移確率行列 $(\boldsymbol{P})$	0	1/3	1/3	1/3
	1/2	0	0	1/2
	1/2	0	0	1/2
	1	0	0	0

表 1: 例 1 のパラメータ設定

Step 3 収束条件 $\max_{i,n} |\mu_i(n) - \nu_i(n)| < \varepsilon$ の判定.
満たすならば, $p_i(n) := \pi_i(n)$ として終了する.
満たさないならば, $\mu_i(n) := \nu_i(n)$ として Step 1 へ戻る.

改良指数化法では, $\pi_i(n)$ を解析的に求めているため, Coxian 分布を扱う場合に比べて, 計算量を大幅に減らすことができる. また, サービス時間分布が Coxian タイプに制限されないため, 従来の手法よりも一般的である. 次節では, いくつかの数値例を通して, 本論文で提案する改良指数化法の精度を確かめる.

4 数値例

まず, 例 1 として図 1 で表されるような, ノード数 4, 客数 8 の単一サーバの例を考える. 各ノードにおけるサービス時間分布は, ノード 1 が平均 1, 変動係数 $\sqrt{1.3}$ の 2 次の超指数分布, ノード 2 が平均 2, 変動係数 1 の指数分布, ノード 3 が平均 1, 変動係数 $\sqrt{0.5}$ のアーラン分布, ノード 4 が平均 2, 変動係数 $\sqrt{1.5}$ の 2 次の超指数分布である. 通常, 生産システムへの応用を考える場合, 変動係数の 2 乗の値は 1 よりも小さく, ほぼゼロに近い. また, 情報通信システムの場合でも 1 前後である. したがって, 例 1 における変動係数の値は実用上妥当なものであるといえる. 例 1 のパラメータ設定を表 1 にまとめた.

実験では, 本論文で提案した拡散近似による改良指数化法と, シミュレーション・モデル, Cox-2 モデルによる指数化法を比較した. 改良指数化法, Cox-2 モデルともに 4 回で収束した. 図 2 から図 5 は状態確率を表している. 改良指数化法, Cox-2 モデルともに非常に精度の良いことが分かる. 表 2 は, 性能評価指標の比較である. 利用率の場合, 改良指数化法では最大で -1.5%, 従来の指数化法では最大 -1.1%程度の誤差であった. いずれのノードにおいても過小評価されていることが分かる. 平均客数の場合には, 改良指数化法で最大 0.5%, 従来の指数化法で最大 -7.5%の誤差がでた. Cox-2 モデルでは, 平均客数について -2.5%, -7.5%の誤差がでるなどあまり安定していないことが読み取れる. この数値例から, 変動係数が妥当な範囲にあるときには, 本論文で提案した拡散近似がうまく機能していることが分かる.

拡散近似では, 変動係数の値が大きくなると精度が落ちるという性質がある. そこで,

例2として変動係数の値が極端に大きい場合について考察する．例1と同じネットワークにおいて，ノード4におけるサービス時間分布を，平均2，変動係数10の2次の超指数分布とする．図6，図7は状態確率を表している．特にノード4をみると本論文による手法の誤差がさらに大きくなっているのに対し，Cox-2モデルでは誤差は小さいままであることが分かる．これは，超指数分布がCoxian分布によってうまく近似できていることを表している．このことは逆に，Coxモデルが特定の分布しか近似できないことを示唆している．本論文で提案した改良指数化法は，一般の分布に適用できる点においても従来の手法の拡張であるといえる．表3は，性能評価指標の比較である．利用率の場合について本論文の手法が−20%程の誤差がでているのに対し，Cox-2モデルでは，2%以内にとどまっている．また平均客数については，本論文の手法で最大−24%の誤差がでているのに対し，Cox-2モデルではノード2を除くと，2%内の誤差に収まっている．例2の数値例によって，変動係数が極端に大きい場合には本論文の手法は有効でないことが分かった．しかし，変動係数が10になるようなことは，実用上考えられない．

5 結論

本論文では，閉鎖型待ち行列ネットワークに対する近似手法である指数化法の改良を行った．指数化法は，指数ネットワークが解析解を持つことを利用した単純な近似手法であるが，閉鎖型待ち行列ネットワークに対する近似手法のなかで最も精度のよいものの一つとして知られている．しかし，従来の指数化法ではCoxian分布を仮定して状態確率を求めているために計算負荷が大きくなるという欠点があった．さらに，指数化法の適用範囲がサービス時間分布がフェーズタイプの場合に限られることを述べた．そこで本論文では，拡散近似による$M(n)/G/s/n$待ち行列に対する状態確率の近似式を紹介した．さらに，この近似式を指数化法に適用することで従来の手法の欠点であった計算負荷を軽減する手法を提案した．

数値実験によって，サービス時間分布の変動係数が小さい場合には，非常に精度が良いことが確かめられた．しかし，変動係数が大きくなるに従い精度が悪くなるという欠点も明らかとなった．これは，状態確率の導出に拡散近似を用いていることが原因と考えられる．したがって，変動係数の影響をいかに小さくできるかが改良指数化法の課題の一つといえる．しかし，生産システムへの応用を考える際には，変動係数の値は1よりも小さく，ほぼゼロであることが通常である．また，情報通信システムへの応用においても2を超えることはまれである．したがって，本論文の数値例は，改良指数化法が応用に耐え得るだけの精度を保っていることを示している．また改良指数化法は，Yao & Buzacottの拡張指数化法と組み合わせることで，有限容量の場合や推移確率行列が状態に依存する場合にも有効に適用できる．

本論文におけるモデルの仮定には重大な問題点もある．つまり，本論文で扱ったモデルにおいては，客をすべて同じ特徴を持つものとして扱った点である．応用上，すべての客が同じサービスを要求することは稀である．そこで今後の課題としては，客を特徴ごとにクラスという概念で区別する複数クラスへの拡張が考えられる．また，ネットワーク内に複数のサブネットワークが存在する複数連鎖への拡張も考慮するべきである．複数クラス・複数連鎖になると起こりうる状態の数が組み合わせ的に増加するため，計算負荷がよ

り小さくなるよう，さらなるアルゴリズムの改良が課題となる．

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 classes of customers," *Journal of the Assiciaton for Computing Machinary*, **22** (1975) 248–260.

[2] Jackson, J.R., "Networks of waiting lines," *Operations Research*, **5** (1957) 518–521.

[3] Marie, R.A., "An approximate analytical method for general queueing networks," *IEEE Transactions on Software Engineering*, **5** (1979) 530–538.

[4] Willits, C.J. and Dietz, D.C., "Rapid, efficient analysis of the $\lambda(n)/C_k/r/N$ queue, with apprication to decomposition of closed queueing networks," *Computers and Operations Research*, **25** (1998) 543–556.

[5] William, J.S. and Marie, R.A., "A numerical solution for the $\lambda(n)/C_k/r/N$ Queue," *European Journal of Operational Research*, **5** (1980) 56–68.

[6] Yao, D.D. and Buzacott, J.A., "The exponentialization approach to flexible manufacturing system models with general processing times," *European Journal of Operational Research*, **24** (1986) 410–416.

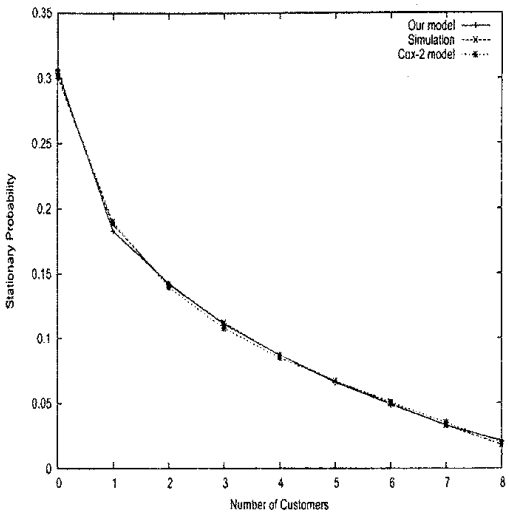

図 2: ノード 1 の状態確率 (例 1)

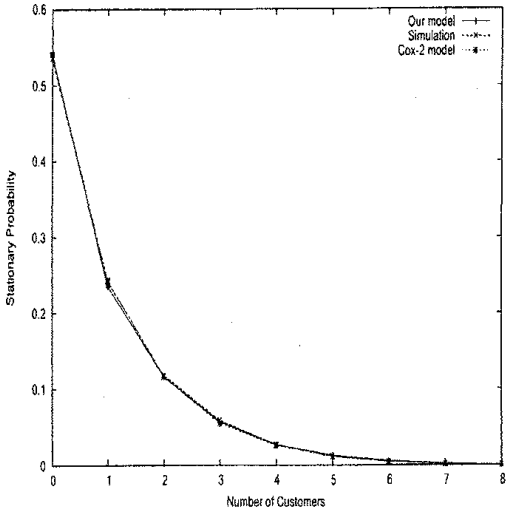

図 3: ノード 2 の状態確率 (例 1)

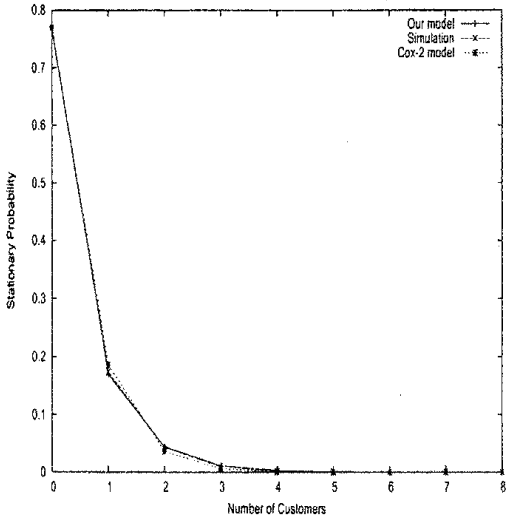

図 4: ノード 3 の状態確率 (例 1)

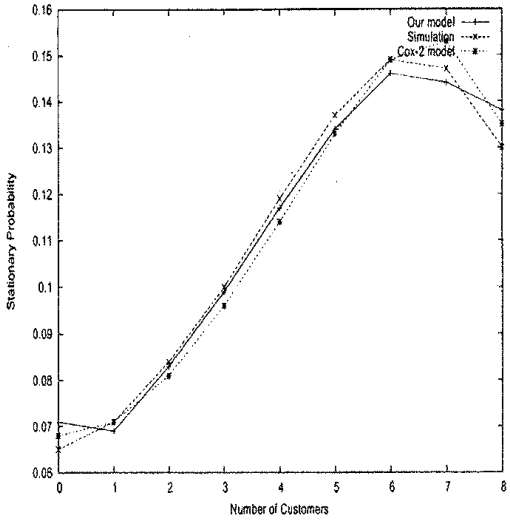

図 5: ノード 4 の状態確率 (例 1)

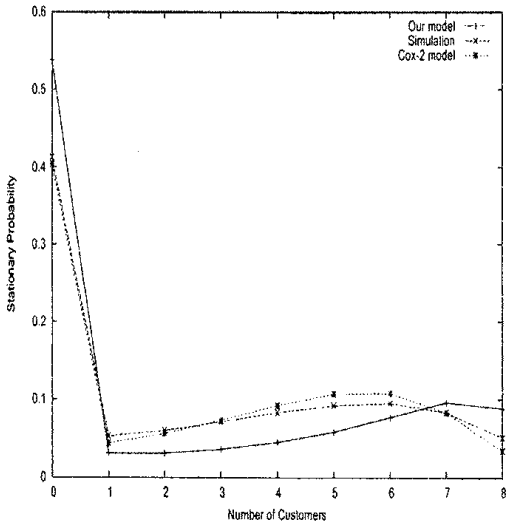

図 6: ノード 1 の状態確率 (例 2)

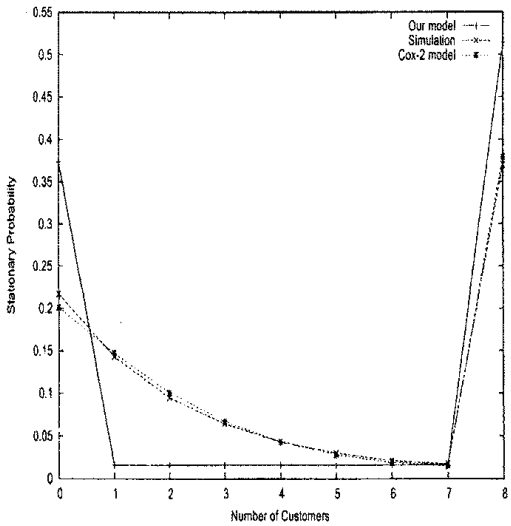

図 7: ノード 4 の状態確率 (例 2)

表 2: 性能評価指標の比較 (例 1)

ノード		1	2	3	4
u_i	(a)	0.693 (–0.7)	0.457 (–1.5)	0.228 (–0.9)	0.929 (–0.6)
	(b)	0.698 (–)	0.464 (–)	0.230 (–)	0.935 (–)
	(c)	0.696 (–0.3)	0.459 (–1.1)	0.230 (0)	0.932 (–0.3)
$\bar{n}_i$	(a)	2.172 (0.2)	0.867 (–0.5)	0.306 (0.3)	4.655 (–0.0)
	(b)	2.168 (–)	0.871 (–)	0.305 (–)	4.657 (–)
	(c)	2.179 (0.5)	0.849 (–2.5)	0.282 (–7.5)	4.689 (0.7)

表 3: 性能評価指標の比較 (例 2)

ノード		1	2	3	4
u_i	(a)	0.462 (–21.3)	0.298 (–23.6)	0.150 (–22.7)	0.626 (–20.1)
	(b)	0.587 (–)	0.390 (–)	0.194 (–)	0.783 (–)
	(c)	0.596 (1.5)	0.394 (1.0)	0.197 (1.5)	0.799 (2.0)
$\bar{n}_i$	(a)	2.510 (–8.2)	0.684 (–24)	0.242 (–22.4)	4.563 (12.6)
	(b)	2.734 (–)	0.900 (–)	0.312 (–)	4.054 (–)
	(c)	2.781 (1.7)	0.789 (–12.3)	0.316 (1.3)	4.114 (1.5)

u_i:利用率;
$\bar{n}_i$:平均滞在客数;
(a) 改良指数化法;
(b) シミュレーション;
(c) 指数化法 (Cox-2 モデル);

注) カッコ内の数値は，シミュレーションとの相対誤差 (%) である.