6B－1

効率の良いA／D法

田中 公幸　塩山 忠義

京都工芸繊維大学

1．はじめに

通信システムやFMS等の自動生産システムを性能評価する際に、これらのシステムを待ち行列ネットワークとしてモデル化する方法がある。これらのシステムが大規模になると、システムの性能を厳密に解析することが困難となるため、効果的な近似解法が必要となる。システムの平衡状態確率を求める近似解法の一つとして、aggregation-disaggregation法(A/D法)[1]があるが、この場合、遷移確率行列（t.p.m.）の形により収束するまでに要する反復回数が左右される。このため、本研究では、A/D法の実行に先立ち、システムのt.p.m.を出来るだけlumpableな形にすることにより、A/D法の反復回数を可能な限り減少させる手法を提案する。

2．A／D法

有限状態空間を$\Omega=\{1,2,\cdots,n\}$とし、$n\gg 1$とする。状態iの平衡状態確率をπ_i, $i\in\Omega$, とし、遷移確率行列（t.p.m.）を$P=[P_{ij}]$, $P_{ij}\geqq 0$, $\Sigma_j P_{ij}=1$, とした時、π_iは、

$$\pi_i=\sum_{j\in\Omega}\pi_j P_{ji} \qquad (1)$$

という式を満足する。しかし、これを直接解くことはシステムが大規模な場合、困難となるため近似解法が必要となり、A/D法は以下のようにして近似解を得るアルゴリズムである。マルコフ連鎖は既約であり非周期的であるとする。状態は次のようなN個のブロック(N≪n)に分割されるものとする。

$$\Omega=\sum_{\alpha=1}^{N}\Omega(\alpha) \qquad (2)$$

（Ⅰ）初期化：初期推定値$\pi(0)>0$を入力する。($\pi\equiv(\pi_1,\pi_2,\cdots,\pi_n)$)。$m=0$とする。

（Ⅱ）Aggregation step：mと$\pi(m)$を入力する。mを1増す。次に、$\bar{\pi}(m)\equiv(\bar{\pi}(m)_1,\bar{\pi}(m)_2,\cdots,\bar{\pi}(m)_N)$, $\bar{\pi}(m)_\alpha=\sum_{i\in\Omega(\alpha)}\pi(m)_i$, $\alpha=1,\cdots,N$, を次式より求める。

$$\bar{\pi}(m)_\alpha=\sum_{\beta=1}^{N}\bar{\pi}_\beta\bar{P}(\pi(m-1))_{\beta\alpha} \qquad (3)$$

ここで、任意の$x=(x_1,x_2,\cdots,x_n)$に対して、

$$\bar{P}(x)_{\beta\alpha}\equiv\sum_{j\in\Omega(\beta)}\sum_{i\in\Omega(\alpha)}x_j P_{ji}\Big/\sum_{k\in\Omega(\beta)}x_k \qquad (4)$$

とする。

（Ⅲ）Disaggregation step：各$\Omega(\alpha)$に対して、$\pi(m)_i$, $i\in\Omega(\alpha)$, を次式から求める。

$$\pi(m)_i=\sum_{j\in\Omega(\alpha)}\pi(m)_j P_{ji}+\sum_{\substack{\beta=1\\ \beta\neq\alpha}}^{N}\bar{\pi}(m)_\beta P(\pi(m-1))^{\#}_{\beta i} \qquad (5)$$

ここで、任意の$x=(x_1,x_2,\cdots,x_n)$に対して、

$$P(x)^{\#}_{\beta i}\equiv\sum_{j\in\Omega(\beta)}x_j P_{ji}\Big/\sum_{k\in\Omega(\beta)}x_k \qquad (6)$$

とする。

（Ⅳ）終了テスト：もし、残差$\pi(m)-\pi(m)P$が十分に小さければ終了。さもなければ、ステップ(Ⅱ)へ戻る。

このように、状態空間を幾つかのブロックに分けて考えることにより一度に解く方程式の数を減らし、計算機のメモリと計算時間を減らす方法である。

3．本手法

もし、全てのαとβの組み合せで$\Sigma_i P_{ij}$, $i\in\Omega(\alpha)$, が任意の$j\in\Omega(\beta)$に関して一定ならば、t.p.m.$[P_{ij}]$は式(2)の分割に関してlumpableであるという。lumpableなt.p.m.に対してA/D法のアルゴリズムは式(1)の解πを1回の反復で得る。これより、t.p.m.がlumpableなものに近くなる程、A/D法の反復回数が減ることが予想される。そこで、本研究では、できるだけt.p.m.をlumpableな形にする状態の分割を見つけるため次式の目標関数を定義する。

$$F\equiv\sum_{\alpha=1}^{N}\sum_{\beta=1}^{N}\Big\{\sum_{j\in\Omega(\beta)}\Big|\sum_{i\in\Omega(\alpha)}P_{ij}-\frac{1}{|\Omega(\beta)|}\sum_{j\in\Omega(\beta)}\sum_{i\in\Omega(\alpha)}P_{ij}\Big|\Big\} \qquad (7)$$

このFを最小にする分割(2)を分枝限定法を用いて求め、その分割に対応するlumpableに最も近いt.p.m.を得てA/D法を実行し(1)式の近似解を求める。また、分枝限定法において行なわれる下界値テストに用いる下界値として、第1～γ番目までのブロックの状態の選び方が決定している場合、次式で表わされるものを用いる。

$$\begin{aligned}LB(\gamma)=&\sum_{\alpha=1}^{\gamma}\sum_{\beta=1}^{\gamma}\Big\{\sum_{j\in\Omega(\beta)}\Big|\sum_{i\in\Omega(\alpha)}P_{ij}-\frac{1}{|\Omega(\beta)|}\sum_{j\in\Omega(\beta)}\sum_{i\in\Omega(\alpha)}P_{ij}\Big|\Big\}\\&+\sum_{\beta=1}^{\gamma}\Big\{\sum_{j\in\Omega(\beta)}\Big|\sum_{i\in\bar{\Omega}(\gamma)}P_{ij}-\frac{1}{|\Omega(\beta)|}\sum_{j\in\Omega(\beta)}\sum_{i\in\bar{\Omega}(\gamma)}P_{ij}\Big|\Big\}\\&+\sum_{\alpha=1}^{\gamma}\sum_{\beta=\gamma+1}^{N}\Big\{\sum_{k=k1}^{k2}\Big|\max_{j\in\bar{\Omega}(\gamma)}(k)\sum_{i\in\Omega(\alpha)}P_{ij}\\&\qquad-(1/M)\sum_{k=k1}^{k2}\max_{j\in\bar{\Omega}(\gamma)}(k)\sum_{i\in\Omega(\alpha)}P_{ij}\Big|\Big\}\\&+\sum_{\beta=\gamma+1}^{N}\Big\{\sum_{k=k1}^{k2}\Big|\max_{j\in\bar{\Omega}(\gamma)}(k)\sum_{i\in\bar{\Omega}(\gamma)}P_{ij}\\&\qquad-(1/M)\sum_{k=k1}^{k2}\max_{j\in\bar{\Omega}(\gamma)}(k)\sum_{i\in\bar{\Omega}(\gamma)}P_{ij}\Big|\Big\}\end{aligned} \qquad (8)$$

An Efficient A/D Algorithm
Kimiyuki TANAKA, Tadayoshi SHIOYAMA
Kyoto Institute of Technology

ここで、式の中に用いた各記号は以下のことを示す。

$$\overline{\Omega}(\gamma) \equiv \Omega - \sum_{\beta=1}^{\gamma} \Omega(\beta)$$

$M \equiv n/N$ （n：状態数，N：ブロック数）

$k1=(\beta-\gamma-1)M+1$

$k2=(\beta-\gamma)M$

$|\Omega(\beta)|$：$\Omega(\beta)$の位数

$\max_j(k)f(j)$：f(j)のk番目に大きいもの

本研究で用いた分枝限定法のアルゴリズムを以下に示す。ただし、用いる各記号は次に定義される。

S：分枝対象ノードの候補の集合

f：暫定値（初期暫定値をf_0とする）

N_i^ℓ:第ℓ番目のブロックの状態の選び方（ノードと呼ぶ）のｉ番目

N_0^ℓ:第ℓ番目のブロックの分枝対象ノード

$L(N_i^\ell)$:ノードN_i^ℓの下界値

[Step 1]（初期化） $f=f_0, \ell=0, S=\{N_0^0\}$ （N_0^0は原問題）とおく。

[Step 2]（分枝操作） N_0^ℓを分枝し、$\ell=\ell+1$として、新しいノードN_i^ℓ, i=1, ・・・, $\binom{\overline{\Omega}(\ell-1)}{m}$,を生成する。$N_0^{\ell-1}$をSから削除する。

[Step 3]（下界値テスト） 生成された全ての新しいノードN_i^ℓの下界値$L(N_i^\ell)$を(8)式から求める。もし、ℓが最終ブロックの値であれば、N_i^ℓの中で一番小さい下界値を持つノードを探し、ｆとの比較を行ない、ｆよりも小さい値であれば、暫定値ｆの更新を行ない、Ｓから新しいｆ以上の下界値のノードをＳから削除する。ℓが最終ブロックの値でない時は、ｆよりも小さいノードN_i^ℓのみをＳに保存する。

[Step 4] 最大のℓをもつ集合Ｓのノードのうちで、最小の下界値をもつノードを次の分枝対象ノードN_0^ℓとする。もし、集合Ｓが空であれば終了。そうでなければ、[Step 2]に戻る。

４．数値実験モデル

数値実験には、Courtoisのモデル[2]でサーバ数は５、各サーバのバッファは２、システム内の全客数は２、各サーバ間のrouting probabilityは表１の値，各サーバの処理率は表２の値を用いた。この時、システムの状態数は１５となり、これを５状態ずつの３個のブロックに分ける場合について考えた。A/D法の終了条件は1.0×10^{-9}として数値実験を行なった。

表１　routing probability

i \ j	0	1	2	3	4
0	0.500	0.475	0.010	0.010	0.005
1	0.455	0.500	0.020	0.020	0.005
2	0.040	0.040	0.620	0.100	0.200
3	0.140	0.140	0.060	0.630	0.030
4	0.100	0.100	0.100	0.100	0.600

表２　各サーバの処理率

	0	1	2	3	4
μ_i	0.70	0.60	0.20	0.10	0.20

５．実験結果

本手法によるＦの値は最小で0.764となり最大で6.527となり、この時のA/D法の終了までに要した反復回数は、Ｆが最小となる分割の時には14回であり、最大のＦの時には72回であった。この両者の収束の様子を図示したものが図１である。

次に、種々のＦの値でのA/D法の反復回数を求め散布図で示したものが図２である。これより、Ｆの値が小さい程、反復回数は減少し、Ｆの値と反復回数の相関係数を求めると0.854であった。

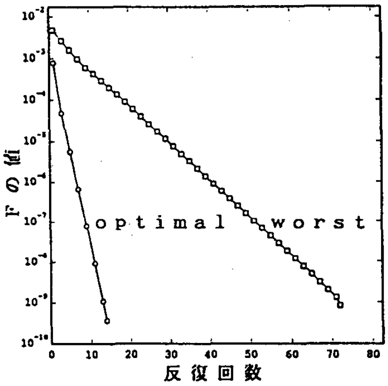

図１　収束の様子

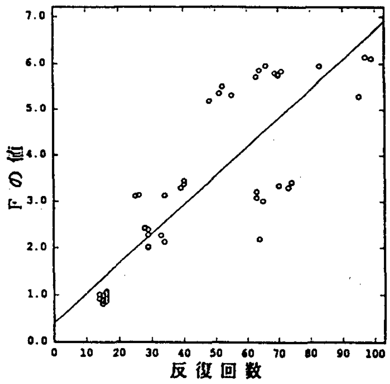

図２　散布図

６．おわりに

本実験例においては、t.p.m.の形が最良の場合は最悪の場合と比べ約１／５の反復回数で近似解が得られた。また、Ｆの値が小さい程、反復回数は減少し、反復回数のバラツキも小さくなった。以上のことから、本手法はA/D法の効率を改善することが確かめられた。

参考文献

[1]P.J.Schweitzer," Mathematical Computer Performance and Reliability" , G.Iazeolla , P.J.Courtois and A.Hordijk (editors) Elsevier Science Publishers B.V. (North-Holland),pp.275-286 (1984).

[2]P.J.Courtois," Decomposability-Queueing and Computer System Applications" ,Academic Press, New York,London (1977).