

# フローキングのある待ち行列モデルの数値解析

逆瀬川 浩孝  
(筑波大学・社会工学系)

## 1. はじめに

計算機システムの性能評価のための確率モデルとして、待ち行列モデル、特に積形式解を持つ待ち行列網モデルが良く使われている。これは、各サービスノードで待ち行列の長さに制限がなく、ある一定のサービス規律に従ってサービスを受け、ノード間をマルコフ的に推移する、というモデルでは、各ノードの待ち客数の同時定常分布が、ノードごとの待ち客数の定常分布の積として表される、すなわち、各ノードでの客のふるまいは、他のノードと独立になされる、という結果を得、という。このモデルでは、単純な結果を得るために、待ちは無制限に許されるとか、世帯はマルコフ的であるとか、サービス分布が指数型でない場合は、到着した客に対して直ちにサービスを開始しなければならないとかの、技巧的な仮定を設けたり、理想化を行ったりしている。ある場合には、これによって十分実用的な近似解が得られるが、モデルの前提条件が大きくくずれた時には別の解析法が必要になる。たとえば、待ち行列の長さには制限があるのが普通で、待ち客数が少なからぬ確率でその限度いっぱいになる場合などでは、他のノードへの影響が大きく、積形式解と近似解として用いるのは妥当でない。

あるサービス窓口で、サービスを待つ客がいるにもかかわらず、外部条件（他のノードでの混雑など）でサービスが実行できない状態（サービスを終了した客が先のノードへ進めないので、その窓口を占有し続ける場合など）を窓口のフローキングと呼ぶ。一般の待ち行列網モデルでフローキングを考えると、デッドロックを引き起こすおそれがあり、解析が複雑になるので、ここではループを含まず、客は一方方向に流れるだけのモデル（直列型待ち行列モデル）を考える。もし、待ち行列の長さには制限がなければ、このモデルの最大出力率は、最も能率の悪いノードのサービス率に等しいが、フローキングを起こすモデルを考えたとき、この数値がどれほど低下するか、ということ、待ち客のためのスペースを考えた上での基本的な関心事の一つになる。その他にも、系内滞在時間、系内客数、窓口の稼働率など、数量的に把握することが必要である。この小論では、ある種の直列型待ち行列モデルの定常分布を求める問題を取りあげ、それを数値的に求めるアルゴリズムを提示し、いくつかの数値例を使って積形式解との比較を行う。

## 2. モデル

ここでは次のような直列型待ち行列モデルを考える。

- 1)  $K$  個のサービスノードが、一列に並んでいる。 $k$  番目のノードを  $\sigma k$  段と呼ぶ。 $\sigma k$  段は  $c_k$  個の窓口と、最大  $N_k - c_k$  の待ちスペースとからなる。各窓口のサービス時間は、独立に、パラメータ  $\mu_k$  の指数分布に従う。
- 2) 系外からの客は、パラメータ  $\lambda$  のポアソン過程に従って  $\sigma 1$  段に到着する。もし、 $\sigma 1$  段の客の数が  $c_1$  未満ならば、直ちにサービスを開始される。もし

到着時に  $N_1$  人の客がいれば、到着した客は系内に入らず、失われる（呼損という）。それ以外の場合は、到着順に待ち行列を作る。

- 3) 客は、各段のサービスをその番号順に先着順でサービスを受け、第  $k$  段のサービスが終了したら系外に退去する。途中の段からは退去しない。
- 4) ある客が、第  $k$  段 ( $k < K$ ) の窓口でサービスを終えた時、第  $k+1$  段に  $N_{k+1}$  人の客がいた場合は、その客は次の段へ進むことができず、今迄サービスを受けていた窓口を占有したまま第  $k$  段に留まる。この時、もし、第  $k$  段の待ち行列が空でない場合、待ちっている客に対して、新たなサービスを開始することができない（窓口がフロックされた、という）。フロックされた窓口は、その客が第  $k+1$  段に進んだ時に、フロックを解除される。
- 5) ある段で、一人の客が退去した時（次の段へ進むか、あるいは、その段が最終段であれば、系外へ退去する）、もし、その直前の段で窓口をフロックしている客がいれば、そのうちの一人の客をとり込み、窓口のフロックを解除する。ある段の影響で、その直前の 2 つ以上の段において窓口がフロックされる場合が生ずるが、この時、フロックの解除は同時に行われる。

待ち行列モデルにおける基本的特性量の一つとして、最大処理速度、あるいは最大容量というものがある。これは上述のモデルで、到着率  $\lambda$  を無限大とした時の出力率として定義されるが、実際の計算には、ポアソン到着過程の代わりに、第 1 段の前に無尽蔵の倉庫を設けたモデルを考えるとよい。2 つのモデルを同時に扱う時、前者をモデル I、後者をモデル II と呼ぶ。以下断りなしに述べた部分はモデル I に関するものである。

### 3. 解析

前節で記述したモデルの系内客数の同時定常分布を求める問題を考える。モデルに含まれる分布の性質から、連続時間マールコフ連鎖として定式化することが可能である。第  $k$  段の客数を  $Q_k$ 、フロックされている窓口の数を  $B_k$  とすると、

$$Z = (Q_1, B_1; Q_2, B_2; \dots; Q_K, B_K) \quad (B_K \equiv 0)$$

はマルコフ連鎖となり、その状態集合  $S_K$  は

$$S_K = \{ (q_1, b_1; \dots; q_K, b_K); 0 \leq q_k \leq N_k, 0 \leq b_k \leq c_k, q_k \geq b_k, b_K = 0 \}$$

生成作用素の各元  $r(A \rightarrow A')$  ( $A, A' \in S_K$ ) は次のように与えられる。

$$r(A \rightarrow \xi_1(A)) = \lambda \quad (q_1 < N_1 \text{ の場合})$$

$$r(A \rightarrow \xi_k^{-1} \xi_{k+1}(A)) = \mu_k (\min(q_k, c_k) - b_k)$$

$$(k < K, q_{k+1} < N_{k+1}, b_{k-1} = 0 \text{ であるか}, k = K, b_{K-1} = 0 \text{ の場合})$$

$$r(A \rightarrow \gamma_k(A)) = \mu_k (\min(q_k, c_k) - b_k) \quad (k < K, q_{k+1} = N_{k+1} \text{ の場合})$$

$$r(A \rightarrow \xi_i^{-1} \gamma_j^{-1} \dots \gamma_{k-1}^{-1} \xi_{k+1}(A)) = \mu_k \cdot \min(q_k, c_k)$$

$$(k < K, q_{k+1} < N_{k+1} \text{ 又は } k = K \text{ で、} b_{k-1} \neq 0, \dots, b_j \neq 0, b_{j-1} = 0 \text{ の場合})$$

但し、 $\xi_k(A)$  は  $A = ((q_j, b_j)_j)$  の  $q_k$  に  $q_{k+1}$  に変えることを表し

$\gamma_k(A)$  は  $b_k < c_k$  に  $b_{k+1}$  に変えることを表わす。また、

$$\xi_{k+1}(A) = A \text{ である。}$$

状態集合の大きさ  $M$  は次の漸化式で与えられる。

$$f(0) = 1$$

$$f(1) = N_1 + 1$$

$$f(k) = (N_k + C_{k-1} + 1) f(k-1) - \frac{1}{2} C_{k-1} (C_{k-1} + 1) f(k-2) \quad k=2, 3, \dots, K$$

$$M = f(K)$$

$f(k)$  は、 $k$  段直列型モデルの状態集合の大きさを表わす。  
例として  $C_k = 2, N_k = 3 (k=1, 2, \dots)$  の場合の  $f(k)$  の値は次のようになる。

| $k$    | 1 | 2  | 3   | 4   | 5    | ... |
|--------|---|----|-----|-----|------|-----|
| $f(k)$ | 4 | 21 | 114 | 621 | 3384 | ... |

なお、モデル II では、初期値として  $f(0) = 0, f(1) = 1$  を用いる。

状態集合が確定し、生成作用素が与えられると、定常状態における分布は、連立一次方程式を解くことにより、与えられるが、その次元はかなり大きい。そこで生成作用素の構造を利用して、次元を減らすことを考える。

$\mathcal{S}_{k,n}$  とし、

$$(1) \quad \mathcal{S}_{k,n} = \{ (q_k, l_k)_k; \sum_k q_k = n \}$$

を定義すると、

$$\mathcal{S}_k = \bigcup_{n=0}^{\infty} \mathcal{S}_{k,n} \quad (\text{但し、} N = \sum_{k=1}^K N_k)$$

となり、 $\mathcal{S}_{k,n}$  の状態から推移可能な状態は  $\mathcal{S}_{k,n-1}, \mathcal{S}_{k,n}, \mathcal{S}_{k,n+1}$  のいずれかに含まれる。したがって、生成作用素  $A$  を  $\mathcal{S}_{k,n}$  に対応する行ごとにまとめ、 $n$  の順で並べかざると、小行列を要素とする行列の意味で、 $A$  は三重対角になる。

$$A = \begin{bmatrix} P_0 & Q_0 & & & \\ R_1 & P_1 & Q_1 & & \\ & R_2 & P_2 & Q_2 & \\ & & & \ddots & \\ 0 & & R_{N-1} & P_{N-1} & Q_{N-1} \\ & & & R_N & P_N \end{bmatrix} \quad \left( \begin{array}{l} R_n, P_n, Q_n \text{ は } \mathcal{S}_{k,n} \text{ から、それぞれ} \\ \mathcal{S}_{k,n-1}, \mathcal{S}_{k,n}, \mathcal{S}_{k,n+1} \text{ への} \\ \text{推移率をまとめたものである} \end{array} \right)$$

定常分布を与えるベクトルも、 $A$  と同じように  $\mathcal{S}_{k,n}$  の順に、分割して考える。

$$\pi = (\pi_0, \pi_1, \dots, \pi_N)$$

$$\pi_n = (\pi_{n1}, \pi_{n2}, \dots, \pi_{nM_n}) \quad (\text{但し、} M_n = |\mathcal{S}_{k,n}|)$$

この時、 $\pi$  は次の方程式の解として与えられる。

$$\begin{cases} \pi A = 0 \\ \pi \xi = 1 \end{cases} \quad \left( \begin{array}{l} \text{但し、}\xi \text{ は要素がすべて 1 のベクトルを表わし、} \\ \text{以後、次元が違っても同じ記号を使う} \end{array} \right)$$

あるいは

$$(2) \quad \begin{cases} \pi_0 P_0 + \pi_1 R_1 = 0 \\ \pi_{n-1} Q_{n-1} + \pi_n P_n + \pi_{n+1} R_{n+1} = 0 \quad (n=1, 2, \dots, N-1) \\ \pi_{N-1} Q_{N-1} + \pi_N P_N = 0 \\ \pi \xi = 1 \end{cases}$$

である。

$$p_n = P(\sum_{k=1}^K Q_k = n) = \sum_{j=1}^{M_n} \pi_{nj}$$

$$z_{n,j} = \pi_{n,j} / p_n$$

$$z_n = p_n^{-1} \pi_n$$

を導入すると、

$$(3) \quad \begin{cases} \tau_n = -\frac{p_{n-1}}{p_n} \tau_{n-1} Q_{n-1} P_n^{-1} - \frac{p_{n+1}}{p_n} \tau_{n+1} R_{n+1} P_n^{-1} & (n=1, 2, \dots, N-1) \\ \tau_N = -\frac{p_{N-1}}{p_N} \tau_{N-1} Q_{N-1} P_N^{-1} \\ \sum_{n=0}^N p_n = 1 \end{cases}$$

一方、 $A$  の行和が 0 になることを使えば、(3) 式から

$$\pi_{n-1} Q_{n-1} \xi = \pi_n R_n \xi$$

あるいは、

$$(4) \quad p_n = \frac{\tau_{n-1} Q_{n-1} \xi}{\tau_n R_n \xi} p_{n-1}$$

が導かれる。(3), (4) を使えば、適当な初期値  $\{p_n^{(0)}\}$  と初期ベクトル  $\{\tau_n^{(0)}\}$  から出発して、(3) 式を満たす  $\{p_n\}$ ,  $\{\tau_n\}$  を反復法で求めることができる。

(3) 式で、 $P_n$  の逆行列が使われるところから、 $\mathcal{A}_{k,n}$  の状態を適当に並べるとして、 $P_n$  を上三角行列にするることができるから、その逆行列は

$$P_n^{-1} = T_1 T_2 \cdots T_{M_n} \quad \text{但し、} T_j = \begin{pmatrix} 1 & 0 & -p_{nj}/p_{jj} & & \\ & 1 - p_{nj}/p_{jj} & 0 & & \\ & 0 & 1/p_{jj} & & \\ & & 0 & \ddots & \\ & & & 0 & 1 \end{pmatrix} \quad P_n = (p_{nij})$$

のように、簡単な行列の積として表すことができる。反復法のアルゴリズムは次の通りである。

#### アルゴリズム A (定常分布)

0.  $\varepsilon$  (許容誤差) を定める。  $\alpha_n^{(0)} = \lambda$  ( $n=0, 1, \dots, N-1$ ),  $\beta_n^{(0)} = c_k \mu_k$  ( $n=1, 2, \dots, N$ )

$$\tau_{n,j}^{(0)} = M_n^{-1} \quad (j=1, 2, \dots, M_n; n=0, 1, \dots, N), \quad l \leftarrow 1$$

1. ( $l$  回目反復)

$n=1, 2, \dots, N$  について順に

$$\tau_{n,j}^{(l)} = \frac{1}{-p_{nj}} \left\{ \sum_{i < j} \tau_{ni}^{(l)} p_{nij} + \frac{\beta_n^{(l-1)}}{\alpha_{n-1}^{(l-1)}} \sum_i \tau_{n-i}^{(l-1)} g_{n-i,j} + \frac{\alpha_n^{(l-1)}}{\beta_{n+1}^{(l-1)}} \sum_i \tau_{n+1,i}^{(l-1)} r_{n+1,j} \right\}$$

( $j=1, 2, \dots, M_n$ )

$$\alpha_n^{(l)} = \sum_{i,j} \tau_{n,i}^{(l)} g_{n,i,j} \quad , \quad \beta_n^{(l)} = \sum_{i,j} \tau_{n,i}^{(l)} r_{n+1,j}$$

2. (判定)

$$\max_{n,j} |(\tau_{n,j}^{(l)} - \tau_{n,j}^{(l-1)}) / \tau_{n,j}^{(l)}| > \varepsilon$$

ならば、 $l \leftarrow l+1$  として 1 からやり直し。そうでなければ収束したとみなし、

$$p_0 = \left\{ 1 + \sum_{n=1}^N \pi_{j=1}^n \frac{\alpha_{j-1}^{(l)}}{\beta_j^{(l)}} \right\}^{-1}, \quad p_n = \frac{\alpha_{n-1}^{(l)}}{\beta_n^{(l)}} p_{n-1}, \quad \pi_n = p_n \tau_n^{(l)}$$

とする。

アルゴリズムに見られる  $p_{nij}$ ,  $g_{nij}$ ,  $r_{nij}$  は生成作用素の要素であるが、生成作用素が 11 かに疎な行列であることも、それらと密に記憶しておくことは大変なので、必要の都度、求めるようにした。この為には、 $\mathcal{A}_{k,n}$  の状態ベクトルは順に番号を付して整列させること、与えられた順番の状態ベクトルを生成することが必要である。それらのアルゴリズムを以下に述べる。

アルゴリズム B ( $\delta_{k,m}$  から  $\{0, 1, \dots, M_n - 1\}$  への 1 対 1 写像)

0.  $(g_1, l_1, \dots; g_k, l_k)$  ( $\sum g_k = n$ ) が与えられる。  
 $s \leftarrow l_1, m \leftarrow g_1, k \leftarrow 2$
1.  $m \leftarrow m + g_k$   
 $\text{もし } k = K, \text{ または } g_{k+1} < N_{k+1} \text{ ならば } s \leftarrow s + \sum_{j=0}^{g_k-1} g_{k-1, m-j} \text{ と } k \geq 3 \text{ へ。}$
2. ( $k < K$  から  $g_{k+1} = N_{k+1}$  の場合)  
 $\text{もし } n_k < N_k \text{ ならば } s \leftarrow s + \sum_{j=0}^{g_k-1} (\min(j, c_k) + 1) g_{k-1, m-j} + l_k g_{k-1, m-g_k}$   
 $\text{もし } n_k = N_k \text{ ならば } s \leftarrow s + \sum_{j=0}^{g_k-1} (\min(j, c_k) + 1) g_{k-1, m-j} + l_k h_{k-1, m-g_k}$
3.  $k < K$  ならば  $k \leftarrow k + 1$  と加えて 2 へ。そうでなければ、 $s$  が目的の値である。

且し、 $g_{1,n} = 1$  ( $n = 0, 1, \dots, N_1$ ),  $= 0$  (それ以外)  
 $h_{1,n} = \min(n, c_1) + 1$  ( $n = 0, 1, \dots, N_1$ ),  $= 0$  (それ以外)

$$(5) \quad \begin{cases} g_{k,n} = \sum_{j=0}^{\min(n, N_k-1)} g_{k-1, n-j} + \chi_{N_k}(n) h_{k-1, n-N_k} \\ h_{k,n} = \sum_{j=0}^{\min(n, N_k-1)} (\min(j, c_k) + 1) g_{k-1, n-j} + \chi_{N_k}(n) (c_k + 1) h_{k-1, n-N_k} \end{cases}$$

( $n = 0, 1, \dots, N_1 + \dots + N_{K+1}; k = 2, 3, \dots, K$ )

$\chi_n(m) = 1$  (if  $m \geq n$ ),  $= 0$  (if  $m < n$ )

アルゴリズム C ( $\{0, 1, \dots, M_n - 1\}$  から  $\delta_{k,m}$  への 1 対 1 写像)

0.  $s$  ( $0 \leq s < M_n$ ) が与えられる。 $m \leftarrow n, k \leftarrow K, l_k \leftarrow 0$ .
1.  $\text{もし } k = K \text{ または } g_{k+1} < N_{k+1} \text{ ならば } l_k \leftarrow 0,$   
 $g_k \leftarrow \min_{0 \leq j < N_k} (\sum_{i=0}^j g_{k-1, m-i} > s) \quad \text{または } N_k,$   
 $s \leftarrow s - \sum_{i=0}^{g_k-1} g_{k-1, m-i}, \quad m \leftarrow m - g_k \quad \text{と } k \geq 3 \text{ へ。}$
2. ( $k < K$  から  $g_{k+1} = N_{k+1}$  の場合)  
 $g_k \leftarrow \min_{0 \leq j < N_k} (\sum_{i=0}^j (\min(i, c_k) + 1) g_{k-1, m-i} > s) \quad \text{または } N_k,$   
 $s \leftarrow s - \sum_{i=0}^{g_k-1} (\min(i, c_k) + 1) g_{k-1, m-i}, \quad m \leftarrow m - g_k$   
 $\text{もし } g_k < N_k \text{ ならば } l_k \leftarrow \lfloor s / g_{k-1, m-g_k} \rfloor, \quad s \leftarrow s - l_k \cdot g_{k-1, m-g_k}$   
 $\text{もし } g_k = N_k \text{ ならば } l_k \leftarrow \lfloor s / h_{k-1, m-g_k} \rfloor, \quad s \leftarrow s - l_k \cdot h_{k-1, m-g_k}$
3.  $\text{もし } k > 2 \text{ ならば } k \text{ から } 1 \text{ まで } 3 \text{ 回 } k \geq 1 \text{ となる。} \quad \text{そして } g_1 \leftarrow m, l_1 \leftarrow s$

モデル II の定常分布を求める場合は、状態集合の分割とし、(1) 式の代りに

$$(6) \quad \delta_{k,m} = \{ (g_k, l_k)_k; \sum_{k=2}^K g_k + l_1 = m \}$$

とある。これは、次のように考えるとよい。モデル I では、第 1 段をなくし、第 2 段の待ちスペースを 2 倍  $c_1$  増やす。客は第 2 段にポアソン到着するが、その倍々  $\lambda$  は第 2 段の客の数に依存し、もし客数  $g_2$  が  $N_2$  以下ならば  $c_1 \mu_1$ , それ以上ならば  $(N_2 + c_2 - g_2) \mu_2$  と変化する。モデル II と同等のふるまいをする  $K-1$  段直列型モデル (変形モデル I) が与えられる。(6) 式はこの変形モデルの状態集合を系内客数に層別したものとなる。(6) 式を用いた場合のアルゴリズム A, B, C の変更箇所は以下の通りである。1) アルゴリズム A のステップ 200 へ  $\alpha_n^{(0)} = c_1 \mu_1$

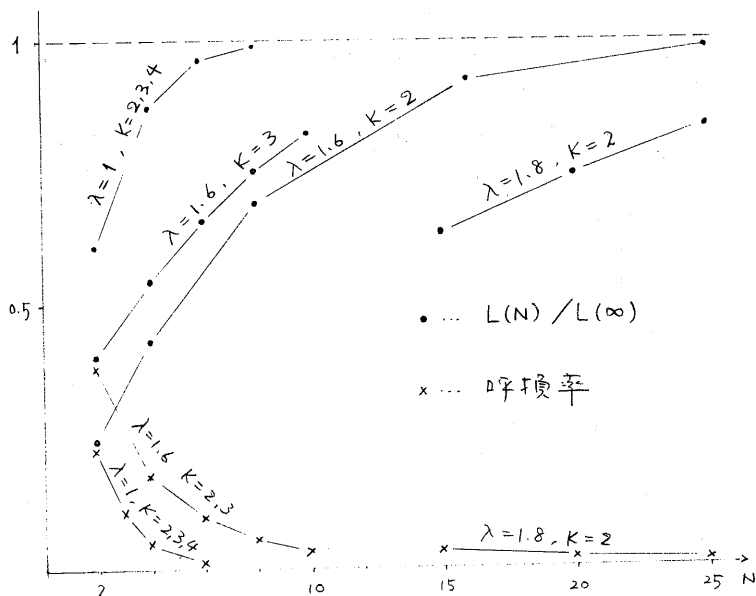
( $n=0,1,\dots,N-1$ )。2) アルゴリズム A のステップ 0 まで。  $S \leftarrow 0, m \leftarrow l_1$ 。3)  $g_{k,m}$  の初期値として、  $g_{1,0} = 1, l_{1,n} = 1$  ( $n=0,1,\dots,C_1$ ),  $= 0$  (それ以外)。4) アルゴリズム A のステップ 3 まで。  $g_1 \leftarrow C_1, l_1 \leftarrow m$ 。

#### 4. プログラムと数値例

前節で述べたアルゴリズムに基いて、直列型待ち行列モデルの定常分布を求めるプログラムが、FORTRAN により書かれ、数値実験に用いられている。プログラムは約 500 ステップで、主たる作業領域は  $\tau_n^{(k)}, \tau_n^{(k-1)}$  のための大きさ  $M + \max M_n$  の 1 つの次元配列と、  $g_{k,m}, l_{k,m}$  のための大きさ  $K \times N$  の 2 つの 2 次元配列、  $\alpha_n^{(k)}, \beta_n^{(k)}$  などのための大きさ  $N \times 11$  以下の 1 次元配置に与えられる。収束までの反復回数、実行時間は、モデルのパラメータによってそれぞれ異なり一定の傾向をとらなないが、次元 ( $M$ ) の増加に従い、緩やかに指数的に増大する、  $\lambda$  の増加に伴い、急速に増大する、などがある。数値例として、  $C_k = 2, \mu_k = 1$  ( $\forall k$ ), 誤差  $\varepsilon = 10^{-9}$  の場合、  $K=4, N_k=6$  ( $\forall k$ ),  $\lambda=0.5$  まで、  $M=4491$  となるが、約 75 秒、62 回の反復で収束し、  $K=3, N_k=10$  ( $\forall k$ ),  $\lambda=0.75$  まで、  $M=1787$  となり、約 56 秒、150 回の反復で収束した。(使用した計算機は筑波大学にある FACOM M-200 である)。以下、いくつかの数値例を挙げる。

##### 4.1. 数値例 1

$C_k = 2, \mu_k = 1, N_k = N$  (一定) の場合、系全体の平均客数  $L(N)$  を  $N \rightarrow \infty$  の場合と比較する。右の図は  $L(N)/L(\infty)$  を示しているが、  $\lambda, K$  の値に対してプロットしたものである。  $\lambda$  が増加するにつれて待ちスペースの制限による影響が指数的に増大する様子がわかる。トラフィック密度が 1 に近い場合、有限待ちモデルを無限待ちモデルで近似することは危険である。



##### 4.2. 数値例 2

系全体の効率を決める尺度の 1 つに最大利用率がある。これは、モデル II の出力率を、最も能率の悪いノードのサービス率で割ったものである。  $C_k = N_k = C, \mu_k = C^{-1}$  とした場合の最大利用率を示している  $K, C$  の値を計算したものが右の数表である。これは、ノードとして同じサービス率を保った時、窓口の数を増やすと、効率がどれだけ改善されるかを示すものである。

| C | K    |      |      |      |  |
|---|------|------|------|------|--|
|   | 2    | 3    | 4    | 5    |  |
| 1 | .667 | .564 | .515 | .486 |  |
| 2 | .750 | .666 | .624 | .599 |  |
| 3 | .791 | .718 | .681 |      |  |
| 4 | .816 | .751 | .717 |      |  |