



## 論文

## Gate のある M/M/m 待行列ネットワーク・モデルの解析\*

石黒美佐子\*\*

## Abstract

The results contained herein relate to the problem of obtaining a product-form solution to the equilibrium distribution in the queuing network model of the open batch multi-job processing system.

The model is characterized by the gate through which jobs are passed to executing stages under the restriction of the job multiplicity. Excessive jobs are kept waiting in a queue in front of the gate.

It is easily shown that the original equilibrium equations do not have the product-form solutions since the local balance condition does not hold. So we attempt to approximate the equilibrium probabilities of such an untractable case by the product-form solutions. The system is modeled with two Markovian processes, where the executing stages are independently analyzed as a sub-module and interconnected to the global module.

The method is also applied to generalized open batch system and the system with job classes.

## 1. はじめに

オープンバッチ多重ジョブ処理システムを M/M/m 待行列モデルを用いて解析することは、文献 1) で既に試みた。この時の解析手法を定式化していくと、基本的には Fig. 1 で示すような局所平衡 (local balance) が成立しない待行列ネットワーク・モデルの平衡確率を求めることに帰着する。

ごく普通の M/M/m 待行列ネットワーク・モデルでは、各待行列長、いわゆる waiting room が有限であっても、その平衡確率は有限乗積の積の形をなし、特に  $m$  が 1 の場合には等比数列の積となることが知られている<sup>2)</sup>。また複数のサブチェーンを持つシステムに対しても、waiting room の大きさに制約がない場合はやはり積の形の解 (product-form solution) が求められることが示されている<sup>3)</sup>。

これから解析しようとする Gate を持つ待行列ネッ

トワークモデルでは、複数の待行列のネットワークで構成されるサブモジュール内の待行列長の和が有限であるという制約を持つために、全パラメータに対して同時に平衡方程式を立て、平衡確率を求めようとしても、境界で式の両辺の各項の局所平衡に矛盾が生じ、積の形のように簡単な解にならない上に解を式として導くことが困難である。しかしながら、この種の待行列モデルに対しても近似的な解法を用いて積の形の解を得ようというのが本文の課題である。

今まで平衡確率の定式化ができたほとんどの場合は局所平衡が成立する場合である。しかるに局所平衡が成立しない場合は、Fig. 1 のモデルのように非常に単純なものでも、3つのパラメータ ( $v, x, z$ ) に対して

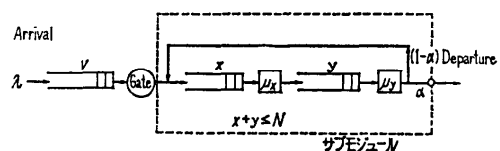


Fig. 1 Example of the queuing network model with a gate (M-1 model).

\* Analysis for M/M/m Queuing Network Model with a Gate by Misako ISHIGURO (JAPAN Atomic Energy Research Institute).

\*\* 日本原子力研究所

同時に平衡方程式を立て解を得ようとしても、解の形が非常に複雑になり定式化できない。

ここで試みた近似的解法の骨子は、Gate により系を2分し、2つマルコフ過程によりモデルを近似し、それぞれ積の形の解を得、これらを合成して近似解を導くというものである。

この解法を、文献 1) の原研オープンバッチ・システム、文献 2) で示された一般化されたオープンバッチ・システムおよびジョブ・クラスを持つシステムに適用する。

近似法により簡単な形の解を得る試みは、多重待行列を単一待行列で近似するなど報告されている<sup>9)</sup>。また一般の分布に対する平衡確率を再帰法 (recursive technique) を用いて数値的に近似解を求めることが試みられている<sup>9)</sup>。

どのような指数分布型ネットワークに対しても、平衡方程式が成立するかぎり、平衡確率を数値的に求めることは、変数の数が多くはなるが単に連立一次方程式の解を求める問題に過ぎない。しかしながら、平衡確率が定式化できれば、式の示す意味が直観的に把握できるだけでなく、これを用いて導かれる平均待行列長、タンアランド・タイムなどの解析結果も定式化できる。特に積の形のように簡単な形で解が得られれば、類似モデルの解が容易に推測できる点で有難い。

## 2. 解析方法

Fig. 1 を例にとって解析を試みる。

### 2.1 M-1 モデル

Fig. 1 のモデルは、文献 1) で取扱ったオープンバッチ・モデルを単純化したものである。ここではバッチ処理におけるジョブの実行段階をサブモジュールとして点線で囲まれた部分で表現している。一般的にジョブの多重度に上限があることから、このサブモジュールの系内の和が一定値： $N$  以下に保つことが必要となる。したがってこのモデルでは、サブモジュールの入口に Gate を設けて、Gate の外側にサブモジュールへの入力待行列つまりジョブの実行待行列を置き、サブモジュールの系内の和が  $N$  未満となったとき、Gate が開き、次のジョブが通過できるようになっている。またサブモジュールは、CPU 処理と I/O 処理のくり返しにより表現されているが完全なサイクリック・モデルではなく、ジョブの終了・開始による系の増減つまり多重度の増減が相俟込まれている。

ジョブは  $\lambda$  を入力率とするポアソン到着し、CPU と

I/O に対してそれぞれ  $\mu_x, \mu_y$  をパラメータとする指数型のサービスを受け、FCFS のスケジュールにより処理される。I/O 終了後、 $(1-\alpha)$  の確率で CPU に処理がもどり、 $\alpha$  の確率でシステムを去る。 $v, x, y$  をそれぞれ実行待、CPU 待、I/O 待行列長とすると、 $x+y \leq N$  の条件を持つ。 $x+y=N$  のときは、到着したジョブは実行待行列で待たされ、1つのジョブの実行が終了すると直ちに次のジョブの実行が開始される。 $x+y < N$  のときは、到着と同時にジョブの実行が開始される。

### 2.2 近似解法の概略

Fig. 1 の点線の枠内を1つのサービスを提供する窓口と考え、まず Gate の外側の状態  $(v, n)$ 、ここで  $n$  は Gate の内側の多重度、について平衡方程式を立てる。他方、Gate の内側の状態、つまりサブモジュールに関する状態  $(x, y)$ 、に対し平衡方程式を立てる。この際、Gate の開く条件を考慮し、Gate の内外を関連づけながら両者を独立のマルコフ過程としてモデル化し、別々に平衡確率を得る。Gate の内と外では、Gate を通過して自分の待行列長に増減が生じる場合以外の相手方の動きを考慮しない。つまり相手方の状態は自分の状態に関係なく平均化されたものとして取扱われる。

このようなモデルの近似は、実際の多重ジョブ処理において、実行時の動きは、実行待行列長に関係しないこと、また実行待行列長は、実行時の詳細に関係がなくジョブの終了度合にのみ依存すると考えられるから矛盾がない。

このようにして得た各々の平衡確率  $P_1(v, n), P_2(x, y)$  は等比数列の積の形をなす。そこでこれらを次のように合成すれば、やはり等比数列の積の形の解を得ることができる。

$$\begin{aligned} P(v, a, b) &= P, [V=v, \text{多重度}=n] P, [X=a, \\ & \quad Y=b | X+Y=n] \\ &= P_1(v, n) P_2(a, b) / \sum_{y=0}^n P_2(n-y, y). \end{aligned}$$

ただし  $n=a+b$ 。

この近似解は、全パラメータ  $(v, x, y)$  に対して同時に平衡方程式を立てて得られる本来の解  $Q(v, x, y)$  と数値的に良く一致することを後に示す。もちろん本来の解とはマルコフ連鎖の作り方が異なるわけだから完全には一致しない。

### 2.3 本来の解

本来の解  $Q(v, x, y)$  を求める際に、平衡方程式を

Table 1 Corresponding relations between the terms.

条件(b)の場合の項	(5)式による対応	(6)式による対応
$\lambda Q(0, x-1, y)$ $+ \mu_y(1-\alpha)Q(0, x-1, y+1)$ $+ \mu_y\alpha Q(1, x-1, y+1)$ $+ \mu_x Q(0, x+1, y-1)$	$\lambda Q(0, x, y)$ $+ \mu_x(1-\alpha)Q(0, x, y)$ $+ \mu_x\alpha Q(0, x, y)$ $+ \mu_y Q(0, x, y)$	$\mu_x\alpha Q(0, x, y)$ $+ \mu_x(1-\alpha)Q(0, x, y)$ $+ \lambda Q(0, x, y)$ $+ \mu_y Q(0, x, y)$

$x+y=N$  ……は副次点に出てくる関係

解く過程で境界で局所平衡に矛盾が生じることをまず示す。

平衡方程式は以下のように書ける。

(a)  $x+y=N, v>0$  のとき

$$\begin{aligned} &\lambda Q(v-1, x, y) + c_x \mu_y \{ \alpha Q(v+1, x-1, y+1) \\ &+ (1-\alpha)Q(v, x-1, y+1) \} \\ &+ c_y \mu_x Q(v, x+1, y-1) \\ &= (\lambda + c_x \mu_x + c_y \mu_y) Q(v, x, y). \end{aligned} \tag{1}$$

(b)  $x+y=N, v=0$  のとき

$$\begin{aligned} &c_x \lambda Q(0, x-1, y) + c_x \mu_y \{ \alpha Q(1, x-1, y+1) \\ &+ (1-\alpha)Q(0, x-1, y+1) \} \\ &+ c_y \mu_x Q(0, x+1, y-1) \\ &= (\lambda + c_x \mu_x + c_y \mu_y) Q(0, x, y). \end{aligned} \tag{2}$$

(c)  $x+y < N$  のとき

$$\begin{aligned} &c_x \lambda Q(0, x-1, y) + \alpha \mu_y Q(0, x, y+1) \\ &+ c_x \mu_y (1-\alpha)Q(0, x-1, y+1) \\ &+ c_y \mu_x Q(0, x+1, y-1) \\ &= (\lambda + c_x \mu_x + c_y \mu_y) Q(0, x, y). \end{aligned} \tag{3}$$

(1), (2), (3)式に加えて確率分布だから

$$\sum_{v=0}^{\infty} \sum_{x+y \leq N} Q(v, x, y) = 1. \tag{4}$$

ここで  $c_x=0(x=0)$  または  $1(x>0)$ .

解が  $c \rho_1^x \rho_2^y \rho_3^v$  なるパラメータ分離した等比数列の積となるためには、係数  $c_x, c_y$  に対する項が両辺で互に等しくならなければならない。つまり局所平衡しなければならない。そうだとすると(a)の場合には、

(1)式から

$$\begin{aligned} &\{ \lambda Q(v-1, x, y) - \lambda Q(v, x, y) \} \\ &+ c_x \{ [ (1-\alpha) \{ \mu_y Q(v, x-1, y+1) \\ &- \mu_x Q(v, x, y) \} + \alpha \{ \mu_y Q(v+1, x-1, y+1) \\ &- \mu_x Q(v, x, y) \} ] + c_y \{ \mu_x Q(v, x+1, y-1) \\ &- \mu_y Q(v, x, y) \} \} = 0 \end{aligned}$$

より第1項と第3項から次の関係式を得る。

$$\begin{aligned} Q(v, x, y) &= Q(v-1, x, y) \\ &= \mu_x / \mu_y \cdot Q(v, x+1, y-1), \\ &x+y=N. \end{aligned} \tag{5}$$

同じく(c)の場合には、(3)式から

$$Q(0, x, y) = \lambda / (\alpha \mu_x) \cdot Q(0, x-1, y)$$

$$\begin{aligned} &= \mu_x / \mu_y \cdot Q(0, x+1, y-1), \\ &x+y < N. \end{aligned} \tag{6}$$

しかるに(b)の場合は(a), (b)の境界をなす場合であるが、(5)式と(6)式のいずれを適用するかによってTable 1に示すように項の対応関係に矛盾が生じる。

これらのことから(1)~(4)の連立一次方程式の解は積の形では得られない。

### 2.4 近似解

既に述べたように、近似モデルの解を次のように与える。

$$P(v, a, b) = P_1(v, n) P_2(a, b) \Big/ \sum_{y=0}^n P_2(n-y, y).$$

ただし、 $a+b=n, n < N$  のときは  $v=0$ .

#### 2.4.1 Gateの外側の平衡方程式

(a)  $n=N, v>0$  のとき

$$\begin{aligned} &\lambda P_1(v-1, N) + \alpha P_2^{(N)} \mu_y P_1(v+1, N) \\ &= \{ \lambda + \alpha P_2^{(N)} \mu_y \} P_1(v, N). \end{aligned} \tag{8}$$

(b)  $n=N, v=0$  のとき

$$\begin{aligned} &\lambda P_1(0, N-1) + \alpha P_2^{(N)} \mu_y P_1(1, N) \\ &= \{ \lambda + \alpha P_2^{(N)} \mu_y \} P_1(0, N). \end{aligned} \tag{9}$$

(c)  $n < N$  のとき

$$\begin{aligned} &\lambda P_1(0, n-1) + \alpha P_2^{(n+1)} \mu_y P_1(0, n+1) \\ &= \{ \lambda + \alpha P_2^{(n)} \} P_1(0, n), \end{aligned} \tag{10}$$

$$\sum_{v=0}^{\infty} \sum_{n=0}^N P_1(v, n) = 1. \tag{11}$$

ここで  $P_2^{(n)}$  は多重度が  $n$  のときの I/O の使用率を表わす。

$$\begin{aligned} P_2^{(n)} &= P_r[y > 0 | x+y=n] \\ &= \sum_{y=1}^n P_2(n-y, y) \Big/ \sum_{y=0}^n P_2(n-y, y). \end{aligned} \tag{12}$$

(8)~(11)の方程式の解は以下のようになる。

$$P_1(v, N) = \{ \lambda / (\alpha \mu_y P_2^{(N)}) \} \cdot P_1(0, N), \tag{13}$$

$$P_1(0, n) = \{ \lambda / (\alpha \mu_y) \} \cdot \frac{P_1(0, 0)}{P_2^{(n)} P_2^{(n-1)} \dots P_2^{(1)}}. \tag{14}$$

#### 2.4.2 Gateの内側の平衡方程式

$$\begin{aligned} &c_y \mu_x P_2(x+1, y-1) + c_x \{ P_1^{(n)} \alpha + (1-\alpha) \} \\ &\times \mu_y P_2(x-1, y+1) + c_x \lambda P_2(x-1, y) \end{aligned}$$

$$=(c_x \mu_x + c_y \mu_y) P_2(x, y). \tag{15}$$

ここで  $P_1^{(n)}$  は多重度が  $n$  のとき実行待行列が 0 でない確率,

$$P_1^{(n)} = \begin{cases} P_r[v > 0 | x + y = N], & n = N, \\ 0, & n < N, \end{cases}$$

$$P_1^{(N)} = \sum_{v=1}^{\infty} P_1(v, N) \Big/ \sum_{v=0}^{\infty} P_1(v, N). \tag{16}$$

(15)式から次の関係を得る.

$$\begin{aligned} P_2(x, y) &= \mu_x / \mu_y \cdot P_2(x + 1, y - 1) \\ &= (\mu_x / \mu_y)^y P_2(x + y, 0), \end{aligned} \tag{17}$$

$$P_2(x, y) = \lambda / \{\alpha \mu_x (1 - P_1^{(n)})\} P_2(x - 1, y). \tag{18}$$

$$\delta_1 = \mu_x / \mu_y, \rho_y = \lambda / (\alpha \mu_y), \rho_x = \lambda / (\alpha \mu_x) \text{ とおき,} \tag{12}$$

式に (17), (18) の関係を代入すると,

$$\begin{aligned} P_2^{(n)} &= \frac{\delta_1 \rho_x \sum_{y=0}^{n-1} P_2(n-y, y)}{(1 - P_1^{(n)}) \sum_{y=0}^n P_2(n-y, y)} \\ &= \frac{\delta_1 (1 - \delta_1^n)}{1 - \delta_1^{n+1}}, \quad 1 \leq n \leq N. \end{aligned} \tag{19}$$

以上のことから,  $a + b = N$  の場合は, (13), (14), (19)式から,

$$\begin{aligned} P_1(v, N) &= (\rho_y / P_2^{(N)})^v P_1(0, N) \\ &= (\rho_y / P_2^{(N)})^v (1 - P_1^{(N)}) \sum_{y=0}^N P_2(N-y, y), \end{aligned} \tag{20}$$

$$P_2(a, b) = \rho_x^a \delta_1^b (1 - P_1^{(N)}) \cdot P_2(0, 0). \tag{21}$$

$P_1(0, 0) = P_2(0, 0) = P_0$  とおき, (20), (21) 式を (7) 式に代入すれば

$$P(v, a, b) = (\rho_y / P_2^{(N)})^v \rho_x^a \rho_y^b P_0. \tag{22}$$

$a + b < N$  のとき,  $P(0, a, b) = \rho_x^a \rho_y^b P_0$  となり (22) 式に含めることができる.

### 2.5 計算結果の比較

本来の解は, パラメータ  $\lambda, \mu_x, \mu_y, \alpha$  に数値を設定し, (1)~(4)式から成る連立一次方程式を解くことにより数値的に求めることが可能である.

この数値解  $Q(v, x, y)$  と (22)式に数値を代入して得られる近似モデルの解  $P(v, x, y)$  を比較する. Fig. 2 は,  $N=3, \alpha=1, \lambda=0.5, \mu_x=1, \mu_y=0.8$  の場合の両解をプロットしたものである.  $v=0$  のとき, つまり境界では両者の値に差が出る傾向にある.

M-1 モデルで  $\alpha=1$  の場合は, ジョブは CPU と I/O を 1 回ずつ使用すれば終るという極端な場合であり, 実行待行列と CPU 待行列を 1 つにまとめてもよいわけだから実際には意味がないが  $\alpha=1$  の場合に誤差が最大となるはずだから敢えて計算してみた.

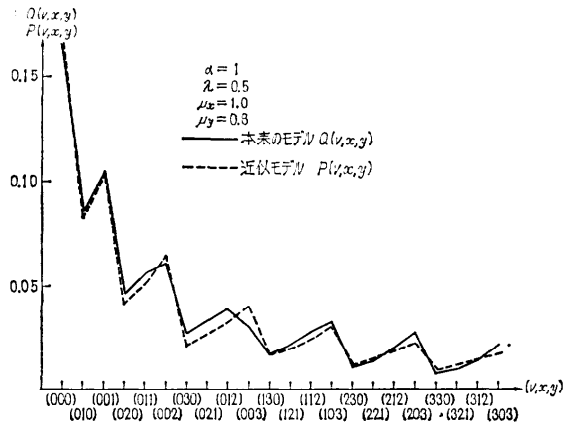


Fig. 2 Comparison of the equilibrium probabilities in M-1 model ( $\alpha=1$ ).

Table 2 Comparison of the equilibrium probabilities in M-1 model ( $\alpha < 1$ ).

$(v, x, y)$	平衡確率	$Q(v, x, y)$	$P(v, x, y)$
(000)		0.1657	0.1657
(010)		0.0829	0.0829
(001)		0.1036	0.1036
(020)		0.0414	0.0414
(011)		0.0518	0.0518
(002)		0.0637	0.0647
(030)		0.0209	0.0207
(021)		0.0261	0.0259
(012)		0.0324	0.0324
(003)		0.0403	0.0405
(130)		0.0157	0.0157
(121)		0.0196	0.0196
(112)		0.0245	0.0245
(103)		0.0307	0.0306
(230)		0.0118	0.0119
(221)		0.0148	0.0148
(212)		0.0185	0.0185
(203)		0.0232	0.0231

$\alpha=0.01, \lambda=0.05, \mu_x=1.0, \mu_y=0.8$

実際のシステムでは, ジョブは CPU と I/O の数千回のくり返しにより処理されるので  $\alpha < 1$  なる値となっている.  $\alpha=0.001$  の場合には両解は小数第 4 位まで完全に一致する.  $\alpha=0.01$  の場合でも Table 2 に示すように, 両計算値はほぼ等しい. つまり近似モデルの解は, Gate の出入り時に誤差を含むので, Gate の出入り率 ( $\alpha$ ) が小さい場合には, 本来の解に非常に近くなる. このことから近似モデルの解が充分使いものになることが数値的に実証できる.

### 3. システム例とその解析

Gate を持つ待行列ネットワーク・モデルにより, オープンバッチ・多重ジョブ処理システムの 2, 3 の例

について解析を試みる。いずれの場合も積の形の解を得ることができる。M-1 モデルと同じくシステム的全パラメータに対して同時にモデル化しても解の定式化が困難なため、Gate の外側と内側に分けて別々に平衡確率を求め、それらを合成して近似解を得る。

3.1 原研オープンバッチ・システム (M-2 モデル)

Fig. 3 で示す原研システムは既に文献1) で解析が済んでいるものであるが、Gate を持つ待行列モデルの考えを入れて改めてその近似モデルによる解を求め、先の解との関連を明らかにする。

M-2 モデルではオープンバッチ処理を、

入力→実行→出力

の3段階に分けてモデル化し、M-1 モデルをより実際のモデルの動きに近づけている。

M-1 モデルと同じく Gate の内外を分けて、それぞれの平衡確率  $P_1(v, z, n), P_2(x, y_1, y_2)$  を求め、近似モデルの解  $P(v, z, a, b_1, b_2)$  を次式から求める。

$$P(v, z, a, b_1, b_2) = P_r[V=v, Z=z, \text{多重度}=n] \times P_r[X=a, Y_1=b_1, Y_2=b_2 | X+Y_1+Y_2=n] = P_1(v, z, n)P_2(a, b_1, b_2) / \sum_{x+y_1+y_2=n} P_2(x, y_1, y_2).$$

ただし  $n = a + b_1 + b_2, n < N$  のとき  $v = 0$ . (23)

$$\delta_1 = 2\mu\sigma_1/\mu_1, \delta_2 = 2\mu\sigma_2/\mu_2, \rho_x^{(j)} = \lambda/(l_j\mu_x), \rho_x = \lambda/(2\mu\alpha\sigma_1), \rho_1 = \lambda/(\alpha\mu_1), \rho_2 = \lambda/(\alpha\mu_2)$$

とおくと(20), (21)式を導いたのと同様にして(24), (25)式を得る。

$$P_1(v, z, n) = (\rho_1/P_2^{(N)})^v \left( \prod_{j=1}^z \rho_x^{(j)} \right) \rho_1^n P_0 / (P_2^{(n)} P_2^{(n-1)} \dots P_2^{(1)}) = (\rho_1/P_2^{(N)})^v \left( \prod_{j=1}^z \rho_x^{(j)} \right) (1 - P_1^{(n)}) \times \sum_{x+y_1+y_2=n} P_2(x, y_1, y_2). \quad (24)$$

$$P_2(a, b_1, b_2) = 2\rho_x^n \delta_1 b_1 \delta_2 b_2 P_0 / (1 - P_1^{(n)}). \quad (25)$$

ここで  $l_j = j (j < L)$  または  $L (j \geq L)$ ,

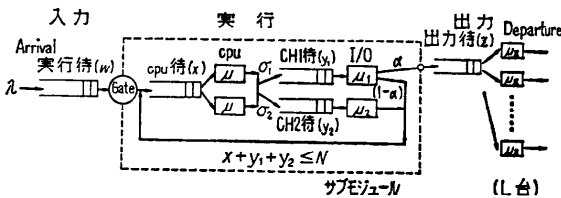


Fig. 3 JAERI model (M-2 model).

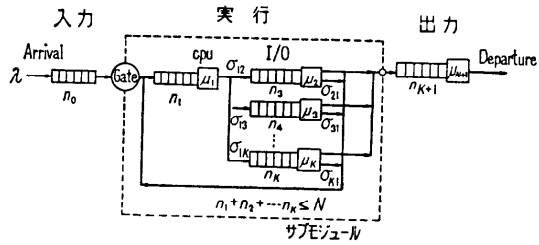


Fig. 4 Generalized open batch processing model (M-3 model).

$$P_1^{(n)} = \begin{cases} P_r[v > 0 | x + y_1 + y_2 = N], & n = N, \\ 0, & n < N, \end{cases} \quad (26)$$

$$P_2^{(n)} = P_r[y_1 > 0 | x + y_1 + y_2 = n]$$

$$= \frac{\delta_1 \rho_x}{1 - P_1^{(n)}} \frac{\sum_{x+y_1+y_2=n-1} P_2(x, y_1, y_2)}{\sum_{x+y_1+y_2=n} P_2(x, y_1, y_2)}. \quad (27)$$

(24), (25)式を(23)式に代入すると

$$P(v, z, a, b_1, b_2) = \begin{cases} 2(\rho_1/P_2^{(N)})^v \left( \prod_{j=1}^z \rho_x^{(j)} \right) \rho_x^n \rho_1^{b_1} \rho_2^{b_2} P_0, & a > 0, \\ (\rho_1/P_2^{(N)})^v \left( \prod_{j=1}^z \rho_x^{(j)} \right) \rho_1^{b_1} \rho_2^{b_2} P_0, & a = 0. \end{cases} \quad (28)$$

(28)式は文献1)で得た解と次のように対応づけられる

$\rho_1/P_2^{(n)}$  は多重度が  $n$  のときの実行待行列の増減のパラメータである。文献1)ではこの値は  $\lambda/\{n\mu(n)\}$  として与えられた。ここで  $\mu(n)$  は多重度が  $n$  のときのジョブの終了率(実行経過時間の逆数)であり、 $n\mu(n)$  は  $n$  個のジョブのうちいずれか1つが終了する率を表わし、

$$n\mu(n) = P_2^{(n)} \mu_1 \alpha \quad (29)$$

で与えられている。しかるに M-2 モデルでは、(24)式から、

$$P_1(0, 0, n) = \lambda / \{P_2^{(n)} \mu_1 \alpha\} P_1(0, 0, n-1) \quad (30)$$

だから本質的には、同じ解法である。

3.2 オープンバッチ・システムの一般化 (M-3 モデル)

Fig. 4 で示すモデルは、文献2)で一般化されたオープンバッチ・システムとして取上げられたモデルに多重度の制約を加えたものである。ジョブは到着率  $\lambda$  のポアソン到着し、 $i$  番目の装置のサービスは  $\mu_i$  をパラメータとする指数分布にしたがうものとし、 $\sigma_{ij}$  は  $i$  番目の処理が  $j$  番目に先行する確率とする。0 番目は実行待行列、1 から  $K$  までは実行時段階であり、

特に1の場合はCPU, 2からKまではI/Oとする。どのI/Oからもジョブの終了は有り得るものとする。ジョブの実行終了後は、出力待行列(K+1)に入る。 $n_i$ を*i*番目の待行列長とすると文献2)で扱われたのは、実行時の多重度に制限を加えない場合であり、その平衡確率  $R(n_0, n_1, \dots, n_{K+1})$  は、次の形を持つ。

$$R(n_0, n_1, \dots, n_{K+1}) = \prod_{i=0}^{K+1} (1-\rho_i)^{K+1} \cdot \prod_{i=0}^{K+1} \rho_i^{n_i} \quad (31)$$

ここで  $\rho_i = \lambda_i / \mu_i$  とし、 $\lambda_i$  は各待行列長への相対入力率で、各待行列間でバランスしている。

$$\lambda_j = \lambda_0 \delta_{j0} + \sum_{i=0}^{K+1} \lambda_i \sigma_{ij}, \quad j=1, 2, \dots, K+1, \quad (32)$$

$$\delta_{j0} = \begin{cases} 1, & j=0, \\ 0, & j>0. \end{cases}$$

実行時の多重度に次の制限の加えた場合には、

$$n_1 + n_2 + \dots + n_K \leq N.$$

近似モデルにより解  $P(n_0, n_1, \dots, n_{K+1})$  を得る。

$$\begin{aligned} P(n_0, n_1, \dots, n_{K+1}) &= P_r[X_0=n_0, X_{K+1}=n_{K+1}, \text{多重度}=n] \\ &\times P_r[X_1=n_1, X_2=n_2, \dots, X_K=n_K | X_1 \\ &+ X_2 + \dots + X_K = n] \\ &= P_1(n_0, n_{K+1}, n) P_2(n_1, n_2, \dots, n_K) \\ &\quad / \sum_{x_1+x_2+\dots+x_K=n} P_2(x_1, x_2, \dots, x_K) \end{aligned}$$

ただし  $n = n_1 + n_2 + \dots + n_K, n < N$  のとき  $n_0 = 0$ 。 (33)

このモデルの平衡確率も以下に示すような等比数列の積の形をなす(導出については付録Aを参照)。

$$P(n_0, n_1, \dots, n_{K+1}) = \rho_0^{n_0} \rho_1^{n_1} \dots \rho_{K+1}^{n_{K+1}} P_0. \quad (34)$$

ここで  $\rho_i = \lambda_i / \mu_i$ , 各  $\lambda_i$  および  $\mu_0$  は以下の関係式から得られる値である。

$$\lambda_0 = \lambda_{K+1} = \lambda = \sum_{i=2}^K (1-\sigma_{i1}) \lambda_i,$$

$$\lambda_1 = \sum_{i=2}^K \lambda_i, \quad \lambda_i = \sigma_{i1} \lambda_1. \quad (35)$$

$\mu_0^{(n)}$  を多重度が  $n$  の場合のジョブの終了率とする。

$$\mu_0^{(n)} = \sum_{i=2}^K P_2^{(n,i)} (1-\sigma_{i1}) \mu_i. \quad (36)$$

特に  $\mu_0 = \mu_0^{(N)}$  とおく。ここで  $P_2^{(n,i)}$  はI/O装置*i*の使用率を示し

$$P_2^{(n,i)} = P_r \left[ x_i > 0 \mid \sum_{i=1}^K x_i = n \right]. \quad (37)$$

### 3.3 ジョブ・クラスを持つオープンバッチ、システム(M-4モデル)

Fig. 5は、A, B, Cの3つのジョブ・クラスを持つシステムを表現している。ジョブは各クラスについて

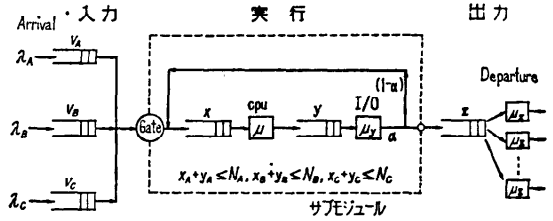


Fig. 5 Model with job classes (M-4 model).

独立にそれぞれ入力率  $\lambda_A, \lambda_B, \lambda_C$  のポアソン到着をする。 $v_A, v_B, v_C$  は各ジョブ・クラスの実行待行列長とする。ジョブの実行時は M-1 と同じサブモジュールによるものとする。CPU と I/O は各クラスに共通のサービス率  $\mu_x, \mu_y$  を持つ指数分布にしたがって処理されるものとし、I/O 終了後、各クラスが異なる確率  $\alpha_A, \alpha_B, \alpha_C$  で実行を終了する。もし A, B, C が小中、大のジョブと対応しているなら  $\alpha_A > \alpha_B > \alpha_C$  となっている。

実行時の待行列  $x, y$  には、各クラスのジョブが実行多重度に比例して乗っているものとする。実行終了後は共通に1つの出力待行列に入り、*L* 台の出力装置を使って出力される。

ジョブの起動時は、各ジョブ・クラスに対して多重度の制限があり、その値を  $N_A, N_B, N_C$  とする。 $x_A, x_B, x_C$  および  $y_A, y_B, y_C$  をそれぞれ A, B, C のジョブ・クラスに対するCPUおよびI/O待行列長とする。

$$\begin{aligned} x &= x_A + x_B + x_C, \quad y = y_A + y_B + y_C, \\ n_B &= x_B + y_B \leq N_B, \quad (\beta = A, B \text{ または } C), \\ x_B &= n_B x / n, \quad y_B = n_B y / n, \end{aligned}$$

および

$$n = n_A + n_B + n_C, \quad N = N_A + N_B + N_C. \quad (38)$$

このモデルの解は以下のようにして求められる。

$$\begin{aligned} P(v_A, v_B, v_C, n_A, n_B, n_C, a, b, z) &= P_r[V_A=v_A, V_B=v_B, V_C=v_C, M_A=n_A, \\ &M_B=n_B, M_C=n_C, Z=z] P_r[X=a, Y=b, \\ &M_A=n_A, M_B=n_B, M_C=n_C | X+Y=n] \\ &= P_1(v_A, v_B, v_C, n_A, n_B, n_C, z) \end{aligned}$$

$$\times \frac{P_2(a, b, n_A, n_B, n_C)}{\sum_{y=0}^n P_2(n-y, y, n_A, n_B, n_C)}$$

ただし  $x+y=n$  であり、 $n_B < N_B$  のとき  $v_B=0$ 。 (39)

同様の解析手法により

$$\begin{aligned} P(v_A, v_B, v_C, n_A, n_B, n_C, a, b, z) &= \{ \rho_A / P_2^{(n_A, N_A)} \}^{n_A} \{ \rho_B / P_2^{(n_B, N_B)} \}^{n_B} \end{aligned}$$

$$\times \{ \rho_C / P_2(N_C, N) \}^{v_C} \left( \prod_{j=1}^x \rho_A^{(j)} \right) \rho_A^{n_A} \rho_B^{n_B} \rho_C^{n_C}$$

$$\times n! P_0 \{ \delta_1^{n_A} n_A! n_B! n_C! \}. \quad (40)$$

ここで  $\rho_\beta = \lambda_\beta / (\alpha_\beta \mu_\beta)$ ,  $\delta_1 = \mu_x / \mu_y$ ,  $\rho_A^{(j)} = (\lambda_A + \lambda_B + \lambda_C) / (l_j \mu_x)$  で  $P_2^{(n_A, n_B)}$  は実行多重度が  $n$  でジョブ・クラス  $\beta$  の多重度が  $n_\beta$  の場合の I/O の使用率を示す。

$$P_2^{(n_A, n_B)} = P_r [ y > 0 | x + y = n, x_\beta + y_\beta = n_\beta ]$$

$$= n_\beta \delta_1 (1 - \delta_1^{n_A}) / \{ n(1 - \delta_1^{n_A+1}) \}. \quad (41)$$

#### 4. おわりに

M-1 モデルに対して、本来のモデルの解は積の形とならないことを示したが、この場合の解を式で求めることは逆マトリックスを求める段階で式が複雑になり成功しなかった。

原研の FACOM 230-75 システムにおける実測値との比較検討を行うべきであるが、ここで取上げたモデルは、次の点でまだ不充分である。原研のシステムでは、ジョブ・クラスの分け方を、ジョブの大、小の他に、事務計算、磁気テープの使用が目的としたセミ・オープン、長時間でも優先して行うジョブ・クラスなどあり、クラス内のジョブの性質の特徴づけが困難である。また、小さいジョブに対するジョブ・クラス A は、空時間の有効利用のために無制限ジョブ・クラスと定義され、全体の多重度が  $N$  以下のときは、クラス A の多重度  $N_A$  を越えてジョブが起動される。この点をモデル化すると積の形の解での近似が困難となる。

平衡確率分布がこのように、等比数列または、有限乗積の積で表現されると、種々の解析に要する時間がシミュレーションなどとは較べものにならない程少なくて済む。FACOM 230-75 では数秒で終る。

#### 参考文献

- 1) 石黒美佐子：オープンバッチ・多重ジョブ処理システムの待行列モデルの解析，情報処理，Vol. 18, No. 5 (1977)。
- 2) H. Kobayashi: Some Recent Progress in Analytic Studies of System Performance, IBM Research Report (1972)。
- 3) M. Reiser and H. Kobayashi: Queuing Network with Multiple Closed Chains, IBM J. Res. and Dev. Vol. 19, No. 3, pp. 283~294 (1975)。
- 4) M. A. Leibowitz: An Approximate Method for Treating a Class of Multiple Queue Problem,

IBM J. Res. and Dev. Vol. 5, pp. 204~209 (1961)。

- 5) U. Herzog, L. Woo and K. M. Chandy: Solution of Queuing Problems by a Recursive Techniques, IBM J. Res. and Dev. Vol. 19, No. 3, pp. 295~300 (1975)。

#### 付録 A

(37)式より  $\delta_i = \mu_i \sigma_{1i} / \mu_i$  ( $2 \leq i \leq K$ ) とおくと、

$$P_2^{(n, i)} = \sum_{x_i=1}^n \frac{\sum_{x_1+x_2+\dots+x_{i-1}+x_{i+1}+\dots+x_K=n-x_i} P_2(x_1, x_2, \dots, x_K)}{\sum_{x_1+x_2+\dots+x_K=n} P_2(x_1, x_2, \dots, x_K)}$$

$$= \frac{\delta_i \rho_i}{1 - P_1^{(n)}}$$

$$= \frac{\sum_{x_1+x_2+\dots+x_K=n-1} P_2(x_1, x_2, \dots, x_K)}{\sum_{x_1+x_2+\dots+x_K=n} P_2(x_1, x_2, \dots, x_K)}. \quad (A-1)$$

ここで

$$P_1^{(n)} = \begin{cases} P_r [ n_0 > 0 | n_1 + n_2 + \dots + n_K = N ], & n = N, \\ 0, & n < N. \end{cases}$$

(36)式から

$$\mu_0^{(n)} = \frac{\sum_{x_1+x_2+\dots+x_K=n-1} P_2(x_1, x_2, \dots, x_K)}{\sum_{x_1+x_2+\dots+x_K=n} P_2(x_1, x_2, \dots, x_K)}$$

$$= \sum_{i=0}^K \frac{\delta_i \rho_i (1 - \sigma_{1i}) \mu_i}{1 - P_1^{(n)}}. \quad (A-2)$$

ところが、 $\rho_i = \lambda_i / \mu_i$ ,  $\delta_i = \mu_i \sigma_{1i} / \mu_i$  および (35) の関係から、

$$\sum_{i=2}^N \delta_i \rho_i (1 - \sigma_{1i}) \mu_i = \lambda, \quad (A-3)$$

同様にして、

$$P_1(n_0, n_{K+1}, n) = \rho_0^{n_0} \rho_{K+1}^{n_{K+1}} P_1(0, 0, n), \quad (A-4)$$

$$P_1(0, 0, n) = \lambda^n P_0 / (\mu_0^{(n)} \mu_0^{(n-1)} \dots \mu_0^{(1)})$$

$$= (1 - P_1^{(n)})$$

$$\sum_{x_1+x_2+\dots+x_K=n} P_2(x_1, x_2, \dots, x_K). \quad (A-5)$$

一方、 $P_2(n_1, n_2, \dots, n_K)$

$$= \delta_2^{n_2} \delta_3^{n_3} \dots \delta_K^{n_K} P_2(n, 0, 0, \dots, 0)$$

$$= \rho_1^{n_1} \delta_2^{n_2} \delta_3^{n_3} \dots \delta_K^{n_K} P_0 / (1 - P_1^{(n)}). \quad (A-6)$$

故に式(A-4, 5, 6)と(33)から、次の結果を得る。

$$P(n_0, n_1, \dots, n_{K+1}) = \rho_0^{n_0} \rho_1^{n_1} \dots \rho_K^{n_K} P_0. \quad (A-7)$$

(昭和51年12月10日受付)

(昭和52年4月8日再受付)