マルチプロセッサシステムにおける主記憶最適配分問題の解析†

坂　田　真　人††

最近の大型計算機システムは仮想記憶を採用し，マルチ・プロセッサ構成とすることが多い．普通，多重プログラミングの実行環境のもとで複数個のバッチジョブとTSSジョブがCPU，主記憶などのシステムリソースを共用する形態で処理され，システムのスループットやTSSのレスポンスはこの仮想記憶の制御方式に著しい影響を受ける．すなわち，主記憶を処理中のプログラムにどのように配分するのが最適かという問題が提起される．本論文では，主記憶割当て方式の単純な一方式である固定割当て法について，マルチ・プロセッサ環境下で検討するため，計算機システムを BCMP 型の閉じた待ち行列網にモデル化して議論している．従来，シングル・プロセッサ・モデルでは，主記憶をおのおののプログラムに均等に配分するかあるいは極端に偏って配分するかのいずれかの方策を採れば，最適となることが明らかになっていた．本論文ではこの結果がマルチ・プロセッサ環境ならびにより一般化したライフタイム関数の場合にも成立することを示した．

1. まえがき

近年，おのおのの大型計算機システムは主記憶として数十Mバイトの容量を実装するようになってきたが，このような状況においても実装容量を超える大規模プログラムの処理が要求されている．一方，TSS (タイムシェアリング・システム) による利用が通常化して，同時に多数の利用者プログラムを処理することが必須要件となっている．ページ方式による現実の仮想記憶システムは多重プログラミングの環境下で動作するものであり，主記憶制御方式はシステム全体の性能にきわめて大きな影響を与えることから，理論的・実験的立場から種々の議論がなされてきた[1),2)]．すなわち，プログラムの多重度をいくつにするか，また，個々のプログラムにどのように主記憶を割り当てるかなどの課題である．しかし，現在のところ，決定的な解を得るにいたっていない．この困難さは，現実に処理されているプログラムの動作特性が多様であること，システムの負荷が常時変動すること，システム構成や利用環境が個々に異なることなどに起因する．

本論文では，主記憶制御方式の一つである固定割当て法について待ち行列モデルに基づき理論的立場から考察する．これまで，待ち行列モデルに基づく主記憶制御方式の固定割当て法の解析は，Spirn[3)], Ghanem[4)], Fin と亀田[5)]によってなされてきたが，いずれもシングル・プロセッサ (CPU) のモデルを取り扱っている．本論文ではモデルをマルチ・プロセッサ・システムへ拡張して解析する．さらに，ライフタイム (lifetime) 関数についてもより一般化したもので議論する．

取り扱っているモデルは仮想記憶システムを単純化したものであり，主記憶の配分問題のみの検討である．そこで，実際の主記憶制御方式にそのまま適用できないが，得られた一般的な結果はオペレーティング・システムの主記憶制御プログラムの設計に際しての基礎的な知見を与えるものである．

2. ライフタイム関数

ページ方式の主記憶制御方式が採用されているシステム環境下において，主記憶量としてmページが割り当てられているプログラムに注目する．ライフタイム関数はこのプログラムがページ・フォールト (page fault) やファイル I/O などで CPU の実行を中断するまでの CPU のサービス時間の平均と定義される．ここに，ライフタイム関数を変数 m の関数とみなして $L(m)$ と表せば，ページ置換えアルゴリズムに依存してその形は異なるが，普通，$L(0)=0$，m については単調増加，$\lim_{m\to\infty} L(m)=L_\infty<\infty$ なる性質をもつと考えられる．さらに，関数 $L(m)$ は，**図1**に示すごとく，ある程度の主記憶を割り当てるまでの範囲(図1の $0\leq m\leq m_a$)では下に凸であるが，そこから先では上に凸なる特性を示すことが多いと指摘されている[6)]．

ライフタイム関数の全域にわたって検討するため，その関数を微係数が不連続的に変化する場合をも含むところの次の一般的な関数で近似して議論を進める．ただし，各プログラムのライフタイム関数は同一とす

† Analysis of Memory Fartitioning for Multiprogramming-Multiprocessor Computer Systems by MASATO SAKATA (Computer Center, Tohoku University).

†† 東北大学大型計算機センター

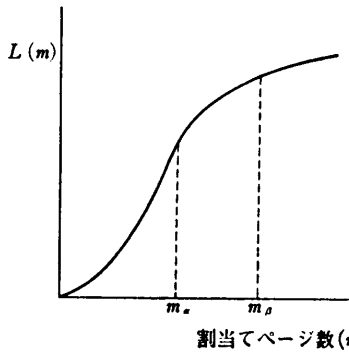

図 1 ライフタイム関数
Fig. 1 Lifetime function.

る.

(i) $0 \leqq m \leqq m_\alpha$ なる範囲では次の下に凸なる関数

$$L(m)=\beta m^\alpha \tag{2.1}$$

ここで, α は $\alpha>1$, β は $\beta>0$ なる定数とする.

なお, β はプログラムの特性や計算機の処理能力に依存する係数であるが, α は比較的これらに独立で2に近い値をとることが実験的に確かめられている[6].

(ii) $m_\alpha \leqq m<\infty$ なる範囲では上に凸なる関数.

なお, $m_\alpha \leqq m<\infty$ で任意の $\delta(>0)$ に対し

$$L(m+\delta)-L(m) \leqq \delta\beta \tag{2.2}$$

を満たす最小の m を m_β で表す. ここに, $m \geqq m_\beta$ なる m は式(2.2)を満たすことは明らかである.

Ghanem および Fin と亀田のモデルで取り扱っているライフタイム関数は, (i)の設定は同じであるが, (ii)の設定で $L(m)=$一定 (Ghanem) または $m_\alpha=m_\beta$ (Fin と亀田) と限定されている. これらの近似ではライフタイム関数は下に凸から上に凸に移る点 m_α でその微係数が不連続となり, 実際のライフタイム関数[6]の全域的近似としては不十分であった. 本論文では上に凸なる範囲と下に凸なる範囲におのおの主記憶が割り当てられているところの複数のプログラム相互間の主記憶配分をも取り扱うため, 点 m_α において微係数が連続な場合をも含むところのより一般的なライフタイム関数を仮定して議論する.

3. 待ち行列網モデルの設定

複数台の CPU といくつかの I/O ステーションからなる計算機システムを**図 2**に示すような BCMP 型[7]の閉じた待ち行列網 (closed queuing network) でモデル化する. ここで, CPU ステーションのサービス規律 (service discipline) は PS (Processor Sharing), 個々の I/O ステーションのサービス規律は FCFS

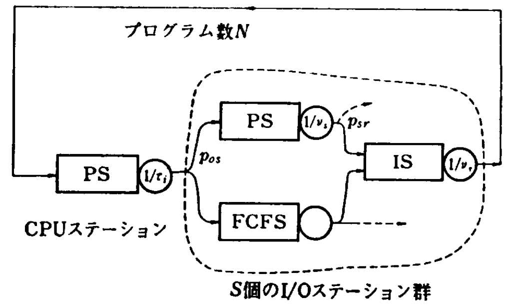

図 2 閉じた待ち行列網モデル
Fig. 2 Closed queuing network model.

(First-Come First-Served), PS, IS (Infinite Server) または LCFS-PR (Last-Come First-Served Preemptive-Resume) のいずれかとする. また, おのおののプログラムは CPU または I/O ステーション間をマルコフ連鎖に従って順次推移しながら処理されるものとする.

以下, 次の記法を採用する.

N: プログラムの数 (=多重度)

M: 主記憶の総ページ数

m_i: プログラム i に割り当てられている主記憶のページ数 $\left(\sum_{i=1}^{N} m_i \leqq M\right)$

C: CPU ステーションのプロセッサ台数

$1/\tau_i$: CPU ステーションでプログラム i が毎回要求するサービス時間の平均 (ライフタイム関数 $L(m_i)$ に等しい); サービス時間は Coxian[8] (分布のラプラス・スティルチェス変換が有理関数で表現可能) に従う.

S: I/O ステーションの数

C_s: I/O ステーション s の装置台数

$1/\nu_s$: I/O ステーション s でプログラムが毎回要求するサービス時間の平均 ($1/\nu_s>0$); サービス時間は FCFS のステーションでは指数分布, PS・IS・LCFS-PR のステーションでは Coxian に従い, 個々のプログラムには依存しないものとする.

$\mu_s(k)$: k 個のプログラムがステーション s ($s=0, 1, \cdots, S$; $s=0$ は CPU ステーションを示し, $C_0=C$ とする) に存在するときの処理能力比; おのおののステーションでの処理にオーバヘッドがないと仮定すれば, 次式で表される.

$$\mu_s(k)=\begin{cases} k & (k<C_s) \\ C_s & (k \geqq C_s) \end{cases} \tag{3.1}$$

p_{sr}：ステーション s での処理を終えたプログラムが次にステーション r に移る推移確率；この確率はシステムの状態に依存しないものとする．さらに，一般性を失うことなく，$p_{00}=0$，また，任意の r に対して $\sum_{s=0}^{S} p_{sr}>0$（既約なること）を仮定する．

ステーション間の推移確率はすべてのプログラムについて同一と仮定したので，あるプログラムがステーション s の処理を要求する単位時間当りの回数（スループット）θ_s は，マルコフ連鎖の性質に基づき，平衡状態では，次の連立方程式の解として与えられる．

$$\theta_s=\sum_{r=0}^{S}\theta_r p_{rs} \qquad (s=0, 1, \cdots, S) \tag{3.2}$$

この方程式の解は定数倍を除いて一意に定まる．そこで，$\theta_0=1$（CPU ステーションのスループットで正規化）として求めた相対的なスループット θ_s を用いて，プログラム i の CPU ステーションにおけるトラヒック密度 ρ_i，および任意のプログラムの I/O ステーション s におけるトラヒック密度 σ_s は次式で表される．

$$\rho_i=1/\tau_i=L(m_i) \tag{3.3}$$

$$\sigma_s=\theta_s/\nu_s \qquad (s=1, 2, \cdots, S) \tag{3.4}$$

ステーション s に存在するプログラム i の数を k_{is} とし，ステーション s の状態は $\boldsymbol{k}_s=(k_{1s}, k_{2s}, \cdots, k_{Ns})$，網全体の状態は $\bar{\boldsymbol{k}}=(\boldsymbol{k}_0, \boldsymbol{k}_1, \cdots, \boldsymbol{k}_S)$ で表す．ここで，k_{is} はプログラム i がステーション s に存在すれば1を，存在しなければ0をとる変数で，任意の i に関して $\sum_{s=0}^{S} k_{is}=1$ を満たさねばならない．

BMCP 型待ち行列網の設定に基づき，平衡状態における状態確率 $\pi(\bar{\boldsymbol{k}})$ は次の積形式で表される[7]．

$$\pi(\bar{\boldsymbol{k}})=\frac{1}{G}\pi_0(\boldsymbol{k}_0)\pi_1(\boldsymbol{k}_1)\cdots\pi_S(\boldsymbol{k}_S) \tag{3.5}$$

ここで，

$$\pi_0(\boldsymbol{k}_0)=a_0(|\boldsymbol{k}_0|)\rho_1^{k_{10}}\rho_2^{k_{20}}\cdots\rho_N^{k_{N0}} \tag{3.6}$$

$$\pi_s(\boldsymbol{k}_s)=a_s(|\boldsymbol{k}_s|)\sigma_s^{|\boldsymbol{k}_s|} \quad (s=1, 2, \cdots, S) \tag{3.7}$$

$$|\boldsymbol{k}_s|=k_{1s}+k_{2s}+\cdots+k_{Ns} \tag{3.8}$$

$$a_s(n)=\begin{cases}1 & (n=0)\\ n!\prod_{k=1}^{n}(1/\mu_s(k)) & (n>0)\end{cases} \tag{3.9}$$

G は正規化定数．

CPU ステーションの処理ではオーバヘッドがないものと仮定し，CPU ステーションに存在するプログラム数のみに着目すれば次の結果が得られる．

［定理 1］ CPU ステーションに n 個のプログラムが存在する確率 $p(n)$ は次式で求められる．

$$p(n)=\frac{1}{G}a(n)h(N-n)v_N(n) \tag{3.10}$$

ここで，

$$a(n)=a_0(n)=\begin{cases}1 & (n\leqq C)\\ n!/(C!C^{n-C}) & (n>C)\end{cases} \tag{3.11}$$

$$h(l)=\sum_{r_1+r_2+\cdots+r_S=l}\frac{l!}{r_1!r_2!\cdots r_S!}\prod_{s=1}^{S}a_s(r_s)\sigma_s^{r_s} \tag{3.12}$$

$v_N(n)$ は N 変数 $\rho_1, \rho_2, \cdots, \rho_N$ からなる n 次の基本対称式で

$$v_N(n)=\sum_{k_{10}+k_{20}+\cdots+k_{N0}=n}\rho_1^{k_{10}}\rho_2^{k_{20}}\cdots\rho_N^{k_{N0}} \tag{3.13}$$

G は正規化定数で

$$G=\sum_{n=0}^{N}a(n)h(N-n)v_N(n). \tag{3.14}$$

［証明］ Fin と亀田[5] の Theorem 1 の証明と同様であり，付録1に示す．□

式(3.10)を用いて，CPU の使用率 U は

$$U=\sum_{n=1}^{C-1}\frac{n}{C}p(n)+\sum_{n=C}^{N}p(n)$$

$$=1-\frac{1}{C}\sum_{n=0}^{C-1}\sum_{k=0}^{n}p(k) \tag{3.15}$$

と計算される．特別な場合として，$C=1$ とすれば，

$$U=1-p(0)=1-h(N)/G \tag{3.16}$$

を得る．Ghanem および Fin と亀田はこの式(3.16)の最大問題（G についての最大問題）を解いている．次章以降で，式(3.15)の最大問題を議論する．

4. 問題の定式化

計算機システムの処理能力は単位時間当りに処理するプログラム（ジョブ）数で評価することができる．処理すべきプログラムが十分多く，多重度 N の多重プログラミングで処理している状況のもとで，あるプログラムの処理が完了したときにそのプログラムに割り当てていた主記憶を待合せ中のプログラムに引き継いで処理するモデルを考えれば，単位時間当りの処理プログラム数は CPU の使用率に比例し，3 章の閉じた待ち行列網のモデルと等価である．

そこで，固定割当て法の最適問題はおのおののプログラムに割り当てる（配分する）主記憶量 m_i $(i=1, 2, \cdots, N)$ を変数として CPU の使用率 U を目的関数とする次の最大問題を解くことに帰着させて考える．

$$\begin{cases} U=F/G\to\max & (4.1)\\ F=\sum_{n=1}^{C-1}\frac{n}{C}a(n)h(N-n)v_N(n) \\ \quad +\sum_{n=C}^{N}a(n)h(N-n)v_N(n) & (4.2)\\ G=\sum_{n=0}^{N}a(n)h(N-n)v_N(n) & (4.3)\\ \rho_i=L(m_i) & (4.4)\\ \sum_{i=1}^{N}m_i\leqq M & (4.5)\\ m_i\geq 0\quad (i=1,2,\cdots,N;\ N\geq 2) & (4.6)\end{cases}$$

ここで，関数 $a(n)$, $h(l)$ および $v_N(n)$ はそれぞれ式(3.11), (3.12) および (3.13) で定義されている．

ここで，変数 m_i は割当てページ数であるから，厳密には整数の値を取るべきであるが，解析ならびに結果を簡潔にするため連続量（実数）とみなして以下の解析を進める．

この最大問題（最適配分）を解くための前準備として，CPU の使用率 U の性質を明らかにする．つまり，U を ρ_i $(i=1,2,\cdots,N)$ の関数とみなせば次の結果が得られる．

［定理 2］ CPU の使用率 U は任意の ρ_i に関して，狭義の単調増加関数で上に凸である（証明は付録2に示す）．□

この定理から CPU の使用率 U を最大にする配分は式 (4.5) で等号が成立する場合を調べればよいことがわかる．

CPU の使用率 U を2変数 ρ_i, ρ_j の関数とみなせば，式(4.2), (4.3)の F, G は ρ_i, ρ_j に関してたかだか一次の項からなるので，

$$U(\rho_i,\ \rho_j)=\frac{F_0+F_1(\rho_i+\rho_j)+F_2\rho_i\rho_j}{G_0+G_1(\rho_i+\rho_j)+G_2\rho_i\rho_j} \qquad (4.7)$$

と書き直せる．ここで，F_0, F_1, F_2, G_0, G_1 および G_2 は変数 ρ_i, ρ_j を含まない係数で，係数間に次の性質がある．

［定理 3］ 式(4.7)の係数 F_0, F_1, F_2, G_0, G_1 および G_2 には

$$F_0/G_0<F_1/G_1\leqq F_2/G_2 \qquad (4.8)$$

なる関係式が成り立つ．ただし，$C=1$ の場合は $F_1/G_1=F_2/G_2$, $C>1$ の場合は $F_1/G_1<F_2/G_2$.

［証明］ 定理2から，$\rho_i<\rho_i'$, $\rho_j<\rho_j'$ ならば，$U(\rho_i,\ \rho_j)<U(\rho_i',\ \rho_j')$ なる関係が成り立つ．一方，式 (4.7) から，$F_0/G_0=U(0,0)$, $F_1/G_1=\lim_{\rho\to\infty}U(\rho,0)$, $F_2/G_2=\lim_{\rho\to\infty}U(\rho,\rho)$ となる．そこで，$F_0/G_0<F_1/G_1$, $F_1/G_1\leqq F_2/G_2$ は明らかである．さらに，$C=1$ ならば，$\lim_{\rho\to\infty}U(\rho,0)=1$ より等号が成り立ち，$C>1$ ならば，$\lim_{\rho\to\infty}U(\rho,0)<1$ より不等号が成り立つ．□

5. 最適配分

N個のプログラムのうちから，まず，番号1と2の二つのプログラムを選び，この二つのプログラム間の主記憶配分問題を考える．注目しているプログラムの主記憶量を m_1, m_2 と表し，その和 $M_{12}(=m_1+m_2)$ を固定する．このとき，CPU の使用率を $U(m_1,\ m_2)$ で表すことにすれば，式 (4.7) は

$$U(m_1,\ m_2)=\frac{F_0+F_1y+F_2z}{G_0+G_1y+G_2z} \qquad (5.1)$$

と書き直すことができる．ここで，

$$y=L(m_1)+L(m_2),\ z=L(m_1)L(m_2) \qquad (5.2)$$

二つのプログラム間の配分に関して，ライフタイム関数 $L(m)$ が上に凸なる区間あるいは下に凸なる区間で以下の定理に示す結果が得られる．

［定理 4］（上に凸なる区間） $m_1\geq m_a$, $m_2\geq m_a$ なる条件のもとで，$M_{12}=m_1+m_2=m_1^*+m_2^*$ かつ $m_1>m_1^*\geqq m_2^*>m_2$ ならば，

$$U(m_1,\ m_2)\leqq U(m_1^*,\ m_2^*) \qquad (5.3)$$

なる関係が成り立ち，$m_1^*=m_2^*=M_{12}/2$ が最適配分となる．

［証明］ ライフタイム関数は区間 $[m_a,\infty)$ で上に凸なることから，$y^*=L(m_1^*)+L(m_2^*)$ とおけば，$y\leqq y^*$ は明らか．$L(m_1)+L(m_2)\leqq L(m_1^*)+L(m_2^*)$ が成り立てば，この不等式の両辺から $L(m_2)$ を引き，ついで，$L(m_2)$ を掛けた不等式を整理すれば，

$$\begin{aligned}&L(m_1)L(m_2)\\ &\quad\leqq L(m_1^*)L(m_2^*)-(L(m_2^*)-L(m_2))(L(m_1^*)-L(m_2))\\ &\quad\leqq L(m_1^*)L(m_2^*)\end{aligned}$$

よって，$z\leqq z^*$ が成り立つ．そこで，定理3の関係式から式 (5.3) が導ける．ここで，式 (5.3) で等号は，$y=y^*$, $z=z^*$ なる場合，つまり，$m\geqq m_2$ において $L(m)$ が一定となる場合に限る．$m_1^*=m_2^*=M_{12}/2$ が最適なることは明らか．□

［定理 5］（下に凸なる区間） $m_1\leqq m_a$, $m_2\leqq m_a$ なる条件のもとでの最適配分は，$m_1\geqq m_2$ とすれば，次のいずれかの場合である．

（i） $m_1=m_2=M_{12}/2$

（ii） $m_1=\min(m_a,\ M_{12})$, $m_2=M_{12}-m_1$

［証明］ 式の取扱いを簡単化するため，$m_1=(1+x)M_{12}/2$, $m_2=(1-x)M_{12}/2$ $(0\leqq x\leqq\min(1, 2m_\alpha/M_{12}-1))$ とおいて，式 (2.1) のライフタイム関数の定義式に代入し，式 (5.2) を整理すれば

$$y=A\{(1+x)^\alpha+(1-x)^\alpha\},\ z=A^2(1-x^2)^\alpha \quad (5.4)$$

となる．ただし，$A=\beta(M_{12}/2)^\alpha$. CPU の使用率 $U(x)$ の x に関する増減を調べるため，式(5.4)を式(5.1)に代入して，微係数を求めれば次の式を得る．

$$\frac{dU(x)}{dx}=\frac{\alpha A(1-x^2)^{\alpha-1}\psi(x)}{(G_0+G_1y+G_2z)^2} \quad (5.5)$$

$$\left.\frac{d^2U(x)}{dx^2}\right|_{x=0}=\frac{\alpha A\psi'(0)}{(G_0+G_1y+G_2z)^2} \quad (5.6)$$

ここで，関数 $\psi(x)$ の導出は付録 3 に示すが，関数 $\psi(x)$ は区間 $[0, 1)$ で次の性質をもつ．

（a） $\psi(x)$ は連続で，$\psi(0)=0$, $\lim_{x\to 1}\psi(x)=\infty$

（b） $\psi'(x)$ は連続で，$\lim_{x\to 1}\psi'(x)=\infty$，また，ある x^* で $\psi'(x^*)\geqq 0$ ならば，区間 $[x^*, 1)$ で狭義の単調増加

そこで，$\psi(x)$ の正負を次の二つの場合で調べる．

（1） $\psi'(0)\geqq 0$ の場合：性質（b）より，$\psi'(x)>0$ $(0<x<1)$，ゆえに，$\psi(x)=\int_0^x\psi'(x)dx+\psi(0)>0$ $(0<x<1)$ となる．$d^2U(x)/dx^2|_{x=0}\geqq 0$ であるから，$U(x)$ は $x=0$ で極小値（最小値）をとり，$x=\min(1, 2m_\alpha/M_{12}-1)$ で最大値をとる．

（2） $\psi'(0)<0$ の場合：性質（b）より，$\psi'(x)=0$ なる根は区間$(0, 1)$でただ一つ存在する．この根を x^* とすれば，区間 $(0, x^*)$ で $\psi'(x)<0$, $\psi(x)<0$ となる．また，区間 $(x^*, 1)$ で $\psi'(x)>0$，つまり，$\psi(x)$ は狭義の単調増加となる．よって，$x^*<x<1$ で $\psi(x)=0$ なる根はただ一つ存在し，この根を x_0 とすれば，

$x=0$ では極大値 $(d^2U(x)/dx^2|_{x=0}<0)$

$0<x<x_0$ で $dU(x)/dx<0$

$x=x_0$ で $dU(x)/dx=0$（極小値）

$x_0<x\leqq 1$ で $dU(x)/dx>0$

となるから，$U(x)$ は $x=0$ または $x=\min(1, 2m_\alpha/M_{12}-1)$ で最大値をとる．□

［定理 6］

（i） 条件 $m_1>m_\beta$, $m_\beta>m_2\geqq 1$ を満たす配分は最適配分ではない．

（ii） 条件 $m_1>m_\beta+1$, $m_\beta>m_2\geqq 0$ を満たす配分は最適配分ではない．

［証明］ m_2 が $m_\beta>m_2\geqq m_\alpha$ ならば，最適配分でないことは定理 4 より明らかである．次に $m_\alpha>m_2$ の場合を示す．まず，（i）の場合には，

$0<\delta\leqq\min(m_1-m_\beta, m_\alpha-m_2)$

とすれば，$m_1>m_2\geqq 1$ より

$L(m_2+\delta)-L(m_2)=\beta((m_2+\delta)^\alpha-m_2{}^\alpha)>\beta\delta$

（ii）の場合には，

$0<\delta\leqq\min(m_1-m_\beta, m_\alpha-m_2)$

かつ $1<m_2+\delta$ とすれば，$m_2<1$ としても，

$$\begin{aligned}&L(m_2+\delta)-L(m_2)\\&\quad=\{L(m_2+\delta)-L(1)\}+\{L(1)-L(m_2)\}\\&\quad>\beta(m_2+\delta-1)+\beta(1-m_2)\end{aligned}$$

となるから，

$L(m_2+\delta)-L(m_2)>\beta\delta$

一方，式(2.2)の定義より，

$L(m_1)-L(m_1-\delta)\leqq\beta\delta$

ゆえに，$m_1^*=m_1-\delta$, $m_2^*=m_2+\delta$ とすれば，

$L(m_1)+L(m_2)<L(m_1^*)+L(m_2^*)$

を得る．そこで，定理 4 の証明と同様な手順で

$$U(m_1, m_2)<U(m_1^*, m_2^*) \quad (5.7)$$

なる関係式が導け，命題が成り立つ．□

定理 5 の証明の式(5.7)を用いれば，$m_1+m_2(=M_{12})\geqq m_\alpha+m_\beta$ ならば，$m_2<m_\alpha$ としても

$$U(m_1, m_2)<U(M_{12}-m_\alpha, m_\alpha)\leqq U(M_{12}/2, M_{12}/2) \quad (5.8)$$

なる関係式を得る．以上の定理 4〜6 は Fin と亀田の Theorem 3 を $C>1$ へ拡張したものとなっている．

一般の N 個のプログラム間の配分については次の結果が得られる．

［定理 7］ 条件 $M\geqq N(m_\beta+1)$ を満たせば，N 個のプログラムの割当て主記憶量 m_i $(i=1, 2, \cdots, N)$ における最適配分は

$$m_1=m_2=\cdots=m_N=M/N \quad (5.9)$$

である．

［証明］ ある主記憶配分 $(m_1, m_2, \cdots, m_N)$ が与えられれば，まず，$m_1\geqq m_2\geqq\cdots\geqq m_N$ となるように番号をふり直す．ついで，$m_1\neq m_N$ ならば，$m_1>M/N\geqq m_\beta+1$ であるから，定理 6 の証明の式 (5.7) および式 (5.3) を用いて，$m_1^*=M/N$, $m_N^*=m_1+m_N-M/N$ とすれば（$m_1+m_N>2M/N$ ならば，$m_1^*=m_1+m_N-M/N$, $m_N^*=M/N$ とする），$U(m_1, m_2, \cdots, m_N)\leqq U(m_1^*, m_2, \cdots, m_N^*)$ が成り立ち，さらに，$m_i=M/N$ なるプログラム数が 1 個以上増加する．そこで，この手順をたかだか $N-1$ 回繰り返せば，$U(m_1, m_2, \cdots, m_N)\leqq U(M/N, M/N, \cdots, M/N)$ を得る．□

［定理 8］ $M<N(m_\beta+1)$，かつ，$L(m)$ は区間 $(0,$

$m_\beta+1$) で狭義の単調増加ならば，N個のプログラムの割当て主記憶量 $m_i(i=1,2,\cdots,N)$ における最適配分は，$0\leqq p\leqq q\leqq N$ なるところのある p, q が存在して次の条件を満たさねばならない．

$$\begin{cases} m_\beta+1\geqq m_1=m_2=\cdots=m_p\geqq m_\alpha \\ m_\alpha>m_{p+1}=m_{p+2}=\cdots=m_q>0 \\ \qquad m_{q+1}=m_{q+2}=\cdots=m_N=0 \end{cases} \tag{5.10}$$

ただし，$m_1\geqq m_2\geqq\cdots\geqq m_N$ を仮定する．

［証明］ 式(5.10)を満たさない最適配分 ($m_1, m_2, \cdots, m_N$) が存在するとすれば，矛盾することを示す．

まず，$m_i>m_\beta+1$ なる m_i が存在すれば，$M<N(m_\beta+1)$ から，$m_j<m_\beta+1$ なるプログラム j が少なくとも一つ存在するので，定理4または定理6の(ii)より最適配分ではない．ついで，$m_\beta+1\geqq m_i>m_j\geqq m_\alpha$ なる i, j が存在すれば，定理4から最適配分とはなりえない．同様に，$m_\alpha>m_i>m_j>0$ なる i, j が存在すれば，定理5から最適配分とはなりえない．□

定理8の特別な場合として，$M\leqq m_\alpha$，つまり，ライフタイム関数が下に凸ならば，定理2の性質を加味して，最適配分の条件は

$$\begin{cases} m_1=m_2=\cdots=m_q=M/q \\ m_{q+1}=m_{q+2}=\cdots=m_N=0 \end{cases} \tag{5.11}$$

となる．しかし，この条件は定理5の証明のところで述べたように，$\sum_{i=1}^{N} m_i=M$ なる拘束条件のもとで，極小値をとる場合がある．

6. むすび

本論文では主記憶制御方式のうちの固定割当て法について待ち行列網を用いた理論的検討を行い，マルチ・プロセッサ環境においても，従来のシングル・プロセッサ環境において得られていた結果が成り立つことを証明した．

ライフタイム関数は図1に示すごとく，下に凸なる範囲で式(2.1)の関数を仮定したが，たとえば，$L(m)=\beta(e^{\alpha m}-1)/(e^\alpha-1)$ なる関数を仮定しても，定理4～8が証明できる．しかし，関数が下に凸という条件のみでは明確な結論は得られていない．

一方，定理5，8の結論からすれば，あるプログラムの主記憶配分を零とする最適配分が存在することとなる．この場合はそのプログラムはむだなディスクI/Oを繰り返すこととなる．これは，一見矛盾するようだが，本論の考察が式(4.1)～(4.6)の最大問題を取り扱う際に，多重度Nを固定して議論したことによる．このように，多重度とおのおののプログラムの主記憶の配分の関連やライフタイム関数が個々のプログラムごとに異なる場合など現実のシステム設計上解明せねばならない点も多い．

謝辞 本文をまとめるにあたり，有益なご教示ならびにご討論いただいた本学電気通信研究所野口正一教授に感謝の意を表します．

参 考 文 献

1) 益田：仮想記憶の制御方式，情報処理，Vol. 21, No. 4, pp. 369-377 (1980).
2) 益田，亀田：オペレーティングシステムの性能解析，情報処理叢書 9，オーム社，東京(1982).
3) Spirn, J. R.: A Model for Dynamic Allocation in a Paging Machine, Proc. 8th Annual Princeton Conf., pp. 331-334 (1974).
4) Ghanem, M. Z.: Study of Memory Partitioning for Multiprogramming Systems with Virtual Memory, *IBM J. Res. Dev.*, Vol. 19, No. 5, pp. 451-457 (1975).
5) Fin, T-H. and Kameda, H.: An Extension of a Model for Memory Partitioning in Multiprogrammed Virtual Memory Computer Systems, *JIP*, Vol. 2, No. 2, pp. 89-96 (1979).
6) Belady, L. A. and Kueher, C. J.: Dynamic Space Sharing in Computer Systems, *Comm. ACM*, Vol. 12, No. 5, pp. 282-288 (1969).
7) Baskett, F., Chandy, K. M., Muntz, R. R. and Palacios, F. G.: Open, Closed, and Mixed Networks of Queues with Different Classes of Customers, *J. ACM*, Vol. 22, No. 2, pp. 248-260 (1975).
8) Cox, D. R.: A Use of Complex Probabilities in the Theory of Stochastic Processes, Proc. Cambridge Phil. Soc. 51, pp. 313-319 (1955).

付録1（定理1の証明）

状態 $\boldsymbol{k}_s=(k_{1s}, k_{2s}, \cdots, k_{Ns})$ のうち $\sum_{i=1}^{N} \boldsymbol{k}_{is}=r$ なる集合を $R_s(r)$ とし，状態 $\tilde{\boldsymbol{k}}=(\boldsymbol{k}_1, \boldsymbol{k}_2, \cdots, \boldsymbol{k}_S)$ のうち $\sum_{s=1}^{S} \boldsymbol{k}_s=\boldsymbol{l}$ なる集合を $\boldsymbol{R}(\boldsymbol{l})$ とする．さらに，$\boldsymbol{R}(\boldsymbol{l})$ を

$$\boldsymbol{S}(r_1, r_2, \cdots, r_S)=\left\{\tilde{\boldsymbol{k}} \mid |\boldsymbol{k}_1|=r_1, |\boldsymbol{k}_2|=r_2, \cdots, |\boldsymbol{k}_S|=r_S, \sum_{s=1}^{S} r_s=|\boldsymbol{l}|\right\}$$

なる集合に分割して考える．

式(3.5)で $k_{10}+k_{20}+\cdots+k_{N0}=n$ なるものを加え合わせれば $p(n)$ が求まるから，$\boldsymbol{N}=(1, 1, \cdots, 1)$ として

$$p(n)=\frac{1}{G}\sum_{\boldsymbol{k}_0\in R_0(n)} \pi_0(\boldsymbol{k}_0) \sum_{\tilde{\boldsymbol{k}}\in \boldsymbol{R}(\boldsymbol{N}-\boldsymbol{k}_0)} \pi_1(\boldsymbol{k}_1)\cdots\pi_S(\boldsymbol{k}_S)$$

ここで，$\tilde{\boldsymbol{k}}\in\boldsymbol{S}(r_1, r_2, \cdots, r_S)$ ならば，

$$\pi_1(\boldsymbol{k}_1)\pi_2(\boldsymbol{k}_2)\cdots\pi_S(\boldsymbol{k}_S)=\prod_{s=1}^{S} a_s(r_s)\sigma_s^{r_s}$$

と計算でき，同一な値をとる．また，$\boldsymbol{S}(r_1, r_2, \cdots, r_S)$ の元の数は $(r_1+r_2+\cdots+r_S)!/(r_1!r_2!\cdots r_S!)$ である．

$\sum_{\boldsymbol{k}\in \boldsymbol{R}(N-\boldsymbol{k}_0)}$ を $\sum_{r_1+r_2+\cdots+r_S=N-n}\ \sum_{\boldsymbol{k}\in \boldsymbol{S}(r_1,r_2,\cdots,r_S)}$ と分解して，$p(n)$ は

$$p(n)=\frac{1}{G}\sum_{\boldsymbol{k}_0\in \boldsymbol{R}_0(n)} a_0(n)\rho_1{}^{k_{10}}\rho_2{}^{k_{20}}\cdots\rho_N{}^{k_{N0}}$$
$$\times\sum_{r_1+r_2+\cdots+r_S=N-n}\frac{(N-n)!}{r_1!\,r_2!\cdots r_S!}$$
$$\times\prod_{s=1}^{S} a_s(r_s)\sigma_s{}^{r_s}$$

と書き表され，この式を整理して定理1の式(3.10)〜(3.14)が導出される．

付録 2（定理 2 の証明）

式(3.15)の $\sum_{k=0}^{n} p(k)\ (n<N)$ が任意の ρ_i に関して狭義の単調減少関数で下に凸なることを示せばよく，ρ_N について証明すれば十分である．

式(3.10)の右辺の分子・分母は ρ_N に関してたかだか一次であるから，

$$\sum_{k=0}^{n} p(k)=\frac{A_n+B_n\rho_N}{A_N+B_N\rho_N}=\frac{B_n}{B_N}+\frac{(A_nB_N-B_nA_N)}{B_N(A_N+B_N\rho_N)}\qquad \text{(A. 1)}$$

ただし，

$$A_n=\sum_{k=0}^{n} g_k u_k,\quad B_n=\sum_{k=1}^{n} g_k u_{k-1}$$
$$g_k=a(k)h(N-k),\ u_k=v_{N-1}(k)$$

また，式(A.1)の $A_nB_N-B_nA_N$ は次のように整理できる．

$$\begin{aligned}A_nB_N-B_nA_N&=\sum_{k=0}^{n}\sum_{j=1}^{N} g_k u_k g_j u_{j-1}-\sum_{k=0}^{N}\sum_{j=1}^{n} g_k u_k g_j u_{j-1}\\&=\sum_{k=0}^{n}\sum_{j=n+1}^{N} g_k u_k g_j u_{j-1}-\sum_{k=n+1}^{N}\sum_{j=1}^{n} g_k u_k g_j u_{j-1}\\&=\sum_{k=1}^{n}\sum_{j=n+1}^{N} g_k g_j(u_k u_{j-1}-u_j u_{k-1})\\&\quad+\sum_{j=n+1}^{N} g_0 u_0 g_j u_{j-1}\end{aligned}$$

そこで，$k<j$ において $u_k u_{j-1}-u_j u_{k-1}>0$ が証明できれば，$A_nB_N-B_nA_N>0$ となり，命題が成り立つ．

u_k は $N-1$ 変数の k 次の基本対称式であり，一般に，r 変数 $x_1, x_2, \cdots, x_r\ (x_j>0\,;\ j\leqq r)$ の k 次の基本対称式を $v_r(k)$ とすれば，

$$\frac{v_r(1)}{v_r(0)}>\frac{v_r(2)}{v_r(1)}>\cdots>\frac{v_r(k)}{v_r(k-1)}>\cdots>\frac{v_r(r)}{v_r(r-1)}\qquad \text{(A. 2)}$$

が成り立つことを以下に数学的帰納法で証明する．

まず，$r=2$ の場合は，$v_2(0)=1$，$v_2(1)=x_1+x_2$，$v_2(2)=x_1x_2$ となるから，$v_2(1)/v_2(0)=x_1+x_2$，$v_2(2)/v_2(1)=x_1x_2/(x_1+x_2)$ より不等式(A. 2)が成り立つ．

$r(r\geqq 2)$ において不等式(A. 2)が成り立つとすれば，$v_{r+1}(k)=v_r(k)+x_{r+1}v_r(k-1)$ と表せるから，

$$\frac{v_{r+1}(k)}{v_{r+1}(k-1)}=\frac{v_r(k)+x_{r+1}v_r(k-1)}{v_r(k-1)+x_{r+1}v_r(k-2)}$$

よって，$k=2, 3, \cdots, r$ において

$$\begin{aligned}v_r(k-1)/v_r(k-2)&>v_{r+1}(k)/v_{r+1}(k-1)\\&>v_r(k)/v_r(k-1)\end{aligned}$$

さらに，次の関係が成り立つ．

$$v_{r+1}(1)/v_{r+1}(0)=\sum_{j=1}^{r+1} x_j>v_r(1)/v_r(0),$$
$$v_{r+1}(r+1)/v_{r+1}(r)=1\Big/\Big(\sum_{j=1}^{r+1}1/x_j\Big)<v_r(r)/v_r(r-1)$$

よって，$r+1$ において不等式(A. 2)が成り立つ．

付録 3（関数 $\phi(x)$ の性質）

式(5.4)を式(5.1)に代入して，x で微分すれば，

$$dU(x)/dx=V(x)/(G_0+G_1y+G_2z)^2$$
$$V(x)=h_0y'+h_1z'+h_2(yz'-y'z)$$

ただし，$y'=\alpha A\{(1+x)^{\alpha-1}-(1-x)^{\alpha-1}\}$，$z'=-2\alpha A^2x\times(1-x^2)^{\alpha-1}$，また，$h_0$，$h_1$，$h_2$ は定数で，$h_0=(F_1G_0-F_0G_1)>0$，$h_1=(F_2G_0-F_0G_2)>0$，$h_2=(F_2G_1-F_1G_2)\geqq 0$．$V(x)=\alpha A(1-x^2)^{\alpha-1}\phi(x)$ と式(5.5)の形に整理すれば，

$$\begin{aligned}\phi(x)=h_0\{&(1-x)^{1-\alpha}-(1+x)^{1-\alpha}\}-2Ah_1x\\&-A^2h_2[2x\{(1+x)^{\alpha}+(1-x)^{\alpha}\}\\&+(1-x^2)^{\alpha}\{(1-x)^{1-\alpha}-(1+x)^{1-\alpha}\}]\end{aligned}$$

よって，$\phi(0)=0$，$\lim_{x\to 1}\phi(x)=\infty$ なる性質(a)を得る．$\phi(x)$ を微分して整理すれば，

$$\begin{aligned}\phi'(x)=\{&(1-x)^{-\alpha}+(1+x)^{-\alpha}\}[h_0(\alpha-1)\\&-h_2A^2(1-x^2)^{\alpha}(1+\alpha)]-2Ah_1\end{aligned}$$

となり，$\lim_{x\to 1}\phi'(x)=\infty$ は明らか．さらに，{ }，[ ] 内の式は狭義の単調増加であるから，ある $x^*(0\leqq x^*<1)$ において，$\phi'(x^*)\geqq 0$ ならば，[ ]>0 でなければならないから，$x^*<x$ において $\phi'(x^*)<\phi'(x)$ が成り立つ．この性質は $\phi'(x)=0$ は区間 $[0, 1)$ においてたかだか一つの根をもつことを示している．

$x=0$ における2次の微係数は，$y'=0$，$z'=0$，$V(0)=0$ より

$$\left.\frac{d^2u(x)}{dx^2}\right|_{x=0}=\left.\frac{V'(x)}{(G_0+G_1y+G_2z)^2}\right|_{x=0}=\frac{\alpha A\phi'(0)}{(G_0+G_1y+G_2z)^2}$$

と表され，式(5.6)を得る．

（昭和58年 6月14日受付）
（昭和58年11月15日採録）