SE13-3
(1976. 1. 30)

多重プログラミングシステムにおける資源配分の適応制御方式について*

亀田壽夫（電気通信大学）

〔要旨〕

本稿は，多重プログラミング・システムにおける資源利用効率向上のための，適応的な負荷調節の一方式について述べ，その解析を試みたものである．この方式は，古典経済学にのべられた'みえざる手'のようなものの働きによって，各資源利用のかたよりを是正しようとするものである．つまり各ユーザプログラム U_j が資源 R_i に申し出る価格 W_{ij} が，優先度を表わすものとして周期的に計算される．これらの価格は，ユーザが直接関知するものではなく，各ユーザプログラムの性質や予算制約と，各資源における輻輳の度合，等によって決められる，いわば影の価格である．この結果，ある資源における輻輳によるかたよりが，その資源を多く用いるプログラムの優先度を下げさせ，その資源を少なく用いるプログラムの優先度を上げさせることによって，ただされるようになる．本稿では簡単さのため，処理装置型資源に限定して話を進めた．更に，この方式を実現する場合の問題点についても論じた．

1．はじめに

計算機のシステムは，いくつかの中央および周辺処理装置と記憶装置等から構成されるが，これらの諸装置のことを'資源（Resource）'と呼ぶことが多い．計算機システムのオペレーティング・システムの目的の一つは，与えられた資源を用いていかに多くの仕事を処理するかということにある．

本稿は，多重プログラミング・システムについて考察するが，そこで資源配分上問題となることは，各時点において，各処理装置にどのプログラムからの要求を処理させるかということである．（ここでは簡単のために記憶装置の取り扱いは行なわない．）このような選択は，一つには各プログラムに装置利用の優先度を割り当てることによって行なうことができる．よく採用される方式としては，プログラムからの装置使用要求を先着順に処理する First-Come-First-Served 方式があるが，この論文では，各資源における輻輳の度合と各プログラムがどれ位の資源を必要とするかにもとづいて優先度をわりあてる方式を考察し，それがどのように，より効率のよい資源の利用につながるかをみることにする．

しばしば，多重プログラミング・システムにおいても，ある資源に使用が集中し，他の資源があまり使われないということがおこり得る．このような場合，この輻輳のある資源は bottleneck をなすといわれ，このために各資源の利用のバランスがくずれると考えられる．もし，この輻輳のある資源をあまり使わず他の空いている資源をよく用いるプログラムを，より重点的に処理するようにして，資源利用のつりあいがとれるようにすれば，システム全体の効率がもっとよくな

---

* 本稿は，筆者の IBM T. J. Watson Reseach Center 滞在中の研究報告を，University of Toronto 滞在中に手を加えたもの[4]をさらに発展させたものである．

ることが期待できる。この目的のためには，各プログラムが各装置をどの位使用するかを測定し，それに基づいて適当な制御を，優先度わりあてのようなスケジューリング方式でほどこすことが必要となるが，これは Feedback 制御とよんでもよいものである。

本論文では，Feedback を組み込んだ，ある資源配分の方式を論じ，この方式のもとでユーザプログラムの処理がどのようになるかを解析する。この方式の底流をなす考え方は，経済学のことばでいえば，資源配分の問題が価格のメカニズムで解決されるとすること（Adam Smith のみえざる手）に類似しているものである。計算機システムにおいても価格のメカニズムのようなものが考えられるのは，各ユーザのプログラムが，それぞれの計算機使用料というものを考えた上で，計算機センターに委託され，計算機使用料が実際に支払われることが前提とされているからである

ここでは，'みえざる手'のようなものは，次のように働くと考えられる。つまり，ある資源がほぼ完全に利用されて他の資源が空くような場合，ほぼ完全に使われる資源の価格が比較的高くなり，他はより低くなる。このとき，こみ合っている資源の方をあまり使用せず，他の資源を多く使うユーザプログラムの方が，同じ予算制約の下では，より有利になり，より高い優先度とより多くの処理を受けることになる結果，空いている資源の方の利用率も高くなるというわけである。

このような価格のはたらきを計算機資源利用の改善に用いようとする提案は既にいくつかあらわれている[2,5,6,7]。これらの方式のねらいは，価格を変えることによってユーザからの需要を直接制御しようとするところにある。つまりユーザ自身が価格の変動を知って，プログラムの性質を変えたりするということである。このような方式の一つの問題点は，ユーザの需要を適応的に制御するためには，輻輳のゆらぎに応じて価格を頻繁に変えなければならないが，逆にそのように価格が頻繁に変わることはユーザにとって不便であり不都合であるということにある。

本論文に述べる方式が，これらの方式と違うところは，実際にユーザが知らなければならない実価格を用いず，ユーザが直接に知らなくてよい，いわば影の価格を，内在的制御パラメタとして用いて負荷のつりあいをはかることにある。

まず方式を記述することからはじめて，次にシステムのふるまいについて論ずることにし，後に実際に具現化するときに生じ得る問題についてふれる。

2. 資源配分方式

多重処理計算機システムでは，次の2種類の資源がある

1) 処理装置型資源（CPU および周辺処理装置）

$R_i \quad i = 1, 2, \cdots, n$

2) 記憶装置型資源

これらに加えて，ユーザプログラム

$U_j \quad j = 1, 2, \cdots, m$

があり，それらは単位処理時間あたり C_j の処理価値を有するものとする。

ここでは前述したように記憶装置型資源のことは考慮に入れないことにする．

システムの性能に制御を及ぼす可能性を導入するために，次のような方式を考える．つまり，各ユーザプログラム U_j に，各資源 R_i の使用優先度が，パラメタ W_{ij} の大きさに従って次のようにわりあてられる．つまり，

$$U_j\text{の優先度} \gtrless U_k\text{の優先度（資源}R_i\text{の利用における）} \iff W_{ij} \gtrless W_{ik} \qquad (1)$$

$W_{ij} = W_{ik}$ のときは，到着順が処理順をきめるものとする．

ユーザプログラム $U_1, \cdots U_m$ がメモリ内に存在し，その間はどのユーザプログラムもシステムに入ってこずどれも完了して出ていくことがないような時間間隔 T を考えることにしよう．X_{ij} を，この時間間隔 T において，プログラム U_j が資源 R_j を使用する時間長，とする．そうすると，次のようになる．

$\sum_i^n X_{ij}$ ＝ プログラム U_j が各資源を使用した時間の総和
＝ プログラム U_j についての処理時間

ここで，このような時間間隔 T において $\sum_i^n X_{ij} = 1$ になるような場合，このプログラム U_j は'単位処理'をうけると考える．以後 X_{ij} について述べるときには，いつもこのような単位処理の場合であるとする．つまり，つねに $\sum_i^n X_{ij} = 1$ であるとする．更に，プログラム U_j がこのような単位処理をうけるために主メモリに存在する時間間隔を T_j と記すことにする．（T_j は $\sum_i^n X_{ij} = 1$ のときの T の値である．）通常，処理装置は，要求したらすぐ使えるというわけでなく，待ち時間が必ず存在し，$T_j > 1$ となる．

本稿で提案する方式では，W_{ij} を，プログラム U_j が資源 R_i に申し出る影の価格とみなす．そして，プログラム U_j が申し出る価格と資源使用量との積の総和は，次のような予算制約条件をみたすものとする．

$$\sum_i^n W_{ij} X_{ij} \leq C_j \qquad (2)$$

つりあいのとれた資源利用をはかるため，W_{ij} を決定するのに，一種の競争的解法を考える．つまり，各ユーザプログラム U_j にとって，その予算制約内で，単位処理当り必要な滞在時間 T_j が最小になるように，W_{ij} を決定するのである．

このような方法で，優先度 W_{ij} がどのように決められ，この方式がどのようにつりあいのとれた資源配分につながるかを次にみることにしよう．

方式の解析

この方式を解析するために，ミクロ経済分析にたような近似を用いる．つまり，メモリに存在し並行処理されるプログラム数が大きく，その中の一つだけのユーザのふるまいのちがいがシステム全体に及ぼす影響が無視できるという仮定を設ける．ただし，多くのユーザのふるまいが変われば，システム全体の状態は当然変わるのである．つまり，ただ一つだけのユーザプログラムを付け加えたり

取り除いたりしても，ただ一つだけのユーザプログラムの優先度を変えてみても，多くの他のプログラムの性質や優先度が変わらなければ，システム全体としての性能は，殆ど変化しないと考えるのである。

この仮定の下に，資源 R_i を単位時間使用することに際し価格 W を申し出るプログラムが，その資源を使用するに際してこうむる待ち時間の期待値を表わすところの関数 $f_i(W)$，を考えることができる．Wは優先順位をあらわすので，待ち時間の期待値 $f_i(W)$は，申し値 W を増すと，単調に減少するはずである．つまり，プログラムが高い価格を申し出るほど，その待ち時間の期待値は短かくなるからである．ユーザプログラムの数が多いという仮定の下で，$f_i(W)$は，連続微分可能であると仮定する．

各ユーザプログラムは各資源を順次用いていくと考えられるので，ユーザプログラム U_j の単位処理に要する全待ち時間（あるいは，経過時間）の期待値 T_j は次のようにあらわされる．

$$T_j = \sum_i^n f_i(w_{ij})\, x_{ij} \tag{3}$$

かくて，ユーザプログラムが各資源を順次用いると仮定して，各ユーザプログラムにとっての最適方針は，次のようにあらわされる．

制約 $C_j = \sum_i^n w_{ij}\, x_{ij}$ のもとで

$$T_j = \sum_i^n f_i(w_{ij})\, x_{ij} \tag{4}$$

を最小にするような w_{ij} の値の組を生かすこと

ここで，制約条件の不等式がここでは等式になっていることに注意されたい．これは，f_i が単調減少関数であることから明らかである．

これらの仮定の下で，Lagrange 乗数法を用いて，w_{ij} を求めることができる．プログラム U_j の Lagrangean は，λ_j を Lagrange 乗数として，

$$L_j = T_j + \lambda_j \left(C_j - \sum_i^n x_{ij}\, w_{ij} \right) \tag{5}$$

最適条件は，

$$\frac{\partial L_j}{\partial w_{ij}} = 0 \qquad i = 1, 2, \cdots, n \tag{6}$$

つまり，

$$\frac{d\, f_i(w_{ij})}{d\, w_{ij}} = \lambda_j \qquad i = 1, 2, \cdots, n \tag{7}$$

かくて，$n+1$ 個の未知数 w_{ij} $(i = 1, 2, \cdots, n)$ と λ_j が，次の $n+1$ 個の方程式から決められる．

$$\begin{aligned} &\frac{d\, f_i(w_{ij})}{d\, w_{ij}} = \lambda_j \qquad i = 1, 2, \cdots, n \\ &C_j = \sum_i^n x_{ij}\, w_{ij} \end{aligned} \tag{8}$$

このようにして，$f_i(w)$ が分らであれば，各ユーザプログラムに対して最適の w_{ij}（これを w_{ij}^* と記す）が得られる．

ところで，これまでの話とは逆になるが，すべてのプログラムがそれ自身の w_{ij} の値を決めると，システム全体の状態も決定され，$f_i(w)$ が決定されるのである．つまり $\{w_{ij}\} \to \{f_i(w)\}$．一方各ユーザは $f_i(w)$ に基づいて w_{ij} をきめる．つまり $\{f_i(w)\} \to \{w_{ij}'\}$．従って次のような変換が考えられる．

$$\{w_{ij}\} \to \{f_i(w)\} \to \{w_{ij}'\} \tag{9}$$

最適解は，もちろん均衡条件をも満たさなければならない．つまり，(9) において，すべての i,j について $w_{ij} = w_{ij}'$ でなければならない．

このような均衡解が存在することは，次のように示される．w_{ij} の値の組の集合を考え，その集合から自分自身への写像 F として (9) を考えることにし，$F: \{w_{ij}\} \longrightarrow \{w_{ij}'\}$ とする．ところで w_{ij} は，すべての j について条件 $C_j \geq \sum_i^n w_{ij} x_{ij}$ を，またすべての i,j について条件 $w_{ij} \geq 0$ を満たすものと考えられるので，w_{ij} の値の組の集合は，R^{mn} 空間における空でないコンパクトで凸な部分集合であるといえる．更に，仮定により F は連続写像とみなせる（つまり，w_{ij} の微少変化は w_{ij}' の微少変化のみをもたらすと考えられる）．従って Brouwer の固定点定理（例えば[1]参照）によって，$\{w_{ij}\} = F(\{w_{ij}\})$ であるような w_{ij} の値の組み合わせが存在することが保証される．すなわち，

$$\{w_{ij}\} \longrightarrow \{f_i(w)\} \longrightarrow \{w_{ij}\}$$

このようにして，w_{ij} の最適解として，少なくとも一組の均衡解が存在することが示される．

比較静学的分析をするために，パラメタ t を関数 f_i と C_j に導入する．t は時間パラメタとみなすこともでき，その場合 f_i, C_j は時間の関数となる．

(8) は，次式のようになる．

$$\begin{aligned} &\frac{\partial f_i(w_{ij}, t)}{\partial w_{ij}} = \lambda_j \qquad i = 1, 2, \cdots, n \\ &C_j(t) = \sum_i^n x_{ij} w_{ij} \end{aligned} \tag{10}$$

(10) から，λ_j, w_{ij} がての関数として得られ，これを $\lambda_j = \lambda_j^*(t)$, $w_{ij} = w_{ij}^*(t)$ $(i = 1, 2, \cdots, n)$ と記す．ここで星印は最適値であることを示すとする．これを (4) の2つの式に代入すると次のようになる．

$$\begin{aligned} T_j^*(t) &= \sum_i^n f_i(w_{ij}^*(t), t)\, x_{ij} \\ C_j(t) &= \sum_i^n x_{ij}\, w_{ij}^*(t) \end{aligned} \tag{11}$$

(11) で適当に微分して $w_{ij}^*(t)$ を消去すると

$$\frac{d}{dt} T_j^*(t) = \sum_i^n \frac{\partial f_i}{\partial t} x_{ij} + \lambda_j^* \frac{\partial C_j}{\partial t} \tag{12}$$

ここで，システム全体の状態はかわらず，ある一つのプログラムの予算 C_j だけが変わる場合を考えてみよう．この場合，$\partial f_i / \partial t = 0$ となるので，(12) より，

$$\frac{d}{d\tau} T_j^*(\tau) = \lambda_j \frac{dC_j}{d\tau} \tag{13}$$

ところで，(10)式と $f_i(w)$ が w について単調減少であることから，

$$\lambda_j \leq 0$$

従って(13)式は，一般に予算制約 C_j が大きいほど，待ち時間は短かくなることをあらわしている．

次に，同じプログラムが，輻輳が異なる状況におかれた場合を考えよう．この場合，$\partial f_i/\partial\tau \neq 0$ ではないが，予算制約は変わらず $dC_j/d\tau = 0$ である．このとき(12)から，

$$\frac{d}{d\tau} T_j^*(\tau) = \sum_i^n \frac{\partial f_i}{\partial \tau} X_{ij} \tag{14}$$

これは，例えば資源 R_i の輻輳がより大きくなる（つまり，$\partial f_i/\partial\tau > 0$）と，資源 R_i を多く使う（つまり X_{ij} の大きな）ユーザプログラムほど待ち時間が長くなることを示している．

更に，2つの資源 R_1 と R_2 のうち，R_1 は輻輳が増え，R_2 は減りつつあり，他の資源については変化がないとしよう．これは，a_1, a_2 を正として $\partial f_1/\partial\tau = a_1$, $\partial f_2/\partial\tau = -a_2$, $\partial f_i/\partial\tau = 0$ $(i = 3, 4 \cdots n)$ とあらわせる．このとき，(14)から

$$\frac{d}{d\tau} T_j^*(\tau) \gtreqless 0 \quad \Longleftarrow \quad a_1 X_{1j} \gtreqless a_2 X_{2j}$$

すなわち，資源 R_1 よりも資源 R_2 を相対的に多く使うユーザプログラムは，その処理率 $(1/T_j^*)$ が増大し，R_1 よりも R_2 を相対的に少なく使うプログラムの処理率は減少する．従って，この方式では，各ユーザプログラムの処理率は，全体の資源使用状態のバランスが回復されるように変わることがわかる．

更に，w に対する f_i の関係について考えてみよう．これは，大体，経済学上の標準的な効用曲線と同じようだと考えられる．すなわち w の大きな値に対しては df_i/dw は比較的小さく，w の小さな値に対しては比較的大きい．従って限界効用は w の増大とともに減少し，$d^2f_i/dw^2 \geq 0$ と近似することができよう．

各ユーザプログラム U_j の最適条件が式(10)であらわされ，上の仮定のもとで df_i/dw_{ij} が単調増加であることから，すべてのユーザプログラムは λ_j の大きさによって並べることができ，各処理装置における優先度もその順となる．つまり

$$\lambda_j \gtreqless \lambda_k \Longleftrightarrow w_{ij} \gtreqless w_{ik} \quad (i = 1, 2, \cdots, n) \tag{15}$$

更に，λ_j^* と w_{ij}^* の微分を調べることにしよう．式(10)，(11)より，

$$\frac{d\lambda_j^*}{d\tau} = \left[\sum_i \left(X_{ij} \Big/ \frac{\partial^2 f_i}{\partial w_{ij}^2}\right) \frac{\partial^2 f_i}{\partial\tau \partial w_{ij}}\right] \Big/ \left[\sum_i \left(X_{ij} \Big/ \frac{\partial^2 f_i}{\partial w_{ij}^2}\right)\right]$$

$$\frac{dw_{kj}^*}{d\tau} = \left[\sum_i \left(X_{ij} \Big/ \frac{\partial^2 f_i}{\partial w_{ij}^2}\right) \frac{\partial^2 f_i}{\partial\tau \partial w_{ij}} - \frac{\partial^2 f_k}{\partial\tau \partial w_{kj}} \sum_i \left(X_{ij} \Big/ \frac{\partial^2 f_i}{\partial w_{ij}^2}\right)\right] \Big/ \left[\frac{\partial^2 f_k}{\partial w_{kj}^2} \sum_i \left(X_{ij} \Big/ \frac{\partial^2 f_i}{\partial w_{ij}^2}\right)\right] \tag{16}$$

$k = 1, 2, \cdots, n$

ここで，資源利用におけるふつりあいが生じた場合，つまり，例えば資源 R_e が bottleneck となるような場合，各プログラムのふるまいがどうなるかみてみよう．ここで，輻輳の増加をパラメタτの増加であらわすことにする．そして，資

源 R_ℓ の輻輳の増大は，資源 R_ℓ を要求するプログラムの数が，その資源の容量に対して相対的に増大することを意味すると考える．このとき，同じ W の値に対しての待ち時間の期待値 f_ℓ は，（W よりも高い値を申し出るプログラム数の資源 R_ℓ の容量に対する割合が，輻輳増大とともに大きくなるので，）増大するであろう．つまり $\partial f_\ell/\partial \tau > 0$．また，どのような ΔW についても，申し出価格の値が W と $W+\Delta W$ の間にあるプログラム数の，当の資源に対する割合も増大すると考えられるので，$f_\ell(W) - f_\ell(W+\Delta W)$，すなわち，申し出値がこの区間にあるプログラムの処理がすむのを待つ時間は，増大すると考えられる．従って，$\partial f_\ell/\partial w$ は，資源のこみあいにともなって減少し $\partial/\partial\tau(\partial f_\ell/\partial w) < 0$ となる．R_ℓ 以外については，輻輳が変化しないとしたので，これに対し $\partial f_i/\partial\tau = 0$，および $\partial/\partial\tau(\partial f_i/\partial w) = 0$ となる．これらの条件より，(16) から

$$\frac{d w_{\ell j}^*}{d\tau} = \left(x_{\ell j} \Big/ \frac{\partial^2 f_\ell}{\partial w_{\ell j}^2}\right) \frac{\partial}{\partial\tau}\left(\frac{\partial f_\ell}{\partial w_{\ell j}}\right) \Big/ \sum_i \left(x_{ij} \Big/ \frac{\partial^2 f_\ell}{\partial w_{ij}^2}\right) < 0 \qquad (17\text{-}1)$$

$$\frac{d w_{kj}^*}{d\tau} = \left(x_{\ell j} \Big/ \frac{\partial^2 f_\ell}{\partial w_{\ell j}^2}\right) \frac{\partial}{\partial\tau}\left(\frac{\partial f_\ell}{\partial w_{\ell j}}\right) \Big/ \left[\frac{\partial^2 f_\ell}{\partial w_{\ell j}^2} \sum_i \left(x_{ij} \Big/ \frac{\partial^2 f_\ell}{\partial w_{ij}^2}\right)\right] < 0 \quad (k \neq \ell) \qquad (17\text{-}2)$$

$$\frac{d w_{\ell j}^*}{d\tau} = \left[\frac{x_{\ell j} \Big/ \frac{\partial^2 f_\ell}{\partial w_{\ell j}^2}}{\sum_i \left(x_{ij} \Big/ \frac{\partial^2 f_\ell}{\partial w_{ij}^2}\right)} - 1\right] \frac{\frac{\partial}{\partial\tau}\left(\frac{\partial f_\ell}{\partial w_{\ell j}}\right)}{\frac{\partial^2 f_\ell}{\partial w_{\ell j}^2}} > 0 \qquad (17\text{-}3)$$

すなわち，(17-1) より，資源 R_ℓ の輻輳の増大は，もし他の状態が不変ならば，大きな $x_{\ell j}$ をもつプログラムほど（つまり資源 R_ℓ を多く使うものほど）多くの優先度を失なうことになる．また (17-2), (17-3) より，各プログラムの最適行動は，他の資源への申し出価格を下げ，資源 R_ℓ へより高い価格を申し出ることになる．従って，資源 R_ℓ の輻輳をあらわす f_ℓ は更に高価格の方にずれ，他は低価格の方にずれることになるが，このような状況によく適応できるのは，資源 R_ℓ をあまり用いない（$x_{\ell j}$ の小さい）プログラムのみである．つまり，同じ予算制約の下で他の資源への価格をあまり下げずに，資源 R_ℓ への価格を十分高くすることができる．その結果，資源 R_ℓ をあまり用いないプログラムが浮上し，R_ℓ を多く使うプログラムが相対的に沈むことになる．このようにしてシステム資源利用のかたよりが是正される方向に進むと考えられるのである．

3．具現化についての考察

以上の方式を実現させる際に生じ得る問題点のいくつかをここに検討する．

i. この方式は，各プログラムの性質についてのデータを要する．特に各ユーザプログラム U_j の単位処理当りの，各資源 R_i に対する需要の比率 —— x_{ij}，と処理価値（あるいは，予算制約）—— c_j が必要となる．全計算時間にわたる総和でなく，各時点における値が必要なのである．このため，各プログラムが実際にどのように資源を用いたかを常に測定し（モニタリング）それに基づいて予測するような方法が必要となる．

ii. 以上の解析では，プログラムの数が大きく連続性の近似ができるということを用いたが，これで実際の場合の特徴がどの程度把握できるかが問題である．

(1). パラメタの見積り

ここで，x_{ij}，c_j の一つの予測法について述べる．全くの初めには，実測値が何もないので，ユーザの予測のみによることになる．

次の時間間隔からは，x_{ij}，c_j の実測値にもよることができる．r 番目の時間間隔での実測値を ${}_rO^e$ と表わすとしよう．ここで O は x_{ij} も c_j も代表して表わすとする．これに基づいて $r+1$ 番目の時間間隔に対する予測値 ${}_{r+1}O^e$ を見積ることになる．それは例えば次のようにすればよい．

$$ {}_{r+1}O^e = (1-p)\,{}_rO^o + p\,{}_rO^e \qquad \text{ただし } 0<p\leq 1 \tag{18}$$

p は適当に選ばれる定数である．ところで (18) より，

$$ {}_rO^e = (1-p)\sum_{n=0}^{r-1} p^n\,{}_{r-n-1}O^o + p^r\,{}_0O^e \qquad \text{及び} \quad \sum_{n=0}^{r-1}(1-p)p^n + p^r = 1 \tag{19}$$

つまり，この方法は過去の実測値の重みはけ平均になっている．ここで $r=0$ が一番初めの時間間隔とする．

更に，各時間間隔における資源 R_i の平均価格 $\overline{w_i}$ というものを考えることができ，その実測値 $\overline{w_i}^o$ は次のように求められる．

$$ \overline{w_i}^o = \sum_i^m w_{ij}\left(\frac{x_{ij}^o}{T_j^o}\right) \tag{20}$$

ここで，w_{ij} はその時間間隔における実際の各ユーザの申し出価格である．これより (18) と同様に次期の $\overline{w_i}$ の見積り $\overline{w_i}^e$ が得られる．この際全くの初期値としてはレンタル・コストなどを用いればよいであろう．

次の問題は，このようにして得られた x_{ij}，c_j，$\overline{w_i}$ の値に基づいて，優先度 w_{ij} を計算することである．ここで，各ユーザプログラムの方針として，(8) の中の $df_i/dw_{ij}=\lambda_j$ をそのまま用いることは困難なので，そのかわり，

$$ \frac{\overline{w_i}}{w_{ij}} = -\lambda_j \tag{21}$$

と近似してみよう．ちなみにこれは (15) も満たす．ユーザプログラム U_j の方針は，

$$ \begin{aligned} &\overline{w_i}/w_{ij} = -\lambda_j \\ &c_j = \sum_i^n w_{ij}\,x_{ij} \end{aligned} \tag{22}$$

となる．(22) から直ちに次の関係が得られる．

$$ \begin{aligned} \lambda_j &= -\left(\Sigma_i^n x_{ij}\,\overline{w_i}\right)/c_j \\ w_{ij} &= c_j\,\overline{w_i}\,/\,\Sigma_i^n x_{ij}\,\overline{w_i} \end{aligned} \tag{23}$$

(ii). システムのふるまい

この具体的近似法において，ある資源 R_i のこみ具合が変化するとどうなるか調べてみよう．(23) より

$$ \frac{\partial \lambda_j}{\partial \overline{w_i}} = -\frac{x_{ij}}{c_j} \tag{24}$$

これは，資源 R_i の輻輳が増す（平均価格 $\overline{w_i}$ が高くなる）と，大きな x_{ij} が小さな c_j をもつプログラムの優先度（λ_j で示される）が大きく減少することを示す．つまり，R_i を多く使うプログラムが予算許容の小さいプログラムの優先度が大きく下がることになり，資源利用のかたよりが是正される方向に動くことがわ

かる．

(iii) 考えられうる問題点

ここに述べた近似法は，解析のところで用いた連続性近似に基づく微分方程式による記述を，実際のシステムの離散的性格を反映した差分方程式による記述でおきかえたもののように考えることができる．この点でいくつかの問題が生じうる．実際には，各ユーザプログラムの優先度決定は，ある時間間隔ごとに行なわれるが，この時間間隔の長さをどうとるかの問題がある．これは，各時間間隔ごとのパラメタの実測値と予測値のずれがどうなるかの問題をも含む．

ここでは論じなかった大きな問題は，記憶装置のことを考慮に入れた場合とのようになるかということである．これは別に検討中である．

以上のような問題点にもかかわらず，輻輳が増加する資源に対する需要を減らすようなことをして，資源利用のつりあいを自動的にとっていくようなシステムの可能性は，十分検討に値いすると思われる．

参照文献

1. Berge, C.: Topological Spaces, New York: The MacMillan Company, 1963

2. Cunning, Inc.: "The effects of charge-back policies" EDP Analyzer, Vol. 11, No. 11, (Nov. 1973), 1-14.

3. Intriligator, M. D.: Mathematical Optimization and Economic Theory Englewood Cliffs, N.Y.: Prentice-Hall, 1971.

4. Kameda, H.: "The analysis of an adaptive workload balancing strategy in computing system resources management" International Journal of Computer and Information Sciences, Vol. 4, No. 4, (Dec. 1975), 295-306.

5. Nielson, N.: "Flexible pricing: an approach to the allocation of computer resources" Proc. 1968 FJCC, 521-531.

6. ———— : "The allocation of computer resources - is pricing the answer ?" Comm. ACM, Vol. 13, No. 8, (Aug. 1970), 467-474.

7. Smidt, S.: "Flexible pricing of computer services" Management Science, Vol. 11, No. 11, (Nov. 1973), 1-14.