計算機システムの制御と評価 17－2
（1982. 10. 4）

応答性能の実現可能性

亀田壽夫（電気通信大学）

概要：ＴＳＳの解析などに使われるFinite-source queueing modelは、single-server (processor)とinfinite-server (terminals) とから成る２ノードMarkov循環待ち行列ネットワークと同等である。ここで、モデル内の各ジョブのサービス時間は、processor において異なった平均を持つとする。本稿では、processor における各ジョブの平均応答時間の要求値の組が与えられたとき、それを実現するスケジューリング方式が存在するための必要十分条件を求めた。

1．はじめに

Coffman らは、複数クラスのジョブ到着のある、infinite-source single-server queueにおけるスケジューリングについて考察した：各クラスの到着はPoisson で、サービス時間は、クラスごとに異なる平均を持つ指数分布をなす場合である。性能指標として、各クラスのジョブに対する平均応答時間を各要素とする性能ベクトルが考えられた。そして、Coffman らは、要求される性能ベクトルが与えられたとき、それを実現するスケジューリング方式が存在するための必要十分条件を得た(Coffman and Mitrani[1980])。

タイムシェアリング・システムの解析には、より現実的なモデルとして、finite-source queueing modelが考えられている。本稿では、異種ジョブを持つfinite-source queueing modelにおいて、上述と類似の結果が得られることを示す。ここで考える性能ベクトルは、各ジョブに対する(processorの）利用率を各要素とする。［これから、(processorにおける）各ジョブの平均応答時間を各要素とする応答時間ベクトルが直ちに求められる。］そして、与えられた性能ベクトルを実現するスケジューリング方式が存在するための必要十分条件を求める。また、性能指標の最適化の問題についても若干論ずる。

2．モデル

ここで考えるモデルは、２つのサービスセンタとN個のジョブとから成る閉循環待ち行列ネットワークである。一方のサービスセンタ（processor と呼ぶ）は、single-server から成りジョブは待つことがある。他方のサービスセンタ(terminalsと呼ぶ）は、multiple-server から成り、その各serverは各1 つのジョブに対応し、ジョブは待つことがない。ジョブ j は、processor で平均$1/u_j$, terminals で平均$1/v$ の、指数サービス時間分布を経験するとする。

processor で使われるスケジューリング方式は、次のように特徴付けられる：work-conserving であり、スケジューリングの決定において、現在および過去の情報のみ用いる。スケジューリング方式の範囲は、Kameda[1982]と同じであり、次のものを含む：FCFS, preemptiveおよびnonpreemptive priority, preemptive

およびnonpreemptive LCFS, processor sharing, generalized processor sharing 等。

性能ベクトル：ジョブjに対する、processor における平均応答時間をT_jと表わそう。応答時間ベクトル$(T_1, T_2, ..., T_N)$ をTと表わす。ジョブjに対するprocessor の利用率をU_jと表わそう(Uをprocessor 利用率とすると$\Sigma_{j=1}^{N} U_j = U$ となる).利用率ベクトル$(U_1, U_2, ..., U_N)$ をUと表わす。これから明らかに

$$U_j = 1/[u_j(T_j + 1/v)], \; j = 1, 2, ..., N. \qquad (2,1)$$

今後、主にUをシステムの性能ベクトルと考えることにする。processor におけるスケジューリング方式がSのとき、性能ベクトルの値がUであると、SがUを実現するという。与えられた性能ベクトルは、それを実現するスケジューリング方式が存在するとき、実現可能であるという。実現可能な性能ベクトルすべての集合をHと表わす。

Finite-source queueing modelの性質：ジョブ番号の任意の集合MとIとに対し、MがMの要素数（濃度）、IがIの要素数（濃度）とし、$r_j = v/u_j$ とするとき、

$$U(M) = 1 - 1/(\Sigma_{n=0}^{M} n! \Sigma_{I \subseteq M, I=n} \Pi_{j \in I} r_j) \qquad (2.2)$$

と定義する。さらに、モデル内の全ジョブの番号の集合をNと表わす。このとき、次の性質が得られている（Kameda[1982]）。

性質1. processorの利用率は、processor におけるスケジューリング方式によらず一定で、次のように表わせる：

$$\Sigma_{j \in N} U_j = U(N) \qquad (2.3)$$

性質２．Nの、任意の空でない真部分集合Zのジョブに対する、processor 利用率が最大になるための必要十分条件は、Zの全ジョブが、他のジョブよりも高いpreemptive priorityを与えられることであり、

$$\Sigma_{j \in Z} U_j \leq U(Z). \qquad (2.4)$$

式（2.3)および、ZがNの任意の真部分集合であるとき式（2.4)を、満足させる性能ベクトルUの集合をH^*と表わそう。すなわち、H^*内の任意のUは次の条件を満たす：

$\Sigma_{j \in N} U_j = U(N)$ ，および

Nの任意の空でない真部分集合Zに対して
$\Sigma_{j \in Z} U_j \leq U(Z)$.

3．実現可能な性能ベクトル

前節の性質1,2 は、性能ベクトルUが実現可能であるための必要条件を示している。本節では、それらが十分条件でもあることを示す。性質1,2 から、

$$H \subseteq H^* \tag{3.1}$$

本節では$H \supseteq H^*$となることを示す。
まず、mixing strategy と呼ばれる、スケジューリング方式を考える。そしてH^*の各要素が、mixing preemptive priority strategy の一つによって実現されることを示す。

Mixing strategy:スケジューリング方式S1,S2,...,Skがあるとしよう。processor の各busy period 開始点において、そのbusy period 中のスケジューリング方式としてSi を確率p(i)で選ぶようなスケジューリング方式を考えよう。このようなスケジューリング方式を、S1,S2,...,Skに対する、パラメータp(1),p(2),...,p(k-1)であるmixing strategy と呼ぶ。明らかに、前節で述べたスケジューリング方式に対するmixing strategy も、前節のスケジューリング方式のクラスに含まれる。これについて次の定理が得られる。

定理1．S1, S2, ...,Skに対する、パラメータp(1),p(2),...p(k-1)の、mixing strategy による利用率ベクトルUは、次式で与えられる。

$$U = \Sigma_{i=1}^{k} p(i) U^i$$

ただし、$p(k) = 1 - \Sigma_{i=1}^{k-1} p(i)$であり、$U^i$はスケジューリング方式Siが用いられたときの利用率ベクトルを表わす。

この定理は、processor のbusy period の平均がスケジューリング方式によらないこと(Kameda[1982],Corollary 2)から明らかである。

実現可能性の十分条件：preemptive priority 方式は、N!とおりあることに注意されたい。ジョブj_1, j_2, ..., j_N の順にpriorityが与えられるpreemptive priority方式による性能ベクトルを、$P(j_1, j_2, ..., j_N)$で表わそう。(2.3)を満たす性能ベクトルは、P(1, 2, ..., N), ...,P(N, N-1, ..., 1) のN!個の点を通るN 次元超平面上にある。H^{**}を、これらの点から得られる閉包であるとしよう：すなわち、UがH^{**}に含まれるための必要十分条件は、P(1, 2, ..., N), ..., P(N, N-1, ..., 1) のうちからのN個の点P_1, P_2, ..., P_N とN個の数q_1, q_2, ..., q_N ($q_j > 0$ で$\Sigma_{j=1}^{N} q_j = 1$)があり、次式が成立つことである。

$$U = \Sigma_{j=1}^{N} q_j P_j.$$

定理1の各Siをpreemptive 方式の一つと考えると、H^{**} 内のすべての性能ベクトルは実現可能であり、Hの定義から

$$H^{**} \subseteq H. \tag{3.2}$$

補題1. 任意の異なる2つの集合Z^*と$Z°$とに対して、(2.4) における等号が成り立つとすると、$Z^* \subset Z°$か$Z^* \supset Z°$である。

これは、性質2より明らかである。

補題2. H^*の各頂点$U°$に対して、$U° = P(j_1, j_2, ..., j_N)$ なるようなpreemptive priority 方式($j_1, j_2, ..., j_N$)が存在する。

証明. $U° = (U_1, U_2, ..., U_N)$ がH^*の頂点であるとしよう、H^*の定義により、$U°$は、N個の超平面の交点にあり、それらの超平面の 1つが(2.3) で表わされ、他が(2.4) のうち等号が成り立つもので表わされる。したがって、$U°$の各要素は、次の N個の連立一次方程式を満たす:

$$\Sigma_{j \in Z_i} U_j = U(Z_i), \ i = 1, 2, ..., N,$$

ただし、Z_i のうち 1つが集合Nであり、他がNの空でない真部分集合である。

補題1より、部分集合Z_i、i = 1, 2, ..., Nのすべてについて、(番号付けを変えることにより) 次の関係が成立する:

$$Z_1 \subset Z_2 \subset ... \subset Z_N = N, \quad \text{かつ}$$

Z_1 はj_1を、 $Z_i - Z_{i-1}$ (i=2, 3, ..., N)はj_iを、一要素のみ含む。

それゆえ、$U°$ は、性質2より、preemptive priority 方式($j_1, j_2, ..., j_N$)の性能ベクトルになっていることがわかる。□

定理2. 任意の性能ベクトルUを実現するスケジューリング方式が存在するための必要条件は、$U \in H^*$となることである。

証明. 既に、$H \subseteq H^*$ (3.1), $H^{**} \subseteq H$ (3.2)を得ている。H^*は bounded polytopeであるので、補題2より$H^* \subseteq H^{**}$. したがって, $H \subseteq H^* \subseteq H^{**} \subseteq H$となり、これから $H = H^* = H^{**}$. □

性能ベクトルUが実現可能であることが示されると、次の問題は、それを実現するスケジューリング方式を求めることである。一つの答えが、上述のmixing strategyによって与えられる。mixされるべきstrategyと、パラメータとは、Coffman and Mitrani[1980] に示されたのと同様に、線型計画法を用いて決定できる。preemptive priority 方式の性能ベクトルの名前を、P_1, P_2, ..., $P_{N!}$と呼び変えると、問題が次のように表現される:

N!個の非負整数p_1, p_2, ..., $p_{N!}$のうちN個のみ零でないようにして、次の関係を満足させるようにせよ：

$$\Sigma_{i=1}^{N!}\ p_i = 1 \quad かつ \quad \Sigma_{i=1}^{N!}\ p_i \mathbf{P}_i = \mathbf{U}\ .$$

この問題には、N ＋ 1個の線型制約条件があるが、そのうち N個が独立であり((2.3) をみよ)，たかだか N個の変数が零でないような解を見つけることになる。これは、q_0, $\underline{q}$ = (q_1, q_2, ..., q_{N-1})の N個の変数を導入し、次の線型計画問題を解くことによって求められる：

$$制約条件 \quad p_i \ \geq\ 0\ (i=1,2,...,N),\ q_0\ \geq\ 0,\ \underline{q}\ \geq\ 0,$$

$$q_0 + \Sigma_{i=1}^{N!}\ p_i = 1,\ \underline{q} + \Sigma_{i=1}^{N!}\ p_i \mathbf{P}_i = \mathbf{U} \quad の下で$$

$\Sigma_{i=1}^{N!}$ p_iを、初期基底p_i = 0(i = 1, 2, ..., N) q_0= 1, $\underline{q}$ = Uとおき simplex 法で最大化する。目的値1 が得られたとき、p_iの値がUを得るためのパラメータ値となっている。

Coffman and Mitrani[1980] のinfinite-source queue の場合と同様に、mixing strategyでは、各ジョブに対する平均応答時間の分散が大きくなることがあるので、実際への応用には不適切かも知れない。しかし、性能ベクトルが実現可能であることがわかりさえすれば、別の適当なスケジューリング方式をさがすことができる。

4．性能指標の最適化

性能指標には、性能ベクトルの関数として表わされるもの（たとえば、システムの平均応答時間T_{mr}= $N/(\Sigma_{j\in N}\ u_j U_j) - 1/v$)が多い。そのような性能指標を最適化する問題は、前節の結果より、性能指標を最適化する性能ベクトルを求める問題に帰着される。今、f(U_1, U_2, ..., U_N)で表わされる性能指標を考えよう。この性能指標の最適化問題は次のように表現される：

制約条件$\Sigma_{j\in N}\ U_j$ = U(N) および
Nの任意の空でない真部分集合Zに対して$\Sigma_{j\in Z}\ U_j \leq$ U(Z) の下で
A = f(U_1, U_2, ..., U_N) を最小化する。

f(U_1, U_2 ..., U_N) が凸であるとしよう。このとき、この問題は、凸計画となり、局所最小値が最適値となる。さらに、f(•)がすべてのU_iについて微分可能であるとすると、最適解が次のようなKuhn-Tucker 条件によって特徴付けられる(たとえば、Shapiro[1979] 参照)：

$\partial f(U_1,U_2,...,U_N)/\partial U_j + y_N + \Sigma_{Z\ni j,\ Z\subset N,\ Z\geq 1}\ y_Z = 0,\quad j\in N$
$\Sigma_{j\in N}\ U_j = U(N)$, かつNの任意の空でない真部分集合Zに対して、
$y_Z \geq 0,\ \Sigma_{j\in Z}\ U_j \leq U(Z),\ y_Z[\ \Sigma_{j\in Z} U_j - U(Z)] = 0,$

ただし、y_Nとy_ZはLagrange乗数をあらわす。

5．結言

求めた結果は次のようである：ここで述べたfinite-source queueing modelにおいて、任意の利用率ベクトルUを実現するスケジューリング方式が存在するための必要十分条件は、Uが次の条件を満たすことである：

$\Sigma_{j\in N} U_j = U(N)$　かつNの任意の空でない部分集合Zに対して

$\Sigma_{j\in N} U_j \leq U(Z)$　ただし、U(N)，U(Z)は(2.2)で与えられる。

応答時間ベクトルTが実現可能であるための条件は、上述の条件と（2.1)とから直ちに求められる。

参考文献

Coffman,E.G., Jr., and Mitrani, I. 1980. A characterization of waiting time performance realizable by single-server queues. Operations Resarch 28 3, Part II, 810-821.

Gelenbe, E., and Mitrani, I. 1980. Analysis and Synthesis of Computer Systems. Academic Press, London

Kameda, H. 1982. A finite-source queue with different customers. J. Assoc. Comput. Mach. 29, 2, 478-491.

Shapiro, J. F. 1979. Mathematical Programming: Structures and Algorithms Wiley, New York.