アルゴリズム 3－2
（1988．9．22）

動的ジョブに対するスケジュール長最小化のためのアルゴリズム

川口剛† 喜屋武盛基† 榎原博之†† 中野秀男†† 中西義郎††

†琉球大学工学部 ††大阪大学工学部

あらまし

発生時刻の異なるジョブを、スケジュール長が最小となるように$m(\geq 1)$台の均質プロセッサにスケジュールする問題が考察される。各ジョブi（$1\leq i\leq n$）は時刻a_iに発生し、その処理のためにp_i時間要す。またジョブは互いに独立である。この問題で特にすべてのa_iが等しい場合に対しては、これまでに多くのヒューリスティック・アルゴリズムが提案されている。しかし、a_iが任意の値をもつ場合に対しては、得られるスケジュールの良さが理論的に保証されるアルゴリズムは報告されていない。本論文では、a_iが任意の値をもつ場合に対して、$O(n\log n)$の時間計算量をもつヒューリスティック・アルゴリズムを提案する。そして、このアルゴリズムによって得られるスケジュールのスケジュール長$C(S_A)$と最適スケジュールのスケジュール長$C(S_0)$の相対比$C(S_A)/C(S_0)$が、最悪の場合でも$2-1/m$以下となることを示す。また$C(S_A)$と$C(S_0)$の差が、最悪の場合でもp_iの最大値以下となることを示す。

Minimizing Schedule-Length for Jobs with Release Times

Tsuyoshi KAWAGUCHI† Seiki KYAN†

Hiroyuki EBARA†† Hideo NAKANO†† Yoshiro NAKANISHI††

†University of the Ryukyus

††Osaka University

Abstract

In this paper we study the problem of nonpreemptively scheduling jobs on m identical processors so as to minimize schedule-length. Jobs have release times and processing times. There exists no precedence relation among jobs. An $O(n\log n)$ algorithm is presented for the problem. Let $C(S_A)$ denote the schedule-length obtained by the proposed algorithm and $C(S_0)$ be that of an optimal schedule. We show that the ratio $C(S_A)/C(S_0)$ is bounded by $2-1/m$ and the difference between $C(S_A)$ and $C(S_0)$ does not exceed the processing time of the largest job.

1．まえがき

本論文では、発生時刻の異なるジョブを、スケジュール長が最小となるようにm(≧1)台の均質プロセッサにスケジュールする問題が考察される。この問題は次のように記述される（図１参照）。

（A・1）　各ジョブｉは時刻a_iに発生し、任意の１台のプロセッサでの処理を要求する。

（A・2）　ジョブは、一度プロセッサを割り当てられると、処理が完了するまで中断できない（ジョブはnonpreemptive である）。そして、各ジョブｉは､プロセッサで処理を開始されてから、完了するまでにp_i時間要す。p_iをジョブｉの処理時間と呼ぶ。

（A・3）　ジョブ間には先行関係は存在しない（ジョブは互いに独立である）。

（A・4）　スケジュール長（最初のジョブが発生してから、最後のジョブが完了するまでの時間)を最小にするスケジュールを求める。

上記の（A・1）～（A・4）で規定される問題を、本論文ではＣＭＡＸ問題と呼ぶ。ＣＭＡＸ問題の１つの問題例と、この問題例に対する１つのスケジュールを図２に示す（図２に示されたスケジュールはジョブを発生順に処理するスケジュールである）。

ＣＭＡＸ問題は多くのスケジューリング問題の中でも最も注目を集めてきた問題の１つである[1]。特に、すべてのジョブの発生時刻が等しい場合に対しては多くのヒューリスティック・アルゴリズムが提案されている。文献（２），（３）には､最悪相対比 $2-1/m$, $4/3-1/3m$, $1.2+2^{-k}$ のアルゴリズムが示されている。また、これらのアルゴリズムの時間計算量は、それぞれ$O(n)$，$O(n\log n)$，$O(n\log n+kn\log m)$である。但し､アルゴリズムAの最悪相対比とは、Aによって得られるスケジュールのスケジュール長 $C(S_A)$と最適スケジュールのスケジュール長 $C(S_o)$の相対比 $C(S_A)/C(S_o)$ のすべての問題例に対する最大値をいう。

本論文では、ジョブの発生時刻a_iや処理時間p_iが任意の値をもつＣＭＡＸ問題に対して、時間計算量$O(n\log n)$のアルゴリズムを提案する。そして、このアルゴリズムによって得られるスケジュールのスケジュール長 $C(S_A)$と最適スケジュールのスケジュール長 $C(S_o)$の間に

$$C(S_A)/C(S_o) \leqq 2-1/m \qquad (1)$$

$$C(S_A)-C(S_o) \leqq P \qquad (2)$$

が成り立つことを示す。但し P は次式で定義される値である。

$$P=\frac{(m-1)}{m}\cdot\min\left(\max_{1\le i\le n}\{p_i\},\ \frac{1}{m}\sum_{i=1}^{n}p_i\right) \qquad (3)$$

式（１）は最悪相対比に関する結果である。また式（２）から、$C(S_A)$と$C(S_o)$の差は処理時間p_iの最大値以下であることがわかる。

更に、上記の結果を導出する過程の中で、納期d_i、処理時間p_iをもつジョブに対する最大納期ずれ最小化問題、および、最大納期遅れ最小化問題が考察される。そして納期の非減少順スケジュールの最大納期ずれ $L(S_E)$と最適スケジュールの最大納期ずれ $L(S_o)$の間に

$$L(S_E)-L(S_o) \leqq P \qquad (4)$$

が成り立つことを示す。また､納期の非減少順スケジュールの最大納期遅れ $T(S_E)$と最適スケジュールの最大納期遅れ $T(S_o)$の間に

$$T(S_E)-T(S_o) \leqq P \qquad (5)$$

が成り立つことを示す。但し、ジョブｉの完了時

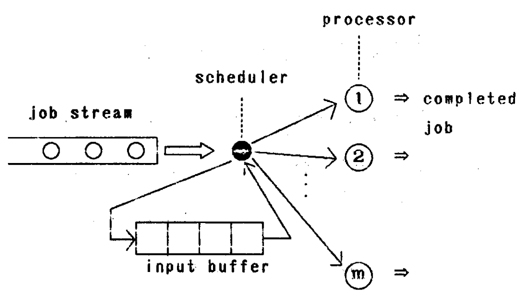
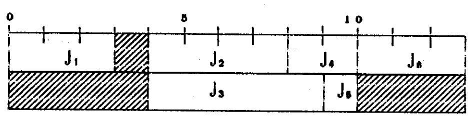


図１．スケジューリング・モデル

J_i	J_1	J_2	J_3	J_4	J_5	J_6
P_i	3	4	5	2	1	3
a_i	0	4	4	6	8	8

(a) 問題例



(b) 先着順スケジュール（スケジュール長 13）

図２．CMAX問題とスケジュールの例

刻をf_iとするとき、$L_i = f_i - d_i$で定義されるL_iをジョブｉの納期ずれと呼ぶ。また、$f_i > d_i$ならば $T_i = f_i - d_i$、 かつ 、 $f_i \leqq d_i$ ならば$T_i = 0$で定義されるT_iをジョブｉの納期遅れと呼ぶ。そして、すべてのジョブについての L_iの最大値が最大納期ずれであり、 T_iの最大値が最大納期遅れである。

２．準備

この章では、本論文を通して用いる記号を定義するとともに、式（１）～（５）を導出するために必要な従来の研究結果を記述する。

［記号の定義］

- n： ジョブ数
- m： プロセッサ数
- a_i： ジョブｉの発生時刻（３.で考察する問題ではすべてのa_iは等しく、４.で考察する問題では a_iは任意の実数値をとる）。なお、最初に発生するジョブの発生時刻（a_iの最小値）を時刻０と定義する。
- d_i： ジョブｉの納期（３.で考察する問題でのみジョブは納期をもち、４. で考察する問題ではジョブは納期をもたない）。
- p_i： ジョブｉの処理時間（任意の実数値）。
- P： 式（３）で定義される値。すなわち、p_iの最大値と$\Sigma p_i / m$のうち大きくない方の値を$1-1/m$ 倍した値。
- $f_i(S)$： スケジュールＳによって決まるジョブｉの完了時刻。スケジュールが明らかな場合はＳは省略される。
- $s_i(S)$： スケジュールＳによって決まるジョブｉの開始時刻。
- $C(S)$： スケジュールＳのスケジュール長。
- $L(S)$： スケジュールＳの最大納期ずれ。
- $T(S)$： スケジュールＳの最大納期遅れ。

また文献（２）では、すべてのジョブの発生時刻が等しいＣＭＡＸ問題に対して、次のアルゴリズムが提案されている。

ＬＩＳＴ（List Scheduling）法

Ｓｔｅｐ１．ジョブを適当な順序に並べたリストをつくる。

Ｓｔｅｐ２．あるプロセッサが空き状態になるたびに、リストの先頭のジョブを取り出し、このプロセッサに割り当てる。

ＬＩＳＴ法は、Ｓｔｅｐ１ でジョブをどのような順序で並べるかによって、色々なスケジュールを生成することができる。これらのスケジュールを総称してリストスケジュールと呼ぶ（例えば３．で述べるＥＤＤスケジュールもリストスケジュールの一種である）。任意のリストスケジュールに対して次の結果が文献（２）に示されている。

［補題１］　すべてのジョブの発生時刻が等しいＣＭＡＸ問題に対する任意のリストスケジュールをS_Lで表す。またスケジュール長を最小にするスケジュールをS_0で表す。このとき次式が成り立つ。

$$C(S_L) - C(S_0) \leqq P \tag{6}$$

また、次式が成り立つ。

$$C(S_L) / C(S_0) \leqq 2 - 1/m \tag{7}$$

更に式（６），（７）を等号で成り立たせる例が存在する。

なお、文献（２）には、ジョブ間に任意の先行関係が存在する場合でも式（７）が成り立つことが示されている。

任意のプロセッサ数ｍに対して、式(6),(7)を等号で成り立たせるジョブ集合$J = \{J_1, \cdots, J_n\}$が次のようにつくれる。

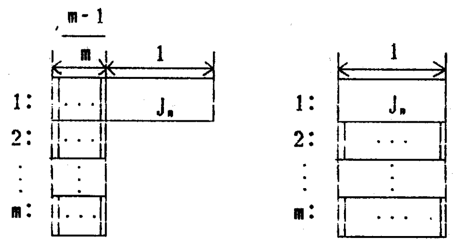
（ｉ）　ジョブ数ｎを$n=(m-1)N+1$で与える。但し、Ｎはｍの任意の倍数とする。

（ii）　そしてジョブJ_iの処理時間p_iを

$$p_i = \begin{cases} 1/N & (1 \leqq i \leqq n-1) \\ 1 & (i = n) \end{cases}$$

で与える。

このジョブ集合に対して、番号の小さい順にジョブを並べた後ＬＩＳＴ法を適用すると、図３(a)の



（a） S_L　　（b） S_0

図３．式(6),(7)を等式で成り立たせる例

スケジュールS_Lが得られる。一方、最初にJ_nを処理すると図3(b)のスケジュールS_0が得られる。そして図3のS_LとS_0の間では、式（6），（7）が等号で成り立つ。

3．最大納期ずれ、最大納期遅れ最小化問題

この章では、納期d_i、処理時間p_iをもつジョブに対して最大納期ずれ（又は最大納期遅れ）を最小にする問題が考察され、式（4）、（5）が導出される。

これらの問題は次の（B．1）～（B．4）によって規定される。

（B．1）　すべてのジョブは時刻0に発生し、任意の1台のプロセッサでの処理を要求する。

（B．2）
（B．3）｝（A．2）～（A．3）と同じ

（B．4）　各ジョブiは納期d_iをもつ。そして最大納期ずれ（又は最大納期遅れ）を最小にするスケジュールを求める。

上記の（B．1）～（B．4）で規定される問題のうち、最大納期ずれを最小にする問題をLMAX問題と呼び、最大納期遅れを最小にする問題をTMAX問題と呼ぶ。

次に述べるアルゴリズムはLMAX問題とTMAX問題の両方に利用できる。このアルゴリズムをEDD（Earliest Due Date First）法と呼ぶ。

<u>EDD法</u>

Step1.　納期d_iの非減少順にジョブを並べたリストをつくる。但し、d_iが等しいジョブの間では任意の順序とする。

Step2.　あるプロセッサが空き状態になるたびに、リストの先頭のジョブを取り出し、このプロセッサに割り当てる　□

EDD法から得られるスケジュールを<u>EDDスケジュール</u>と呼ぶ。EDDスケジュールに関して次の定理が成り立つ。

［定理1］　EDDスケジュールをS_Eで表し、最大納期ずれを最小にするスケジュールをS_0で表す。このとき次式が成り立つ。

$$L(S_E)-L(S_0)\leqq P \qquad (8)$$

また、上式を等号で成り立たせる例が存在する。　□

図3に示したジョブ集合で、すべてのジョブが等しい納期dをもつとき、式（8）が等号で成り立つ（dは任意の値でよい）。また式（8）は、以下に述べる補題2～4を用いて証明できる。

［補題2］　すべてのd_iが等しいジョブ集合に対しては、式（8）が成り立つ。

（証明）　$d_i=d\ (1\leqq i\leqq n)$とおき、

$$\Delta=L(S_E)-L(S_0)$$

とおくと、次式が得られる。

$$\Delta=C(S_E)-d-\{C(S_0)-d\}$$
$$=C(S_E)-C(S_0)$$

更に、S_Eはリストスケジュールの一種であるので、補題1を用いて$C(S_E)-C(S_0)\leqq P$がいえ、$\Delta\leqq 0$が成り立つ。　（証終）

［補題3］　Rを次式によって定義する。

$$R=\{L(S_E)-L(S_0)\}/P \qquad (9)$$

そしてRを最大にするジョブ集合の中で、ジョブ数が最小のジョブ集合を$J=\{J_1,\cdots,J_n\}$とし、$d_1\leqq\cdots\leqq d_n$と仮定する。このとき、Jに対するEDDスケジュールにおいて最大納期ずれを与えるジョブはJ_nである。

（証明）　Jに対するEDDスケジュールをS_Eで表す。以下、$L(S_E)$を与えるジョブがJ_i（$i<n$）であると仮定し矛盾を導く。

ジョブ集合J'を　$J'=J-\{J_n\}$によって定義する。またS_EからJ_nを取り除いて得られるスケジュールをS_E'で表す。このS_E'は、J'に対する1つのEDDスケジュールである。更にJに対する最適スケジュールとPの値をS_0およびPで表し、J'に対する最適スケジュールとPの値をS_0'およびP'で表す。

まず、$L(S_E)$を与えるジョブはJ_nではないので$L(S_E)=L(S_E')$がいえる。更にJ'の定義より

$$L(S_0')\leqq L(S_0) \quad かつ \quad P'\leqq P$$

が成り立つことは明らかである。このようにして、

$$\frac{L(S_E)-L(S_0)}{P}\leqq\frac{L(S_E')-L(S_0')}{P'}$$

が得られる。S_E'はJ'に対するEDDスケジュールである。それゆえ上式は、J'に対するRの値がJに対するRの値より小さくないことを意味する。これは、Jが、Rの値を最大にするジョブ数最小のジョブ集合であるという仮定に矛盾する。

このようにして補題が成り立つ。　（証終）

［補題４］　式（９）のＲを最大にするジョブ集合の中の少なくとも１つのジョブ集合は

$$d_1 = d_2 = \cdots = d_n \qquad (10)$$

を満たす。

（証明）　Ｒを最大にするジョブ集合を $J = \{J_1, \cdots, J_n\}$ で表し、各 $J_i (1 \leq i \leq n)$ の処理時間と納期をそれぞれ p_i および d_i で表す。以下、Ｊにおいて、式（１０）が成り立たない場合には、Ｊよりも小さくないＲの値をもちしかも式（１０）を満たすジョブ集合が存在することを示す。

一般性を失うことなく、$d_1 \leq \cdots \leq d_n$ と仮定できる。また $d_1 = d_i$ を満たす最大のｉをｇで表す。Ｊの仮定によって $g < n$ であるので、

$$d_1 = \cdots = d_g < d_{g+1} \leq d_{g+2} \leq \cdots \leq d_n$$

が成り立つ。更に $J' = \{J_1', \cdots, J_n'\}$ とし、ジョブ $J_i' (1 \leq i \leq n)$ は J_i と同じ処理時間 p_i および次式で定義される納期 d_i' をもつとする。

$$d_i' = \begin{cases} d_{g+1} & (1 \leq i \leq g) \\ d_i & (g+1 \leq i \leq n) \end{cases} \qquad (11)$$

そして、スケジュール S_E、S_0、S_E'、S_0' を次のように定義する。

S_E ：Ｊに対するＥＤＤスケジュール。

S_0 ：Ｊに対する最適スケジュール。

S_E'：S_E の中の各 $J_i (1 \leq i \leq n)$ をそれぞれ J_i' で置き換えて得られるスケジュール（S_E' は J_i' に対する１つのＥＤＤスケジュールである）。

S_0'：Ｊ' に対する最適スケジュール。

補題３によって、一般性を失わずに、$L(S_E)$ を与えるジョブは J_n であると仮定できる。また S_E' の定義から、S_E における J_n の完了時刻は、S_E' における J_n' の完了時刻に等しい。更に式（１１）から $d_n' = d_n$ である。それゆえ $L(S_E') = L(S_E)$ が成り立つ。

また、すべての $J_i'(1 \leq i \leq n)$ の納期は J_i の納期以上であるので $L(S_0') \leq L(S_0)$ がいえる。しかもＰの値は、ＪとＪ' で同じである。

それゆえ、Ｊ' に対するＲの値は、Ｊに対するＲの値より小さくないことになる（ＪはＲを最大にするジョブ集合であるので、ＪとＪ' のＲの値は等しいことになる）。しかも、$d_i' (1 \leq i \leq n)$ の中の異なる値の数は、$d_i (1 \leq i \leq n)$ の中の異なる値の数より１小さい。従って、上記の議論を繰り返すことによって補題を得る。（証終）

補題２と補題４から、ＬＭＡＸ問題の任意の問題例に対して式（８）が成り立つことがいえる。

またＴＭＡＸ問題に対して次の定理が成り立つ。この定理は、定理１の証明と同様な手順により証明できる。

［定理２］　ＥＤＤスケジュールを S_E で表し、最大納期遅れを最小にするスケジュールを S_0 で表す。このとき次式が成り立つ。

$$T(S_E) - T(S_0) \leq P \qquad (12)$$

また、上式を等号で成り立たせる例が存在する。

□

図３に示したジョブ集合で、すべてのジョブが等しい納期ｄをもつとき式（１２）が等号で成り立つ（ｄは任意の値でよい）。

４．スケジュール長最小化問題

この章では、１．で述べたＣＭＡＸ問題に対して、１つのヒューリスティック・アルゴリズムを提案するとともに、このアルゴリズムの性能を評価する。

<u>アルゴリズムＡ</u>

｛発生時刻 a_i、処理時間 p_i をもつジョブ J_i $(1 \leq i \leq n)$ に対して、スケジュール長を最小にするスケジュールを求めるためのヒューリスティック・アルゴリズム｝

Ｓｔｅｐ１．　各ジョブ J_i の仮想納期 d_i を $d_i = -a_i$ で与える。

Ｓｔｅｐ２．　各ジョブ J_i が納期 d_i、発生時刻０をもつと仮定して、ＥＤＤ法を適用する。このとき得られるスケジュールを S_E で表す。

Ｓｔｅｐ３．　S_E におけるジョブ $J_i (1 \leq i \leq n)$ の完了時刻 $f_i(S_E)$ と最大納期ずれ $L(S_E)$ を求める。

Ｓｔｅｐ４．　｛スケジュール長を最小にするスケジュールへの変換｝
ジョブ $J_i (1 \leq i \leq n)$ の開始時刻 s_i を $s_i = L(S_E) - f(S_E)$ で与える。そして、すべての J_i を、S_E と同じプロセッサ上で時刻 s_i に開始させる（S_E 上のすべてのジョブを負の時刻の方向に $L(S_E)$ だけ並行移動し、その後、時刻０を軸とし

て折り返すことに対応する）。

□

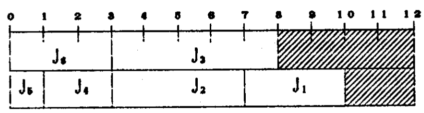
アルゴリズムAの適用例を図４に示す。プロセッサの数$m=2$とし、図４（a）に示されたジョブ集合に対してアルゴリズムAを適用すると、まずStep２で図４（b）のスケジュールが得られる。そしてStep３で$L(S_E)=12$が得られる（J_3が$L(S_E)$を与える）。更にStep４で図４（c）のスケジュールが得られる（すべてのジョブが a_i以後に開始されていることに注意せよ）。このスケジュールのスケジュール長は$L(S_E)=12$である（Step４で得られるスケジュールのスケジュール長は常に $L(S_E)$ に等しい）。

アルゴリズムAが $O(n\log n)$ 時間で実行できることは明らかである。更に、アルゴリズムAから得られるスケジュールに対して次の定理が成り立つ。

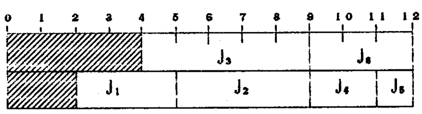
［定理３］　アルゴリズムAから得られるスケジュールS_Aは、CMAX問題に対する１つの可能スケジュール（すべてのジョブがa_i以後に開始されるスケジュール）である。また、CMAX問題の任意の問題例に対して、S_Aと最適スケジュールS_0の間に次式が成り立つ。

J_i	J_1	J_2	J_3	J_4	J_5	J_6	
p_i	3	4	5	2	1	3	入力データ
a_i	0	4	4	6	8	8	
$d_i=-a_i$	0	-4	-4	-6	-8	-8	→ 仮想納期

（a）　ジョブ集合



（b）　Step２で得られるスケジュールS_E



（c）　Step４で得られるスケジュールS_A

図４　アルゴリズムAの適用例

$$C(S_A)-C(S_0)\leqq P \qquad (13)$$

$$C(S_A)/C(S_0)\leqq 2-1/m \qquad (14)$$

更に式（13）、（14）を等号で成り立たせる例が存在する。

（証明）　任意の発生時刻a_iおよび任意の処理時間p_iをもつジョブ集合を$J=\{J_1,\dots,J_n\}$とする。またJに対して$J'=\{J_1',\dots,J_n'\}$を次のように定義する；J_i'は発生時刻０、処理時間p_i、納期$d_i=-a_i$をもつ（アルゴリズムAのStep１～３では、J_iが仮想納期d_iや仮想発生時刻０をもつと記述された。しかし定理３の証明の中では、混乱をさけるため、上記のようなJ'を定義しておく）。 そして、アルゴリズムAのStep２で得られるスケジュールS_Eを、J'に対するスケジュールと考える。

まず、すべての$i\ (1\leqq i\leqq n)$に対して

$$L(S_E)\geqq f_i(S_E)-d_i=f_i(S_E)+a_i$$

が成り立つ。またアルゴリズムAのStep４より、S_AにおけるJ_iの開始時刻$s_i(S_A)$は

$$s_i(S_A)=L(S_E)-f_i(S_E) \qquad (15)$$

を満たす。それゆえ、$s_i(S_A)\geqq a_i$がいえ、S_AはCMAX問題に対する可能スケジュールとなる。

更に、図３に示した問題例は、CMAX問題に対する１つの問題例（すべてのa_iが等しい問題例）であり、すでに述べたように式（13）、（14）を等号で成り立たせる。

従って、証明の残りの部分では、式（13）、（14）を導出する。

まずJに対する最適スケジュールS_0に対して、S_0'を次のように定義する。

S_0'：　S_0上のジョブ$J_i\ (1\leqq i\leqq n)$を各々J_i'で置き換えた後、すべてのJ_i'を負の時刻の方向に $C(S_0)$ だけ並行移動して得られるスケジュールを、更に時刻０を軸として折り返して得られるスケジュール。

このとき、S_0'におけるJ_i'の開始時刻$s(S_0')$は

$$s_i(S_0')=C(S_0)-f_i(S_0)$$

を満たす。よってS_0'におけるJ_i'の納期ずれ$L_i(S_0')$は

$$\begin{aligned} L_i(S_0') &= s_i(S_0')+p_i-d_i \\ &= s_i(S_0')+p_i+a_i \\ &= C(S_0)-f_i(S_0)+p_i+a_i \end{aligned}$$

で与えられ、$f_i(S_0) \geq p_i + a_i$であるので$L_i(S_0') \leq C(S_0)$がいえる。このようにして

$$L(S_0') \leq C(S_0) \quad (16)$$

が成り立つ。

また、式（15）の両辺にp_iを加えると$f_i(S_A) = L(S_E) - s_i(S_E)$であり、$s_i(S_E) \geq 0$を用いて$f_i(S_A) \leq L(S_E)$（$1 \leq i \leq n$）がいえる。特に$s_i(S_E) = 0$である$i$については、上式で等号が成り立つ。それゆえ

$$C(S_A) = L(S_E) \quad (17)$$

更に、式（16）、（17）より

$$C(S_A) - C(S_0) \leq L(S_E) - L(S_0') \quad (18)$$

であり、S_EはJ'に対するEDDスケジュール、S_0'はJ'に対するLMAX問題の1つの可能スケジュールである。それゆえ定理1を用いて、式（18）の右辺はP以下となる。このようにして式（13）が得られる。

また文献（2）において式（6）から式（7）を導かれたのと同じ手順で、式（13）から式（14）が導かれる。（証終）

5・むすび

本論文では、発生時刻が異なるジョブをスケジュール長が最小となるようにm台の均質プロセッサにスケジュールする問題に対して、時間計算量 O（n logn） のヒューリスティック・アルゴリズムを提案した。そして、このアルゴリズムから得られるスケジュール長と最適スケジュールのスケジュール長との差が、ジョブの処理時間p_iの最大値以下であることを示した。また、両者の相対比が 2-1/m 以下であることを示した。これらの結果は、ジョブの発生時刻や処理時間が任意の値をもつ場合に対して成り立つ。

なお、本論文で提案したアルゴリズムは、スケジューリング開始時点ですべてのジョブの発生時刻が既知であることを要求する。しかし例えば、個々のジョブが一定の時間間隔で発生するシステムやジョブ集合全体が繰り返し処理されるシステムなど、ジョブの発生時刻がスケジューリング開始時点に既知であるシステムも多い。提案したアルゴリズムは、これらのシステムにおける実用的スケジューリング法として利用できる。

また、式（2）から得られる$C(S_0) \geq C(S_A) - P$は、最適スケジュールのスケジュール長の下界値を与える。そして、この下界値は、スケジューリング開始時点でジョブの発生時刻が既知であるか否かにかかわらず、種々の実用的スケジューリング法から得られるスケジュール（例えば図2に示した先着順スケジュール）の良さを計算機シュミレーションによって評価するために利用できる。

文献

（1）R.L.Graham, E.L.Lawler, J.K.Lenstra and A.H.G.Rinnooy Kan: "Optimization and Approximination in deterministic sequencing and scheduling - a survey", Ann. Distcrete Math., Vol.5, pp.286-326 (1979).

（2）R.L.Graham: "Bounds on multiprocessing timing anomalies", SIAM J.Appl. Math. Vol.17, pp.416-429(1969).

（3）D.K.Friesen: "Tighter bounds for the MULTIFIT processor scheduling algorithm", SIAM J.Comput., Vol.13, pp.170-181(1984).