

# スケジューリング問題の近似解法

石井 博昭

## 1. はじめに

昭和60年度のプロ野球は阪神タイガースの日本1であげくれた。ところで野球の日程はどのように決められるのだろうか。終り頃になるとダブル・ヘッダーがあるかと思えば、あるチームは3日間ほど試合がなかったりして、おかしいと思われる方も多いと思う。おまけに雨で流れても空いている日に埋めないで後まわしになったりした。このあたりがスケジューリング問題のむずかしさをよく表わしており、よく調べると球場の移動とか次の対戦相手とかの関係でそうせざるを得ないことがわかる。このように6チームでもスケジューリングはむずかしい。

人、設備、機械等を用いて処理すべき対象物があるとする。前者を**機械**と呼び、後者を**仕事**と呼ぶことにする。1つ、あるいは複数台の機械を用いていくつかの仕事を処理する時、各仕事についてそれを処理する機械および各機械については処理する仕事の順序、より厳密に言えば、各機械上で各仕事が処理される時間区間を決定すること、すなわちスケジュールを決定することがスケジューリング問題である。

スケジュールを示すには Gantt chart が便利でよく用

いしい ひろあき

大阪大学 工学部 応用物理学科  
〒565 吹田市山田丘2-1

表1  $a_i$  は  $M_1$  での処理時間,  $b_i$  は  $M_2$  での処理時間を示す

|       | $J_1$ | $J_2$ | $J_3$ | $J_4$ | $J_5$ |
|-------|-------|-------|-------|-------|-------|
| $a_i$ | 10    | 9     | 4     | 12    | 7     |
| $b_i$ | 9     | 3     | 8     | 6     | 11    |

いられる(図1参照)。多くの場合、仕事間の先行関係や分割処理可能かどうか等の制約のもとで、全仕事の完了時間などの評価関数を最小にするスケジュールを求めなければならない。

最近のスケジューリング理論の発展にはコンピュータの飛躍的進歩に依るところが大であるが、逆にコンピュータ・システムの最適運用はスケジューリング理論の重要な応用分野となってきた。スケジューリング理論は1950年代の2機械フローショップ問題に対する Johnson [20] および1機械最大納期ずれ最小化問題に対する Jackson [18] の成功によりはじまった。しかし、1970年代初期の Cook [8] や Karp [22], [23] たちの仕事を嚆矢とする計算の複雑性の研究は、スケジューリング問題はその見かけの簡単さに較べて一般に非常にむずかしいことを明らかにした。現実

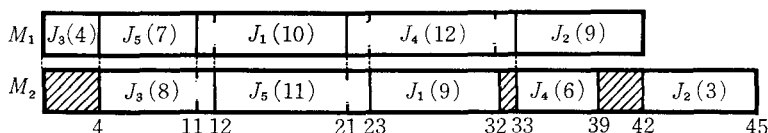


図1 表1で与えられる機械フローショップ問題に対する最適スケジュールを表わすガントチャート( $J_3(4)$ などの4は処理時間を示す)

発生するスケジューリング問題のほとんどが厳密解法には膨大な計算を要求するのが現状で、近似解法に頼らざるをえない。

近似解法の目的は妥当な計算時間で最適解に近い実行可能な解を求めることである。もちろん近似解法は常に最適解を与える保証はない。近似解法が作り出す近似解と最適解の近さは、目的関数の対応する値を比較して測られるべきである。しかしながら最適解が容易に求めがたいので近似解法に頼っているのであるから、きっちりした近さを測ること自身困難である。このことから、近似解法の性能は基本的には2つの方法で測られる。

1つは多数の問題例を近似解法で解くとともになんらかの方法で最適解を求めて、その誤差を計算する絶対評価法である。この絶対評価法では小規模な問題例にしか適用できない上、どのような問題例を解けばよいかも明白でない。もっと根本的な疑問点は、近似解法が必要である大規模な問題例にはたして小規模な問題例からの判断が通用するのかという点である。もう1つはその近似解法にとって最も都合の悪いと思われる問題例を考え、その時の相対誤差を評価する方法である。相対誤差の取り方にはいくつかあるが、よく用いられるのは次の2つ、またはその変形である。

- (a)  $\max_{Q(I) \in Q} |F^*(Q(I)) - \hat{F}(Q(I))| / F^*(Q(I))$
- (b)  $\max_{Q(I) \in Q} \hat{F}(Q(I)) / F^*(Q(I))$

ここで  $Q$  は、いま考えている問題<sup>3)</sup>、 $Q(I)$  問題例、 $F^*(Q(I))$  は  $Q(I)$  の最適値、 $\hat{F}(Q(I))$  は、いま考えている近似解法を用いた時の目標関数の値。(a)、(b)どちらの場合も厳密に求めることはむずかしい。どのようにシャープな上界を求めるかがポイントとなる。

また、スケジューリング問題の中には時間割問題等のように実行可能なスケジュールを求めること自身むずかしい場合もよく見られる。私の属する大阪大学工学部を例にとれば、ある先生の授業は朝一番ばかりという時間割をなかなか変えられないといったことになる。

ここでスケジューリング問題によく使われる基本的な考え方を述べよう。1つはリスト・スケジューリングである。これは、仕事をある基準にもとづいて順序をつけてリストに入れておき、この順に現在空いている機械に仕事を割当てていく方法で、基準の決め方により種々のスケジュールを生成する。もう1つは、いま考えている問題より単純な問題における最適解法を適当に修正して近似解法を作る方法である。

以下 2. では、スケジューリング問題の構成要素を系統的に示す。3. はビンパッキング問題の近似解法など4. への準備である。4. では、いくつかのスケジューリング問題に対して具体的に近似解法を紹介し、その相対誤差および計算手間を与える。5. はまとめである。

## 2. スケジューリング問題

仕事の集合を  $J = (J_1, \dots, J_n)$ 、機械の集合は  $M = (M_1, \dots, M_m)$  で示す。スケジューリング問題は、大別すると、機械が1台だけの時と複数台の時の2通りが考えられ、アプローチが異なってくる<sup>3)</sup>。通常、仕事は各機械ごとに処理内容が異なっている要素からなり、この要素を操作 (operation) と呼ぶ。仕事  $J_j$  に対して機械  $M_i$  での操作を  $O_{ij}$  で示し、その処理時間を  $p_{ij}$  で示す<sup>4)</sup>。複数台の機械がある時には、機械の機能によって大きく次の2つ (並列型、ジョブ型) に分けられる。その一方は並列型と呼ばれるものであり、各機械の機能が同じであって、仕事はどの機械で処理されてもよく、各仕事がただ1つの操作からなっている場合である。並列型はさらに次の3つに分けられる。(c)等価機械型 (d)一様機械型 (e)非一様機械型である。(c)は同じ機械が複数台ある場合で、(d)の特別な場合と考えられるが、近似解法は特にこの場合によく研究されている。(d)は同一の仕事でも処理時間は機械ごとに異なるが、その比は仕事によらず一定な場合である。すなわち各機械の性能が決まっていて、各仕事を処理するスピードが

機械ごとに異なっている。この機械ごとのスピードを  $s=(s_1, s_2, \dots, s_m)$  で示す。  $s_1=\dots=s_m=1$  の場合が(c)となる。(e)はこの形で最も一般の場合で、同一の仕事でも各機械ごとに処理時間が異なっていて、その比も必ずしも一定でない場合である。

もう一方は**ショップ型**と呼ばれるものであり、各機械の機能がすべて異なり、各仕事が複数の操作から構成されている場合である。ショップ型はさらに、すべての仕事でその操作が処理されていく機械の順序が同一の場合の**フロー・ショップ型**、各操作が処理されていく機械の順序が仕事ごとに決まっている場合の**ジョブ・ショップ型**、および各操作が処理されていく機械の順序がどの仕事においても自由である場合の**オープン・ショップ型**の3つに分類される<sup>5)</sup>。

次に、スケジューリング問題ではスケジュールを評価する判定基準が重要であり、次のようなものが一般的によく用いられる。ここで  $C_j$  は仕事  $J_j$  の完了時間、すなわち最後の操作の完了する時間を示す。(f)**最大完了時間**：すべての仕事の操作を完了するのに要する時間で、 $C_{\max}$  で示される。すなわち  $C_{\max} = \max_{1 \leq j \leq n} C_j$  である。(g)**最大納期ずれ(最大納期遅れ)**： $C_j$  から  $J_j$  の納期  $d_j$ <sup>6)</sup> を差し引いた時間の  $J_j$  についての最大値(納期から遅れた時間の最大値)で  $L_{\max}$  で示される。 $L_{\max} = \max_{1 \leq j \leq n} (C_j - d_j)$  である。(h)  $\sum_{j=1}^n C_j/n$ ：平均流れ時間。(i)**遅れ仕事数**。

スケジューリング問題の目的は以下で述べる条件を満足し上記の評価尺度の1つまたはいくつかに関する目標関数を最小にするスケジュールを求めることである。以下制約条件について述べる。

一般に、仕事間には技術的制約などにもとづく**先行関係**が存在する。先行関係は図2のような有向グラフで示すことができる。 $J_i \rightarrow J_j$  は仕事  $J_j$  は  $J_i$  が完了するまで処理を開始できないことを示し<sup>7)</sup>、●は全仕事の開始時点、◎は全仕事の終了時点を示す節点である。そして、節点○上の数字は所要時間を示している。また、仕事の内容によ

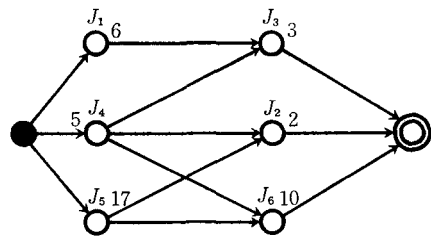


図2 先行関係のグラフ

っては、途中で処理を中断しても後で中断した所から開始できる場合、すなわち**分割処理(preemption)**が可能な場合がある。この場合、問題が比較的とり扱いやすくなるが分割処理が不可能でない(nonpreemptive)場合は途端にむずかしくなる。以下では特に断わらないかぎり各仕事は非分割とする。さらに、各仕事には**処理開始可能な最早時間  $r_j$** が決まっていたり、**納期  $d_j$** と呼ばれるそれまでに処理を完了すべき時間が決まっていたりする。最後にすべてのスケジューリング問題において常に前提とされる2つの仮定で2.を締めくくる。(j)各仕事は2つ以上の機械で同時処理は許されない。(k)各機械は一度には2つ以上の仕事を処理できない。

### 3. アルゴリズムの効率性およびビンパッキング問題

問題例の規模は通常その問題例をコンピュータ入力するのに要するデータの長さで評価される。人によって種々の考え方があると思うが、アルゴリズムがある問題を効率的に解くというのは問題例の規模の高々多項式時間のオーダー<sup>8)</sup>の計算手間(規模を  $l$  とすると  $O(l)$ ,  $O(l^2)$ ,  $O(l^{2.5})$ ,  $O(l \log l)$ <sup>9)</sup>等)で解を見いだす場合である。

次に並列機械問題と密接な関係があるビンパッキング問題について考察する。

**ビンパッキング問題**：容量  $C$  をもつビン<sup>10)</sup>とこれらのビンに詰められるべき品物のリスト  $L=(I_1, I_2, \dots, I_n)$  がその大きさ  $C \geq v(I_j) \geq 0, j=1, 2, \dots, n$  とともに与えられている。この時  $\sum_{I_i \in B_j} v(I_i)$

$\leq C$  を満足するリスト  $L$  の分割  $L=B_1 \cup \dots \cup B_m$  が存在する最小の  $m$  およびその分割を求める, すなわち各  $B_i$  は容量  $C$  の  $i$  番目のビンの内容と考えると,  $L$  をつめるのに必要なビンの最小数およびその詰め方を求める問題 ■

このビンパッキング問題に対する近似解法はいくつかあるが, 特に次の FFD (First Fit Decreasing 法) は並列機械スケジューリング問題においてよく用いられる.

**FFD**:  $v(I_1) \geq \dots \geq v(I_n)$  となるように  $L$  での順番を入れかえる. 最初すべてのビン は空であるとして  $L$  での品物の順番にそれが入り得る一番番号の小さいビンに詰めていく ■

図3は  $n=30, C=40$ , 大きさ;  $21 \times 6$  個,  $12 \times 6$  個,  $11 \times 6$  個,  $8 \times 12$  個に対する FFD による結果を示し, 図4は対応する最適パッキングを示す. FFD によって必要となるビンの数を  $Z_{\text{FFD}}$ , 最適パッキングによるビンの数を  $Z_{\text{opt}}$  とすると,

$$Z_{\text{FFD}} \leq (11/9)Z_{\text{opt}} + 4$$

が成り立つことを Johnson [21] が示した. 計算時間は  $O(n \log_2 n)$  である. 他の近似解法については Garey & Johnson [14] を見られたい.

## 4. スケジューリング問題に対する近似解法

### 4.1 1 機械問題に対する近似解法

仕事  $J_1, J_2, \dots, J_n$  を 1 機械で処理する.  $J_i$  の最早開始可能時間  $r_i$ , 納期  $d_i$ , 処理時間を  $p_i$  とし完了時間を  $C_i$  で示す.

**Q1**:  $L_{\max}(\equiv \max_i (C_i - d_i))$  を最小にするスケジュールを求めよ. ■

Q1 に対して次の 2 つの近似解法が知られている.

**A1**: 納期を非減少順に並び変えて  $d_{\pi(1)} \leq \dots \leq d_{\pi(n)}$  となったとすると,  $J_{\pi(1)} \rightarrow J_{\pi(2)} \rightarrow \dots \rightarrow J_{\pi(n)}$  の順に仕事を処理せよ ■

A1 は  $r_1 = \dots = r_n$  である, すなわち実際上納期による制約しかない問題に対しては最適スケジュールを与える (Jackson [18]). (a) 型の相対誤差

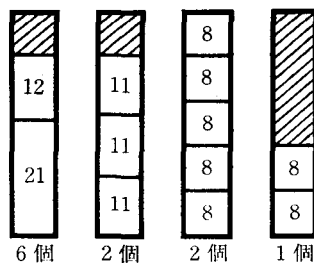


図3 FFD による結果  
(全部で11個必要)

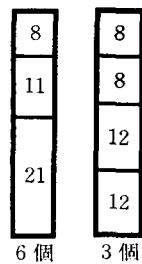


図4 最適パッキング (9 瓶必要)

に対する上界は  $1 - (1/\sum_{i=1}^n p_i)$  である.

**A2**:  $r_i$  が最小の仕事からはじめる. 処理中の仕事が完了するごとに, その時点までにははや開始時間がきている未処理の仕事の中から納期が最小の仕事を次に処理するというようにスケジュールを決める. ■

A2の(a)型の相対誤差の上界は  $1 - (3/\sum_{i=1}^n p_i + 1)$  である (Schrage [39]). また A1, A2 とも計算時間は  $O(n \log n)$  である.

### 4.2 並列機械問題に対する近似解法

**Q2**(等価並列機械問題): 同一の機械  $M_1, M_2, \dots, M_n$  とどれか 1 つで各仕事  $J_1, J_2, \dots, J_n$  を処理して, 最大完了時間を最小にするスケジュールを求めよ. ■

Q2は分割処理を許すと簡単に最適解が求まり, その時の  $C_{\max} = \max(\sum_{i=1}^n p_i/n, \max_{1 \leq i \leq n} p_i)^{11)}$  となる.  $p_i$  は  $J_i$  の処理時間である. ところが非分割処理とすると途端にむずかしくなるが, 次の LPT (Longest processing time) アルゴリズムと MFD (Multi-fit decreasing) アルゴリズムは良い近似解法となる.

**A3**(LPTアルゴリズム): 処理時間の非減少順, すなわち LPT ルールにしたがってリストを作り, リスト・スケジューリングを行なえ (リストに並んでいる順に仕事を空いている機械に割り当てていく). ■

例:  $m=30, n=21$ , 処理時間  $p_i$ ; 19, 18, 17, 16, 15, 14, 13, 12, 11 各 2 個と 10 の仕事 3 個. これを

|               |    |    |    |
|---------------|----|----|----|
| $M_1$         | 19 | 10 | 10 |
| $M_2$         | 19 | 10 |    |
| $M_3$         | 18 | 11 |    |
| $\vdots$      |    |    |    |
| $M_9$         | 15 | 14 |    |
| $M_{10}$      | 15 | 14 |    |
| $C_{\max}=39$ |    |    |    |

|               |    |    |    |
|---------------|----|----|----|
| $M_1$         | 19 | 11 |    |
| $M_2$         | 19 | 11 |    |
| $M_3$         | 18 | 12 |    |
| $\vdots$      |    |    |    |
| $M_9$         | 15 | 15 |    |
| $M_{10}$      | 10 | 10 | 10 |
| $C_{\max}=30$ |    |    |    |

|               |    |    |    |
|---------------|----|----|----|
| $M_1$         | 19 | 15 |    |
| $M_2$         | 19 | 10 |    |
| $M_3$         | 18 | 11 |    |
| $\vdots$      |    |    |    |
| $M_9$         | 10 | 14 | 10 |
| $M_{10}$      | 15 | 14 |    |
| $C_{\max}=34$ |    |    |    |

図 5 (a) LPT アルゴリズムによるスケジュール ( $\mathcal{L}^*$  に決まる) (b) 最適スケジュール (19 と 11, 18 と 12  $\dots$  15 と 15,  $10 \times 3$  と組み合わせる) (c) 修正 LPT アルゴリズムによるスケジュール ((a) での  $M_9$  の 15 と  $M_{10}$  の  $10 \times 2$  個を入れかえたスケジュール)

図 5

LPT ルールで並べると 19, 19, 18, 18,  $\dots$ , 10, 10, 10 となる。結果は図 5(a) となる。比較のため、最適スケジュールを図 5(b) で、後に触れる修正 LPT アルゴリズムの結果を図 5(c) で示す。A3 による (b) 型の相対誤差は  $(4/3) - (1/3m)$  以下となり、計算手間は  $O(n \log n)$  となる。最近 M. A. Langston [29] は A3 を改良し、計算手間は  $O(kn \log m)$  で (b) 型の相対誤差が  $(5/4) + (1/12)2^{-k}$  以下であるアルゴリズムを考え出した。その基本的考え方は A3 でうまくゆかない場合を解析し、この都合の悪い場合に適当に仕事を入れかえてスケジュールを改良しようというものである。 $k$  はこの入れかえの回数に対応するパラメータである。この考え方は興味深いが詳細はかなり複雑であるのでここでは割愛する。

一方、MFD アルゴリズムは以下のようにビンパッキング問題に対する FFD (First fit decreasing) アルゴリズムを十分に利用する (Friesen [11])。

**A4 (MFD アルゴリズム):** 各機械をビンと各仕事  $J_i$  を大きさ  $v(I_i) = p_i$  である品物  $I_i$  と考える。時間  $C$  までにすべての仕事が完了することは、容量  $C$  の  $m$  個のビンにこれらの品物を詰めることに成功

することを意味する。最初  $LOW = \lfloor \sum_{i=1}^m p_i / m \rfloor^{12}$ ,  $UP = 2LOW$  とおく。この  $LOW$  から  $UP$  までの範囲で  $C$  を変えて各容量をもつビンパッキング問題が FFD を用いて詰められるかどうか調べる。2 分割法で最小整数の成功容量  $C$  でのパッキングに対応するスケジュールを近似解として採用する。最初  $C = \lfloor (UP + LOW) / 2 \rfloor$  を調べ、成功した時  $UP =$  現在の  $C$ 、失敗したとき  $LOW =$  現在の  $C$  とする。この改訂された  $LOW$ ,  $UP$  から再び  $C = \lfloor (LOW + UP) / 2 \rfloor$  として調べるといいうようにくりかえす。■

A3 に対する例題に A4 を適用する。最初の  $LOW = \lfloor (\sum p_i) / 2 \rfloor = 30$ ,  $UP = 2 \times LOW = 60$ , 最初  $C = \lfloor (30 + 60) / 2 \rfloor = 45$  である。 $C = 45$  では明らかに FFD で詰めるのに成功する。したがって  $UP =$  現在の  $C = 45$  として  $C = \lfloor (45 + 30) / 2 \rfloor = 37$  とする。この場合も成功する。次に  $C = \lfloor (37 + 30) / 2 \rfloor = 33$  とすると図 6(a) のようなスケジュールが得られる。さらに  $C = 31$ , 最後に  $C = 30$  でも成功するので A4 は最適スケジュール (図 6(b)) を与える。A4 の (b) 型の相対誤差の上限は  $13/11$  以上  $6/5$  以下であることが知られている。A4 の計算手間は  $O(\log p_{\max} \times n \log n)$  ( $p_{\max} = \max_i p_i$ ) である。

|          |    |    |
|----------|----|----|
| $M_1$    | 19 | 14 |
| $M_2$    | 19 | 14 |
| $M_3$    | 18 | 15 |
| $M_4$    | 18 | 15 |
| $M_5$    | 17 | 16 |
| $M_6$    | 17 | 16 |
| $M_7$    | 13 | 13 |
| $M_8$    | 12 | 12 |
| $M_9$    | 11 | 11 |
| $M_{10}$ | 10 | 10 |

|          |    |    |
|----------|----|----|
| $M_1$    | 19 | 11 |
| $M_2$    | 19 | 11 |
| $M_3$    | 18 | 12 |
| $M_4$    | 18 | 12 |
| $M_5$    | 17 | 13 |
| $M_6$    | 17 | 13 |
| $M_7$    | 16 | 14 |
| $M_8$    | 16 | 14 |
| $M_9$    | 15 | 15 |
| $M_{10}$ | 10 | 10 |

図 6(a)  $C=33$  に対する FFD の結果 (b)  $C=30$  での成功すなわち最適スケジュール

処理時間の最大値と最大値の比が限定された  $Q_2$  の部分クラスの問題に対する近似解法は

Achugbue & Chin [1] が与えている。また、処理時間が  $k$  個の特定の値しかとらない場合に対しては Leung [32] が近似解法を与えている。

**Q3:**  $Q_2$  で最大完了時間のかわりに  $L_{\max}$  を最小にするスケジュールを求めよ。■

この問題に対してわれわれは次の近似解法 A 5 を考えた ([33])。

**A5:** LPT ルールにしたがって作られたリストにしたがってリスト・スケジュールを行なう。すなわち処理時間が長い仕事から空いている機械に割り当ててゆく。次に、各機械上で、納期の非減少順に仕事を並べかえる。■

A 5 の相対誤差は、 $(\hat{L}_{\max} - L_{\max}^*) / (L_{\max}^* + d_{\max}) \leq \min \{ (4/3) - (1/3m) - (m \times p_{\min}) / p, (1/3) - (1/3m) - m(d_{\max} - d_{\min}) / p \}$  となる。ここで  $p_{\min} = \min_i p_i$ ,  $d_{\max} = \max_i d_i$ ,  $d_{\min} = \min_i d_i$  である。計算手間は  $O(n \log n)$  となる。

**Q4:**  $Q_2$  でさらに先行関係  $<$  が存在するとき最大完了時間を最小にするスケジュールを求めよ。■

先行関係が与えられている時、 $J_i < J_j$  は  $J_j$  の処理開始以前に  $J_i$  の処理が完了していなければならないことを意味し、閉路を含まない方向グラフとして表現される (図 2 参照)。仕事  $J_l$  の直前仕事集合を  $IP(J_l) \triangleq \{J. | J. \in J, J. < J_l \text{ かつ } J. < J_{\Delta} < J_l \text{ となる } J_{\Delta} \text{ が存在しない}\}$ ,  $l=1, 2, \dots, n$  と、直後仕事集合を  $IS(J_l) \triangleq \{J. | J. \in J, J_l \in IP(J.)\}$ ,  $l=1, 2, \dots, n$  と定義する。

**A6(LP (Longest path) アルゴリズム):** まず、各仕事  $J_k$  の長さ  $l(J_k)$  を次のように逐次計算する。

$$\begin{cases} l(J_k) = p_k \dots\dots\dots (IS(J_k) = \phi) \\ l(J_k) = p_k + \max_{J. \in IS(J_k)} l(J.) \dots\dots (IS(J_k) \neq \phi). \end{cases}$$

図 2 のグラフでは  $IS(J_2) = IS(J_3) = IS(J_6) = \phi$  より、 $l(J_2) = 2$ ,  $l(J_3) = 3$ ,  $l(J_6) = 10$ , そして  $l(J_1) = 6 + \max_{J. \in IS(J_1)} p. = 6 + p_3 = 9$ ,  $l(J_4) = 5 + \max_{J. \in IS(J_4)} p. = 5 + \max(p_2, p_3, p_6) = 5 + \max(2, 3, 10) = 15$ ,  $l(J_5) = 17 + \max(2, 10) = 27$  となる。

$l(\cdot)$  の非増加順に仕事を並べてリスト  $L = (J_{i_1}, J_{i_2}, \dots, J_{i_n})$  を作る。すなわち、 $1 \leq j < k \leq n$  に対して  $l(J_{i_j}) \geq l(J_{i_k})$  となっていて、LP リストと呼ばれる。LP リストの順にしたがって、リスト・スケジュールを行なう。この結果できるスケジュールを LP スケジュールと言う。先ほどの  $l(\cdot)$  についての LP リストは  $\{J_5, J_4, J_6, J_1, J_3, J_2\}$  となる。A 5 の良さは先行関係に対応する有向グラフの形に依存する。(b)型の相対誤差で比較的簡単な先行関係の時次になる<sup>13)</sup>。(ただし、 $|\cdot|$  は集合の要素数。)

(1) **連鎖システム:** すべての  $J_k \in J$  に対して  $|IP(J_k)| \leq 1$ , かつ  $|IS(J_k)| \leq 1$ . 相対誤差は  $1 + \max[(m-2k+1)(2k+1)/\{(m-2k+1)(3k-1)+k^2\}] \leq k \leq (m+1)/3$ , 整数] 以下。

(m) **木型システム:** すべての  $J_k \in J$  に対して  $|IS(J_k)| \leq 1$ . この先行関係に対する(b)型の相対誤差は  $2 - (2/(m+1))$  以下となる。

**Q5(一様機械問題):**  $J_1, J_2, \dots, J_n$  を機械スピード  $s_1, s_2, \dots, s_m$  で与えられる  $m$  個の機械上で処理して、最大完了時間を最小にするスケジュールを求

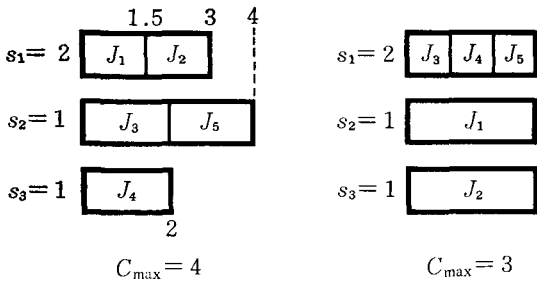


図 7(a) LPTによるスケジュール (b) 最適スケジュール

めよ。■

Q 5に対して2つの近似解法, LPT アルゴリズムおよび  $MFD(k)$  (Multi-fit decreasing( $k$ )) アルゴリズムを紹介する。

**A7(LPTアルゴリズム)**: まだ割当てられてなくて, 処理時間が最大の仕事をその完了時間が最も早く機械へ逐次割当てる。■

$m=3, s_1=2, s_2=1, s_3=1, p_1=p_2=3, p_3=p_4=p_5=2$ を適用すれば図 7 (a) のようになる。図 7 (b) は対応する最適スケジュールである。A 7 の (b) 型の相対誤差は  $2 - (2/m + 1)$  以下となり, 計算手間は  $O(n \log n + nm)$  である (Coffman [5])。

**A8( $MFD(k)$  アルゴリズム)**: 処理時間の非増加順に仕事番号をつかえて  $p_1 \geq \dots \geq p_n$  とする。機械のほうも番号を適当につかえて  $s_1 \geq \dots \geq s_m$  とする。そして,  $X_k = \sum_{i=1}^k p_i, k=1, 2, \dots, n, Y_h = \sum_{i=1}^h s_i, h=1, 2, \dots, m, S_m = (s_1, \dots, s_m)$  とおく。最初  $LOW = \max(X_n/Y_m, \max_{1 \leq k \leq m} X_k/Y_k)^{14}$ ,  $UP = \{2 - (1/m)\} \times LOW$  とおく。C をこの LOW と U P のあいだで二分法で  $k$  回変え, FFD で詰め得る最小の C を求め, そのパッキングに対応するスケジュールを近似解とする。A 2 と異なるのは各機械を各々  $s_1 C, s_2 C, \dots, s_m C$  の容量のビンとすることで, 各仕事は大きさ  $p_1, p_2, \dots, p_n$  の品物と考える。最初  $C = (LOW + UP)/2$  とし, FFD で詰める。詰めることができれば  $UP =$  現在の C, できなければ  $LOW =$  現在の C として次の  $C = (LOW + UP)/2$  として,  $k$  回くりかえす。最後の C のパッキングに対応するスケジュールを採用する。■

$MFD(k)$  の計算手間は  $O(n \log n + kmn)$  となり, (b) 型の相対誤差は,  $k$  を適当にくりかえせば, 本質的に A 7 より良くなる。詳しくは Kunde & Steppat [27] を見られたい。

Q 5 で分割処理を許すと最適スケジュールは比較的簡単に求まる (Coffman [5]) が, この場合でも一般の先行関係をもつときはやはりむずかしい。M. Jaffe<sup>15)</sup> はこの問題に対して最大利用スケジュールという近似解法を提案している。その (b) 型の相対誤差は  $\sqrt{m} + 1/2$  以下である。また, 非一様機械問題に対しては Davis たち [9] が近似解法を与えているが, あまり良いものは知られていない。

### 4.3 2 機械問題に対する近似解法

**Q6**: 2 機械フローショップ問題で  $L_{\max}$  を最小にするスケジュールを求めよ。■

**A9**: 納期の非減少順に仕事を処理するスケジュールを作れ。■

フローショップ問題では各機械上で仕事の処理順を変えない, いわゆる順序スケジュールの中に 1 つは最適スケジュールが存在することが知られている。A 9 は  $1 \leq i < j \leq n$  で,  $d_i \leq d_j \Leftrightarrow \min(a_i, b_j) \leq \min(a_j, b_i)$  が満足される時, 最適スケジュールとなる。A 9 の相対誤差については  $(\hat{L}_{\max} - L_{\max}^*) / (L_{\max}^* + d_{\max}) \leq 1$  ( $d_{\max} = \max_i d_i$ ) となる。詳細はわれわれの論文 [33] を見られたい。

## 5. ま と め

ここで紹介したスケジューリング問題は比較的小さい問題であり, 全体的一端にすぎない。最適解法はもちろん, 近似解法すら見つけることがジョブ・ジョブ問題などの一般の問題ではむずかしい。私自身, この原稿を書くのに納期に間に合うように仕事をスケジュールできず事務局の方に迷惑をかけた次第です。

スケジューリング問題の近似解法の研究はゆきづまっているように思われるが, その原因の 1 つは相対誤差の上界の計算が非常に難しいことであ

り、もう1つは使える考え方が限られていることであると思われる。その意味で新しい考え方が必要とされるが、1つの発展方向と考えられるのは[29]に見られるように、どのような場合にいま考えている近似解法がうまくゆかないか調べ、そこから更にスケジュールを改良するという研究であろう。近似解法も含めてスケジューリング理論の現状については[15][31]を見られたい。

最後に、いつも研究で公私ともに御世話になっております大阪大学、西田俊夫先生に感謝致します。また、いつも共同研究しております大阪府立大学、益田照雄氏に協力を感謝いたします。

#### 参 考 文 献

- [1] Achugbue, J. O. and Chin, F. Y.: Bounds on Schedules for Independent Tasks with Similar Execution Times. *JACM*, Vol. 28, No. 1 (1981), 81-99
- [2] Adolphson, D.: Single Job Sequencing with Precedence Constraints. *SIAM J Computing*, Vol. 6 (1977), 40-54
- [3] Baker, K. R.: *Introduction to Sequencing and Scheduling*. John Wiley & Sons, Inc., New York, 1974
- [4] Brucker, P., Garey, M. R. and Johnson, D. S.: Scheduling Equal Tasks under Tree like Precedence Constraints. *Math. of Operations Res.*, Vol. 2 (1977), 275-284
- [5] Coffman, E. G. Jr (ed.): *Computer and Job-Shop Scheduling*. Willey Interscience Pub. Comp., 1976
- [6] Coffman, E. G. Jr., Garey, M. R. and Johnson D. S.: An Application of Bin-Packing to Multiprocessor Scheduling. *SIAM J. Computing*, Vol. 7, No. 1 (1978) 1-17
- [7] Conway, R. W., Maxwell, W. L. and Miller, L. M.: *Theory of Scheduling*. Addison-Welsey pub. Comp., 1966
- [8] Cook, S. A.: The Complexity of Theorem Proving Procedures. *Proc. 3rd Annual ACM Symp. Theory of Computing*, 1971, 151-158
- [9] Davis, E. and Jaffe, J. M.: Algorithms for Scheduling Tasks on Unrelated Processors. *JACM*, Vol. 28(1981), 721-736
- [10] Fisher, M. L.: Worst Case Analysis of Heuristic Algorithms for Scheduling and Packing. In *Deterministic and Stochastic Scheduling* ed. by D. A. H. Dempster et al., 1982, 15-33
- [11] Friesen, D. K.: Tighter Bunds for the Multifit Processor Scheduling Algorithm. *SIAM J. Computing*, Vol. 13, No. 1 (1984), 170-181
- [12] Garey, M. R. and Johnson, D. S.: Approximation Algorithms for Combinatorial Problems: an Annotated Bibliography. In *Algorithm and Complexity: New Direction and Recent Results*(ed. J. F. Traub), Academic Press, New York, 1976, 41-52
- [13] \_\_\_\_\_: *Computer and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman Comp., San Francisco, 1979
- [14] \_\_\_\_\_: Approximation Algorithms for Bin Packing Problem: A Survey. In *Analysis and Design of Algorithms in Combinatorial Optimization* (ed. G. Ausiello and M. Luckertini), 1981, 147-172
- [15] Graham, R. L., Lawler, J. K., Lenstra, J. K. and Rinnooy Kan, A. H. G.: Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey. *Ann. Discrete Math.*, Vol. 5(1979), 287-326
- [16] Horowitz, E. and Sahni, S.: Exact and Approximation Algorithms for Scheduling Nonidentical Processors. *JACM*, Vol. 23 (1976), 317-327
- [17] Harvath, E. C. and Sethi, R.: A Level Algorithm for Preemptive Scheduling. *JACM*, Vol. 24 (1977), 32-43



- [18] Jackson, J. R. : Scheduling a Production Line to minimize the Maximum Tardiness, *Res Report No.43, Management Sci. Res. Project*, University of California at Los Angeles, 1955
- [19] \_\_\_\_\_ : An Extension of Johnson's Results on Job-Lot Scheduling. *Nav. Res. Log. Quart.*, Vol. 3 (1956), 201-203
- [20] Johnson, S. M. : Optimal Two-and Three-Stage Production Schedules with Setup Times Included. *Nav. Res. Log. Quart.*, Vol. 1 (1954), 61-68
- [21] Johnson, D. S. : Near Optimal Bin Packing Algorithm. *Report MAC TR-109*, Massachusetts Institute of Technology, 1973
- [22] Karp, R. M. : Reducibility among Combinatorial Problems. In *Complexity of Computer Computations* (ed. R. E. Miller and J. W. Thatcher), Plenum Press, New York, 1972, 85-103
- [23] \_\_\_\_\_ : On the Computational Complexity of Combinatorial Problem. *Networks*, Vol. 5 (1975), 45-68
- [24] 木瀬 洋 : 機械スケジューリング問題の近似解法, オペレーションズ・リサーチ **25**(1980), 765-771
- [25] Kise, H., Ibaraki, T. and Mine, H. : Performance Analysis of Six Approximation Algorithm for the One-Machine Maximum Lateness Scheduling Problem with Ready Times. *J. of Operations Res. Soc. of Japan*, Vol. 22, No. 3 (1979), 205-224
- [26] Kunde, M. : Nonpreemptive LP-Scheduling on Homogeneous Multiprocessor Systems, *SIAM J. Computing*, Vol. 10, No. 1 (1981), 151-173
- [27] Kunde, M. and Steppat, H. : First fit Decreasing Scheduling on Uniform Multiprocessors. *DAM.*, Vol. 10 (1985), 165-177
- [28] Lageweg, B. J., Lenstra, J. K. and Rinnooy Kan, A. H. G. : Minimizing Maximum Lateness on One Machine : Computational Experience and Some Application. *Statistica Netherlendica*, Vol. 30(1976), 25-41
- [29] Langston, M. A. : Improved LPT Scheduling for Identical Processor Systems. *Technique et science informatiques*, Vol. 1 (1982), 69-76
- [30] Lawler, E. L. : Optimal Sequencing of a Single Machine subject to Precedence Constraints. *Management Sci.*, Vol. 19 (1973), 544-546
- [31] Lawler, E. L., Lenstra, J. K. and Rinnooy Kan, A. H. G. : Recent Development in deterministic Sequencing and Scheduling : A Survey. In *Deterministic and Stochastic Scheduling* (ed. D. M. Dempster et al), D. Reidel Pub. Comp., 1982, 35-73
- [32] Leung, J. Y. T. : On Scheduling Independent Tasks with Restricted Execution Times. *Operations Research*, Vol. 30, No. 1 (1982), 163-171
- [33] Masuda, T., Ishii, H. and Nishida, T. : Some Bounds on Approximation Algorithms for  $n/m/I/L_{\max}$  and  $n/2/F/L_{\max}$  Scheduling Problems. *J. of Oper. Res. Soc. of Japan*, Vol. 26 (1983), 212-224
- [34] \_\_\_\_\_ : Mixed Shop Scheduling Problem. *DAM.*, Vol. 11 (1985), 175-186
- [35] Moore, J. M. : An  $n$  Job, One-Machine Sequencing Algorithm for Minimizing the Number of Late Jobs. *Man. Sci.*, Vol. 15 (1968), 102-109
- [36] 鍋島一郎 : スケジューリング理論, 森北出版 (1974)
- [37] Potts, C. N. : Analysis of Heuristic for One Machine Sequencing with Release Dates and Delivery Times. *Operations Research*, Vol. 28 (1980), 1436-1441
- [38] Sahni, S. : Algorithms for Scheduling Independent Tasks. *JACM*, Vol. 23 (1976), 116-127
- [39] Schrage, L. : Obtaining Optimal Solutions

to Resources Constraint Network Scheduling Problem. Unpublished Manuscript.

[40] Smith, W. E.: Various Optimizers for Single Stage Production. *Nav. Res. Log. Quart.*, Vol. 3 (1956), 59-66

注

- 1) 必要なら適当に大きい正数を目標関数につけ加えることにより, 一般性を失うことなく  $F^*(Q(I)) > 0$  と仮定できる.
- 2)  $Q$  でいま考えているスケジューリング問題と問題例全体の集合の両方をあらわす.
- 3) 問題の解法のむずかしさから言えば, 1 台, 2 台, 3 台以上の 3 つに分けるべきであろう.
- 4)  $p_{ij} = 0$  は  $J_j$  は  $M_i$  では処理を必要としない, すなわち  $O_{ij}$  が存在しないことを意味する. また, 操作が 1 つだけの仕事の処理時間を  $p_j$  で示す. これは 1 機械および並列機械の場合である.
- 5) 最近, われわれは新しく混合ジョブ型を考察したが, これはオープン・ジョブ型とフロー・ジョブ型の仕事が混ざっている場合である[34]. もちろん, フロー・ジョブ型はジョブ・ジョブ型の特殊な場合と考えられることができる.
- 6) 納期については後述
- 7)  $[J_i \rightarrow J_j]$  と書くことにする. 先行関係は  $<$  で示す.  $<$  は推移律  $J_i < J_k$  かつ  $J_k < J_l \Rightarrow J_i < J_l$  を満足する.
- 8) オーダー  $f(x)$  は  $O(f(x))$  とかき, 常に  $f(x)$  の定数倍  $c \cdot f(x)$  で上から押さえられていることを示す(ただし  $c$  は  $x$  に無関係な定数).
- 9)  $I^{2.5}$  や  $I \log_2 I$  などにも実際には多項式でなくても多項式で上から押さえられるものも含む.
- 10) ビンごとに容量が異なってもよい. 後の一様機械問題では, 容量が異なったビンを設定する.
- 11) 詳しくは Coffman [5] を見られたい.
- 12)  $[x]$  は  $x$  を超えない最大の整数を示す. たとえば  $[2.3] = 2, [2] = 2$  である.
- 13) 詳しくは Kunde [26] を見られたい.
- 14) この LOW は Q5 で分割処理を許した時の最大完了時間の最小値である.
- 15) Jaffe, J. M.: An Analysis of Preemptive Multiprocessor Job Scheduling, *Mathematics of OR*, Vol. 5 (1980), 415-420

1986 年 1 月号

© 日本オペレーションズ・リサーチ学会. 無断複写・複製・転載を禁ず.

新時代のコンピュータ総合誌

# Computer Today

●61年1月号/好評発売中

定価880円

## エキスパートシステムズ

エキスパートシステム——現状と課題 石塚 満  
 エキスパートシステム研究開発の意義 上野晴樹  
 エキスパートシステム構築のステップ 溝口文雄  
 人工知能とエキスパートシステム 丹羽 清  
 エキスパートシステムの言語とツール 武田正之  
 エキスパートシステムのアーキテクチャ 金田悠紀夫  
 エキスパートシステムとツール

XCON—村上憲郎 MACSYMA—村尾裕一

YES/MVS—戸沢義夫 OPS5—小林重信

Knowledge Craft—亀山 靖他

お得な年間購読のおすすめ 年間5000円(6冊)

●11月号特集

定価880円

## プログラミング言語の動向

別冊

特集

# PAD

好評発売中  
定価1200円

—構造化プログラム開発技法—

# 数 理 科 学

2月号予告

定価 880 円

## 特集=確率過程

確率過程, 理論と応用 飛田 武幸  
 量子場の確率論的描像 江沢 洋  
 確率過程 量子化その周辺 並木美喜雄  
 スピン系のランダムウォーク表示 服部 哲弥  
 集団遺伝学の系図過程と拡散過程 高畑 尚之  
 白色雑音と網膜 中 研一・安藤祐一郎  
 潜在特性モデル 鮫島 史子  
 待ち行列システムの近似解析 山田 敬吾

<連載> 電磁気学を考える⑦

パズル⑦

今井 功

高木 茂男

数理科学・別冊

好評発売中

## ソリトン

定価2000円

## 知識と認知のソフトウェア

定価1800円

## サイエンス社

東京都千代田区神田須田町2-4 安部徳ビル

☎03(256)1091 振替 東京7-2387