実践講座

スケジューリング問題の新解法

(2)：分枝限定法で大規模問題例を解く

木瀬　洋

1．はじめに

前号[7]では茨木先生が組合せ最適化の計算複雑性の立場からスケジューリング問題の難しさを明確に論じられた。一言で言うと，(簡単に）解けるスケジューリング問題はむしろ例外的で，ほとんどの場合は，解くことが難しい，いわゆるNP困難問題であることを認識しなければならない，という事である。実際，この連載講座で以後，いろいろな近似的手法が展開されるのもこの認識を反映しての事である。しかし，ここで前向きの立場からこの難しさを考えてみる。NP完全の理論は，NP困難問題に対して最悪の場合に問題規模の指数関数程度の計算量を要する事を推測しているだけであって，全ての問題例に対して困難性を指摘しているのではない。また，その様な最悪例がどの程度存在するかはそれを解くアルゴリズムに依存すると思われる。言換えると，良く工夫されたアルゴリズムのもとでは大多数の問題例が効率よく解かれても不思議ではない。この意味で分枝限定法は現時点で最も有力な最適化手法である。ここでは分枝限定法の枠組を概説すると共に2種のNP困難なスケジューリング問題に対して1000ジョブ程度の大規模（？）問題例を解いている分枝限定法アルゴリズムを紹介する。これら2つの問題はいずれも“強い意味”でのNP困難問題である。

2．分枝限定法

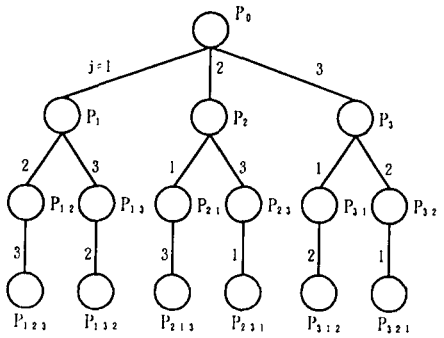
分枝限定法（以後，B＆B法と記す）では例えば図1の様な分枝図で示されるように，直接解く事が困難な原問題（P_0）を1つ以上の部分問題（P_1，P_2，P_3）に分割し，それらを全て解く事によってP_0を解こうとする。この様な分割は部分問題にも解く事に関し何らかの結論が得られるまで繰り返し適用される。以下では（P_0も含めて）部分問題P_iを（分枝図の）節点と言う。一般に分割されるべき節点を親，分割によって生成された節点を子と言い，結論の出た節点は終端，そうでなく分割される可能性のある節点は活性であると言う。節点の要件として次の2つが必要である。

（1）親節点の解集合はその全ての子節点の解集合の和と一致する。これによって全ての子節点を解く事によって親節点が解かれる事が保証される。

（2）子節点は親に比べて何らかの意味で解き易い。これによって原問題の解に近づける事になる。

たとえば，nジョブ$J=\{1,2,\cdots,n\}$の最適処理順序を決定するスケジューリング問題（図1は$n=3$の場合）において各ジョブ$j(=1,2,\cdots,n)$を処理すべき最初のジョブとして固定し，残り$n-1$個のジョブの最適順序を決定するn個のスケジューリング問題（P_1，P_2，P_3）は(1)と(2)を満している。この様にB＆B法の基本構造は極めて簡単で，どの様なスケジューリング問題にも容易に適用できる。しかし，これだけでは単に実行可能解を数え上げるだけの手段にすぎないのであって，現実的な解法とはなり得ない。たとえば，$n=100$のスケジュールの数は$n!\approx 10^{159}$であり，スーパーコンピュータが1千億世紀かかっても数え上げられない数である。

B＆B法では生成された全節点が分枝されているか，終端している，すなわち，活性節点の無くなった時点



n！(＝6)個のスケジュール

図1: 分枝図（n=3）

きせ　ひろし　京都工芸繊維大学工芸学部
〒606 京都市左京区松ケ崎御所海道町

で探索を終える事ができる。したがってできるだけ早い段階で各節点を終端化する事が望ましい。このため次の様な方策がとられる。なお，説明の都合上，以下では目的関数 f の最小化，すなわち，実行可能解中の最小値 $f^*(P_i)$ を節点 P_i の最適値とする。

（3）上界値：節点 P_i の近似解を求め，その f の上界値を $u(P_i)$ とする。特に，$u(P_i) = f^*(P_i)$ を証明できれば，P_i を終端する。（図1の P_{123} は自明な例の1つである）

$$u(P_0) \leftarrow \min\{u(P_0), u(P_i)\}$$

とする。$u(P_0)$ は原問題 P_0 の最適値候補という意味で暫定値と言い，以下では f^0 と記す。

（4）下界値：$f^*(P_i)$ の下界値 $l(P_i)$ を計算する。このとき，

$$(f^*(P_i) \geq)\ l(P_i) \geq f^0\ (\geq f^*(P_0))$$

ならば，P_i 以外に P_0 の最適解があり，P_i は終端する。この様な下界値テストはB & B法にとって最も普遍的かつ有力な終端化の方策である。

（5）優越関係：節点 P_j に対して別の P_i が存在し，$f^*(P_i) < f^*(P_j)$ が証明できるとき，P_i は P_j に優越すると言い，$P_i D P_j$ と記す。当然 P_j は終端する。

（6）探索法：活性節点の探索順序を決定する。（4）のためにはできるだけ早く最適値に到達できる（すなわち，$f^0 = f^*(P_0)$ となる）探索法が望ましい。代表的な探索法としては，次に探索すべき活性節点として最も新しく生成された節点を選ぶ“深さ優先探索”と，最も下界値が小さい節点を選ぶ“最良下界探索”がある。

図2は以上の説明を図1と同じ例で図示したものである。（○: 活性節点，●: 終端節点，⊙: 分割節点，→: 深さ優先探索による探索順序）

以上からB & B法の効率化のためにはできるだけ最適値に近い上界値（暫定値）と下界値をできるだけ早

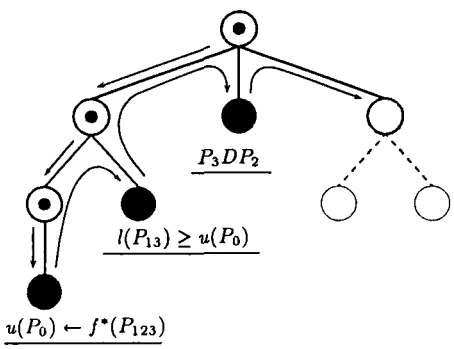


図 2: 分枝限定法の原理

くかつ簡単に求め，できるだけ多くの優越関係を見いだす必要がある。この事が容易でない事は明かであるが，n が 100 以上の様な大規模例を短時間で解くためにはこの事が実現されなければならない事も事実である。なお，B & B 法の詳しい解説については文献 [6] を参照されたい。

3．1 機械スケジューリング問題 [2]

3.1 問題の設定: n ジョブ $I = \{i | i = 1, 2, \cdots, n\}$ を1機械で処理する。各ジョブ i には最早開始時刻（たとえば，到着時刻）$a(i)$，処理時間 $p(i)$，納期 $d(i)$ が与えられている。各ジョブ i の開始時刻を $s(i)$，終了時刻を $c(i)$ とすると，

$$\begin{aligned} s(i) &\geq a(i) \quad \text{（開始時刻の制約）} \\ c(i) &= s(i) + p(i) \quad \text{（分割処理は不可）} \end{aligned} \tag{1}$$

でなければならないとする。また，納期遅れは

$$l(i) = \max\{0, c(i) - d(i)\} \tag{2}$$

で与えられる。このとき，最大納期遅れ

$$L_{max} = \max_i l(i) \tag{3}$$

を最小にする1機械でのジョブ処理順序を最適とする。ここで，

$$q(i) = \max_j d(j) - d(i) \tag{4}$$

とすると，

$$l(i) = \max\{0, f(i) + q(i) - \max_j d(j)\}$$

となるので，式(3)の L_{max} の代りに

$$f = \max_i \{f(i) + q(i)\} \tag{5}$$

としても最適スケジュールは変らない。よって以後では表現が容易な式(5)の最小化を考える。

3.2 グラフ表現: 問題をグラフ $G = (X, U)$ で表現する。ここで $X = I \cup \{0, *\}$ は節点集合で，I はジョブに対応し，0と*はそれぞれ全ジョブの開始と終了を表す仮節点である。U はアーク集合で，$U = U_1 \cup U_2 \cup U_3$ とする。$U_1 = \{(0, i) | i \in I\}$ で，アーク $(0, i)$ の長さを $a(i)$ とする。$U_2 = \{(i, *) | i \in I\}$ で，アーク $(i, *)$ の長さを $q(i) + p(i)$ とする。$U_3 = \{(i, j) |$ ジョブ i はジョブ j に先行する$\}$ とし，アーク (i, j) の長さを $p(i)$ とする。これらのアークはジョブ処理順序を決定する。図3は表1の7ジョブの例に対して順序 $(6, 1, 2, 3, 4, 5, 7)$ を表す

表 1: 1機械スケジューリング問題の例

ジョブ	1	2	3	4	5	6	7
a(i)	10	13	11	20	30	0	30
p(i)	5	6	7	4	3	6	2
q(i)	7	26	24	21	8	17	0

グラフである。このとき，節点0から節点$i \in I$までの最長経路長がジョブiの開始時刻，0から$*$までの最長経路長が式(5)のfを表す。よってスケジューリング問題は最短な最長経路長（最適値）をとるU_3（順序）を決定する問題に帰着する。

3.3 最適スケジューリングに関する性質: 時刻tでジョブi,jが処理可能（$a(i),a(j) \leq t$）で，$d(i) < d(j)$ならば，納期の小さい方，すなわち，$q(i) > q(j)$（式(4)参照）となるジョブiをjに先行させるべきであることは容易に理解できる。よって，「時刻$t = \min a(i)$から処理を開始し，加工中のジョブiが終了する毎に，残りのジョブRの最早開始時刻$t = \max\{c(i), \min_{j \in R} a(j)\}$（式(1)参照）において開始可能なジョブの中から$q(j)$最大のジョブを次に処理する」スケジューリングは必ずしも最適でないが，合理的である。この様なスケジューリングはSchrageアルゴリズムとしてよく知られている。図3はSchrageアルゴリズムを適用した結果を示している。

Schrageアルゴリズムが最適である場合を考えてみる。容易に分かる様に，$a(i)$あるいは$q(i)$が一定である場合は最適である。次にこれを一般化する。このアルゴリズムによるスケジュールに対するグラフの最長

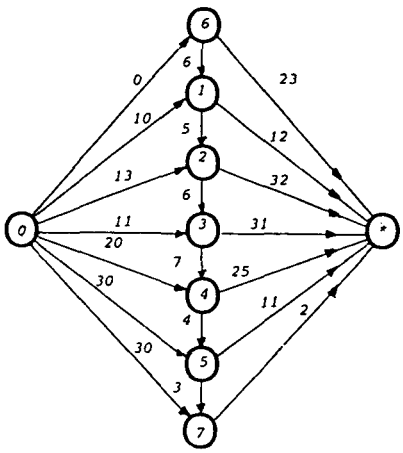


図 3: グラフによる問題表現

経路を簡単のため，$(0,1,\cdots,p,*)$とし，その長さをuとする。すなわち，

$$u = a(1) + \sum_{i=1,\cdots,p} p(i) + q(p). \tag{6}$$

ただし，pは式(6)を満たす最大の値とする。この事はSchrageアルゴリズムの定義より，

$$s(1) = a(1) = \min_{i=1,\cdots,p} a(i), \tag{7}$$

を意味する。次に，

$$l = a(1) + \sum_{i=1,\cdots,p} p(i) + \min_{i=1,\cdots,p} q(i), \tag{8}$$

とする。式(7)よりlは最適値f^*の下界であるから，$u = l$ならば，$f^* = u$である。次に，Schrageアルゴリズムが最適でない場合を考える。このとき，$q(c) > q(p)$となるジョブ$c(1 \leq c < p)$が存在する。そうでなければ，$q(p) = \min q(i)$となり，$u = l$である。

$$c = \max\{j | q(j) < q(p), j = 1,\cdots,p-1\} \tag{9}$$

とし，$J = \{c+1,\cdots,p\}$とする（図3の例では$c = 1$，$J = \{2,3,4\}$）。定義より

$$q(j) \geq q(p) > q(c), j \in J \tag{10}$$

であるから，Schrageアルゴリズムの定義より，$s(c) < a(j), \forall j \in J$，である。この意味で$c$を$J$に優先する事が有利となる可能性があるが，式(10)はまたcをJに後続させる方が優利となる可能性も示している。結局，以上の議論から最適スケジュールにおいてはcをJに先行させるか，後続させるかのどちらかである。

3.4 B & B法アルゴリズム: 2の分枝限定法の説明に従って述べる。

(1) 部分問題：親節点Pが3項組$(a(i),p(i),q(i))$で表されるnジョブスケジューリング問題であるとしたとき，Pが活性であるならば，PにSchrageアルゴリズムを適用し，3.3のcと$J = (c+1,\cdots,p)$を計算して，以下の2つの子節点P_1とP_2を生成する。

P_1: cがJに先行する場合を考える。この場合，

$$q(c) = \max\{q(c), \sum_{j \in J} p(j) + q(p)\}, \tag{11}$$

としたときのnジョブスケジューリング問題をP_1とする。

P_2: cがJに後続する場合を考える。この場合，

$$a(c) = \max\{a(c), \min_{j \in J} a(j) + \sum_{j \in J} p(j)\}, \tag{12}$$

としたときの n ジョブスケジューリング問題を P_2 とする。

(2) 上界値と下界値：各節点 P_i の上界値 $u(P_i)$ は Schrage アルゴリズムによって得られる（式(6)参照)。下界値 $l(P_i)$ は式(8)によって得られる（この他に式(1)の分割処理不可を緩和した最適分割処理スケジューリングの結果も用いる）。なお，u と l の計算手間は $O(n\log n)$ である。f^0 を暫定値としたとき，$u(P_i) < f^0$ ならば，$f^0 = u(P_i)$ とする。$l(P_i) \geq f^0$ ならば，P_i を終端する。

(4) 探索法：最良下界探索を用いる。

3.5 数値実験: $a(i), p(i), q(i)$ はそれぞれ $[1, kn]$，$[1, 50]$，$[1, kn]$ の範囲の整数型一様乱数で与えられた。ジョブ数 n は $n = 50 \sim 1000$ の 20 種類，また，各 n に対し，50 種類の $k = 1 \sim 200$ を適用し，合計 1000 種類の問題例に B & B 法が適用された。結果として $k = 19$，$n = 850$ 以外の全ての例題に対して最適解が得られた。解かれなかった例題の暫定値の相対誤差は 7×10^{-5} 以下であった。なお，使用言語は FORTRAN，使用計算機は IRIS80 である。

3.6 まとめと文献: ここで述べた1機械スケジューリング問題の NP 困難性は文献 [14] で証明されている。この問題に対する研究報告はここで紹介した Carlier の論文 [2] 以前にも多数ある。実際，上界値，下界値の計算法，また，探索法で Carlier のそれらと共通した B & B 法も提案されている [15]。しかし，それまでに提案されている最良のものでも 80 ジョブ程度を解いているにすぎない [12]。他と比して Carlier の B & B 法が決定的に違う点は c と J の関係に基づいて部分問題を設定したところにある。この様な手法は並列機械問題にも適用され，100 ジョブ程度までは高い確率で解かれる事を示している [3]。さらに，Carlier らは一般のジョブショップ問題にも適用し，ベンチマークとして悪名高い 10×10 問題例を初めて解いている [4]。他方，Adams らは一般のジョブショップ問題に対して Carlier の B & B 法をくり返し用いる近似解法（Shifting Bottleneck Procedure）を提案し，良好な結果を得ている [1]。

4．3 機械フローショップ問題 [10]

4.1 問題の定義: n ジョブ $J = \{1, 2, \cdots, n\}$ が3機械 $m = 1, 2, 3$ によってこの順序で次々と処理される。このとき，1)n ジョブは時刻0で処理可能である，2)各ジョブは一時に高々1機械でしか処理されない，3)各機械は一時に高々1ジョブしか処理できない，4)処理の中断は許されない，5)ジョブ処理順序はどの機械においても同一である（追抜き禁止）という条件のもとで最大完了時間を最小にするジョブ処理順序（以後，順序と略記）を最適とする。すなわち，機械 m におけるジョブ i の処理時間を $p_m(i)$ とすると，順序 $s = (j_1, j_2, \cdots, j_n)$ における k 番目のジョブ j_k の機械 m での終了時間 $F_m(j_k)$ は

$$\begin{aligned} F_1(j_k) &= F_1(j_{k-1}) + p_1(j_k), \\ F_2(j_k) &= \max\{F_1(j_k), F_2(j_{k-1})\} + p_2(j_k), \\ F_3(j_k) &= \max\{F_2(j_k), F_3(j_{k-1})\} + p_3(j_k), \end{aligned} \quad (13)$$

で与えられ，最大完了時間は $F_3(j_n)$ で与えられる。

4.2 優越規則のファジィ近似: 最初の k ジョブの順序 $s_k = (j_1, j_2, \cdots, j_k)$ に対して機械 m $(= 2, 3)$ における j_k の滞留時間を

$$T_m(s_k) = F_m(j_k) - F_1(j_k) \quad (14)$$

で定義する。このとき，次の優越規則が証明されている（文献 [16] の定理 3.11）。

<優越規則> 最適順序において部分順序 s_k の直後に処理する2ジョブを i，j とするとき，

$$T_m(s_k, i, j) < T_m(s_k, j, i),\ m = 2, 3 \quad (15)$$

が成立つならば，最適順序においてはジョブ j が i に先行することはない。

式(15)が $m = 2, 3$ の両方において同時に成立する事はむしろ稀で，これだけでは B & B 法の効率化に大きな期待はできない。そこでここでは式(15)が十分条件であることに注目して，以下のファジィ近似を用いることを考える。すなわち，式(15)が近似的に成立するならば，ジョブ i が j に先行する可能性が高いと見なし，この可能性をメンバーシップ関数で表現する。

$$\begin{aligned} D_m(s_k, i, j) &= T_m(s_k, i, j) - T_m(s_k, j, i) \\ D(s_k) &= \alpha D_2(s_k, i, j) + (1-\alpha) D_3(s_k, i, j) \end{aligned}$$

とするとき，ある α $(0 \leq \alpha \leq 1)$ に対して

$$\mu_{s_k}(i, j) = 0.5 - \{D(s_k)/(2D_{max}(s_k))\} \quad (16)$$

を i が j に先行する程度を表すメンバーシップ関数とする。図4は式(16)を図で表している。

4.3 B & B 法アルゴリズム: 以下では式(16)の $\mu_{s_k}(i, j)$ を用いた B & B 法アルゴリズムを述べる。

(1) 部分問題:最初の k ジョブの順序 $s_k = (j_1, \cdots, j_k)$ が固定され，残りのジョブの集合 $R(|R| = n - k)$ に対

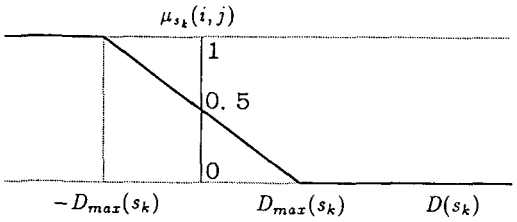


図 4: メンバーシップ関数

する最適順序を決定する問題を親節点 $P(s_k)$ としたとき，$n-k$個の子節点，$P(s_k,i)$，$i \in R$が生成される(図1参照)。

（2）上界値：原節点 $P(\emptyset)$ に対する初期暫定値 f^0が以下の手順（ファジィスケジューリングと呼ぶ）に従って決定される。

手順1　最初の kジョブの順序を s_k，残りのジョブの集合を Rとする（$k=0$ から出発する)。このとき，

$$\mu^*_{s_k}(i) = \min_{j \in R} \mu_{s_k}(i,j),\ \forall i \in R$$

を計算する（式(16)参照)。

手順2　以下の順序で決定された i^*を次に処理すべきジョブとする。

（a）$\mu^*_{s_k}(i^*) = \max_{i \in R} \mu^*_{s_k}(i)$

（b）$\mu_{s_k}(i,j) > 0$ となる個数が最大の i，

（c）$\mu_{s_k}(i,j)$ の平均値最大の i。

手順3　$s_k \leftarrow (s_k, i^*)$，$R \leftarrow R - \{i^*\}$ とする。$R = \emptyset$ ならば，順序 $s = s_n$に対する最大完了時間を f^0として計算終了，そうでなければ，手順1へ戻る。

（3）下界値：2機械フローショップ問題はJohnsonルール[9]よって解かれるので，これを利用する。すなわち，機械 $m=1$ の全ジョブの処理時間を $\min p_1(i)$ に緩和したときの最適値は元の問題の下界値である。$m=2,3$ についても同様の事が言える。

（4）優越関係：式(15)の優越規則を用いる。

（5）探索法:深さ優先探索を用いる(図2参照)。ただし，節点 $P(s_k)$ から生成された子節点 $P(s_k,i)$，$i \in R$ の中から次にどれを探索するかは次の規則による。a) 最小下界値の節点，b)a)で同値の場合は $P(s_k,i^*)$ を選ぶ。ただし，i^*はファジィスケジューリングの手順2と同じ手順で選ばれる。以上の探索法をファジィサーチと呼ぶ事にする。

4.4　数値実験: 提案したB＆B法におけるファジィスケジューリングやファジィサーチの効果を主として検討する数値実験を行った。ジョブ処理時間 $p_m(i)$ は [1,50] の範囲の整数型一様乱数で与えられた。$n = 10,20,\cdots,200$ の20種類のジョブ数のそれぞれに対して30例題ずつ，合計600例題が解かれた。メンバーシップ関数(16)にはαが含まれるが，最適なαは問題例によって異なる。よってファジィスケジューリング際してはα = 0, 0.4, 0.8 の3種を適用し，最良のαを決定した。ファジィサーチに際してはα = 0 に固定した。なお，使用言語はFORTRAN 77，使用計算機はSun Sparc Station IPC(15.8 MIPS)である。

表2: ファジィサーチによる暫定解更新回数の率(%)

	N.I.S.R.			
n	0	1	2	≥ 3
10–20	69	21	10	0
30–40	82	18	0	0
50–60	85	15	0	0
70–80	79	21	0	0
90–100	78	22	0	0
110–120	73	27	0	0
130–140	75	25	0	0
150–160	78	22	0	0
170–180	71	29	0	0
190–200	81	19	0	0
Average	77	22	1	0

表2は提案したアルゴリズムによって最適値に至るまでに更新された暫定解の回数（N.I.S.R.）の相対頻度(%)を示す。N.I.S.R.= 0 はファジィスケジューリングによる初期暫定解が最適である事を，N.I.S.R. = 1 はファジィサーチによって得られた最初の実行可能解が最適である事を意味する。この両者を無謬探索という事にすれば，表2の結果は約99%が無謬探索であった事を示している。ファジィスケジューリングとファジィサーチの効果をさらに調べるために，提案したアルゴリズム（以下ではAと記す）の他に，次の3種のB＆B法アルゴリズムA1，A2，A3及び近似解法としてのB＆BアルゴリズムA* について数値実験を行った。

A1：ファジィスケジューリングは用いるが，ファジィ探索を用いない。他はAと同じ。

A2：ファジィ探索は用いるが，ファジィスケジューリングは用いない。他はAと同じ。

A3：ファジィスケジューリングもファジィも用いない。他はAと同じ。

A*：アルゴリズムAで最適値から1%以内の誤差を

表3: B&Bアルゴリズムによって解けた問題例の率(%)

n	A	A1	A2	A3	A*
10–20	92	90	92	83	98
30–40	90	87	85	62	98
50–60	98	98	92	52	100
70–80	92	88	90	53	100
90–100	97	93	97	62	100
110–120	97	95	95	63	100
130–140	93	92	93	52	100
150–160	98	97	95	48	100
170–180	98	92	88	53	100
190–200	98	95	88	37	100
Average	95	93	92	57	100

許す。アルゴリズムAにおいて全ての下界を1.01倍することによって得られる。

表3は各アルゴリズムによってCPU時間30分以内に解かれた率を示す。ジョブ数nが増大するとき，ファジィ近似を用いたアルゴリズムA，A1，A2は高い率を維持しているのに対し，ファジィ近似を用いないA3は急激に低下しているのが分かる。アルゴリズムA2とA3の結果に大きな差が出ている事は同じ親節点を持つ子節点の間で同値の下界値を持つ場合が多い事に帰因している。なお，ジョブ数nがむしろ小さいところで解ける率が低いのはやや意外に思えるが，このへんがNP困難性の意味するところかも知れない。

4.5 まとめと文献: 3機械フローショップ問題のNP困難性は文献[5]で証明されている。この問題に対し初めてB＆B法を適用したのはIgnallら[8]である。一般のフローショップ問題に対するLagewegらのB＆B法は下界値の計算法に工夫があるが，3機械フローショップ問題に対しては50ジョブでも5割程度しか解けていない[13]。久保らは100ジョブ程度までは非常に高い確率で解く事ができるB＆B法を提案している[11]。なお，ここで提案されたアルゴリズムAはその後，下界値計算法に改良が加えられ，現在，1000ジョブ程度までは95%以上の確率で解けるまでになっている。また，優越規則(15)は一般のm機械にも拡張可能である。

5．むすび

ここで述べた2つのスケジューリング問題においても確かに解く事が難しい問題例（最悪例）が存在する。しかし，大半の問題例については1000ジョブ程度まで解く事が可能である。これをもってスケジューリング問題に対するB＆B法の実用性を主張するのは論議の余地があるかも知れないが，読者の判断にゆだねたい。現実のもっと複雑な問題に対してはここで述べた様には簡単ではないという意見もある。現在はその通りである。しかし，ここで述べた問題も長い間小さな問題例しか解かれていなかった。これは計算機の差だけではない。したがって将来，より複雑な問題に対してもB＆B法あるいは他の最適化アルゴリズムが大規模問題例に適用できる様になる事を信じたい。

参考文献

[1] Adams,J.,Balas,E.,Zawack,D.,Management Sci., 34(1988),391-401

[2] Carlier,J.,European J. Oper. Res.,11(1982),42-47

[3] Carlier,J.,European J. Oper. Res.,29(1987)

[4] Carlier,J.,Pinson,E.,Management Sci.,35(1989), 164-176

[5] Garey,M.R.,Johnson,D.S.,Sethi,R.,Math. Oper. Res.,1(1976),117-129

[6] 茨木，組合せ最適化，産業図書(1983)

[7] 茨木，本誌，39-10(1994)

[8] Ignall,E.,Schrage,L.,OR,13(1965),400-412

[9] Johnson,S.M.,Naval Res. Log. Quart.,1(1954),61-68

[10] 木瀬，程，松本，電学論C,114(1994),470-475

[11] 久保，沖野，精密機械，42(1976),20-25

[12] Lageweg,J.,Lenstra,J.,Rinnooykan,A.H.G., Statistica Neerlandica,30(1976),25-40

[13] Lageweg,J.,Lenstra,J.,Rinnooykan,A.H.G.,OR, 25(1978),53-67

[14] Lenstra,J.,Rinnooykan,A.H.G.,Brucker,P., Anals of Discrete Math.,1(1977),343-362

[15] Mcmahon,G.,Florian,M.,OR,23(1975),475-482

[16] 鍋島，スケジューリング理論，森北出版(1974)