



Title	フロー・ショップ スケジューリング問題の下界について
Author(s)	関口, 恭毅
Citation	北海道大學 經濟學研究, 29(1), 139-161
Issue Date	1979-03
Doc URL	https://hdl.handle.net/2115/31447
Type	departmental bulletin paper
File Information	29(1)_P139-161.pdf



フロー・ショップ スケジューリング問題の 下界について

関 口 恭 毅

1. は じ め に

フロー・ショップ スケジューリング問題 (FSP) は順序づけ問題の中でも主要な課題の1つである。FSPは技術的順序 (routing) が同一のN個の仕事が与えられたとき、何らかの意味で最良な、各機械での仕事の処理順 (スケジュール) を決定する問題である。技術的順序が同じであるから、各仕事は機械 m で処理されてから機械 $m+1$ ($1 \leq m \leq M-1$) で処理され、機械 M での処理によって完成すると考える。

従来の研究は、機械 m と $m+1$ の間に必要なだけ中間在庫を持てると仮定するもの [2, 4, 7~11, 14~16, 18] と、機械間の在庫容量は0であるとするもの [6, 13, 17, 18, 21] に大別できる。スケジュールの良さの尺度には総所要時間が多く用いられるが、平均滞留時間 [6, 8, 10], 納期遅れ [19], 段取時間 [20] などとも検討されている。仕事については、その間に先行関係の存在を仮定するものもあるが [12], 多くは任意のスケジュールが実行可能であるような状況を扱っている。各機械でのスケジュールが同一であるとする、いわゆる追越禁止の条件下では、中間在庫容量無限大の場合の最適スケジュールは必ずしも求められない [例えば15]。しかし、この条件下でのスケジュールは管理上の利点を持ち [4], またその最適スケジュールは真の最適スケジュールに十分近いものであることが期待できる [6, 8]。さらに、この条件は問題を易しくする (考察すべきスケジュールは $(N!)^{M-2}$ から $N!$ に減少) ので多くはこの条件下での最適スケジュールを探索している。

本報告で検討するのは最も標準的な設定の問題であって、「機械 M 台から

なる中間在庫容量無限大の工場で処理すべき N 個の技術的順序が同一の仕事が所与で、各仕事の各機械での処理所要時間は既知で確定している。このとき、追越禁止条件の下で総所要時間を最小にするスケジュールを求めよ。」というものである。

この問題の解法の多くは分岐—界値法 (Branch-and-Bound Method, B—B法) に基づくものであり、そこでは総所要時間の下界の算定法、下界を用いない廃棄法 (elimination method)、分岐の規則などが解法改善の着眼点として研究されている。中でも高精度な下界算法はB—B法の基本的課題であるとされ、多くの手法が提案されている (後節参照)。^[11] には、機械準拠下界 (machine-based bound, MB B) を巧妙に利用すれば、 $M=3$ であれば $N=100$ の問題を平均0.36秒 (FACOM 230—75による) 程度で解けることが示されているが、仕事の増加に伴う所要の計算時間 (CPU time) の増加速度の低さから見て充分実用に耐えるものと思われる。しかし、機械台数の増加につれてMB Bの精度が急激に悪化するであろうことはその計算法から見て容易に推定できる。既存の計算結果^[2, 4] を見ても所要計算量が急激に増大することが看取される。^[4] はより高精度の下界算法を考案してその有効性を確かめているが、その効果が実用に耐える程充分であるとは言い難い。

以下では Mitten 規則^[3, p. 147] を利用する新しい下界算法を提案し、それが従来どの算法よりも高精度の下界を与えることを立証し、最後に計算実験の結果を示して定量的検討を行うと共に従来算法の特性を調べることにとする。

2. 問題の定式化

N 個の仕事には $1 \sim N$ の通番が付されているとし、この通番の集合 J で N 個の仕事を示す。各仕事 i の各機械 m ($1 \leq m \leq M$) での処理所要時間を $\tau_{i,m}$ で示す ($\tau_{i,m}=0$ であっても、機械 m で所要時間0の処理を行うものと考えれば、 $\tau_{i,m}>0$ の場合と区別する必要はない)。機械は仕事の技術的順序に

従って通番が付されており, 機械 m は $t_m(\phi)(t_1(\phi) = 0)$ を時刻の原点とする) 以後に仕事群 J の処理に使用できるものとする。 $t_m(\phi) (1 \leq m \leq M)$ をこの F S P の初期条件と呼ぶ。

N 個の仕事はどれも任意の時刻に着手可能であり, 追越禁止条件の下では任意の順列がスケジュールとなる。 J のすべてのスケジュールの集合を P (P の要素数は $N!$) とする。 σ を P の要素の 1 つとし, $\sigma_n (1 \leq n \leq N)$ は σ の先頭から n 番目までの部分スケジュールを表わす。 σ_n はまた, n 個の仕事の順列を示すためにも用いる。このとき, $\bar{J}_{N-n}$ を σ_n に含まれていない仕事の集合とする。 $\bar{P}_{N-n}$ は $\bar{J}_{N-n}$ のすべての順列の集合を表わす。 $j \in \bar{J}_{N-n}$ であるとき $\sigma_{n+1} = \sigma_n \cdot j$ は σ_n の最後に仕事 j を追加して得る長さ $n+1$ の順列, $\sigma'_q (q \leq N-n)$ が $\bar{J}_{N-n}$ の q 個の要素の順列であるとき, $\sigma_{n+q} = \sigma_n \cdot \sigma'_q$ は σ_n の後に σ'_q を接続して得る (部分) スケジュールとする。

$t_m(\sigma_{n+1})$ を部分スケジュール $\sigma_{n+1} = \sigma_n \cdot j$ を機械 m が完了する時刻であるとすると, これは次の漸化式を用いて計算できる。

$$t_1(\sigma_{n+1}) = t_1(\sigma_n) + \tau_{j,1} \dots \dots \dots (1)$$

$$t_m(\sigma_{n+1}) = \max\{t_{m-1}(\sigma_{n+1}), t_m(\sigma_n)\} + \tau_{j,m}$$

$$\text{但し, } \sigma_0 = \phi, \quad 2 \leq m \leq M, \quad 0 \leq n \leq N-1.$$

我々の問題は次のような σ^* を求めることである。

$$t_M(\sigma^*) = \min_{\sigma \in P} t_M(\sigma) \dots \dots \dots (2)$$

機械 r と $m (r < m)$ の間にある機械での処理所要時間の和 $G_i(r, m)$ を次のように定義する。

$$G_i(r, m) = \begin{cases} \sum_{p=r+1}^{m-1} \tau_{i,p} & (1 \leq r \leq m-2) \\ 0 & (r = m-1) \end{cases} \dots \dots \dots (3)$$

N 個の仕事のスケジュール σ が与えられ, 各仕事は機械 r, m での処理だけで完了するが, r と m の間に $G_i(r, m)$ だけの加工時間遅れ (通常, 加工時間遅れは $\tau_{i,r} + G_i(r, m)$ あるいは $G_i(r, m) + \tau_{i,m}$ に対応しているが, そのスケジュールに対する影響の本質的部分は $G_i(r, m)$ にあるのでここでは上のように定義する) があるものとする。このとき, r, m 以外のすべ

ての機械を無視して2機械F.S.Pと考えると、 σ_{n+1} の機械 $P(P=r, m)$ での完了時刻 $t_p(\sigma_{n+1}|r)$ は次式で算定される。

$$t_r(\sigma_{n+1}|r) = t_r(\sigma_n|r) + \tau_{j,r} \dots\dots\dots (4)$$

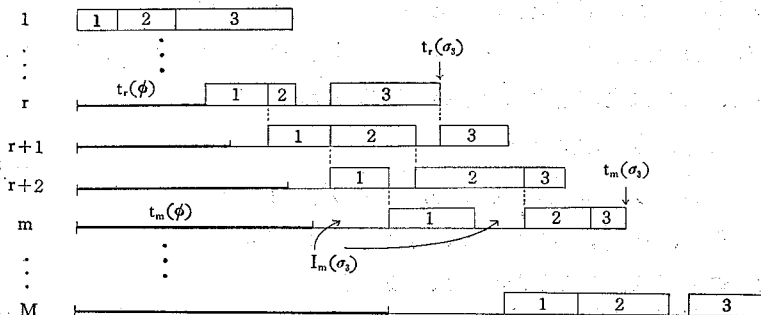
$$t_m(\sigma_{n+1}|r) = \max\{t_r(\sigma_{n+1}|r) + G_j(r, m), t_m(\sigma_n|r)\} + \tau_{j,m}$$

但し、 $\sigma_0 = \phi$, $1 \leq r < m \leq M$, $0 \leq n \leq N-1$

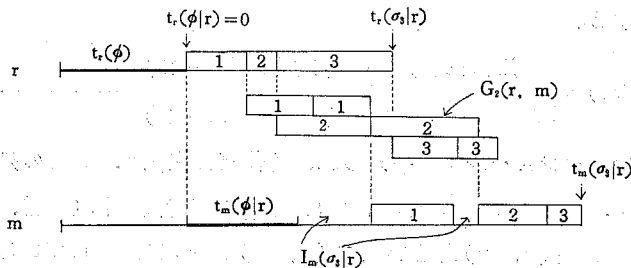
$$t_r(\phi|r) = 0, t_m(\phi|r) = t_m(\phi) - t_r(\phi) \dots\dots\dots (5)$$

$S(r, m)$ を、上のようにして形式的に作った加工時間遅れ $G_i(r, m)$ を持つ2機械総所要時間問題(図1参照)とすれば、これは次のような $\sigma^*(r, m)$ を求める問題である。

機械



(a) 元の問題



(b) $S(r, m)$

図1 元の問題と $S(r, m)$ の関係

$$t_m(\sigma^*(r, m)|r) = \min_{\sigma \in P} t_m(\sigma|r) \dots\dots\dots (6)$$

$I_m(\sigma_n)$ を σ_n を実施するとき機械 m に発生する遊休時間の総和とし、 $I_m(\sigma_n|r)$ を $S(r, m)$ における同様のものとすれば、これらは次の(7)、(8)両式で計算される。

$$I_m(\sigma_n) = t_m(\sigma_n) - \sum_{i=1}^n t_{[i], m} - \tau_m(\phi) \dots\dots\dots (7)$$

$$I_m(\sigma_n|r) = t_m(\sigma_n|r) - \sum_{i=1}^n \tau_{[i], m} - t_m(\phi|r) \dots\dots\dots (8)$$

ここで $[i]$ は σ_n の i 番目の仕事の番号を示す(厳密には $[i]\sigma_n$ とでもすべきであるが混乱の恐れのない限り上の記法を用いる)。(7)および(8)の右辺の第2、3項は $n=N$ に対しては任意の $\sigma \in P$ に対して等しい値をとるから、定義(2)と(6)は各々、(9)、(10)と等価である。

$$I_m(\sigma^*) = \min_{\sigma \in P} I_m(\sigma) \dots\dots\dots (9)$$

$$I_m(\sigma^*(r, m)|r) = \min_{\sigma \in P} I_m(\sigma|r) \dots\dots\dots (10)$$

3. 新しい下界算法*

$K(\sigma)$ を、 σ を実行する際に各機械に生ずる遊休時間の加重和であるとする

$$K(\sigma) = \sum_{m=2}^M \alpha_m I_m(\alpha) \quad \alpha_m \geq 0 \dots\dots\dots (1)$$

$K(\sigma)$ は関口等の定理1[18, p. 43]における関数 F の条件を満たしているので $\sigma = \sigma_n \cdot \sigma'_{N-n}$ とすれば、(2)式が成立する。

$$K(\sigma) \geq K(\sigma_n) + \sum_{m=2}^M \alpha_m I_m(\sigma'_{N-n}|r_m) \dots\dots\dots (2)$$

但し、 $t_{r_m}(\phi|r_m) = 0$, $t_m(\phi|r_m) = t_m(\sigma_n) - t_{r_m}(\sigma_n)$

我々は一般的な初期条件の下で検討しており、下界算法に興味があるのであるから、一般性を損うことなく $n=0$ として良い。また、 $K(\sigma_0) = 0$ に注

* 本稿を整理中に、同じ算法が[22]で提案されていることを知った。[22]は本節と5.に述べた内容に言及しているがその接近法は全く異っている。また、新しい下界算法を利用した実験結果も述べているが、6.で述べる筆者の結果から見て、取り上げた問題の範囲に偏りがあるように思われる。

意する。そこで(2)式の左右両辺に、各々2.の(7),(8)式を代入し、両辺の σ には無関係に定まる項を相殺すれば、(3)式を得る。

$$\sum_{m=2}^M \alpha_m t_m(\sigma) \geq \sum_{m=2}^M \alpha_m \{t_m(\sigma|r_m) + t_{r_m}(\phi)\} \dots\dots\dots (3)$$

この式は $1 \leq r_m < m$ なる任意の r_m に対して成立するから右辺をできるだけ大きくすれば、

$$\sum_{m=2}^M \alpha_m t_m(\sigma) \geq \sum_{m=2}^M \alpha_m \cdot \max_{1 \leq r < m} \{t_m(\sigma|r) + t_r(\phi)\}.$$

さらに、任意の σ に対しても成立することから、

$$\begin{aligned} \min_{\sigma \in P} \sum_{m=2}^M \alpha_m t_m(\sigma) &\geq \min_{\sigma \in P} \sum_{m=2}^M \alpha_m \max_{1 \leq r < m} \{t_m(\phi|r) + t_r(\phi)\} \\ &\geq \sum_{m=2}^M \alpha_m \cdot \min_{\sigma \in P} \max_{1 \leq r < m} \{t_m(\phi|r) + t_r(\phi)\} \end{aligned}$$

最後の式の $\min$ と $\max$ の順序が可換であることに注意しよう。そこで、 $\alpha_m = 1$, $\alpha_p = 0$ ($p \neq m$, $1 \leq p \leq M$) とおけば $t_m(\sigma)$ の下界 L_m を与える基本的な定理1を得る。

$$\text{【定理1】} \quad \min_{\sigma \in P} t_m(\sigma) \geq L_m \quad (2 \leq m \leq M) \dots\dots\dots (4)$$

$$\text{但し, } L_m = \max_{1 \leq r < m} L_m(r) \dots\dots\dots (5)$$

$$L_m(r) = \min_{\sigma \in P} \{t_m(\sigma|r) + t_r(\phi)\} \dots\dots\dots (6)$$

ここで、(6)の右辺の最小値は2.で定義した $\sigma^*(r, m)$ によって与えられることに注目する。また、 $\sigma^*(r, m)$ は Mitten の規則[3, p.147]によって与えられることを利用すれば、次のような L_m の計算手順を構成することができる。Step1 と Step2 が Mitten 規則に対応しており $\sigma_1 \cdot \sigma_{II}$ が $\sigma^*(r, m)$ である。

【Mitten 規則による下界算法】

Step 0 : $L_m \leftarrow 0$, $r \leftarrow 1$

Step 1 : $J_I \leftarrow \{i \in J \mid \tau_{i,r} < \tau_{i,m}\}$

$J_{II} \leftarrow J - J_I$

Step 2 : $\sigma_I \leftarrow (J_I \text{の要素を}\tau_{i,r} + G_i(r, m) \text{の小さい順に並べたもの})$

$\sigma_{II} \leftarrow (J_{II} \text{の要素を}\tau_{i,r} + G_i(r, m) + \tau_{i,m} \text{の大きい順に並べたもの})$

Step 3 : $\bar{L}_m \leftarrow \max\{L_m, L_m(r)\}$

但し, $L_m(r) = t_m(\sigma_I \cdot \sigma_{II}|r) + t_r(\phi)$

Step 4 : $r = m - 1$ なら終了。

$r < m - 1$ なら $r \leftarrow r + 1$ として Step 1 へ。

定理 1 から, L_m を各 m に関して計算しておけば, $t_m(\sigma)$ ($1 \leq m \leq M$) に関する任意の非減少関数に対する下界を求められることが明らかである。また, $m = M$ についての計算を行えば, 2. の(2)で定義した $t_m(\sigma^*)$ に対する下界を得ることができる。

上の算法によって B-B 法における下界を求めようとする場合, Step 1 と Step 2 を毎回実行する必要はないことに注意する。つまり, B-B 法の中では部分スケジュール σ_n に対する下界が, J_{N-n} に対する L_m を求めることによって算定されるが, J_{N-n} に対して Step 1, Step 2 によって得られるスケジュールは, J に対する $\sigma_I \cdot \sigma_{II}$ から σ_n に含まれる仕事を間引いたものになる。従って, B-B 法に入る前に必要とする各 (r, m) の組合せに対する $\sigma_I \cdot \sigma_{II}$ を求めておけば, 特定の m に対する下界算定に要する計算量は, $1 \sim m - 1$ の各 r に対する間引きと $t_m(\sigma|r)$ の算定を行うための $O((m - 1)(N - n))$ である。これは最も単純な MBB と同じオーダーの計算量である。

4. 算 法 の 改 良

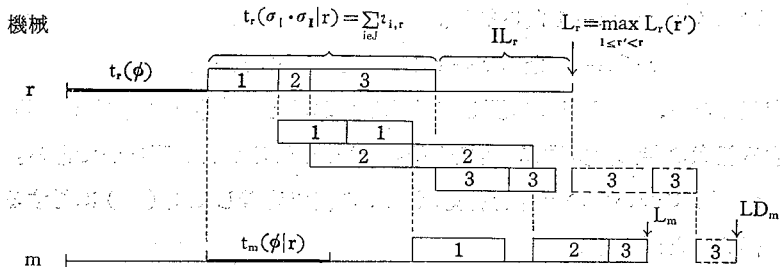
前節では $t_m(\sigma^*)$ の下界を計算するのに L_m だけを求めることを暗黙の内に仮定して検討していた。ここでは $L_m(r)$ の計算に先行して L_r が計算されている場合を考える。

L_r は機械 r が J のすべての仕事を完成するのに必要な時間の下限であるから, 機械 r には少なくとも IL_r だけの遊休時間が発生する。

$$IL_r = L_r - \left\{ \sum_{i \in J} \tau_{i,r} + t_r(\phi) \right\} \dots \dots \dots (1)$$

前節の算法は $IL_r = 0$ を仮定していたのであるから, このことを利用すればより高精度の下界 LD_m を求めることができる(図 2 参照)。しかし, IL_r が正であることから Mitten 規則に多少の修正を加える必要が生じる。この点を検討しよう。

総計 IL_r の遊休時間がスケジュール実行に際してどのように分布して現わ

図2 $S(r, m)$ と IL_r ときの $S(r, m)$

れるかが不明なので σ の i 番目の仕事の着手直前に $I_{(i)}$ なる遊休が生じると考えよう。

$$\sum_{i=1}^N I_{(i)} = IL_r \quad \dots\dots\dots (2)$$

この時、機械 r, m でのスケジュールの完了時刻 t_r^d, t_m^d は(3)式で計算できる。

$$t_r^d(\sigma_n | r) = \sum_{i=1}^n \{\tau_{(i),r} + I_{(i)}\} \quad \dots\dots\dots (3)$$

$$t_m^d(\sigma_n | r) = \max \{t_r^d(\sigma_n | r) + G_{(n)}(r, m), t_m^d(\sigma_{n-1} | r)\} + \tau_{(n),m}$$

仮りに、 IL_r だけの遊休時間が σ の N 番目の仕事の着手直前に集中して生ずるものとすれば、この時の完了時刻 t_r^o, t_m^o は $1 \sim N-1$ 番目の仕事に対しては 2. の(4)式で計算できるから次の式が成立する。

$$t_r^d(\sigma_{N-1} | r) = t_r^o(\sigma_{N-1} | r) + \sum_{i=1}^{N-1} I_{(i)} \quad \dots\dots\dots (4)$$

$$t_m^d(\sigma_{N-1} | r) \geq t_m^o(\sigma_{N-1} | r)$$

従って、

$$t_r^o(\sigma | r) = t_r^o(\sigma_{N-1} | r) + IL_r + \tau_{(N),r} = t_r^d(\sigma | r) \quad \dots\dots\dots (5)$$

$$t_m^o(\sigma | r) = \max \{t_r^o(\sigma | r) + G_{(N)}(r, m), t_m^o(\sigma_{N-1} | r)\} + \tau_{(N),m}$$

(3)と(5)を比較すれば、(4)の第2式の存在から、 $t_m^o(\sigma | r) \leq t_m^d(\sigma | r)$ となる。

以上から、次の定理を得る。

【定理2】 $S(r, m)$ において機械 r に総計 IL_r の遊休時間が強制されているとき $\sigma \in P$ の完了時刻を最小にするには、 IL_r を最後の仕事の直前に集中して置けば良い。

IL_r を考慮した下界算法を構成するには、次のようなスケジュール σ^0 を求める手順を見つけないければならない。

$$t_m^c(\sigma^0|r) = \min_{\sigma \in p} t_m^c(\sigma|r) \quad \dots\dots\dots (6)$$

σ^i を $J - \{i\}$ の仕事を Mitten 規則によって並べたものの最後に仕事 i を追加してできるスケジュールとする。 σ^i は仕事 i を最後に処理するスケジュールの中の最良のものであるから、明らかに次式が成立する。

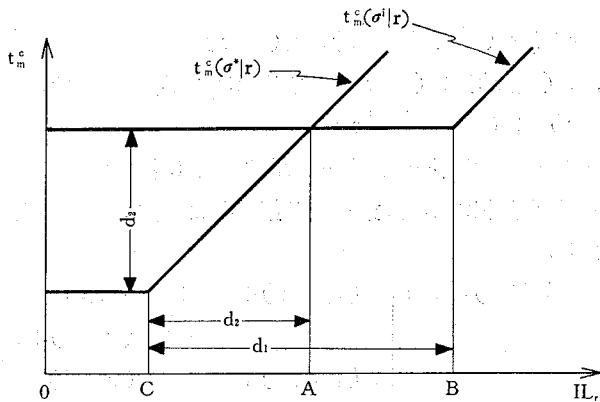
$$t_m^c(\sigma^0|r) = \min_{i \in J} t_m^c(\sigma^i|r) \quad \dots\dots\dots (7)$$

$\sigma^* = \sigma_I \cdot \sigma_{II}$ とし、その最後の仕事を i^* とする。 $i \neq i^*$ であれば σ^i の $N-1$ 番目の仕事は i には無関係に i^* であるから、 σ^* と σ^i の総所要時間は次のように書くことができる。

$$t_m^c(\sigma^i|r) = \max \left\{ \begin{array}{l} t_r(\sigma_{N-2}^i|r) + \tau_{i^*,r} + IL_r + \tau_{i,r} + G_i(r,m) + \tau_{i,m} \\ t_r(\sigma_{N-2}^i|r) + \tau_{i^*,r} + G_{i^*}(r,m) + \tau_{i^*,m} + \tau_{i,m} \\ t_m(\sigma_{N-2}^i|r) + \tau_{i^*,m} + \tau_{i,m} \end{array} \right\} \quad \dots\dots\dots (8)$$

$$t_m^c(\sigma^*|r) = \max \left\{ \begin{array}{l} t_r(\sigma_{N-2}^i|r) + \tau_{i,r} + IL_r + \tau_{i^*,r} + G_{i^*}(r,m) + \tau_{i^*,m} \\ t_m(\sigma_{N-1}^*|r) + \tau_{i^*,m} \end{array} \right\} \quad \dots\dots\dots (9)$$

σ^i と σ^* の定義により $IL_r = 0$ なら明らかに $t_m^c(\sigma^i|r) \geq t_m^c(\sigma^*|r)$ であるから、



$$B = t_m(\sigma^i|r) - \{t_r(\sigma_{N-2}^i|r) + \tau_{i^*,r} + \tau_{i,r} + G_i(r,m) + \tau_{i,m}\}$$

$$C = t_m(\sigma^*|r) - \{t_r(\sigma_{N-2}^i|r) + \tau_{i,r} + \tau_{i^*,r} + G_{i^*}(r,m) + \tau_{i^*,m}\}$$

図3 IL_r の t_m^c に対する影響

IL_r の増大に伴って σ^* の最適性がおびやかされる。この様子を図3に示す。

各折点は(8)あるいは(9)式の第一項がその $\max$ を与える IL_r の値を示し、 IL_r がこの値より大のところでは直線の傾きは1である。図から、 $B > A$ かつ $IL_r > A$ のときその時にきぎり、 σ^i は σ^* より良いスケジュールである。 $B > A$ は $d_1 > d_2$ と同値であり、 $d_1 > d_2$ は

$$d_2 = t_m(\sigma^i | r) - t_m(\sigma^* | r)$$

$$d_1 = B - C$$

を考慮して変形すれば次式と同値である。

$$G_{i^*}(r, m) + \tau_{i^*, m} > G_i(r, m) + \tau_{i, m} \quad (10)$$

i^* の定義から(10)は $i \in J_1$ のときだけ成立する可能性がある。

$A = C + d_2$ であるから、 $t_m(\sigma^i | r)$ に(8)式の $IL_r = 0$ としたものを代入すれば、 $i \in J_1$ であるとき次式が成立することがわかる。

$$A \geq \tau_{i, m} - \tau_{i, r}$$

従って、 $IL_r > A$ なるための必要条件は、

$$IL_r + \tau_{i, r} > \tau_{i, m} \quad (11)$$

次に σ^i の i^* と i を入れ換えたスケジュール $\sigma_{N-2}^i \cdot i \cdot i^*$ を考える。 $IL_r = 0$ であれば

$$t_m(\sigma^* | r) \leq t_m(\sigma_{N-2}^i \cdot i \cdot i^* | r) \leq t_m(\sigma^i | r)$$

であるから、 $t_m^o(\sigma^* | r) > t_m^o(\sigma^i | r)$ であるための必要条件は次のものである。

$$t_m^o(\sigma^* | r) > t_m(\sigma_{N-2}^i \cdot i \cdot i^* | r)$$

(12)式の第一項が(9)式の第一項と同一であることに注目すれば、上の式が成立するための必要条件は(12)の式第一項が第二項より大であることであり、これは(13)式である。

$$t_m^o(\sigma_{N-2}^i \cdot i \cdot i^* | r) = \max \left\{ \begin{array}{l} t_r(\sigma_{N-2}^i | r) + \tau_{i, r} + IL_r + \tau_{i^*, r} \\ \quad + G_{i^*}(r, m) + \tau_{i^*, m} \\ t_r(\sigma_{N-2}^i | r) + \tau_{i, r} + G_i(r, m) \\ \quad + \tau_{i, m} + \tau_{i^*, m} \\ t_m(\sigma_{N-2}^i) + \tau_{i, m} + \tau_{i^*, m} \end{array} \right\} \quad (12)$$

$$IL_r + \tau_{i^*, r} + G_{i^*}(r, m) > \tau_{i, m} + G_i(r, m) \quad (13)$$

以上から次の定理とその系を得る。

【定理 3】 $S(r, m)$ において機械 r に IL_r だけの遊休時間が強制されているとする。このとき次式が成立する。

$$t_m^0(\sigma^0 | r) = \min_{i \in J(IL_r)} t_m^0(\sigma^i | r) \dots \dots \dots (14)$$

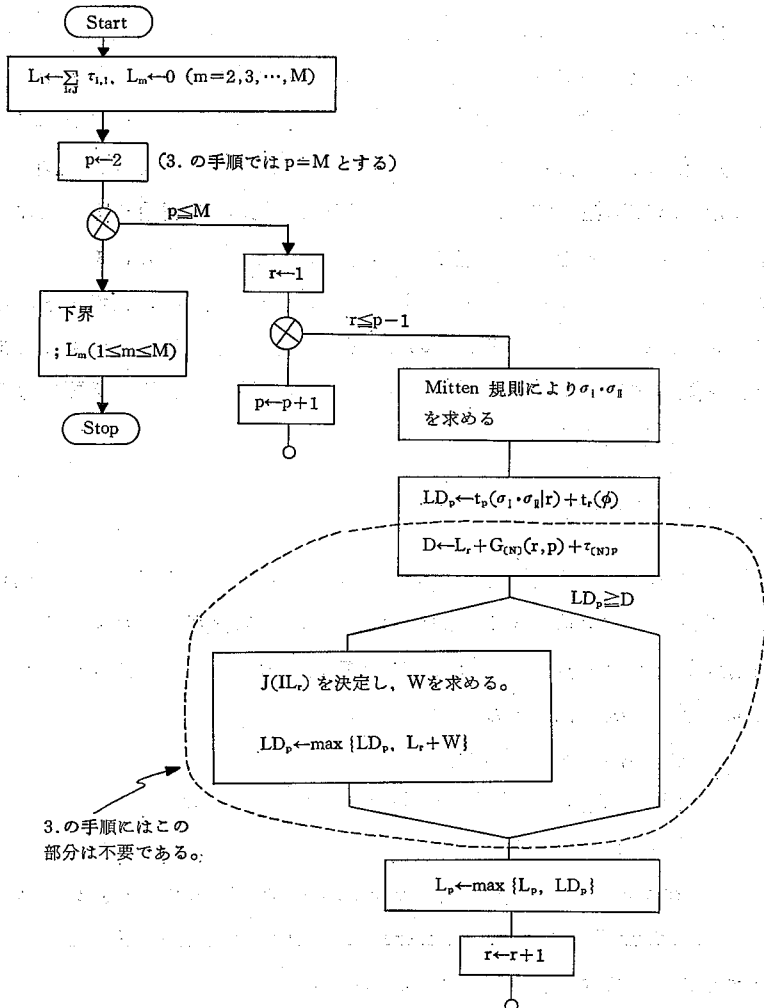


図 4 Mitten 規則を用いる下界算法およびその改善法

但し, $J(IL_r) = \{i^*\}^U \{i \in J_1 | (10), (11), (13) \text{ 成立}\}$

[系] $W = \min_{i \in J(IL_r)} \{G_i(r, m) + \tau_{i,m}\}$ とするとき,

$$t_m^0(\sigma^0 | r) \geq \max \{t_r^0(\sigma^* | r) + W, t_m^0(\sigma^* | r)\} \quad (15)$$

この系が成立することは(15)の右辺第一項が(8)式の右辺第一項の $i \in J(IL_r)$ に関する最小値であることから明らかである。

(15)式に基づいて、図4に示す手順で改良された下界 LD_m (図では L_p が LD_m に対応する) を求めることができる。この手順では LD_m を求めるために $LD_2 \sim LD_{m-1}$ をすべて計算しており前節で述べた方法に比べて約 $(m-2)$ $(m-1)/2$ 倍の計算量が必要になる。後に述べる計算実験の結果から見て、ネックになる機械に注目するなど、ここで述べた改良法の適用に際しては、計算量の増大することを十分考慮してその方法を検討する必要がある。($\sigma^* = \hat{\sigma} \cdot i \cdot \hat{\sigma} \cdot i^*$, $\sigma^i = \hat{\sigma} \cdot \hat{\sigma} \cdot i^* \cdot i$ として、(8)式のより詳しい表現を求めることによって、 σ^i が σ^* より良いための必要十分条件を得ることが可能であるがそれを適用するには図4の場合よりさらにN倍程度の計算が必要になるので、ここでは割愛する。)

5. 従来の下界算法との比較

ここでは3.で述べた下界 L_m が従来提案されているどの算法よりも高精度の下界であることを論証する。副産物として、従来提案されたものの中でも高精度な下界を改善する、ほとんど計算量の増加しない算法が得られる。

従来の下界算法の代表的なものとして、

機械準拠下界 (MBB) [8, 14]

仕事準拠下界 (job-based bound, JBB) [15]

Charleton-Death 下界 (CDB) [4, 5]

Bestwick-Hastings 下界 (overrun-based bound, OBB) [4]

Ashour 下界 (non-interference bound, NIB) [2, 1]

を上げることができる。このうち[4]に、

$$OBB \geq MBB, \quad OBB \geq CDB$$

が示されているので、ここではJBB, OBB, NIBの3つをとり上げる。

(i) JBB

機械 m に対するJBB $_m$ は、(1)式で定義される。

$$JBB_m = \max_{1 \leq r < m} JBB_m(r) \dots\dots\dots (1)$$

$$\text{但し, } JBB_m(r) = t_r(\phi) + \max_{i \in J} [\sum_{p=r}^m \tau_{i,p} + \sum_{j \in J - \{i\}} \min\{\tau_{j,r}, \tau_{j,m}\}] \dots\dots\dots (2)$$

$L_m(r)$ と $JBB_m(r)$ の間には次式で示す関係があるので、 $L_m \geq JBB_m$ である。

$$\begin{aligned} L_m(r) &= t_r(\phi) + t_m(\sigma_I \cdot \sigma_{II} | r) \\ &\geq t_r(\phi) + \max_{1 \leq n \leq N} \{ \sum_{j=1}^n \tau_{(j),r} + G_{(n)}(r, m) + \sum_{i=n}^N \tau_{(i),m} \} \\ &\geq t_r(\phi) + \max_{i \in J} \{ \sum_{j \in J - \{i\}} \tau_{j,r} + \sum_{p=r}^m \tau_{i,p} + \sum_{j \in J_{II} - \{i\}} \tau_{j,m} \} \\ &\geq t_r(\phi) + \max_{i \in J} [\sum_{p=r}^m \tau_{i,p} + \sum_{j \in J - \{i\}} \min\{\tau_{j,r}, \tau_{j,m}\}] \\ &= JBB_m(r) \end{aligned}$$

(ii) OBB

σ^+ をJの仕事に $G_i(r, m) + \tau_{i,m}$ の大きい順に並べたスケジュールとする。任意のスケジュール σ の i 番目の仕事のオーバーラン $y_{(i)}(\sigma)$ は(3)式で定義される。

$$y_{(i)}(\sigma) = G_{(i)}(r, m) + \tau_{(i),m} - \sum_{j=i+1}^N \tau_{(j),r} \dots\dots\dots (3)$$

このとき、

$$\max_{1 \leq i \leq N} y_{(i)}(\sigma^+) = \min_{\sigma \in P} \max_{1 \leq i \leq N} y_{(i)}(\sigma) \dots\dots\dots (4)$$

であること[4]に注目して、OBB $_m$ は(5)で定義される。

$$OBB_m = \max \{ \max_{1 \leq r < m} OBB_m(r), L_m(m-1) \} \dots\dots\dots (5)$$

$$\text{但し, } OBB_m(r) = t_r(\phi) + \sum_{i \in J} \tau_{i,r} + \max_{1 \leq i \leq N} y_{(i)}(\sigma^+) \dots\dots\dots (6)$$

(3)に着目すれば次のように変形できて、 $L_m \geq OBB_m$ を示すことができる。

$$\begin{aligned} L_m(r) &\geq t_r(\phi) + \max_{1 \leq i \leq N} \{ \sum_{j=1}^i \tau_{(j),r} + G_{(i)}(r, m) + \tau_{(i),m} \} \\ &= t_r(\phi) + \sum_{i \in J} \tau_{i,r} + \max_{1 \leq i \leq N} y_{(i)}(\sigma_I \cdot \sigma_{II}) \end{aligned}$$

(4)が成立するから、

$$\geq t_r(\phi) + \sum_{i \in J} \tau_{i,r} + \max_{1 \leq i \leq N} y_{(i)}(\sigma^+) = OBB_m(r) \dots\dots\dots (7)$$

なお、(7)の最後の不等式は σ^+ のかわりに $\sigma_I \cdot \sigma_{II}$ を用いることによって

OBB_m を改善できることを示している。 $\sigma_1 \cdot \sigma_{II}$ を求めるのに必要な計算量は σ^+ のそれとほとんど変わらない。

(iii) NIB

σ^- を $\tau_{i,r} + G_i(r, m)$ の小さい順に並べたスケジュールとし、この i 番目の仕事に対して次式により $C_{(i)}$ を定義する。

$$C_{(0)} = 0 \quad \dots\dots\dots (8)$$

$$C_{(i)} = \max\{\tau_{(i),r} + G_{(i)}(r, m), C_{(i-1)}\} + \tau_{(i),m}$$

この時 NIB_m は次のように定義される。

$$NIB_m = \max_{1 \leq r < m} NIB_m(r) \quad \dots\dots\dots (9)$$

$$\text{但し, } NIB_m(r) = C_{(N)} + t_r(\phi) \quad \dots\dots\dots (10)$$

(8)を逐次計算する過程で第二式 $\{ \}$ の中の第一項が $C_{(i-1)}$ より大であるところの最大の i を k とすれば $C_{(N)}$ は次式で与えられる。

$$\begin{aligned} C_{(N)} &= \tau_{(k),r} + G_{(k)}(r, m) + \sum_{j=k}^N \tau_{(j),m} \\ &= \max_{1 \leq i \leq N} \{ \tau_{(i),r} + G_{(i)}(r, m) + \sum_{j=i}^N \tau_{(j),m} \} \\ &= \max_{1 \leq i \leq N} \{ \tau_{(i),r} + G_{(i)}(r, m) - \sum_{j=i}^{i-1} \tau_{(j),m} \} + \sum_{j=1}^N \tau_{(j),m} \quad \dots\dots\dots (11) \end{aligned}$$

この最後の表現と(6)に(3)を代入して得る表現の類似性に注目を要する。すなわち、

$$\begin{aligned} OBB_m(r) - t_r(\phi) &= \max_{1 \leq i \leq N} \{ \tau_{(i),m} + G_{(i)}(r, m) \\ &\quad - \sum_{j=i+1}^N \tau_{(j),r} \} + \sum_{j=1}^N \tau_{(j),r} \quad \dots\dots\dots (12) \end{aligned}$$

$S(r, m)$ の逆順序問題 $\bar{S}(r, m)$ [18]とは(12)式で定義される $\bar{t}_r(\sigma | m)$ を最小にするスケジュール $\bar{\sigma}^*$ を求める問題である。

$$\begin{aligned} \bar{t}_m(\sigma_n | m) &= \bar{t}_m(\sigma_{n-1} | m) + \tau_{(n),m} \\ \bar{t}_r(\sigma_n | m) &= \max\{ \bar{t}_m(\sigma_n | m) + G_{(n)}(r, m), \bar{t}_r(\sigma_{n-1} | m) \} + \tau_{(n),r} \quad \dots\dots\dots (13) \end{aligned}$$

但し、 $\bar{t}_m(\phi | m) = 0$, $\bar{t}_r(\phi | m) = 0$

これは $S(r, m)$ がその完了時刻を $\text{---}\searrow$ のように計算するのに対して、 $\text{---}\nwarrow$ のように、スケジュールを逆の順序に実施した時、逆方向から計算した完了時刻を求めることに相当する。(11)式の σ^- を逆転した $\bar{\sigma}^-$ を考え、逆順序問題のように逆方向から計算するものと考え、(11)式は次の式と等価

である。

$$C_{[N]} = \max_{1 \leq i \leq N} \{ \tau_{[i],r} + G_{[i]}(r, m) - \sum_{j=i+1}^N \tau_{[j],m} \} + \sum_{j=1}^N \tau_{[j],m} \quad (14)$$

但し、(14)の $[i] = (1)$ の $[N-i+1]$

σ^- が $\bar{S}(r, m)$ に対する σ^+ に相当することに注意して(14)と(12)を比較すれば、 $C_{[N]}$ が逆順序問題の min・max オーバーラン(4)であることがわかる。

(12)は $t_r(\phi|r) = t_m(\phi|r) = 0$ である $S(r, m)$ の総所要時間の下界であり、 $C_{[N]}$ は $\bar{t}_m(\phi|m) = \bar{t}_r(\phi|r) = 0$ とする $\bar{S}(r, m)$ の総所要時間の下界である。この時、 $S(r, m)$ と $\bar{S}(r, m)$ の総所要時間の最小値は各々、 $\sigma_I \cdot \sigma_{II}$ とそれと逆順序のスケジュールによって実現され、互に等しい。(これは [18] と同様の手法で容易に示すことができる。) 従って、 $L_m \geq NIB_m$ が成立する。

OBB_m の場合と同様に、 $\sigma_I \cdot \sigma_{II}$ に対して(14)式を計算することによって、より高精度の NIB_m を求めることができることを指摘しておく。

6. 計 算 実 験

L_m が他の下界より定性的には良いことがわかったが、ここではその改善量が定量的にどの程度になるかを実験によって調べる。

$\tau_{i,m}$ として、1～10あるいは1～100の整数型乱数を用いる例題を、 $M=1, 2, \dots, 20$, $N=10, 20, 30, 50, 100$ の計 $20 \times 5 = 100$ ケースについて、各々100題ずつ作り、 L_M , OBB_M, MBB_M, JBB_M を計算した。各々についての100題の平均値を基準データとして、以下結果を整理して述べる。なお、下界の良さの尺度には次式の S_M と各下界の差を用いている。

$$S_M = \max_{1 \leq r < M} t_M^*(\sigma^*(r, M)|r) \quad (1)$$

S_M は4. の(9)式の r に関する最大値であって、4. で述べた改善法によって得られる下界の上限である。

図5は、1～10の整数型乱数を用いる場合の、各下界の S_M に対する比率の平均値を示したものである。但し、 $N=20$ と100については図が混雑するのを押えるために選択的に記入してある。1～100の乱数を用いる場合は、図5とはほとんど同一であるので割愛する。各々のデータ点での下界値の差の絶

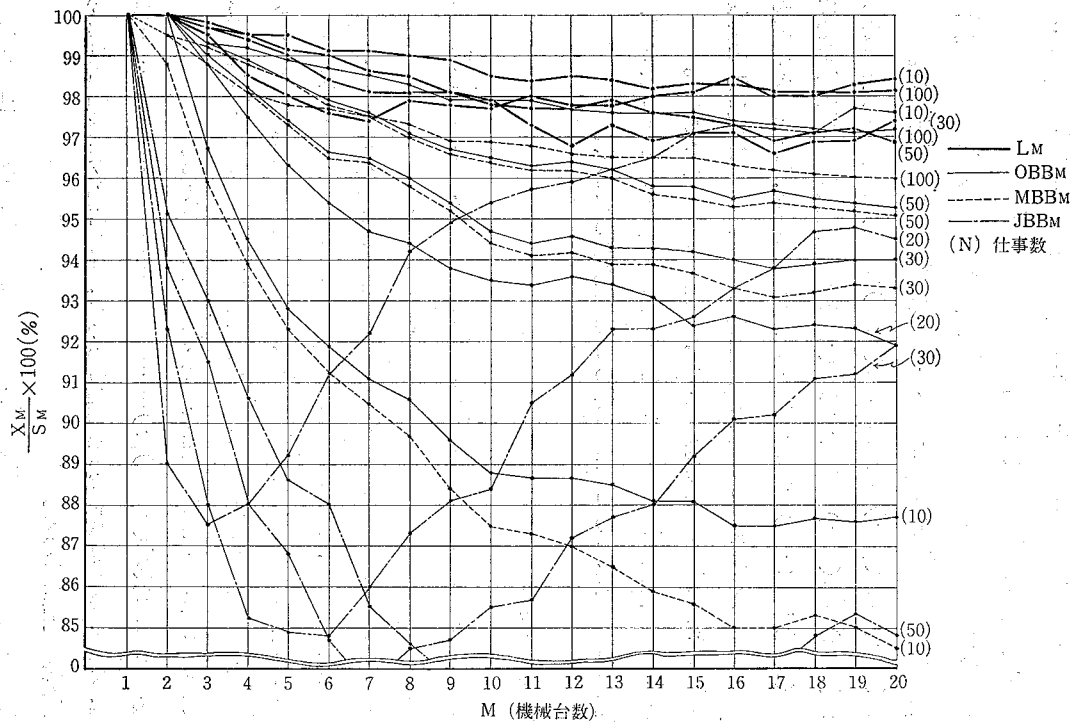


図5 各種下界計算法の良さ

対量は、表 1 を参照して求めれば良い。表 1 は S_M の 1% が何単位時間に相当するかを示すものである。例えば、 $N=30$, $M=10$ の場合には、図 5 より $L_M: 97.9$, $OBB_M: 94.7$ であり、表 1 から 2.20 を得るので

$$L_M - OBB_M = (97.9 - 94.7) \times 2.20 = 7.04 \text{ (単位時間)}$$

だけ L_M の方が大きい下界を与えることがわかる。1~100 の場合にも図 5 を使って同様に行えば良い。

こうして得る S_M との差の絶対量を N および M に関して平均したものが図 6 である。(N) とあるものは、その各データ点が

$$\frac{\sum_{N=10, 30, 50, 100} \{S_M - X_M\}}{4} \quad \text{但し } X = L, OBB, MBB, JBB$$

の値を示しており (M) とあるものは、

$$\frac{\sum_{M=1}^{20} \{S_M - X_M\}}{20} \quad \text{但し } X = L, OBB, MBB, JBB$$

を各 N の値に対して求めたものである。非常に粗いまとめ方の図であるが、各々 M あるいは N の変化に対する各下界の挙動を特徴的に示していると言える。

表 2 は、図 5 の $N=10, 50$ に対応するデータ点での分散を示したものであるが、 N の他の値に対しても類似の傾向があることを付記しておく。

以上の図および表を検討すれば、以下に述べるような諸点が明らかとなる。

a. 新しく提案した L_M は他の算法に比べて格段に良い下界である。 L_M の計算に要する計算量はオーダーでは、例えば OBB_M と変わらないので、 L_M の採用によって B-O 法が大幅に改善されそうである。特に [4] では MBB の代りに OBB を用いるだけでかなりの改善となることを示していることに着目する。

b. OBB と MBB の差は L と OBB の差の $1/5 \sim 1/6$ 程度である。

c. JBB は他の 3 種に比べて、極端に悪い、しかし、仕事数に比較して機械台数が多い場合には、OBB や MBB より良い下界を与える傾向が見える。

d. 4. で提案した改善法による改善量は機械台数の増加につれて急激に大

表1 下界改善1%当りの下界改善量

(単位: 単位時間)

N \ M	2	4	6	8	10	12	14	16	18	20
10	0.61	0.74	0.86	0.97	1.10	1.22	1.34	1.47	1.58	1.70
	5.63	6.83	7.76	8.90	10.01	11.14	12.29	13.48	14.49	15.69
20	1.20	1.31	1.42	1.53	1.64	1.74	1.85	1.96	2.08	2.19
	10.73	11.80	12.94	13.91	14.94	15.81	16.85	17.87	18.84	19.97
30	1.75	1.87	1.98	2.07	2.20	2.30	2.41	2.52	2.62	2.72
	15.92	17.13	18.13	19.22	20.22	21.08	21.94	23.06	24.05	25.00
50	2.88	3.01	3.13	3.25	3.35	3.45	3.56	3.67	3.76	3.87
	26.35	27.65	28.69	29.66	30.71	31.55	32.54	33.51	34.37	35.39
100	5.70	5.89	6.02	6.13	6.24	6.34	6.43	6.54	6.65	6.74
	52.19	53.73	54.88	56.06	57.05	57.98	58.89	59.88	60.84	61.77

上段: 1~10の整数型乱数, 下段: 1~100の整数型乱数。

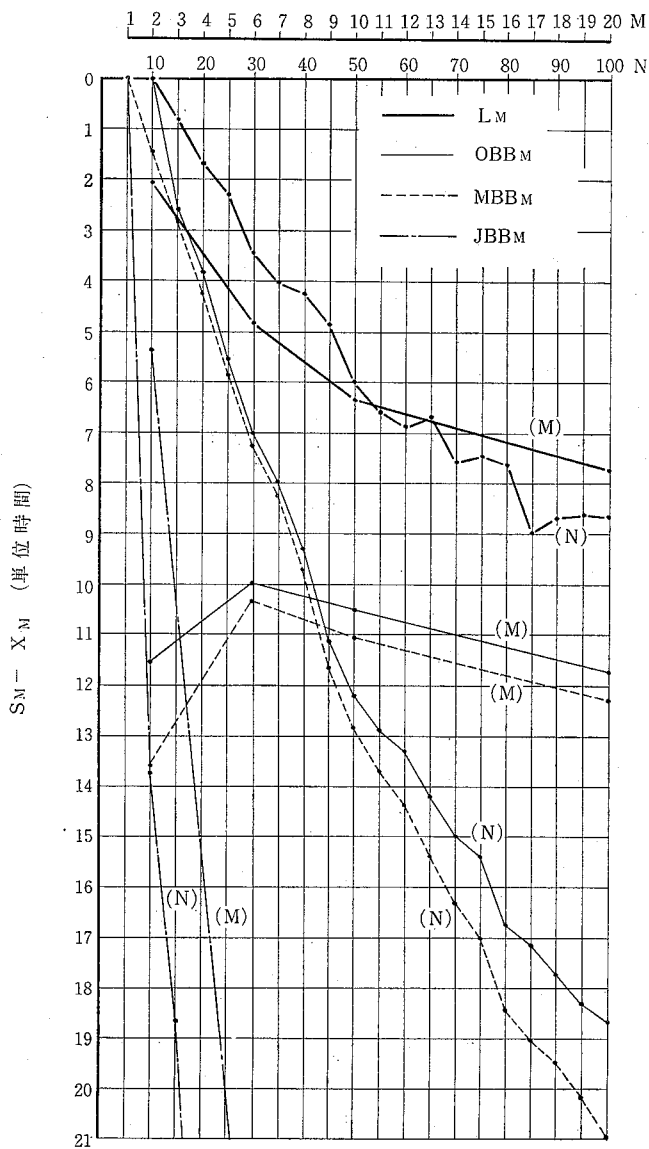


図6 仕事数および機械台数増加に対する各下界算法の特性 (1~10整数型乱数)

表2 各下界算法の分散 (1~10の整数型乱数の場合)

(単位: %)

$\begin{matrix} M \\ \text{下界} \end{matrix}$	2	4	6	8	10	12	14	16	18	20
L_M	0	2.57	3.58	3.19	3.41	2.66	2.79	2.76	2.26	2.10
	0	0.63	1.26	1.79	2.35	2.30	2.79	2.83	2.45	2.97
DBB_M	0	3.83	5.34	5.08	5.78	5.23	4.87	4.45	4.36	4.37
	0	0.77	1.55	2.12	2.61	2.86	3.07	3.30	2.81	2.98
MBB_M	1.52	4.27	5.73	5.48	6.22	5.94	5.99	5.41	5.64	5.36
	0.45	0.89	1.58	2.21	2.63	2.81	3.10	3.27	2.85	3.05
JBB_M	10.22	8.22	6.89	5.34	4.94	4.49	3.81	3.52	2.76	2.72
	5.35	6.75	6.37	6.03	4.95	4.39	3.89	4.21	3.70	3.89

上段: $N=10$ の場合, 下段: $N=50$ の場合。

となる。このことは、計算量の増大があまり急でないように適用方法を工夫すれば (4. の最後の段落を参照), この改善法は相当有効であろうことを示すものと考えられる。

e. S_M と OBB_M および MBB_M との差は仕事数が20~40のところで最も小さい。しかし, MBB を用いる場合, この位の仕事数の所で計算量が極端に大きくなる問題が増加する [例えば11を見よ]。この位の規模の問題が最も解きにくいもののように思われる。

f. 平均的に悪い下界ほどその分散が大である。これは, 平均的には悪くとも時に, 非常に良い下界を与える場合のあることを示すものと考えられ, 混合下界 (composed bound) が有効となる根拠と言えよう。

7. む す び

本稿では Mitten 規則を利用する下界算法を提案し, それが従来どの算法よりも高精度の下界を与えることを立証した。また, この過程で, Mitten 規則に基づくスケジュールを利用することによって, 既存のものの中でも高精度な Bestwick-Hastings 下界, Ashour 下界を改善できること, さらにこの2つの下界が同じ計算法を各々もとの問題とその逆順序問題に適用したものであることを示した。各種下界算法について行った計算実験の結果からは, 新提案の方法による下界改善量が特に大きいことを示すとともに, 既存の算法のこれまで言及されていなかった特性を2~3明らかにすることができた。

新しい下界算法を利用するB-B法の計算効率に関するデータはまだ十分に蓄積されていないので別の機会に述べたい。しかし, これまでのところ, 機械台数の増加とともに仕事数の増加に対する以上に, その効率が悪化する傾向が強い, 例えば, 準備的な計数では $M=3$ の場合には S_M が真の最適値にほぼ等しいのに対し, $M=10$ ではそれが90%程度となる。このことは, 機械台数が多い場合に, これまでのように機械干渉に関する制約を緩和する方法によって, 高精度の下界を得る見込が非常に少いことを感じさせる。

最後に、日頃、ご討論、ご激励下さる久保洋助教授（北海道大学大型計算機センター）、大内 東助手（北海道大学工学部）、若林信夫助教授（小樽商科大学）の諸氏に感謝する。

なお、本研究における計算実験は北海道大学大型計算機センター(FACOM 230—75)を利用して行われた。さらに、本研究は一部文部省科学研究費補助金に依っている。

参 考 文 献

- 1) Ashour, S., "A Branch-and-Bound Algorithm for Flow-Shop Scheduling Problems", AIIE Trans., Vol. 2 pp. 172—176 (1970).
- 2) Bakes, R.B., "A Comparative Study of Flow-Shop Algorithms", Opns. Res. Vol. 13, No. 1, pp. 62—73 (1965).
- 3) ———, Introduction to Sequencing and Scheduling, John Wiley (1974).
- 4) Bestwick, P.F. and Hastings, N.A.J., "A New Bound for Machine Scheduling", Opl. Res. Quart., Vol. 27, No. 2-ii, pp. 479—487 (1976).
- 5) Charlton, J.M. and Death, C.C., "A Method of Solution for General Machine-Scheduling Problems", Opns. Res. Vol. 18, No. 4, pp. 689—707 (1970).
- 6) Deman, J.M.V. and Baker, K.R., "Minimizing Mean Flowtime in the Flow Shop with No Intermediate Queues", AIIE Trans., Vol. 6, No. 1, pp. 28—34 (1974).
- 7) Heller, J., "Some Numerical Experiments for an $M \times J$ Flow-Shop and its Decision Theoretical Aspects", Opns. Res., Vol. 8 pp. 178—184 (1960).
- 8) Ignall, E and Schrage, L., "Application of the Branch-and-Bound Technique to Some Flow-Shop Scheduling Problems", Opns. Res., Vol. 13, pp. 400—412 (1965).
- 9) Johnson, S.M., "Optimal Two-and-Three-Stage Production Schedules with Set-up Time Included", Naval. Res. Log. Quart., Vol. 1 pp. 61—68 (1954).
- 10) Kohler, W.H. and Steiglitz, K., "Exact, Approximate and Guaranteed Accuracy Algorithms for the Flow-Shop Problem $n/2/F/\bar{F}$ ", Journal of ACM, Vol. 22, No. 1, pp. 106—114 (1975).
- 11) 久保 洋, 沖野教郎 "フロー・ショップ スケジューリング問題の一解法—三機械総所要時間問題の厳密解法—", Journal of JSPE, Vol. 42, No. 6 Extra, pp. 20—25 (1976).

- 12) Kurisu, T., "Three-Machine Scheduling Problem with Precedence Constraints", Journal of ORSJ, Vol. 20, No. 3, pp. 231—242 (1977).
- 13) Levner, E.V., "Optimal Planning of Parts Machining on a Number of Machines", Automation and Remote Control, Vol. 30, No. 12, pp. 1972—1978 (1969).
- 14) Lomnicki, Z.A., "A "Branch-and-Bound" Algorithm for the Exact Solution of the Three Machine Scheduling Problems", Opl. Res. Quart., Vol. 16, No. 1, pp. 89—100 (1965).
- 15) McMahon, G.B. and Burton, P.G., "Flow Shop Scheduling with the Branch and Bound Method", Opns. Res., Vol. 15, No. 3, pp. 473—481 (1967).
- 16) Nabeshima, I., "On the Bound of Makespans and its Application in M Machine Scheduling Problem", Journal of ORSJ, Vol. 9, No. 3 & 4, pp. 98—136 (1967).
- 17) Reddi, S.S. and Ramamoorthy, C.V., "On the Flow-Shop Sequencing Problem with No Wait in Process", Opl. Res. Quart., Vol. 23, No. 3, pp. 323—331 (1972).
- 18) 関口恭毅, 小山昭一, 三浦良一 "フロー・ショップとそのスケジュールの解析", Trans of SICE, Vol. 8, No. 5, pp. 40—48 (1972).
- 19) Townsend, W., "Sequencing n Jobs on m Machines to Minimize Maximum Tardiness : A Branch-and-Bound Solution", Management Sci., Vol. 23, No. 9, pp. 1016—1019 (1977).
- 20) Uskup, E. and Smith, S.B., "A Branch-and-Bound Algorithm for Two-Stage Production-Sequencing Problems", Opns. Res., Vol. 23, No. 1, pp. 118—136 (1975).
- 21) Wismer, D.A., "Solution of the Flow-Shop Scheduling Problem with No-intermediate Queues", Opns. Res., Vol. 20, No. 3, pp. 689—697 (1972).
- 22) Lageweg, B.J., Lenstra, J.K. and Rinnooy Kan, A.H.G. "A General Bounding Scheme for the Permutation Flow-Shop Problem", Opns. Res., Vol. 26, No. 1, pp. 53—67 (1978).