



HOKKAIDO UNIVERSITY

Title	初期状態・合成仕事と多機械フロー・ショップ問題の解法
Author(s)	関口, 恭毅
Citation	経済學研究, 37(1), 17-30
Issue Date	1987-06
Doc URL	https://hdl.handle.net/2115/31762
Type	departmental bulletin paper
File Information	37(1)P17-30.pdf



初期状態・合成仕事と 多機械フロー・ショップ問題の解法

関 口 恭 毅

要 約

m 機械問題がある 2 機械問題と等価になるための十分条件のいくつかは従来から知られており、これを満たすときは古典的な Johnson の規則によって最適スケジュールを求めうことは良く知られている。著者はまず、これらの十分条件が満たされる場合であっても、フロー・ショップの各機械がそのスケジュールの処理に着手できる時期、つまり初期状態を考慮に入れると、Johnson 規則によるスケジュールが最適ではない場合のあることを指摘する。次いで、初期状態を明言的にとり入れて、m 機械問題が 2 機械問題と等価になるための十分条件を明らかにする。また、任意のスケジュールが最適スケジュールにくらべ、どの位悪いかを判定する簡単な関数を与える。形式的に構成される、m 機械問題と等価な 2 機械問題が、負の処理時間をも許す Sekiguchi (1983) の 2 機械問題と等価なことが示される。その結果、等価な 2 機械問題を利用する m 機械問題の解法が、仕事に対して処理時間に含まれない段取時間、後始末時間、各種の時間遅れが指定されている場合、グループ・スケジューリング問題となる場合、さらに、これらの問題に先行関係制約が与えられる場合、などにも適用可能なことが明らかにされる。

1. はじめに

順序づけ問題は、製造システムや計算機システムにおけるスケジューリング問題のモデルと解釈できることが多い。これらの現実の状況を思い描くと、順序づけされる対象(仕事)を処理する場(工場)は仕事の処理とともに変化する、ある種の“状態”を持っていると考えるのが自然である。例えば、フロー・ショップ問題では、各機械が利用可能であるかどうか、この状態に相当すると考えることができる。与えられた一群の仕事の処理に着手する時の、工場の状態が、あるスケジュール(順列)に従ってその仕事群を処理する際の効率に影響し、それ故に、最適な順序の決定に影響を与えると考えるのは自然である。しかし、順序づけ問題の分

野では、慣例として問題の記述の中に明言的に初期状態を組み込むことは少ない。わずかに、Gilmore and Gomory [2], Sekiguchi [7-11] が明言的に初期状態を扱っている。

本稿ではフロー・ショップ問題を研究するが、まず、3 機械問題になると、初期状態によって最適スケジュールが変化することを示す。従って、Johnson [3], Burns and Rooker [1], Monma [4] が与えた、3 機械問題がある 2 機械問題と等価となるための十分条件(初期状態を明言的に導入していない)を満たす場合であっても、一般的な初期状態の下では、必ずしもこの 2 機械問題と等価にならないことが明らかになる。初期状態を明言的に扱うことが、理論的にも必要な理由である。

しかる後に 3 節では一般的な初期状態を明言的に扱い、多機械の問題を 2 機械問題として解

くことができるための十分条件を導く。この検討は一般化した時間遅れを利用する Szwarc [13] の分析手法を利用して行う。この方法は、仕事に、処理時間とは独立の開始遅れ、終了遅れ、運搬遅れ、段取時間、後始末時間のいくつかあるいは全部が指定されている問題を統一的に、しかも、簡単に扱うことを可能にする。従って、ここでの結果は Nabeshima and Maruyama [5] の成果を包含するものである。

しかし、この手法では、負の処理時間を有する仕事が発生したり、 k 番目の機械での処理を、 $k-1$ 番目の機械での処理に着手する前に、開始する状況が発生する。そこで4節では、その物理的意味を明らかにし、3節で提案する解法の適用領域を拡大するために、Sekiguchi [9-11] の合成仕事を利用する手法との等価性が指摘される。その結果、3節で提案する(近似)解法を、先行関係が指定される問題やグループ・スケジューリング問題[6, 10]に応用するためには、等価な2機械問題をSekiguchi [10, 11] の方法で解けば良いことが明らかにする。

2. 記 法

$N=\{1, 2, \dots, n\}$, $M=\{1, 2, \dots, m\}$ は各々仕事と機械の集合とする。 $t_{i,k}$ と $a_{i,k}$ ($i \in N$, $k \in M$) は各々、仕事 i の機械 k における処理時間と時間遅れであるとする。ただし、 $a_{i,1}=0$ ($i \in N$) を仮定する。 N は当面順序づけすべき仕事の集合である。 N はある母集合 S , $2M$ 個の実数 $t_{i,k}$, $a_{i,k}$ ($k \in M$) の組から成る要素全体の集合、の部分集合でもあるとする。つまり、 N は仕事につけた番号の集合であるとともに、仕事を定義するパラメータ値の組の集合でもある。

S あるいは N の任意の部分集合の要素の順列を σ で表わす。4節以外では順列とスケジュールは同義語として使われる。順列 σ_1 の直後に別の順列 σ_2 を接続したものを、 $\sigma_1 \cdot \sigma_2$ で表

わす。任意の順列 σ の直後に仕事 i を処理する時、各機械の完了時刻は次式で定まるものとする。

$$\begin{aligned} C^k(\sigma \cdot i) \\ = \max\{C^{k-1}(\sigma \cdot i) + a_{i,k}, C^k(\sigma)\} \\ + t_{i,k} \quad k \in M \quad \dots(1) \end{aligned}$$

ただし、 $C^0(\sigma)=0$ とする。

空の順序を ϕ で記す。 N が与えられたとき、 $C^k(\phi)$ は機械 k が N の中の仕事の処理に利用可能となる時刻であり、利用可能時刻(available time)と呼ぶ。簡単のため、 $C^1(\phi)=0$ とし、これを時刻の原点とする。

$$\begin{aligned} t_{0,k} &= 0 \\ t_{0,k} &= C^k(\phi) - C^{k-1}(\phi) \\ &\quad k \in M - \{1\} \quad \dots(2) \end{aligned}$$

によって、初期状態を定義する。従って、

$$C^k(\phi) = \sum_{p=1}^k t_{0,p} + C^1(\phi)$$

である。

$\langle i \rangle$ を σ の i 番目の仕事であるとする、機械 k におけるこの仕事の完了時刻は $C_{\langle i \rangle}^k(\sigma)$ で表わされる。しかし、以下では簡単のために、特に断らない限り $\sigma=1 \cdot 2 \cdot \dots \cdot n$ となるように、仕事の番号は常につけ替えられるものとする。つまり、 $\langle i \rangle=i$ であるから、 $C^k(\phi)$ を C_0^k , $C_{\langle i \rangle}^k(\sigma)$ を C_i^k を書くことにする。

(1), (2)及び $C^1(\phi)=0$ から、

$$C_1^k = \max\{C_1^{k-1} + a_{1,k}, \sum_{p=1}^k t_{0,p}\} + t_{1,k} \quad \dots(3)$$

である。これから出発して、数学的帰納法を用いれば次式を得る。

$$\begin{aligned} C_n^1 &= \sum_{i=0}^n t_{i,1} \\ C_1^k &= \max_{0 \leq v_1 \leq \dots \leq v_{k-1} \leq 1} \{ \sum_{i=0}^{v_1} t_{i,1} + a_{v_1,2} \\ &\quad + \sum_{i=v_1}^{v_2} t_{i,2} + a_{v_2,3} + \dots + \sum_{i=v_{k-2}}^{v_{k-1}} t_{i,k-1} \\ &\quad + a_{v_{k-1},k} + \sum_{i=v_{k-1}}^1 t_{i,k} \} \\ &\quad k \in M - \{1\} \end{aligned}$$

$$C_n^k = \max_{0 \leq v_1 \leq \dots \leq v_{k-1} \leq n} \left\{ \sum_{i=0}^{v_1} t_{i,1} + a_{v_1,2} \right. \\ \left. + \sum_{i=v_1}^{v_2} t_{i,2} + a_{v_2,3} + \dots + \sum_{i=v_{k-2}}^{v_{k-1}} t_{i,k-1} \right. \\ \left. + a_{v_{k-1},k} + \sum_{i=v_{k-1}}^n t_{i,k} \right\} \quad k \in M - \{1\} \quad \dots(4)$$

C_1^k において $v_1 = v_2 = \dots = v_{k-1} = 1$ となるための条件は,

$$\sum_{p=1}^h (t_{i,p} + a_{i,p+1}) \geq \sum_{p=1}^h t_{0,p+1} \quad \dots(5)$$

$$h=1, 2, \dots, k-1.$$

である。これはどの機械においても、初期状態（つまり、利用可能時刻）が隘路となって最初の仕事の処理開始を遅らせる必要が生じることはない、ということである。 N の任意の順列に対して(5)が成立するための必要・十分条件は,

$$\sum_{p=1}^k (t_{i,p} + a_{i,p+1}) \geq \sum_{p=1}^k t_{0,p+1} \quad \dots(6)$$

$$v_i \in N, \quad v_k \in M - \{m\}$$

が成立することである。

古典的な Johnson 問題 [3] は、形式的に、 $t_{0,k}=0, a_{i,k}=0, t_{i,k} \geq 0 (v_i \in N, v_k \in M)$ であるとき(1)を最小化する問題であるから、そこでは(6)が常に成立している。

(6)が成立していれば、上の C_1^k の式は次のように簡略化される。

$$C_1^k = \sum_{p=1}^k (t_{i,p} + a_{i,p+1}) \quad v_k \in M$$

この式から出発して数学的帰納法を用いれば次式を得る。

$$C_n^k = \max_{1 \leq v_1 \leq \dots \leq v_{k-1} \leq n} \left\{ \sum_{i=1}^{v_1} t_{i,1} + a_{v_1,2} \right. \\ \left. + \sum_{i=v_1}^{v_{k-1}} t_{i,k-1} + a_{v_{k-1},k} + \sum_{i=v_{k-1}}^n t_{i,k} \right\} \quad \dots(7)$$

$$k \in M - \{1\}$$

(4)と(7)で $k=m$ に対し、 $v_1 = v_2 = \dots = v_{k-1} = v$ のところでの右辺の最大値を M_n^m とすると、

$$M_n^m = \max_{0 \leq v \leq n} \left\{ \sum_{i=0}^v t_{i,1} + \sum_{p=2}^{m-1} (t_{v,p} + a_{v,p+1}) \right. \\ \left. + a_{v,m} + \sum_{i=v}^n t_{i,m} \right\} \quad \dots(8)$$

$$M_n^m = \max_{1 \leq v \leq n} \left\{ \sum_{i=1}^v t_{i,1} + \sum_{p=2}^{m-1} (t_{v,p} + a_{v,p+1}) \right. \\ \left. + a_{v,m} + \sum_{i=v}^n t_{i,m} \right\} \quad \dots(9)$$

を得る。そこで、

$$A_i = \sum_{p=1}^{m-1} (t_{i,p} + a_{i,p+1}) \quad \dots(10)$$

$$B_i = \sum_{p=1}^m (t_{i,p} + a_{i,p+1})$$

$$i=0, 1, 2, \dots, n$$

とおくと、次の恒等式が成立する。

$$(8)の左辺の \{ \} 中 = \sum_{i=0}^v A_i - \sum_{i=0}^{v-1} B_i + \sum_{i=0}^n t_{i,m}$$

$$(9)の左辺の \{ \} 中 = \sum_{i=1}^v A_i - \sum_{i=1}^{v-1} B_i + \sum_{i=1}^n t_{i,m}$$

従って、(8), (9)は次のように変形される。

$$M_n^m = \max_{0 \leq v \leq n} \left\{ \sum_{i=0}^v A_i - \sum_{i=0}^{v-1} B_i \right\} + \sum_{i=0}^n t_{i,m} \quad \dots(11)$$

$$M_n^m = \max_{1 \leq v \leq n} \left\{ \sum_{i=1}^v A_i - \sum_{i=1}^{v-1} B_i \right\} + \sum_{i=1}^n t_{i,m} \quad \dots(12)$$

Johnson 問題で $m=3$ の場合、

$$t_{i,2} \leq \min_{1 \leq j \leq n} \{ t_{j,3} \} \quad v_i \in N \quad \dots(13)$$

$$t_{i,2} \leq \min_{1 \leq j \leq n} \{ t_{j,1} \} \quad v_i \in N \quad \dots(14)$$

のいずれか一方が成立する場合、 $C_n^3 = M_n^3$ つまり、3 機械問題が機械 1, 2 の処理時間が A_i, B_i で与えられる 2 機械問題と等価であることが指摘されている [3]。また、その後も数人の研究者が、これとは異なる 2 機械問題と等価になる（従って、Johnson 規則を用いて最適スケジュールを定めることができる）ための十分条件を導いている。先に述べた Johnson 問題の状況から明らかなように、これらの分析は(9)と(12)式の比較に基づくものである（ただし、時間遅れを含まないので、両式で $a_{i,k}$ は削除される）。初期状態は明言的に扱われていないから、どんな初期状態の下でなら、彼らが求めた十分条件が成立するかという議論は存在しない。

しかし、図 1 の例が示すように、たとえ、Johnson が与えた十分条件を満たしたとして

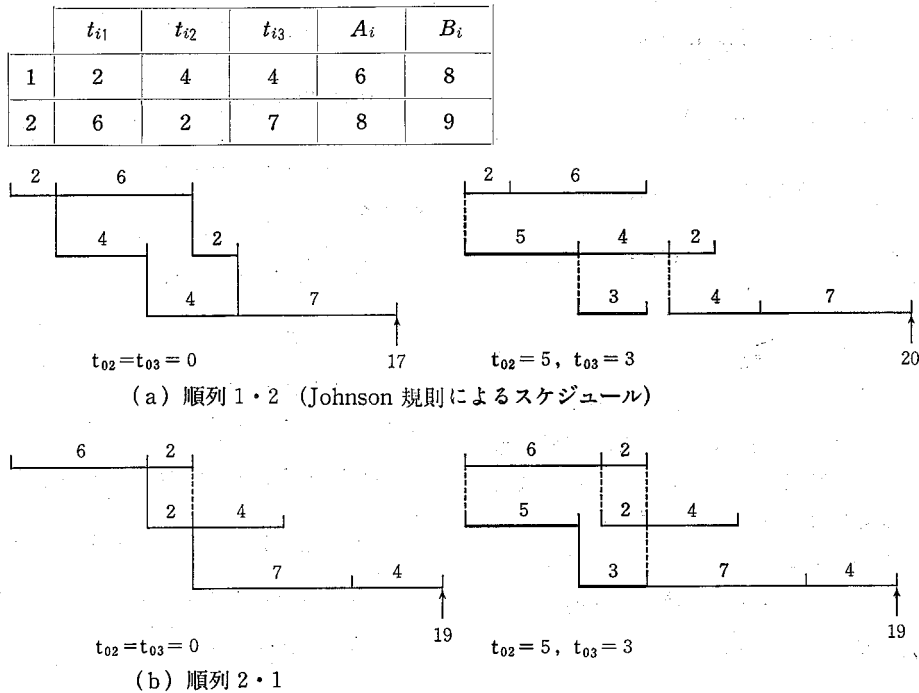


図 1 Johnson 条件(3)を満足するが特定の初期条件の下では 2 機械問題として解くことができない 3 機械問題

も、初期状態によっては、Johnson 規則で求めた順列が最適でないかも知れない。このことが、その他の十分条件についても同様であることを調べるのは容易である。

〔定理 1〕

Johnson [3], Burns-and-Rooker [1], Monma [4] で与えられた、3 機械問題が 2 機械問題と等価になるための十分条件は、一定範囲内の初期状態の下でのみ正しい。

定理 1 は、3 機械問題の最適順列が初期状態に依存することの一つの反映であり、 $m \geq 3$ であれば、常に、最適順列が初期状態に依存することが観察できる。一般に、工場に新しい一群の仕事が投入される瞬間に、仕掛中の仕事が存在することは、現実には良くあることである。従って、 m 機械問題の記述に初期状態を明言的

に導入することは、理論的にも、実際的にも、意義の認められるところである。

3. m 機械問題の解きうる場合

(1), (2)の M_n^m は、右辺の第 1 項を最小化する順列によって、最小化される。優先関係 $\leq$ を次のように定義する。

$$i \leq j \quad \text{iff} \quad \begin{cases} A_i \leq B_i, A_j \leq B_j, A_i \leq A_j, \text{ or} \\ A_i \leq B_i, A_j \geq B_j, \text{ or} \\ A_i \geq B_i, A_j \geq B_j, B_i \geq B_j \end{cases}$$

もし、 $i \leq j$ だが $j \leq i$ は成立しないなら、 $i < j$ と書くことにする。Sidney の一般化順序づけ定理 [12] を M_n^m に適用すれば、 M_n^m は「 $i < j$ なら i は j に先行する順列」によって最小化される。これは Johnson 規則の一般化である。

M_n^m を最小化する N の中の仕事の順列を σ^*

と書き, M_n^m の最小値を $M_n^m(\sigma^*)$ と書くものとする。 M_n^m は, (4)あるいは(7)の右辺に $v_1 = \dots = v_{m-1}$ という制限を追加して導いたものであるから, 明らかに

$$C_n^m \geq M_n^m \quad \dots(15)$$

が任意の順列に対して成立している。従って,

$$C_n^m(\sigma^*) = M_n^m(\sigma^*) \quad \dots(16)$$

が成立するときには σ^* はもとの m 機械問題の最適順列にもなっている。しかし, 当然, σ^* は必ずしも C_n^m を最小化しない。 C_n^m を最小化する順列を σ^0 と書くことにすれば, 次の定理は簡単に証明できる。

〔定理2〕

$$C_n^m(\sigma) - C_n^m(\sigma^0) \leq C_n^m(\sigma) - M_n^m(\sigma^*) \quad \dots(17)$$

〔証明〕

$$C_n^m(\sigma) - C_n^m(\sigma^0) \leq C_n^m(\sigma) - M_n^m(\sigma^0)$$

(15)による,

$$\leq C_n^m(\sigma) - M_n^m(\sigma^*) //$$

$$U(\sigma) = C_n^m(\sigma) - M_n^m(\sigma^*) \quad \dots(18)$$

を定義する。 $U(\sigma)$ は σ の (σ^0 からの) 乖離上界である。乖離上界は一意ではないから他の形のものも後述するが, σ^* が C_n^m を最小化する場合には $U(\sigma^*) = 0$ であり, その意味で最良の上界である。

ある順列 σ を構成するのに必要な計算量はオーダーとしては, N の要素のソートに必要な量と同じであると仮定する。すると, σ が C_n^m を最小化する順列に近いかどうかを判定するには次の2段階手続きを実行すれば良い。

- (1) σ を構成する
- (2) $U(\sigma)$ を評価する

第2段階の実行には一般に(1)の実行よりも必要計算量のオーダーが大きくなるから, この手続きは, $O(n \log n)$ の計算量で十分実行できる。順列 σ を求めて, これを近似解として利用しようというのであれば, $U(\sigma)$ がいわば近

似の程度を表わすと考えられるので, 極めて, 好都合であろう。

m 機械問題の近似最適順列として, M_n^m を最小化する σ^* を利用する方法は有望である。 σ^* が真に m 機械問題の最適順列であることが保証される, 例えば定理1の十分条件が正しい状況下では, $U(\sigma^*) = 0$ が必ず成立するので, 定理2と上の2段階手続きは, 実用的には, $C_n^m = M_n^m$ となる十分条件を求めて, その成否を判定する方法以上に効果的であると考ええる。一つには, 求めた十分条件を満たさない場合でも, $U(\sigma^*) = 0$ となることがあるからである。

定理2は Nabeshima and Maruyama [5] の Remark (p. 358) の直接的な一般化である。 σ が任意であるという点と, C_n^m , M_n^m は任意の時間遅れを含んでいるという2点で。以下に見るように, 処理時間 $t_{i,k}$ と時間遅れ $a_{i,k}$ を適切に定義してやれば, (1)は, 開始遅れ $d_{i,k}$, 完了遅れ $d'_{i,k}$, 運搬遅れ $b_{i,k}$, 段取時間 $s_{i,k}$, 後始末時間 $r_{i,k}$ ($b_{i,k}$ 以外は非負) のすべて, あるいは, 一部を含む場合のフロー・ジョブにおける完了時刻に一致する。特に, $m=2$ の場合には $C_n^2 = M_n^2$ が常に成立するから, σ^* を Sidney の一般化順序づけ定理によって求めるアルゴリズムは, [5] の2機械問題の解と一致する。さらに, 運搬遅れは開始遅れ/完了遅れのどちらによっても表現することができるから, このアルゴリズムは2機械問題に直・並列先行関係が指定されている場合をも解いた Sekiguchi [10] の結果に包含されるものである。

機械 k における仕事 i の段取は, この仕事の他の機械での処理の進行とは独立に実施できると仮定されているから, 機械 k が仕事 i の処理に着手できる時刻は $C_{i,k-1}^k + s_{i,k}$ で与えられる。後始末も同様の性質を持ち, 仕事 i の機械 k での処理が終了した時刻に着手可能である。従って, 機械 $k-1$ が仕事 i の処理を完了する時刻は, $C_{i,k-1}^{k-1} - r_{i,k-1}$ で与えられる。各処理は

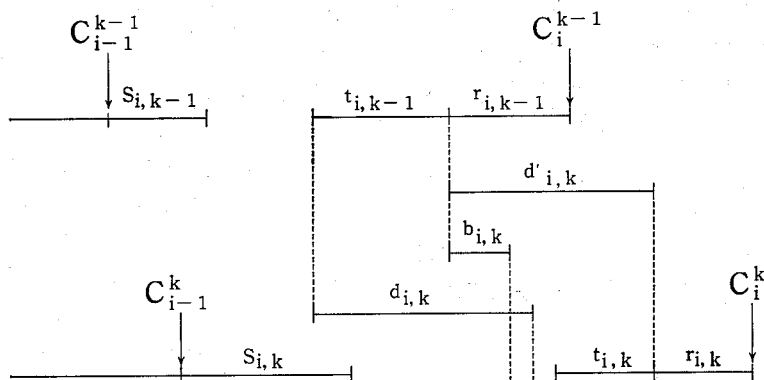


図2 処理時間とは独立な5つのパラメータが指定された時の完了時刻

中断なく完了されると仮定すると、機械 $k-1$ が仕事 i の処理に着手する時刻は $C_{i-1}^{k-1} - r_{i,k-1} - t_{i,k-1}$ である。機械 $k-2$ における処理の進行状況との干渉によって、 $s_{i,k-1}$ の完了と $t_{i,k-1}$ の開始の間には遊休が存在するかも知れないが、 $t_{i,k-1}$ の完了と $r_{i,k-1}$ の着手の間には決して遊休を生じない。開始遅れ $d_{i,k}$ は、機械 $k-1$ における仕事 i の処理着手後、機械 k においてこの仕事の処理に着手するまでに経過すべき最小時間である。運搬時間 $b_{i,k}$ と完了遅れ $d'_{i,k}$ は、各々、機械 $k-1$ における仕事 i の処理完了後、機械 k においてこの仕事に着手あるいは完了するまでに経過すべき最小時間である。従って、機械 $k-1$ が隘路となる場合には、機械 k における仕事 i の完了時刻は

$$C_{i-1}^{k-1} - r_{i,k-1} - t_{i,k-1} + d_{i,k} + t_{i,k}$$

$$C_{i-1}^{k-1} - r_{i,k-1} + b_{i,k} + t_{i,k}$$

$$C_{i-1}^{k-1} - r_{i,k-1} + d'_{i,k}$$

の中の最大値に一致する。他方、機械 k が隘路となる場合には、この完了時刻は

$$C_{i-1}^{k-1} + s_{i,k} + t_{i,k}$$

となる (図2参照)。以上から、機械 k において、仕事 i の後始末が完了する時刻は、

$$C_i^k = \max \left\{ \begin{array}{l} C_{i-1}^{k-1} - r_{i,k-1} - t_{i,k-1} + d_{i,k} + t_{i,k} \\ C_{i-1}^{k-1} - r_{i,k-1} + b_{i,k} + t_{i,k} \\ C_{i-1}^{k-1} - r_{i,k-1} + d'_{i,k} \\ C_{i-1}^{k-1} + s_{i,k} + t_{i,k} \end{array} \right\} + r_{i,k} \quad \dots (19)$$

である。もし、上のようなパラメータを指定された仕事を(20)の時間遅れと処理時間を持つ仕事 (これは一種の合成仕事である) で置き換えるものとする、(1)で計算される完了時刻を最小化する問題と、(19)を最小化することは等価である。

$$a_{i,k} = \max \left\{ \begin{array}{l} d_{i,k} - t_{i,k-1} \\ d'_{i,k} - t_{i,k} \\ b_{i,k} \end{array} \right\} - r_{i,k-1} - s_{i,k} \quad \dots (20)$$

$$t'_{i,k} = s_{i,k} + t_{i,k} + r_{i,k}$$

実際、(20)を代入すれば(19は次のように変形される。

$$C_i^k = \max \{ C_{i-1}^{k-1} + a_{i,k}, C_i^k \} + t'_{i,k}$$

5つのパラメータのうち、指定されないものがあるときには、仮りに次のような値を割り当てて(20)を用いれば良い。

- $s_{i,k}$ ($r_{i,k}$) が指定されないときはこれを0とおく。
- $d_{i,k}$ ($d'_{i,k}$) が指定されないときは $d_{i,k} = t_{i,k-1}$ ($d'_{i,k} = t_{i,k}$) とおく。
- $b_{i,k}$ が0あるいは指定されないときは、これを負の無限大とおく。

こうする理由は、これらのパラメータが完了時刻に及ぼす効果を考えれば明らかであろう。

(20)で定義される (合成) 仕事は、処理時間は常に非負であるが、時間遅れは負となることもあり得る。特に、時間遅れが負で絶対値が処理

時間にくらべて大きいときには、機械 $k-1$ で処理に着手する以前に、機械 k で処理に着手するということも起こり得る。この伝統的なフロー・ショップの概念からすると理解しにくい状況の意味の説明は次章で行う。

次に、 $C_n^3 = M_n^3$ となるための十分条件を、初期状態を明言的に組み込んで求める。 $m \geq 4$ の場合にも同様な検討が可能であるが、得られる十分条件は簡潔ではなく、前述の 2 段階手続きの計算量の少さから考えて、実用的な意義は乏しい。以下の $m=3$ の場合の検討は、初期状態と一般化した時間遅れを組み込むと、従来から知られている結果、例えば (13), (14) とどう異なってくるかを理解するのに有効であろう。

(6) が成立する (これを制限付きの場合という [13]) 場合をまず考える。 C_n^3 と M_n^3 は、各々 (7), (9) で与えられる。(13) が成立しているから、 $C_n^3 \leq M_n^3$ が $C_n^3 = M_n^3$ と等価である。このための十分条件は、(7) と (9) を比較することにより、次式が成立することである。

$$\sum_{i=1}^{v_1} t_{i,1} + a_{v_1,2} + \sum_{i=v_1+1}^{v_2} t_{i,2} + a_{v_2,3} + \sum_{i=v_2+1}^n t_{i,3} \\ \leq \max \left\{ \begin{array}{l} \sum_{i=1}^{v_1} t_{i,1} + a_{v_1,2} + t_{v_1,2} + a_{v_1,3} + \sum_{i=v_1+1}^n t_{i,3} \\ \sum_{i=1}^{v_2} t_{i,1} + a_{v_2,2} + t_{v_2,3} + a_{v_2,3} + \sum_{i=v_2+1}^n t_{i,3} \end{array} \right\} \quad (21)$$

(任意のスケジュール, $1 \leq v_1 < v_2$)

この式を整理すれば、

$$\left. \begin{array}{l} \sum_{i=v_1+1}^{v_2} (t_{i,2} + a_{i,3}) \leq \sum_{i=v_1+1}^{v_2-1} (t_{i,3} + a_{i,3}), \text{ or } \\ \sum_{i=v_1+1}^{v_2-1} (t_{i,2} + a_{i,2}) \leq \sum_{i=v_1+1}^{v_2} (t_{i,1} + a_{i,2}) \end{array} \right\} \quad (21)$$

(任意のスケジュール, $1 \leq v_1 < v_2$)

(21) が成立するための十分条件を求めるには、時間遅れがない場合の Johnson [3] や Burns and Rooker [1] のやり方を模倣すれば良い。Johnson を模倣すれば (22), (23) を得、Burns and Rooker をまねれば (24) を得る。これらはいづれも、任意のスケジュールの下で、 $v_1 < v_2$ の値が

なんであっても (21) が成立するための十分条件の代表例である。

$$t_{i,2} + a_{i,3} \leq \min_{\substack{1 \leq j \leq n \\ j \neq i}} \{t_{j,3} + a_{j,3}\} \quad \forall i \in N \quad \dots (22)$$

$$t_{i,2} + a_{i,2} \leq \min_{\substack{1 \leq j \leq n \\ j \neq i}} \{t_{j,1} + a_{j,2}\} \quad \forall i \in N \quad \dots (23)$$

$$t_{i,2} + \max\{a_{i,2}, a_{i,3}\} \leq \min\{t_{i,2} + a_{i,2}, t_{i,3} + a_{i,3}\} \quad \forall i \in N \quad \dots (24)$$

(6) が成立しない、いわゆる制限されない場合には、(4) と (8) を各々 (7), (9) の代わりに使って、同様の分析を行えば、次の (25) ~ (27) を得る。

$$t_{i,2} + a_{i,2} \leq \min_{\substack{0 \leq j \leq n \\ j \neq i}} \{t_{j,3} + a_{j,3}\} \quad \forall i \in N \quad \dots (25)$$

$$t_{i,2} + a_{i,2} \leq \min_{\substack{1 \leq j \leq n \\ j \neq i}} \{t_{j,1} + a_{j,2}\} \quad \forall i \in N \cup \{0\} \quad (26)$$

$$t_{i,2} + \max\{a_{i,2}, a_{i,3}\} \leq \min\{t_{i,1} + a_{i,2}, t_{i,3} + a_{i,3}\} \quad \forall i \in N \cup \{0\} \quad (27)$$

ちょっと考えると、(25) ~ (27) に、(6) が成立するという制限を加えると、(22) ~ (24) が導びかれそうに思えるが、これは実は、正しくない。 $m=3$ の場合、(6) は具体的には (28) になる。

$$\left. \begin{array}{l} t_{0,2} \leq t_{i,1} + a_{i,2} \quad (\forall i \in N) \\ t_{0,2} + t_{0,3} \leq t_{i,1} + a_{i,2} + t_{i,2} + a_{i,3} \quad (\forall i \in N) \end{array} \right\} \quad (28)$$

(26) と (28) が同時に成立すれば (25) を得る。しかし、(25), (27) に (28) を追加しても、対応する (22), (24) を生じない。かくして次の定理が確立された。

〔定理 3〕

(1) を最小化する 3 機械問題は、(6) と (22), (23), (26), (6) と (24), (27) のどれかが成立すれば、(1) あるいは (2) を最小化する 2 機械問題と等価である。

定理 3 は Johnson や Burns and Rooker の結果にくらべて、初期状態と任意の時間遅れを含んでいる点で一般化されている。(20) による合成仕事を利用すれば、先に述べたようなパラメータを含んだ問題に対しても、適用可能な定理である。

時間遅れがなく、仕事 i の機械 1, 2 での処理時間が $A_i, B_i (i \in N \cup \{0\})$ であるとき、2 機械問題の完了時刻は次式の T_n^2 である [3]。

$$T_n^1 = \sum_{i=0}^n A_i$$

$$T_n^2 = \max\left\{\sum_{0 \leq v \leq n} A_i - \sum_{i=0}^{v-1} B_i\right\} + \sum_{i=0}^n B_i \quad \dots(29)$$

$$T_n^3 = \max\left\{\sum_{1 \leq v \leq n} A_i - \sum_{i=1}^{v-1} B_i\right\} + \sum_{i=1}^n B_i \quad \dots(30)$$

(30)は(6)が成立する場合のもの。これらは、(4)、(7)で $m=2$, $a_i, k=0$, $A_i=t_{i,1}$, $B_i=t_{i,2}$ といった式と一致する。(29), (30)を各々(11), (12)と比較すれば、 M_n^3 は T_n^2 に対し、定数だけ異っている。尚、 T_n^2 の最後の項は機械2が busy 状態になる総時間であるから、 $\max$ の項は機械2の総遊休時間である。

C_n^m を最小化する問題が2機械問題と等価になるためのもう1つの十分条件が、Monma [4] に示唆されている方法を模倣することによって得られる。

(4)が次のように変形できることに注目する。

$$C_n^m = \max_{0 \leq v_1 \leq \dots \leq v_{m-1} \leq n} \left\{ \sum_{p=1}^{m-1} \left[\sum_{i=0}^{v_p} (t_{i,p} + a_{i,p+1}) - \sum_{i=0}^{v_{p-1}} (t_{i,p+1} + a_{i,p+1}) \right] + \sum_{i=0}^n t_{i,m} \right\} \quad \dots(31)$$

$$A_i = t_{i,p} + a_{i,p+1}, \quad B_i = t_{i,p+1} + a_{i,p+1} \quad \dots(32)$$

とおくと、(31)は、(11)に良く似た式になることに注目する。もし、

$$t_{i,p} + a_{i,p+1} \leq t_{j,p+1} + a_{j,p+1} \quad \dots(33)$$

$$\forall i \in N, \forall j \in N^u \setminus \{0\}$$

が成立すれば、任意の順列に対し、

$$(31)の [] の中 \leq t_{0,p} + a_{0,p+1} \quad \dots(34)$$

であり、

$$t_{i,p} + a_{i,p+1} \geq t_{j,p+1} + a_{j,p+1} \quad \dots(35)$$

$$\forall i \in N, \forall j \in N^u \setminus \{0\}$$

が成立するときには、任意の順列に対し、

$$(31)の [] の中 \leq \sum_{i=0}^n (t_{i,p} + a_{i,p+1}) - \sum_{i=0}^{n-1} (t_{i,p+1} + a_{i,p+1}) \quad \dots(36)$$

が成立する。

さて、ある q ($1 \leq q \leq m-1$) が存在して

$$p(1 \leq p \leq q-1) \text{ に対し } (33)$$

$$p(q+1 \leq p \leq m-1) \text{ に対し } (35)$$

が成立するものとする、(31)は次のように簡単化される。

$$C_n^m = \sum_{p=1}^{q-1} (t_{0,p} + a_{0,p+1}) + \max_{0 \leq v_q \leq n} \left\{ \sum_{i=0}^{v_q} (t_{i,q} + a_{i,q+1}) - \sum_{i=0}^{v_q-1} (t_{i,q+1} + a_{i,q+1}) \right\} + \sum_{i=0}^n t_{i,q+1} + \sum_{p=q+2}^m (t_{n,p} + a_{n,p})$$

$$\equiv W_{n,q}^m \quad \dots(37)$$

$W_{n,q}^m$ の式(37)でスケジュールに依存するのは第2, 4項である。(37)は、(32)の代入をほどこせば、(11)と非常に良く似た式になり、特に $q=m-1$ であれば第4項が消滅するので、第1項の定数項が異なるだけである。

[定理4]

ある q ($1 \leq q \leq m-1$) が存在し、 p ($1 \leq p \leq q-1$) に対し(33), p ($q+1 \leq p \leq m-1$) に対し(35)が成立するとき、 C_n^m を最小化する問題は、 $W_{n,q}^m$ を最小化する問題と等価である。特に、 $q=m-1$ であれば、後者の問題は(32)により A_i, B_i を定めたとき、 M_n^m (T_n^2) を最小にする問題と等価である。

$W_{n,q}^m$ の第4項は順列の最後の仕事にのみ依存することに着目すれば、 $W_{n,q}^m$ が次の手続きによって定まる順列 σ_q^* によって最小化されることは明らかである。

(1) $i \in N$ に対し(32)によって A_i, B_i を定め、優先関係 $\leq$ を用いて Sidney の一般化順序づけ定理によって1つの順列を定めよ。その中の仕事 i ($i \in N$) を最後尾に回わし、他の順序は変えないで作った順列を σ_q^i とする。

(2) σ_q^i ($i \in N$) の中で、 $W_{n,q}^m$ の最小値を与えるものを σ_q^* とせよ。

この手続きの(1)の所要計算量は $O(n \log n)$ 程度と評価されるが、(2)は $O(n^2)$ 程と考えられる。従って、オーダー的には先に述べた M_n^m を利用する場合にくらべると若干大きな計算量を

必要とする。但し、 $q=m-1$ のときはその限りでない。

σ^* および $\sigma_q^*(q=1, 2, \dots, m-1)$ をすべて構成し、その中の最良のものを採用するには、必要な計算量は

$$O(n \log n) + (m-2) O(n^2) + O(n \log n) \\ \sigma^* \quad \sigma_1^* \sim \sigma_{m-2}^* \quad \sigma_{m-1}^*$$

程度である。この時得られる最良の順列の近似度は、後述の定理 6 によって評価される。

$W_{n,q}^m$ は C_n^m の式に、 $v_1=v_2=\dots=v_{q-1}=0$, $v_{q+1}=v_{q+2}=\dots=v_{m-1}=n$ なる制限を追加して得たものであるから、明らかに

$$C_n^m \geq W_{n,q}^m \quad \forall q \in M - \{m\} \quad \dots (38)$$

が任意の順列に対して成立する。従って、定理 2 の場合と同様にして、次の定理を得る。

〔定理 5〕

$$C_n^m(\sigma) - C_n^m(\sigma^0) \leq C_n^m(\sigma) - W_{n,q}^m(\sigma_q^*) \quad \dots (39) \\ \forall q \in M - \{m\}$$

$$V_q(\sigma) = C_n^m(\sigma) - W_{n,q}^m(\sigma_q^*) \quad \dots (40)$$

と定義すれば、これも $U(\sigma)$ と同じく σ の乖離上界であり、定理 4 によれば、 $U(\sigma)$ がそうであるのと同じ意味で、最良の上界である。つまり、 q が定理 4 の条件を満たすとき $V_q(\sigma_q^*) = 0$ である。

(39) を検討すれば、 $V_q(\sigma_q^*) = 0$ を満たす q に対し、機械 q と $q+1$ が隘路を構成していることが理解できる。これに対し、 $U(\sigma^*) = 0$ のとき、機械 1 と機械 m が隘路となる。類推すれば、これら以外の機械 k と $q (k < q)$ が隘路となる場合を検討することによって、 C_n^m を最小化する問題が 2 機械問題と等価になる別の十分条件を発見することができるであろう。

定理 2 と定理 5 を結合すれば容易に、次の定理が正しいことがわかる。

〔定理 6〕

$$C_n^m(\sigma) - C_n^m(\sigma^0) \\ \leq C_n^m(\sigma) - \max\{C_n^m(\sigma^*), W_{n,1}^m(\sigma_1^*), \dots, W_{n,m-1}^m(\sigma_{m-1}^*)\} \quad \dots (41)$$

a_{ip} は負であることも許されるから、(10) や (32) の A_i , B_i は負になる可能性がある。負であっても数学的には全く困難を生じないが、しかし、処理時間が負であるときの完了時刻 T_n^2 が何を表わしているかを知ることは実用上興味深い。この点は次章で述べる。

4. 負の処理時間・時間遅れおよび合成仕事

前節の解法では、 m 機械問題を解くのに利用する 2 機械問題に、負の処理時間を持つ合成仕事が生じる可能性がある。また、処理時間に含まれないパラメータを持つ仕事を適当な合成仕事で置き換えて扱うとき、第 $k-1$ 機械での処理に着手する以前に第 k 機械での処理に着手するという場合が生じる。この節では、これらのことが発生する状況を少し詳しく検討し、その実際の意味を明らかにする。

N は次のような仕事集合 $G_i (i=1, 2, \dots, n)$ の集合であると仮りに考える。 $G_i = G1_i \cup G2_i \cup G12_i$ である。ただし、

$G1_i$: 機械 1 における処理だけで完成する仕事 (type-1 の仕事) の集合

$G2_i$: 機械 2 における処理だけで完成する仕事 (type-2 の仕事) の集合

$G12_i$: 機械 1 について機械 2 で処理されて完成する仕事 (type-12) の仕事の集合。
各仕事は、機械 2 における着手が機械 1 における着手に先行しない範囲の時間遅れを持っても良い。

G_i 中の仕事を処理するスケジュールは、機械 1 では機械 2 にはない type-1 の仕事があり、逆に機械 2 には type-2 の仕事があるので、機械 1 と機械 2 では異った順列となる。しかし、

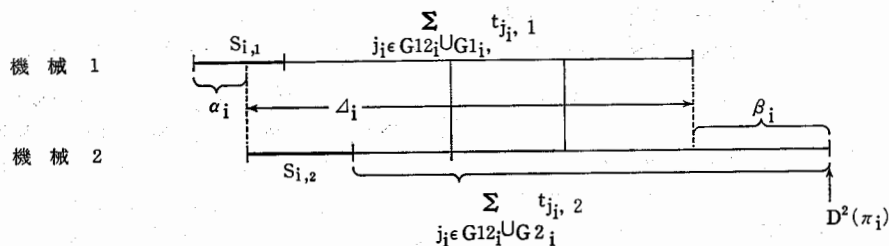


図 3 Sekiguchi [10] の合成仕事

type-12 の仕事の順序は、機械 1 と機械 2 で一致しているものとする。機械 1 では type-12 の仕事の順列の間に type-1 の仕事がある、また、機械 2 では type-2 の仕事がある、はさまっても良いとする。このような条件を満たす機械 1 と 2 のスケジュールの対をグループ内順序と呼ぶ。 G_i の仕事の処理に先立って、すべての仕事に共通の一定の段取りが必要であり、 G_i の全仕事は、連続して（途中で他のグループの仕事の処理をはさまずに）処理されなければならない。 G_i の段取り時間を $S_{i,1}$, $S_{i,2}$ とする。そこで、 N に含まれる各仕事集合に含まれるすべての仕事を処理するには、各仕事集合のグループ内順序とともに、仕事集合の処理順序つまり N の要素の順列（これをグループ間順序と言う）が与えられる必要がある。何らかの意味で最適なグループ間、ならびに、グループ内順序を求める問題をグループ・スケジューリング問題と呼ぶ[6]。各タイプの仕事の完了時刻は次のように算定されるとする。

タイプ-1 の仕事： $D_{j_i}^1 = D_{j_{i-1}}^1 + t_{j_i,1}$

$$D_{j_i}^2 = D_{j_{i-1}}^1$$

タイプ-2 の仕事： $D_{j_i}^1 = D_{j_{i-1}}^1$

$$D_{j_i}^2 = D_{j_{i-1}}^2 + t_{j_i,2}$$

タイプ-12 の仕事：

$$D_{j_i}^1 = D_{j_{i-1}}^1 + t_{j_i,1}$$

$$D_{j_i}^2 = \max\{D_{j_i}^1 + a_{j_i,2}, D_{j_{i-1}}^2\} + t_{j_i,2}$$

ここで $D_{j_{i-1}}^k$ は機械 k の j_i に対する利用可能時刻であり、 j_i は G_i の仕事 j である。 $t_{j_i,k}$, $a_{j_i,k}$ は処理時間、時間遅れを示す。この完了

時刻の算定式は (1) と整合している（タイプ-12 の $D_{j_i}^k$ は、(1) と一致する）。

G_i のグループ内順序 π_i が与えられたとき、機械 1 における段取り作業終了時点時刻を時刻 0 とし、初期状態 $D_0^1 = 0$, $D_0^2 = -\infty$ とし、算定される π_i の完了時刻を $D^2(\pi_i)$ とする。次式によって 3 つ組 $(\alpha_i, \beta_i, \Delta_i)$ を定義する（図 3 参照）。

$$\left. \begin{aligned} \alpha_i &= D^2(\pi_i) - \sum_{j_i \in G_{12i} \cup G_{2i}} t_{j_i,2} + S_{i,1} - S_{i,2} \\ \beta_i &= D^2(\pi_i) - \sum_{j_i \in G_{12i} \cup G_{1i}} t_{j_i,1} \end{aligned} \right\} \quad (42)$$

$$\Delta_i = D^2(\pi_i) + S_{i,1} - \max\{\alpha_i, 0\} - \max\{\beta_i, 0\}$$

第 1, 第 2 機械における処理時間が α_i, β_i である仕事は、グループ G_i のグループ内順序 π_i に対応する合成仕事と言われる [10]。グループ間順序 $\sigma = 1 \cdot 2 \cdot \dots \cdot n$ が与えられた時、合成仕事 i の完了時刻は次式により定まるとする。

$$R_i^1 = R_{i-1}^1 + \max\{\alpha_i, 0\} - \min\{0, \beta_i\}$$

$$R_i^2 = \max \left\{ R_{i-1}^1 + \max\{\alpha_i, 0\} \right\} + \max\{\beta_i, 0\} \\ \left\{ R_{i-1}^2 - \min\{0, \alpha_i\} \right\} \quad \dots (43)$$

$\alpha_i \geq 0$, $\beta_i \geq 0$ であれば (43) は (1) で $a_{i,p} = 0$ の場合に一致する。 $D_n^k(\sigma)$ を、各グループ内順序を π_i , グループ間順序を σ とするとき、 N のすべての仕事を処理完了する時刻とし、 $R_n^k(\sigma)$ は各グループ内順序に対応する合成仕事を順列 σ に従って処理するとき、(43) によって定められる完了時刻とする。この時、次式が成立する [10] (また [9] も見よ)。

$$D_n^1(\sigma) = R_n^1(\sigma) + \sum_{i \in N} \Delta_i$$

$$D_n^2(\sigma) = R_n^2(\sigma) + \sum_{i \in N} \Delta_i \quad \dots (44)$$

α_i, β_i は、一方あるいは両方が負になる可能性がある。タイプ-12の仕事の定義によれば

$$D^2(\pi_i) \geq \sum_{j \in a_{12i}} t_{j,i,k} \quad (k=1, 2)$$

が成立するから、負の α_i は、機械2における段取時間 $S_{i,2}$ やタイプ-2の仕事の処理時間の反映である。また、負の β_i はタイプ-1の仕事の処理時間の反映である。従って、負の α_i, β_i は各々、機械2、機械1の処理時間と考えるのが自然であろう。(43)はそのような観点から定義されている。実際、(44)は、(42)によって定められた完了時刻が、本来の完了時刻 $D^k(a)$ に対し、一定のバイアス $\sum \Delta_i$ だけ異なるにすぎないことを示し、上記の考え方の妥当性を示している。言い換えれば、(43)は、 D^k の計算式における機械間の干渉に関する構造を保持したまま、両方の機械が共に必然的に busy (あるいは遊休) 状態になる期間 Δ_i の分を省略したものと考えることができる。

前節で検討した、5つのパラメータを持つ仕事 i は、 $m=3$ のときタイプ-1の仕事 1_i (処理時間 $r_{i,1}$) とタイプ-12の仕事 2_i (機械1, 2の処理時間は各々 $t_{i,1}, t_{i,2}+r_{i,2}$ 、時間遅れが $\max\{d_{i,2}-t_{i,1}, d'_{i,2}-t_{i,2}, b_{i,2}\}$) から成り、段取時間が $s_{i,1}, s_{i,2}$ であるような仕事集合であると解釈することができる。グループ内順序 π_i として、機械1では $2 \cdot 1_i$ 、機械2では 2_i であるものを使って、 $(\alpha_i, \beta_i, \Delta_i)$ をこの仕事に対して定義できることに注意する。具体的には α_i, β_i は次式で与えられる。

$$\begin{aligned} \alpha_i &= t'_{i,1} + a_{i,2} \\ \beta_i &= t'_{i,2} + a_{i,2} \end{aligned} \quad \dots(45)$$

ここで $t'_{i,k}, a_{i,2}$ は (20) で定められる値である。

以上の準備をしたうえで、2機械フロー・ショップにおいて、(20) の $a_{i,k}$ が負で絶対値が大きく、第1の機械で着手するより前に、第2の機械(もし、利用可能であれば)で着手する状況を例によって検討する。図4を参照する。

(a)は2機械問題の2つの仕事を示しており、(b)はその左半分に(20)で定義される合成仕事を、

右半分は(42)によって上述の考え方で定めた合成仕事を示している。機械1, 2の利用可能時刻がともに0であるとき、仕事1の処理に引き続き仕事2を処理する経過を、(19)の計算式に従ってガントチャートに表わしたのが(c)である。(20)によって定義される合成仕事を同じく1・2の順に処理する経過を、(1)に従ってガント・チャートに表現したのが(d)である。また、(42)による合成仕事を同じく1・2の順で(43)に従って処理する際のガント・チャートが(e)である。

(c)の $s_{i,2}$ に相当する処理が右づめで実行されていることを除くと、(d)は(c)と全く同一の進行状況を表わしている。これに対し、(e)は(d)の ~~+++++~~ で示した部分をそっくり削除して左右を短絡させた形である。~~+++++~~ の部分は両方の機械が共に仕事1あるいは仕事2の処理で busy な時であり、機械間の干渉は(e)で十分表現されていることが理解される。

(c)を見ると、仮りに機械2が遊休状態であるために仕事2の段取作業の開始を、あと1単位時間早める(左へずらす)ことができれば、全体の完了時刻も1単位短縮できる。しかし、たとえ2単位以上仕事2の段取着手を早めても、仕事2の完了遅れ制約とぶつかって、全体の完了時刻は1単位しか短縮できない。結局、仕事2の完了時刻は、機械2での段取着手を機械1での着手より3単位以内で先行させれば、先行させた分だけ短縮されることがわかる。これが、 $t'_{2,1} + a_{2,2} = -3$ ((20) による合成仕事2の機械2での処理着手は機械1での着手より3単位まで先行させることができる)、および $\alpha_2 = -3$ (負の処理時間) の意味である。

(45)が成立することは、(b)で確認できる。 $W_{\pi,q}^m$ における機械1, 2の処理時間である(32)の A_i, B_i が負であるとき、(32)と(45)の類似性に着目すれば、これは α_i, β_i が負である時の意味と同じであることがわかる。また、(10)の A_i, B_i は $m=2$ であるとき(32)の $p=1$ であるときと一致する。このことから、(10)の A_i, B_i が負になるときの意味も、(32)の場合と類似であることが理解

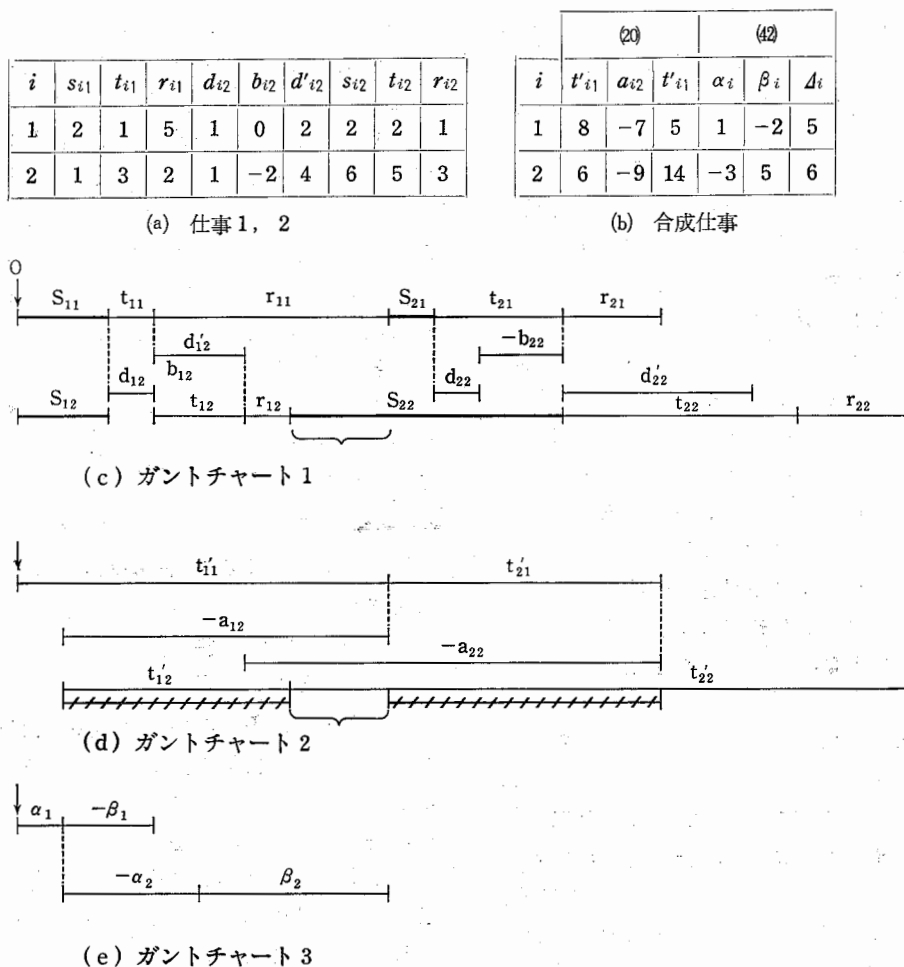


図 4 大きな負の時間遅れ, 負の作業時間

される。

5. 先行関係制約件がある場合の多機械問題の近似解法

Sekiguchi [10] によれば機械 1, 2 の処理時間が α_i, β_i で与えられる仕事の集合の (43) を最小化する順序は, 3 節の冒頭で定義した優先関係 $\leq$ を用い, Sidney の一般化順序づけ定理に基づいて定めることができる。また, $W_{n,m-1}^m$ や M_n^m を最小化する, つまり, (10) や (32) によって A_i, B_i を定義するとき, (29), (30) を最小化する問題も同じ優先関係 $\leq$ と定理に基づいて, 最

適順序が定められる [3] (Johnson の分析は A_i, B_i が正であると仮定して進められているが, 負の場合であっても困難を生じない)。このことは, (29) と (43) が本質的に同じものであることを示唆している。実際,

$$T_n^p(\sigma) = R_n^p(\sigma) + \sum_{i \in N} \{ \min(0, \beta_i) + \min(0, \alpha_i) \} \quad p=1, 2 \quad (46)$$

が成立する。従って, $W_{n,m-1}^m, M_n^m$ を最小化することは, R_n^2 を最小化することと同値である。 $W_{n,q}^m$ ($q \leq m-1$) を最小化するのも R_n^2 に対応する変形を施したものを最小化するのと, 同値

であることがわかる。以上により、3節で構成した2段階手続きの(1)に、(4)で定義される合成仕事を利用して得られた、Sekiguchi [9, 10], さらに、処理時間がすべて非負の場合の Sekiguchi [11] の分割アルゴリズムに関する結果が、そっくり適用可能であることが示された。

M_n^m や $W_{n,q}^m$ は C_n^m の式における $v_1 \sim v_{m-1}$ の変域を制限することによって作られたものであるから、もとの m 機械問題に対して先行関係が追加された場合、グループ・スケジューリング問題である場合などにも、定理2, 5, 6が成立する。そこで、これらのより一般化された m 機械問題を次の2段階手順で(近似的に)解くことができる。

- (1) M_n^m や $W_{n,q}^m$ を最小化する順列 σ^* , σ_q^* を課された制約条件の下で求める。そのためには Sekiguchi [9, 10, 11] の解析的解法を利用する。
- (2) (4)の右辺の値を(1)で求めた各順列に対して計算し、その値の最小な順列を解として採用する。

この手順の所要計算量は、(1)の必要計算量によってほぼ定まる((2)の計算量は $O(n^2)$ 程度である)。(1)の計算量については[10]に簡単な記述がある。(2)で求めた(4)の右辺の最小値が0であるときには、採用した順列が、もとの m 機械問題の最適順列であることが保証される。

初期状態を明言的に扱うことの意義を強調するために、 m 機械のグループ・スケジューリング問題を解く際の留意点を指摘しておく。 $m=2$ の場合のグループ・スケジューリング問題は概略が次のような解法で解かれる[10]。

- (1) 各仕事集合内の仕事の最適順列を定めよ。
- (2) 各仕事集合に対し、(1)で求めたグループ内順序に対応する合成仕事を(4)によって定めよ。
- (3) 求めた合成仕事の最適順列を定め、これをグループ間順序とせよ。

この解法は、(1)で求めるグループ内順序が、

(3)で求められるグループ間順序が何であっても、そのグループ間順序の下で D_n^2 を最小化する最適なグループ内順序である、という性質にもとづいている。このことは、2機械問題では、Johnson 規則あるいはその適当な変種によって定められる順列が、任意の初期状態の下で最適であるということの直接的な帰結である。

同様のアプローチは、 $m \geq 3$ に対しては採用できないことが明らかである。最適順列が初期状態に依存することは2節で例をあげて説明した。 M_n^m , $W_{n,q}^m$ を最小化する順序が最適なグループ内順序となるための十分条件(定理3, 4の条件)は初期状態に依存し、グループ間順序が定まるまで判定不可能である。この点が、初期状態を明言的に記述しない従来の方法では、見逃される危険性が非常に大きいと考える。例えば、各仕事集合 G_i ($i \in N$) が個別に(13)を満たしているとき、 G_i の最適グループ内順序が Johnson 規則によって求められないかも知れないと疑う手掛りは、初期状態を明言的に記述しない理論の中には何もないのである。

References

- 1) Burns, F. and J. Rooker. Three-Stage Flow-Shop with Recessive Second Stage. *Operations Research*. Vol. 26, No. 1. (1978) pp. 207-208.
- 2) Gilmore, P. C. and R. E. Gemory. Sequencing a One State-Variable Machine: A Solvable Case of the Travelling Salesman Problem. *Operations Research*. Vol. 12, No. 5. (1964). pp. 655-679.
- 3) Johnson, S. M. Optimal Two- and Three-Stage Production Schedules with Setup Times Included. *Naval Research Logistics Quarterly*. Vol. 1, No. 1. (1954). pp. 61-68.
- 4) Monma, C. L. The Two-Machine Maximum Flow Time Problem with Series-Parallel Precedence Constraints: An Algorithm and Extensions. *Operations Research*. Vol. 27, No. 4. (1979, July-August). pp. 792-798.
- 5) Nabeshima, I. and S. Maruyama. Two- and Three-Machine Flow-shop Makespan Scheduling Problems with Additional Times

- Separated from Processing Times. *Journal of the Operations Research Society of Japan*. Vol. 27, No. 4. (December, 1984). pp. 348-367.
- 6) Nakamura, N. and K. Hitomi. Study on Production Scheduling-Optimality Analysis of Group Scheduling in a Multi-Stage Process (in Japanese). *Trans. of Japan Society of Mechanical Engineers*. Vol. 42 (1976). pp. 2964-2975.
- 7) Sekiguchi, Y., S. Koyama and R. Miura. Flow Shop and Analysis of the Schedules (in Japanese). *Trans. of Soc. Instrum. & Control Eng.* Vol. 8, No. 5. (1972). pp. 552-560.
- 8) Sekiguchi, Y. Branch-and-Bound Algorithms. In T. Nagao (ed.). *Inference for Decision Makings* (in Japanese), Part II. Tokyo: Koseishakoseikaku. (November, 1973).
- 9) Sekiguchi, Y. Inter- and Intra-Group-of-Jobs Schedule for Minimizing Makespan in a 2-Machine GT Shop. *Preprints of the 8th IFAC Triennial World Congress, Kyoto*. (1981, August). pp. XIV164-XIV169.
- 10) Sekiguchi, Y. Optimal Schedule in GT-Type Flow-Shop under Series-Parallel Precedence Constraints. *Journal of the Operations Research Society of Japan*. Vol. 26, No. 3. (1983). pp. 226-252.
- 11) Sekiguchi, Y. A Decomposition Theory Based on a Dominance Relation and Composite Jobs. *Discrete Applied Mathematics* (In press).
- 12) Sidney, J. B. A Decomposition Algorithm for Sequencing with General Precedence Constraints. *Mathematics of Operations Research*. Vol. 6, No. 2. (May, 1981). pp. 190-204.
- 13) Szwarc, W. Flow Shop Problems with Time Lags. *Management Science*. Vol. 29, No. 4. (1983). pp. 477-481.