

ある種の順序づけ問題の一般解法[†]

鍋 島 一 郎*

§1. は し が き

完成時刻のみならず、納期を考慮した罰金つきの場合など、種々の目的関数を最小ならしめる順序づけの問題については、1台の機械の場合 Smith [1] により解が与えられている。

この論文では、 m 台の機械の場合に対して、ある仮定の下に種々の目的関数を最小ならしめる最適仕事順を決定するための、一つの統一的な一般解法（アルゴリズム）を述べ、具体例への応用についてふれる。

§2. 問 題 の 型

次の各事柄を仮定する。

1. n 個の仕事の各々を m 台の機械 $M_1, M_2, \dots, M_m$ でこの機械順に加工する。
2. 各機械における仕事順は同一である。
3. 各機械は当面する一つの仕事のみを、それが全部完成するまで加工し次の仕事に移る。
4. 工程中、仕掛在庫をゆるす。
5. 仕事間の先行関係はあっても、全然なくともよいが、先行関係のあるときは、その先行関係を満たすように以下の一般解法を用いればよいので、一応、先行関係はないものとする。
6. 各仕事の各機械での加工時間は所与で、仕事の順位に関せず一定とする。
7. 各仕事の各機械での段取時間については、§3 では、段取時間を加工時間に含めて考える場合について、§4 では、それを加工時間とは別に考え、その前の仕事如何により変化すると考える場合について、以下順次考察する。
8. 最小ならしめるべき目的関数 f は、仕事順を変えることなく各仕事の完成時刻を遅らせると、その値は大きくなるものとする。

以上の仮定の下に、問題は、 n 個の仕事全部を m 台の機械で加工したとき、種々の目的関数を最小ならしめる仕事順（最適順序）を定めるアルゴリズムを構成することである。

§3. 加工時間が段取時間を含む場合

3.1. 相隣る2つの仕事の順序

[†] 1966年10月25日受理

* 電気通信大学

仕事 i の機械 M_k における加工時間を $m_{k,i} (i=1, \dots, n, k=1, \dots, m)$ とおく.

$(n-3)$ 個以下の先行仕事の列 S の加工が終ったとし, S の最後の仕事を l とする.

$T_p(l)$ = 仕事 l の M_1 での加工完了後, M_p での加工完了までの経過時間 ($p=2, \dots, m$), とおく.

そこで, いま先行仕事列 S の次に一対の仕事の組 (i, j) を加工するとき, S の次に, ij の順に加工するときと, ji の順に加工するときとで, S, i, j 以後同じ仕事を加工する場合どちらの方が早く以後の仕事の加工を完了せしめるかを調べると, 次のようになる.

先ず Sij の順に加工するとき, $T_p(l)$ と同様に,

$T_p(i)$ = 仕事 i の M_1 での加工完了後, M_p での加工完了までの経過時間 ($p=2, \dots, m$),

$T_p(ij)$ = 仕事 j の M_1 での加工完了後, M_p での加工完了までの経過時間 ($p=2, \dots, m$), とおく.

Sji の順に加工するとき, 上と同様にそれぞれ $T_p(j)$, $T_p(ji)$ ($p=2, \dots, m$) を定める. (図 1)

このとき, 次の補助定理 1 の成立することは $T_p(ij)$, $T_p(ji)$ ($p=2, \dots, m$) の定義より明らかである. 以下, S は $(n-3)$ 個以下の仕事の列とする.

補助定理 1. 既知の先行仕事列 S の次の 2 つの仕事の組 (i, j) に対し, $(m-1)$ 個の不等式:

$$(3.1) \quad T_p(ij) \leq T_p(ji), \quad (p=2, \dots, m)$$

が成立すれば, S, i, j 以後同じ仕事を加工するとき, 仕事順 Sij の後の方が仕事順 Sji の後より, 以後の各仕事の完成時刻が早くなる.

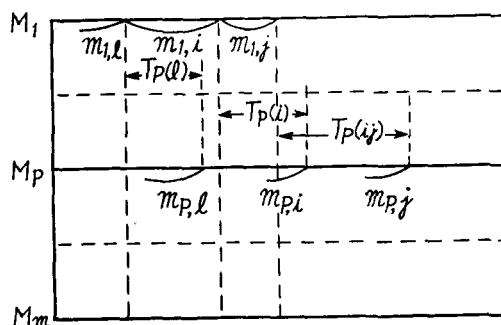


図1. $T_p(l), T_p(i), T_p(ij)$. ($p=2, \dots, m$)

仕事列 Sij , Sji の完成時刻については上記補助定理の如くであるが, 他方, §1 の仮定 8 を満たす目的関数 f の, 仕事列 Sij , Sji のときの値を, それぞれ $f(Sij)$, $f(Sji)$ とおくと, f, S および i と j の順序に依存する関数 g が存在し, ij の順のときの値を $g(i, j)$, ji の順のときの値を $g(j, i)$ とするとき,

$$(3.2) \quad g(i, j) \leq g(j, i)$$

のとき,

$$(3.3) \quad f(Sij) \leq f(Sji)$$

となるものとする.

補助定理 1 と目的関数 f についての仮定より次の定理をうる。

定理 1. 既知の仕事列 S の次の 2 つの仕事の組 (i, j) に対し, S, i, j 以後同じ仕事を加工するとき, 全仕事の加工が完了した時の目的関数の値を, より小さくするための i と j の順序は次の条件により定まる:

$$(3.2) \quad g(i, j) \leq g(j, i)$$

が成立し, さらにすべての $p(p=2, \dots, m)$ に対し,

$$(3.1) \quad T_p(ij) \leq T_p(ji)$$

が成立すれば, S の次は, ij の順とする。

もし, (3.2) が成立するとき (3.1) が成立しなければ, ij の順と ji の順の両方を考える。

3.2. 優先仕事列

k 個の同じ仕事群, または k 個の仕事群より成る 2 つの先行仕事列 S_k, S'_k において, S'_k 以後最適順序を生ずるならば, S_k 以後も必ず最適順序を生ずる場合, 仕事列 S_k は仕事列 S'_k に優先すると呼ぶこととする。例えば §3.1 定理 1 において, (3.2), (3.1) 共に成立するとき, 仕事列 S_{ij} は仕事列 S_{ji} に優先していることになる。

そこで, 少なくとも 1 つの最適順序を求めるとき, 同一個数の仕事群より成る 2 つの先行仕事列の間に優先関係が存在すれば, 他方に優先する仕事列以後の仕事順のみを考えればよい。

§3.1 の補助定理 1 と目的関数 f についての仮定より次の定理が成立する。

定理 2. $2q$ 個の同じ仕事より成る 2 つの先行仕事列 $(i_1 i_2 \dots i_{2q-1} i_{2q}), (j_1 j_2 \dots j_{2q-1} j_{2q})$ に対し,

$$(3.4) \quad f(i_1 i_2 \dots i_{2q-1} i_{2q}) \leq f(j_1 j_2 \dots j_{2q-1} j_{2q})$$

および, $(m-1)$ 個の不等式:

$$(3.5) \quad T_p(i_{2q-1} i_{2q}) \leq T_p(j_{2q-1} j_{2q}), \quad (p=2, \dots, m)$$

が成立すれば, 少なくとも 1 つの最適順序を求めるとき, 仕事列 $(i_1 i_2 \dots i_{2q-1} i_{2q})$ が優先する。

次に, 後述 §3.4 のアルゴリズムの最終段階で用いてもよい事柄として次のことが考えられる:
すなわち,

〔下界による判定〕

(i) 仕事の個数 n が偶数, $n=2r+2$ のとき,

n 個の仕事列のおのおの $(i_1 i_2 \dots i_{2r} i_{2r+1} i_{2r+2})$ に対して, 部分列 $(i_1 i_2 \dots i_{2r})$ および i_{2r+1}, i_{2r+2} の関数である下界 LB が,

$$(3.6) \quad f(i_1 i_2 \dots i_{2r} i_{2r+1} i_{2r+2}) \geq LB(i_1 i_2 \dots i_{2r}, i_{2r+1}, i_{2r+2})$$

と求まる場合, この LB が既に求めた他の n 個の仕事列に対する f の値より大きいならば, 当仕事列 $(i_1 \dots i_{2r+2})$ に対する f の値を求める必要はない。同様のことが,

(ii). n が奇数, $n=2r+1$ のときに対しても成り立つ。(i) の各仕事列, 式において i_{2r+2}

を除けばよい。

3.3 $T_p(i), T_p(ij); T_p(j), T_p(ji)$ に関する繰返し関係と (3.1) 成立のための十分条件

前節 (3.1) における $T_p(ij), T_p(ji)$ ($p=2, \dots, m$) の逐次計算には次の繰返し関係を用いる。(図2)

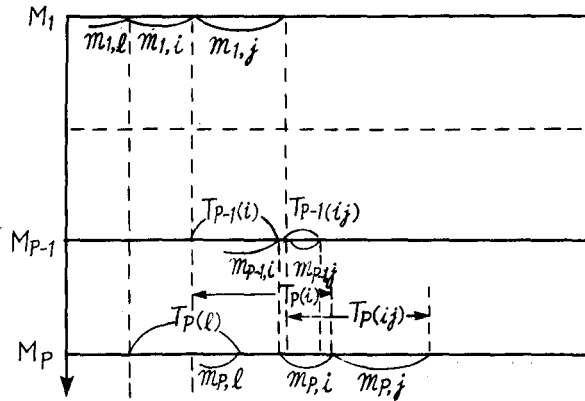


図 2. 繰返し関係

仕事列 Sij のとき、例えば j の M_p での加工開始は、 M_{p-1} で j の加工が完了し、 M_p で j の前の仕事 i の加工が完了した後であることから、仕事 i の前の仕事は l であるから、次式が得られる。

$$(3.7) \quad T_p(i) = m_{p,i} + \max [T_p(l) - m_{1,i}, T_{p-1}(i)]$$

$$(3.8) \quad T_p(ij) = m_{p,j} + \max [T_p(i) - m_{1,j}, T_{p-1}(ij)] \quad (p=2, \dots, m)$$

ただし、 $T_1(i) \equiv 0, T_1(ij) \equiv 0$ である。

仕事列 Sji のとき、同様に、

$$(3.7)' \quad T_p(j) = m_{p,j} + \max [T_p(l) - m_{1,j}, T_{p-1}(j)]$$

$$(3.8)' \quad T_p(ji) = m_{p,i} + \max [T_p(j) - m_{1,i}, T_{p-1}(ji)] \quad (p=2, \dots, m)$$

となる。ただし、 $T_1(j) \equiv 0, T_1(ji) \equiv 0$ である。

(3.7) と (3.7)' は同じ式であり、上記4式は全く類似の演算である。

次に、(3.1) : すべての p ($p=2, \dots, m$) に対して、 $T_p(ij) \leq T_p(ji)$ が成立するたきの十分条件が、逐次 $p=2, 3, \dots, p-1$ に対し、それぞれ (3.7) を (3.8) に、(3.7)' を (3.8)' に代入して、 $T_p(ij), T_p(ji)$ を $T_k(l)$ ($k=2, \dots, p$) で表わし、各 $T_k(l)$ に対応する項を比較することにより、帰納的に次の定理が得られる。(詳細な証明は省略する。)

定理 3. 相隣る2つの仕事の組 (i, j) に対して、次の $\frac{1}{2}m(m-1)$ 個の不等式：

$$(3.9) \quad \text{Min}[m_{k,i}, m_{k+1,j}] \leq \text{Min}[m_{k,j}, m_{k+1,i}], \quad (k=1, \dots, m-1)$$

および,

$$(3.10) \quad \text{Min} \left[\sum_{k=u}^v m_{k,i}, \sum_{k=u+1}^{v+1} m_{k,j} \right] \leq \text{Min} \left[\sum_{k=u}^v m_{k,j}, \sum_{k=u+1}^{v+1} m_{k,i} \right]$$

$$(u < v; u=1, \dots, m-2; v=2, \dots, m-1)$$

が成立すれば, i, j の前の仕事 l の如何にかかわらず常に $T_p(ij) \leq T_p(ji)$ ($p=2, \dots, m$) となる.

すなわち, 組 (i, j) の位置にかかわらず常に仕事順 ij は仕事順 ji に時間的に優先する.

定理 3 の仮定が成立しないときは, 繰返し関係を用いて, (3.1) を確かめる.

3.4 一般解法 (アルゴリズム)

前節の結果を用い, 次の段階により最適順序を求める. $n=2r+2$, または $n=2r+1$.

(i) 最初の 2 つの仕事列群の決定

nC_2 個の組合せ (i, j) のおのおのに対して, $T_p(l) \equiv 0$, $T_p(i) = \sum_{k=2}^p m_{k,i}$, $T_p(j) = \sum_{k=2}^p m_{k,j}$

($p=2, \dots, m$) の下に, 定理 1 の (3.2) および (3.1) が成立するとき ij の順とし, (3.2) は成立するが (3.1) が成立しないとき, ij の順と ji の順の両方をとる. かくして 2 個の仕事の部分列 $(i_1 i_2)$ の集合をうる.

(ii) 3 番目と 4 番目の仕事群の決定

(i) で求めた各部分列 $(i_1 i_2)$ に対し, 残りの仕事よりの $n-2C_2$ 個の組合せ (i, j) の各々に対し, $T_p(l) \equiv T_p(i_1 i_2)$ ($p=2, \dots, m$) [$T_p(i_1 i_2)$ は (i) で既知] の下に, (i) におけると同様にして i と j の順を定める. かくして 4 個の仕事の部分列 $(i_1 i_2 i_3 i_4)$ の集合をうる.

[定理 2 の適用] この段階では定理 2 を適用して以後必要でない部分列 $(i_1 i_2 i_3 i_4)$ を消す.

(iii) $(2r-1)$ 番目と $2r$ 番目迄の仕事群の決定

(ii) と同様にして次々に 2 つの仕事をつないでゆき, $(r-1)$ 番目と $2r$ 番目の仕事まで定め, $2r$ 個の仕事の部分列 $(i_1 i_2 \dots i_{2r-1} i_{2r})$ の集合をうる. 各段階で定理 2 を適用する.

(iv) 最後の 2 個, または 1 個の仕事の決定

(iii) で求めた各部分列 $(i_1 i_2 \dots i_{2r-1} i_{2r})$ に対し, $T_p(l) \equiv T_p(i_{2r-1} i_{2r})$ ($p=2, \dots, m$) [前段で既知] の下に, $n=2r+2$ のときは, 残りの 2 つの仕事 (i, j) に対し定理 1 の (3.2) が成立するとき列 $(i_1 i_2 \dots i_{2r-1} i_{2r} ij)$ をとる. $n=2r+1$ のときは, 残りの 1 つの仕事 i を加えた列 $(i_1 i_2 \dots i_{2r-1} i_{2r} i)$ をとる.

かくして, n 個の仕事列 $(i_1 i_2 \dots i_n)$ の集合をうる.

[下界による判定] この段階において §3.2 における下界が求まれば, この下界と既に得た $f(i_1 i_2 \dots i_n)$ の大小により最適順序を生じうる $(i_1 i_2 \dots i_{2r})$ についてのみ最後の 2 個, または

1 個の仕事の決定を行なえばよい。

(v) 最適順序の決定

(iv) により求めた順序 $(i_1 i_2 \cdots i_{2r} i_{2r+1} i_{2r+2})$, または $(i_1 i_3 \cdots i_{2r} i_{2r+1})$ の集合に対する、最小の

$$(3.9) \quad f(i_1 i_2 \cdots i_{2r+1} i_{2r+2})$$

または,

$$(3.9)' \quad f(i_1 i_2 \cdots i_{2r+1})$$

を与える順序が最適順序で、この最小値が目的関数 f の最小値である。(一般解法終)

なお、(3.9) または (3.9)' は定理 1 の関数 g を用いて与えられることもあり、(iv) の段階において同時に (3.9), または (3.9)' の最小値を求めうる場合もある。(3.5 応用例参照)

また、§2 の仮定 5 で述べた如く、仕事間に先行関係のある場合には、それを満足するように (i)~(iv) の各段階を適用すればよい。

さらに、もし新しい仕事追加された場合には、これらともの仕事との組合せについて (i)~(v) の計算を補えばよい。

3.5. 具体例 [2]

問題の仮定を満足するいくつかの例につき、定理 1 の (3.2) における $g(i, j)$ [i と j を交換すれば $g(j, i)$ をうる.] と (3.2), 定理 2 の (3.4), および一般解法の (v) の (3.9) (3.9)' 等を示すにとどめる。

1. 総所要時間最小

$g(i, j) \equiv T_m(ij)$ で、(3.2) は (3.1) に含まれる。

(3.4) は、 $T_m(i_{2q-1} i_{2q}) \leq T_m(j_{2q-1} j_{2q})$ となり (3.5) に含まれる。(3.9), (3.9)' はそれぞれ $T_m(i_{2r+1} i_{2r+2})$, $T_m(i_{2r+1})$ としてよく、総所要時間は、それぞれ、

$$\sum_{i=1}^n m_{1,i} + T_m(i_{2r+1} i_{2r+2}), \quad \sum_{i=1}^n m_{1,i} + T_m(i_{2r+1})$$

である。〔Time Lag のある場合も可能〕

2. 各仕事の完成時刻の重みつき平均〔和〕を最小

各仕事 i の完成時刻 T_i のとき、 $\sum_{i=1}^n a_i T_i$ を最小にする。($a_i \geq 0$)

$$(3.2) \text{ は, } a_i T_m(i) + a_j \{m_{1,i} + T_m(ij)\} \leq a_j T_m(j) + a_i \{m_{1,j} + T_m(ji)\}$$

$$(3.4) \text{ は, } \sum_{k=1}^{2q} a_{i_k} T_m(i_k) + \sum_{k=1}^{2q-1} (m_{1,i_k} \cdot \sum_{t=k+1}^{2q} a_{i_t}) \leq \sum_{k=1}^{2q} a_{j_k} T_m(j_k) + \sum_{k=1}^{2q-1} (m_{1,j_k} \cdot \sum_{t=k+1}^{2q} a_{j_t})$$

(3.9), (3.9)' はそれぞれ (3.4) の左辺で $2q=2r+2$, $2q=2r+1$ とおいてうる式を考えればよい。

2'. 各仕事の完成時刻の平均〔和〕を最小

2. で各重み $\alpha_i \equiv 1$ ($i=1, \dots, n$) とおけばよい.

$g(i, j) \equiv 2 \sum_{k \in S} m_{1,k} + \{m_{1,i} + T_m(i)\} + \{m_{1,i} + m_{1,j} + T_m(j)\}$ より, (3.2) は,

$$m_{1,i} + T_m(i) + T_m(j) \leq m_{1,j} + T_m(j) + T_m(ji)$$

(3.4) 式は, $T_m(ji) \equiv T_m(j)$ とおくと,

$$\sum_{k=1}^{2q-1} (2q-k) m_{1,i_k} + \sum_{k=1}^{2q} T_m(i_k) \leq \sum_{k=1}^{2q-1} (2q-k) m_{1,j_k} + \sum_{k=1}^{2q} T_m(j_k)$$

(3.9), (3.9)' はそれぞれ,

$$\sum_{k=1}^{2r+1} (2r+2-k) m_{1,i_k} + \sum_{k=1}^{2r+2} T_m(i_k), \quad \sum_{k=1}^{2r} (2r+1-k) m_{1,i_k} + \sum_{k=1}^{2r+1} T_m(i_k)$$

である.

3. 各仕事の完成時刻の増加関数である費用の平均 (和) を最小

仕事 i の完成時刻 T_i の増加関数である費用を $C_i(T_i)$ とおく.

$$(3.2) \text{ は, } C_i(\sum_{k \in S} m_{1,k} + m_{1,i} + T_m(i)) + C_j(\sum_{k \in S} m_{1,k} + m_{1,i} + m_{1,j} + T_m(j)) \\ \leq C_j(\sum_{k \in S} m_{1,k} + m_{1,j} + T_m(j)) + C_i(\sum_{k \in S} m_{1,k} + m_{1,j} + m_{1,i} + T_m(ji)).$$

$$(3.4) \text{ は, } \sum_{l=1}^{2q} C_{i_l} (\sum_{k=1}^l m_{1,i_k} + T_m(i_l)) \leq \sum_{l=1}^{2q} C_{j_l} (\sum_{k=1}^l m_{1,j_k} + T_m(j_l))$$

(3.9), (3.9)' は, 上式の左辺で, それぞれ $2q=2r+2$, $2q=2r+1$ とおいてうる式である.

4. 各仕事の納期よりの遅れの最大値を最小

仕事 i の完成時刻を T_i , 納期を D_i , 最大の遅れを $\text{Max}_i [T_i - D_i, 0]$ とする. ($\text{Max}_i [T_i - D_i, 0]$ としてもよい.)

$$g(i, j) = \text{Max} \left[\sum_{k \in S} m_{1,k} + m_{1,i} + T_m(i) - D_i, \sum_{k \in S} m_{1,k} + m_{1,i} + m_{1,j} \right. \\ \left. + T_m(j) - D_j, 0 \right]$$

で, (3.2) は, $g(i, j) \leq g(j, i)$

$$(3.4) \text{ 式は, } \text{Max}_{k=1 \sim q} g(i_{2k-1}, i_{2k}) \leq \text{Max}_{k=1 \sim q} g(j_{2k-1}, j_{2k})$$

(3.9), (3.9)' は, それぞれ,

$$\text{Max}_{k=1 \sim r+1} g(i_{2k-1}, i_{2k}),$$

$$\text{Max} \left[\text{Max}_{k=1 \sim r} g(i_{2k-1}, i_{2k}), \text{Max} \left[\sum_{k=1}^{2r+1} m_{1,i_k} + T_m(i_{2r+1}) - D_{i_{2r+1}}, 0 \right] \right]$$

である.

4'. 各仕事の納期よりの遅れの増加関数である費用の最大値を最小

費用を $C_i(\text{Max}_i [T_i - D_i, 0])$, または $C_i(T_i - D_i)$ として 4. と同様に考えればよい.

5. 各仕事の納期よりの遅れの平均〔和〕を最小 $\left(\sum_{i=1}^n (T_i - D_i)\right)$ を最小とする.)

$$(3.2) \text{ は, } T_m(i) + m_{1,i} + T_m(ij) \leq T_m(j) + m_{1,j} + T_m(ji)$$

としてよい.

$$(3.4) \text{ 式は, } \sum_{k=1}^{2q} T_m(i_k) + \sum_{k=1}^{2q-1} (2q-k)m_{1,i_k} \leq \sum_{k=1}^{2q} T_m(j_k) + \sum_{k=1}^{2q-1} (2q-k)m_{1,j_k}$$

でよい. また, (3.9), (3.9)' として, それぞれ上式の左辺で, $2q=2r+2$, $2q=2r+1$ として
うる式を考えればよい.

なお, 遅れの和として, $\sum_{i=1}^n \text{Max}[T_i - D_i, 0]$ をとるときは式が複雑となるが, 同様に考えうる.

5'. 各仕事の納期よりの遅れの増加関数である費用の平均 (和) を最小

費用の和を, $\sum_{i=1}^n C_i(T_i - D_i)$, または $\sum_{i=1}^n C_i(\text{Max}[T_i - D_i, 0])$ とする. 例えば前者の場合,

$$(3.2) \text{ 式は, } C_i(\sum_{k \in S} m_{1,k} + m_{1,i} + T_m(i) - D_i) + C_j(\sum_{k \in S} m_{1,k} + m_{1,i} + m_{1,j} + T_m(ij) - D_j) \\ \leq C_j(\sum_{k \in S} m_{1,k} + m_{1,j} + T_m(j) - D_j) + C_i(\sum_{k \in S} m_{1,k} + m_{1,j} + m_{1,i} + T_m(ji) - D_i).$$

(3.4) 式は, $T_m(ij) \equiv T_m(j)$ とおくと,

$$\sum_{k=1}^{2q} C_{i_k}(\sum_{l=1}^k m_{1,i_l} + T_m(i_k) - D_{i_k}) \leq \sum_{k=1}^{2q} C_{j_k}(\sum_{l=1}^k m_{1,j_l} + T_m(j_k) - D_{j_k}).$$

(3.9), (3.9)' は, 上式の左辺で, それぞれ $2q=2r+2$, $2q=2r+1$ とおいてうる式である.

6. 各仕事の各機械間での仕掛在庫時間 (待時間) の総和を最小

仕事 i の M_{k-1} での加工終了後, M_k での加工が始まるまでの待時間を $w_k(i)$ ($i=1, \dots, n, k=2, \dots, m$) とおくと, i についての待時間の和は $W(i) \equiv \sum_{k=2}^m w_k(i) = T_m(i) - \sum_{k=2}^m m_{k,i}$. 同様に, i の
次の仕事 j に対し, $w_k(j) \equiv w_k(ij)$ とし, j の待時間の和は, $W(ij) \equiv \sum_{k=2}^m w_k(ij) = T_m(ij) - \sum_{k=2}^m m_{k,j}$ である.

$$(3.2) \text{ は, } T_m(i) + T_m(ij) \leq T_m(j) + T_m(ji) \text{ となり,}$$

$$(3.4) \text{ は, } \sum_{k=2}^{2q} \{T_m(i_k) - \sum_{l=2}^m m_{l,i_k}\} \leq \sum_{k=2}^{2q} \{T_m(j_k) - \sum_{l=2}^m m_{l,j_k}\}.$$

(3.9), (3.9)' は上式の左辺で, それぞれ $2q=2r+2$, $2q=2r+1$ とおいてうる式である.

§ 4. 加工時間と別に段取時間を考慮する場合

§3 の場合と異なる点は, §3 の加工時間とは別に, 各仕事の各機械における段取時間 (運搬時間, および準備時間) を考え, これはその前の仕事如何によって変化すると考える.

すなわち, 仕事 i の次に仕事 j を機械 M_{k-1} より M_k に運ぶに要する運搬時間を $t_{k,ij}$ ($k=2, \dots, m$) とし, これは M_{k-1} , M_k および仕事順 ij に依存するとする.

また、機械 M_k で仕事 i を加工完了後、次の仕事 j を加工するときの M_k の準備時間を $s_{k,ij}(k=1, \dots, m)$ とする。なお、以下において、 $t_{k,ij} \equiv 0$ ($k=2, \dots, m$)、または $s_{k,ij} \equiv 0$ ($k=1, \dots, m$) と考えてもよい。従って、仕事 j の機械 M_k における加工は、 j が M_{k-1} より到着し、かつ M_k の j を加工する準備が完了した後に始まる。

4.1 2つの仕事の間にある2つの仕事の順序

($n-4$) 個以下の先行仕事の列 S の加工が終わったとし、 S の最後の仕事を l とする。

$T_p(l)$ = 仕事 l の M_1 での加工完了後、 M_p での加工完了までの経過時間 ($p=2, \dots, m$)、とおく。

そこで、いま先行仕事列 S の次に一対の仕事の組 (i, j) を加工し、これらの次に仕事 d を加工するとき、 S の次、 ij の順に加工するときと、 ji の順に加工するときとで、 S, i, j, d 以後同じ仕事を加工する場合、どちらの方が早く以後の仕事の加工を完了せしめるかを調べると次のようになる。

まず、 $Sijd$ の順に加工するとき、

$E_p(i) \equiv E_p(li) = S$ の最後の仕事 l の M_1 での加工完了後、仕事 i を M_p で加工完了するまでの経過時間 ($p=1, \dots, m$),

$E_p(ij) \equiv E_p(lij) =$ 仕事 l の M_1 での加工完了後、 ij の順に M_p で加工し終るまでの経過時間 ($p=1, \dots, m$),

$E_p(ijd) \equiv E_p(lijd) =$ 仕事 l の M_1 での加工完了後、 ijd の順に M_p で加工し終るまでの経過時間 ($p=1, \dots, m$),

とおく。

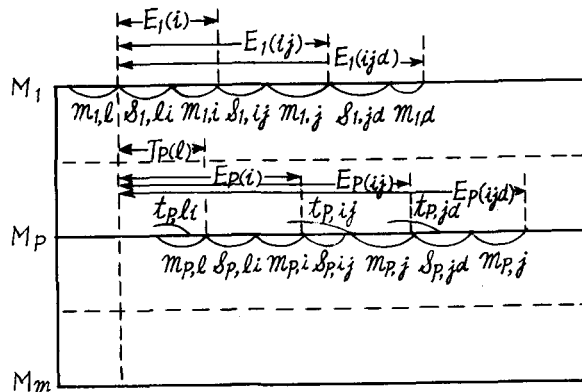


図3. $T_p(l); E_p(i), E_p(ij), E_p(ijd)$. ($p=1, \dots, m$)

$Sjid$ の順に加工するとき、上と同様にそれぞれ $E_p(j)$, $E_p(ji)$, $E_p(jid)$ を定める。(図3) このとき、次の補助定理2の成立することは $E_p(ijd)$, $E_p(jid)$ の定義より明らかである。

補助定理 2. 既知の先行仕事列 S の次の3つの仕事 i, j, d に対して、 m 個の不等式

$$(4.1) \quad E_p(ijd) \leq E_p(jid), \quad (p=1, \dots, m)$$

が成立すれば、 S, i, j, d 以後同じ仕事を加工するとき、仕事順 $Si jd$ の後の方が仕事順 $Sjid$ の後より、以後の仕事の完成時刻が早くなる。

次に §3 の場合と同様に、目的関数 f の仕事列 $Si jd$, $Sjid$ のときの値を、それぞれ $f(Si jd)$, $f(Sjid)$ とおくと、 f, S, i と j の順序、および d に依存する関数 g が存在し、それぞれのときの値を、 $g(ijd)$, $g(jid)$ とおけば、

$$(4.2) \quad g(ijd) \leq g(jid)$$

のとき、

$$(4.3) \quad f(Si jd) \leq f(Sjid)$$

となるものとする。

補助定理 2 と目的関数 f についての仮定より次の定理をうる。

定理 4. 既知の仕事列 S の次の 3 つの仕事 i, j, d に対し、 S, i, j, d 以後同じ仕事を加工するとき、全仕事の加工が完了した時の目的関数の値を、より小さくするための i と j の順序は次の条件により定まる。

(i) S が $(n-4)$ 個以下の仕事より成るとき、

$$(4.2) \quad g(ijd) \leq g(jid)$$

が成立し、さらにすべての $p(p=1, \dots, m)$ に対し、

$$(4.1) \quad E_p(ijd) \leq E_p(jid)$$

が成立すれば、 S の次は ijd の順とする。

もし、(4.2) が成立し、(4.1) が成立しなければ、 ijd の順と jid の順の両方を考える。

(ii) S が $(n-3)$ 個の仕事より成るとき、

$$(4.2) \quad g(ijd) \leq g(jid)$$

が成立するとき、 ijd の順とする。

(iii) S が $(n-2)$ 個の仕事より成るとき、

$$(4.4) \quad f(Sij) \leq f(Sji)$$

が成立するとき、 ij の順とする。

4.2. 優先仕事列

§3.2 と全く同様に、同一個数の仕事より成る 2 つの先行仕事列の間の優先関係を定める。

例えば定理 4 において、(i) の場合、(4.2), (4.1) 共に成立するならば、仕事列 $Si jd$ は仕事列 $Sjid$ に優先することになる。

そこで、少なくとも 1 つの最適順序を求めるとき、同一個数の仕事より成る 2 つの先行仕事列の間に優先関係が存在すれば、他方に優先する仕事列以後の仕事順のみを考えればよいことになるので、補助定理 2 と目的関数 f についての仮定より次の定理が成立する。

定理 5. $3q$ 個の同じ仕事より成る 2 つの先行仕事列 $(i_1 i_2 \dots i_{3q-3} i_{3q-2} i_{3q-1} i_{3q}), (j_1 j_2$

$\cdots j_{3q-3} j_{3q-2} j_{3q-1} j_{3q}$; $i_{3q} \equiv j_{3q}$, に対して,

$$(4.5) \quad f(i_1 i_2 \cdots i_{3q}) \leq f(j_1 j_2 \cdots j_{3q})$$

で, かつ, m 個の不等式

$$(4.6) \quad T_1(i_1 i_2 \cdots i_{3q-3}) + E_p(i_{3q-2} i_{3q-1} i_{3q}) \\ \leq T_1(j_1 j_2 \cdots j_{3q-3}) + E_p(j_{3q-2} j_{3q-1} j_{3q}), \quad (p=1, \dots, m)$$

ここに, $T_1(S)$ は仕事列 S の M_1 での完成時刻を表わす; が成立すれば, 少なくとも 1 つの最適順序を求めるとき, 仕事列 $(i_1 i_2 \cdots i_{3q})$ が他方に優先する.

次に, 後述 §4.4 のアルゴリズムの最終段階で, 優先する全仕事列を発見するための判定条件として, 次の下界による判定が考えられる. すなわち,

〔下界による判定〕

(i) 仕事の個数 $n=3r+3$ のとき, n 個の仕事列の各々 $(i_1 i_2 i_3 \cdots i_{3r} i_{3r+1} i_{3r+2} i_{3r+3})$ に対し, 部分列 $(i_1 i_2 i_3 \cdots i_{3r})$, および i_{3r+1} , i_{3r+2} , i_{3r+3} の関数である下界 LB が,

$$(4.7) \quad f(i_1 i_2 i_3 \cdots i_{3r+3}) \geq LB(i_1 i_2 \cdots i_{3r}; i_{3r+1}, i_{3r+2}, i_{3r+3})$$

と求まる場合, この LB が既に求めた他の n 個の仕事列に対する f の値より大きいならば, 当仕事列はその仕事列によって優先されるので, 当仕事列に対する f の値を求める必要はない.

(ii) $n=3r+2$ のとき, (i) の仕事列, 式等で i_{3r+3} を除いて考えれば全く (i) と同様である.

(iii) $n=3r+1$ のとき, (i) の仕事列, 式等で i_{3r+2} と i_{3r+3} を除いて考えれば全く (i) と同様である.

4.3. $E_p(i)$, $E_p(ij)$, $E_p(ijd)$; $E_p(j)$, $E_p(ji)$, $E_p(jid)$ に関する繰返し関係

既に述べた如く, 仕事 i の次の仕事 j について考えると, 仕事 j の機械 M_p での加工開始

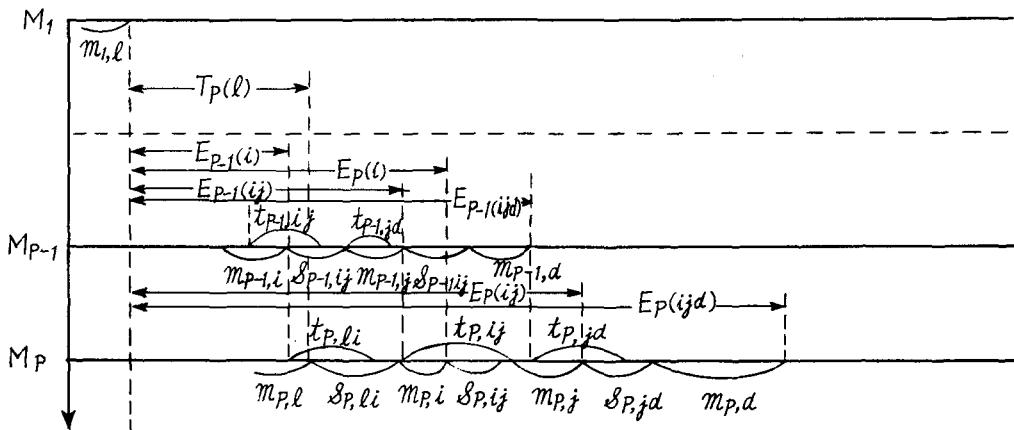


図 4. 繰返し関係

は, M_{p-1} で j の加工が終了し M_p に到着し, かつ, M_p での i の加工が終了し j を加工する準備が完了した後であるから, 列 $Sijd$ に対する, (4.1) の $E_p(ijd)$ ($p=1, \dots, m$) の計算に, 次の $E_p(i)$, $E_p(ij)$, $E_p(ijd)$, ($p=1, \dots, m$) の逐次計算のための繰返し関係を用いる.

(図4)

$$(4.8) \quad \begin{cases} E_1(i) = s_{1,li} + m_{1,i} \\ E_p(i) = m_{p,i} + M_{ax} [T_p(l) + s_{p,li}, E_{p-1}(i) + t_{p,li}], \quad (p=2, \dots, m) \end{cases}$$

$$(4.9) \quad \begin{cases} E_1(ij) = s_{1,li} + m_{1,i} + s_{1,ij} + m_{1,j}, \\ E_p(ij) = m_{p,j} + M_{ax} [E_p(i) + s_{p,ij}, E_{p-1}(ij) + t_{p,ij}], \quad (p=2, \dots, m) \end{cases}$$

$$(4.10) \quad \begin{cases} E_1(ijd) = s_{1,li} + m_{1,i} + s_{1,ij} + m_{1,j} + s_{1,jd} + m_{1,d}, \\ E_p(ijd) = m_{p,d} + M_{ax} [E_p(ij) + s_{p,jd}, E_{p-1}(ijd) + t_{p,jd}], \quad (p=2, \dots, m) \end{cases}$$

同様に, 列 $Sjid$ に対する, (4.1) の $E_p(jid)$ ($p=1, \dots, m$) の計算に用いる, $E_p(j)$, $E_p(ji)$, $E_p(jid)$ ($p=1, \dots, m$) の逐次計算のための繰返し関係は, それぞれ, (4.8), (4.9), (4.10)において i と j を交換してうる式 [(4.8)', (4.9)', (4.10)'] とおく.] である.

上記, (4.8) [(4.8)'], (4.9), (4.9)', (4.10), (4.10)' は全く類似の計算である.

4.4. 一般解法 (アルゴリズム)

前節の結果を用い, 次の段階により最適順序を求める. $n=3r+3$, $3r+2$, または $3r+1$.

(i) 最初の 3 つの仕事列群の決定

${}_nC_2$ 個の組合せ (i, j) の各々に対し ij と ji の 2 つの順序を考え, これらに残りの $(n-2)$ 個の仕事のおのおの d をつけて, ijd の順と jid の順に対して定理 4 の (4.2) が成立するとき, (4.1) が成立すれば ijd の順とし, (4.1) が成立しなければ ijd の順と jid の順の両方を考える. この場合, (4.1) の両辺の計算に §4.3 の繰返し関係を用いるが, $T_p(l) \equiv 0$ ($p=2, \dots, m$), $E_1(i) = s_{1,li} + m_{1,i}$, $E_p(i) = m_{p,i} + \text{Max} [s_{p,li}, E_{p-1}(i) + t_{p,li}]$, ($p=2, \dots, m$) [$E_1(j)$, $E_p(j)$ ($p=2, \dots, m$) は上式で i を j とおく.] で, 例えば $s_{p,li}$ は i の M_p における準備時間, $t_{p,li}$ は i を M_{p-1} より M_p に運ぶに要する時間を意味する. かくして, 3 個の仕事の部分列 $(i_1 i_2 i_3)$ の集合をうる.

(ii) 4, 5, 6 番目の仕事群の決定

(i) で求めた各部分列 $(i_1 i_2 i_3)$ のおのおのに対し, 残りの $(n-3)$ 個よりの ${}_{n-3}C_2$ 個の組合せ (i, j) の各々に対し残りの $(n-5)$ 個の仕事のおのおの d を i と j の後につけて, 各 $(i_1 i_2 i_3)$ の次の 3 つの仕事列として ijd の順と jid の順のうち, 他に優先する順序を定理 4 により (i) と同様に定める. この場合, $T_p(l) \equiv E_p(i_1 i_2 i_3) - E_1(i_1 i_2 i_3)$ ($p=2, \dots, m$) [(i) により既知] である. また定理 5 を適用して以後必要でない部分列を消し, 6 個の仕事の部分列 $(i_1 i_2 i_3 i_4 i_5 i_6)$ の集合をうる.

(iii) $3r-2, 3r-1, 3r$ 番目までの仕事群の決定.

(ii) と同様にして、次々に 3 つの仕事の列をつないでゆき、最初より $3r$ 番目までの仕事列 $(i_1 i_2 i_3 \cdots i_{3r-2} i_{3r-1} i_{3r})$ の集合をうる.

(iv) 最後の 3 個, 2 個, または 1 個の仕事の決定.

(iii) で求めた各部分列 $(i_1 i_2 i_3 \cdots i_{3r-2} i_{3r-1} i_{3r})$ に対し, $T_p(I) \equiv E_p(i_{3r-2} i_{3r-1} i_{3r}) - E_1(i_{3r-2} i_{3r-1} i_{3r})$ ($p=2, \dots, m$) (前段より既知) とし, $n=3r+3$ のときは, 残りの 3 つの仕事よりの 2 個の組合せ (i, j) のおのおのに最後の 1 つの仕事 d を定め, 定理 4 の (4.2) が成立するとき ijd の順とする. $n=3r+2$ のときは, 残りの 2 つの仕事 (i, j) に対し, 定理 4 の (4.4) が成立するとき ij の順とする. $n=3r+1$ のときは, 残りの 1 つの仕事を最後につける. かくして, 上の各場合に n 個の仕事列の集合をうる.

なお, この段階で, §4.2 の下界による判定が適用できれば, 下界を用いて, 最適順序を生じうる部分列 $(i_1 \cdots i_{3r})$ についてのみ最後の仕事の順序を定めればよい.

(v) 最適順序の決定

(iv) で求めた各場合における順序 $(i_1 i_2 \cdots i_n)$ のうち, 最小の f を与える順序が最適順序でその最小値が目的関数 f の最小値である. (終)

なお, §3.4 の一般解法の次に述べた事はこの場合にもあてはまる.

4.5. 具体例

§4.4 のアルゴリズムは §3.5 の具体例の各場合に応用できる. 次に, その 2 つの場合につき, 定理 4 の (4.2) : $g(ijd) \leq g(jid)$, 定理 5 の (4.5) : $f(i_1 i_2 \cdots i_{3q}) \leq f(j_1 j_2 \cdots j_{3q})$ [$3q$ 個の同じ仕事より成る 2 つの列 $(i_1 i_2 \cdots i_{3q})$, $(j_1 j_2 \cdots j_{3q})$ で最後の仕事と同じ : $i_{3q} \equiv j_{3q}$] 等を示すにとどめる.

1. 各仕事間の段取時間を考慮するとき総所要時間を最小.

(4.2) 式で, $g(ijd) \equiv E_m(ijd)$, $g(jid) \equiv E_m(jid)$ となり, (4.2) 式は (4.1) 式に含まれる.

(4.5) は (4.6) に含まれ, (4.6) は,

$$\begin{aligned} & \sum_{k=1}^{3q-3} (s_{1, i_{k-1} i_k} + m_{1, i_k}) + E_p(i_{3q-2} i_{3q-1} i_{3q}) \\ & \leq \sum_{k=1}^{3q-3} (s_{1, i_{k-1} i_k} + m_{1, j_k}) + E_p(j_{3q-2} j_{3q-1} j_{3q}), \quad (p=1, \dots, m) \end{aligned}$$

となる. ここに, 例えば i_{k-1} は i_k の前の仕事を示し, $i_{3q} \equiv j_{3q}$ である.

また, アルゴリズムの段階 (iv) で用いる下界の式 (4.7) 等は, $n=3r+3$, $n=3r+2$, $n=3r+1$ のとき, それぞれ,

$$E_m(i_{3r+1} i_{3r+2} i_{3r+3}) \geq E_m(i_{3r+1} i_{3r+2}) + s_{m, i_{3r+2} i_{3r+3}} + m_{m, i_{3r+3}}$$

$$E_m(i_{3r+1} i_{3r+2}) \geq E_m(i_{3r+1}) + s_{m, i_{3r+1} i_{3r+2}} + m_{m, i_{3r+2}}$$

$$E_m(i_{3r+1}) \geq E_m(i_{3r-2} i_{3r-1} i_{3r}) - E_1(i_{3r-2} i_{3r-1} i_{3r}) + s_{m, i_{3r} i_{3r+1}} + m_{m, i_{3r+1}}$$

となる。また段階 (v) における $f(i_1 i_2 \cdots i_n)$ の値は、それぞれ、

$$\begin{aligned} f(i_1 i_2 i_3 \cdots i_{3r+3}) &= \sum_{k=1}^{3r} (s_{1, i_{k-1} i_k} + m_{1, i_k}) + E_m(i_{3r+1} i_{3r+2} i_{3r+3}) \\ &\equiv T_{3r} + E_m(i_{3r+1} i_{3r+2} i_{3r+3}), \\ f(i_1 i_2 i_3 \cdots i_{3r+2}) &= T_{3r} + E_m(i_{3r+1} i_{3r+2}) \\ f(i_1 i_2 i_3 \cdots i_{3r+1}) &= T_{3r} + E_m(i_{3r+1}) \end{aligned}$$

である。

2. 巡回セールスマン問題

上の例1において1台の機械 ($m=1$) の場合、 $\sum_{i=1}^n m_{1, i}$ が一定ゆえ、 $\sum_{k=2}^n s_{1, i_{k-1} i_k}$ を最小ならしめる列 $(i_1 i_2 \cdots i_{k-1} i_k \cdots i_n)$ を求めればよい。 $s_{1, i_{k-1} i_k}$ を町 i_{k-1} より町 i_k へ行くに要する距離、または費用 $C(i_{k-1}, i_k)$ と考えれば、これは巡回セールスマン問題である。出発点となる町 i_0 を固定し、最後にその町 i_0 にもどるとしてよいから、上例1の解法を少しく変更すればよい。

(4.2) は、 $c(l, i) + c(i, j) + c(j, d) \leq c(l, j) + c(j, i) + c(i, d)$ となり、(4.4) は、 $c(l, i) + c(i, j) \leq c(l, j) + c(j, i)$ となる。アルゴリズムの段階 (i) では、 $c(i_0, i) + c(i, j) + c(j, d) \leq c(i_0, j) + c(j, i) + c(i, d)$ のとき $i_0 i j d$ の順をとる。また段階 (v) で列 $(i_0 i_1 \cdots i_{n-1} i_0)$ に対する目的関数 f の値は、 $\sum_{k=1}^{n-1} c(i_{k-1} i_k) + c(i_{n-1} i_0)$ である。

§5. 類似の問題への一般解法の適用

1. 所与の n 個の仕事のうち適当な t 個の仕事 ($t < n$) の順序で目的関数 f を最小ならしめるものを求める問題 (§2 の諸仮定を満足する) に対し、各一般解法 (§3.4, 4.4) を t 個の仕事列をうるまで適用すればよい。

2. 機械、仕事などは仮の名称と考えてよいので、§2 の諸仮定をみたす類似の問題にも、各一般解法が適用される。

§6. むすび

以上、§2 の諸仮定を満たす2種類の順序づけ問題に対し、統一的な一般解法 (§3.4, および 4.4) を示し、それらが容易に各具体例に応用されうることを理論的に示した。

実際問題としては、機械、仕事の数が多くなれば電子計算機の利用にまたねばならないが、現在、富士通の坪井武夫、原亭両氏の御協力により、§3.5 具体例の総所要時間最小の場合につき、FACOM 231 によるプログラム化がなされている。

参 考 文 献

- [1] Smith, W.E., "Various Optimizers for Single-Stage Production," *Nav. Res. Log. Quart.*, **3** (1956), 59—66.
- [2] 1. 鍋島, "m 台の機械に対する仕事の順序づけ問題の一般解," 1965年日本OR学会秋季研究発表会アブストラクト, 6—17.
2. 2'. 4. 各1台の場合は [1].
- 2'. 2台の場合は, Ignall, E and L. Schrage, "Application of the Branch and Bound Technique to Some Flow-Shop Scheduling Problems," *Oper. Res.*, **13** (1965), 7, 400—412.
3. 1台の場合, 任意の $C_i(T_i)$ に対し, Held, M and R.M. Karp, "A Dynamic Programming Approach to Sequencing Problems," *J. Soc. Ind. Appl. Math.* **10** (1962), 1, 196—210.