

プロジェクト・ネットワーク上での 人員機械配分計画法 (第1部)[†]

山 本 正 明*

§1. ま え が き

普通 PERT/CPM で用いるネットワークによる日程管理の手法では、人員・機械・設備など作業遂行上に必要な資源の制約については考慮していない。しかし実際の日程計画においては、資源の制約を考えないでよいというような場合はむしろ稀である。日程計画担当者はネットワークを組む過程で、たとえ explicit に資源の制約を考えない時でも、その制約から持込まれる順序関係を頭に入れた上で、それをネットワークに反映させているものである。この場合、資源の制約は2つの作業間での新たな順序関係の形で与えられる。

資源の制約から持込まれる複雑な諸条件を正しく日程計画の上に反映させ、能率の良い日程を組むためには、前もってプロジェクト全体での資源に関する制約条件を明らかにし、その下で最も有利な日程計画を作る方法が必要となる。この論文はそのような人員機械配分計画法について検討したものである。

1つのプロジェクトを完成させるのに必要な作業の集合は、各作業間の技術的な順序関係によってネットワークを形づくる。一方各作業を実行するのに必要な資源は、プロジェクト全体での利用可能量を制限されている。この制限の下で実行可能な日程計画を作るためには、ネットワークに固有な技術的順序関係の外に、資源による順序関係をネットワークに導入する必要がある。

資源による順序関係は技術的順序関係とちがって、その順序は固定されたものではなく日程計画者の意志で自由に選択しうるものである。したがって、計画作製の際の管理目標に対して最もふさわしい選択が望ましく、この順序関係の指定がそのまま実行可能日程のネットワークを与えることになる。

このような考え方による人員・機械配分計画法を順序指定型の計画法と呼ぶことにする。これに対してガント・チャート上の時点に個々の作業を割付けることによって実行可能日程を得るタイプの計画法は時点割付型と名付けることができよう。従来のこの種の日程計画法はいずれも時点割付型に属している。

[†] 1966年8月16日受理, 1966年5月12日春季研究発表会講演

* 法政大学工学部

順序指定型の計画法としては次の2種の方法が考えられる。

1) 資源順序列 (resource sequence) を与える法

2) 資源順序対 (resource order) を与える法

本論文第1部では資源順序列を与える方法, 第2部では資源順序対を与える方法を検討する。

§2. ネットワーク上での日程計算^{[1],[2]}

プロジェクト・ネットワークは FLOW 型の表現による。プロジェクトを構成する n 個の作業の集合を P とすると, P に属する各作業の間には次のような諸関係が成立っている。

1) 直接的な順序関係 $i \ll j \quad i, j \in P$

2) 先行作業集合 $PJ_j = \{i | i \ll j\}$

後続作業集合 $SJ_j = \{k | j \ll k\} = \{k | j \in PJ_k\}$

3) 先行作業行列 $\{P\} \quad p_{ij} = \begin{cases} 1 & i \ll j \\ 0 & \text{直接関係なし} \end{cases}$

後続作業行列 $\{S\} \quad s_{ij} = \begin{cases} 1 & j \ll i \\ 0 & \text{直接関係なし} \end{cases}$

4) 間接的な順序関係 $i \prec j$

$$i \ll c_1 \ll c_2 \ll \cdots \ll c_k \ll j$$

5) ネットワークにループあり。

$$i \prec i$$

6) 冗長な関係あり。

$$\begin{cases} i \prec j \\ i \ll j \end{cases}$$

今ループのないネットワークが与えられると, P の中に出発作業 s

7) $s = \{j | PJ_j = \phi\}$

と, 完了作業 f

8) $f = \{j | SJ_j = \phi\}$

とが少なくともそれぞれ1つは存在する。

各作業の所要時間 d_j とプロジェクト開始時刻 S , 終了時刻 F が与えられると, 各作業の日程時刻は次のように求められる。

9) 前向き日程計算 (最早開始時刻 ES_j の計算)

$$\begin{cases} ES_s = S \\ ES_j = \max_{i \in PJ_j} EF_i \\ EF_j = ES_j + d_j \end{cases}$$

10) 後向き日程計算 (最遅完了時刻 EF_j の計算)

$$\begin{cases} LF_f = F \\ LF_j = \min_{k \in SJ_j} LS_k \\ LS_j = LF_j - d_j \end{cases}$$

§ 3. 資源順序列の導入

次の条件の問題を考える.

- 1) プロジェクト P は n 個の作業からなり, その先行作業行列 $\{P\}$ が与えられている. $\{P\}$ はループを持たない.
- 2) 各作業の所要時間 d_j , およびこの作業の実行に必要な資源の集合が与えられている. これによって資源 $k(k=1, 2, \dots, m)$ を必要とする r_k 個の作業の集合 R_k が作られる. これを共通資源集合と呼ぶ.
- 3) 資源は次の性質を持つものとする.
 - i) 各資源はそれぞれ特有の共通資源集合を持ち, 同種類の資源は 1 つしかない.
 - ii) 同時に 2 個の作業では使用できず, 一度一つの作業に使われ始めたら, その作業中は連続して使用される.
 - iii) 作業は, それに必要なすべての資源がそろうまで開始できない.
- 4) プロジェクト中の全作業に必要な m 個の資源はプロジェクト期間中常に利用可能である.

この問題では, 資源 k が R_k に属する r_k 個の作業を順次 1 個ずつ作業していかななくてはならないから, このための順序関係が発生する. 今資源 k が作業 i を終った後, 直ちに作業 j にかかけられるという関係を記号 $i \xrightarrow{k} j$ で示すことにする (k は省略することがある). すると, R_k に属する r_k 個の作業の間に,

$$(3.1) \quad j^{(1)} \rightarrow j^{(2)} \rightarrow j^{(3)} \rightarrow \dots \rightarrow j^{(r_k)}$$

なる関係が必要となる. これを資源順序列と呼ぶ. $i \xrightarrow{k} j$ は作業間の順序関係という点では $i \ll j$ と全く同等の性質を持っているから, (3.1) からできる (r_k-1) 個の直接的な順序関係から $\{P\}$ と同じ形式で, 資源順序列行列 $\{\bar{P}_k\}$ が作られる.

今 m 個の資源について, それぞれ $\{\bar{P}_1\}, \{\bar{P}_2\}, \dots, \{\bar{P}_m\}$ が定まれば, これらを技術的順序関係 $\{P\}$ と重ね合せた

$$(3.2) \quad \{P^*\} = \{P\} + \sum_{k=1}^m \{\bar{P}_k\}$$

は, 資源の制約を満足したネットワークとなる. ただし行列の加法で 1 以上の値は 1 とみなすものとする.

$\{P^*\}$ から §2 で与えた日程計算により, 直ちに資源の制約を考慮した日程計画が計算できる. ただ, 新たに持ち込まれた順序関係のためにネットワーク中にループが形成され, そのため日程計算が不可能となる場合が起こり得る. $\{P^*\}$ にループがなければ, それはすべて実行可能な

日程を与えていることになる。

$\{P^*\}$ が実行可能であることを保証するためには、適当なループ・テストが必要である。今ネットワークの出発作業から $PJ_j = \phi$ である作業を順次外していくと、もしループのない場合はすべての作業がネットワークから外されてしまう。しかし、 $i < j$ が成立するような作業 i があると、 PJ_i は空集合になれない。したがって i の後続作業はすべてネットワークから外せない。この手順は、§2, 9) の前向き日程計算手順第2式によって ES_j を計算する場合、常に実行しているものである。この計算手順中 $PJ_j = \phi$ なる作業が1つもなくなり日程計算不可能となれば、ネットワークにループが検出されたことになる。したがって特別なループ・テストは必要ない。

$\{P^*\}$ には冗長な関係も含まれる可能性があるが、これは日程計算の結果には影響しないから、特別にこれを除くというようなことは考えないことにする。

$\{P^*\}$ から1つの日程計画が得られるのだから（ループになる場合を除いて）、もし $\{P^*\}$ のとりうるすべての場合を列挙することができれば、その中から目的関数に最もよく適合した日程計画を選び出すことができる。

m 個の資源によって得られる $\sum_{k=1}^m \{\bar{P}_k\}$ のとりうる組合せの数は、

$$r_1! r_2! \cdots r_m! \text{ 通り}$$

となるから、実際問題としては列挙法は膨大な計算量のため実行困難である。しかし、この種の組合せ問題で、最適解に確実に達する方法は列挙法しか考えられない。以後の解法はすべて実行可能な計算量の範囲で、いかにして良い近似解をうるかという問題に関するものである。

§ 4. 資源順序列の指定法

プロジェクト遂行時間 λ を最小化するという目的関数に対して、資源順序列の指定法を検討する。

4.1. 優先順位法

プロジェクトに属するすべての作業に対して、次の性質を満たすような優先順位 PR_i を与えるための優先順位規則を決める。

- 1) PR_i は作業 i に関して一意的に定まる。
- 2) 任意の2つの作業 i, j に関して、

$$PR_i \neq PR_j$$

この優先順位規則によって、 R_k に属する作業を優先順位の小さい作業順に並べて、これを資源順序列とする。すなわち、任意の2つの作業 i, j に関して

$$PR_i < PR_j \text{ ならば } i < j$$

なる順序関係が満たされるように資源順序列を作る。

優先順位は上の性質(2)により、等号になった時の処置を与えるため数段にわたる優先順位規則を決めておくのが普通である。それを $PR_i^1, PR_i^2, \dots$ で示す。

PR_i としてはプロジェクトの技術的順序関係 $\{P\}$ に矛盾しないものが望ましい。そこで資源による順序関係を入れない $\{P\}$ から計算した日程時刻 EF_i , ES_i , LS_i , LF_i を用いるのが実用的である。

〔例題 1〕

図 1 に示す機械製作プロジェクトを考える。

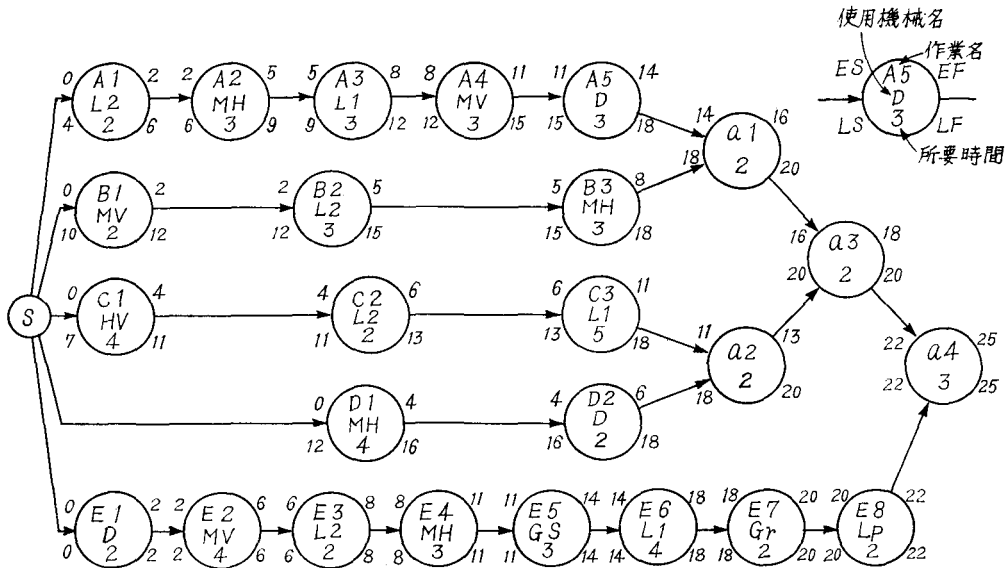


図 1. 機械製作プロジェクトのネットワーク

このプロジェクトを構成する作業は次の25工程の作業である。

機械加工作業	部品 A	5 工程	} 21 工程
	部品 B	3 "	
	部品 C	3 "	
	部品 D	2 "	
	部品 E	8 "	

組立作業	部品 A + 部品 B	→ a ₁	} 4 工程
	部品 C + 部品 D	→ a ₂	
	a ₁ + a ₂	→ a ₃	
	a ₃ + 部品 E	→ a ₄ (完成品)	

機械加工に使用する工作機械は次の 8 種である。—— L_1 , L_2 , MH , MV , D , G_s , G_r , L_p このうちはじめの 5 種の工作機械は工程間でその使用がぶつかる可能性があるので適当な配分計画が必要である。各工作機械は 1 台ずつしか利用できないものとして、このプロジェクトの日程計画を作れ。

i) 資源の制約を入れない日程計算を図 1 に直接記入。

ii) 優先順位 ($PR_i^1=LS_i$, $PR_i^2=d_i$, $PR_i^3=i$) を用いて p_k を作る.

$$\left\{ \begin{array}{l} p_{L1} : A3 \rightarrow C3 \rightarrow E6 \\ p_{L2} : A1 \rightarrow E3 \rightarrow C2 \rightarrow B2 \\ p_{MH} : A2 \rightarrow E4 \rightarrow D1 \rightarrow B3 \\ p_{MV} : E2 \rightarrow C1 \rightarrow B1 \rightarrow A4 \\ p_D : E1 \rightarrow A5 \rightarrow D2 \end{array} \right.$$

iii) 上の p_k を書き込んだネットワークについて, 日程計算を行なうと図2の結果を得る.

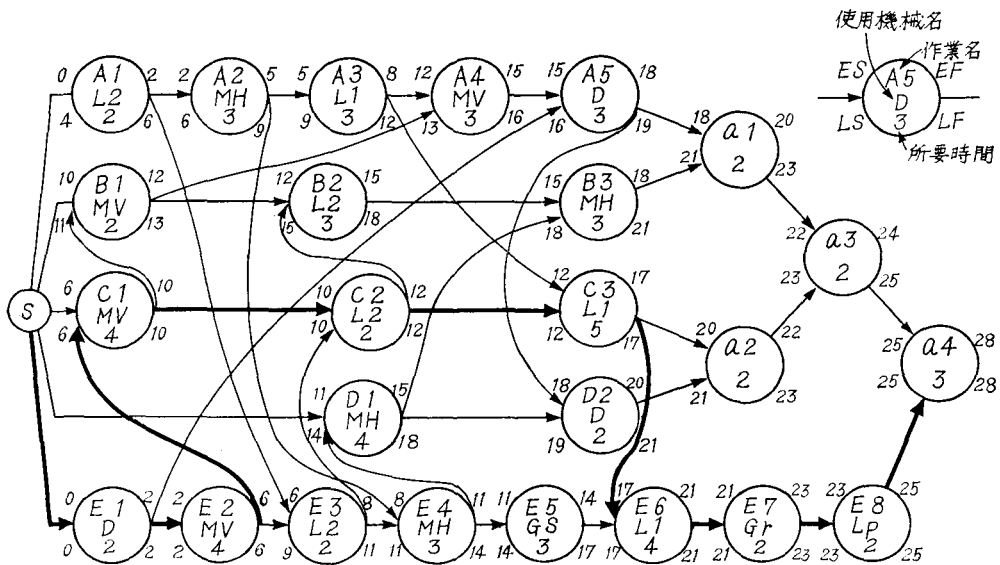


図2. L_3 優先順位による p_k を加えたネットワーク

プロジェクト遂行時間 λ は 25 から 28 にのびた. この問題では $\lambda=28$ は最適解であることが解っている.

($PR_i^1=ES_i$, $PR_i^2=1/d_i$, $PR_i^3=i$) を用いると $\lambda=29$ が得られる. 優先順位法は手軽に計算できるという点が強みで, 手元にあるネットワークにもとづいてプロジェクト中の特定の機械の使用計画を検討するというような場合, 簡単な手計算で問題点が把握できる.

4.2. 競合作業集合による列挙法

§3 の手順で作られる先行作業行列 $\{P^*\}$ のとりうるすべての組合せの集合を S とする. プロジェクト遂行時間の最小値を λ_{opt} とすると, このプロジェクトの最適日程は

$$(4.1) \quad \lambda_{opt} = \min_{\{P^*\} \in S} EF_f$$

を与えるような $\{P^*\}_{opt}$ を求めればよい. (4.1) 式で f は完了作業の作業番号を示す. この式を実際に計算する (列挙法) のは実行困難である.

そこで S の部分集合 S' を考える. S' に属する日程計画については

$$(4.2) \quad \lambda_{opt} \leq \min_{\{P^*\} \in S'} EF_f \leq \lambda_{max}$$

が成立つ。ここで λ_{max} は S についての最大値を示す。今 S' として次のような性質のものを選びだすことにする。

- i) S' から得られる λ の集合が、比較的高い確率で λ_{opt} を含んでいる。
- ii) S' に属する $\{P^*\}$ の個数は、実際に計算できる範囲の大きさである。

すると、 S の代りに S' を用いて、

$$(4.3) \quad \lambda_{min} = \min_{\{P^*\} \in S'} EF_f$$

を λ_{opt} の近似値として採用することができる。

S' を作る方法は種々考えられるが、どれを用いるかは問題の条件に応じて実験的に決めるより手がない。

[A] $LF-ES$ 競合作業集合

$\{P\}$ による日程計算の結果を用いて、 R_k に属する 2 つの作業 i, j の間に

$$(4.4) \quad LF_i > ES_j \quad i, j \in R_k$$

の関係が成立てば、作業 i, j は資源 k の専有をめぐって時間的に競合する可能性が大きい。そこで (4.4) の成立っている関係を $LF-ES$ 競合関係とよび、この関係が互に成立っている作業の集合を $LF-ES$ 競合作業集合と呼ぶ。ある特定の関係で定まる競合作業集合を $\bar{R}_{kg}$ で示すと、 $\bar{R}_{kg}$ は R_k の部分集合である。同じ k に対して複数個の $\bar{R}_{kg}$ が得られることもあるので、時間的に早いものから $g=1, 2, \dots$ を与えて区別する。

S が R_k のすべての作業の順列を組合せて得られるのと同様に、 S' は $\bar{R}_{kg}$ のすべての作業の順列を組合せて作ることにする。この場合いずれの $\bar{R}_{kg}$ にも属さない作業及び互に異なる $\bar{R}_{kg}$ の間の順序関係は $\bar{R}_{kg}$ に矛盾しない適当な優先順位によって固定するものとする。

[B] $LF-LS$ 競合作業集合

$$(4.5) \quad LF_i > LS_j \quad i, j \in R_k$$

なる関係により $\bar{R}_{kg}$ を作る。

[C] $EF-ES$ 競合作業集合

$$(4.6) \quad EF_i > ES_j \quad i, j \in R_k$$

なる関係により $\bar{R}_{kg}$ を作る。

[D] フロート調整 $LF-ES$ 競合作業集合

作業 i, j の競合関係に各作業のフロート

$$TF_i = LF_i - EF_i = LS_i - LF_i$$

が寄与する割合を減量係数 k_i, k_j で調整すると、(4.4) あるいは

$$LF_i - k_i(LF_i - EF_i) > ES_j + k_j(LS_j - ES_j)$$

となる。したがって

$$(4.7) \quad (1-k_i)LF_i + k_iEF_i > (1-k_j)ES_j + k_jLS_j$$

なる関係によって $\overline{R}_{kg}$ を作る事ができる。(4.7)において

$k_i=0, k_j=0$ ならば [A] の場合となり,

$k_i=1, k_j=0$ " [B] の場合となり,

$k_i=0, k_j=1$ " [C] の場合となる.

k_k, k_j を適当に調節することによって, S' の大きさを与えることができる.

例題1の機械製作プロジェクトについて, $k_i=1, k_j=0$ すなわち LF - LS 競合作業集合を作り, これを LF 優先順位による p_k の中に加えると,

$$\left\{ \begin{array}{ll} \overline{R}_{L1,1} = \{C3, E6\} & p_{L1} : A3 \rightarrow \{C3, E6\} \\ \overline{R}_{L2,1} = \{C2, B2\} & p_{L2} : A1 \rightarrow E3 \rightarrow \{C2, B2\} \\ \overline{R}_{MH,1} = \{A2, E4\} \\ \overline{R}_{MH,2} = \{D1, B3\} \end{array} \right\} p_{MH} : \{A2, E4\} \rightarrow \{D1, B3\}$$

$$\left\{ \begin{array}{ll} \overline{R}_{MV,1} = \{C1, B1\} & p_{MV} : E2 \rightarrow \{C1, B1\} \rightarrow A4 \\ \overline{R}_{D,1} = \{A5, D2\} & p_D : E1 \rightarrow \{A5, D2\} \end{array} \right.$$

となる. 6組の $\overline{R}_{kg}$ 内で作業の順序をいれかえ, その組み合わせを作ると, $(2!)^6=64$ 通りの $\{p^*\}$ が得られる. この中で λ の最も小さい日程計画を近似解として採用すればよい. $\lambda_{\min}=28$ の日程が得られる.

[E] 優先順位比較法

資源 k について2組の優先順位規則を用いて2個の p_k を作る.

$$\left\{ \begin{array}{l} p_k : i_1 \rightarrow i_2 \rightarrow \dots \rightarrow i_r \\ p'_k : i'_1 \rightarrow i'_2 \rightarrow \dots \rightarrow i'_r \end{array} \right.$$

この両資源順序列において, 作業順序が互に入れ換っている部分を競合作業集合 $\overline{R}_{kg}$ とする.

2組の優先順位としては, 最早時刻と最遅時刻を組み合わせたものが有効である. たとえば例題1について ES - LS 優先順位比較を行なって競合作業集合を作ってみると, p_k は

$$\left\{ \begin{array}{l} p_{L1} : A3 \rightarrow C3 \rightarrow E6 \\ p_{L2} : A1 \rightarrow \{B2, C2, E3\} \\ p_{MH} : \{D1, A2, B3, E4\} \\ p_{MV} : \{B1, C1, E2\} \rightarrow A4 \\ p_D : E1 \rightarrow \{D2 \rightarrow A5\} \end{array} \right.$$

となる. $3! \times 4! \times 3! \times 2! = 1728$ 通りの $\{p^*\}$ が得られる.

[例題2]

図3に示す $6 \times 6 \times 6$ の Job Shop 型問題について競合作業集合を作れ.

この問題を列挙法で扱くと $(6!)^6$ 個の日程計画が生ずる. S' によってこれがどの位減るかを

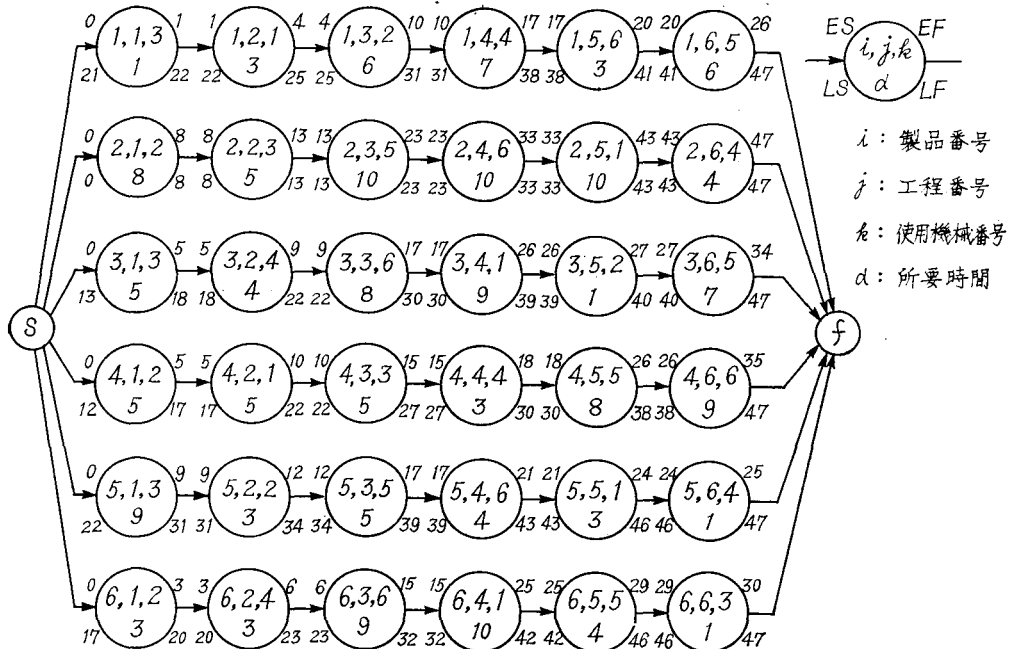


図3. 6×6×6 Job Shop 型問題のネットワーク

表1. 競合作業集合の大きさ (例題2)

方 法	k_i	k_j	p_1	p_2	p_3	p_4	p_5	p_6	総組合せ数
LF—ES 競合	0	0	6!	6!	6!	6!	6!	6!	1.39×10^{17}
EF—ES 競合	1	0	3!	5!	5!	$2! \times 2!$	6!	$2! \times 2! \times 2!$	1.99×10^9
LF—LS 競合	0	1	3!	1	2!	$2! \times 2!$	$2! \times 3!$	$3! \times 3!$	20,739
フロート調整	0.5	0.5	$2! \times 4!$	5!	5!	$2! \times 2! \times 2!$	5!	$2! \times 4!$	3.19×10^{10}
"	0.8	0.5	3!	3!	2!	1	2!	$2! \times 3!$	1,728
"	0.8	0.8	2!	1	2!	1	1	1	4
EF—LS 競合	1	1	1	1	1	1	1	1	1
優先順位組合せ*	$LS(1/d_i) - EF(d_i)$		$2! \times 2! \times 2!$	3!	4!	$2! \times 2! \times 2!$	$3! \times 3!$	$2! \times 4!$	1.59×10^7
	$LF(d_i) - ES(1/d_i)$		$2! \times 2! \times 2!$	3!	5!	$2! \times 2!$	$3! \times 2!$	$3! \times 2!$	3.32×10^6
	$LF(d_i) - ES(LF)$		$2! \times 2! \times 2!$	1	5!	$2! \times 2!$	$3! \times 2!$	$3! \times 2!$	138,240
	$\frac{EF+LF}{2}(d_i) - ES(1/d_i)$		2!	3!	$2! \times 2!$	2!	2!	3!	1,152

* () 内は PR_i^2 を示す。 PR_i^3 はすべて i である。

表1に示す。

表1から解るように S' の数を極端に減らすのはなかなか難しい。もちろん k_i, k_j を適当に調整すれば数は小さくなるが、あまり小さくすると競合作業集合を作る本来の意味が失なわれる。というのは資源順序列が競合作業の組合せではなくて、競合しない作業の順序を決める優先順位によってほとんど決められる。したがって、優先順位法で求める場合とあまり変りがなくなるからである。

S の数の非常に大きい問題では、他の考え方による数の減らし方が欲しい。

4.3. 日程計画集合からのランダムな選択

$\bar{R}_{kg}$ の組合せによって S' を得た前節の方法では、問題によって S' に属する日程の数変動してあらかじめ予想ができないから、集合のつくり方を前もって確定するのは難かしい。また問題に対してあまり小さい S' を作ると、最適日程に近い計画を全く含まなくなるおそれがある。

そこで、 S' をあまり小さくすることは考えず、むしろ最適日程に近いものを含むことが保障される程度には大きくしておいて、 S' の中からランダムな選択によって、大きさが一定の S'' を作ることにした方がよい。 $S' \in S'$, $S' \in S$ であるから、 S'' も当然前節で述べた S' に必要な性質を持っている。したがって

$$(4.8) \quad \lambda_{\min} = \min_{\{P^*\} \in S''} EF_j$$

を与える $\{P^*\}$ から最適日程に近い日程計画が得られる。

4.4. 資源修正優先順位による日程の改善

前節まで用いてきた優先順位は主としてネットワークの技術的順序関係から決められたもので、実際の日程を組む段階でも一定不変のものとして扱ってきた。しかし実際の日程では資源の制約から多くの資源順序列が持込まれ、その結果プロジェクトの日程時刻は変更を受ける。

そこで、他の資源についてはすべての資源順序列が確定している状態で、特定の R_k に属する作業のについての日程時刻から優先順位を決めて、これを資源修正優先順位 PR_i^* と名付ける。

今ある $\{P^*\}$ において、 R_k に属する作業について次のような日程時刻を定める。 PJ_i の中に p_k にもとづく先行作業があればこれを取除き PJ_i^* と名付ける。すなわち、

$$\begin{aligned} h \xrightarrow{k} i \text{ なる } h \text{ が存在すれば } PJ_i^* &= PJ_i - h \\ " \quad " \quad \text{存在しなければ } PJ_i^* &= PJ_i \end{aligned}$$

PJ_i^* を用いて次の日程時刻を計算する。

$$(4.9) \quad ES_i^* = \max_{j \in PJ_i^*} EF_j$$

$$(4.10) \quad EF_i^* = ES_i^* + d_i$$

同様にして、

$$\begin{aligned} i \xrightarrow{k} h \text{ なる } h \text{ が存在すれば } SJ_i^* &= SJ_i - h \\ " \quad " \quad \text{存在しなければ } SJ_i^* &= SJ_i \end{aligned}$$

$$(4.11) \quad LF_i^* = \min_{j \in SJ_i^*} LS_j$$

$$(4.12) \quad LS_i^* = LF_i^* - d_i$$

4.1 節で述べた優先順位 PR_i の代りに PR_i^* として EF_i^* , ES_i^* , LS_i^* が利用できる。 PR_i^* は他の p_k の状態によって変っていくから、すべての資源につい PR_i^* を満たす日程を作るためには、逐次的な接近法が必要となる。その手順を要約すると次のようになる。

1) 適当な優先順位を用いて初期日程計画 $\{P^*\}$ を作る。 $\{P^*\}$ について各作業の日程時刻を

表 2. 資源修正優先順位による日程の改善

a) 初期日程 ($PR_i^1 = ES_i$, $PR_i^2 = LS_i$, $PR_i^3 = i$)資源修正優先順位 $PR_i^* = ES_i^*$ $PR^2 =$ 現状優先

日程番号		λ	入換順序数
1	初期日程計画	67	3
11	Step 1	67	2
12	Step 2	64	4
13	Step 3	60	4
14	Step 4	65	並びかえ完了

b) 初期日程 a) の日程14

資源修正優先順位 $PR_i^* = LS_i^*$ $PR^2 =$ 現状優先

日程番号		λ	入換順序数
2 (=14)	初期日程計画	65	6
21	Step 1	65	4
22	Step 2	64	4
23	Step 3	58	1
24	Step 4	58	1
25	Step 5	58	並びかえ完了

c) 初期日程 a) の日程14

資源修正優先順位 $PR_i^* = LF_i^*$, $PR^2 =$ 現状優先

日程番号		λ	入換順序数
3 (=14)	初期日程計画	65	7
31	Step 1	65	4
32	Step 2	60	2
33	Step 3	57	2
34	Step 4	57	3
35	Step 5	60	2
36	Step 6	61	並びかえ完了

計算する。

2) R_k ごとに資源修正優先順位 PR_i^* を用いて修正された資源順序列 p_k を作る。3) p_1^* , p_2^* , p_3^* , ……によって $\{P^*\}$ を作り, これを新たな初期日程計画として 1) にもどる。4) もし, すべての p_k が修正を必要としなくなれば, その時の $\{P^*\}$ は PR_i^* を満足した日程となっている。

例題 2 にこの方法を適用した一例を表 2 に示す。

初期日程計画と資源修正優先順位との組合せによって, 日程が改善される場合もされない場合も起こる。しかも並びかえの完了した時よりも, 完了前に λ の小さい日程が現われることが多い。

そこでいくつかの優先順位と初期計画 (前の Step の最終段の日程を用いてもよい) の組合せ

を作って、並べ換えを繰返すと、比較的 λ の小さい日程計画の集合を選択的に作り出すことができる。そこでこの手順を 4.3 節で述べた S' を作る手段として利用すると比較的少い試行で良好な日程計画を得ることができる。この場合修正優先順位として ES_i^* , EF_i^* のグループと LS_i^* , LF_i^* のグループを交互に繰返すと非常に結果がよい。

§ 5. 複数機械問題への拡張

前節までに取扱った問題は 1 作業 1 資源でプロジェクト全体で利用できる資源の数も 1 個に限られていたが、一般には次のような分類が考えられる。

- | | |
|---------------------------|--------------|
| 1) 1 作業に必要な資源の種類 | a. 1 種類 |
| | b. 複数種類 |
| 2) 1 作業に必要な資源の個数 (各種類ごとに) | a. 1 個 |
| | b. 複数個 |
| | c. 可変 (工数指定) |
| 3) プロジェクト全体で利用可能な資源の種類 | a. 1 種類 |
| | b. 複数種類 |
| | c. 時間的に可変 |
| 4) プロジェクト全体で利用可能な資源の個数 | a. 1 個 |
| | b. 複数個 |
| | c. 時間的に可変 |

たとえば「例題 1」は (1a, 2a, 3b, 4a) の問題であった。

5.1. 同種機械が 2 台以上使用できる場合 (1a, 2a, 3b, 4b)

資源 k がプロジェクト全体で $\bar{Q}$ 個使用可能であるとする。この場合には共通資源集合 R_k に属する作業から $\bar{Q}$ 本の資源順序列 p_{kq} を作る必要がある。

$$(5.1) \quad \begin{cases} p_{k \cdot 1} : j_1^{(1)} \rightarrow j_1^{(2)} \rightarrow j_1^{(3)} \rightarrow \dots \rightarrow j_1^{(r_{k \cdot 1})} \\ p_{k \cdot 2} : j_2^{(1)} \rightarrow j_2^{(2)} \rightarrow \dots \rightarrow j_2^{(r_{k \cdot 2})} \\ \vdots \\ p_{k \cdot Q} : j_Q^{(1)} \rightarrow j_Q^{(2)} \rightarrow \dots \rightarrow j_Q^{(r_{k \cdot Q})} \end{cases}$$

$$(5.2) \quad r_{k \cdot 1} + r_{k \cdot 2} + \dots + r_{k \cdot Q} = r_k$$

この場合は次のような方針によって分けることにする。

- 1) 各 p_{kq} に属する作業時間の和

$$(5.3) \quad D_{kq} = \sum_{i \in p_{kq}} d_i$$

をできるだけ等しくする。

- 2) 各 p_{kq} 内では競合作業集合の発生をできるだけ少くする。

上の方針により具体的手順を次のように進めることにする。

- 1) 適当な優先順位規則を用いて R_k に属するすべての作業を 1 本の資源順序列 p_k に並べる。
- 2) p_k のはじめ $\bar{Q}$ 個をとり出して、1 個ずつ p_{kq} の最初の作業 $j_q^{(1)}$ とする。
- 3) p_k の残りの順序列の先頭の作業から順番に、 D'_{kq} の最も小さい p_{kq} に付け加えていく。
 D'_{kq} のダッシュは (5.3) の和がその時までには作られた p_{kq} についての和であることを示す。 D'_{kq} の等しい場合は、1) の優先順位がより離れた作業の後に続ける。

〔例題 3〕

例題 1 の機械作製プロジェクトで機械 MV が 2 台使用できるものとして機械配分計画を作れ。

$$\hat{p}_{MV} : E2 \rightarrow C1 \rightarrow B1 \rightarrow B1 \rightarrow A4 \quad (LS \text{ 優先})$$

$$\begin{cases} p_{MV.1} : E2 \rightarrow B1, & D_1 = 6 \\ p_{MV.2} : C1 \rightarrow A4, & D_2 = 7 \end{cases}$$

新らたな資源順序列によって日程計算を行なうと図 4 の結果を得る。

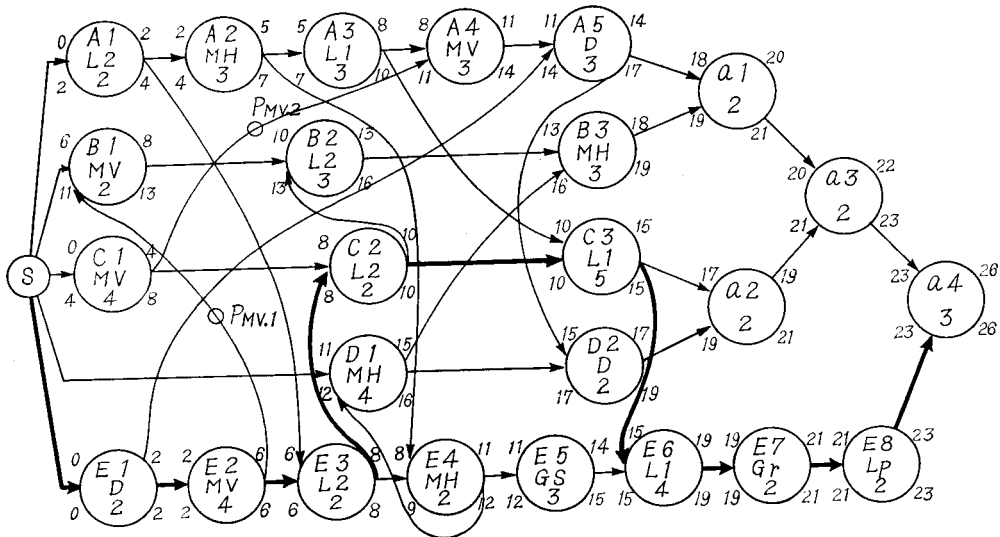


図 4. MV 2 台使用の場合の日程計画

5.2. 1 作業で複数台の機械を必要とする場合 (1a, 2b, 3b, 4b)

作業 i が資源 k を u_{ki} 個必要とする。この場合 (5.1) の q_k 個の資源順序列のうち u_{ki} 個の系列が同一の作業をその中に含んでいればよい。そこで 5.1 の手順に若干の変更を加える。

- 1) $\hat{p}_k$ を作る時 2 個以上の資源を必要とする作業 i については、 u_{ki} 個だけ i の順序列を繰返ししておく。

$$\hat{p}_k : j^{(1)} \rightarrow j^{(2)} \rightarrow \dots \rightarrow \underbrace{i \rightarrow i \rightarrow \dots \rightarrow i}_{u_{ki} \text{ 個の繰返し}} \rightarrow j^{(l)} \rightarrow \dots \rightarrow j^{(rk)}$$

- 2) q_k 個の資源順序列に $\hat{p}_k$ を配分する過程で、 p_{kq} 中に自分と同一の作業 i がすでに割当てられている場合には、 D'_{kq} がより大きい p_{kq} の方に繰上げて割当てる。

(1b, 2b, 3b, 4b) の問題では同一の作業がいくつかの R_k に同時に属していることになるが、1つの R_k について考えれば、従来と同様の取扱が可能である。ただ i を通じて異なった R_k に属する作業の間に新たな順序関係が持込まれることになる。

5.3. 必要な資源数が可変の場合 (1a, 2c, 3b, 4b)

作業 i は資源 k を $\bar{u}_{ki}$ 個までは同時に使用できるが、それ以下でも作業は開始でき、全体で $(d_i \times \bar{u}_{ki})$ の工数を必要とする。従って i の所要日数が使用できる資源の数に応じて可変の問題となる。この場合、1個の資源は所要時間 d_i の整数倍でしか作業に用いられないものとすれば、図5のような作業の分割を考えて、今までの手法がそのまま利用できる。

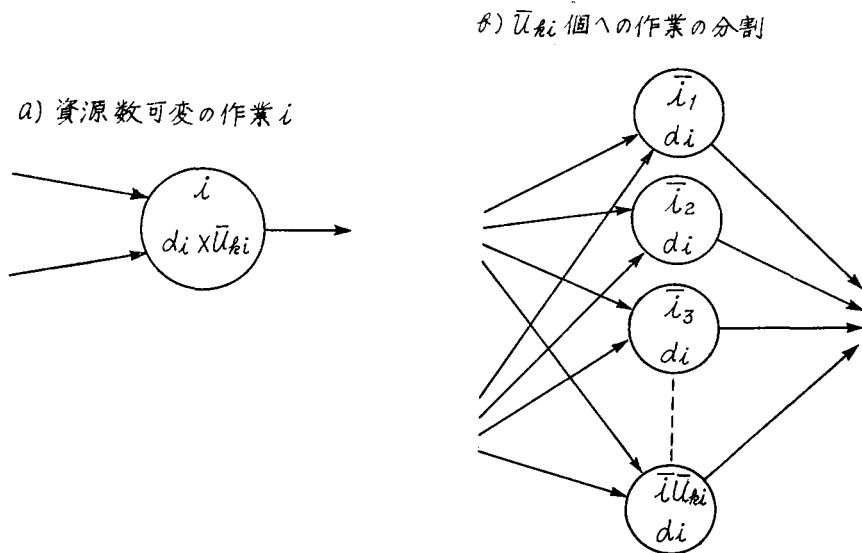


図5. 資源数可変の場合の作業の分割

以上の説明で明らかなように、資源の種類・台数が増加するとともに資源順序列が増してネットワークが非常に複雑となる。プロジェクトの総人員の配分計画を立てるといような形の問題をこの方法で処理するのは著しく不利である。このためには、はじめに述べた資源順序対を指定していく方法を用いる必要がある。

§ 6. 実用上の問題点

{ P^* } から日程時刻を計算するには能率の良い PERT 計算のアルゴリズムが与えられている。この場合資源順序列によって作業間の順序は変動するから、いわゆるトポロジカル・オーダリングを行なわないで、作業ロケーションのリンク構造だけに必要な変更を与える方式のアルゴリズムが適している^[3]。このような計算方式を用いた場合は、かなりの数の S' (または S'') について日程を検討することが可能となる。

計算量は、より正確な解を求めようとすれば、当然増加する。しかし問題は日程計画であるか

ら、拙速を尊ぶという場合もしばしば起こり、4.1 節の優先順位規則だけで日程を決める方法も有効となる。すでにプロジェクトがネットワークで与えられている時、特定の資源についてその配分計画をたてる必要に迫られたような場合には、ネットワーク上の手計算で簡単に実行可能解が求められる。資源順序列を図面に書き込みながら検討すれば、全体の見通しも良くきて、PERT 手法の利点が十分活用できる。

たとえば、〔例題 1〕に示したような機械製作の基準日程を、実際の Shop に割付けていくような場合、直接機械加工に加わらない関連作業との時間関係（部品発注のリード・タイムや機械の定期補修など）も自由にプロジェクト・ネットワークの中に組込んで、それらとの関連の上で資源配分を行なうことができる。

また資源配分の対象となる資源の選択も自由であるから、日程にクリティカルにきいてくると思う資源だけにこの方法を適用して、他の資源は手直しによって割付けるといような弾力的な利用法も考えられる。このような利点は、ネットワーク上で技術的序順関係にもとづくマスター・プランを確立した上で、資源配分を取扱っていることによるものである。

資源配分の結果として得られた日程計画もまたネットワークの形をしているから、これがそのままプロジェクトの初期日程計画となってその後の日程管理システムに継がり、フォロー・アップを続けていくことができる。

従来のこの種の手法との比較検討は第 2 部で資源順序対を与える方法について述べた後総合的に検討することにする。

参 考 文 献

1. 関根智明, PECT・CPM 入門, 日科技連, 1965.
2. Levy, F.K., G.L. Thompson, and J.D. Wiest, "Introduction to the Critical-Path Method," Chap. 20 of *Industrial Scheduling*, Prentice-Hall, 1963. 関根智明訳, インダストリアル・スケジューリング, 竹内書店, 1966.
3. Lass, S.E., "PERT Time Calculations without Topological Ordering," *Communications of the ACM*, 8, 3 (1965), 172-174.