PERT／MANPOWERの最適解法について
——作業の分割が可能な場合——

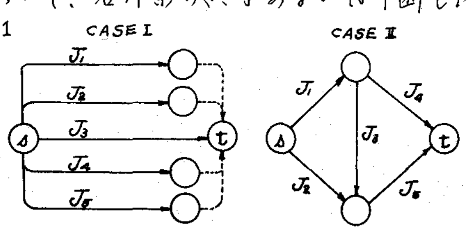
京都大学工学部　正員　工博　吉川和広
京都大学大学院　学生員　工修○春名　攻

1. はじめに　工事用資源の量的制約のもとでのスケジューリングの問題は、種々の分野において研究が行なわれ、解法の開発の努力が払われているようであるが、一部のJob Shop Schedulingの問題あるいは特殊な場合を除いては最適解法と呼ばれるものは殆んど開発されていない。その理由は、この種の問題の数学モデルの作成は可能であっても、複雑な組合せ問題であるため解くことができないことである。ここでとりあげたPERT/MANPOWERの問題においても、主として実用的な側面からの要請にしたがって、比較的小さな完了時刻を与えると考えられる近似解をスケジュールとして求める手法が開発されているにすぎない。しかし、列挙法によって最適解を求めた場合、これらの近似解と最適解の間に大きなへだたりが存在することが多い。したがって、計算法が若干複雑になったとしても最適解法へのアプローチは是非とも行なっておく必要があると考える。本研究においてはこのような観点にたって、作業分割の可能な場合についてのPERT/MANPOWERの最適解法の提案を行なうことにする。

2. パターンおよびスケジュールの実行可能性の検討　これまでに我々が行なってきた一連の事例研究をとおして、スケジュールにおける作業間の順序関係には①施工技術的な側面から先決的に定められる技術的な順序関係と②各種工事用資源の配分方法から定められる管理的順序関係の2通りのものが存在することが明らかになった。PERT/MANPOWERの問題は、「①の技術的順序関係を与件とした場合、資源制約量Rの下で、プロジェクト完了時刻λが最小になるような②の管理的順序関係すなわちスケジュールを求めよ。」のように表わされる。さて、①の技術的順序関係の有無をネットワークとして考えると図－1のCASE I, CASE IIのようにあらわされる。すなわち、CASE Iの場合には先決的な順序関係は存在していないがCASE IIの場合には明らかに先決的な順序関係が存在している。

(1)パターン　さて、図－1に示したネットワークのそれぞれに対して、作業所要時間d_i, 所要資源量r_iおよび資源制約量Rを表－1のように与える。このような条件の下で、1つの実行可能なスケジュールを求めたものを図－2に示した。このスケジュールチャートにおいて、各作業の終了あるいは中断した時刻で時間軸を区分すると、時間区間$I_1, I_2, \cdots, I_5$

図－1

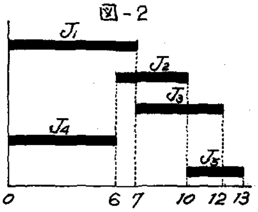


表－1

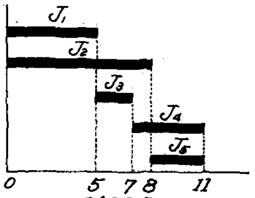
	CASE I (R=7)					CASE II (R=10)				
	i					i				
J_i	1	2	3	4	5	1	2	3	4	5
d_i	7	4	5	6	3	5	8	2	4	3
r_i	3	2	5	4	2	3	5	4	5	4

が求められる。いま作業J_iが区間I_kで実施されているときに$a_{ik}=1$とし、そうでないときには$a_{ik}=0$とおくと、各区間において、同時刻に実施されている作業の組合せが求められる。このような作業の組合せをパターンと定義し、P_kであらわすと、CASE I, CASE II の場合のパターンは表-2のようになる。さらに、各区間I_kの長さすなわち時間をx_kであらわすと、$\mathbb{P}_k$, x_k, d_iの間には、

図-2



CASE I



CASE II

① $$\sum_k \mathbb{P}_k x_k = \mathbb{D} \qquad ここで、\mathbb{D}=\begin{pmatrix} d_1 \\ \vdots \\ d_m \end{pmatrix}$$

の関係が成立している。

(2)実行可能なパターン　以上のことから明らかなように、スケジュールは、各区間に対するパターンの割当てによって規定される。もちろん、各パターンは資源制約条件、

② $$\mathbb{r}\cdot\mathbb{P}_k \leqq \mathbb{R} \quad (k=1,2,\cdots) \qquad ここで、\mathbb{r}=(r_1, r_2, \cdots, r_m),\ \mathbb{R}=\left.\begin{pmatrix} R \\ \vdots \\ R \end{pmatrix}\right\} m$$

を満していなければならない。さらにCASE II のように、技術的順序関係が成立している場合には、1つのパターンに含まれる作業は同時作業が可能なものでなければならない。従って、パターンを求める場合には、同一パターンに同時作業の不可能な作業を含まないように予め検討しておかなければならない。以下に、実行可能なパターンを判別するための条件について示すことにする。いま、S_Tを与えられた技術的順序関係をあらわすマトリックスとする。

表-2

		区間	I_1	I_2	I_3	I_4	I_5
CASE I	$\mathbb{P}_k$		1	1	0	0	0
			0	1	1	0	0
			0	0	1	1	0
			1	0	0	1	0
			0	0	0	0	1
	x_k		6	1	3	2	1
		区間	I_1	I_2	I_3	I_4	I_5
CASE II	$\mathbb{P}_k$		1	0	0	0	–
			1	1	1	0	–
			0	1	0	0	–
			0	0	1	1	–
			0	0	0	1	–
	x_k		5	2	1	3	0

③ $$S_T=(s_{ij}^T),\quad s_{ij}^T=\begin{cases} 1, & J_i < J_j \left(\begin{array}{l}作業J_iが作業J_jの先行作業;\\ 直接的な先行作業でなくてもよい\end{array}\right) \\ 0, & その他の場合 \end{cases}$$

つぎに、技術的な順序関係に矛盾するような作業間の順序関係をあらわすマトリックスを$\bar{S}_T$とおくと、$\bar{S}_T$は、

④ $$\bar{S}_T=(\bar{s}_{ij}^T),\quad \bar{s}_{ij}^T=\begin{cases} 1, & s_{ji}^T=1 のとき \\ 0, & その他の場合 \end{cases}$$

のように求められる。さて、同時作業が可能な作業間の対応関係を表わすマトリックスをS_0であらわすと、S_0は、

⑤ $$S_0=(s_{ij}^0),\quad s_{ij}^0=\begin{cases} 1, & J_i \otimes J_j\ (作業J_iと作業J_jの同時作業が可能) \\ 0, & その他の場合 \end{cases}$$

とあらわされるが、s_{ij}^0は、s_{ij}^Tおよび$\bar{s}_{ij}^T$を用いて次式のように求められる。

⑥ $$s_{ij}^0 = 1 - s_{ij}^T - \bar{s}_{ij}^T$$

このようにS_0を定めると、実行可能なパターン$\mathbb{P}_k$において、$a_{ik}=1$であるような作業の集合をA_kとおき、つぎに、作業J_iに対してマトリックスS_0から$s_{ij}^0=1$であるような作業J_jの集合をとりだしこれをA_iとおくと、A_kとA_iの間には、

⑦ $\bigcap_{J_i \in A_k} A_i \supseteq A_k$

が成立していなければならない。式⑦が、実行可能なパターンの判別式である。

(3)実行可能スケジュール　実行可能なパターンが求められ、各区間へこれらのパターンを割当ててスケジュールを定めるとき、CASE I の場合には何の問題もないが、CASE II の場合には、パターンの割当てによって求められるスケジュールが、与えられた技術的順序関係を満足させていなければならない。いま、パターン$\mathbb{P}_k$を用いて表わされるスケジュールのマトリックス

⑧ $\mathbb{B} = (\mathbb{P}_1\ \mathbb{P}_2 \cdots\cdots \mathbb{P}_m)$

から一意的に導びかれる順序関係および同時作業の対応関係を表わしたものをSとすると、

⑨ $S = (s_{ij})\ ,\ s_{ij} = \begin{cases} 1, & J_i < J_j \text{ あるいは作業} J_i \text{と作業} J_j \text{が同時作業のとき、} \\ 0, & \text{その他の場合} \end{cases}$

のように示される。ここで、$\mathbb{B}$が実行可能かどうかを調べるが、この基準は以下のようにして導びかれる。すなわち、$\mathbb{B}$が実行可能であることは、SがS_Tに矛盾していないことを示せばよいから、これを調べるためにつぎのような作業間関係のマトリックスS_aを設定する。S_aは作業間の順序関係と同時作業関係のような許容的な関係をすべて表わしたもので、

⑩ $S_a = (s_{ij}^a)\ ,\ s_{ij}^a = \begin{cases} 1, & J_i < J_j \text{ あるいは } J_i \otimes J_j \text{ のとき} \\ 0, & \text{その他の場合} \end{cases}$

のように示される。ここで、S_aは、

⑪ $S_a = S_T + S_0$　（ここで、$1+1=1,\ 1+0=0,\ 0+0=0$とする。）

であるから、上式の関係に式⑥を代入すると、

⑫ $s_{ij}^a = 1 - \bar{s}_{ij}^T$

として求められる。以上のような定義によって定めたS_aを用いると、スケジュールが実行可能であるための条件は、

⑬ $\Delta S = S_a - S \geq O$（零行列）

が成立することである。さて、ΔSが上式を満さないときには、現在のスケジュールは実行可能ではないが、スケジュールとパターンの関係からつぎの２つの場合が考えられる。

ⓐパターンの区間への割当てを適当に変更することによって$\Delta S \geq O$とすることができる。

ⓑどのように割当てを変更しても$\Delta S \geq O$とすることはできない。

ⓑの場合には、新しいパターンの組合せを考えなければ実行可能なスケジュールとはならないことを示している。さて、以上でモデル作成のための準備を終えたのでつぎにモデルの作成を行なうこととした。

3. PERT/MANPOWER問題の数学モデルと解法

(1)すべての実行可能なパターンが求められている場合　プロジェクトに含まれる作業の数が少ない場合には、予め実行可能なパターンをすべて求めておくことは不可能ではない。いま、すべての実行可能なパターンが求められているとき、モデルの定式化は、

「制約条件　⑭ $\sum_k \mathbb{P}_k x_k = \mathbb{D},\ x_k \geq 0$ の下で、目的関数　⑮ $\lambda = \sum_k x_k$ を最小にする

ような解$\{x_k^0\}$を求めよ。ただし、Basisマトリックス$\mathbb{B}$から求められるSとS_dによって計算されるΔSは常に式⑬を満していなければならない。」のようになる。上記の定式化をみれば明らかなように、後の条件がなければLPのモデルとなっている。従って、CASE Iの場合のように後の条件の存在しないときにはシンプレックス法によって解くことができる。またCASE IIの場合においても、新しい解x_sの導入時にΔSを調べ式⑬が成立しないとき@, ⑥のいずれの場合かを判別し、⑥の場合には$x_s \equiv 0$とおいて、新しい解ベクトルを探せばよい。この場合の最適解の存在は容易に証明される。

(2)すべての実行可能なパターンが求められない場合。　プロジェクトの作業の数が多くなるとパターンの数は幾何級数的に増大する場合が多い。従って、必要最小限のパターン数でもって最適解$\{x_k^0\}$を求める方法について述べることにする。まず、準備としてシンプレックス基準をつぎのように変更しておく。いま、C_kを目的関数の初期の係数、$\overline{C}_k$を現在の係数、$\mathbb{B}$をBasisマトリックスとすると、$\overline{C}_k$は、

⑯　$\overline{C}_k = C_k - \mathbb{C}\mathbb{B}^{-1}\mathbb{P}_k$　,　$\mathbb{C} = (c_1, c_2, \cdots, c_n)$

として求められ、シンプレックス基準は、

⑰　$\overline{C}_s = \min_k \overline{C}_k \ (<0)$

として求められる。ここで、式⑰に式⑯を代入すると、

⑱　$\overline{C}_s = \min_k \{C_k - \mathbb{C}\mathbb{B}^{-1}\mathbb{P}_k < 0\} = \max_k \{\mathbb{C}\mathbb{B}^{-1}\mathbb{P}_k - \mathbb{C}_k > 0\}$

のように示されるから、シンプレックス基準をつぎのように変更することができる。すなわち、「まず、⑲ $U_s = \max_k \{\mathbb{C}\mathbb{B}^{-1}\mathbb{P}_k\}$によって$s$を求め、$U_s > C_s$ならば解の改良が可能である。$U_s \leq C_s$ならば現在の解が最適解を与える。」このように、$U_s$を用いてシンプレックス基準を用いると、つぎのような有利な点が考えられる。すなわち、1つの実行可能スケジュールが求められれば、$\mathbb{B}$および$\mathbb{B}^{-1}$が容易に計算され、新しく導入されるパターン$\mathbb{P}_s$は式⑲によって求められるので、ピボット操作をすべてのパターンを対象にして行なう必要がなくなる。このことは、作業数が多くなればなるほど、全計算量を大巾に減少せしめることになる。さて、以上に計算に用いるパターン数の減少のための工夫について示したが、つぎに、式⑲を満すような$\mathbb{P}_s$の効率的な求め方について述べることにする。式⑲を書き改めると、「補助問題；制約条件　⑳ $\sum_i r_i y_i \leq R$, $y_i = 0$ or 1の下で、目的関数、㉑ $U = \sum_i \beta_i y_i$を最大にするような$\{y_i^0\}$を求めよ。ただし、解$\{y_i\}$は、式⑦を満していることが必要である。」ここで、β_iは、$\beta = \mathbb{C}\mathbb{B}^{-1} = \mathbb{1}\cdot\mathbb{B}^{-1}$の要素であり、$\mathbb{P}_s$は、$\mathbb{P}_s = y^0$, $y^{0\prime} = (y_1^0, \cdots, y_m^0)$として求められる。この補助問題は式⑦に関する条件さえなければ、Integer Programmingの問題であり、Gomory, Balas等の解法で解くことができる。しかし、ここでは式⑦の条件があるため、Balasの開発したAdditive Algorithmに若干の変更を加えた解法を用いることにした。すなわち、計算開始時の解を$\{y_i = 0\}$とし、ある基準にもとづいて順次y_iに0あるいは1の値を与えていき、最終的に最適解を求めるという方法を用いる。この場合、各段階で$y_i = 1$と与えられているiすなわちJ_iの集合をA_kとして求め、A_kが式⑦を満しているかどうかをたえず検討する。この方法を用いれば容易に補助問題を解くことができるが、紙面の関係上この方法についての詳細と計算例は講演時に示すことにする。