

施工計画における最適ネットワークの 作成法に関する一考察

STUDY ON THE METHOD AND TECHNIQUES FOR THE OPTIMAL NETWORK PLANNING IN CONSTRUCTION PLANNING

吉川和広*・春名 攻**

By Kazuhiro Yoshikawa and Mamoru Haruna

1. 緒 言

近年の土木工事の傾向として工事の大規模化、多様化、複雑化および工事量の増大化等が顕著である。このような土木工事における量的、質的变化のもとで工事施工を合目的に遂行していくためには、土木技術の革新、とくに工事施工の分野における技術革新が必要である。工事施工の分野における近年の技術革新は、建設機械、施工技術等に関しては急速に進展してきており、大型高性能建設機械、特殊建設機械および数多くの新工法の開発などにその成果をみることができている。

一方、工事の大規模化、工事量の増大化および工事内容の多様化、複雑化の傾向は、施工管理者に対して処理すべき情報の急激な増加をもたらした。したがって、従来のように経験的あるいは数少ない試行錯誤に基づく管理技術では、管理のための複雑で多量な情報を合目的に処理することが不可能であると考えられる。合目的にかつ円滑に工事を遂行せしめるためには、建設機械や施工技術に関する技術革新とともにそれらを十分に活用するための管理技術の革新が必要である。

われわれは、現在までに現工事体制下において実行可能でかつ最も有効な施工管理システムとして、工程管理を中心とする施工管理システムに着目して研究を進めてきた。さらに、このシステムにおける工程管理手法として PERT, CPM などのネットワーク手法の導入を行なった。しかし、ネットワーク手法が装置産業のための管理手法として開発されたものであるから、装置産業とは多くの異なった管理特性を有する土木工事にそのまま適用することは困難である。したがって、土木工事の特性を十分に考慮した新しい管理手法の開発が行なわれなければならない。本研究においてはこれらの特性のうちで

もネットワーク手法を導入する工程管理システムにおいて、最も基本的であると考えられる工程ネットワークの作成、スケジュールの作成のプロセスに着目した。

土木工事の工程計画としてネットワークを作成し、スケジュールを作成する場合、そこにはさまざまな施工技術あるいは管理技術の組み合わせが可能であり、これにもなってさまざまな形の工程ネットワークが存在するはずである。しかるに、これまで土木工事にネットワーク手法が用いられた場合には、管理者の想定する工程をただ一通りのネットワークに表現し、これに基づいて PERT 計算を行なってきた。その場合ほとんどの管理者が、作成したネットワークが合目的かどうかをあまり深く検討することなく工程計画として採用してきた。このような工程計画・管理の方法が本来の管理目的を十分果たしているとはいえず、管理者の管理行為に対する検討が欠けていることは明らかである。このような管理における工程ネットワーク上の問題点を解決し、ネットワーク手法を土木工事の有効な管理技法として定着させるためには、合目的なネットワークをどのような基準に基づいて作成すべきかを明らかにして行くことが必要であると考えられる。以下においては、このような観点に立ったネットワークの作成、スケジュールの作成のプロセスについて論じることとする。

2. 作業間順序関係

(1) 土木工事における作業間順序関係

ネットワーク手法によるシステムアプローチにおいて最も基本的でかつ本質的なプロセスは、工事の工程に示される作業の流れの構造を、所要時間を特性値とするグラフとして表現することである。さらに、この作業の流れの構造の基礎となっているのは、プロジェクトに含まれる作業間の順序関係である。この場合、すべての作業

* 正会員 工博 京都大学教授 工学部土木工学教室
** 学生会員 工修 京都大学大学院工学研究科 土木工学専攻

の順序関係が一意的に定められるならばネットワークは1通りだけ作成される。たとえば、装置産業の工程においては生産の技術的な側面、あるいは機械・設備の位置的な関係から作業の流れはほとんど一意的に定められている。したがって、1通りだけの工程ネットワークを作成し、このネットワークに対して作業の所要時間を与えてスケジュールを計算した結果を工程計画として採用すればよい。もちろん、この計画に基づいて作業管理を行なっても、工程計画に対するなんらの問題も生じてこないことは明らかである。

土木工事の工程においても施工法が決定している場合には、施工技術的な側面からの作業順序はほとんど一意的に定められるはずである。しかし、実際にはこのような基本的な作業順序を与えるだけでは工事を遂行することができない。というのは、土木工事の施工に際して、まず対象となる施工区間あるいは構造物をいくつかの施工ブロックに分割し、つぎに、これらのブロックに対応する作業集合を施工技術的な順序関係に基づいて施工していくことによって順次構造物を建設していき、最終的に完成品としての施設を得るという手段がとられるからである。この場合、相異なるブロック間において、同一工程に含まれる作業に対する施工順序はすべての工程をとおして同じであってもよいし、同じでなくてもよい。

さて、ここで示した施工対象のブロック分割および相異なるブロック間の作業の施工順序は、施設、機械、要員などよりなる作業グループの配分方法によって規定される。したがって、全体工程としての作業の順序関係は、これらをどのように配分するかによって大きく変わり得るものである。同一工事に対していく通りもの工程ネットワークを作成することが可能であるということは、これらの作業グループの配分方法がいく通りも考えられるということに基づいている。

以上において土木工事の工程が、

① 基本的施工順序の決定 → ② 施工対象のブロック分割

→ ③ ブロック内作業順序決定 → ④ ブロック間の作業順序の決定

というプロセスを経て作成されることを示したが、②のブロック分割に際して、一つのブロック内で行なわれる作業量については、施工能率を最高度に発揮することのできる大きさのうちで最小のものを分割の基準とすればよい。これは、一般にブロック分割をある程度まで多くすれば一つの作業の施工量も少なくなり、したがって、所要時間も小さくなる。このため、これらの単位作業（アクティビティ）を用いてネットワークを作成し、スケジュール計算を行なうと、全体的に余裕時間（ロス）は少なくなるとともに、プロジェクトの完了時刻も小さくなる傾向がある。しかし、分割の数をあまり多くしすぎると機械の運用の面で不都合が生じたり、時間的にみ

て適正作業量とならず、かえって施工能率が極度に低下し、施工時間が増大する傾向を示すということに基づいている。

以上のことから明らかなように、分割単位もある程度までは一意的に決められることがわかる。以下において、われわれは①～③までのプロセスで定められる順序関係を施工技術的な側面から決められる先決的な順序関係であると考えて、これを技術的順序関係と呼ぶことにする。また、④のブロック間の作業順序が、施設、機械、要員などよりなる作業グループの配分方法によって決められる管理的側面を有する順序関係であることから、これを管理的順序関係と呼ぶことにする。つぎに、ネットワークプランニングからみたこれら二つの順序関係の位置づけについて述べることにする。

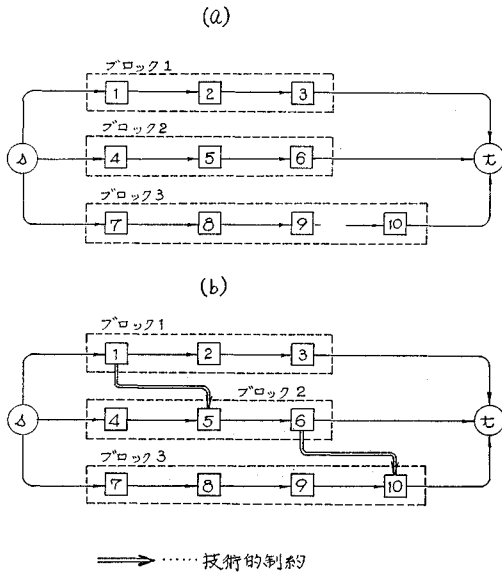
（2）技術的順序関係と管理的順序関係

前述のように、技術的順序関係は対象工事が与えられればほぼ一意的に定められる先決的な順序関係である。一方、管理的な順序関係は施設、機械、要員などの作業グループの配分という管理的な側面を有しており、土木工事においては、この点に着目してネットワークを作成して行くことがきわめて重要である。すなわち、技術的順序関係を与件として与えた場合、どのような管理的順序関係がより合目的であるかということを知ることが必要である。

a) 技術的順序関係

施工対象の分割を行なうことによって施工の単位ブロックが求められれば、各ブロックを施工するための作業順序は容易に求められる。施工の単位ブロックとしては、道路工事のように構造物に明確な区分の存在しない場合には、導入する機械の形状、寸法、性能および施工能率の関係からブロックの大きさを求めねばならないし、施工対象が橋梁のような場合には、橋台、橋脚、橋桁などのように、すでに分割されている構造物が適正な施工単位となっていることが多いので、これらをブロックとすればよい。また、各ブロックにおける単位作業を求める場合には、使用機械の機種および作業職種をもとにして決めることが重要である。

施工ブロック、および各ブロック内の作業順序が決められれば、つぎにこれらの作業のうち特定の作業の間に施工技術的な制約があり、これに基づく順序関係が存在すれば、それらを新たに付加することによってネットワークを作成することができる。たとえば、施工上の安全性からみて相異なる二つのブロックに含まれる作業が連続して行なわれるほうが望ましい場合、あらかじめ、これらの二つの作業の間に順序関係をつけておけばよい。図一1にその例を示す。



図一 技術的制約の付加

さて、作業の技術的順序関係が求められれば、これをネットワークとして表現するために、つぎのような行列を用いると便利である。すなわち、

$$P^{(T)} = (p_{ij}^{(T)})$$

$$p_{ij}^{(T)} = \begin{cases} 1, & i \leq j \text{ (作業 } i \text{ が作業 } j \text{ に直接先行している場合)} \\ 0, & \text{その他の場合。} \end{cases} \quad i, j \in J \quad \dots\dots\dots (1)$$

のように表わす。ここに、 J はプロジェクトに含まれる作業の集合である。いま、図一 (b) のようなネットワークに対して、行列 $P^{(T)}$ は 図二 のように求められる。

$$P^{(T)} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

図二 技術的順序関係 $P^{(T)}$

b) 管理的順序関係

実際の工程計画を策定する場合、前述の作業グループに対して利用可能量を与えて、これらの量的な制約の下で合目的な工程計画を作成しようとする立場と、どのような作業グループをどの程度の数量与えれば最も合目的な計画が作成されるかを求める立場とがある。さらに、このような作業グループの配分の方式に関しても、工種と作業グループとが1対1に対応する場合と、1対1に

対応しない場合が考えられる。すなわち、前者は作業の専門化の進んでいる Job shop scheduling においてよくみられるような追抜き禁止型の工程であり、後者は作業の専門化の進んでいない場合の工程である。

これらのいずれの場合に対しても工程ネットワークの作成プロセスとしては、管理的順序関係をすでに求められている技術的順序関係に合目的に付加しなければならないことには変わりがない。

いま、一つの作業工程に対応する作業グループ l に対する競合作業 (同種の作業グループを必要とする作業) 集合 $J^{(Rl)} \subseteq J$ における作業順序関係 $P^{(Rl)}$ が求められるとすると、プロジェクトにおける作業間順序関係 P はこれらの順序関係をすべて重ね合わせたものとして求められる。すなわち、

$$P = P^{(T)} + P^{(R)} \\ = P^{(T)} + \sum_l P^{(Rl)} \quad \dots\dots\dots (2)$$

ここで、 $P^{(Rl)}$ は競合作業間の順序関係を表わしており、

$$P^{(Rl)} = (p_{ij}^{(Rl)})$$

$$p_{ij}^{(Rl)} = \begin{cases} 1, & i \leq j, \\ 0, & \text{その他の場合} \end{cases} \quad i, j \in J^{(Rl)} \quad \dots\dots\dots (3)$$

また、競合作業集合は互いに共通な作業を含まないように定める。

$$J^{(Rl)} \cap J^{(Rl')} = \emptyset \text{ (空集合)} \\ (l \neq l', l, l' = 1, 2, \dots, L) \quad \dots\dots\dots (4)$$

ここで、式 (2) における演算の加法則には、

$$1+1=1, 1+0=1, 0+1=1, 0+0=0$$

を用いることにする。

c) 工事工程と作業グループの配分方法

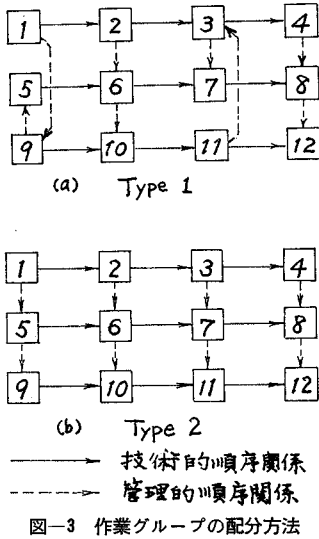
前述したように、資源配分方法に基づく工事工程の型にはつぎの2通りの場合が考えられる。

Type 1: ブロックごとに構造物を順次完成させていく方法はとらず、施工技術的順序関係に矛盾が起らないように、作業グループの配分順序を決める。この場合、作業の追抜きは許される (図三 (a))。

Type 2: ブロックごとに構造物を順次完成させていく方法で、ブロック間の施工順序にしたがって、ブロック内に含まれる作業に作業グループを配分していく方法である。これは、Job shop scheduling における作業の追抜きが禁止されている場合に対応する (図三 (b))。

Type 2 の作業グループ配分方法は、組立生産工業のように、作業の専門化が高度に進んだ工種を多く含んでいる工程の場合に適した配分方法であり、ブロック間に施工順序をつければ同時に作業順序も決まる。

Type 1 は作業の専門化があまり進んでいない作業を多く含んでいるような場合で、このため、かえって作業順序にフレキシビリティが生まれてきているようなとき



に適しており、作業ごとに作業グループの配分順序をつけることが必要である。土木工事の場合には、近年 Type 2 の工事工程が増加していく傾向にあるが、それでも大部分の工事工程は Type 1 のパターンである。したがって、まず、Type 1 を一般的な場合としてとりあげていくこととする。また、Type 2 の問題は 6. で後述することにする。

3. 追抜き自由の場合の作業間順序決定のモデル

(1) 目的関数の設定

工程管理は時間を中心とする管理であるから、実行可能でかつ工期が最小となるようにネットワークの型、すなわち作業間順序 P を決めることが必要である。したがって、以下において取り扱うモデルにおいても工期最小という目的を設定することとする。すなわち、このモデルで用いる目的関数は各種作業グループ l ($l=1, 2, \dots, L$) の配分を表わす競合作業間の順序関係のマトリックスを、 $P^{(R_1)}, P^{(R_2)}, \dots, P^{(R_L)}$ で与えると、目的関数 λ は、

$$\lambda = \lambda(P) = (P^{(T)}, P^{(R_1)}, \dots, P^{(R_L)}) \dots\dots\dots (5)$$

と表わされる。ここで $P^{(T)}$ が一定であると仮定したから目的関数 λ は上式の代わりに、

$$\lambda = \lambda(P^{(R_1)}, P^{(R_2)}, \dots, P^{(R_L)}) \dots\dots\dots (6)$$

のように書き改めることができる。したがって、プロジェクト完了時刻 λ は管理的順序関係のみの関数として表わされることになる。

(2) 制約条件

(1) ではモデルにおける変数および目的関数について述べたが、ある競合作業集合 $J^{(R)}$ に着目し、これらの作業間の順序関係、すなわちモデルの変数に対する制約条件について順次示していくこととする。

a) 連続の条件

すべての競合作業が、与えられた作業グループによって順次、連続的に施工されていくためには、一つの競合作業の施工が終了すると、それがこの作業グループにとって最終の作業でないかぎり、必ずつぎの競合作業に移って施工を行なう必要がある。これを連続の条件と呼び、順序関係 $p_{ij}^{(R)}$ を用いてあらわすと、

$$\left. \begin{array}{l} \textcircled{1} \sum_{i \in J^{(R)}} p_{ij}^{(R)} \leq 1 \\ \textcircled{2} \sum_{j \in J^{(R)}} p_{ij}^{(R)} \leq 1 \end{array} \right\} \dots\dots\dots (7)$$

ようになる。上式の ① において不等号が成立するのは、作業 $i \in J^{(R)}$ が作業グループにとって最終の作業である場合に限り、② において不等号が成立するのは、作業 $j \in J^{(R)}$ が作業グループにとって最初の作業である場合に限る。また、その他の場合には常に等号が成立する。

b) サイクル非構成の条件

作業間順序関係に基づいて定められるスケジュールが実行可能であるためには、作業間順序関係 P および $P^{(R)}$ が、サイクルを構成しないように定めることが必要である。

いま、 $L(P) = 0$ によって P がサイクルを構成しないことを表わすことにすると、常に、

$$L(P) = L(P^{(R)}) = 0 \dots\dots\dots (8)$$

が成立しなければならない。

c) 作業間順序関係の数に関する条件

つぎに、全体として求めるべき順序関係の数、すなわち 1 の値を与える変数の数に関する制約条件を求めておかねばならない。これは、施工のために与えられる作業グループの数によって変化する。以下においては、競合作業集合 $J^{(R)}$ に着目した場合の作業グループの数と、求めなければならない競合作業間の順序の数の関係について示すこととする。

まず、作業グループの数 r が与えられている場合を考える。この場合、競合作業集合 $J^{(R)}$ はそれぞれの作業グループに対応する r 個の部分作業集合 $J_t^{(R)}$ ($t=1, 2, \dots, r$) に分割される。すなわち、 $J_t^{(R)}$ は t 番目の作業グループによって施工される作業の集合を表わしており、互いに共通な作業を含むことはできない。

$J^{(R)}$ と $J_t^{(R)}$ の関係は、

$$\left. \begin{array}{l} J_1^{(R)} \cup J_2^{(R)} \cup \dots \cup J_r^{(R)} = J^{(R)} \\ J_{t_1}^{(R)} \cap J_{t_2}^{(R)} = \emptyset \text{ (空集合)} \\ (t_1, t_2 = 1, 2, \dots, r, t_1 \neq t_2) \end{array} \right\} \dots\dots\dots (9)$$

のように表わされる。

さて、 n 個の作業を含む競合作業に対して、 n 個の作

業グループを与えた場合を考えてみる。この場合、式(9)を満たすような $n (=r)$ 個の部分作業集合 $J_t^{(R)}$ ($t=1, \dots, n$) が求められるが、これらの部分作業集合 $J_t^{(R)}$ は競合作業を1つだけ含むような集合である。すなわち、

$$\left. \begin{aligned} J_t^{(R)} &= \{j_t\} \\ j_t &\in J^{(R)} \\ (t &= 1, 2, \dots, n) \end{aligned} \right\} \dots\dots\dots(10)$$

この場合には、管理的順序関係を付加する必要は全くない。

つぎに、作業グループの数が $n-1$ 個である場合を考える。この場合も、式(9)を満たすような $n-1$ 個の部分集合が求められるが、上述の場合とは異なって、ただ1つの部分集合 $J_{t^*}^{(R)}$ だけが、集合の要素として2つの競合作業 $j_l, j_k \in J^{(R)}$ を含み、他の部分集合 $J_t^{(R)}$ ($t \neq t^*$) は上と同様にただ一つの競合作業 j_t を要素として含んでいる。すなわち、

$$\left. \begin{aligned} J_{t^*}^{(R)} &= \{j_l, j_k\} \\ J_t^{(R)} &= \{j_t; t \neq l, t \neq k\} \\ j_l, j_l, j_k &\in J^{(R)} \\ (t &= 1, 2, \dots, n-1, t \neq t^*) \end{aligned} \right\} \dots\dots\dots(11)$$

この場合、付加すべき管理的順序関係は、作業 j_l と j_k の間の順序関係ただ1つであり、順序行列の要素では $p_{lk}^{(R)}=1$ あるいは $p_{kl}^{(R)}=1$ で表わされている。

作業グループの数が $r=n$ と $r=n-1$ の場合を比較すれば明かなように、1つの順序関係を付加することが1つの作業グループ数の減少と対応する。このことから一般に、 r の作業グループに対しては作業の順序関係にサイクルを生じさせないような $n-r$ 個の順序関係が必要であることを類推することができる。いま、この関係が一般に成立することを証明するためには、すでに $r=n$ および $r=n-1$ については証明されているから、一般の r および $r-1$ の場合で上述の関係が成立することを示せばよい。以下にそのことを示してみよう。

いま、 r 個の作業グループに対して、 $J_u^{(R)}$ ($u=1, 2, \dots, r$) という部分集合が求められ、さらに作業間の順序関係はサイクルを持たず、その個数は $n-r$ であるとする。つぎに、作業グループの数を1だけ減らして、 $r-1$ 個の作業グループを与えた場合を考えてみる。この場合には、

$$\left. \begin{aligned} I &= J_{u_1}^{(R)} \cup J_{u_2}^{(R)} \\ J_{u_1}^{(R)} \cap J_{u_2}^{(R)} &= \emptyset \text{ (空集合), } (u_1 \neq u_2) \end{aligned} \right\} (12)$$

を満たすような I を構成する部分作業集合 $J_{u_1}^{(R)}, J_{u_2}^{(R)}$ が存在する。また、 I は、

$$\left. \begin{aligned} I &\subseteq J^{(R)} \\ I \cap J_u^{(R)} &= \emptyset \text{ (空集合)} \\ (u &= 1, 2, \dots, r, u \neq u_1, u_2) \end{aligned} \right\} \dots\dots\dots(13)$$

をも満たす。さて、このような I に対してはただ1個の

作業グループが対応するから、 $J_{u_1}^{(R)}$ に含まれるある競合作業 j_{u_1} と $J_{u_2}^{(R)}$ に含まれるある競合作業 j_{u_2} の間には、1つの順序関係が付加されていなければならない。したがって、全体として付加される順序関係の数は1つだけ増加し、 $n-r+1$ となり、かつこのような順序関係は競合作業間にサイクルを構成しないことも明らかである。

以上のことから明かなように、 r 個の作業グループが与えられ、さらに競合作業間の順序関係にサイクルを考えないような場合には、付加すべき競合作業間の順序関係の数は $n-r$ 個であることがわかる。すなわち、このことを順序行列の要素 $p_{ij}^{(R)}$ を用いてあらわすと、

$$\sum_{i \in J^{(R)}} \sum_{j \in J^{(R)}} p_{ij}^{(R)} = n-r \dots\dots\dots(14)$$

のようになる。

(3) 定 式 化

以上述べてきたことから明かなように、本節でとりあげる作業順序決定問題は、「 N 個の作業を含むプロジェクトにおいて、作業所要時間 D_j 、作業間の技術的順序関係 $P^{(T)}$ および L 種の作業グループをおのおの r_l 組ずつ与えた場合に、最小のプロジェクト完了時刻を与えるような管理的順序関係 $P^{(R)}$ を求めよ。ただし、作業グループ l に対する競合作業集合 $J^{(R,l)}$ に含まれる作業の個数はそれぞれ n_l 個であるとする」のように表わされる。また、この問題を順序関係 P および $P^{(R,l)}$ を用いてつぎのように定式化することができる。

「制約条件

$$\left. \begin{aligned} \textcircled{1} \quad & \sum_{i \in J^{(R,l)}} p_{ij}^{(R,l)} \leq 1 && (l=1, 2, \dots, L) \\ \textcircled{2} \quad & \sum_{j \in J^{(R,l)}} p_{ij}^{(R,l)} \leq 1 && (l=1, 2, \dots, L) \\ \textcircled{3} \quad & \sum_{i \in J^{(R,l)}} \sum_{j \in J^{(R,l)}} p_{ij}^{(R,l)} = n_l - r_l && (l=1, 2, \dots, L) \\ \textcircled{4} \quad & L(P) = L(P^{(R,l)}) = 0 && (l=1, 2, \dots, L) \end{aligned} \right\} \dots\dots\dots(15)$$

の下で、目的関数

$$\lambda = \lambda(P^{(R,1)}, P^{(R,2)}, \dots, P^{(R,L)}) \dots\dots\dots(16)$$

を最小にするような $P^{(R)}$ を求めること。」

定式化から明かなように、式(16)に示した目的関数が非線形であるから、すべての変数を同時に決定するような便利な解法はない。本研究においては、DP あるいはブランチ・バウンド法のような逐次計算法を導入する解法についての考察を行なった。後述するが、DP を用いる解法は、作業グループが1種類の場合には有効な解法となりうるが、作業グループが2種類以上の場合には、DP における最適性の原理が成立するような多段階決定過程としてとらえることができず、したがって最適解も求められない。一方、ブランチ・バウンド法を用いた解法によると、2種類以上の作業グループの場合に

対しても最適解を求めることができる。以下、4、5. においてこれらの解法を示していくことにする。

4. DP を用いた解法 (作業グループが1種類の 場合)

いま、説明を容易にするために作業を適当にならべかえて、順序行列が図-4 に示すような形になるように変える。このように示すと、図の斜線部の小行列 $\bar{P}^{(R)}$ において、変数 $p_{ij}^{(R)}$ が制約条件式 (15) の下で0あるいは1の値をとるが、 $P^{(R)}$ の $\bar{P}^{(R)}$ に含まれない他の部分においては、 $p_{ij}^{(R)}$ が常に0の値をとることがわかる。

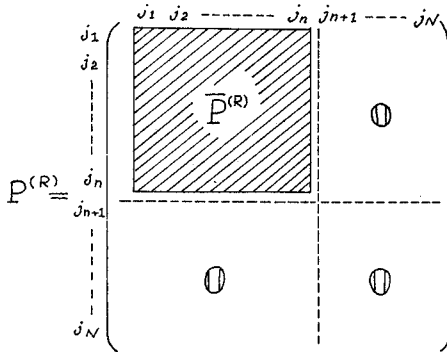


図-4 $P^{(R)}$ と $\bar{P}^{(R)}$ の関係

このように $\bar{P}^{(R)}$ を定義すると、モデルにおける実際の変数としては $\bar{P}^{(R)}$ だけを考えればよいことが明らかである。以下においては、この小行列 $\bar{P}^{(R)}$ を求めるべき変数と考えて論をすすめることにする。

いま、この小行列の列ベクトルを $\bar{Q}_k^{(R)}$ ($k=1, 2, \dots, n$) とおくと、 $\bar{Q}_k^{(R)}$ は、

$$\bar{Q}_k^{(R)} = \mathbf{0} \text{ or } \mathbf{e}_i \dots\dots\dots(17)$$

$$\left(\begin{matrix} k=1, 2, \dots, n \\ i=1, 2, \dots, n \end{matrix} \right)$$

でなければならない。ここで $\mathbf{0}$ は n 次元ゼロベクトルであり、 \mathbf{e}_i は i 番目の要素が1で、他の要素がすべて0であるような n 次元単位ベクトルを表わしている。このことは、制約条件式 (15) ① をみれば容易に理解できる。さて、式 (17) で $\bar{Q}_k^{(R)}$ を定義したが、この $\bar{Q}_k^{(R)}$ を用いると与えられた問題はつぎのように書き改められる。

「制約条件

$$\left. \begin{matrix} \sum_{k=1}^n \bar{Q}_k^{(R)} \leq \mathbf{1} \\ \sum_{k=1}^n \bar{Q}_k'^{(R)} \bar{Q}_k^{(R)} = n - r \\ L(P) = L((\bar{Q}_1^{(R)} \bar{Q}_2^{(R)} \dots \bar{Q}_n^{(R)}) = 0 \end{matrix} \right\} \dots\dots(18)$$

の下で、目的関数

$$\lambda = \lambda(\bar{Q}_1^{(R)}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_n^{(R)})$$

を最小にするような $(\bar{Q}_1^{(R)'}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_n^{(R)})$ を求めよ。」

ここで1はすべての要素が1であるような n 次元列ベクトルである。また、 $\bar{Q}_k'^{(R)}$ は $\bar{Q}_k^{(R)}$ の転置ベクトルである。

前述したように、上記の問題の解 $(\bar{Q}_1^{(R)}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_n^{(R)})$ を同時に求めることはできないので、DP の多段階決定過程としてとらえることにより解を求めていくことにする。すなわち、まず競合作業の数に等しいだけの配分ステップを設定し、各ステップにおいてはベクトル $\bar{Q}_k^{(R)}$ を一つずつ決めていくようにする。したがって、ステップ k においては、ベクトル $\bar{Q}_1^{(R)}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_k^{(R)}$ が定められていることになる。

このように配分過程を定めると、DP の最適性の原理に基づいて目的関数式 (19) はつぎのような漸化式として表わすことができる。

$$\left. \begin{matrix} \lambda_0(\emptyset) = \lambda(P^{(T)}) \\ \lambda_k(\bar{Q}_1^{(R)}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_k^{(R)}) \\ = \min_{\bar{Q}_k^{(R)}} \{ \lambda_{k-1}(\bar{Q}_1^{(R)}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_{k-1}^{(R)}) \\ + \Delta \lambda_k(\bar{Q}_k^{(R)}) \} \end{matrix} \right\} \dots\dots(20)$$

($k=1, 2, \dots, n$)

ここで、 λ_k はステップ k におけるプロジェクト完了時刻を表わし、 $\Delta \lambda_k$ は $\bar{Q}_k^{(R)}$ を定めることによって生じるプロジェクト完了時刻の増分をあらわしている。

また、 $\bar{Q}_k^{(R)}$ は制約条件、

$$\left. \begin{matrix} 0 \leq \bar{Q}_k^{(R)} \leq \mathbf{1} - \sum_{h=1}^{k-1} \bar{Q}_h^{(R)} \\ \max\{0, k-r\} \leq \bar{Q}_k'^{(R)} \cdot \bar{Q}_k^{(R)} \\ + \sum_{h=1}^{k-1} \bar{Q}_h'^{(R)} \cdot \bar{Q}_k^{(R)} \leq \min\{k, n-r\} \end{matrix} \right\} \dots\dots(21)$$

を満たし、かつ $\bar{Q}_k^{(R)}$ を求めることによって変化する作業間順序関係 P_k および $\bar{P}_k^{(R)}$ がサイクルの非構成条件、

$$L(P_k) = L(\bar{P}_k^{(R)}) = 0 \dots\dots\dots(22)$$

を満たすように定めなければならない。 $\bar{P}_k^{(R)}$ および P_k はつぎのようにして求められる。すなわち、 $\bar{P}_k^{(R)}$ 、 $P_k^{(R)}$ は次式によって表わされる。

$$\bar{P}_k^{(R)} = (\bar{Q}_1^{(R)} \bar{Q}_2^{(R)} \dots \bar{Q}_k^{(R)} \underbrace{\mathbf{0} \dots \mathbf{0}}_{n-k \text{ 個}}) \dots\dots(23)$$

$$P_k^{(R)} = (Q_1^{(R)} Q_2^{(R)} \dots Q_N^{(R)}) \dots\dots\dots(24)$$

ここで、 $Q_k^{(R)}$ は次式で示されるような N 次元列ベクトルである。

$$Q_k^{(R)} = \left(\begin{matrix} \bar{Q}_k^{(R)} \\ 0 \\ \vdots \\ 0 \end{matrix} \right) \left. \vphantom{Q_k^{(R)}} \right\} N - k \text{ 個}, (k=1, 2, \dots, N) \dots\dots\dots(25)$$

最後に、 P_k は $P^{(T)}$ の列ベクトル $Q_k^{(T)}$ と $P_k^{(R)}$ の列ベクトル $Q_k^{(R)}$ を用いれば次式によって求められる。

$$\left. \begin{aligned} P_k &= (Q_1 Q_2 \dots Q_N) \\ Q_k &= Q_k^{(T)} + Q_k^{(R)} \end{aligned} \right\} \dots\dots\dots(26)$$

また、 $\Delta\lambda(\bar{Q}_k^{(R)})$ は $\bar{Q}_k^{(R)}$ の要素のなかで1の値をもつ要素を $p_{ijk}^{(R)}$ とすると、ステップ $k-1$ において求められる最早開始時刻 $ES_{jk}^{(k-1)}$ 、最早完了時刻 $EF_i^{(k-1)}$ 、およびトータル・フロード $TF_{jk}^{(k-1)}$ を用いて容易に求められる。すなわち、

$$\left. \begin{aligned} \Delta\lambda(\bar{Q}_k^{(R)}) &= \max\{0, EF_i^{(k-1)} - ES_{jk}^{(k-1)} \\ &\quad - TF_{jk}^{(k-1)}\}, \bar{Q}_k^{(R)} = e_i \text{ のとき} \\ &0, \bar{Q}_k^{(R)} = \mathbf{0} \text{ のとき} \end{aligned} \right\} \dots\dots\dots(27)$$

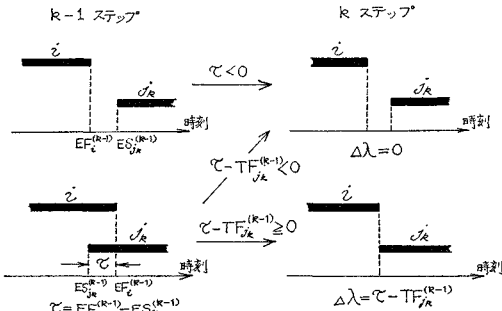


図-5 順序関係の付加によるスケジュールの変化

ここで、上式で用いるスケジュール値は P_{k-1} および所要時間 D_j を与えれば容易に計算できる。スケジュール計算には、PERT/TIME の計算法を用いずに、プロジェクト型でトポロジカルオーダリングなしのスケジュール計算法を用いる。また、式 (20), (27) から明らかのように、ステップ k においてはスケジュール計算を、すべての組み合わせに対して行なう必要はなく、つぎのステップにおいて $\Delta\lambda$ を求めるためにスケジュール値が必要となる場合にだけ行なえばよい。

以上の計算法を用いて、プロジェクト完了時刻 λ_k ($k=1, 2, \dots, n$) を順次求めていき、

$$\lambda_{\text{optimal}} = \min\{\lambda_n(\bar{Q}_1^{(R)}, \bar{Q}_2^{(R)}, \dots, \bar{Q}_n^{(R)})\} \dots\dots\dots(28)$$

を与える順序行列 $\bar{P}^{(R)}$ を求めるとこれが最適解を与える。

以上で作業グループが1種類の場合の解法を示したが、与えられる作業グループが2種類以上になると同様の方法では最適解を求めることができない。このことを示すために、作業グループが2種類の場合をとりあげ、これらに対応する二つの競合作業集合 $J^{(R_1)}$, $J^{(R_2)}$ に含まれる作業間の順序を求める場合を考えることとする。さて、制約条件に関しては、 $P^{(R_1)}$ と $P^{(R_2)}$ が互いに独立であることが式 (15) より明らかである。しかし、式(16)

の目的関数に関しては、与えられた技術的順序関係 $P^{(T)}$ によって2つの競合作業集合 $J^{(R_1)}$, $J^{(R_2)}$ が互いに関係づけられているために、求めるべき $P^{(R_1)}$ と $P^{(R_2)}$ は独立ではなくなる。したがって、各競合作業集合に対する最適順序関係を求める場合には、1つの配分ステップにおいて、相異なる競合作業集合に含まれる作業間の順序関係をもすべて同時に評価することができなければ、DPの最適性の原理は成立しないことになる。このことより、作業グループが2種類以上の場合には、上に述べた方法を拡張しても他の競合作業集合に含まれる作業間の順序関係を評価することはできないから、DPによって最適解が求められないことは明らかである。また、このような評価が可能であるような配分過程を求めることも不可能である。

5. ブランチ・バウンド法を用いた解法

(1) ブランチ・バウンド法

組み合わせ問題においては、与えられた制約条件の下で、解ベクトル X に対して目的関数 $F(X)$ を最小(あるいは最大)にするように整数値(離散値)を割り当てることが多い。ブランチ・バウンド法を用いる場合には、目的関数 $F(X)$ に対する特別な条件はなく、種々の形をとることができる。しかし、ただ一つの条件として必ず有限な解をもつことがあげられる。この条件を満たすならば、計算量の多少、計算効率のちがいはあっても、他の計算法が適用できないような場合に対しても必ず最適解を求めることができる便利な手法である。

さて、具体的な解法を示すにあたって、まずブランチ・バウンド法の一般的な方法について述べることにする。

a) 解空間

いま、つぎのような最適化問題 M を考えてみる。

$$\left. \begin{aligned} &M; \text{制約条件} \\ &G_1^{(0)}(X) \geq 0 \\ &\vdots \\ &G_m^{(0)}(X) \geq 0 \\ &X \in R^{(0)} \end{aligned} \right\} \dots\dots\dots(29)$$

の下で

$$f^{(0)} = F^{(0)}(X) \dots\dots\dots(30)$$

を最小にするような、最適解 X_0 を求めよ。」

ここで、 $R^{(0)}$ は解空間を表わし、 n 次元ユークリッド空間のうち、離散値の集合に対する部分空間である。すなわち、 R_D を離散値の集合とすると、

$$R^{(0)} = \underbrace{R_D \times R_D \times \dots \times R_D}_{n \text{ 個}} \dots\dots\dots(31)$$

ここで、 X はベクトル (x_1, x_2, \dots, x_n) を表わす。

問題 M を直接解くことが困難である場合、解空間 $R^{(0)}$ をいくつかの部分解空間に分割し、それらに対して問題 $M^{(i)}$ を考える。

$$\left. \begin{array}{l} \text{「}M^{(i)}\text{; 制約条件} \\ G_j^{(i)}(X^{(i)}) \geq 0 \\ \vdots \\ G_g^{(i)}(X^{(i)}) \geq 0 \\ X^{(i)} \in R^{(i)} \end{array} \right\} \dots\dots\dots(32)$$

の下で

$$f^{(i)} = F^{(i)}(X^{(i)}) \dots\dots\dots(33)$$

を最小にするような最適解 $X_0^{(i)}$ を求めよ。」

ここで、 $R^{(i)}$ は問題 $M^{(i)}$ に対応する解空間を表わしている。

いま、

$$\bigcup_i R^{(i)} = R^{(0)} \dots\dots\dots(34)$$

を満たすように、 $R^{(i)}$ を設定すると、

$$\left. \begin{array}{l} F^{(0)}(X_0) = \min \{ F^{(i)}(X_0^{(i)}); X_0^{(i)} \in R^{(i)} \} \\ F^{(0)}(X^{(i)}) = F^{(i)}(X^{(i)}), X \in R^{(i)} \end{array} \right\} \dots\dots\dots(35)$$

が成立することは明らかである。

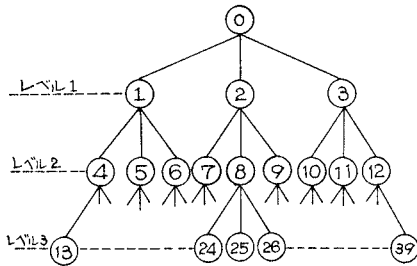


図-6 トリー図 ($n=3$)

いま、解空間の $R^{(i)}$ への分割に対して図に示すようなノード・ブランチをもつトリー表現を用いる。トリーにおける始点ノード 0 には、主問題 M に対する解空間 $R^{(0)}$ を対応させ、その他のノード j には、つぎのような方法で、1つ以上の解ベクトルを含む部分解空間 $R^{(j)}$ を対応させるようにする。すなわち、ノード i における解空間 $R^{(i)}$ を次式の関係を満たすような部分解空間 $R^{(j)}$ の集合に分割する。

$$\left. \begin{array}{l} \bigcup_{j \in N_i} R^{(j)} = R^{(i)} \\ R^{(j)} \cap R^{(k)} = \emptyset \text{ (空集合)} \\ j \in N_i, k \in N_i \end{array} \right\} \dots\dots\dots(36)$$

ここで、 N_i はノード i からのブランチで求められるノードの集合を表わす。

また、解空間の分割にあたっては、 $R^{(i)}, R^{(j)}$ に含まれる解ベクトルの間に、

$$\left. \begin{array}{l} \textcircled{1} \quad x_l^{(j)} = x_l^{(i)}, \quad x_l^{(i)} \in \mathcal{Q}_i \\ \textcircled{2} \quad x_l^{(j)} = a_l, \quad x_l^{(i)} \in \mathcal{Q}_i \\ \textcircled{3} \quad x_l^{(j)} = x, \quad x \in R_D \text{ かつ } x_l^{(i)} \in \mathcal{Q}_i, l \neq t \end{array} \right\} \dots\dots\dots(37)$$

が成立するように $R^{(j)}$ を求める。ここで、 \mathcal{Q}_i はノード i における解空間 $R^{(i)}$ において一定値をとる $x_l^{(i)}$ の集合であり、 l は特殊な場合を除いては、1回のブランチに対して、ただ1つだけ求める。また、式 ③ 右辺の x は R_D に含まれる限りどのような値をもとりうる。さらに、 a_l は R_D においてとり得る離散値のうちの1つの値であるから、 N_i に含まれるノードの数は、とり得る離散値の数だけ存在する。ここで、始点ノード 0 からノード i が求められるまでのブランチ数をレベル d_i で表わすと、 \mathcal{Q}_i に含まれる要素の個数は d_i と一致する。また、 $X^{(i)}$ のすべての要素が一定値をとるのは、一般にはレベル n におけるノードであることも明らかである。

b) 下界値

以上において、解空間を部分解空間に分割する方法について述べたが、1つの変数のとり得る離散値の数を m とすれば、すべての解ベクトルを求めるためには、最大 m^n 個のノードを作成しなければならない。したがって、変数の個数が極端に小さい場合を除いては、すべての解を求めることは計算量の点で困難かつ非常に不経済である。このため、なんらかの評価基準を設定することによってノードの数をできるだけ減少させ、計算を実行可能にしなければならない。以下に述べようとしている下界値の概念はこの目的に対して大変有効な手段であると考えられる。

さて、われわれは与えられた主問題 M が解けなかったり、解くことがむずかしいような場合には、まず、よりやさしい補助問題 AM を考え、これを解いて主問題の解との関係を調べる場合が多い。そして求められた関係に基づいて、主問題を解くためにはどのような補助問題を作成すればよいか、あるいはどのような補助問題の組み合わせを考えればよいかなどを求める。ブランチ・バウンド法においても、先述の問題 $M^{(i)}$ に対する補助問題 $AM^{(i)}$ を作って主問題 M に対する最適解を求めるという方法を用いる。以下においては、ブランチ・バウンド法で用いる補助問題についての説明を行なうこととする。

さて、補助問題 $AM^{(i)}$ を作成するにあたっては、その目的関数は以下に示す2つの条件式を満たすことが必要である。

$$\left. \begin{array}{l} \bar{F}^{(i)}(X^{(i)}) \leq \bar{F}^{(j)}(X^{(j)}), j \in N_i \\ \bar{F}^{(i)}(X_0^{(i)}) \leq F^{(i)}(X^{(i)}) \end{array} \right\} \dots\dots\dots(38)$$

一般には、このような条件式を満足させるような補助問題 $AM^{(i)}$ を作成することはあまり困難ではないが、 $AM^{(i)}$ を作成できないような場合にはこの方法を適用することはできない。

補助問題 $AM^{(i)}$ をつぎのように示す。

$$\left. \begin{array}{l} \text{「}AM^{(i)}\text{; 制約条件} \\ \bar{G}_1^{(i)}(X^{(i)}) \geq 0 \\ \vdots \\ \bar{G}_g^{(i)}(X^{(i)}) \geq 0 \\ X^{(i)} \in R^{(i)} \end{array} \right\} \dots\dots\dots(39)$$

の下で

$$\bar{f}^{(i)} = \bar{F}^{(i)}(X^{(i)}) \dots\dots\dots(40)$$

を最小にするような最適解 $\bar{X}_0^{(i)}$ を求めよ。」

このように、 $\bar{X}_0^{(i)}$ を補助問題 $AM^{(i)}$ の最適解とすると、式 (38) より、

$$\bar{F}^{(i)}(\bar{X}_0^{(i)}) \leq F^{(i)}(X_0^{(i)}) \dots\dots\dots(41)$$

が成立することが明らかである。

以上のことから、補助問題 $AM^{(i)}$ の最適解の与える目的関数値は問題 $M^{(i)}$ の目的関数の下界値となっておりと同時に、ノード $j \in N_i$ に対応する補助問題 $AM^{(j)}$ の目的関数の下界値となっていることがわかる。すなわち、

$$\bar{F}^{(i)}(\bar{X}_0^{(i)}) \leq \bar{F}^{(j)}(\bar{X}_0^{(j)}) \leq F^{(j)}(X^{(j)}), j \in N_i \dots\dots\dots(42)$$

以下においては、問題 $M^{(i)}$ の目的関数の下界値を $LB^{(i)}$ で表わし、ブランチの結果得られる各ノードおよびそれ以後のブランチで求められるノードに対して下界値を計算し、問題 $M^{(i)}$ の解の評価に用いることとする。

$$LB^{(i)} = \bar{F}^{(i)}(\bar{X}_0^{(i)}) \dots\dots\dots(43)$$

いま、計算の途中で主問題 M の許容解であるような補助問題の最適解 $\bar{X}_0^{(i)}$ が少なくとも一つ求まっているとすると、これらのうちで最小の目的関数値をもつものを求め、これを $LB^{(*)}$ とおく。いま、ノード i における下界値が

$$LB^{(*)} \leq LB^{(i)} \dots\dots\dots(44)$$

を満たす場合には、式 (42) よりノード $j \in N_i$ の下界値も

$$LB^{(*)} \leq LB^{(j)}, j \in N_i \dots\dots\dots(45)$$

を満たす。したがって、ノード i から以後のブランチを行なう必要はない。以上のことから明らかなように、このような下界値を用いれば、結果的に全体のノードの数、すなわち、考慮しなければならない補助問題の数を大幅に減少させることが可能となる。また、下界値 $LB^{(*)}$ は、補助問題 $AM^{(i)}$ の最適解が偶然主問題 M の許容解となる場合を除いては、一般にレベル n において求められる。

c) ブランチの手順

b) においては、現在求められているノード i およびノード i からのブランチによって求められるノード j に対する評価に、下界値 $LB^{(i)}$ および $LB^{(*)}$ を用いてノードの数を大幅に減少しうることを示した。しかし、どのような手順でノードを求めていくかについては、まったくふれなかった。実際に主問題の解を求めるときに

は、ブランチを行なうための合理的でかつシステムティックなルールを考えなければならない。現在までいくつかのルールが考えられているが、ここで用いるルールを示すとつぎのように表わされる。

ルール：

① $LB^{(*)}$ が求められていないとき、現在行ないうるすべてのブランチに対して、 $LB^{(i)}$ を計算するとともに、

$$LB^{(s)} = \min_i \{LB^{(i)}\} \dots\dots\dots(46)$$

によって、ノード s を求める。 s が求められれば、すべてのノード $j \in N_s$ をブランチによって求めるが、 $i \neq s$ からのブランチは行なわない。以上の手順をノード 0 から始めて $LB^{(*)}$ が求められるまで順次繰り返す。

② $LB^{(*)}$ が求められているとき、式 (46) の代わりに

$$LB^{(s)} = \min_i \{LB^{(i)}; LB^{(i)} < LB^{(*)}\} \dots\dots(47)$$

を用いて s を求めるが、ブランチに際しては ① で示した方法と同様の方法を用いる。

③ 計算途中において、主問題の許容解となっているような補助問題の最適解がいくつか求められるが、そのような解が求められるたびに現在の $LB^{(*)}$ と比較し、たえず両者の小さいものを $LB^{(*)}$ としておきかえる。

④ 計算の途中で、式 (47) を満たすようなノード s が存在しなくなれば、その時点での下界値 $LB^{(*)}$ を与えているノードにおける解が最適解を与える。

(2) 解 法

以上においては、ブランチ・バウンド法の一般的な方法を示した。つぎに、本モデルに適用する場合の問題 $M^{(i)}$ 、補助問題 $AM^{(i)}$ およびブランチの手順などを、作業グループが 1 種の場合を例にとって述べることにする。いま、 \bar{Q}_k の属する集合を E とすると、 E は式 (17) から、

$$E = \{e_i; i = 1, 2, \dots, n\} \cup \mathbf{0} \dots\dots\dots(48)$$

と表わされるから、始点ノード 0 における解空間 $R^{(0)}$ は、

$$\begin{aligned} R^{(0)} &= \underbrace{E \times E \times \dots \times E}_{n \text{ 個}} \\ &= \{(\bar{Q}_1^{(0)} \dots \bar{Q}_k^{(0)} \dots \bar{Q}_n^{(0)}); \bar{Q}_k^{(0)} \in E\} \dots\dots(49) \end{aligned}$$

で表わされる。つぎに、トリーの各ノードにおける解空間について示すことにする。

ノード 0 からのブランチによって求められるノード i における解空間を $R^{(i)}$ ($i \in N_0$) とすると、

$$R^{(i)} = \underbrace{E \times E \times \dots \times E}_{n-1 \text{ 個}} = \{(\bar{Q}_1^{(i)} \dots \bar{Q}_n^{(i)}); \bar{Q}_t^{(i)} = a_t, \bar{Q}_k^{(i)} \in E, k \neq t\} \dots\dots\dots (50)$$

ここで、 a_t は E に属する要素 ($\mathbf{0}$ or e_i) のうちの 1 つである。すなわち、 $\bar{Q}_k^{(i)}$ は

$$\bar{Q}_k^{(i)} = \left\{ \begin{array}{l} a_t, k=t \\ \bar{Q}_k^{(0)}, k=1, 2, \dots, n, k \neq t \end{array} \right\} \dots\dots\dots (51)$$

で表わされる。ここでノード i において一定ベクトルを与えるベクトルの集合を Ω_i とおくと、 $\Omega_0 = \emptyset$ (空集合)、 $\Omega_i = \{\bar{Q}_i\}$ である。また、ブランチによって求められる解空間の間には、次式のような関係が成立する。

$$\left. \begin{array}{l} R^{(i)} \cap R^{(j)} = \emptyset, i \neq j, i, j \in N_0 \\ \bigcup_{i \in N_0} R^{(i)} = R^{(0)} \end{array} \right\} \dots\dots\dots (52)$$

つぎに、ノード f における解空間 $R^{(f)}$ が求められている場合に、ノード g ($g \in N_f$) の解空間がどのようになるかを求めてみる。 $R^{(g)}$ は、ノード g のレベルを d_g とすると、 E の $(n-d_g)$ 次の直積空間として表わされる。また、 $\bar{Q}_k^{(f)}$ と $\bar{Q}_k^{(g)}$ の間にはつぎのような関係が成立する。

$$\bar{Q}_k^{(g)} = \left\{ \begin{array}{l} a_t, \bar{Q}_i^{(f)} \in \Omega_f, k=t \\ \bar{Q}_k^{(f)}, \bar{Q}_k^{(f)} \in \Omega_f, k \neq t \\ a_k, Q_k^{(f)} \in \Omega_f \end{array} \right\} \dots\dots\dots (53)$$

また、すべての $g \in N_f$ における解空間の間には、式 (52) と同様の関係が成立する。

つぎに、各ノード i における問題 $M^{(i)}$ はつぎのように書ける。ただし、目的関数 $F^{(i)}$ は完了時刻を表わし、ネットワーク手法では記号 λ のほうが一般的であるためここでは λ を用いた。

「 $M^{(i)}; P_i^{(R)}$ を与えた場合、制約条件

$$\left. \begin{array}{l} \sum_{\bar{Q}_k \in \Omega_i} \bar{Q}_k \bar{Q}_k = n - r - \sum_{\bar{Q}_k \in \Omega_i} \bar{Q}_i \bar{Q}_i \\ \mathbf{0} \leq \sum_{\bar{Q}_k \in \Omega_i} \bar{Q}_k \leq \mathbf{1} - \sum_{\bar{Q}_k \in \Omega_i} \bar{Q}_k \\ L(P) = L(P^{(R)}) = 0 \end{array} \right\} \dots\dots\dots (54)$$

の下で、

$$\lambda = \lambda(\{\bar{Q}_k; \bar{Q}_k \in \Omega_i\}) \dots\dots\dots (55)$$

を最小にするような $\{\bar{Q}_k; \bar{Q}_k \in \Omega_i\}$ を求めよ。」

ここで、 N をプロジェクトに含まれる作業の総数とすると、 $P_i^{(R)}$ は、ノード i において求められている $\bar{Q}_k \in \Omega_i$ を用いてつぎのように求められる。すなわち、

$$\left. \begin{array}{l} P_i^{(R)} = (Q_1^{(i)} Q_2^{(i)} \dots Q_N^{(i)}) \\ \text{また、} Q_k^{(i)} \text{ は、} \\ \bar{Q}_k^{(i)} = \left(\begin{array}{c} \bar{Q}_k^{(i)} \\ 0 \\ \vdots \\ 0 \end{array} \right) \end{array} \right\} N-n \text{ 個} \dots\dots\dots (56)$$

であり、 $Q_k^{(i)}$ は $k > n$ のとき常に $\mathbf{0}$ である。

さて、この問題に対応する補助問題 $AM^{(i)}$ をつぎのように作成する。

「 $AM^{(i)}; P_i^{(R)}$ を与えた場合、

$$\lambda(\Omega_i) = \lambda(\{\bar{Q}_k^{(i)}; \bar{Q}_k^{(i)} \in \Omega_i\}) \dots\dots\dots (57)$$

の下界値を求めよ。」

このような $AM^{(i)}$ を用いると上記の補助問題の解すなわち下界値は $P_i^{(R)}$ を与えた場合のプロジェクト完了時刻となっていることは明らかである。すなわち、 $\lambda = \lambda(\{\bar{Q}_k; \bar{Q}_k \in \Omega_i\})$ の下界値は $\bar{Q}_k \in \Omega_i$ がすべてゼロベクトルの場合である。このことは、始点ノード 0 においても、一般のノード i においても全く同様である。このことから下界値が $\bar{P}_i^{(R)}$ の関数、すなわち Ω_i の関数となっていることがわかる。したがって、前述したブランチ操作による解空間の分割は、各ノード g において求められている一定ベクトルの集合 Ω_g ($g=1, 2, \dots$) を用いてつぎのように書きかえることができる。

すなわち、始点ノード 0 における補助問題の解としては、 $\Omega_0 = \emptyset$ であるから、

$$(\bar{Q}_1^{(0)} \dots \bar{Q}_n^{(0)}) = (\mathbf{0} \dots \mathbf{0}) \dots\dots\dots (58)$$

また、 $i \in N_0$ における補助問題の解、すなわち下界値は、 $(\bar{Q}_1^{(i)} \dots \bar{Q}_n^{(i)})$ の要素のうち一つのベクトル $\bar{Q}_t^{(i)}$ だけを一定ベクトル $a_t \in E$ に置きかえることによって求められる。ここで、すべての a_t ($t=1, \dots, n$) の間には、 $a_t \neq a_{t'} (t \neq t', t, t'=1, 2, \dots, n)$ が成立する。すなわち、

$$\bar{Q}_k^{(i)} = \left\{ \begin{array}{l} a_t, k=t \\ \mathbf{0}, k \neq t, k=1, 2, \dots, n \end{array} \right\} \dots\dots\dots (59)$$

つぎにノード f とノード $g \in N_f$ の下界値の関係について述べる。

上述のことから、 $\lambda(\Omega_f) = \lambda(P_f^{(R)})$ と $\lambda(\Omega_g) = \lambda(P_g^{(R)})$ との間につぎの関係が成立していることが明らかである。

$$\left. \begin{array}{l} \lambda(\Omega_g) = \lambda(\Omega_f) + \Delta\lambda(\Omega_g - \Omega_f) \\ \Delta\lambda(\Omega_g - \Omega_f) = \left\{ \begin{array}{l} \max(EF_i^{(f)} - ES_j^{(f)}) \\ -TF_j^{(f)}, 0 \end{array} \right\}, \Omega_g - \Omega_f = e_i \\ \mathbf{0}, \Omega_g - \Omega_f = \mathbf{0} \end{array} \right\} \dots\dots\dots (60)$$

したがって、1回のブランチによって求められるノードにおいては、すべてのノードにおける補助問題を解いて下界値を計算する必要はなく、上述の関係式に基づいて、下界値を計算することができる。ただし、つぎにブランチを行なおうとするノード (ブランチ・ノード) においてはスケジュール計算は必要である (式 (60) 右辺参照)。

以上の議論から一つの実行可能解に対する解ベクトルの集合 $\Omega = \{\bar{Q}_k; k=1, 2, \dots, n\}$ に対応する下界値 $\lambda(\Omega)$ と、 $\lambda(\Omega_g)$ 、 $\Omega_g \subseteq \Omega$ との間にはつぎの関係が成立するこ

とが導かれる。

$$\lambda(\mathcal{Q}) \geq \lambda(\mathcal{Q}_g; \mathcal{Q}_g \subseteq \mathcal{Q}) \dots\dots\dots (61)$$

これらのことから、このモデルが(1)において述べたブランチ・バウンド法を適用するための条件を満たしていることは明らかである。

(3) 作業グループが2種類以上の場合への拡張

(2)においては、作業グループが1種類の場合についての解法を示したが、作業グループが2種類以上の場合に対する解法も、前述のブランチの方法を若干修正することによって容易に求められるので、このことを以下において簡単に述べることにする。

いま、L種の作業グループを投入する一般の場合を考えてみよう。この場合、実行可能な解すなわち全順序関係が求められる最終レベルをnとすると、nは、

$$n = \sum_{i=1}^L n_i \dots\dots\dots (62)$$

によって表わされる。また、集合Qをレベルnにおいて、定値ベクトルとして求められている解ベクトル全体の集合とすると、Qは、

$$\mathcal{Q} = \{ \mathcal{Q}^{(1)}, \mathcal{Q}^{(2)}, \dots, \mathcal{Q}^{(L)} \} \\ \mathcal{Q}^{(l)} = \{ \bar{Q}_k^{(Rl)}; \bar{Q}_k^{(Rl)} = a_k, k=1, 2, \dots, n_l \} \dots\dots\dots (63)$$

のように表わされる。

上記のように、最終のレベルn, QおよびQ^(l)を定めると、前述した解法と異なってくるのは、問題M⁽ⁿ⁾の作成にすることがらだけである。

いま、計算途中でブランチを行なうための始点ノードとしてノードsが選択されたとすると、ノードsのレベルd_sが、

$$\sum_{i=1}^{m-1} n_i \leq d_s < \sum_{i=1}^m n_i \dots\dots\dots (64)$$

を満たすならば、ノードsにおける問題M^(s)はつぎのようになる。

$$\left[\begin{aligned} &M^{(s)}; P_s^{(R)} \text{ を与えた場合、制約条件、} \\ &\sum_{\bar{Q}_k^{(Rp)} \in \mathcal{Q}_s^{(l)}} \bar{Q}_k^{(Rl)}, \bar{Q}_k^{(Rl)} \\ &= n_l - r_l - \sum_{\bar{Q}_k^{(Rp)} \in \mathcal{Q}_s^{(l)}} \bar{Q}_k^{(Rl)'} \cdot \bar{Q}_k^{(Rl)} \\ &0 \leq \sum_{\bar{Q}_k^{(Rp)} \in \mathcal{Q}_s^{(l)}} \bar{Q}_k^{(Rl)} \leq 1 - \sum_{\bar{Q}_k^{(Rp)} \in \mathcal{Q}_s^{(l)}} \bar{Q}_k^{(Rl)} \\ &(l=m, m+1, \dots, L) \\ &L(P) = L(P^{(R)}) = 0 \end{aligned} \right] \dots\dots\dots (65)$$

の下で、

$$\lambda = \lambda(\{ \bar{Q}_k^{(Rl)}; \bar{Q}_k^{(Rl)} \in \mathcal{Q}_s^{(l)} \} | \\ = m, m+1, \dots, L) \dots\dots\dots (66)$$

を最小にするような{Q_k^(Rl); Q_k^(Rl) ∈ Q_s^(l)}, (l=m, m+1, ..., L)を求めよ。」

ここで、Q_s^(l)はノードsにおいて各作業グループl(L

= m, m+1, ..., L)に対応する解ベクトルのうち、すでに定値ベクトルとして求められている解ベクトルの集合であり、ここでは明らかに、

$$\mathcal{Q}_s^{(m+1)} = \dots = \mathcal{Q}_s^{(L)} = \emptyset \text{ (空集合)} \dots\dots\dots (66)$$

である。

前述したように、実際の計算で用いる補助問題においては、Q_s^(l)に含まれないような解ベクトルをすべてゼロベクトル0とおかねばならないから、

$$\bar{Q}_k^{(R(m+1))} = 0, \bar{Q}_k^{(R(m+2))} = 0, \dots, \bar{Q}_k^{(RL)} = 0 \dots\dots (67)$$

となり、すでに始点ノードにおいてすべての解ベクトルが0として与えられているので、ここではベクトルQ_k^(Rm) ∈ Q_s^(m)のみを対象として考えればよいことが容易に理解される。ここで用いる解法においても、ノードsにおけるブランチに際して、どの作業グループに対する制約条件にしたがう必要があるかを、式(64)によって定めれば、その他の部分は前述した解法と同様でよいことがわかる。また、ブランチの評価に用いるLB^(s)も、全作業グループの配分を定めたレベルnにおいて初めて求められるのであるが、これもnを式(62)によって求めるようにすれば(2)の方法と全く同じでよいことが明らかである。

作業グループが2種以上の一般の場合について、計算のためのフローチャートを示すと図-7のようになる。

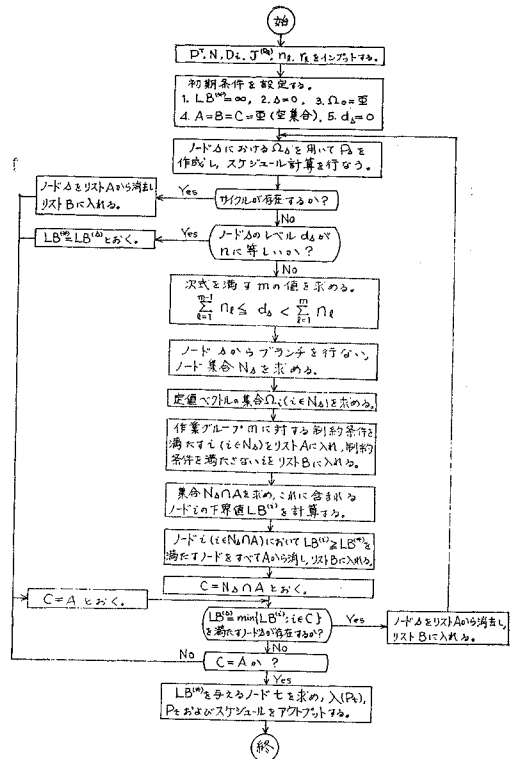


図-7 ブランチ・バウンド法による解法のフローチャート (Type 1)

6. 追抜き禁止の場合の作業間順序決定のモデル

先述のように、近年の土木工事の大規模化、複雑化にともなって、施工現場においては種々の機械化が促進されてきた。さらに、労働力の不足にともなって鉄筋工・とび工などの特殊技能労働者の確保が困難となりつつある。このような現状を反映して工事の各工程においてはそれぞれの作業員・機械などの専門化・規格化が進行してきている。そして、この専従化している作業員や機械などは施工現場の移動とともに移動していく傾向にある。さらに、大規模な道路工事を始めとして建設機械を多種投入しなければならないような土木工事においては、ほとんど同じ工程が繰り返されることによって工事が進められている。

このような場合には、繰り返しの基本となる工程を先決的な順序関係として施工ブロック間の順序関係を求めることにより、全体工事の作業順序関係を定めることができる。

(1) 追抜き禁止型の順序づけ問題の導入

このように、施工技術の面から定められる先決的な基本工程をもち、作業効率の上昇という点から同一工程に対する作業グループの専従化が行なわれている場合、作業間の順序で追抜きが禁止される。この場合の解法として作業グループが2種の場合には Johnson が提案した追抜き禁止型の順序づけ問題として取り扱うことができる。

すなわち、図-8 に示すような基本工程を与件として

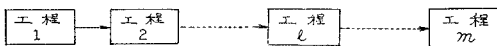


図-8 基本工程

考える。つぎに、施工対象を n 個のブロック $(1, 2, \dots, j, \dots, n)$ に分割すると、問題は完了時刻 λ を最小にするようなブロック間の順序関係を求めることである。いま、ブロック j において工程 l を遂行するために要する時間を t_{lj} として表わすと、 j ブロックにおいて工程 l が行なわれない場合には、 $t_{lj}=0$ とおけば、すべてのブロックに対して同じ工程が与えられると仮定してもなんら矛盾はない。

このように Johnson 型の順序づけ問題を導入すれば、作業間順序関係を決定するモデルを作成することができるが、全工程数 $m > 2$ の場合に対しては Johnson の提案した解法では解くことができない。以下においては $m \geq 3$ の場合に対して、ブランチ・バウンド法を用いた解法を示すこととするが、具体的な解法を述べるまえ

に、まずここでとりあげる主問題 M 、問題 $M^{(g)}$ および補助問題 $AM^{(g)}$ を明らかにしておくこととする。

いま、ブロック間の順序関係を Π とすると、 Π は

$$\left. \begin{aligned} \Pi &= (j_1, j_2, \dots, j_n) \\ j_k &\in J, k=1, 2, \dots, n \end{aligned} \right\} \dots\dots\dots (68)$$

のような順列で表わすことができる。ここで、 J は分割の結果として求めたブロックのインデックスの集合である。このように順列 Π によってブロック間の順序関係を表わすと、モデルにおける目的関数 λ は、

$$\lambda = \lambda(\Pi) \dots\dots\dots (69)$$

のように Π の関数として表わすことができる。したがって、主問題 M をつぎのように表わすことができる。

「 M ; 目的関数 $\lambda = \lambda(\Pi)$ を最小にするような最適解 Π_0 を求めよ。ただし、各基本工程の作業グループを各ブロックへ配分する場合の順序はすべて Π にしたがうものとする。」

この主問題 M は 4. で示した問題と同様にこのままでは解くことがむずかしいが、若干の工夫を加え、これにブランチ・バウンド法を適用すれば比較的容易にかつシステマティックに解くことができる。

(2) ブランチ・バウンド法を適用した解法

ここで用いる解法においても、前述したブランチ操作を適用することとするが、計算途中で求められる各ノード g において求められる解、すなわちブロック間順序関係 $\Pi^{(g)}$ は、

$$\left. \begin{aligned} \Pi^{(g)} &= (\Pi_h^{(g)}, \bar{\Pi}_h^{(g)}) \\ h &= d_g \end{aligned} \right\} \dots\dots\dots (70)$$

のように表わされる。ここで、 $\Pi_h^{(g)}$ はレベル $d_g (=h)$ においてすでに求められている h 個のブロック間の順序関係、すなわちブロック j_1, j_2, \dots, j_h を要素とする順列を表わしている。ここで、 $\Pi_h^{(g)}$ を、

$$\left. \begin{aligned} \Pi_h^{(g)} &= (j_1^{(g)}, j_2^{(g)}, \dots, j_h^{(g)}) \\ j_k^{(g)} &\in J_h^{(g)} \\ k &= 1, 2, \dots, h \end{aligned} \right\} \dots\dots\dots (71)$$

で表わす。ここで $J_h^{(g)}$ は順列 $\Pi_h^{(g)}$ に含まれているブロックのインデックス集合を表わしている。また、 $\bar{\Pi}_h^{(g)}$ は、

$$\left. \begin{aligned} \bar{\Pi}_h^{(g)} &= (j_{h+1}^{(g)}, j_{h+2}^{(g)}, \dots, j_n^{(g)}) \\ j_k^{(g)} &\in J - J_h^{(g)} \\ k &= h+1, \dots, n \end{aligned} \right\} \dots\dots\dots (72)$$

として表わされるが、ノード g においては $\bar{\Pi}_h^{(g)}$ を確定していないので、実際には $n-h$ 個のブロック $\{j \in J - J_h^{(g)}\}$ を要素とする任意の順列である。

したがって、ノード g における問題 $M^{(g)}$ はつぎのように示される。

「 $M^{(g)}$; $\Pi_h^{(g)}$ を与えた場合、目的関数 $\lambda^{(g)} = \lambda^{(g)}(\bar{\Pi}_h^{(g)})$ を最小にするような最適解を求めよ。」

つぎに、問題 $M^{(g)}$ に対する補助問題 $AM^{(g)}$ は、「 $AM^{(g)}; \Pi_h^{(g)}$ を与えた場合、 $\lambda^{(g)}$ の下界値 $LB^{(g)}$ を求めよ。」

として表わされる。

以上で、 $M, M^{(g)}$ および $AM^{(g)}$ について述べてきたが、解法における計算に必要な $AM^{(g)}$ の最適解すなわち $LB^{(g)}$ は次式によって求められる。

$$LB^{(g)} = LB(\Pi_h^{(g)}) = \max_{l \in I} \left[\text{Time } M_l(\Pi_h^{(g)}) + \sum_{j \in J_h^{(g)}} t_{lj} + \min_{j \in J_h^{(g)}} \left\{ \sum_{k=l+1}^n t_{kj} \right\} \right] \quad \dots\dots\dots(73)$$

ここで、 I は工程のインデックスの集合である。また、 $l=n$ のときは、右辺 [] 内の第3項を0とする。さて、上式 [] 内第1項の $\text{Time } M_l(\Pi_h^{(g)})$ は、順列 $\Pi_h^{(g)}$ に含まれる h 個のブロックの最後のブロック $j_h^{(g)}$ において工程 l の作業が完了する時刻を表わしており、この時刻はスケジュールを解析することより容易に計算される。すなわち、

$$\left. \begin{aligned} \text{Time } M_0(\Pi_h^{(g)}) &= 0, \text{ Time } M_l(\Pi_0^{(g)}) = 0, \\ \text{Time } M_l(\Pi_h^{(g)}) &= t_{lj} + \max \{ \text{Time } M_{l-1}(\Pi_h^{(g)}), \\ &\quad \text{Time } M_l(\Pi_{h-1}^{(f)}) \} \quad g \in N_f \end{aligned} \right\} \quad \dots\dots\dots(74)$$

ここで、 $\Pi_{h-1}^{(f)}$ はノード g の始点ノード f で求められている順列であり、 $\Pi_h^{(g)}$ と $\Pi_h^{(f)}$ の間には、

$$\left. \begin{aligned} \Pi_h^{(g)} &= (\Pi_{h-1}^{(f)}, j_h^{(g)}) \\ g \in N_f \text{ かつ } j_h^{(g)} &= J_h^{(g)} - J_{h-1}^{(f)} \end{aligned} \right\} \quad \dots\dots\dots(75)$$

なる関係が成立する。

以上のようにすれば、ノード g における下界値 $LB^{(g)}$ が求められるが、式 (73) で求められる $LB^{(g)}$ は、レベル h の増加に対して非減少関数である。したがって、前述した適用条件を満足させるので、解法としてブランチ・バウンド法を用いることが可能となる。また、実際の解法においては補助問題 $AM^{(g)}$ の最適解で主問題の実行可能解となるような解は、レベル n のノードを求めなくてもレベル $n-1$ のノードにおいて求められる。すなわち、順列 Π の自由度が $n-1$ であるため、 $n-1$ 個のブロックの間の順序関係が求められれば、残りのブロックは自動的に定まってくる。このことを下界値で表わすと、

$$LB(\Pi_n^{(g)}) = LB(\Pi_{n-1}^{(f)}), \quad g \in N_f \quad \dots\dots\dots(76)$$

のように表わされる。したがって計算効率上昇のためにブランチの最終レベルを $n-1$ とすることができる。

本モデルに対するブランチ・バウンド法による解法は図-9 に示すフローチャートのようにになる。

7. 土木工事工程計画への適用例

(1) 追抜き自由の場合のモデルの適用例

3. および 4., 5. で述べた追抜き自由の場合のモデルを国道 171 号線の橋梁下部工事の工程計画の作成に適用してみることにする。橋梁下部の構造物の断面は図-10 に示すとおりである。この図より明らかなように施工対象となる構造物は、橋台 2 基と橋脚 6 基である。したがって、これらを 図-11 のように 8 個の施工ブロックに分割する。つぎに、橋台、橋脚の施工順序を求めると 図-12 に示すとおりである。実際工事においてはこれらの作業のほかに、準備工や工事のための仮道取付け（右岸、左岸とも）および跡片付けなどの作業が必要であるから、これらを付け加えることによって、 図-13 および表-1 に示すような技術的順序に基づくネットワークおよびインプットデータを求めた。本工事は河川によって 2 つの施工区分に分割されているため、作業グループも右岸、左岸の 2 組を同時に導入することとした。

また、最近の土木工事における著しい傾向として、全工事費に対するの機械費用が占める割合は飛躍的に増加しており、その合理的な運用が望まれている。また、工事に含まれる作業において重機械類を含む作業グループの割合は多くなってきている。したがって、本適用例においては、これらの機械・作業員からなる作業グループとして、上述の傾向の最も著しくかつスケジュールへ

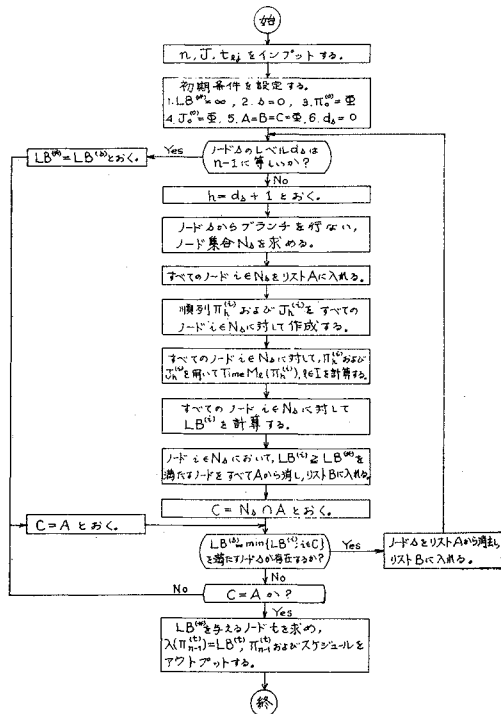


図-9 ブランチ・バウンド法による解法のフローチャート (Type 2)

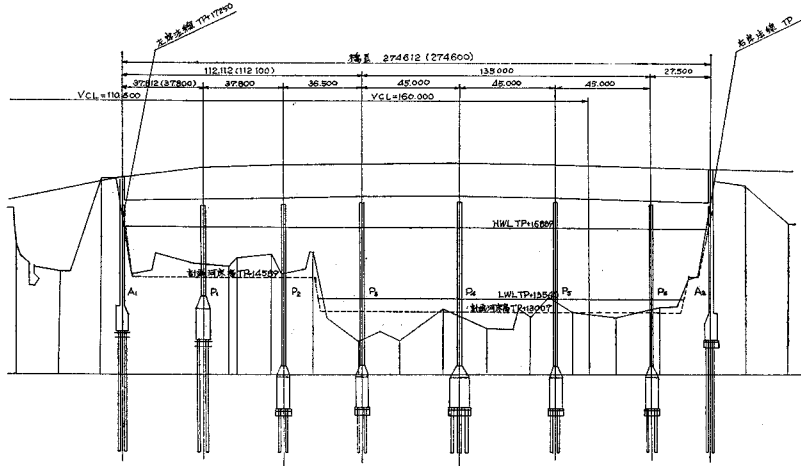


図-10 断面図

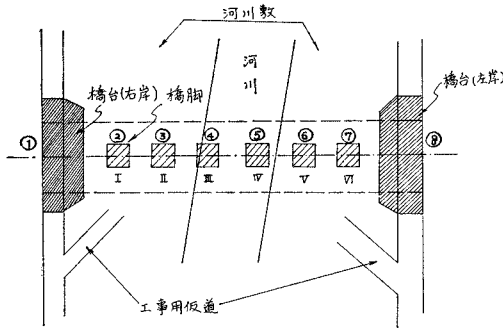
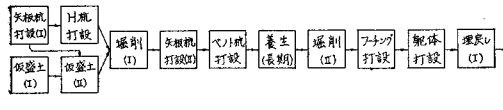
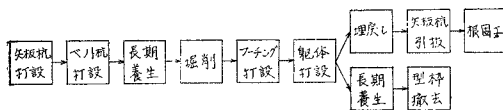


図-11 施工ブロック (①~⑧)



(a) 橋台



(b) 橋脚
図-12 施工順序

の影響の強いと思われるベント杭打設作業グループおよびパイロ打設作業グループに着目することにした。パイロハンマーおよび作業要員からなるグループは右岸，左岸側に対して各2組を導入することとした。競合作業は右岸側 {2, 4, 7, 14, 17, 18, 26, 33, 37, 44, 48, 55}，左岸側 {60, 67, 71, 78, 82, 89, 93, 95, 98, 105, 108, 109} である。また，ベント杭打設機および作業要員からなるグループは右岸，左岸側で各1組導入することと

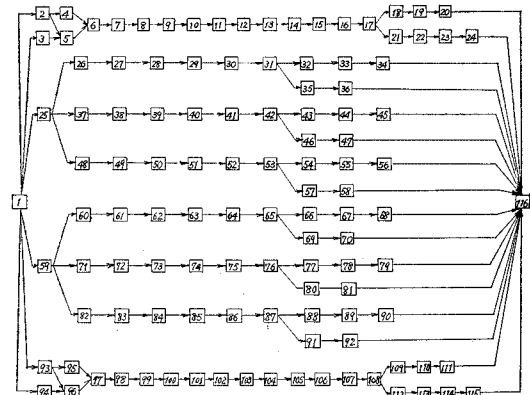


図-13 技術的順序関係に基づくネットワーク

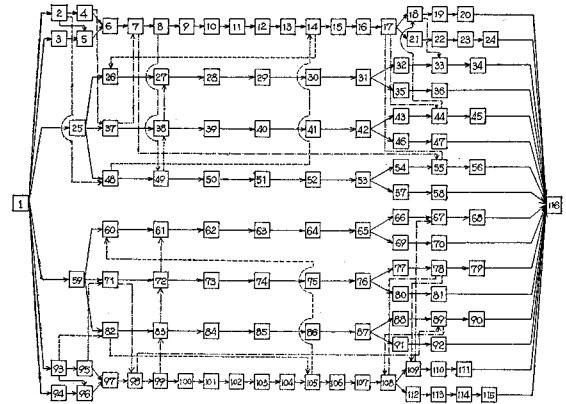


図-14 工程ネットワーク

し，競合作業は右岸側 {8, 27, 38, 49}，左岸側 {61, 72, 83, 99} である。

また，解法としては 5. で述べたブランチ・バウンド法を用いた解法を適用し，計算には京都大学大型電子計算機 FACOM 230-60 を利用した。計算の結果，図-14

表-1 インプットデータ

作業	作業名	所要時間	作業	作業名	所要時間	作業	作業名	所要時間	作業	作業名	所要時間		
1	準備工	15	30	脚	フーチング打設	15	59	左岸工事用仮道取付	15	88	脚	埋戻し	4
2	矢板杭打設(I)	17	31	脚	躯体打設	30	60	矢板杭打設	8	89	脚	矢板杭引抜	8
3	仮盛土(I)	15	32	脚	埋戻し	2	61	ベノト杭打設	10	90	脚	固工	6
4	H杭打設	2	33	(I)	矢板杭引抜	8	62	長期養生	7	91	脚	長期養生	14
5	仮盛土(II)	6	34	脚	根固工	8	63	掘削	5	92	脚	型わく撤去	2
6	橋掘削(I)	5	35	脚	長期養生	14	64	フーチング打設	15	93	脚	矢板杭打設(I)	17
7	矢板杭打設(II)	12	36	脚	型わく撤去	2	65	躯体打設	30	94	脚	仮盛土(I)	8
8	ベノト杭打設	10	37	脚	矢板杭打設	8	66	埋戻し	2	95	脚	H杭打設	2
9	長期養生	7	38	脚	ベノト杭打設	10	67	矢板杭引抜	8	96	脚	仮盛土(II)	6
10	掘削(II)	7	39	脚	長期養生	7	68	根固工	10	97	脚	掘削(I)	3
11	フーチング打設	12	40	脚	掘削	6	69	長期養生	14	98	脚	矢板杭打設(II)	6
12	躯体打設	20	41	脚	フーチング打設	15	70	型わく撤去	2	99	脚	ベノト杭打設	10
13	埋戻し(I)	3	42	脚	躯体打設	30	71	矢板杭打設	9	100	脚	長期養生	7
14	矢板引抜(I)	12	43	脚	埋戻し	2	72	ベノト杭打設	10	101	脚	掘削(II)	7
15	埋戻し(II)	3	44	(II)	矢板杭引抜	8	73	長期養生	7	102	脚	フーチング打設	12
16	仮盛土除去	14	45	脚	根固工	8	74	掘削	8	103	脚	躯体打設	20
17	矢板引抜(II)	16	46	脚	長期養生	14	75	フーチング打設	15	104	脚	埋戻し(I)	3
18	松木杭打設	4	47	脚	型わく撤去	2	76	躯体打設	30	105	脚	矢板杭引抜(I)	7
19	護岸工	45	48	脚	矢板杭打設	10	77	埋戻し	3	106	脚	埋戻し(II)	7
20	埋戻し(III)	3	49	脚	ベノト杭打設	12	78	矢板杭引抜	8	107	脚	仮盛土除去	7
21	擁壁打設	32	50	脚	長期養生	7	79	根固工	6	108	脚	矢板杭引抜(II)	17
22	埋戻し(IV)	8	51	脚	掘削	7	80	長期養生	14	109	脚	松木杭打設	4
23	床版打設	7	52	脚	フーチング打設	15	81	型わく撤去	2	110	脚	護岸工	31
24	長期養生	14	53	脚	躯体打設	30	82	矢板杭打設	9	111	脚	埋戻し(III)	3
25	右岸工事用仮道取付	20	54	脚	埋戻し	2	83	ベノト杭打設	10	112	脚	擁壁打設	32
26	矢板杭打設	8	55	脚	矢板杭引抜	10	84	長期養生	7	113	脚	埋戻し(IV)	8
27	ベノト杭打設	10	56	(III)	根固工	10	85	掘削	8	114	脚	床版打設	7
28	長期養生	7	57	脚	長期養生	14	86	フーチング打設	15	115	脚	長期養生	14
29	橋掘削	5	58	脚	型わく撤去	2	87	躯体打設	21	116	脚	工事跡片付	5

(所要時間の単位：日)

に示すようなネットワーク(作業順序関係)が得られた。そのスケジュールは表-2に示すようになった。

(2) 追抜き禁止の場合のモデルの適用例

本モデルの適用に際して、同じような単純な工程の繰り返しの多い道路よう壁工事を対象にとりあげた。さて、本モデル適用の対象となる工事の平面図および断面図は図-15に示すとおりである。道路よう壁工事のうち、基本工程として、工程1としてベノト杭打設、工程2として床掘工、工程3として基礎工、工程4として躯体工をとりあげた(図-16)。

また、施工現場を10のブロックに分割した。つまり、前述のモデルにおいて $m=4, n=10$ の場合の順序づけ問題として解いた。これらに対応する所要時間 t_{ij} の値を表-3に示した。

さらに、工事現場平面図とブロックとの対照表を図-17および表-4に示した。このブロック分割において、ただ一つのブロックだけは、ベノト杭打設工程の所要時間が0と表わされているが、このブロックは重力式のブロックであってベノト杭は使用されていない。このように基本工程に欠けている工程が含まれている場合でも、その工程の所要時間を0とおくことによって順序づ

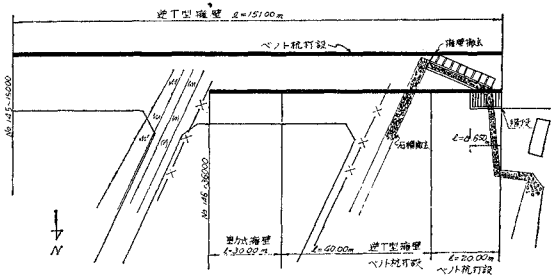


図-15 (a) 平面図

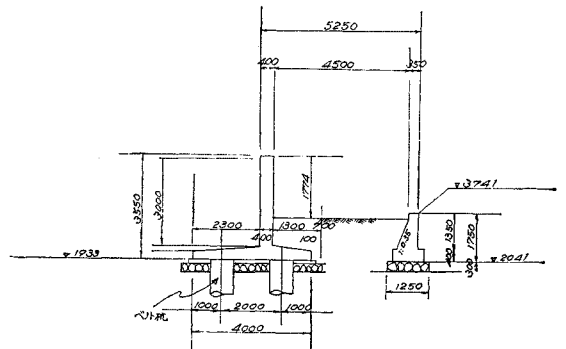


図-15 (b) 断面図

表-2 スケジュール表

作業	所 時	要 間	最早開 始時刻	最早完 了時刻	最遅開 始時刻	最遅完 了時刻	フリ ー フ ロ ー ト	ト ー タ ル フ ロ ー ト	作 業	所 時	要 間	最早開 始時刻	最早完 了時刻	最遅開 始時刻	最遅完 了時刻	フリ ー フ ロ ー ト	ト ー タ ル フ ロ ー ト
i	D_i	ES_i	EF_i	LS_i	LF_i	FF_i	TF_i	i	D_i	ES_i	EF_i	LS_i	LF_i	FF_i	TF_i		
1	15	0	15	0	15	0	0	59	15	15	30	34	49	2	19		
2	17	15	32	15	32	0	0	60	8	115	123	130	138	0	15		
3	15	15	30	17	32	2	2	61	10	123	133	138	148	0	15		
4	2	32	34	33	35	1	1	62	7	133	140	148	155	0	15		
5	6	32	38	32	38	0	0	63	5	140	145	155	160	0	15		
6	5	38	43	38	43	0	0	64	15	145	160	160	175	0	15		
7	12	43	55	43	55	0	0	65	30	160	190	175	205	0	15		
8	10	55	65	55	65	0	0	66	2	190	192	205	207	0	15		
9	7	65	72	70	77	0	5	67	8	192	200	207	215	0	15		
10	7	72	79	77	84	0	5	68	10	200	210	215	225	15	15		
11	12	79	91	84	96	0	5	69	14	190	204	209	223	0	19		
12	20	91	111	96	116	0	5	70	2	204	206	223	225	19	19		
13	3	111	114	116	119	0	5	71	9	34	43	49	58	0	15		
14	12	114	126	119	131	0	5	72	10	69	79	105	115	0	36		
15	3	126	129	131	134	0	5	73	7	79	86	115	122	0	36		
16	14	129	143	134	148	5	5	74	8	86	94	122	130	0	36		
17	16	148	164	148	164	0	0	75	15	94	109	130	145	0	36		
18	4	172	176	173	177	0	1	76	30	109	139	145	175	0	36		
19	45	176	221	177	222	0	1	77	3	139	142	175	178	7	36		
20	3	221	224	222	225	1	1	78	9	149	158	178	187	0	29		
21	32	164	196	164	196	0	0	79	6	158	164	219	225	61	61		
22	8	196	204	196	204	0	0	80	14	139	153	209	223	0	70		
23	7	204	211	204	211	0	0	81	2	153	155	223	225	70	70		
24	14	211	225	211	225	0	0	82	9	32	41	65	74	18	33		
25	20	15	35	15	35	0	0	83	10	59	69	74	84	0	15		
26	8	126	134	132	140	0	6	84	7	69	76	84	91	0	15		
27	10	134	144	140	150	0	6	85	8	76	84	91	99	0	15		
28	7	144	151	150	157	0	6	86	15	84	99	99	114	0	15		
29	5	151	156	157	162	0	6	87	21	99	120	114	135	0	15		
30	15	156	171	162	177	0	6	88	4	120	124	135	139	0	15		
31	30	171	201	177	207	0	6	89	8	124	132	139	147	0	15		
32	2	201	203	207	209	0	6	90	6	132	138	219	225	87	87		
33	8	203	211	209	217	0	6	91	14	120	134	209	223	0	89		
34	8	211	219	217	225	6	6	92	2	134	136	223	225	89	89		
35	14	201	215	209	223	0	8	93	17	15	32	30	47	0	15		
36	2	215	217	223	225	8	8	94	8	15	23	41	49	9	26		
37	8	35	43	35	43	0	0	95	2	32	34	47	49	0	15		
38	10	77	87	95	105	0	18	96	6	32	38	49	55	0	17		
39	7	87	94	105	112	0	18	97	3	38	41	55	58	2	17		
40	6	94	100	112	118	0	18	98	6	43	49	58	64	0	15		
41	15	100	115	118	133	0	18	99	10	49	59	64	74	0	15		
42	30	115	145	133	163	0	18	100	7	59	66	74	81	0	15		
43	2	145	147	163	165	17	18	101	7	66	73	81	88	0	15		
44	8	164	172	165	173	0	1	102	12	73	85	88	100	0	15		
45	8	172	180	217	225	45	45	103	20	85	105	100	120	0	15		
46	14	145	159	209	223	0	64	104	3	105	108	120	123	0	15		
47	2	159	161	223	225	64	64	105	7	108	115	123	130	0	15		
48	10	35	45	55	65	20	20	106	7	115	122	133	140	0	18		
49	12	65	77	65	77	0	0	107	7	122	129	140	147	3	18		
50	7	77	84	77	84	0	0	108	17	132	149	147	164	0	15		
51	7	84	91	84	91	0	0	109	4	158	162	187	191	0	29		
52	15	91	106	91	106	0	0	110	31	162	193	191	222	0	29		
53	30	106	136	106	136	0	0	111	3	193	196	222	225	29	29		
54	2	136	138	136	138	0	0	112	32	149	181	164	196	0	15		
55	10	138	148	138	148	0	0	113	8	181	189	196	204	0	15		
56	10	148	158	215	225	67	67	114	7	189	196	204	211	0	15		
57	14	136	150	209	223	0	73	115	14	196	210	211	225	15	15		
58	2	150	152	223	225	73	73	116	5	225	230	225	230	0	0		

表-3 ブロックと構造物の対照表

ブロック <i>j</i>	1	2	3	4	5	6	7	8	9	10
構造物番号	ベノト 5~7	ベノト 14~15	ベノト 11~13	ベノト 8~10	重力式 構築	ベノト 4'~6'	ベノト 3'	ベノト 1'~2'	ベノト 3~4	ベノト 1~2

表-4 インプットデータ

工程 <i>l</i>	ブロック <i>j</i>										
	1	2	3	4	5	6	7	8	9	10	
A	1	12	12	18	20	0	18	8	20	12	12
B	2	6	5	6	6	2	6	4	6	5	5
C	3	5	4	5	5	1	5	2	5	4	4
D	4	34	31	34	34	8	34	21	30	28	26

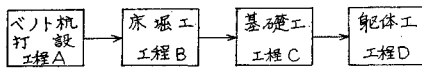


図-16 工事の基本工程

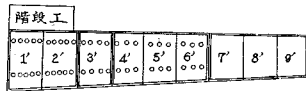


図-17 現場平面図と構造物番号

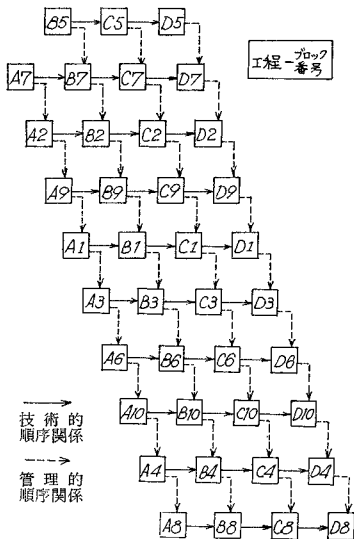


図-18 工程ネットワーク

け問題として解くことが可能であることは以前に示した。これらのインプットデータに基づいて前述の電子計算機を用いて計算した結果、 $\Pi = (5, 7, 2, 9, 1, 3, 6, 10, 4, 8)$ あるいは $\Pi = (5, 7, 2, 9, 1, 3, 6, 10, 8, 4)$ の2通りの最適施工順序が求められ、最小完了時間（総所要日

数）は286日と算定された。

以上の方法によって、施工ブロックの順序づけができれば、ネットワークにおける各アクティビティの順序関係が一意的に定められたことになる。したがって、これらのプロジェクトはすぐさま 図-18 に示すようなネットワークに変換することができる。

8. 結 言

以上においては、土木工事の工程ネットワークの基礎となっている作業間順序関係の多様性をとりあげ、どのような作業順序関係が合目的であるか、また、どのようにして合目的なネットワークを作成するかについて考察を進めてきた。すなわち、土木工事の工程が、

- ① 基本的施工順序の決定 → ② 施工対象のブロック分割 →
- ③ ブロック内作業順序決定 → ④ ブロック間の作業順序の決定

というプロセスを経ることによって求められることを明らかにし、①, ②, ③のプロセスで求められる作業間順序関係に適正な検討さえ加えればほぼ一意的に求める技術的な順序関係であると定めた。したがって、計画者は管理的側面すなわち作業グループの合理的な運用、施工条件の検討という立場から ④ で求める管理的順序関係を定めることが必要であることを示した。さらに、本研究においては、これを施工機械を中心とする作業グループのみを対象としてとりあつかい、完了時間最小を目的とした場合の最適な管理的作業順序を求めるためのモデルを作成し、合わせて解法を示した。モデルとしては、一般の土木工事において見られるような工程間の追抜きを許す場合および作業の専門化のより進んだ追抜き自由の場合の工程、すなわち土木工事の将来に予想される姿として、Job shop scheduling で取り扱われるような Johnson 型の追抜き禁止の場合の2通りを示した。

今後、残された研究課題としては作業グループと適正な施工単位との対応を作業研究、時間研究等とおして求めることであり、これらの研究に基づいて最適な施工ブロック決定法の提案が可能となるであろう。さらに、工程ネットワーク作成に関しても、特定の作業グループのみに着目して行なうだけではなく、できるだけ多くの作業グループやその他の管理要素を取り入れて評価を行なっていくようにすることが必要であろう。

最後に、本研究中で、適用例にとりあげた2つの土木

工事における資料の収集に関して協力していただいた建設省近畿地建第2阪神国道工事事務所の皆様、建設コンサルタツ協会大阪支部施工管理分科会の委員諸氏に対して厚く感謝いたします。また、電子計算機による計算作業に助力をいただいた京都大学谷添（現住友建設勤務）、細見、笹嶋の各氏に対して感謝いたします。

参 考 文 献

- 1) J.F. Muth and G.L. Thompson : Industrial Scheduling, Prentice-Hall, New York, 1963.
- 2) P.P. Schoderbek : Management Systems, John Wiley & Sons Inc., New York, 1967.
- 3) D.R. Ford Jr. and D.R. Fulkerson : Flows in Networks, Princeton University Press, 1962.
- 4) H. Rubey and W.W. Milner : Construction and Professional Management, Macmillan Co., New York, 1966.
- 5) A.S. Manne : On the Job Shop Scheduling Problem, JORSA, Vol. 8, 1960.
- 6) E. Ignall and L. Schrage : Application of the Branch and Bound Technique to Some Flow Shop Scheduling Problems, JORSA, Vol. 13, 1965.
- 7) J.D.C. Little, K.G. Murty, D.W. Sweeney and C. Karrel : An Algorithm for the Travelling Salesman Problem, JORSA, Vol. 13, 1965.
- 8) R.A. Dudek and O.F. Teuton Jr. : Development of M-Stage Decision Rule for Scheduling n Jobs through Machines, JORSA, Vol. 12, 1964.
- 9) S.E. Lass : PERT Time Calculation without Topological Ordering, Comm. ACM, Vol. 8, 1965.
- 10) 吉川和広 : 土木計画と OR, 丸善, 昭 44.
- 11) 吉川和広・春名 攻 : ネットワーク手法による 施工計画のシステムアプローチに関する研究—CPM 計算の簡便化—, 土木学会論文集, 第 151 号, 昭 43.
- 12) 吉川和広・春名 攻 : 施工計画システムへの確率 PERT の導入に関する研究, 土木学会論文報告集, 第 179 号, 昭 45-7.

(1970. 4. 27・受付)