

PERT における Path 概念の一般化について†

—— ネットワーク技法の新しい視点 ——

荒 木 睦 彦*

§ 0. あらすじ

(1) 全体を通じ、現行の作業中心のネットワークに対して、パスを中心にしたネットワークを考察の対象としている。

(2) ネットワーク技法を日程の計画と管理の2局面に分け、プロジェクトの開始結合点からネットワーク上の任意の結合点に到る最長パスによって日程計画を行ない、逆にネットワーク上の任意の結合点からプロジェクトの終了結合点に到る最長パスによって日程管理を行なおうとするものである。

(3) 主に使用される概念は、現行のものが最早、最遅結合点時刻とトータルフロートであるのに対し、ここでは計画局面において最早結合点時刻とフリー・フロート、管理局面において最遅結合点時刻とフリー・フロートを用いている。

(4) このような考え方をとると、PERT の従来の往復計算は片面計算だけで間に合うことになる。しかもフロート概念の取扱いなどは現在のそれよりも厳密になるといえる。

(5) これらの考え方は、現行 PERT のそれと対立するものではなく、むしろそのより有効な利用を狙ったものである。特に PERT の取扱いにおいて最も困難であるといわれる手計算による日程管理には、この考え方はかなり有効であろうと考える。

§ 1. まえがき

PERT (Program Evaluation and Review Technique) の考え方の最も特徴的な点は、プロジェクトを個別作業の単位にバラバラに分解し、この個別作業を基本要素としてそれらを論理的な順序および相互関係にもとづいたアロー・ダイアグラムの型に組み立て、これによって日程の計画と管理を行なおうとしたことにある。

しかし少し観点を变えて PERT を単なる作業の集合としてではなく、プロジェクトの開始から終了に至る作業の系列 (パス) の集合として眺めてみると、従来あまりふれられていない PERT ネットワークの別の側面がうかがい上ってくるように思われる。

† 1967年3月30日受理, 1966年11月11日秋季研究発表会にて梗概発表。

* 清水建設研究所

このような観点から PERT をとりあげてみると、まず手計算の手間が従来の $1/2$ 以下になり、従来面倒であったネットワークの『フォロー・アップ』が非常に容易となる。また従来かなり曖昧であったパスと作業の関係およびそれらとフロートの関係も明確になってくることがわかる。しかし、このような見方は現行の PERT の考え方を否定するものではなく、むしろそれらをさらに一般的な形に展開しようとする試みの一つである。

これらの考え方は、近い将来刊行予定の建築学会編『ネットワークによる工程計画と管理方式の指針』のなかに組み込まれる予定であるが、『学会』の『指針』という性格からいって新しい概念の全般的な展開は困難であるため、この小論によって一応一貫した展開を試みたものである。

ここでは現行のネットワーク技法の知識は既存のものとして、用語、記号、計算方法などについて特に説明を加えていない。したがってそれらについては PERT の一般的な参考文献によらねたい。

なおこの小論は日本 O R 学会秋季研究発表会に前後して清水建設研究所報、第 8 号に発表した同題名の論文の一部に若干の手を加えてリアレンジしたものである。

この小論の展開方法は私自身のものであり、それによって生ずる誤りや問題点の責任の一切が筆者にあることはもち論であるが、ここでの考え方の多くは諸先学の秀れた労作に依存している。なかでも 2 点間最長パスの考え方は電力中央研究所中川友康氏の後述文献、フロートとパスの関係に関する諸定理は、竹中工務店・開発本部・小早川洋太郎氏によって見出されたものなどを利用させていただいている。心からの感謝をささげたい。

§ 2. ネットワーク上の最長パスとその所要工期について

ここでは PERT ネットワーク上の任意の 2 つの結合点 (Node) を結ぶパス (Path) のうち、時間的に最も長くなるパスをその 2 点間の最長パスとよぶことにする。

さてプロジェクトの開始結合点を s 、終了結合点を t 、ネットワーク上の任意の結合点を i もしくは k 、任意の 2 点間、たとえば i, k 間の最長パスを $(i) \Rightarrow (k)$ 、また最長パス (i, k) の所要工期を Δ_{ik} で表わしてみると、通常のアロー・ダイアグラムは図-1 のような単純化されたパスのネットワークで表現することができる。

さて図-1において、結合点 s から t に到る所要工期を考えてみると式(1)の関係がなりたつ。

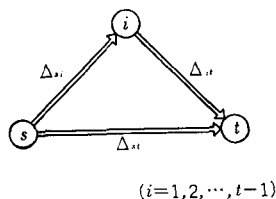


図-1 単純化されたパスのネットワーク [1]

$$(1) \quad A_{si} + A_{it} \leq A_{st}$$

パス (s, i, t) の余裕時間を ϕ' で表わすと、上式はさらに次の式(2)のように変形される。

$$(2) \quad A_{si} + A_{it} + \phi' = A_{st}$$

$$\phi' \geq 0$$

この場合の ϕ' は、結合点 i におけるスラック $(=LT_i - ET_i)$ に等しい。

さてこれらの計算を図-2の簡単な設例について試みてみよう。

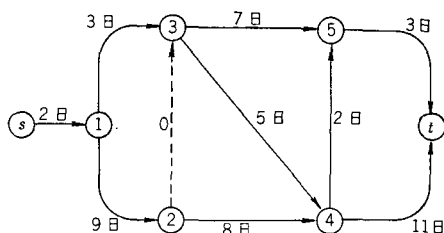


図-2 簡単な設例

a. A_{si} の計算

結合点 s の最早結合点時刻 (ET_s) を 0 とすると、

$$A_{s1} = ET_s + D_{s1} = 0 + 2 = 2 \text{ 日}$$

$$A_{s2} = A_{s1} + D_{12} = 2 + 9 = 11$$

$$A_{s3} = \max \left(\begin{array}{l} A_{s1} + D_{13} = 2 + 3 = 5 \\ A_{s2} + D_{23} = 11 + 0 = 11 \end{array} \right) = 11$$

$$(3) \quad A_{si} = \max_{\substack{(h,i) \in P \\ h < i}} [A_{sh} + D_{hi}]$$

$$(3)' \quad = \max [ET_h + D_{hi} - ET_s]$$

$$(3)'' \quad = ET_i - ET_s$$

となる。

したがって、 $ET_s = 0$ とすれば、 A_{si} は ET_i に一致する。

b. A_{it} の計算

要求工期を λ とし、計算工期は要求工期に一致する、つまり $A_{it} = \lambda$ としてみよう。

ある結合点 i についてプロジェクトの完成までに残されている工期のうち最大のものを結合点 i の残工期といい、 RT_i で表わすことにする。

結合点 i の最遅結合点時刻を LT_i とすれば、それらの関係を当然

$$(4) \quad RT_i = \lambda - LT_i$$

である。

最終結合点 t の残工期 (RT_t) を 0 とすると、

$$A_{st} = RT_t + D_{st} = 0 + 3 = 3 \text{ 日}$$

$$A_{st} = M_{ax} \left(\left. \begin{array}{l} RT_t + D_{st} = 0 + 11 = 11 \\ A_{st} + D_{st} = 3 + 2 = 5 \end{array} \right\} 11 > 5 \right) = 11$$

$$(5) \quad A_{it} = M_{ax} \left[A_{jt} + D_{ij} \right]_{\substack{(i,j) \in P \\ i < j}}$$

$$(5)' \quad = M_{ax} [RT_j + D_{ij} - RT_t]$$

$$(5)'' \quad = RT_t - RT_t$$

となる。式(4)の関係を導入すると上式はさらに、

$$(6) \quad A_{it} = LT_t - LT_i$$

$$(6)' \quad = M_{ax} [LT_t - LT_j + D_{ij}]$$

$$(6)'' \quad = LT_t - M_{in} [LT_j - D_{ij}]$$

という形に表わすことができる。

c. ϕ' の計算

式(2)に式(3)'', 式(6)を代入すると、

$$A_{st} = ET_t - ET_s + LT_t - LT_i + \phi'$$

となる。

ここで、 $ET_s = 0$, $LT_t - ET_s = A_{st}$ であるから、

$$(7) \quad \phi' = LT_i - ET_i$$

となり、 ϕ' は結合点 i のスラック (Slack) であることがわかる。

次に、ある作業 (i, j) を通り結合点 s から t に到る最長パスとその工期を考えてみよう。

この場合のネットワークは図-3のようになる。

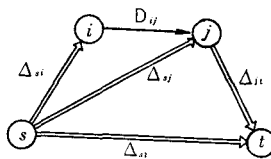


図-3 単純化されたパスのネットワーク [2]

d. ϕ'' の計算

さて図-3において、パス (s, i, j) のもつ余裕工期を ϕ'' で表わすと式(8)の関係がなりたつ。

$$(8) \quad \begin{aligned} A_{st} + D_{ij} + \phi'' &= A_{sj} \\ \phi'' &\geq 0 \end{aligned}$$

また、式(3)'' を用いると上式はさらに式(9)のように変形される。

$$(9) \quad \begin{aligned} \phi'' &= A_{sj} - A_{st} - D_{ij} \\ &= (ET_j - ET_s) - (ET_t - ET_s) - D_{ij} \\ &= ET_j - ET_t - D_{ij} \end{aligned}$$

$$(9)' = FF_{ij}$$

となり、 ϕ'' は作業 (i, j) のフリー・フロート (Free Float) を表わすことがわかる。したがって ϕ'' はパス (s, i, j) で使いつくされても、パス (s, j, t) の工期にはなんの影響も与えないことになる。

次に、パス (s, j, t) の余裕工期を考えると、前記 c. より、

$$(7)' \phi' = LT_j - ET_j$$

$$(7)'' = DF_{ij}$$

となる。つまり、 ϕ' は結合点 j のスラックであると同時に、作業 (i, j) のディペンデント・フロート (Dependent Float) であり、パス (s, i, j, t) がクリティカルになるかどうかはこの ϕ' に依存していることがわかる。

さて、式(8)、式(2)から、

$$(10) \underbrace{A_{si} + D_{ij} + \phi'' + A_{jt} + \phi'}_{\Delta_{sj}} = A_{st}$$

であり、また式(9)', 式(7)''から、

$$(11) \phi' + \phi'' = DF_{ij} + FF_{ij}$$

$$(11)' = TF_{ij}$$

となる。したがってこのことから

$$(12) A_{st} = A_{si} + D_{ij} + TF_{ij} + A_{jt}$$

である。

このことは、作業 (i, j) のトータル・フロート (Total Float) TF_{ij} は作業 (i, j) に対してよりも、むしろ結合点 s から t に到る最長パスについて意味をもつ余裕時間であることを意味している。つまり、作業 (i, j) でトータル・フロートが費消されると、パス (s, i, j, t) 上の他の作業のトータルフロートの如何に拘らず、このパス上の最長パスは『クリティカル・パス』(Critical Path) となる。

そこで次にネットワーク上のパスとフロートの関係について考えてみよう。

§ 3. ネットワーク上のパスとフロート

3.1 最長パスとフリー・フロート

a. パス (s, t) 上のフリー・フロート

前章の d. 式(8)から、

$$(13) \left. \begin{aligned} A_{s1} &= 0 + D_{s1} + FF_{s1} \\ A_{s2} &= A_{s1} + D_{12} + FF_{12} \\ &\vdots \\ A_{st} &= A_{s,t-1} + D_{t-1,t} + FF_{t-1,t} \end{aligned} \right\}$$

という関係が成り立つ.

上式をまとめると,

$$\begin{aligned} A_{st} &= D_{s1} + D_{12} + \cdots + D_{t-1,t} + FF_{s1} + FF_{12} + \cdots + FF_{t-1,t} \\ (14) \quad &= \sum D_{ij} + \sum FF_{ij} \\ &\text{ただし, } (i, j) \in \text{Path}(s, t) \end{aligned}$$

である, なお, クリティカル・パス上では上式は当然,

$$(14)' \quad A_{st} = \sum D_{ij} + \min \sum FF_{ij}$$

となる.

つまり, プロジェクトの開始結合点 s から終了結合点 t に至るパス (s, t) において, そのパスに含まれる作業の所要時間の和とフリー・フロートの和の合計はクリティカル・パスの工期に等しい. このことから2点間の最長パスとは, そのパス上の作業のフリー・フロートの和が最小なものであることがわかる.

b. パス (s, i) 上のフリー・フロート

前章 a. 式(3)より,

$$\begin{aligned} A_{si} &= \max_{\substack{(h,i) \in P \\ h < i}} [A_{sh} + D_{hi}] \\ &= ET_s + D_{s1} + D_{12} + \cdots + D_{hi} \end{aligned}$$

$ET_s = 0$ とすれば, 上式はさらに,

$$(15) \quad A_{si} = \sum D_{hi}$$

ただし, $(h, i) \in \text{Path}(s, i)$

となり, A_{si} はパス (s, i) 上の作業の所要時間の和によって表わされる.

このことはプロジェクト上の開始結合点 s から任意の結合点 i に至るパス (s, i) において A_{si} を決定するフリー・フロートの和がゼロのパスが少なくとも1本はあることを意味している.

c. パス (j, t) 上のフリー・フロート

本章の a, b から明らかなように, プロジェクト上の任意の結合点 j から, 終了結合点 t に至るパス (j, t) のうち, そのパス上のフリー・フロートの和の最小のものが最長パスである.

3.2 最長パスとトータル・フロート

前章にものべたように, ある作業のトータル・フロートはその作業について意味をもつというよりは, むしろその作業を含んで結合点 s から t に至る最長パスについて意味をもつフリー・フロートであり, もしこれをその作業で費消すれば, この作業を含む最長パス (s, t) はクリティカル・パスとなる.

3.3 トータル・フロートとフリー・フロート

a. ある作業 (i, j) のトータル・フロート (TF_{ij}) は, その作業を含みプロジェクトの開始

から終了に至る最長パス (s, i, j, t) 上のフリー・フロートの総和である。このことは本章, 3.1 3.2 から明らかである。

b. ある作業のトータル・フロートは、その作業のフリー・フロートと後続作業のトータル・フロートの最小値との和によっても求められる。

3.4 最長パスとスラック

スラックはその結合点を通してプロジェクトの開始から終了に至る最長パスの余裕時間であり、そのパス上のフリー・フロートの総和に等しい。これは、同時にその結合点で終る作業のデッド・ペンデント・フロート等しい。

3.5 パスについてフリーなフロート

現行の PERT の計算では、フリー・フロートが個別作業について算定されるため、いくつかの作業に共有されるフロートについての情報は欠除している。しかし計算上はフリー・フロートがゼロでもそれがクリティカルな作業でなければフリー・フロートを共有していると考えの方が妥当であろう。

そこで最後にパスについてフリーなフロートの問題をとりあげてみよう。

まず J.J. Moder などの呼び方にならって図-4 の左図のように、ある結合点がいくつかの作業の終了を表わすときこれを合流点 (Merge Event)、また右図のようにいくつかの作業の開始を表わすものを分岐点 (Burst Event) と呼ぶことにする。

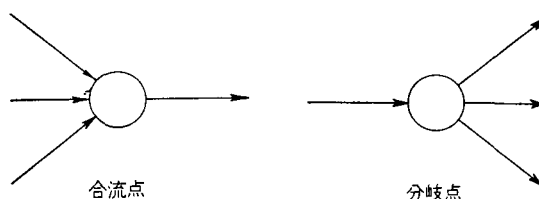


図-4

さて結合点 j の最早結合点時刻 ET_j は、その定義から、

$$ET_j = M_{ax}[ET_i + D_{ij}]$$

である。

この場合、 ET_j に色々な値が算定され、そこから M_{ax} 値を捉えるのは、合流点についてであり、分岐点の ET_j は一義的に決定されるものである。

たとえば、図-5 のネットワークにおいて、フリー・フロートをもつ可能性のある作業は、図上にチェックされたものだけである。

そこで、試みに図-5 に適当な所要時間を与えてフリー・フロートの計算を行なってみると、図-6 のようになる。

たとえば、この図-5、図-6 において $FF_{2,3}=0$, $FF_{3,4}=0$, $FF_{4,14}=18$ であるが、仮りに作業 (2,3) もしくは作業 (3,4) で18日おくれても、作業 (4,14) が4日で完成すれば、後続作業

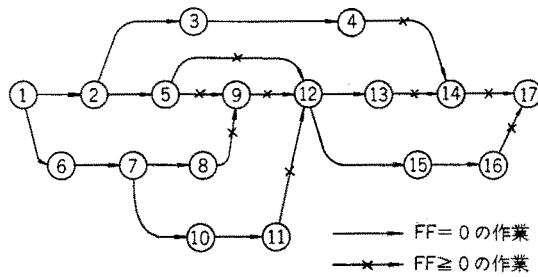


図-5 パスとフリー・フロート [1]

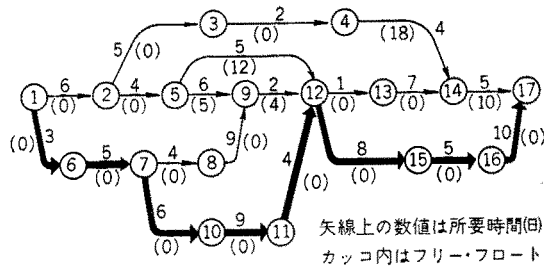


図-6 パスとフリー・フロート [2]

の工期には影響を与えないことがわかる。つまり18日間のフロートは、パス②→③→④→⑭のなかで自由に使ってもよい余裕時間であるといえる。しかし、同じ作業(4,14)に先行するノンクリティカルな作業で $FF=0$ のものでも、たとえば作業(1,2)で使った場合には、作業(2,5)に後続するパスの工期に影響を与えることになるであろう。

これらの問題は通常の PERT の教科書にはあまりふれられていないので、ここで少しとりあげてみよう。

まず第1に、フリー・フロートが発生するのは合流点で終る作業についてのみである。

第2に、合流点で終るいくつかの作業のうち、 $FF=0$ のものが少なくとも1本は必ずある。

第3に、ある作業のフリー・フロートはそれに先行する $FF=0$ の作業系列で、分岐点に至るまでの間では無条件に分配可能である。たとえば図-6 の場合は表-1 のようになる。

表-1 フリー・フロートの分配可能範囲

FF_{ij} の発生した作業	FF_{ij} の数	FF_{ij} の無条件に分配可能な範囲
5, 9	5 (日)	⑤—×→⑨
5, 12	12	⑤—×→⑫
9, 12	4	⑦→⑧→⑨—×→⑫
4, 14	18	②→③→④—×→⑭
14, 17	10	⑫→⑬→⑭—×→⑰

§ 4. 日程計画のための PERT と日程管理のための PERT

現行の PERT において、日程管理は日程計画の繰りかえしとしてとらえられている。そのため、日々変化する状況に対して手作業で日程の管理を行なうことは非常に困難である。この問題は計画と管理を一緒にして取扱っている現行 PERT の技術的欠陥であるといえよう。

そこでここでは、図-1 における最長パス (s, i) によって日程計画を、また最長パス (i, t) によって日程管理を行なうといったように、PERT の適用局面を2つに分離することによって従来とは若干異なるアプローチを試みてみよう。

このような考え方をとると、手作業による日程の計画・管理が極めて容易になるのみならず、PERT における諸概念のもつ意味は逆に現在以上に明確になるように思われる。

4.1 日程計画のための PERT

前章 3.1 のbから、いわゆるクリティカルパスとはプロジェクトの開始結合点 s からネットワーク上の任意の結合点 i に到る最長パスのうち、 $i=t$ といった特殊な場合に過ぎないこと、また最長パス (s, i) とはパス (s, i) 上のフリー・フロートの和がゼロであるようなパスであることがわかる。

この性質を利用すると、クリティカル・パスを見出すための日程計算は従来のような往復計算ではなく、最早結合点時刻の片面計算だけで充分ことがわかる。

そこで図-2 の設例について日程計画の計算を試みてみよう。まず最早結合点時刻 (ET_i) の図上計算を行なってみると図-7 のようになる。

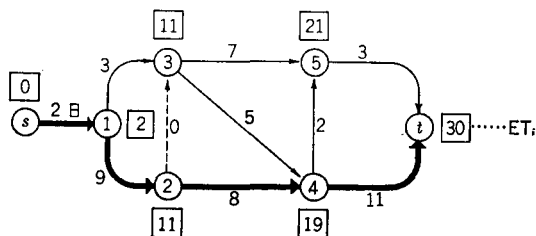


図-7 プランニング用 PERT (ET_i の計算)

作業 (i, j) のフリー・フロート (FF_{ij}) はその定義から、次式のように最早結合点時刻のみによって求められる。

$$(10) \quad FF_{ij} = ET_j - ET_i - D_{ij}$$

上式からフリー・フロートの計算を行なってみると表-2 をうる。

表-2 からクリティカル・パスを見つける方法は比較的簡単である。まずこのプロジェクトの終了結合点 t で終る作業のうち $FF=0$ のものは作業 ($4, t$) である。次に結合点④で終る作業のうち $FF=0$ のものをさがすと作業 ($2, 4$) であることがわかる。このような方法で $FF=0$ の

表-2 フリー・フロートの計算

作業 (i, j)		D_{ij}	ET_i	ET_j	FF_{ij}	CP
s,	1	2	0	2	0	*
1,	2	9	2	11	0	*
1,	3	3	2	11	6	
2,	3	0	11	11	0	
2,	4	8	11	19	0	*
3,	4	5	11	19	3	
3,	5	7	11	21	3	↑
4,	5	2	19	21	0	
4,	t	11	19	30	0	*
5,	t	3	21	30	6	

途を求めていくと、求めるクリティカル・パスは⑤→①→②→④→①で、そのパス上のフリー・フロートの総和はゼロになることがわかる。

なお、このプロジェクトの所要工期30日は計算上の工期であり、要求工期が異なる場合は、 ET_i に主要管理点（マイル・ストーン）として要求工期を与えることにより計画の組みなおしを行なう必要がある。

4.2 日程管理のための PERT

前にのべたように、現行の PERT はその理論的性格からいってプランニングの技法であり、当初たてられた計画を常に現在時点に対してアップデートする技法としては用いにくいところがある。

たとえば図-7 においても、各結合点時刻はプロジェクトの開始日現在においてこのようになることを示しているにすぎない。

極言すれば、翌日になれば今一度プランニングの計算を試みてみないと要求工期に間にあうかわからないことになる。これはコンピューターを中心にした通信・制禦系がしっかりでき上っている米国のような国では重要な問題ではないであろうが、手作業を中心にした単発的な日程管理を行なっている日本では実際上不可能に近い問題であるといえよう。

そこでここでは最早開始時刻を中心にした現行の PERT を裏返した型の最遅開始時刻（残工期）を中心にした管理用 PERT を考えてみよう。

さて図-2 の例における各結合点についての残工期（ RT_i ）の計算を行なってみると図-8 のようになる。

結合点 i の残工期 RT_i は、パス (i, t) 上の作業の所要時間に変更があった場合には変わる可能性があるが、パス (s, i) 上の作業の工期や手順の変更の影響は受けないため、図-8 をアグ

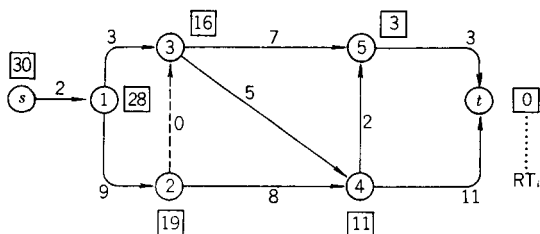
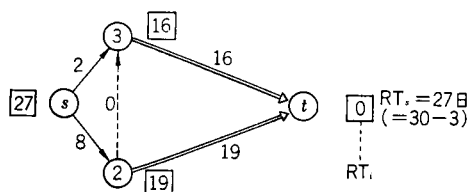
図-8 残工期 (RT_i) の計算

図-9 3日後——順調に進行中

リゲートした形で表わすとネットワークのアップデューティングが非常に容易になる。

たとえば、プロジェクトの開始3日後の状況をアグリゲートしたネットワークで表わすと、図-9 のようになる。

なお、プロジェクトの開始結合点は常に s で表わすことにする。

図-9 を見ると、クリティカル・パスは $(s, 2, t)$ で、その所要工期は27日、これは当初の所要工期30日から経過した3日を引いたものに等しく、工事が順調に進行していることを示している。

次に着工10日後、若干日程に変更があった場合が図-10である。

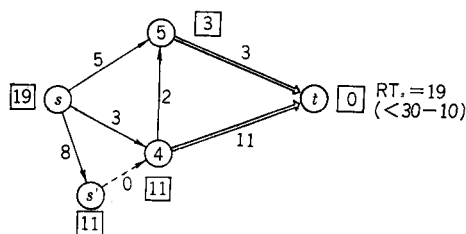


図-10 10日後——若干工期が変更

これを見ると予定の残工期は20日あるはずなのに、実際には19日あればこのプロジェクトは終了すること、つまり工期を1日短縮できたことがわかる。

現実のかなり複雑なネットワークでも、アグリゲートした形で表わすと大体この程度の簡単なものとなり、日程管理の計算も非常に容易に行なうことができる。

また視覚的に日程管理を行なうためには、図-8 を次の図-11 のような LT によるタイム・スケール工程図に表わしてみるとよい。

つまり、ネットワークをタイム・スケール工程図に表わす場合、通常は各作業の開始時刻ない

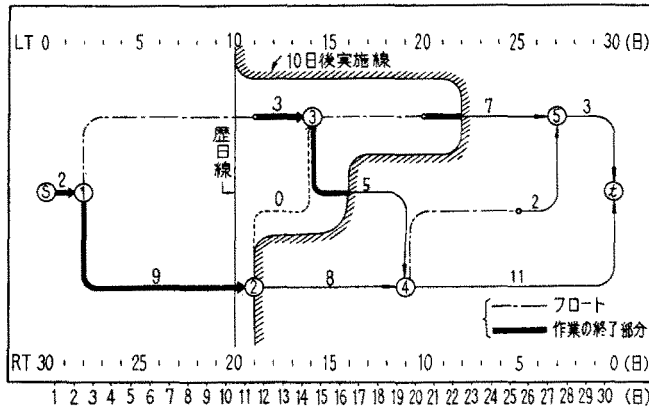


図-11 LT のタイム・スケールによる日程管理 (10日後の例)

{LT: 最遅結合点時刻
RT: 残工期

しは各結合点時刻を **ET** で示している。これに対してここでは各結合点時刻を **LT** で示し、各作業のフロートを作業の前にあらわしてみようというのである。なおこの図に現われてくるフロートは各作業のインディペンデント・フロートとディペンデント・フロートの和である。

このような図は、通常の **ET** によるタイム・スケール工程図にくらべると次のような日程管理上の利点をもっている。

まず第一に、フロートが作業の前にきているため、それに続く作業が共有しているフロートの大きさをあらかじめ知ることができる。

第二に、工事の実施線が当日の歴日線（図-11では工事着手10日後）より先に進んでいる場合は、その2線の差引日数だけトータル・フロートが残っていること、逆に実施線が当日の歴日線に達していない場合は、その差の日数だけ工期が遅れていることを意味している。

第三に、前2者から明らかなように、かなり複雑なネットワークの場合でも、計算や書き換えの手間をかけずに日程の現状把握ができ、しかもその結果は素人でも容易に目で見えてわかるといった利点をもっている。

従来非常に困難であると思われてきていた手作業による PERT の日程管理も、少し観点をかえると意外に容易な途がひらけるのではないかと考えられる。

§ 5. あとがき

PERT 系の日程管理の方法論が日本にやや本格的に紹介されはじめてから、はやいもので5年の才月が過ぎ去ろうとしている。理論的には非常に簡単なものではあるが、実務的に与えたその影響は、対象領域の広さにおいても深さにおいても、戦後のORの幾多の手法のうちでも最も広汎かつ深刻なものであったといえよう。

なかでも従来適切な管理手法をもたなかった建設業においては驚異的な普及をみせて現在に至っている。そして今、この外国に生れた新しい技法は日本の土壌にはっきりと根をおろし始めて

いる。

ここで述べた PERT ネットワークの新しい観点は、殆んどその大部分がプロジェクトごとの手作業によって進められている日本における PERT の現状を考え、これらを前提にして日程の計画と管理に新しい接近を試みたものである。

このような側面からのアプローチがまだ非常に少ない現在では、この試論も若干の意味をもつのではないかと考えている。

参 考 文 献

- 1) 荒木睦彦「ネットワークによる日程管理」建築技術, (1966年2月)
- 2) 同 上「PERT における Path 概念の一般化について」清水建設研究所報, (1966年10月)
- 3) 小早川洋太郎「建築学会ネットワーク分科会資料」No. 21 (1966年)
- 4) 同 上「ネットワークの時間的性質に関する 2, 3 の考察」建築学会論文報告集, (1967年予定)
- 5) 中川友康「日程計画 PERT/CPM の計算方法」電力中研計算機室資料, No. 29 (1965年11月)