

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

§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 (Arrow Diagram) 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。

しかし少し視点を変えて 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び上ってくるように思われる。

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^{注1)}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、将来建築学会の材料施工委員会ネットワーク分科会におけるネットワーク技法の手引き書の中に、なんらかの形で導入される予定であるが、『学会』の『手引き書』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てみたい。

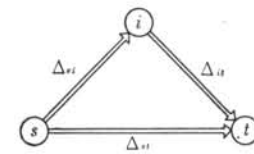
なおここでは現行のネットワーク技法の知識は既存のものとして、用語、記号、計算方法などについて特に説明を加えていない。したがってそれらについては当社の教科書^{注2)}およびPERTの一般的な参考文献によられたい。

注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)

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自身にあることはもち論であるが、この考え方の多くは諸先学の秀れた労作に依存している。なかでもフロートとパスの関係に関する諸定理は、竹中工務店・生産管理部・小早川洋太郎氏によって見出されたもの^{注3)}を、若干手を加えた形で利用させていただいた。また当社設計部宮崎課・権藤英男氏には、草稿作成の過程でいろいろと御示唆をいただいた。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、時間的に最も長くかかるパスをその2点間の最長パスとよぶことにする。



(i=1,2,...,t-1)

図-1 単純化されたパスのネットワーク

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 $\textcircled{i} \rightarrow \textcircled{k}$ 、また最長パス (i, k) の所要

工期を A_{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 に至る所要工期を考えてみると式(1)の関係がなりたつ。

注2) 当社ネットワーク研究委員会：“新しい工程管理手法の手引” 当社工務部、(昭40.4月)

注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)

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

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。

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

$$\phi' \geq 0$$

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。

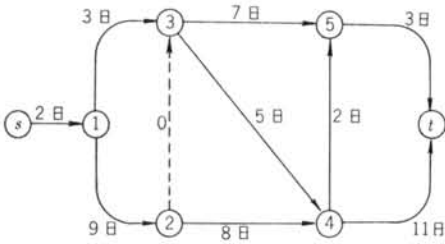


図-2 簡単な設例

a. A_{st} の計算

結合点 s の最早結合点時刻 (ET_s) を 0 とすると、

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

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

$$A_{s3} = \text{Max} \left[\begin{array}{l} A_{s1} + D_{13} = 2 + 3 = 5 \\ A_{s2} + D_{23} = 11 + 0 = 11 \end{array} \right] = 11$$

$$\dots\dots\dots$$

$$A_{st} = \text{Max}_{(h,i) \in P, h < i} [A_{sh} + D_{ht}] \quad \dots\dots\dots(3)$$

$$= \text{Max} [ET_h + D_{ht} - ET_s] \quad \dots\dots\dots(3)'$$

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

となる。

したがって、 $ET_s = 0$ とすれば、 A_{st} は ET_t 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 $A_{st} = \lambda$ 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、それらの関係は当然、

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

である。

最終結合点 t の残工期 (RT_t) を 0 とすると、

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

$$D_{it} = \text{Max} \left[\begin{array}{l} RT_t + D_{it} = 0 + 11 = 11 \\ A_{st} + D_{st} = 3 + 2 = 5 \end{array} \right] = 11$$

$$\dots\dots\dots$$

$$D_{it} = \text{Max}_{(i,j) \in P, i < j} [A_{jt} + D_{ij}] \quad \dots\dots\dots(5)$$

$$= \text{Max} [RT_j + D_{ij} - RT_t] \quad \dots\dots\dots(5)'$$

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

となる。式(4)の関係を導入する上式はさらに、

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

$$= \text{Max} [LT_t - LT_j + D_{ij}] \quad \dots\dots\dots(6)'$$

$$= LT_t - \text{Min} [LT_j - D_{ij}] \quad \dots\dots\dots(6)''$$

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、

$$A_{st} = ET_t - ET_s + LT_t - LT_i + \phi' \text{ となる。}$$

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

$$\phi' = LT_t - ET_t \quad \dots\dots\dots(7)$$

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

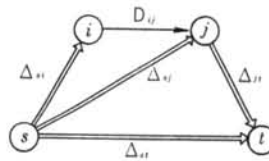


図-3

次に、ある作業 (i, j) を通り結合点 s から t に至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。

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。

$$A_{st} + D_{ij} + \phi'' = A_{sj}$$

$$\phi'' \geq 0 \quad \dots\dots\dots(8)$$

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。

$$\phi'' = A_{sj} - A_{st} - D_{ij}$$

$$= (ET_j - ET_s) - (ET_t - ET_s) - D_{ij}$$

$$= ET_j - ET_t - D_{ij} \quad \dots\dots\dots(9)$$

$$= FF_{ij} \quad \dots\dots\dots(9)'$$

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) の余裕工期を考えてみると、前記 c. より、

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

$$=DF_{ij} \quad \dots\dots\dots(7)''$$

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、

$$\underbrace{D_{st} + D_{ij} + \phi' + D_{jt} + \phi'}_{D_{sj}} = D_{st} \quad \dots\dots\dots(10)$$

であり、また式(9)', 式(7)''から、

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

$$= TF_{ij} \quad \dots\dots\dots(11)'$$

となる。したがってこのことから

$$D_{st} = D_{st} + D_{ij} + TF_{ij} + D_{jt} \quad \dots\dots\dots(12)$$

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 に至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その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、

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

という関係が成り立つ。

上式をまとめると、

$$\begin{aligned} D_{st} &= D_{s1} + D_{12} + \dots + D_{t-1,t} + FF_{s1} + FF_{12} + \dots \\ &\quad + FF_{t-1,t} \\ &= \sum D_{ij} + \sum FF_{ij} \quad \dots\dots\dots(14) \end{aligned}$$

ただし、 $(i, j) \in \text{Path}(s, t)$

である。なお、クリティカル・パス上では上式は当然、

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

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、

$$\begin{aligned} D_{st} &= \max_{\substack{(h,i) \in P \\ h < i}} [D_{sh} + D_{ht}] \\ &= ET_s + D_{s1} + D_{12} + \dots + D_{ht} \end{aligned}$$

$ET_s = 0$ とすれば、上式はさらに、

$$D_{st} = \sum D_{ht} \quad \dots\dots\dots(15)$$

ただし、 $(h, i) \in \text{Path}(s, i)$

となり、 D_{st} 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 D_{st} 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

前章にも述べたように、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、ディペンデント・フロートにも等しい。

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^{注4)}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。

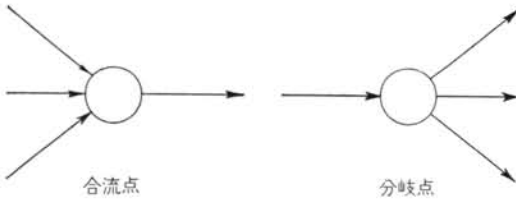


図-4

さて結合点 j の最早結合点時刻 ET_j は、その定義から、

$$ET_j = \text{Max}[ET_i + D_{ij}]$$

である。

この場合、 ET_j に色々な値が算定され、そこから Max 値を採ぶのは、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。

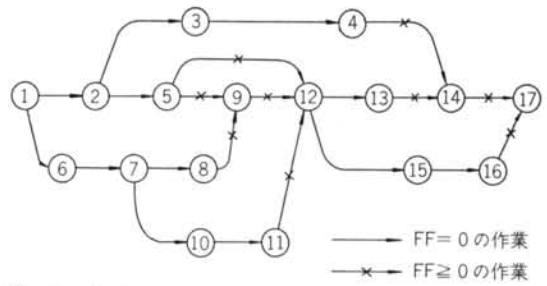


図-5 パスとフリー・フロート

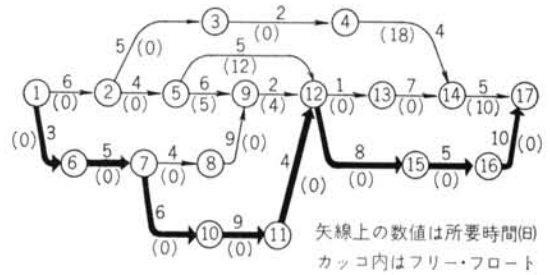


図-6

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)もしくはは作業(3,4)で18日おくても、作業(4,14)が4日で完成すれば、後続作業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

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

表-1

注4) Joseph J. Moder & Cecil R. Phillips: "Project Management with CPM and PERT" Reinhold, (1964)

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。

§ 4. 日程計画のための PERT と 日程管理のための PERT

現行の PERT は日程の計画手法としての性格が強く、管理手法としての性格は弱いように思われる。そのために実際に現場で PERT を行なうにあたって最も困難なのはいわゆる『フォロー・アップ』であるということになる。この『フォロー・アップ』を中心にした、いわば日程管理のためのネットワークの問題については既に別稿^{注5)}で扱ったが、ここでは前章でのべたパスとフロートの性格を利用することにより、計画と管理のためのネットワーク手法としてさらに一般化を試みてみよう。

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・パスはプロジェクトの開始結合点 s から終了結合点 t に至るパスのうちで、そのパス上のフリー・フロートの和がゼロになるようなパスであることがわかる。フリー・フロートはプロジェクト上の各結合点についての最早結合点時刻の計算のみで算定しうることを考えると、クリティカル・パスを見つけるためには通常の PERT のような往復計算をしなくても、片道計算だけで充分なことが分かるであろう。

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。

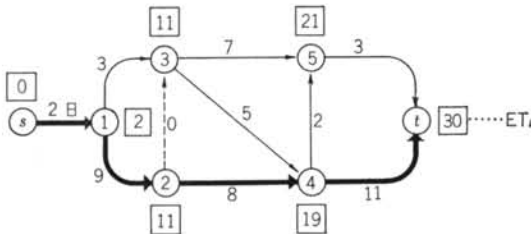


図-7 プランニング用 PERT (ET_i の計算)

作業 (i, j) のフリー・フロート (FF_{ij}) はその定義から次式によって求められる。

$$FF_{ij} = ET_j - ET_i - D_{ij} \quad \dots\dots\dots (6)$$

注5) 拙稿: “前掲論文” 建築技術, (1966年2月)

求められた最早結合点時刻からフリー・フロートの計算を行なってみると表-2 をうる。

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	

表-2 フリー・フロートの計算

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 $FF=0$ の途を求めていくと、求めるクリティカル・パスは ⑧→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日が要求工期と異なる場合については工期制約の問題として後述する。

4.2 日程管理のための PERT

前にものべたように、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 においてもプロジェクトの開始日現在において、各結合点時刻がこのようになるということを示すにすぎない。

つまり翌日になると全部の時刻が変わってしまい、極言すれば毎日プランニングの計算を行なっていなければならない。これはコンピューターを中心にした通信・制御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、実際上不可能なことであるといえる。

そこで残工期とフリー・フロートを用いた2つの方法で、日程のアップデートを試みてみよう。

a. 残工期 (RT_i) を用いて

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。

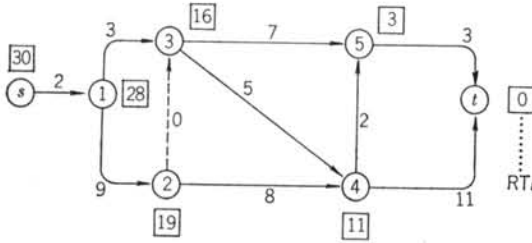


図-8 残工期 (RT_i) の計算

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 をアグリゲートした形で表わすとネットワークのアップデート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。

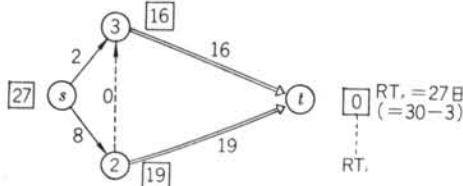


図-9 3日後——順調に進行中

なお、プロジェクトの開始結合点は常に s で表わすことになる。

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。

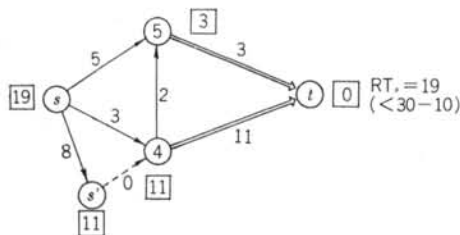


図-10 10日後——若干工期が変更

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、アップデートの計算も非常に容易になる。

b. フリー・フロート (FF_{ij}) を用いて

前章 3.1, 3.3 などに見られるフリー・フロートの性質を利用して、日程のアップデートを行なう方法を考えてみよう。

図-2の例について、各作業のフリー・フロートおよび各結合点について、それらと終了結合点 t を結ぶパス上のフリー・フロートの和の最小のものの値を計算すると、図-11のようになる。

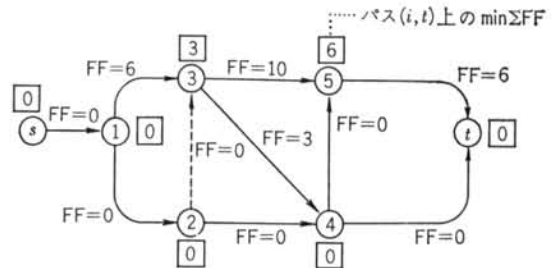


図-11 パス上のフリー・フロート

図-9 の場合と同じように、工事開始3日後のフリー・フロートの状況をこれもアグリゲートしたネットワークで表わしてみると図-12のようになる。

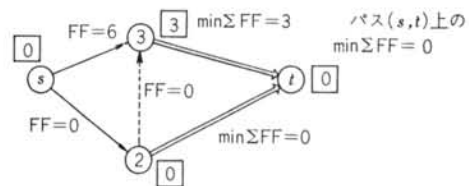


図-12 3日後

この図をみれば明らかなように、フリー・フロートの和がゼロのパスは ($s, 2, t$) でクリティカルとなっており、工事は遅くもなく早くもなく、順調に進行していることがわかる。

次に前と同様に10日後の状況を見ると、図-13のようになる。

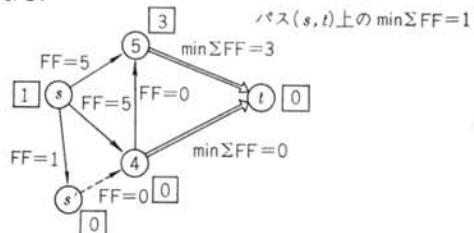


図-13 10日後

これを見ると、パス (s, t) 上のフリー・フロートの和の最小値は1であり、当初の工期に1日余裕ができたこと、つまり1日短縮できたことがわかる。

以上の叙述から明らかのように 4.2 a. では、常に現在時点における残工期そのものを、b. では所定工期の増減を示すことにより、日程の管理を行なおうとしている。実際の日程管理にあたっては、それらの使いやすいう方を用いればよいであろう。

§ 5. 中間工期指定とパス・ネットワーク

今までのべたところでは、ネットワーク上の計算工期とプロジェクトの注文主の要求工期が一致するものとして話を進めてきた。しかし、現実にはこの二者は一致しないと考える方が一般的であろう。

そこでここでは、プロジェクト上の工期が中間的に指定されている場合を考えてみよう。工期指定の種類は表-3 に示されるような、3種類と考えられる。

種類	開始日指定	終了日指定
(1)	××日には開始しなければならない	××日には完了しなければならない
(2)	××日にならなければ開始できない	××日にならなければ完了しない
(3)	××日に開始しなければならない	××日に完了しなければならない

表-3 工期指定の種類

工期指定の場合の処理方法としては、結合点時刻を指定する方法と、リード・タイムを用いる方法があるが、ここでは後者の方法を用い、既のべた片面計算の方法によって話を進めてみよう。

5.1 ××日には開始しなければならない ××日には終了しなければならない

これはある結合点について最遅結合点時刻を指定する場合に等しい。

リード・タイムを用いてこれを表わすと、図-14のようになる。

そこで図-2 の簡単な設例にもとづいて実際の計算を試みてみよう。

たとえば、図-2 の例に対して表-4 のような指定が与えられたとすると、そのネットワークは図-15のように表わされる。

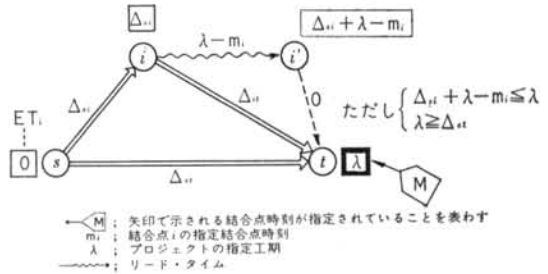


図-14

指定作業	指定の内容	リード・タイムのとり方
開始日指定	3, 5 遅くとも9日には開始すること	結合点③からプロジェクトの終了までの21日間をリード・タイムにとる
	5, t // 20日 //	同上10日間をリード・タイムとする
終了日指定	(プロジェクト全体の工期) (30日)	—
指定	1, 3 遅くとも9日までには終了していること	リード・タイム→21日
	3, 5 // 20日まで //	リード・タイム→10日
	全体のプロジェクト // 30日まで //	—

表-4 日程の指定

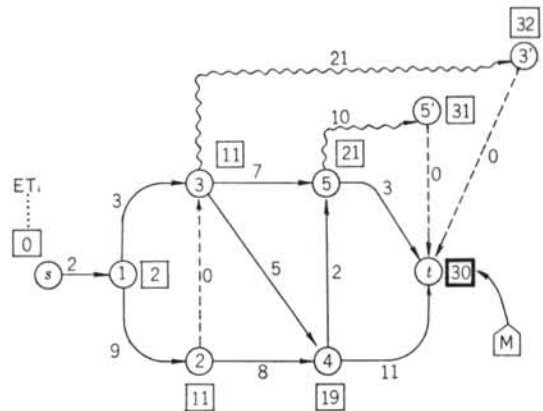


図-15

各結合点について最早結合点時刻の計算を行なった結果は図中の□内になる。

これを見ると、パス③→①→②→③→③' は所要工期32日で要求工期を2日超過する。またパス③→①→②→④→⑤→⑤' は31日で1日超過することがわかる。そこでこれら2つのパスが共有する部分パス③→①→②で2日短縮できれば、計算工期を要求工期内におさめることがわかる。

たとえば図-16のようなリプランニングが行ないうる

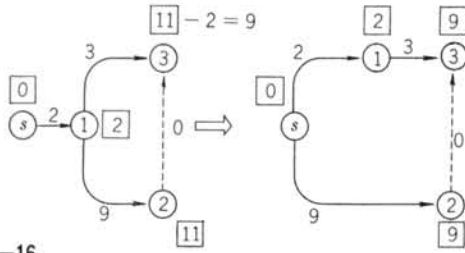


図-16

実際の作業をともなわないパスであり、クリティカルな作業は⑤→②のみであることがわかる。

5.2 ××日にならなければ開始できない ××日にならなければ終了しない

これはある結合点について最早結合点時刻を指定する場合に等しい。前と同様にリード・タイムによって表わすと図-18のようになる。

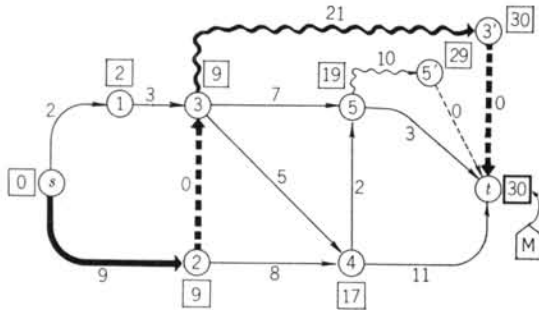


図-17

とすれば、ここで2日間短縮され、図-17のように要求工期内に完成しうることになる。

式(6)および表-2の計算方法を図上に手作業で試みることは充分可能であるが、分かりやすいようにここでは表-5であらわしてみよう。なおこの計算によって求められたクリティカル・パスにおいて、②→③→③'→④はダミーとリード・タイムによって形づくられた

作業 (i, j)	D _{ij}	ET _i	ET _j	ET _j - ET _i - D _{ij}	CP
		ES	—	FF _{ij}	
s, 1	2	0	2	0	*
s, 2	9	0	9	0	
1, 3	3	2	9	4	*
2, 3	0	9	9	0	
2, 4	8	9	17	0	p
3, 4	5	9	17	3	
3, 5	7	9	19	3	*
3, 3'	21	9	30	0	
4, 5	2	17	19	0	p
4, t	11	17	30	2	
5, 5'	10	19	29	0	*
5, t	3	19	30	8	
3', t	0	30	30	0	*
5', t	0	29	30	1	

表-5

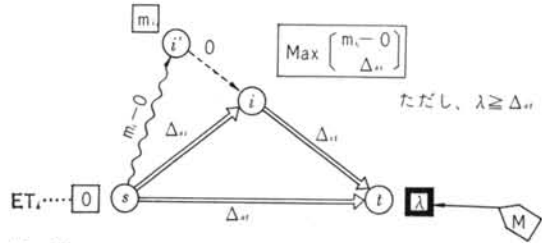


図-18

前と同じ設例を用いて、結合点③、④、⑤について表-6のように日程の指定が与えられた場合を考えてみよう。

指定作業	指定の内容	リード・タイムのとり方
開始 3, 5 3, 4	早くても13日にならないと開始できない	プロジェクトの開始から結合点③までの13日間をリード・タイムにとる
日指定 4, 5 4, t (プロジェクト全体の工期)	早くても20日にならないと開始できない (30日)	結合点④まで20日間をリード・タイムにとる —
終了日指定 1, 3 2, 3 2, 4 3, 4 全体のプロジェクト	早くても13日にならないと終了しない 早くても20日にならないと終了しない 遅くとも30日までは終了していること	リード・タイム→13日 リード・タイム→20日 —

表-6

ネットワークを書いてみると図-19のようになる。

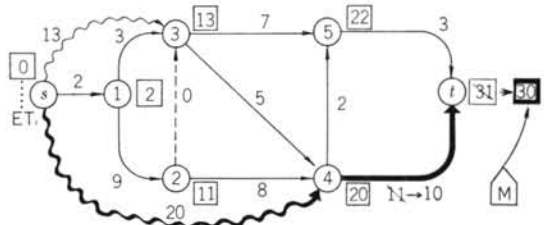


図-19

図-19から明らかなように、結合点 s から t に至る最長パスは⑧→④→①で、その所要工期は31日となり、要求工期を1日超える。リード・タイムは短縮できないので、要求工期に間に合わせるためには作業(4, t)を短縮する以外に方法がない。そこで作業(4, t)に対してコンプレッションを行なうと所要工期は30日となる。クリティカル・パスが⑧→④→①であることは図-19より明らかであるが、参考までに計算を試みた結果は表-7のようになる。

作業 (i, j)	D_{ij}	ET_i	ET_j	$\frac{ET_j - E_{Tj} - D_{ij}}{T_i - D_{ij}}$	CP
		ES	—	FF_{ij}	
$s, 1$	2	0	2	0	*
$s, 3$	13	0	13	0	
$s, 4$	20	0	20	0	
1, 3	3	2	13	8	
1, 2	9	2	11	0	
2, 3	0	11	13	2	
3, 5	7	13	22	2	
3, 4	5	13	20	2	
2, 4	8	11	20	1	
4, 5	2	20	22	0	
4, t	10	20	30	0	
5, t	3	22	30	5	

表-7

5.3 ××日に開始しなければならない
××日に終了しなければならない

これはある結合点について最早・最遅結合点を共に指定した場合に等しい。

これをパスのネットワークで表わすと図-20のようになる。

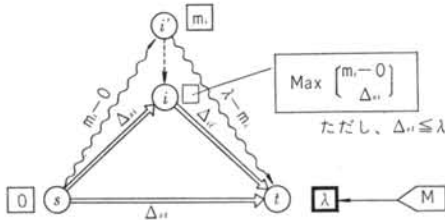


図-20

前と同じ設例において、結合点⑧, ③, ④, ①に対して表-8のような日程の指定が与えられたとする。

そのネットワークは図-21のようになる。

さて、計算工期は33日で要求工期に比べて3日超過する。この計算工期を決定しているパスは⑧→③→④→④'であり、作業(3, 4)を短縮するか、リプランニ

指定作業	指定の内容	リード・タイムのとり方	
開始日指定	$s, 1$	0日に開始する	—
	3, 5	15日に開始する	⑧→③15日間のリード・タイム
	3, 4		③→④ // //
	4, 5	17日に開始する	⑧→④17日間のリード・タイム
	4, t		④→④'13 // //
(プロジェクト全体の工期)	(30日)	—	
終了日指定	1, 3	15日に終了する	⑧→③15日間のリード・タイム
	1, 2		③→① // //
	2, 4	17日に終了する	⑧→④17日間のリード・タイム
	3, 4		④→④'13 // //
	5, t	遅くとも30日までに	—
4, t	は終了	—	

表-8

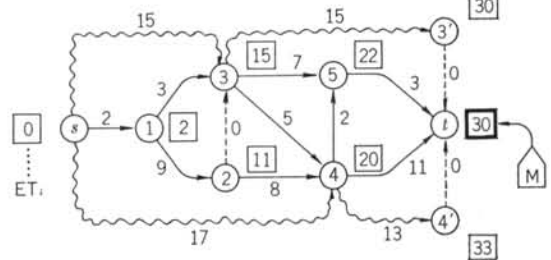


図-21

ングを行なう以外に手はない。ここでは作業(3, 4)の工期は、短縮できないものとしてリプランニングを行なった1つの例を図-22に示してみよう。

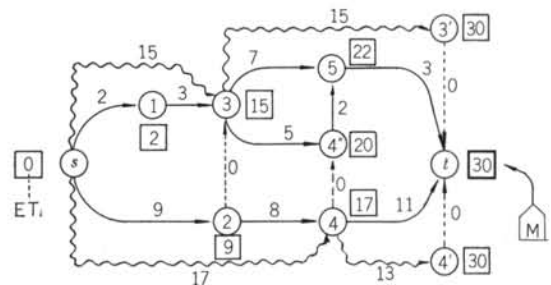


図-22

図-22の事例に対して、表-5, 表-7の手順にしたがってフリー・フロートの計算を図上に試みた結果を、図-23に別図として示してみよう。

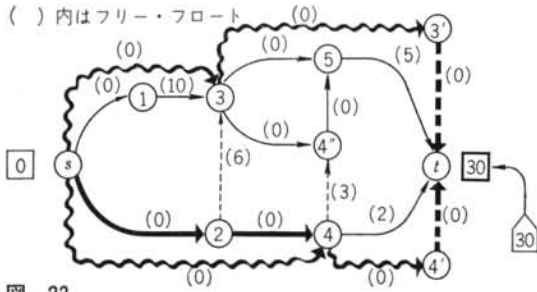


図-23

FF_{ij} の発生した作業	FF_{ij} の日数	FF_{ij} の分配可能範囲
1, 3	10 (日)	③→①-x→③
2, 3	6	計算上のフロートで, 現実的意味はもたない. 分配は不可.
4, t	2	④-x→④
4, 4'	3	作業(2,3)の場合と同じ.
5, t	5	③→⑤-x→④ 2本のパス上で分配可能であるが $FF_{35} \leq 5 - FF_{34} - FF_{45} \leq 5 - FF_{34}$ でなければならない.

表-9

この図から明らかのように, クリティカル・パスは3本であるが, 実際に作業を伴うパスは⑤→②→④にすぎない. 残りの作業は, 一応フロートをもつと考えてよい. そこで第3章の表-1にしたがって, フリー・フロートの状況を見ると表-9のようになる.

§6. あとがき

全体を通じて PERT/TIME の技術的諸問題について若干の問題提起を行なってきた. ここで取り上げたような側面からのアプローチが現在では未だ少ないので, かなり試論的なものにならざるを得なかった. 大方の御批判をいただき, 今後さらにリファインしていきたいと考えている.

ただここでの計算上の工夫は, 現場での手作業を前提にしたものであり, 電算処理を行なう場合には, 現行の方法と大きな違いは出ないと思われる. しかしここでの考え方そのものは, ネットワーク技法の今後の展開, 特に数理計画法との関連において, かなり重要性をもつことになるであろう.

<参考文献>

前掲文献の他には次のものがある.

- 1) 刀根薫編著: "PERT講座" 全四巻, 近刊.
- 2) 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. 11月)
- 3) 小早川洋太郎: "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. 予定)