

# 用電子計算機作計劃控制的研究

楊 維 槟

## 1.1 PERT/TIME

所謂「計劃控制」(Project Control) 常涉及利用電子計算機處理 PERT (計劃評核術) 等計劃管理上有關資料的制度。關於 PERT 的技巧，已相當普遍，其概要載於附錄〔1〕〔2〕。

PERT 與電子計算機之結合應用，已形成強力的管理工具，運用技術並有長足進步；PERT 系統的第一代是 PERT/TIME，第二代是 PERT/COST，第三代是 PERT/LOB，第四代將是 PERT/COST/LOB。這些 PERT 技巧中前二者已由 IBM 公司及 CDC 公司分別製成 PCS 及 PMCS 兩種應用程式，其特點將於下一章介紹。因 PCS 及 PMCS 祇告訴使用人依照格式及順序寫控制卡及資料卡，難令人了解運算之進行全貌，本章特先介紹運算步驟(Algorithm) 及筆者自導之程式。

PERT/TIME 作業如下：

(一) 建立網路圖：

1. 結點，以圓圈附以編號表示時點或事件(event)，編號儘量採用順序號碼 (sequential numbering) 以利電腦處理。
2. 作業，以箭線表示。
3. 虛作業，以虛箭線表示。
4. 時間，以週(或天)表示。

(二) 輸入資料：

- |                                  |                     |
|----------------------------------|---------------------|
| 1. 估計作業時間：                       | 2. 前後結點 (EVENT) 編號。 |
| 設 $a =$ 樂觀估計時間 (optimistic time) | 3. 特定日期：            |
| $m =$ 最可能估計時間 (most likely time) |                     |
| $b =$ 悲觀估計時間 (pessimistic time)  |                     |
- 則 估計時間  $t_e = (a + 4m + b) / 6$   
變異數  $\sigma^2 = [(b - a) / 6]^2$

有 TS (預計完成日期)，TB (規定開始日期)，TR (限定完成日期) 之三種。

(三) 電子計算機進行之計算：

- |              |                              |
|--------------|------------------------------|
| 1. 最早完成日期 TE | 3. 總寬裕 (TOTAL FLOAT) 或 SLACK |
| 2. 最遲完成日期 TL |                              |

〔例〕 1. 網路圖

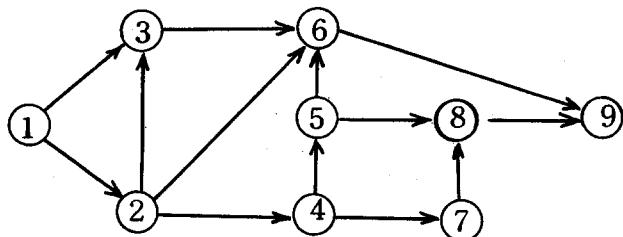


圖 1.1

## 2. 作業時間表：

作業 (i, j)	$t_e$	$\sigma_{ij}^2$	作業 (i, j)	$t_e$	$\sigma_{ij}^2$
1, 2	5.5	1.2	4, 7	10.3	1.7
1, 3	13.0	3.0	5, 6	20.0	3.3
2, 3	7.0	1.0	5, 8	4.0	0.7
2, 4	5.2	0.8	6, 9	3.2	0.5
2, 6	16.5	2.5	7, 8	3.2	0.8
3, 6	14.7	1.7	8, 9	16.5	1.2
4, 5	6.0	1.2			

## 3. ALGORITHM :

Step 1. 記入 ( $t_e$ )<sub>ij</sub>

T <sub>L</sub>	0.0	5.5	22.8	10.7	17.5	37.5	21.0	24.2	40.7	i	T <sub>E</sub>	V
j	1	2	3	4	5	6	7	8	9			
		5.5	13.0							1	0	0
			7.0	5.2		16.5				2	5.5	1.2
						14.7				3	13.0	3.0
					6.0		10.3			4	10.7	2.0
						20.0		4.0		5	16.7	3.2
										3.2	6	36.7
										7	21.0	3.7
									16.5	8	24.2	4.5
										9	40.7	5.7

## PERT/TIME 程式

MS FORTRAN (4.2)/USSOS

11/28/75

## PROGRAM PERTMAIN

C PERT MAIN PROGRAM

C TIME ESTIMATIONS FOR PERT NETWORK

C K = ACTIVITY NUMBER

C I = BEGINNING EVENT OF ACTIVITY

C J = ENDING EVENT OF ACTIVITY

C DIJ = ACTIVITY DURATION

C TA = EARLIEST EVENT DATE

C TB = LATEST EVENT DATE

C ES = EARLIEST STARTING DATE

C TLS = LATEST STARTING DATE

C DF = EARLIEST FINISHING DATE

C TLF = LATEST FINISHING DATE

C TF = TOTAL FLOAT

C FF = FREE FLOAT

C TIF = INDEPENDENT FLOAT

C CCP = CRITICAL PATH

C SIG = VARIANCE OF DIJ

DIMENSION I(100),J(100),A(100),B(100),C(100),DIJ(100),

SIG(100)

INTEGER A,B,C,DIJ,SIG

READ(6,\*) U,I(K),J(K),A(K),B(K),C(K),K=I,N

FORMAT(I4,(2I4,3F 3.0))

C COMPUTATION OF DURATION AND VARIANCE

DO 15 K=1,N

DIJ(K)=(A(K)+4.\*B(K)+C(K))/6.0

SIG(K)=(C(K)-A(K))\*#2/36.

5 CONTINUE

C WRITE DATA

WRITE(6,2)

FORMAT(1H,2 X,7HDATA OF PERT TIME///,4X,1HK,3X,IHI,3X,1HJ,9X,  
11HA,14X,HE,4X,1HC,12X,3HDIJ,12X,3HSIG)

-WRITE(6,25) (K,I(K),J(K),A(K),B(K),C(K),DIJ(K),SIG(K),K=1,N)

5 FORMAT(1H,3I4,3F 5.2)

CALL PERT(U,I,J,DIJ,SIG)

STOP

END

MS FORTRAN (4.2)/MSOS

11/23/71

```

SUBROUTINE PARTI(N,I,J,DIJ,SIG)
DIMENSION I(100),J(100),DIJ(100),SIG(I00),ES(100),TLS(100),EF(100),
      TLF(100),TE(100),FF(100),TIF(100),TA(100),TB(100),V(100),CCP(100),
      S(100)
INTEGER XMAX,E
INTEGER DIJ,SIG,ES,TLS,TLF,TF,FF,TIF,TA,TB,V,CCP,S,BLK,TINF,
      TMAX,MIN,ST
DATA (BLK=11),(STR=11**),(EPS=1.E-04),(TINF=1.E1)
ERROR CHECK
INDEX=
DO 1 K=1,N
IF(I(K).GE.J(K)) GO TO 10
IF(I(K).LT.INDEX) GO TO 110
IF(I(K).GT.INDEX.AND.I(K).NE.INDEX+1) GO TO 120
IF(I(K).EQ.INDEX+) INDEX=I(K)
CONTINUE
DO 2 K=1,N
TA(K)=0.
CCP(K)=BLK
TB(K)=TINF
CONTINUE
V(1)=0.
DO 3 K=1,N
IK=I(K)
JK=J(K)
TMAX=TA(IK)+DIJ(K)
IF(TA(JK).GT.TMAX) GO TO 30
TA(JK)=TMAX
V(JK)=V(IK)+SIG(K)
CONTINUE
JN=J(N)
TB(JN)=TA(JN)
K=N
DO 45 L=1,N
IK=I(K)
JK=J(K)
TMIN=TB(IK)-DIJ(K)
IF(TB(IK).GT.TMIN) TB(IK)=TMIN
K=K-1
CONTINUE
DO 50 K=1,N
IK=I(K)
JK=J(K)
ES(K)=TA(IK)
TLS(K)=T(JK)-DIJ(K)
EF(K)=TA(IK)+DIJ(K)
IF(TE(K).LE.CPS). CCP(K)=STR
EF(K)=TA(.F)-TA(IK)-DIJ(K)
TIF(K)=XMAX(EPS,TB(JK)-TB(IK)-DIJ(K))

```

```

50 CONTINUE
DC 55 L=1,JN
S(L)=TB(L)-TA(L)
55 CONTINUE
WRITE(6,6)
6 FORMAT(1H, //, 5X, "PERT TIME REPORT", //, 2X, #EVENT LIST#, //, , X,
      2LZ, 8X, 2I1 TA, 3X, 2I1 TB, 13X, HS)
      WR1TE(6,65) (L, TA(L), TB(L), S(L), L=1,JN)
65 FORMAT(1H, I4, 3F19.2)
      WRITE(6,7)
7 FORMAT(1H, //, 2X, #ACTIVITY LIST#, //, 5X, #EF#, 8X, #ES#, 13X, #LS#, 13X,
      2#EF#, 13X, #LF#, 13X, #TF#, 13X, #EF#, 13X, #IF#13X, #CP#)
      WRITE(6,75) (K, ES(K), TLS(K), EF(K), TLF(K), TF(K), FF(K), TIF(K),
      3CCP(K), K=1,N)
75 FORMAT(1H, I4, 8F15.2)
      RETURN
100 WRITE(6,200) K
100 FORMAT(1H, #ERROR ~ ERROR OF EVENT ORDERING K = #, I5)
      RETURN
105 WRITE(6,210) K
105 FORMAT(1H, #ERROR 2 ERROR OF ACTIVITY ORDERING K = #, I5)
      RETURN
110 WRITE(6,220) INDEX
110 FORMAT(1H, #ERROR 3 LACK OF EVENT INDEX = #, I5)
      RETURN
END

```

FORTRAN DIAGNOSTIC RESULTS FOR PERTT

Step 2.  $(T_E)_j = \max_i \{ (T_E)_i + a_{ij} \}$  最早期待日期

$$V_j = V_i + \sigma_{ij}^2$$

Step 3.  $(T_L)_i = \min_j \{ (T_E)_j - a_{ij} \}$  最遲容許日期

## 1.2 CPM成本計算程式：

通常想早完成計劃，須付出較高費用；但假如提早完成可減少間接費用，故最好的完成期限是求使總費用為最小的期間，另一方面，不同的作業有不同的趕工方法與趕工成本。故為了以最經濟的方式達成合適的工期，產生所謂 PERT/COST 的問題，亦稱為網路規畫 (CPM) 法。此問題可化成所謂 “NETWORK FLOW ALGORITHM” 如下：

「在限制條件

$$T_E(i) + y(i, j) - T_E(j) \leq 0 \quad \forall (i, j) \in A$$

$$T_E(n) - T_E(1) \leq \lambda$$

$$y(i, j) \leq D(i, j)$$

$$-y(i, j) \leq -d(i, j)$$

之下求使目的函數

$$P(\lambda) = \sum C(i, j)y(i, j) \text{ 為最大。}$$

式中  $d(i, j)$  = 趕工時間 (crash time)

$D(i, j)$  = 正常期間 (normal time)

$C(i, j)$  = 費用增加率 (time-cost slope)

$y(i, j)$  滿足如下條件

$$d(i, j) \leq y(i, j) \leq D(i, j)$$

$\lambda$  = 計畫日期 (project duration)

PERT/COST 的運算程式：

〔例〕1. 網路圖

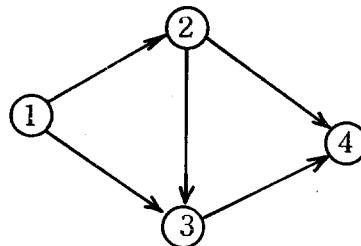


圖 1.2

2. 輸入資料：

ARC	$d(i, j)$	$D(i, j)$	$C(i, j)$
1, 2	1	3	3
1, 3	2	4	1
2, 3	0	2	1
2, 4	2	5	1
3, 4	1	6	3

3. 利用 Fulkerson 的 Algorithm 將問題化爲：

$$\text{在 } f(i, j) + g(i, j) - h(i, j) = C(i, j), \forall (i, j) \in A$$

$$\sum_j [f(i, j) - f(j, i)] = \begin{cases} v, & i = 1 \\ 0, & i \neq 1, n \\ -v, & i = n \end{cases}$$

限制條件下求使

$$\lambda v + \sum_{(i, j) \in A} D(i, j)g(i, j) - \sum_{(i, j) \in A} d(i, j)h(i, j)$$

爲最小的問題。

$$\text{因 } q(i, j) = \max [0, C(i, j) - f(i, j)]$$

$$h(i, j) = \max [0, f(i, j) - C(i, j)]$$

$$\text{而目的函數爲 } G = \lambda v + \sum_{(i, j) \in A} \{D(i, j) \cdot \max [0, C(i, j) - f(i, j)]$$

$$- d(i, j) \cdot \max [0, f(i, j) - c(i, j)]\}$$

$$\text{因 } f(i, j) = f(i, j, 1) + f(i, j, 2)$$

$$\text{此處 } 0 \leq f(i, j, 1) \leq c(i, j) = c(i, j, 1)$$

$$0 \leq f(i, j, 2) \leq \infty = c(i, j, 2)$$

$$\text{同樣令 } d(i, j, 1) = D(i, j)$$

$$d(i, j, 2) = d(i, j)$$

則目的式可化爲：

$$\min \rightarrow G = \lambda v - \sum_{(i, j) \in A_k} \sum_k f(i, j, k) d(i, j, k)$$

$$\text{求CPM時先求各結點之 } T_E(j) = \max_i [T_E(i) + d(i, j, 1)]$$

$$\text{後等 } FF(i, j, k) = T_E(j) - T_E(i) - d(i, j, k)$$

然後就  $FF(i, j, 1) \leq 0$  的作業構成的網路付碼 (Labeling)，則令初結點 (initial node) 的 Label =  $[-, \epsilon(1) = \infty]$

## CPM COST 計算程式

FORTRAN (4.2)/1505

11/2/75

```

      PROGRAM CPM
C   N=NUMBER OF NODES
C   L=NUMBER OF LINKS
C   SOCE=INITIAL EVENT
C   SINK=FINAL EVENT
C   NCDE=NUMBER OF THE EVENTS STARTING TASK
C   JNDE=NUMBER OF THE EVENT ENDING TASK
C   CRSH=CRASH TIME
C   NORM=NORMAL TIME
C   SLOP=COST SLOP
C   LIJ=DURATION OF EACH TASK
C   NODE=NODE OF REALIZATION TIME
C   IMPLICIT INTEGER (A-H,J-Z)
      INTEGER F,CRSH,SLOP,SOCE,SINK,ORD,TB
      DIMENSION LIJ(1),NODE(100),LABL(1),I(100),F(100,3),I(100),J(100),
      ,CRSH(100),NORM(100),SLOP(100)
      READ(60,100)N,SOCE,SINK,IIF
      100 FORMAT(4I5+I7)
      READ(60,110)(I(K)),J(K),CRSH(K),NORM(K),SLOP(K),K=1,L)
      110 FORMAT(5I5)
      LM=L-1
      DO 200 M=L
      IF (I(M).GE.J(M)) OR. I(M).GT.I(M+1).OR. I(M).EQ.I(M+1).AND.J(M)
      .1.GE.J(M+1)) GO TO 500
      200 CONTINUE
      IF (I(L).GE.J(L)) GO TO 500
      COMPUTE CRITICAL PATH BY NORMAL TIME
      DO 210 K=1,N
      LABL(K,1)=
      LABL(K,2)=
      LABL(K,3)=
      210 NCDE(K)=
      DO 220 M=1,L
      F(M,1)=F(M,2)=
      IM=I(M)
      TEMP=NODE(IM)+NORM(M)
      JM=J(M)
      IF (NODE(JM).LT.TEMP) NODE(JM)=TEMP
      220 CONTINUE
      START ALGORITHM.
      NTV=NTV=
      ORD=0
      TB=NODE(SINK)
      A ROUTINE
      225 LABL(SOCE,1)=SOCE
      LABL(SOCE,3)=IIF
      DO 300 M=1,L
      IM=I(M)
      JM=J(M)
      IF (LABL(IM,1).NE.0) GO TO 240
      IF (LABL(JM,1).NE.0) GO TO 300
      NJ=NODE(JM)-NODE(IM)
      IF (NJ.LT.NORM(M)) GO TO 250
      IF (NJ.LT.NORM(M)) GO TO 250

```

```

    LEX = SLOP(M) - F(I, -)
    IF (LEX .LE. ) GO TO 25
    KF =
    GO TO 255
    C A1 ROUTINE
    255 IF (NIJ .NE. CRSH(I)) GO TO 300
    LEX = INF
    KF = 2
    C A2 ROUTINE
    255 LABL(JM, -) = IM
    LABL(JM, 1) = FI
    IF (LABL(IM, 3) .GT. LEX) GO TO 260
    LABL(JM, 3) = LABL(IM, 3)
    GO TO 265
    265 LABL(JM, 3) = LEX
    265 IF (JM = SINK) 300, 310, 300
    C B ROUTINE
    275 IF (LABL(JM, 1) .EQ. 0) GO TO 300
    NIJ = NODE (IM) - NODE (JM)
    IF (-NIJ .NE. NORI(M) .OR. F(M, 1).EQ. 0) GO TO 280
    LEX = F(M, 1)
    KF = -1
    GO TO 285
    C B1 ROUTINE
    285 IF (-NIJ .NE. CRSP(M) .OR. F(M, 2) .EQ. 0) GO TO 300
    LEX = F(M, 2)
    KF = -2
    C B2 ROUTINE
    295 LABL(IM, -) = J(M)
    LABL(IM, 2) = KI
    IF (LABL(JM, -) .GT. LEX) GO TO 290
    LABL(IM, 3) = LABL(JM, 3)
    GO TO 300
    C B3 ROUTINE
    295 LABL(IM, 3) = LEX
    C C ROUTINE
    300 CONTINUE
    DO 315 K = 1, N
    IF (LABL(K, -) .NE. 0) GO TO 315
    NCDE(K) = NODE(K) -
    315 CONTINUE
    C D ROUTINE
    325 IF (LABL(SINK, 3) .EQ. INF) GO TO 400
    NTV = NTV + LABL(SINK, 3)
    NJ = SINK
    C G ROUTINE
    325 NI = LABL(NJ, -)
    IF (LABL(NJ, -) .GT. 0) GO TO 335
    DO 335 M = 1, L
    IF (I(M) .NE. NJ .OR. J(M) .NE. NJ) GO TO 33
    JJ = -LABL(NJ, -)
    F(M, JJ) = F(M, JJ) + LABL(SINK, 3)
    335 CONTINUE
    GO TO 345
    C G1 ROUTINE
    335 DO 340 M = 1, L
    IF (I(M) .NE. NI .OR. J(M) .NE. NI) GO TO 340
    JJ = LABL(NJ, -)
    F(M, JJ) = F(M, JJ) + LABL(SINK, 3)
    340 CONTINUE
    345 IF (NI .EQ. SOCE) GO TO 400

```

用電子計算機作計劃控制的研究

NJ=II

```

      GO TO 325
C   OUT ROUTINE,
4   DO 410 M=1,L
    IM=I(M)
    JM=J(M)
    LIJ(M)=NODE(JM)-NODE(IM)
    IF(LIJ(M).GT.NORM(M)) LIJ(M)=NORM(M)
410 CONTINUE
    ORD=TB-NODE(SINK)*NTV1+ORD
    NTV1=NTV
    TB=NODE(SINK)
    WRITE(6,155) M,(I(M),J(M),LIJ(M),M=1,L)
155  FORMAT(1H,112X,ZMZ,5X,ZI(M)Z,5X,ZJ(M)Z,8X,ZLIJ(M)Z/)
    WRITE(6,156)
156  FORMAT(1H,I3,2I8,4X,I8)
    WRITE(6,160) ORD, TB, NTV
160  FORMAT(1H,1/5X#ORDINATE = #,18, 5X, #BASE TIME = #,18, 5X,
      #NTV = #,18 /)
    IF( LABEL(SINK).EQ.INF) NTV = INF
    IF( NTV.LT. INF) GO TO 23
    GO TO 5
5   WRITE(6,165) I(M),J(M)
165  FORMAT(1H,1/X,ZERROR AT I(M) = #,18, #J(M) = #,18 /)
5   STOP
END

```

FORTRAN DIAGNOSTIC RESULTS FOR CP11

NO ERRORS

利用 Network Flow Algorithm 能夠求解 PERT/COST 的問題，  
 但由於上列程式很繁，不如利用 IBM 公司或 CDC 公司發售出來  
 的 PERT/COST 通用程式。

(1)若  $FF(i, j, k) = 0$  且  $f(i, j, k) < c(i, j, k)$  則對  $j$  取 label  $[i^+, \epsilon(j)]$

$$\epsilon(j) = \min(\epsilon(i), c(i, j, k) - f(i, j, k))$$

(2)若  $FF(i, j, k) = 0$  且  $f(i, j, k) > 0$ , 則對結點  $j$  取 label 為  $[j^-, \epsilon(i)]$   
此時  $\epsilon(i) = \min(\epsilon(j), f(j, i, k))$

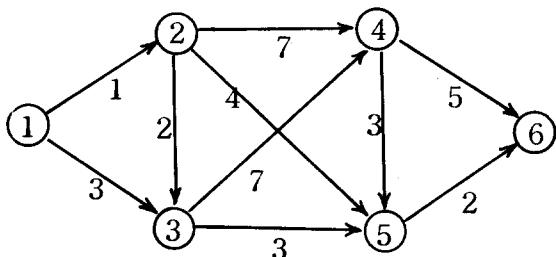
若依上述 Labeling, 而最終結點被 label 為  $[j^+, \epsilon(n)]$  時 (稱為 breakthrough)  
一般結點  $p$  為  $[i^+, \epsilon(p)]$  label 則對  $f(i, p, k)$  加  $\epsilon(n)$ 。

若一般結點為  $[i^-, \epsilon(p)]$  label 則由  $f(p, i, k)$  減  $\epsilon(n)$ ，如此進行 flow 變更  
，到達結點 1 (初結點) 後消除過去的 label 而於新情形下重新 labeling。當最終結點有  
 $\infty$  的 flow 得 breakthrough 時停止計算。

### 1.3 以動態規劃法求要徑 (Critical Path)

#### (a) Arrow Method

〔例〕



利用動態規劃法 (DP) 由向後方向求 Critical Path

$$f_1 = \min(t_{1j} + f_j)$$

$$f_6 = 0$$

$$f_5 = \min(t_{5,6} + f_6) = \min(2 + 0) = 2$$

$$f_4 = \min \left( \begin{array}{l} t_{4,5} + f_5 \\ t_{4,6} + f_6 \end{array} \right) = \min \left\{ \begin{array}{l} 3 + 2 \\ 5 + 0 \end{array} \right\} = 5$$

$$f_3 = \min \left( \begin{array}{l} t_{3,4} + f_4 \\ t_{3,5} + f_5 \end{array} \right) = \min \left\{ \begin{array}{l} 7 + 5 = 12 \\ 3 + 2 = 5 \end{array} \right\} = 5$$

$$f_2 = \min \left( \begin{array}{l} t_{2,3} + f_3 \\ t_{2,4} + f_4 \\ t_{2,5} + f_5 \end{array} \right) = \min \left\{ \begin{array}{l} 2 + 5 = 7 \\ 7 + 5 = 12 \\ 4 + 2 = 6 \end{array} \right\} = 6$$

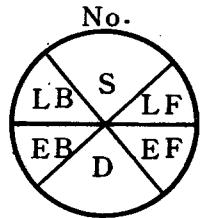
$$f_1 = \min \left( \begin{array}{l} t_{12} + f_2 \\ t_{13} + f_3 \end{array} \right) = \min \left\{ \begin{array}{l} 1 + 6 \\ 3 + 5 \end{array} \right\} = 7$$

Critical Path       $① \rightarrow ② \rightarrow ⑤ \rightarrow ⑥$

(b) Node Method

前法中以圓圖(Circole) 和箭線分別表示事項和作業，本法則以如下記號表示作業，箭線祇表示途徑。

圖中 No.= 作業編號



EB = 最早開始時刻

EF = 最早完成時刻

LB = 最遲開始時刻

LF = 最遲完成時刻

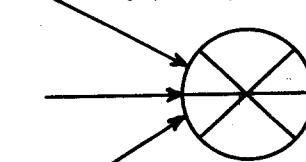
D = 所需期間

S = 總寬裕 (Slack)

要經計算法：

Predecessors

$1 \rightarrow i [(EF)_i]_{ij}$



Successor

$1 \rightarrow k [(LB)_k]_{kj}$

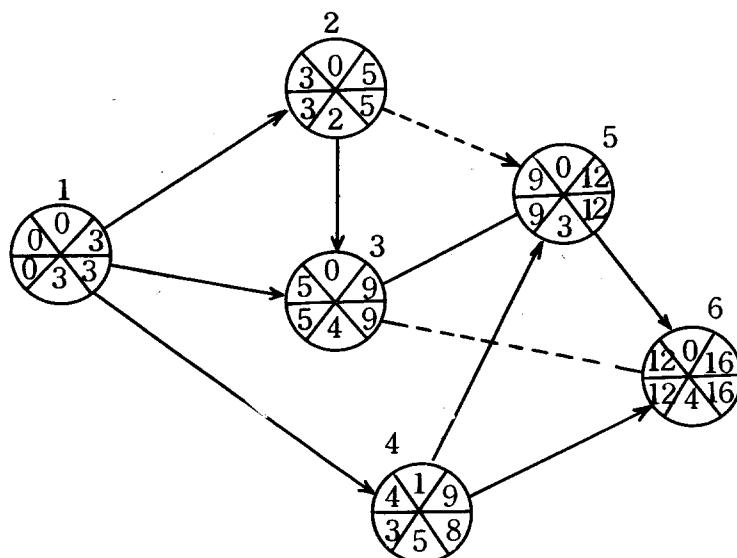
$$(EF)_j = (EB)_j + D_j$$

$$(EF)_j = \max \{D_j + [(EF)_i]_{ij}\}$$

$$(LB)_i = (LF)_i - D_i$$

$$(LB)_i = \min \{-D_i + [(LB)_k]_{jk}\}$$

[例]



$(LF)_6$

||

$$(LB)_6 = \min \{-D_6 + (LB)_7\} = \min \{-4 + 16\} = 12$$

$$(LB)_5 = \min \{-D_5 + (LB)_6\} = \min \{-3 + 12\} = 9$$

$$\begin{aligned}
 (LB)_4 &= \min \left\{ \begin{array}{l} -D_4 + (LB)_5 \\ -D_4 + (LB)_6 \end{array} \right\} = \min \left\{ \begin{array}{l} -5 + 9 \\ -5 + 12 \end{array} \right\} = 4 \\
 (LB)_3 &= \min \left\{ \begin{array}{l} -D_3 + (LB)_5 \\ -D_3 + (LB)_6 \end{array} \right\} = \min \left\{ \begin{array}{l} -4 + 9 = 5 \\ -4 + 12 = 8 \end{array} \right\} = 5 \\
 (LB)_2 &= \min \left\{ \begin{array}{l} -D_2 + (LB)_3 \\ -D_2 + (LB)_5 \end{array} \right\} = \min \left\{ \begin{array}{l} -2 + 5 = 3 \\ -2 + 9 = 7 \end{array} \right\} = 3 \\
 (LB)_1 &= \min \left\{ \begin{array}{l} -D_1 + (LB)_2 \\ -D_1 + (LB)_3 \\ -D_1 + (LB)_4 \end{array} \right\} = \min \left\{ \begin{array}{l} -3 + 3 = 0 \\ -3 + 5 = 2 \\ -3 + 4 = 1 \end{array} \right\} = 0
 \end{aligned}$$

Critical Path ① ← ② ← ③ ← ⑤ ← ⑥

## 二、PCS 及 PMCS 檢討：

### 2.1 PCS/360 概述 [4] :

我們通常把營建工程叫做非重複性工作 (Nonrepetitive Work)，用以與一般的生產工作區別。在執行非重複性工作時，在規劃與監理上，用以幫助經營者完成職責之一套動作與步驟，叫做計劃控制制度 (Project Control System)。IBM公司之 PCS/360 (Project Control System/360) 就是為工程控制制度之資料處理部份工作而設計。茲將該程式之一般性質簡述如次：

PCS/360 可處理五千個作業之網狀圖，PERT·CPM 慣用之箭線圖 (Arrow Diagram) 或居先網狀圖 (Precedence Network 詳後) 均可處理。

PCS/360 本來為官方及民間之營建工程、修理及維護工程而設計，但只要是使用關鍵路線法，任何種類之工程均可使用。

PCS/360 有下述各點之能力：

1. 可處理五千個作業及一萬二千五百個先後關係。
2. 每一先後關係可以錯開。
3. 一星期裡面的工作天數可依每一作業指定。這一點頗有用處，例如混凝土養護在禮拜天不出工亦可進行，一星期之工作天數可指定七天。
4. 正在進行之工作，其進度可用完成百分比表示，或者用剩餘工作天數表示。
5. 對於每一作業之開始與完成日期均可指定編排日期及實際日期。
6. 在日曆裡面可任意指定非工作日 (Non-workdays)。
7. 可算出成本及資源數目。
8. 可提供表格及圖形兩類報告。
9. 卡片或磁碟總檔卷均可使用。在磁碟上最多可儲存五十二個不同之網狀圖。

10. PCS/360 程式係用 FORTRAN 編寫，至少需要三十二 K 之 Core Storage。

## 2.2 計劃控制之概述：

一個控制制度有兩個不同的表徵，第一個表徵為使制度在一個組織裡面發生作用之一套程序。

第二個表徵為用以產生或處理資料之技術或步驟。此項步驟主要有：

1. 工作之規劃：決定什麼工作必須要做，以及它們的次序。
2. 工作之編排：決定某特定工作什麼時候須要做。
3. 資源之分配：對於每一工作項目分配所需之人力或設備。
4. 成本之估計：估算所需總成本。
5. 報告之準備：準備實際監理工作所需之報告。

工作之規劃包含下列步驟：

1. 設定工程之目標。
2. 認明為達成此目標所需之工作項目（作業）。
3. 估計各工作項目所需之時間。
4. 決定完成各工作項目所需之次序。
5. 計算完成該工程所需之時間，認明關鍵路線。
6. 調整計劃，直至滿意為止。

電腦可幫助我們的就是第 5. 項的計算工作。工程越是龐大，越是複雜，電腦越能發揮其力量。

## 2.3 居先網狀圖：

上述步驟裡面，第 4. 項步驟就是繪製網狀圖。PCS/360 推薦的是「居先網狀圖」。居先網狀圖為在 PERT · CPM 之後，近期發展的網狀圖技術，有一點類似於 Flow Diagram，但不盡相同。

居先網狀圖認為兩個工作項目之間，其關係不外三種：

1. B 之開始依賴 A 之開始（始終關係 Start-to-Start Relationship 見圖 2.1）。
2. B 之完成依賴 A 之完成（完完關係 Completion-to-Completion Relationship 見圖 2.2）。
3. B 之開始依賴 A 之完成（始終關係 Completion-to-Start Relationship 見圖 2.3）。

圖 2.4 與圖 2.5 為同一工程分別以箭線圖與居先網狀圖表示者。

居先網狀圖進一步告訴我們它可以表示開始時間之錯開，或完成時間之錯開（即重疊關係 Overlapping Relationship）。這在實際工作上是很實用的。例如圖 2.1 表示 A 與 B 同時開始，若 A 開始一半後才可開始 B，則其關係可表示如圖 2.6；圖 2.7 為其相當之箭線

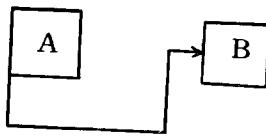


圖 2.1 始始關係

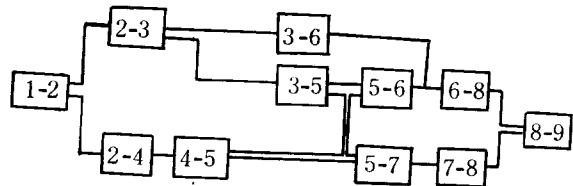


圖 2.5

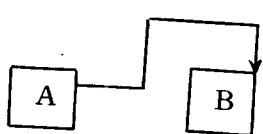


圖 2.2 完完關係

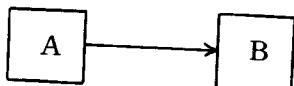


圖 2.3 完始關係

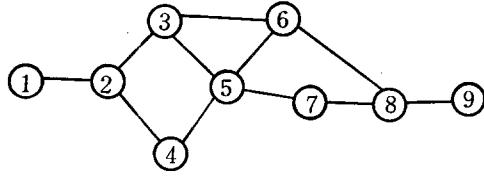


圖 2.4



圖 2.6

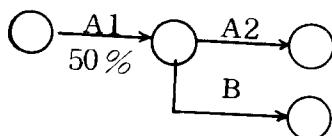


圖 2.7

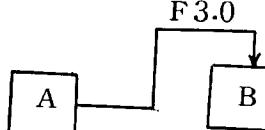


圖 2.8

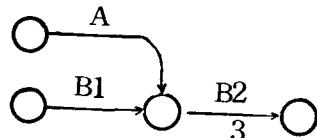


圖 2.9

圖。又如圖 2.2 表示 A 與 B 同時完成，若 A 須在 B 完成之前三天完成，則可表示如圖 2.8；圖 2.9 為其相當之箭線圖。

#### 2.4 PCS/360 程式之說明：

PCS/360 為以網圖為基礎之工程規劃與控制制度內之資料處理部份之電腦程式。  
此電腦程式主要含有下列五種要素：

##### 1. 檔卷之保持：

PCS/360 可容許使用磁碟或卡片之主檔卷。磁碟檔卷之好處有：

- (1)無需保持卡片之檔卷。
- (2)減少更新網狀圖時之電腦使用時間。
- (3)在更新網狀圖印製新報告時，無需再度計算網狀圖之結構。

卡片檔案之好處為不常用之網狀圖，無需佔用磁碟。

## 2. 網狀圖之處理：

PCS / 360 可處理居先網狀圖或一般的箭線圖。最好能使用居先網狀圖。若輸入箭線圖，則程式將把它轉換成居先網狀圖後再加以處理。

PCS / 360 能處理的最大的居先網狀圖為五千個工作項目。每一工作項目均可以有多數的居先工作項目 (Preceding Work Item PWI)，對於一個網狀圖來講，居先工作項目之數目最多可達一萬二千五百個。每一居先工作項目都有標誌表示其關係方式：始始、完完、或完始。並且表示重疊工作之錯開時間，用時間單位或作業期間之百分數表示。

若使用箭線圖，則 PCS / 360 可處理者為最多五千個作業之網狀圖。

注意：PCS / 360 只算出各工作項目之最早開始、最早完成、最遲開始、最遲完成四種日期。至於每一工作應該在什麼時候開始，應由工程主管視人力設備之使用情形來決定。

## 3. 資源之計算：

在 PCS / 360，每一工作項目可指定四種不同之資源，並且對於整個網狀圖來講，最多可以使用一百種不同之資源。資源負荷報告可表示各資源之每日需要量以及每星期需要量與累計數量。

## 4. 成本之估計：

在成本方面，我們可對每一工作項目估計一個總數字，輸入實際成本後，PCS / 360 可印出估計成本與實際成本之對照表。關於成本方面有兩種報告，一種列出所有工作項目之成本，另一種按月列出所需成本。

## 5. 報告：

PCS / 360 可產生十四種報告，計有三種方式，即報表、圖形、與打孔卡片。關於報告將於後節詳細介紹。

另外，PCS / 360 尚有下述兩點特性：

### (1) 日曆：

PCS / 360 擁有的日曆為自一九六五年三月一日至一九九九年十二月三十一日，只要工程期間在這一段時間內，其網狀圖均可以處理。一星期內之工作天數可由使用者決定。若有假期或其他不工作之日子，亦可指定，電腦在處理時能夠在假日或不工作日不安排工作。

### (2) 指定日期：

PCS / 360 可使用五種指定日期。

第一種指定日期儲在檔案內，只供參考用，不與算出之日期產生關係。

第二種指定日期若算出日期比指定日期早時，算出日期無效，所有計算將根據指定日期。若算出日期比指定日期晚，則算出日期有效。當某項工作需要等待別處之人力或機具

時，此種指定日期有用。

第三種指定日期又稱做「命令日期」，經常駕馭於算出日期之上，使算出日期無效。此種命令日期，在需要指定某一工作在某一定日期開始或完成時，可應用。

第四種指定日期經常使算出之「最遲開始」或「最遲完成」日期無效。算出之最早開始及最早完成日期，或經第二種及第三種指定日期修整之最早開始與最早完成日期，並不受第四種指定日期之存在而失效。此種指定日期若應用於某特定工作項目（工程之最後一項或中間之任一項）之最遲開始或最遲完成，因能產生正寬裕或負寬裕，可核對該項工作能否如期完成。

第五種指定日期有兩種使用法。若應用於某一項工作之開始日期，則其作用與第四種指定日期相同，經常使算出之最遲開始日期無效。若應用於某一項工作之完成日期，則能使該項工作之最遲完成日期等於算出之最早完成日期，指定日期失效。

## 2.5 PCS/360 輸入資料：

PCS/360 之輸入卡片共有十二種，分別簡述如次：

屬於工作項目之性質者：

1. 工作項目卡：本卡提供分析關鍵路線時所需之基本資料。主要內容為工作項目編號、工作名稱、期間、一星期內之工作天數、使用日曆等。
2. 里程碑卡：指定工程裡面重要工作之開始或完成。
3. 進度報告卡：用以報告實際進度及修正期間。
4. 編排進度卡：用以輸入指定日期及組織代號。
5. 資源卡：用以輸入成本及資準資料。

屬於網狀圖者：

6. 網狀圖標題卡：用以說明網狀圖標題、工程基準日期、資料日期、處理日期等。
7. 居先工作卡：本卡用以表示工作項目之間之先後關係。
8. 日曆卡：用以指定假日及非工作日。
9. 資源說明卡：用以說明資源之代號及其名稱。
10. 資源歸類卡：將資源歸類，供印製報告者。

屬於輸出報告者：

11. 輸出請求卡：用以指定所需報告名稱。

屬於處理者：

12. 處理控制卡：本卡用以命令程式之邏輯流程。

## 2.6 PCS/360 輸出報告：

PCS/360 可產生十四種報告，分述如下：

1. 桿狀圖：可表示每一工作項目之進度，及其寬裕時間。
2. 摘要桿狀圖：表示每一負責單位所經辦多項工作之開始與完成。
3. 里程碑報告：表列所有里程碑之名稱、算出日期、指定日期、及寬裕時間。
4. 進度報告：表列所有工作項目之最早開始、最遲開始、最早完成、最遲完成、以及寬裕時間。
5. 工作情況及進度報告：表列各工作項目之最早開始與最遲完成日期、剩餘期間、以及居先工作。
6. 工作情況卡：為一組打孔卡片，可用於其他程式之輸入資料。
7. 資源分配報告：表列每一工作項目之按日資源使用量及日合計數。
8. 資源利用報告：為一種圖形，表示每週資源使用量及累計數。
9. 總成本報告：表列各工作項目之估計成本與實際成本。
10. 月成本報告：表列每一月份內之各工作項目之估計成本與實際成本。
11. 居先工作表：表列每一工作項目之所有居先工作項目及其最早與最遲完成日期。
12. 日曆：表示假日與非工作日。
13. 主檔卷：列出所有儲存於磁碟主檔卷之資料。
14. 後援主檔卷：主檔卷之內容可輸出於 SYSPCH，容許日後之修訂或印製報告。

## 2.7 PMCS 概述 [5] :

PMCS 係 CDC ( CONTROL DATA CORPORATION ) 公司之模式程式 (Project Management Control System) 本制度可產生如下三種報告：

### (1) 作業報告 (Activity Report)

1. 表列作業已屆開始日期或正進行中者，居先結點 ( 第 1 欄 ) 前印出 I 。
2. 作業時間欄 ( 第 5 欄 ) 有三種情形：
  - 未開始或正進行中之作業為預估工作週數。
  - 完成之作業為實際工作週數。
  - 選用 TIME - NOW - REPORTING 製表時，所有應開始而未開始或正進行中之作業必需輸入其尚需工作週數 ( 自製表日起算 )，其作業時間乃已過週數與尚需工作週數之和。
3. 最早完成日期欄 ( 第 6 欄 )，已完成者此欄空白。
4. 察際完成日期或預計完成日期欄 ( 第 8 欄 )。
  - 已完成作業列實際完成日期，前面印出字母 A 。
  - 進行中之作業為空白或預計完成日期 TS 。
  - 如係由規定日期算出之 TS 或係限定完成日期，則於日期前分別印出字母 B 或 R 。
5. 總寬裕欄 ( 第 7 欄 ) 印 TF 之週數。

6. 剩餘時間欄（第 10 欄）。
  - 已完成之作業為空白。
  - 進行中之作業為產生報表日期至作業之 TE（最早完成日期）止尚餘可工作週數（不包括 SLACK）。
7. 作業報告顯示之情況：
  - 剩餘時間（REMAINLNG TIME）為負時表示作業延誤及使用 SLACK，應注意佔用 TF（干擾寬裕）情形而採適當措施。
  - TS 或 TF 在報告截止日期之前者顯示作業延誤已用去部分 SLACK，應注意有無佔用 IF 並採適當措施。
  - 總寬裕 SLACK 為負數時顯示作業延誤並超過 SLACK，故應趕工並作 FOLLOWUP。

(2) 標準重點報告 (Standard Milestone Report)

1. 內容與(1)大致相同，但祇以預先選定之若干關鍵結點 (KEY EVENT) 為控制目標。
2. 預先選定之關鍵結點應給以關鍵碼 (MILESTONE CODE) 用作列表順序及印表時的分段指標，碼為三位數。
3. 本報告中之 SLACK 指某結點的 SLACK (即  $TL_j - TE_j$ ) 與作業報告中每個作業的 SLACK 並不盡同。
4. 對網圖的一部或全部 EVENT 按其重要性賦予 1-9 的等級稱為 STRUCTURE LEVEL，製表時在指定等級以內之 EVENT 方予列表。

(3) 記時重點報告 (TIME SCALE MILESTONE REPORT)

1. 內容與(2)略同，但型式不同，本表右半第一行橫向印出月份及年度，表中日期欄僅印出日期而另以字母代表日期別印在相當之月份下：
2. 日期別包括：
  - E 代表 TE (最早完成日期)
  - S 代表 TS (預計完成日期)
  - A 代表 TA (實際完成日期)
  - L 代表 TL (最遲完成日期)
  - M 代表管理界層限定完成日期

(4) 幾種選擇：

上述三種報表之產生許可作下列數種選擇：

1. 印出工作日曆以供對照。
2. 在 FOLLOWUP 時選用現時報告。
3. 按統計數學計算完成機率並在作業報告之 PROB 欄印出完成機率。

4. 作業報告內作業印出：先後可指定下列之一為主順序，另一為副順序。
  - 居先結點                  • 後繞結點                  • 總寬裕
  - TE                          • TL                          • 執行單位別
  - 會計賬號（不得作為主順序）
5. 印表時可刪除某些當時不需考慮之作業，例如：
  - 已完成作業                  • 某日期前之作業                  • TE, TL 在某日以後者
  - 總寬裕大於若干週者（以 TF 為主順序）。

## 2.8 PMCS 制度設計

(1) 先設計輸入資料格式：

輸入卡片分為日曆卡、控制卡及作業卡三大類：

1. 日曆卡 (Calendar Card)：分為四種（圖 2.10）
  - a. K 卡：用以制定一“十年日曆”。
  - b. L 卡：用以指明每週固定的非工作日。
  - c. 日曆修改卡 (Calendar Modification Card)  
將制定的日曆中原來的工作日（或週期）修改為非工作日（或週日期）或將非工作日（或週期）修改為工作日（或週期），若修改的日期未示明年代則表示每年該日均得修改。
  - d. M 卡：表示日曆修改卡到此為止。
2. 控制卡 (control card)：也有四種（圖 2.11）
  - a. X 卡：包括建檔代號及表頭資料。
  - b. Y 卡：包括製表（資料截止）日期，報告送出日期等。
  - c. W 卡：包括選擇不同排列順序之報表及限制報告時間範圍，可選擇七種作業報表，六種重點報表。
  - d. Z 卡：表示作業卡及結點卡到此為止。

日曆卡

K 卡

卡 別 1		開 始 月 910		開 始 年 1920		星 期 30		每週工 作日數 40		選印工 作日曆 50
K										

L 卡

卡 別 1		每週非工作日						
		10			20			30
L								

日曆修改卡

			特定之非工作或工 作日或週期開始日								週 結 束 期 日		
			代 碼	月	日	年			月	日	年		
1	3	4	5	6	7	8	9	10	15	16	17	18	19

M 卡

卡 別 1	
M	

圖 2.10

用電子計算機作計劃控制的研究

X 卡  
控制卡

卡別	網號	網圖	名稱	副案號	計劃案號		報告單位	期別
					36	37		
X	1 2	4 5			49	49	64 65	60

Y 卡

卡別	月	日	年	月	日	年	選擇						
							d <sub>1</sub>	d <sub>2</sub>	d <sub>3</sub>	d <sub>4</sub>	d <sub>5</sub>	d <sub>6</sub>	
1 2	3 4	5 6	7 8	9 10	11 12	13	15	16	17	18	19	20	
Y													

d<sub>1</sub>:以舊主機磁帶為輸入

d<sub>2</sub>:保留新主機磁帶

d<sub>3</sub>:製作出費率/cost 使用之磁帶

d<sub>4</sub>:使用 UNDER TIME NOW 報告

d<sub>5</sub>:計算及列印或然率

d<sub>6</sub>:輸入作業時間以週及日為單位

2-14

W 卡

卡別	業報												重報
	一	二	三	四	五	六	七	一	二	三	四	五	
1	31	36	37	38	39	34	35	36	37	38	39	30	23
W	4	8	13	18	20	24	28	32	36	40	44	52	75

S<sub>1</sub>:主排列代碼

S<sub>2</sub>:副排列代碼

S<sub>3</sub>:已完成作業不列印

m<sub>1</sub>:重點級別

m<sub>2</sub>:跳質空制備

m<sub>3</sub>:報表格式

Z 卡

卡別	業報												重報
	一	二	三	四	五	六	七	一	二	三	四	五	
Z	1	2											

(圖2-11)

## 3. 作業卡 (Activity Card) 及結點卡 (Event Name Card)

- a. 作業卡：包括各作業的資料，作業間的關係等。每一作業由一張卡片表示，當主檔建立後，更新時僅內容有改變之作業需要卡片。

欄位	說	明
1	修正字號：A 新增作業 D 刪除作業 R 更新主檔中現有作業內容	
2 - 5	重點指標：前三位為重點號碼，後三位為重點級別	
6 - 11	居先結點號碼	
12 - 17	後續結點號碼	
18 - 21	樂觀時間估計	
22 - 25	作業時間估計	
26 - 29	悲觀時間估計	
30	說明 31 - 36 欄日期的性質 A 實際完成日期 R 預定完成日期 空白 預計完成日期 B 預定開始日期	
31 - 36	(月、日、年)性質由前欄代號決定	
37 - 64	作業名稱	
65 - 76	作業帳號	
77 - 80	執行單位代號	

- b. 結點卡：包括關鍵結點：資料僅出現於重點報表上。

欄位	說	明
1	修正字碼 A 新增關鍵結點 D 刪除關鍵結點 R 改變結點資料內容	
2 - 5	重點指標	
6 - 11	結點號碼	
12 - 17	結點號碼 (與上項同)	
18 - 30	空白	

31 - 36	預計完成日期
37 - 64	結點名稱

## c. 合計卡 (Batch Total Card)

爲確保作業卡及結點卡各數字欄製卡之正確性，於一組(資料表一頁)作業卡及結點卡之後置一合計卡，填寫該組(頁)中資料各數字欄之總和，以便機器查錯時核計。本卡僅於檢查時使用，正式處理時必須抽出。

欄位	說明
1	0
2, 3	頁次(指資料計表頁次)
4, 5	行數(指該頁所填作業卡、結點卡數)
6 - 11	重點報告指標
12 - 19	居先結點號碼
20 - 27	後續結點號碼
28 - 33	樂觀作業時間估計
34 - 39	近似作業時間估計
40 - 45	悲觀作業時間估計
46 - 48	月
49 - 51	日
52 - 55	年

## (2)原始資料之準備與傳送：

1. 設計網圖與填寫資料製卡表。
2. 建立主檔。
3. 檢討網路與修正。
4. 隨時收集計畫之評核資料。

## (3)決定輸出資料與適用範圍：

## 2.9 PMCS 制度流程：

本制度於處理時可分為三大步驟：

1. 檢查製成之作業卡及結點卡內容有無錯誤，列印資料清單，錯誤部份註以記號以供查閱改正。
2. 網圖作業建檔，讀入程式及資料卡片作必要之計算，印出所需之報表及主檔磁帶。

3. 網圖作業更新：讀入程式（PERT TIME 2-1）、舊主檔磁帶（Old Master Tape）及資料更新卡片，增加，刪除舊主檔中之作業，結點或改變舊主檔中已有之資料，印出所需之報表並將主檔磁帶更新，產生新磁帶（New Master Tape）。
- 建檔，更新工作流程圖及輸入卡片組如圖 2.12~圖 2.15 所示。

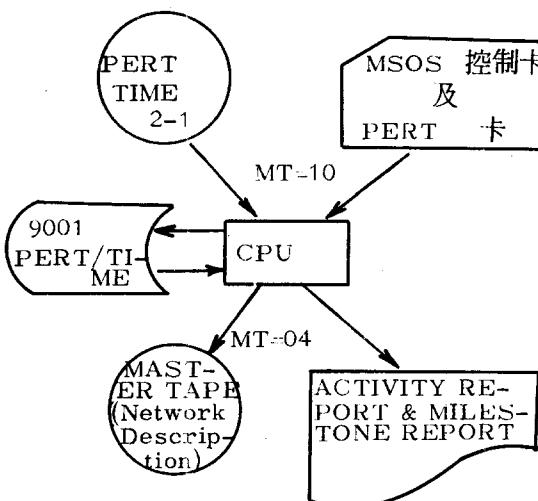


圖 2.12 建檔工作簡圖

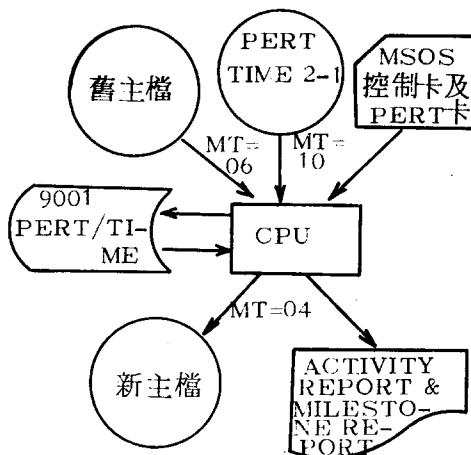


圖 2.13 更新工作簡圖

## 2.10 PCS 及 PMCS 的檢討：

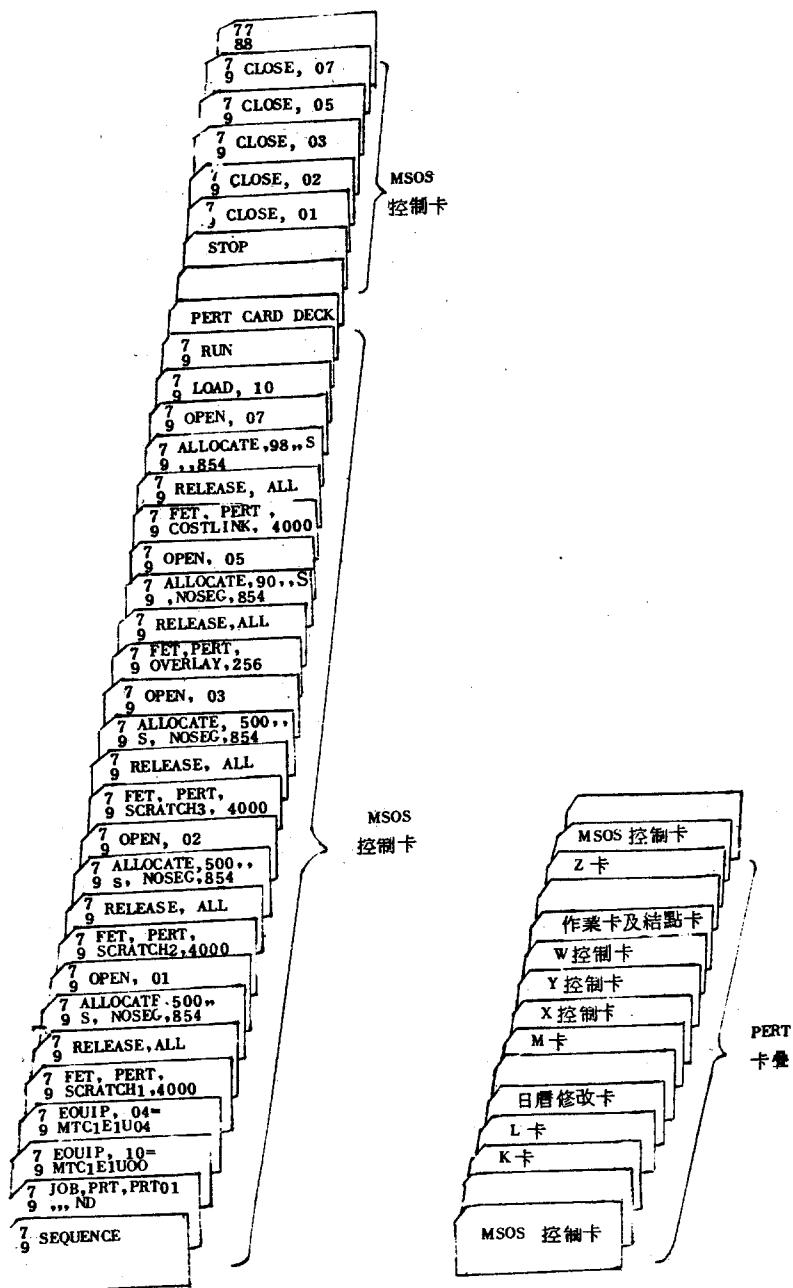
綜上所述，我們可發現採用 PCS 或 PMCS 有如下幾種效益：

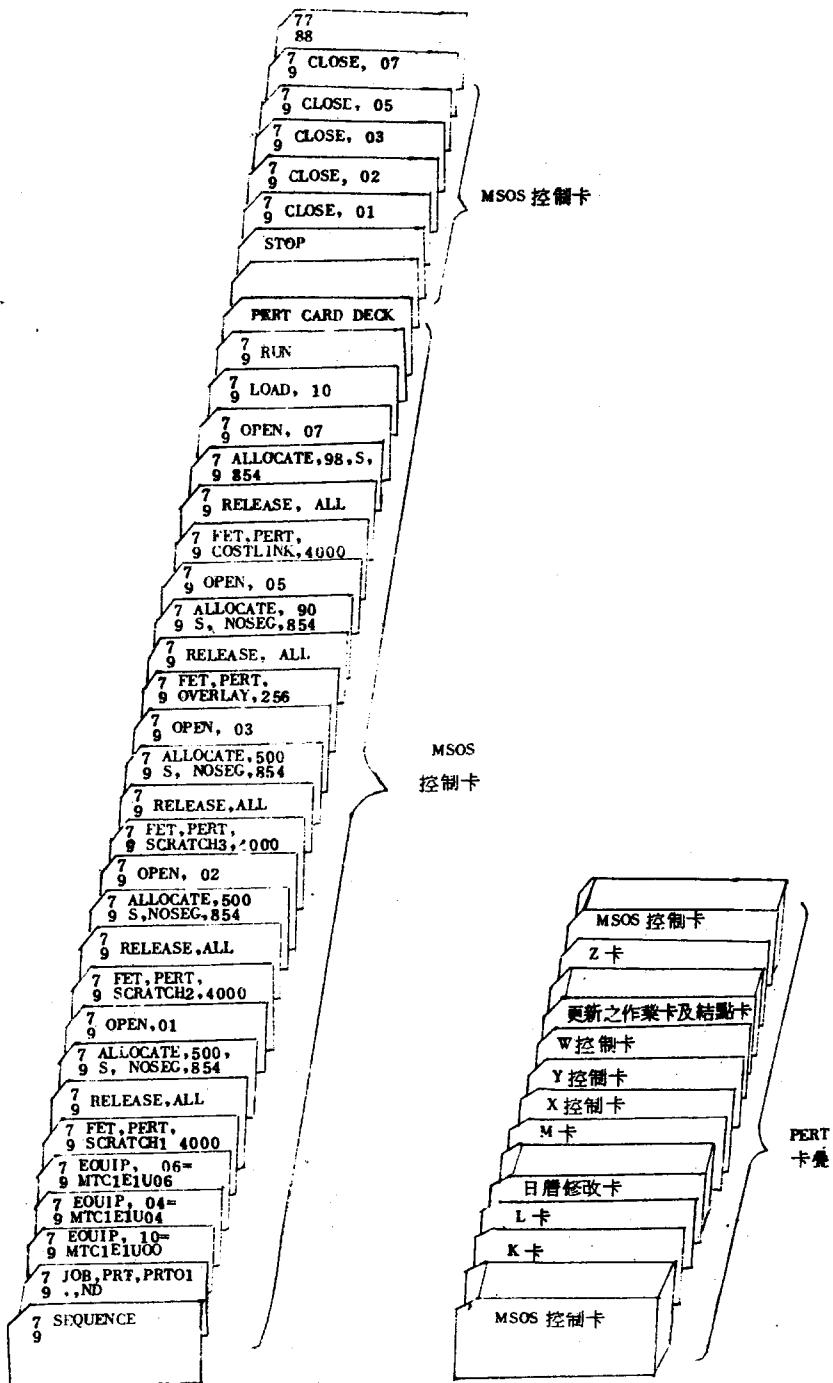
1. 計算迅速正確。
2. 報表資料容易閱讀：人工 PERT 作業之最早、最遲完成日多以自計劃開始後若干日表示，電腦作業報表乃直接以日期（日、月、年）印出，不必經換算。本制報表對各種不正常狀況（延誤）均有診斷訊息（Diagnostic Message）或符號印出，可令人一眼看出問題的所在。
3. 人工作業對複雜網路常有顧此失彼現象，但依 PCS 則顧及全盤計劃，不會遺漏任何有關之作業。
4. 可產生多種報表。
5. 可反復進行計劃之檢討或修改。
6. 隨時可得最新之進度報告，更新容易。

採用 PCS 或 PMCS 時應注意的事項如下：

1. 無論是 PCS 或 PMCS 作業均需 32 K 以上容量，PCS / 360 可處理 5000 作業，PMCS 可容納 4095 個結點，超過時亦可藉擴大容量或濃縮網圖等方式加以處理。

用電子計算機作計劃控制的研究





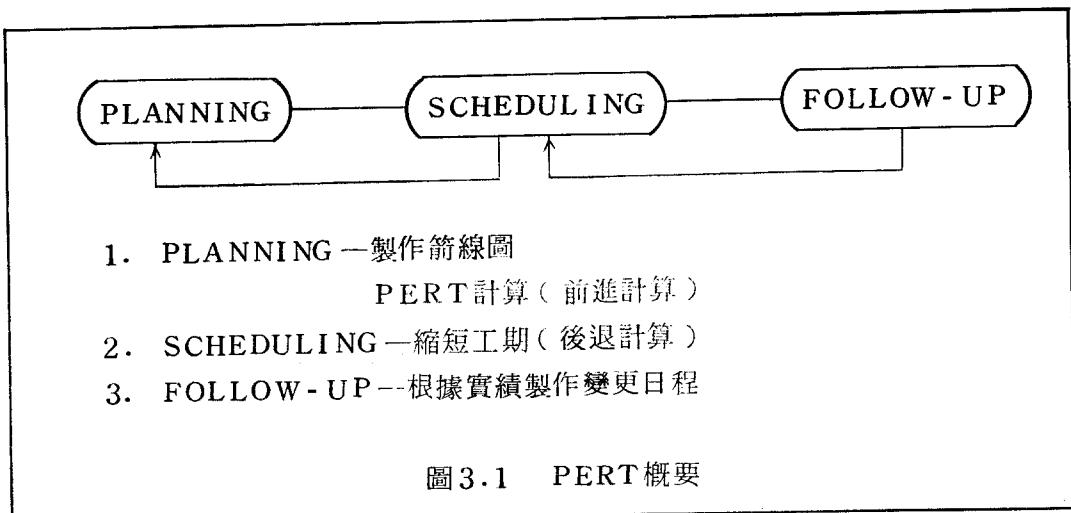
(圖 2-65) PERT/TIME 更新輸入卡片組

2. 何種情況下放棄人工處理而採用電腦以增進效率？通常依下列因素而定：
  - (1)網圖數據：人工處理 PERT 的數據以不超過 100 為宜，超過比數則宜用電腦 PCS。
  - (2)工期：中長程計劃工期在半年以上者可採用電腦 PCS。
  - (3)工作項目：重大工程或計劃，其工作項目繁複或承包廠甚多時可利用電腦。
  - (4)報告頻率：上級所需之報告期限短促，人工難以配合者可採用 PCS。
3. PCS 或 PMCS 的程式週密完整，但輸出報表是否正確可靠，仍賴輸入資料（人工產生）而定，故正確資料管理列為首要條件。
4. PCS 或 PMCS 提供之報表不要使之成為例行公事，應隨時注意配合管考制度，嚴密檢討，無求改革。

### 三、縮短工期法：

#### 3.1 工期（日程）縮短之概念

PERT 由圖 3.1 所示的三部分構成，其中關於計劃（PLANNING）及追從管制（FOLLOW - UP）的計算，各電腦廠家均擁有一套應用程式，但對縮短工期（SCHEDULING）方面的機能較差。但在現實的應用上，將前進計算所求得之完成時間加以縮短，而製作符合要求工期的日程表的情形很多。



此時為了找出應加縮短的作業，可藉後退計算法找出限制路線（Limit Path，則由具備負的 Total Float 的作業構成的路線）而以這些路線為中心檢討縮短方法。所謂後退計算法與前進計算法相反，係將最末作業的完成時刻換算為要求完工期限，然後溯回前面計算者。

茲取簡例說明運算步驟：

### 3.2 Source Data 的製作：

由圖 3.2 所示箭線圖（附有作業名稱、所需時間、結點號碼）將各項目列成如圖 3.3 所示之明細表，將此打孔於卡片（或紙帶）作為進行 PERT/TIME 計算（前進計算）之輸入，而把結果輸出於磁碟，紙帶或卡片。本法的第一步驟是由這些結果依圖 3.4 所示格式，以「逆順序」把後續事項、居先事項分類（Key 項目是由大往小）而輸出於紙帶、磁碟或磁帶，稱之為 Source data。

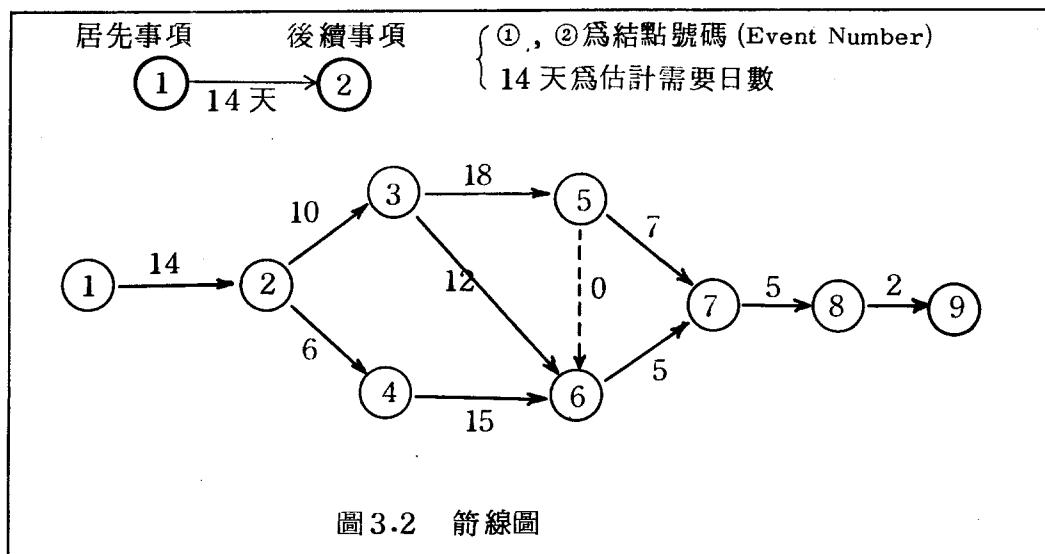


圖 3.2 箭線圖

事項號碼		作業名稱	估計需要天數
前	後		
0001	0002	人員訓練	0014
0002	0003	系統設計	0010
0002	0004	房間佈置	0006
0003	0005	程式製作	0018
0004	0006	機器搬入	0015
0005	0006	打卡	0012
0005	0006	Dummy	0000
0005	0007	製作使用書(手册)	0007
0006	0007	除錯	0005
0007	0008	試機	0005
0008	0009	並行處理	0002

圖 3.3 PERT 作業明細表一例

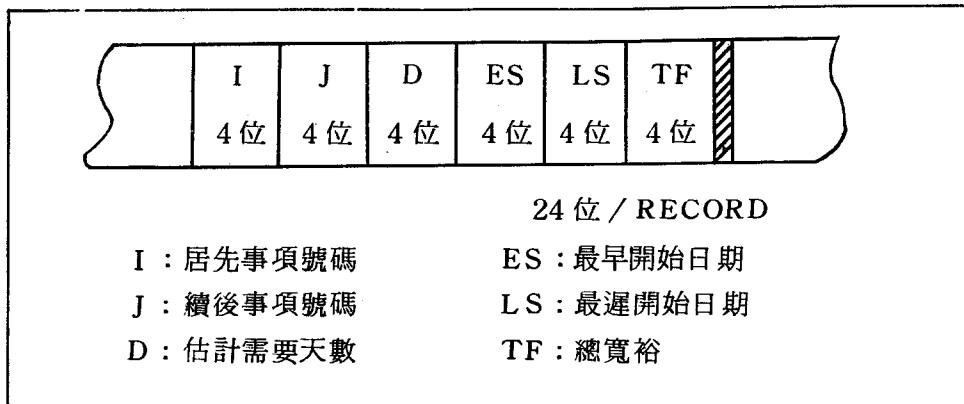


圖 3.4 SOURLE DATA 的格式

### 3.3 Limit Time 的計算：

在此先定義「限制時間」(Limit Time) 則於後退計算法中各結點的續後作業必須開始的界限（圖 3.5）。

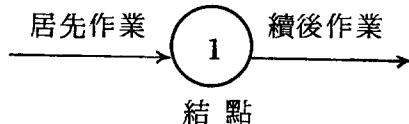
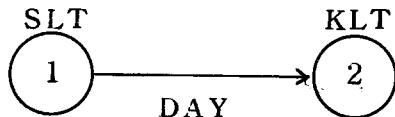


圖 3.5 Limit Time 的定義

設於圖 3.2 中整個計劃的完成時間由 PERT 算出為 56 天。現在重考慮將工作縮短 7 天的場合，就結論而言需找出限制路線 (Limit Path) 藉此考慮縮短方法，為之，需先計算對應「希望完成天數 49 天」的各結點的 Limit Time；其公式如下：

$$SLT = KLT - DAY \quad (3.1)$$



式中 SLT : 居先結點之 Limit Time

KLT : 續後結點之 Limit Time

DAY : 估計需要天數

結點	作業	需要天數	計算式	Limit Time
⑨	⑨	(縮短天數) 7	(完工日) $56 - 7 = 49^*$	49
⑧	⑧ → ⑨	2	$49 - 2 = 47^*$	47
⑦	⑦ → ⑧	5	$47 - 5 = 42^*$	42
⑥	⑥ → ⑦	5	$42 - 5 = 37^*$	37
⑤	⑤ → ⑦	7	$42 - 7 = 35^*$	35
	⑤ → ⑥	10	$37 - 0 = 37$	
④	④ → ⑥	15	$37 - 15 = 22^*$	22
③	③ → ⑤	18	$35 - 18 = 17^*$	17
	③ → ⑥	12	$37 - 12 = 25$	
②	② → ③	10	$17 - 10 = 7^*$	7
	② → ④	6	$22 - 6 = 16$	
①	① → ②	14	$7 - 14 = -7$	-7

圖 3.6 Limit Time 的計算

當網路圖很大時，由 1 結點分出兩個以上作業的情形很多。此時利用 (3.1) 式分別計算後，取最小值為該結點之 Limit Time，圖 3.2 的 Limit Time 的計算如圖 3.6 所示。

包含工期縮短機能的 PERT 系統如圖 3.7 所示，則將縮短天數由外面 Key-in，與 SOURCE DATA 求最終作業之 Limit Time，然後依照圖 3.6 進行計算，並記錄於 Random file；這是因為結點之 Limit Time 由 (3.1) 式知，係反復地出現為居先事項和續後事項，（則一結點對後一作業為居先事項，但對前一作業則為續後事項），故為便於讀寫 (R/W) 之不規則進行而使用 Random File。

計算 Limit Time 的程式要點如下：

- 求最終結點之 Limit Time

= 最終結點的完成日期 - 縮短天數

- 從 drum (或 disk) 讀後續事項之 Limit Time，而與所需時間求先行事項之 Limit Time 寫入 drum (或 disk)。

- 如先行事項相同時採用「最小值」為 Limit Time。

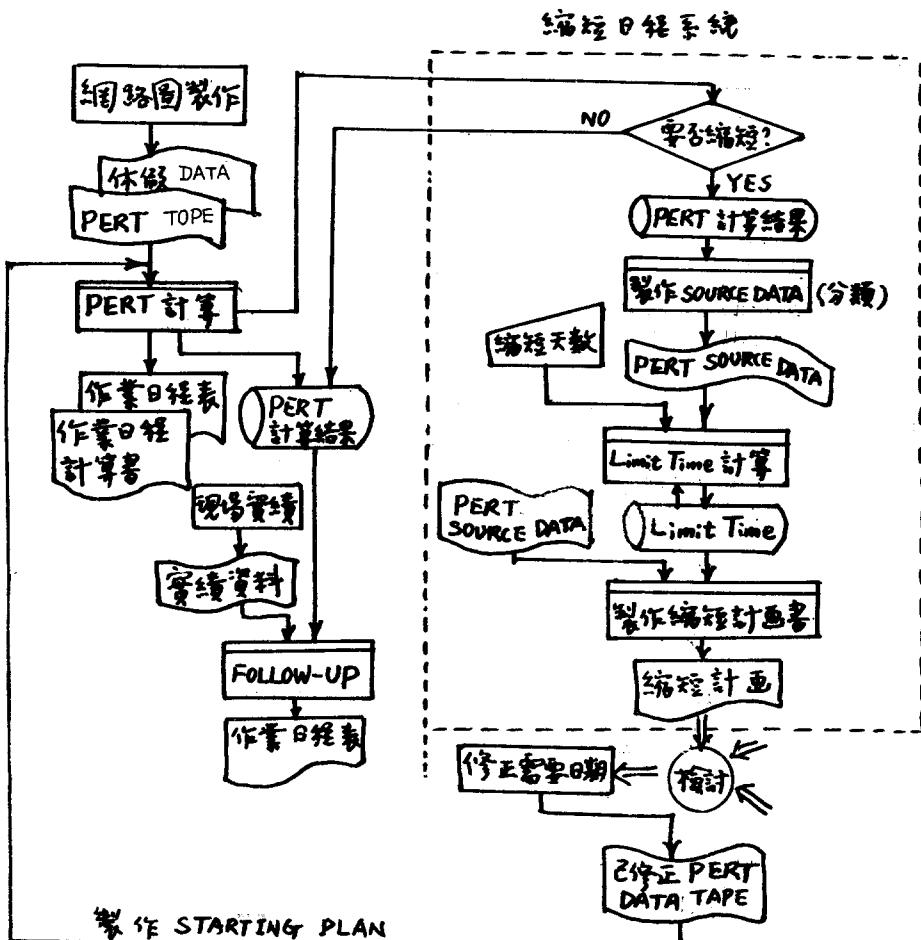


圖 37 包含縮短工期機能的 PERT 系統

4. Random file 上的格式如圖 3.8 所示。

結點號碼 4 位	Limit Time 4 位加符號	
9 位 / RECORD		

圖 3.8 Limit Time  
的 file 格式

### 3.4 限制路線之尋找：

由 drum 或 disk 上所寫 Limit Time 與 SOURCE DATA 可將各作業的總寬裕 (Total float) , 限制路線 (Limit Path) 要徑 (Critical path) 列出。

$$TF = KLT - SES - DAY \quad (3.2)$$

式中 TF : 總寬裕

KLT : 繢後事項之 Limit Time

SES : 居先事項之最早開始日期

DAY : 作業的估計需要時間

就圖 3.2 進行計算之結果如圖 3.9 所示。

### 3.5 STARTING PLAN 的製作：

由圖 3.9 的表可發現在兩個限制路線。

- (a) ① → ② → ③ → ⑤ → ⑦ → ⑧ → ⑨
- (b) ① → ② → ③ → ⑤ → ⑥ → ⑦ → ⑧ → ⑨

需分別縮短 7 天和 5 天。

EVENT-NO		ACTIVITY	START	LIMLT	TOTAL	PATH
I	J	DAY	EAR	LAT	TIME	
8	9	2	54	54	49	7-
7	8	5	49	49	47	7-
6	7	5	42	44	42	5-
5	7	7	42	42	42	7-
5	6	0	42	42	37	5-
4	6	15	20	29	37	2
3	6	12	24	24	37	1
3	5	18	24	24	35	7-
2	4	6	14	14	22	2
2	3	10	14	14	17	7-
1	2	14	0	0	7	7-



\*\* ..... LIMIT-PATH                            \* ..... CRLTLCAL PATH

EAR... EARLIST START TIME    LAT... LATEST START TIME

圖 3.9 縮短日程計算書

假設經檢討後可將③→⑤縮3天，將⑤→⑦縮4天，將⑥→⑦縮2天，那麼我們將此三作業的估計需要時間加以修正重新進行PERT 計算，則該日程必為符合目標的計劃，以後定期地對照實績作 FOLLOW- UP 則可。

#### 四、PERT/LOB

##### 4.1 LOB

所謂LOB (Line of Balance 平衡線) 為在生產期間對有關時間與工作進度，提供消息，資料收集、衡量與報告的一種以管理為導向的圖式工具，表示出計劃中的各相關作業的進度，進行狀況，時間與階段。

這些資料提供了管理者工作實際進度與預定進度之比較。除此，管理者可以適時獲得在重要性作業步驟中的預定進度，目前狀況與計劃的比較，檢查生產是否有超前或落後之現象，而採取適當因應措施。

LOB 與其他的控制技術不太相同，它主要利用在生產程序方面，自原料進廠開始，到生產交貨止的逐站重點控制。主要在調整與監督製程中的原料、零件、組合件的流程與生產量，使配合各製造階段交貨之要求。

##### 4.2 PERT/LOB

PERT/LOB 的應用是組合(Integrate) PERT 的計劃效能和 LOB 的控制效能而成。它對於某產品的由發展階段演進為生產階段的過渡時期和生產開始階段可提供密切的計劃與控制；其效能有如下數端：

1. 着重於關鍵性的生產作業之計劃與控制，如此可減少計劃或任務之複雜性。
2. 對處理生產程序上易生問題之區域，和處理重複性作業的現象一樣有效。
3. 幫助管理者對於生產量有效地計劃控制，如期交貨運送。
4. 幫助管理者，預測任何時期中的控制點之計劃與所需成本，和本來計劃，及目前需努力重點和方向。
5. 可以同時分析生產中時間與成本因素。

總之，PERT/LOB 提供了應什合約的彈性需要和預訂計劃相配合的。作業衡量技術。PERT/LOB 作業的階段中作業內容和相關因素如下表所示。

階 段	作 業 內 容	相 關 因 素
1. 計 劃	設備發展到生產過渡時期，包括工程試驗模型，試驗樣品，以及生產所需作業程序和工作站從屬關係。	工作解析結構和網路工作
2. 估 計	網路作業到達時間之估計，和考慮所需生產量。	學習曲線、生產率、重複性作業計劃
3. 途程安排	網路計算 (TE, TL 和寬裕時間)，分派工作，依據可用機器設備和交貨時間表，調整工作負荷量，交貨日程，平衡生產率。	生產負荷表、直接預訂日程表、預訂目標計劃、生產計劃、作業報告
4. 控 制	檢查生產物件是否配合預訂進度完成。	平衡線、生產狀況報告

#### 4.3 PERT/LOB 作業程序：

##### 1. 工作解析 (Work breakdown)

先確定計劃目標，將目標又分得很細密，然後，依此目標的細分，組成計劃的網路結構。〔6〕

最低級的細分生產件，分入各工作組 (Work packages) 中，形成基本的計劃與控制工具。

##### 2. 重複性作業時間估計 (Repetitive time estimates)

從機器能力、生產週期大小，機器負荷量和學習因素等資料算出重複性作業每一批次作業時間估計。每次生產批量 (Batch) 的大小，可從單一組件到全部生產所需數量，所以時間的估計影響批量的大小。

在重複性作業中，可能有些批次在生產時間上重疊，即前一批次尚未完工，次一批次又開始作業，故需估計各批量所需時間，各批次的最小間隔時間，批量大小及計劃中的生產量而設計計劃。

##### 3. 網路的建立 (Network construction)

PERT/LOB 可以同時描出發展時期到生產期間的過渡時期的作業活動形式，重複性作業活動 (activity) 由作業箭線表示，而重複性作業事件 (event) 由帶有 “×” 號的圓圈代表。

##### 4. 時間計算 (Time calculation)

網路的計算和 PERT 系統相同。

唯一不同的是PERT/LOB的重複性作業配合各工作站整體計算，而非如PERT/TIME單一各站計算。即TE, TL及TF(總寬裕)是以重複性作業的每一批次計算，因為頗為繁複，需借重電子計算機。

#### 4.4 網路模型：

LOB之數學模型可表示如下，已知一有限集合I(工作單元之集合)受優先順序，一數值C(循環時間)及函數t(正實數值)的限制，求I之各子集(S<sub>1</sub>, ……S<sub>J</sub>)，但須滿足下列條件。

1.  $\bigcup_{j \in J} S_j = I$  (工作站的集合包含全部n工作單元)
2.  $S_i \cap S_j = \emptyset$  (一個工作單元不可能分派至兩個以上工作站)
3.  $t(S_j) = \sum_{x \in S_j} t_x, t_x \leq C, j = 1, J$  (工作站時間≤循環時間)
4. 如  $x < y$  且  $x \in S_i, y \in S_j$ ，則  $i \leq j$  ( $x$  優先先於  $y$ )
5.  $\sum t_j = [C - t(S_j)]$  為最小 (使總閒置時間為小)

所求數學模型是將目標函數(5)變為以網路圖表示之最短路徑問題(Shortest route problem)

$$\text{因為 } \sum_{j=1}^J [C - t(S_j)] = \sum_{j=1}^J C - \sum_{j=1}^J t(S_j) = JC - \sum_{j \in J} t_j \\ = JC - \text{常數}$$

故可推斷，欲使總閒置時間為最少，亦即使工作站數目為最少。

#### 4.5 討論：

PERT/LQB的管理報告(Reports for management)有下列三種格式。

1. 工作要站報告(The milestone report)，如同PERT形式，乃有關各工作要站所完成的組件計劃。
2. 生產狀況報告(The production status report)，乃配合PERT/LOB所做網路，加上各控制點的計劃完成日期，供作高階層管理之參考。
3. 作業活動報告(The activity report)：PERT的作業活動報告？多報告生產率批次的資料供中層管理者知道那些工作配合預訂計劃的情形。

PERT/LOB因結合了PERT的計劃功能和LOB的控制功能，有助管理者有效執行計劃與控制，可成為新產品開發的最佳利器，它是PERT/COST結合而成的PERT/LOB/COST可稱為PERT系統第四代(TIME-COST-LOB-LOB/COST)對國防工業、太空發展、電腦工業、和特殊設備製造計劃有重大用途，故值推廣應用。

#### 五、討論：

本研究對利用電子計算機作計劃控制的方法，首先介紹PERT/TIME與PERT/

COST 計算程式的步驟，並舉例說明。除 PERT 的一般概念於附錄中扼要說明外，並以DP法試解 PERT 得到方便的結果。

IBM和CDC公司發展的PCS和PMCS的一般使用法及注意事項，於第二章曾作重點介紹，此二程式，雖稱方便，但因所需記憶容量較大，故一般場合祇得利用單獨的PERT/TIME或PERT/COST應用程式。因PCS和PMCS中對Scheduling(縮短工期)方面機能較弱，本研究提出新創的後退計算的工期縮短法，方法簡單，應用方便。

一般稱計劃控制常以非重複性工作為對象，但對重複性工作如生產活動中LOB發揮很大的控制機能，故第五章對PERT/LOB亦加介紹。

本研究以計算機應用為着眼點，和專以PERT為主的研究不同，對PERT/TIME，PERT/COST及PERT/LOB等總合加以檢討，尚屬首創，由於實地應用經驗少，故討論有不夠深入之處，敬請閱者鑒諒。

本研究中PERT/COST程式費時月餘，始告完成，利用流量問題解答，富有創意，與工期縮短法及動態規劃法解CRITICAL PATH問題等均值得供人參考。

#### 謝 辭

本研究承國家科學委員會資助，特此誌謝。

#### 參考文獻

- (1) 楊維楨：PERT/CPM講義，經濟部專業人員研究中心講義，63年8月。
- (2) Harry F. Evarts, Introduction to PERT, Allyn & Bacon Inc. Boston, 1964
- (3) L. R. Ford, Jr. and D. R. Fulkerson, Flows in Networks, Princeton Univ. Press, Princeton New Jersey, 1962
- (4) IBM Corporation, PCS/360 Manual
- (5) CDC Corporation, PMCS Manual
- (6) P.P. Schoderbrk & L.A. Digman, "Third generation, PERT/LOB," Harvard Business Review, Sep.-Oct. 1967, pp100-110.