韓國 軍事運營分析 學會誌
第9卷 第1號, 1983. 6

OR 컴퓨터 프로그램 개발

朴 淳 達*

이번에는 5개의 프로그램을 소개한다. 특히 I-NVENF는 참고문헌〔1〕의 프로그램 INVENT의 보완적 프로그램이며 MOLP는 참고문헌〔1〕의 프로그램 LPLA01에 내포되어있는 LA01B를 사용하고 있다.

Ⅰ. 프로그램 이름 : MLPGAM

1. 目 的

MLPGAM는 多目的函數를 가진 선형계획법 문제를 目的函數의 가중치에 2人零合 게임의 개념을 사용하여 풀기위한 것이다.

2. 構造와 重要變數

MLPGAM는 2개의 Subroutine, GAMAT와 LA01B로 구성되어 있고, 각각의 Subroutine의 기능은 다음과 같다.

(1) Subroutine GAMAT : 입력 자료를 게임문제의 L.P형으로 풀기 위한 변환 기능을 한다.

(2) LA01B : L.P문제를 푸는 프로그램

重要變數

M : 제약식의 수
N : 구조변수(X)의 갯수
L : 제약식 속의 부등식의 수
NOBJFN : 목적함수의 수
COSTFN : 목적함수의 계수
RHS : 제약식의 RHS의 값
AMATRX : 제약식의 계수 행렬

3. 使用方法

(1) 入 力

카 드	例	入力형태	변수와 내용
1	1~5	I5	M
	6~10	I5	N
	11~15	I5	L
	16~20	AI5	NOBJFN
2~a	1 ~80	8F10.2	COSTFN
a+1~b	1~80	8F10.2	RHS
b+1~c	1~80	8F10.2	AMATRX

(2) 出 力

- 입력 자료에 대한 정보
- 각각의 목적함수 한개만을 고려할때의 최적해
- 각각의 목적함수에 대해서 L.P로 풀었을 경우에 얻어진 이득행렬

이 연구는 1982년도 문교부 학술연구조성비에 의하여 완성된것임.
* 서울대학교 산업공학과

o 각각의 목적함수에 대한 최적 가중치

o 多目的線型計劃法의 최적해와 그 때의 각각의 目的函數에 대한 값

4. 參考事項

MLPGAM는 Subroutine으로 LA01B를 사용하므로 컴파일 후 반드시 LA01B와 link 시켜야 원하는 결과를 얻을 수 있다. 그렇지 않다면 LA01B를 하나의 Subroutine으로 덧붙여야 할 것이다.

5. 使用例

(1) 문 제

Max $\{(0.1X_1+0.2X_2),(10X_1-5X_2)\}$

S.t. $-X_1+X_2 \leqq 1$

$X_1 \quad X_2 \leqq 7$

$X_1 \quad \leqq 5$

$X_2 \leqq 3$

all $X_j \geqq 0$

(2) 入 力

카 드	자 료					
1	4	2	4	2		
2		0.1		0.2		
3		10.0		-5.0		
4		1.0		7.0	5.0	3.0
5		-1.		1.		
6		1.		1.		
7		1.		0.		
8		0.		1.		

(3) 出 力

```
**********************************************
*                                            *
* MULTIOBJECTIVE LINEAR PROGRAMMING PROBLEM  *
*                                            *
**********************************************


** INPUT DATA SPECIFICATION **

   # OF CONSTRAINTS IS     4
   # OF VARIABLES IS      2
   # OF INEQUALITIES      4
   # OF OBJECTIVE FNS IS     2


** COEF. OF COST FNS **


  ++ COEF. OF  1-TH OBJFN ++
           .100         .200

  ++ COEF. OF  2-TH OBJFN ++
         10.000       -5.000



** VALUE OF RHS **

           1.000        7.000        5.000        3.000
```

```
** COEF. OF CONSTRAINTS **

        -1.000        1.000
         1.000        1.000
         1.000         .000
          .000        1.000

7
** OPTIMAL SOLUTION IS AS FOLLOWS,CONSIDERING ONLY A  1-TH OBJFN **

         4.000        3.000


** OPTIMAL SOLUTION IS AS FOLLOWS,CONSIDERING ONLY A  2-TH OBJFN **

         5.000         .000




** PAYOFF MATRIX OBTAINED BY SOLVING L.P. FOR EACH OBJFN **

         1.000         .500
        25.000       50.000

** NORMALIZED PAYOFF MATRIX **

         1.000         .500
          .500        1.000


** MAXIMUM VALUES OF FK(X) **

         1.000       50.000

** PAYOFF MATRIX TO BE USED FOR OBTAINING OPTIMAL WEIGHTS **

         1.000         .500
          .500        1.000


** OPTIMAL WEIGHTING VECTOR OF NORMALIZED GAME **

          .500         .500


** OPTIMAL WEIGHTS CORR. TO THE ORIGINAL OBJECTIVE FNS **

  ++ OPT. WEIGHT FOR  1-TH OBJFN ;          .980

  ++ OPT. WEIGHT FOR  2-TH OBJFN ;          .020
```

** EFFICIENT SOLUTION OF MOLP **

5.000 2.000

** VALUES OF EACH OBJFN FOR THE EFFICIENT SOLUTION

THE VALUE OF 1-TH OBJFN = .900000E+00

THE VALUE OF 2-TH OBJFN = .400000E+02

```
C      PROGRAM MLPDAT
C      THIS PROGRAM SOLVES A MOLP PROBLEM BY TWO-PERSON GAME.
C
C      PROGRAMMED BY PARK,TAE HO
C      DIRECTED BY PROF. PARK,SOONDAL
C      SEOUL NATIONAL UNIVERSITY
C
C      LAST REVISED ON FEB.21,1983
C
       DIMENSION AMATRX(100,100),RHS(100),COSTEN(2100),X(100),ROWMAX(20
      *),XK(11000),IND(110),GAMEFN(20,20),GAMEB(20),GAMEC(20),Y(100),
      *COSTMI(100)
       OPEN(5,FILE='MOLPDAT',MAXREC=80,PAD='YES',BLANK='ZERO').
       OPEN(6,FILE='MOLPOUT',CARRIAGECONTROL='FORTRAN')
       DATA IR,IW/5,6/
       IA=100
       IBB=20
       READ(IR,10) M,N,L,NOBJFN
10     FORMAT(4I5)
       WRITE(IW,600) M,N,L,NOBJFN
       NX2=0
       DO 30 I=1,NOBJFN
       NX1=NX2+1
       NX2=N*I
       READ(IR,20) (COSTEN(J),J=NX1,NX2)
       WRITE(IW,605) I
       WRITE(IW,610) (COSTEN(J),J=NX1,NX2)
20     FORMAT(8F10.2)
30     CONTINUE
       WRITE(IW,620)
       READ(IR,20) (RHS(I),I=1,M)
       WRITE(IW,610) (RHS(I),I=1,M)
       WRITE(IW,630)
       DO 40 I=1,M
       READ(IR,20) (AMATRX(I,J),J=1,N)
       WRITE(IW,610) (AMATRX(I,J),J=1,N)
40     CONTINUE
       NOBJ=0
       DO 80 I=1,NOBJFN
       DO 45 LM=1,N
       NOBJ=NOBJ+1
45     COSTMI(LM)=-COSTEN(NOBJ)
       CALL LA01B(N,M,L,AMATRX,RHS,COSTMI,X,Y,F,IA,0,IND,WK,IER)
       IF(IER.LE.1) GO TO 50
        IF(IER.GE.3) GO TO 170
       WRITE(IW,640) I
50     NOB=0
       WRITE(IW,635) I
       WRITE(IW,610) (X(KN),KN=1,N)
       DO 70 J=1,NOBJFN
       FIJ=0.
       DO 60 K=1,N
       NOB=NOB+1
       FIJ=FIJ+X(K)*COSTEN(NOB)
60     CONTINUE
       GAMEFN(J,I)=FIJ
70     CONTINUE
80     CONTINUE
       WRITE(IW,650)
       DO 90 I=1,NOBJFN
       WRITE(IW,610) (GAMEFN(I,J),J=1,NOBJFN)
90     CONTINUE
       CALL GAMAT(GAMEFN,GAMEB,GAMEC,ROWMAX,NOBJFN,IBB)
       WRITE(IW,660)
       WRITE(IW,610) (ROWMAX(I),I=1,NOBJFN)
       WRITE(IW,670)
       DO 100 I=1,NOBJFN
       WRITE(IW,610) (GAMEFN(I,J),J=1,NOBJFN)
100    CONTINUE
       CALL LA01B(NOBJFN,NOBJFN,NOBJFN,GAMEFN,GAMEB,GAMEC,X,Y,F,IBB,
```

```
     *0,IND,WK,IER)
       IF(IER.GE.3) GO TO 170
       DO 105 I=1,NOBJFN
105    Y(I)=Y(I)/F
       WRITE(Iw,680)
       WRITE(Iw,610) (Y(I),I=1,NOBJFN)
       SUMW=0
       DO 110 I=1,NOBJFN
       SUMW=SUMW+Y(I)/RO.MAX(I)
110    CONTINUE
       DO 120 I=1,N
120    COSTMI(I)=0.
       WRITE(Iw,685)
       NOB=0
       DO 140 I=1,NOBJFN
       RAMDA=Y(I)/(SUMW*RO.MAX(I))
       WRITE(Iw,687) I,RAMDA
       DO 130 J=1,N
       NOB=NOB+1
       COSTMI(J)=COSTMI(J)-RAMDA*COSTFN(NOB)
130    CONTINUE
140    CONTINUE
       CALL LA01B(N,M,L,AMATRX,RHS,COSTMI,X,Y,F,IA,0,IND,WK,IER)
       IF(IER.GE.3) GO TO 170
       WRITE(Iw,690)
       WRITE(Iw,610) (X(I),I=1,N)
       WRITE(Iw,700)
       NOB=0
       DO 160 I=1,NOBJFN
       FIJ=0.
       DO 150 J=1,N
       NOB=NOB+1
       FIJ=FIJ+X(J)*COSTFN(NOB)
150    CONTINUE
       WRITE(Iw,710) I,FIJ
160    CONTINUE
       GO TO 180
170    WRITE(Iw,720) IER
600    FORMAT(////24X,45(1H*)/24X,1H*,43X,1H*/24X,1H*,43X,1H*/24X,45H* MULTIOBJECTIVE LIN
     *EAR PROGRAMMING PROBLEM */24X,1H*,43X,1H*/24X,45(1H*)/////2X,30H**
     * INPUT DATA SPECIFICATION **//5X,19H# OF CONSTRAINTS IS,I5/5X,17H#
     * OF VARIABLES IS,I5/5X,17H# OF INEQUALITIES,I5/5X,21H# OF OBJECTIV
     *E FNS IS,I5////2X,23H** COEF. OF COST FNS **//)
605    FORMAT(/4X,12H++ COEF. OF ,I2,12H-TH OBJFN ++)
610    FORMAT(5X,8F12.3)
620    FORMAT(////2X,18H** VALUE OF RHS **//)
630    FORMAT(////2X,26H** COEF. OF CONSTRAINTS **//)
640    FORMAT(//2X,54HXXXX MAX BASIC MARGINAL COST > TOLERANCE IN L.P. WI
     *TH ,I2,9H-TH OBJFN//)
635    FORMAT(///2X,53H** OPTIMAL SOLUTION IS AS FOLLOWS,CONSIDERING ONLY
     * A ,I2,12H-TH OBJFN **/)
650    FORMAT(////2X,59H** PAYOFF MATRIX OBTAINED BY SOLVING L.P. FOR EAC
     *H OBJFN **//)
660    FORMAT(////2X,29H** MAXIMUM VALUES OF FK(X) **//)
670    FORMAT(////2X,60H** PAYOFF MATRIX TO BE USED FOR OBTAINING OPTIMAL

     * WEIGHTS **//)


680    FORMAT(////2X,49H** OPTIMAL WEIGHTING VECTOR OF NORMALIZED GAME **
     *//)
685    FORMAT(////2X,57H** OPTIMAL WEIGHTS CORR. TO THE ORIGINAL OBJECTV
     *E FNS **//)
687    FORMAT(4X,19H++ OPT. WEIGHT FOR ,I2,12H-TH OBJFN ; ,F12.3/)
690    FORMAT(////2X,32H** EFFICIENT SOLUTION OF MOLP **//)
700    FORMAT(////2X,50H** VALUES OF EACH OBJFN FOR THE EFFICIENT SOLUTIO
     *N/)
710    FORMAT(/4X,13HTHE VALUE OF ,I2,11H-TH OBJFN =,E13.6)
720    FORMAT(////2X,44HXXXX ANY SEVERE ERROR OCCURS IN SOLVING L.P.//12X
     *,13HERROR NO. IS ,I2)
180      STOP
```

```
      END
C
C
      SUBROUTINE GAMAT(A,B,C,ROWMAX,M,IBB)
      DIMENSION A(IBB,1),B(1),C(1),ROWMAX(1)
      DATA IW/6/
      DO 10 I=1,M
      B(I)=1.
      C(I)=-1.
10    CONTINUE
      NORO=0
      DMINA=1.0E30
      DMIN=1.0E30
      DO 30 I=1,M
      DMAX=-1.0E30
      DO 20 J=1,M
      IF(A(I,J).LT.DMAX) GO TO 15
      DMAX=A(I,J)
15    IF(A(I,J).GE.DMIN) GO TO 20
      DMIN=A(I,J)
20    CONTINUE
      IF(DMAX.LE.0.) NORO=NORO+1
      ROWMAX(I)=DMAX
30    CONTINUE
      IF(DMIN.GE.0.) GO TO 35
      IF(NORO.EQ.0) GO TO 35
      DMIN=-DMIN
      GO TO 40
35    DMIN=0.
40    DO 60 I=1,M
      ROWMAX(I)=ROWMAX(I)+DMIN
      DO 50 J=1,M
      A(I,J)=(A(I,J)+DMIN)/ROWMAX(I)
      IF(A(I,J).GE.DMINA) GO TO 50
      DMINA=A(I,J)
50    CONTINUE
60    CONTINUE
      WRITE(IW,600)
600   FORMAT(////2X,30H** NORMALIZED PAYOFF MATRIX **//)
      DO 65 KK=1,M
65      WRITE(IW,610) (A(KK,KB),KB=1,M)
610   FORMAT(5X,8F12.3)
      IF(DMINA.GT.0.) GO TO 90
      DM=-DMINA+1.
      DO 80 I=1,M
      DO 70 J=1,M
      A(I,J)=A(I,J)+DM
70    CONTINUE
80    CONTINUE
90    RETURN
      END
```

Ⅱ. 프로그램 이름 : ILPG

1. 目 的

ILGP는 目標計劃法의 문제를 풀기 위한것이며, 사용자가 임의로 整數解 또는 實數解를 구할 수 있도록 되어 있다.

2. 構造와 重要變數

ILPG는 8개의 Subroutine 과 1개의 function으로 되어 있고, 각 Subroutine 의 기능은 다음과 같다.

(1) Subroutine PLACE : achievement function의 값들을 내부에 기억시킨다.

(2) Subroutine CINDX : index rows를 계산한다.

(3) Subroutine TEST :다음번 후보변수의 열과 탈락변수의 행을 계산한다.

(4) Subroutine PERM :후보변수의 열과 탈락변수의 행이 주어진 상태에서 새로운 행렬표를 계산한다.

(5) Subroutine FCPL :요구되는 절단평면을 계산하고 행렬표에 덧붙임으로써 수정한다.

(6) Subroutine ALTST :유효한 다른해가 있는 지를 구해진 행렬표에서 검사한다.

(7) Subroutine POUT :결과를 사용자가 정해준 초기 조건에 따라 출력한다.

(8) Subroutine INTST :整數解를 구하는 경우에 구한 해가 바른가를 검사한다.

• 重要變數

NOBJ :목적 함수의 갯수
NPRI :우선 순위(priority level)의갯수
NVAR :구조변수 (X)의 갯수
NTAF : achievement function속의 총변수의 갯수
INSW :解의 성질에 따른 option
　　=0, 일반적 LGP의 해
　　=1, 整數解
TE　:목적 함수의 계수 행렬
TB　:목적 함수의 RHS의 값
IPRI : achievement function속의 각 변수 들의 우선 순위
ISUB :편차 변수의 (+), (-) 표시
WHTF :편차 변수의 가중치

3. 使用方法

(1) 入　力

카 드	列	入力형태	변수와 내용
1	1～5	I5	NOBJ
	6～10	I5	NPRI
	11～15	I5	NVAR
	16～20	I5	NTAF
	21～25	I5	INSW
2～a	1～80	8F10.0	TE(i,j)
a+1～b	1～80	8F10.0	TB(i)
b+1～c	1～5	I5	IPRI
	6～10	I5	ISUB
	11～20		WHIF

(2) 出　力

ZBAR :각 우선 순위에서의 최적해에 대한 achievement function의 값
XSTAR :구조변수(X)의 최적해
PSTAR :(+)편차변수의 최적해
NSTAR :(-) 편차변수의 최적해

4. 參考事項

ILGP는 변수의 갯수가 10개, priority level의 갯수가 10개, 목적함수의 갯수가 20개를 넘지 못한다. 만약, 이 이상의 문제를 풀려고 하면 dimension을 확장하면 된다. 그리고 몇가지의 문제를 계속하여 풀고자 하면 입력형태에 맞게 자료를 뒤에 덧붙이면 된다.

5. 使用例

(1) 問　題

$$\text{Min}\{(2P_1+3P_2),\ (n_3),\ (n_4)\}$$

$$X_1+X_2+n_1-P_1=10$$
$$X_1+n_2-P_2=4$$
$$5X_1+3X_2+n_3-P_3=56$$
$$X_1+X_2+n_4-P_4=12$$
$$\text{all } X_j \geqq 0$$

(2) 入 力

카 드	자 료
1	4 3 2 4
2	1 1. 1. 0. 5. 3. 1. 1.
3	10. 4. 56. 12.
4	1 1 2.
5	1 2 3.
6	2 -3 1.
7	3 4 1.

(3) 出 力

```
PROBLEM    1 READ IN SUCCESSFULLY

SUBSCRIPT      3 TERMS ZSAR      2 TERMS XSTAR      4 TERMS PSTAR      4TERMS NSTAR

    1              .0000             4.0000             .0000              .0000
    2            18.0000             6.0000             .0000              .0000
    3             2.0000                                .0000            18.0000
    4                                                   .0000             2.0000
```

```
C     PROGRAM   LLGP
C
C     ORIGINAL PROGRAM FROM IGNIZIO'S GOAL PROGRAMMING
C     TESTED BY  KIM, TAE HO
C     SEOUL NATIONAL UNIVERSITY

C
C     LAST REVISED ON FEb. 21, 1983
C
C
C
C          THE PURPOSE OF THIS PROGRAM IS TO SOLVE LINEAR GOAL
C          PROGRAMMING PROBLEMS WITH MULTIPLE OBJECTIVES.
C
C          CARD1: NOBJ,NPRI,NVAR,NTAF,INSW   FORMAT 5I5
C                 NOBJ--NUMBER OF OBJETIVES
C                 NPRI--NUMBER OF PRIORITY LEVELS
C                 NVAR--NUMBER OF VARIABLES(X'S)
C                 NTAF--NUMBER OF TERMS IN ACHIEVEMENT FUNCTION
C                 INSW--INTEGER SOLUTION(REQUIRED SWITCH)
C
C          CARD2: COEFFICIENTS OF TE MATRIX  FORMAT 8F10.0
C
C          CARD3: RIGHT HAND SIDE VALUES FOR TB MATRIX  FORMAT 8F10.0
C
C          CARD4: 1 CARD FOR EACH TERM IN ACHIEVEMENT FUNCTION
C                 IPRI,ISUB,WHTF     FORMAT 2I5,F10.0
C                 IPRI--PRIORITY LEVEL OF PARTICULAR TERM
C                 ISUB--  +P OR -N VALUE(DESIGNATED WITH A+ OR -SUBSCRIPT
C                         VALUE)
C                 *EXAMPLE: N2 IS CODED AS -2; P2 CODED A 2
C
C          PRESENT ARRAY DIMENSIONS ACCOMODATE 10 VARIABLES, 10 PRIORITY
C          LEVELS, AND 20 OBJECTIVES AS A MAXIMUM.
C          NOTE; THE 20 OBJECTIVES MAXIMUM INCLUDES ANY REQUIRED CUTTING
C                PLANES AS WELL AS INPUT
C          THE COMPUTATION TABLEAU IS BROKEN INTO 6 ARRAYS:
C              TE(NO,NC)=EQUATION FIELD (COEFFICIENTS CIJ)
C              TB(NO)    =B FIELD
C              TL(NO,NP)=LEFT STUB
C              TT(NP,NC)=TOP STUB
C              TI(NP,NC)=INDEX ROWS
C              TA(NP)    =A FIELD
C          THE SUBSCRIPT AS UTILIZED ABOVE INDICATE:
C              NO=1,NOBJ   THE NUMBER OF OBJECTIVES;IE, THE NUMBER OF
C                          ROWS IN TE,TB, AND TL
C              NC=1,NCOL   THE NUMBER OF COLUMNS IN TE,TT,AND TI
C              NP=1,NPRI   THE NUMBER OF PRIORITIES (NUMBER OF ROWS
C                          IN TT,TI,TA, AND THE NUMBER OF COLUMNS IN TL)
C          THE COLUMN AND ROW HEADINGS ARE ENCODED INTO THE
C              JCOL(NSUB,NTYPE)    AND
C              JROW(NSUB,NTYPE)    FIELDS.
C          NSUB IS THE SUBSCRIPT NUMBER ;NTYPE IS INTERPRETED
C          AS:
C          2=X
C          3=P
C          4=N
C
C          SCALARS IN THE COMMON AREA ARE:
C             NOBJ     = NUMBER OF OBJECTIVES
C             NPRI     = NUMBER OF PRIORITIES
C             NVAR     = NUMBER OF VARIABLES
C             NCOL     =NUMBER OF COLUMNS
C             NROW     = NUMBER OF THE ROW CURRENTLY BEING INVESTIGATED.


C             INSW     = INTEGER SOLUTION SWITCH
C                      = 0 NO INTEGER SOLUTION
C                      = 1 INTEGER SOLUITON REQUIRED
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
```

```
C
C        IMAX IS THE MAXIMUM NUMBER OF TABLEAU GENERATIONS ALLOWED

      DATA IMAX/200/
C        NPRR IS INCREMENTED EACH TIME A NEW DATA SET IS READ.THIS IS
C        DONE TO DISTINGUSH DATA SETS DURING BATCH PROCESSING.
      DATA NPRB/0/
C        THE NUMBER OF OBJECTIVES, THE NUMBER OF PRIORITY LEVELS,
C        THE NUMBER OF VARIABLES, THE NUMBER OF TERMS IN ACHIEVEMENT
C        FUNCTION AND INTEGER SOLUTION SWITCH ARE READ IN.
      DATA IR/5/,IW/6/
      OPEN(5,FILE='XILGPDAT',MAXRECL=80,PAD='YES',BLANK='ZERO')
      OPEN(6,FILE='XILGPOUT',CARRIAGECONTROL='FORTRAN')
25    READ(IR,30,END=40,ERR=21) NOBJ,NPRI,NVAR,NTAF,INSW
C        INPUTS ARE CHECKED TO MAKE CERTAIN THAT THEY DO NOT
C        EXCEED THE MAXIMUM DIMENSIONS OF THE ARRAY.
      IF(NOBJ.LT.1.OR.NPRI.LT.1.OR.NVAR.LT.1) GO TO 22
      IF(NOBJ.GT.20.OR.NPRI.GT.10.OR.NVAR.GT.10) GO TO 22
      NCOL=NOBJ+NVAR
C        SET COLUMN HEADINGS
C        X CODED WITH 2
C        P CODED 3
C        N CODED 4
      DO 1 NV=1,NVAR
      JCOL(NV,1)=2
1     JCOL(NV,2)=NV
      DO 2 NO=1,NOBJ
      NC=NO+NVAR
C                  DETERMINE P COLUMNS
      JCOL(NC,1)=3
      JCOL(NC,2)=NO
C                  DETERMINE N ROWS
      JROW(NO,1)=4
2     JROW(NO,2)=NO
C                  READ IN COEFFICIENTS CIJ'S
      READ(IR,31,END=20,ERR=21)((TE(NO,NV),NV=1,NVAR),NO=1,NOBJ)
C                  PLACE -1'S ON THE DIAGONAL OF TE AND 0'S ELSEWHERE
      DO 3 NOR=1,NOBJ
      DO 3 NO=1,NOBJ
      NOC=NO+NVAR
      TE(NOR,NOC)=0
      IF(NO.EQ.NOR) TE(NOR,NOC)=-1
3     CONTINUE
C                  READ RIGHT HAND SIDE VALUES INTO TB ARRAY
      READ(IR,31,END=20,ERR=21) (TB(NO),NO=1,NOBJ)
C                  THE STUBS ARE ZEROED
      DO 6 NP=1,NPRI
      DO 4 NO=1,NOBJ
4     TL(NO,NP)=0
      DO 5 NC=1,NCOL
5     TT(NP,NC)=0
6     CONTINUE
C                  READ STUB VALUES IN.
C                   IPRI=PRIORITY LEVEL
C                   ISUB=+ISUB THE N'TH P
C                       =-ISUB THE N'TH N
C                   WHTF=WEIGHTING FACTOR
      DO 7 NT=1,NTAF



      READ(IR,32,END=20,ERR=21)IPRI,ISUB,WHTF
C          SUBROUTINE PLACE INSERTS VALUES INTO STUBS.
      CALL PLACE(IPRI,ISUB,WHTF)
7     CONTINUE
C          THE PROBLEM COUNTER IS INCREMENTED AND WRITTEN OUT.
      NPRB=NPRB+1
      WRITE(IW,34) NPRB
C          THE CUTTING PLANE AND PRINTOUT COUNTERS ARE ZEROED FOR THE
C          PROBLEM.
```

```
      NCPL=0
      NPRI=0
C           OPTIMIZE ROW COUNTER AND THE LOOP-CHECK COUNTER
C           ARE ZEROED FOR THE START AND FOR EACH NEW CUTTING PLANE ADDED
8     NROW=0
      IVAL=0
C           WHEN NROW BECOMES NPRI,THE ALGORITHM IS TERMINATED.
9     IF(NROW.EQ.NPRI) GO TO 11
C           NROW IS THE ROW THAT THE PROGRAM IS UP TO IN THE OPTIMIZING
C           ROUTINE.
      NROW=NROW+1
C           CALLING CINDX(0) CAUSES INDEX ROW NROW TO BE COMPUTED
      CALL CINDX(0)
C           SUBROUTINE TEST COMPUTES THE ENTERING VARIABLE'S COLUMN
C           AND DEPARTING VARIABLE'S ROW
10    CALL TEST(NEVC,NDVR)
C           IF TEST RETURNS AN ENTERING VARIABLE COLUMN=0,THE ROW
C           CAN NOT BE OPTIMIZED FURTHER.
      IVAL=IVAL+1
      IF(IVAL.GE.IMAX) GO TO 19
      IF(NEVC.LE.0) GO TO 9
C           SUBROUTINE PERM COMPUTES THE NEW TABLEAU.
      CALL PERM(NEVC,NDVR)
      GO TO 10
C           SUBROUTINE INTST TESTS THE SOLVED TABLEAU TO DETERMINE
C           WHETHER IT MEETS ANY REQUIRED INTEGER STIPULATIONS.
11    CALL INTST(NSW)
C           SUBROUTINE POUT(PRINTOUT) IS CALLED IF THE SOLUTION
C           IS SATISFACTORY.
      IF(NSW.EQ.0) CALL POUT(NPRT,NCPL)
C           SUBROUTINE ALTST TESTS FOR ALTERNATE VALID SOLUTIONS AND
C           PRINTS OUT ANY THAT IT FINDS.
      CALL ALTST(NPRT,NCPL)
C           POUT AND ALTST INCREMENT THE PRINT-OUT COUNTER WHENEVER
C           THEY OUTPUT SOLUTIONS. THEREFORE, IF NPRT=0, NO SOLUTION
C           HAS BEEN FOUND,SO A CUTTING PLANE IS EMPLOYED.
      IF(NPRT.NE.0) GO TO 25
C           SUBROUTINE FCPL FORMS A CUTTING PLANE AND ADDS IT TO
C           THE TABLEAU. IF IT RETURNS AN ISW VALUE=0, IT FAILED
C           TO DERIVE A VALID CUTTING PLANE AND THE EXECUTION OF THE
C           CURRENT PROBLEM IS TERMINATED.
      CALL FCPL(ISW)
      IF(ISW.EQ.0) GO TO 25
C           THE CUTTING PLANE COUNTER IS INCREMENTED.
      NCPL=NCPL+1
      GO TO 8
19    WRITE(IW,38) IMAX
      GO TO 25
20    WRITE(IW,35)
      GO TO 40
21    WRITE(IW,36)
      GO TO 40
22    WRITE(IW,37)
30    FORMAT(5I5)



31    FORMAT(8F10.0)
32    FORMAT(2I5,F10.0)
34    FORMAT(1H1,////,' PROBLEM',I4,' READ IN SUCCESSFULLY')
35    FORMAT(//,' **DATA  STREAM ERROR--MISSING DATA CARDS**')
36    FORMAT(//,' DATA STREAM ERROR--UNREADABLE DATA CARD**')
37    FORMAT(//,' **INPUT VARIABLE EXCEEDS ALLOWABLE DIMENSION RANGE**')
38    FORMAT(//, ' **  ALGORITHM DID NOT FINISH ',I5,'  ITERATIONS **')
40    CONTINUE
      CALL EXIT
      END
      SUBROUTINE PLACE(IPRI,ISUB,WHTF)
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
C          INPUT DATA IS CHECKED TO BE CERTAIN THAT IT IS IN RANGE.
```

```
      IF(IPRI.LT.1.OR.IPRI.GT.NPRI) GO TO 2
      IF(ISUB.EQ.0.OR.IABS(ISUB).GT.NOBJ) GO TO 2
C        POSITIVE VALUES OF ISUB INDICATE TOP STUB VALUES.
      IF(ISUB.GT.0) GO TO 1
C        PLACE VALUES IN LEFT STUB.
      ISUB=-ISUB
C        OVERWRITES ARE DETECTED AS BEING ERROR.
      IF(TL(ISUB,IPRI).NE.0) GO TO 3
      TL(ISUB,IPRI)=WHTF
      RETURN
1     ICOL=ISUB+NVAR
C        PLACE VALUE IN RIGHT STUB
      IF(TT(IPRI,ICOL).NE.0) GO TO 3
      TT(IPRI,ICOL)=WHTF
      RETURN
2     WRITE(IW,4)
      GO TO 6
3     WRITE(IW,5)
4     FORMAT(//,' **SUBSCRIPT OUT-OF-RANGE WHILE READING SUBSCRIPT**')
5     FORMAT(//,' **OVERWRITE ATTEMPTED IN A STUB**')
6     CALL EXIT
      END
      SUBROUTINE CINDX(ISW)
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
      I=1
      IF(ISW.NE.1) I=NROW
C           FOR CINDX(0), THE LOOP IS EXECUTED FOR I=NROW,NROW
C           THAT IS,FOR INDEX ROW NROW ONLY.
      DO 3 NP=I,NROW
      TA(NP)=0.
C           THE RIGHT HAND SIDE OF THE INDEX ROWS ARE COMPUTED
      DO 1 NO=1,NOBJ
1     TA(NP)=TA(NP)+TB(NO)*TL(NO,NP)
      DO 2 NC=1,NCOL
      TI(NP,NC)=-TT(NP,NC)
      DO 2 NO=1,NOBJ
      TI(NP,NC)=TI(NP,NC)+TE(NO,NC)*TL(NO,NP)
2     CONTINUE
3     CONTINUE
      RETURN
      END
      SUBROUTINE TEST(NEVC,NDVR)
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
      NEVC=0
C           IF THE LEVEL OF ACHIEVEMENT OF PRIORITY NROW IS ZERO,
C           THIS LEVEL IS OPTIMUM--RETURN NEVC=0
      IF(TA(NROW).LE.0.) RETURN


C           DETERMINE COLUMN OF ENTERING VARIABLE
      VEVC=0
      NRMW=NROW-1
      DO 3 NC=1,NCOL
      IF(TI(NROW,NC).LE.0.) GO TO 3
      IF(NROW.EQ.1) GO TO 2
      DO 1 N=1,NRMW
      IF(TI(N,NC).LT.0.) GO TO 3
1     CONTINUE
2     IF(TI(NROW,NC).LE.VEVC) GO TO 3
      NEVC=NC
      VEVC=TI(NROW,NC)
3     CONTINUE
C           IF NO POSITIVE VALUES EXIST WHICH HAVE NO NEGATIVE
C           INDEX VALUE ABOVE-RETURN NEVC=0,SINCE NO FURTHER
C           OPTIMIZATION IS POSSIBLE AT THIS LEVEL.
      IF(NEVC.EQ.0) RETURN
C           DETERMINE DEPARTING VARIABLE'S ROW
```

```
      NDVR=0
      DO 7 NR=1,NOBJ

      IF(TE(NR,NEVC).LE.0.) GO TO 7
      V=TB(NR)/TE(NR,NEVC)
      IF(NDVR.EQ.0) GO TO 6
      IF(V-VDVR)6,4,7
4     DO 5 NP=1,NPRI
      IF(TL(NR,NP)-TL(NDVR,NP))7,5,6
5     CONTINUE
6     VDVR=V
      NDVR=NR
7     CONTINUE
      IF(NDVR.GE.1) RETURN
      WRITE(IW,8)
8     FORMAT(//,' **PROGRAM TERMINATED-FAILED PIVOT-COMPUTION**')
      CALL EXIT
      END
      SUBROUTINE PERM(NEVC,NDVR)
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
C        SWAP HEADINGS FOR ROW NDVR AND COLUMN NEVC
      DO 1 I=1,2
      J=JCOL(NEVC,I)
      JCOL(NEVC,I)=JROW(NDVR,I)
1     JROW(NDVR,I)=J
C        SWAP VECTORS FOR TL(NDVR,NP) AND(TT(NP,NEVC),NP=1,NPRI)
      DO 2 NP=1,NPRI
      TEMP=TL(NDVR,NP)
      TL(NDVR,NP)=TT(NP,NEVC)
2     TT(NP,NEVC)=TEMP
C        COMPUTE NEW TE ARRAY
      PIV=TE(NDVR,NEVC)
      PIB=TB(NDVR)
      DO 31 NO=1,NOBJ
      IF(NO.EQ.NDVR) GO TO 31
      PIX=TE(NO,NEVC)/PIV
      TB(NO)=FIX(TB(NO)-PIX*PIB)
6     DO 3 NC=1,NCOL
      IF(NC.EQ.NEVC) GO TO 3
      TE(NO,NC)=FIX(TE(NO,NC)-TE(NDVR,NC)*PIX)
3     CONTINUE
31    CONTINUE
      DO 4 NC=1,NCOL
4     TE(NDVR,NC)=FIX(TE(NDVR,NC)/PIV)
      DO 5 NO=1,NOBJ


5     TE(NO,NEVC)=FIX(-TE(NO,NEVC)/PIV)
      TB(NDVR)=FIX(TB(NDVR)/PIV)
      TE(NDVR,NEVC)=FIX(1/PIV)
C         RECOMPUTE INDEX ROWS 1 THRU NROW
      CALL CINDX(1)
      RETURN
      END
      FUNCTION FIX(Z)
C          THIS FUNCTION BRINGS FLOATING POINT VALUES
C          THAT ARE EITHER + OR - .0001 FROM AN INTEGER TO THAT INTEGER
      X=1
      DO 1 N=1,3
      IF(N.NE.1) X=10*X
      F=X*Z
      I=F
      J=I-2
      DO 1 K=1,3
      G=J+K
      IF(ABS(F-G)-.005) 2,2,1
1     CONTINUE
      FIX=Z
      RETURN
```

```
2       FIX=G/X
        RETURN
        END
        SUBROUTINE FCPL(ISW)
        COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
       *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
C          IF THE NUMBER OF OBJECTIVES IS ALREADY THE MAXIMUM,NO NEW
C          CUTTING PLANE OBJECTIVE CAN BE FITTED INTO THE TABLEAU.
        IF(NOBJ.EQ.20) GO TO 10
        CALL INTST(NSW)
        IF(NSW.EQ.0) GO TO 11
        ISW=1
C          THE NUMBER OF OBJECTIVES IS INCREMENTED FOR THE ADDED
C          CUTTING PLANE
        NOBJ=NOBJ+1
C          THE TE VALUES OF THE CUTTING PLANE  ARE COMPUTED.
        DO 4 NC=1,NCOL
        X=TE(NSW,NC)
        I=X
        Y=I
        IF(FIX(X-Y))1,2,3
1       TE(NOBJ,NC)=1.+X-Y
        GO TO 4
2       TE(NOBJ,NC)=0
        GO TO 4
3       TE(NOBJ,NC)=X-Y
4       CONTINUE
C           THE TB VALUE OF THE CUTTING PLANE IS COMPUTED.
        X=TB(NSW)
        I=X
        Y=I
        IF(FIX(X-Y))5,12,6
5       TB(NOBJ)=1.+X-Y
        GO TO 7
6       TB(NOBJ)=X-Y
C           THE NEW COLUMN IS ADDED TO THE TABLEAU WITH A=1 IN THE
C           CUTTING PLANE OBJECTIVE'S ROW
7       NCOL=NCOL+1
        DO 8 NO=1,NOBJ
8       TE(NO,NCOL)=0
        TE(NOBJ,NCOL)=-1
C             THE STUBS ARE EXTENDED AND A 1 IS PLACED IN THE PRIORITY
C             1 SLOT OF THE LEFT STUB IN THE CUTTING PLANE OBJECTIVE'S ROW
        DO 9 NP=1,NPRI
        TT(NP,NCOL)=0
9       TL(NOBJ,NP)=0
        TL(NOBJ,1)=1
C             TITLE NEW ROWS & COLUMNS
        JROW(NOBJ,1)=4
        JROW(NOBJ,2)=NOBJ
        JCOL(NCOL,1)=3
        JCOL(NCOL,2)=NOBJ
        GO TO 14
10      WRITE(IW,15)
        GO TO 13
11      WRITE(IW,16)
        GO TO 13
12      NOBJ=NOBJ-1
        WRITE(IW,17)
13      WRITE(IW,18)
        N=0
        CALL POUT(N,N)
        ISW=0
14      RETURN
15      FORMAT(/,' FCPL FAILURE-DIMENION EXCEEDED ')
16      FORMAT(/,' FCPL FAILURE-ILLEGAL PATH')
17      FORMAT(/,' FCPL FAILURE-ABNORMAL VALUE')
18      FORMAT(/,' VALUES AT THE TIME OF FAILURE FOLLOW')
        END
        SUBROUTINE ALTST(NPPT,NCPL)
        COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
```

```
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
      DO 4 NC=1,NCOL
C          THE TI FIELD IS TESTED FOR AN ALL-ZERO COLUMN,
C          INDICATING ALTERNATE SOLUTIONS ON THAT COLUMN.
      DO 1 NP=1,NPRI
      IF(TI(NP,NC).NE.0) GO TO 4
1     CONTINUE
      DO 3 NO=1,NOBJ
      IF(TE(NO,NC).LE.0) GO TO 3
      IF(TB(NO).LE.0) GO TO 3
C          IF A+ VALUE OF TE IS FOUND AND TB CORRESPONDING TO IT IS
C          POSITIVE, PERM IS CALLED TO PIVOT ON THAT TE ELEMENT
2     CALL PERM(NC,NO)
      DO 5 NB=1,NOBJ
      IF(TB(NB).LT.0) GO TO 6
C          IF THE NEW TABLEAU HAS NO NEGATIVE TB ELEMENT,THE ALTERNATE
C          SOLUTION IS VALID.
5     CONTINUE
C          INTST IS CALLED TO DETERMINE WHETHER THE SOLUTION MEETS ANY
C          INTEGER REQUIREMENTS.
      CALL INTST(NSW)
C          POUT IS CALLED IF THE SOLUTION IS GOOD.
      IF(NSW.EQ.0) CALL POUT(NPRT,NCPL)
C          PERM IS CALLED AGAIN TO RETURN THE TABLEAU TO ITS OTIGINAL
C          FORM FOR FURTHER ALTERNATE SOLUTION SEARCH
6     CALL PERM(NC,NO)
3     CONTINUE
4     CONTINUE
      RETURN
      END
      SUBROUTINE POUT(NPRT,NCPL)
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10),
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW


      DIMENSION WOUT(20,4)
      NPRT=NPRT+1
      IF(NPRT.NE.1) WRITE(IW,31) NPRT
      IF(NCPL.NE.0.AND.NPRT.EQ.1) WRITE(IW,32)NCPL
C          THE OUTPUT ARRAY IS ZEROED.
11    DO 12 I=1,20
      DO 12 J=1,4
12    WOUT(I,J)=0
C          THE OUTPUT ARRAY IS FILLED.
      DO 13 NP=1,NPRI
13    WOUT(NP,1)=FIX(TA(NP))
C          THE FUNCTION FIX RETURNS VALUES THAT HAVE STRAYED FROM
C          BEING INTEGER VALUED DUE TO THE PRECISION LIMITATIONS OF
C          THE COMPUTER,BACK TO INTEGERS.
      DO 14 NO=1,NOBJ
      IC=JROW(NO,1)
      ID=JROW(NO,2)
14    WOUT(ID,IC)=FIX(TB(NO))
C           THE OUTPUT ARRAY IS WRITTEN OUT.
      WRITE(IW,33)NPRI,NVAR,NOBJ,NOBJ
      I=MAX(NPRI,NVAR,NOBJ)
      DO 20 K=1,I
      IF(K.GT.NPRI) GO TO 16
      IF(K.GT.NVAR) GO TO 15
      WRITE(IW,34)K,(WOUT(K,J),J=1,4)
      GO TO 20
15    WRITE(IW,35)K,WOUT(K,1),(WOUT(K,J),J=3,4)
      GO TO 20
16    IF(K.GT.NVAR) GO TO 17
      WRITE(IW,36)K,(WOUT(K,J),J=2,4)
      GO TO 20
17    WRITE(IW,37)K,(WOUT(K,J),J=3,4)
20    CONTINUE
31    FORMAT(/,' ALTERNATE SOLUTION NUMBER ',I3)
32    FORMAT(/,' NUMBER OF CUTTING PLANES USED= ',I3)
```

```
33    FORMAT(/,' SUBSCRIPT',I6,' TERMS ZBAR',I6,
     *' TERMS XSTAR ',I6,' TERMS PSTAR ',I6,'TERMS NSTAR'/)
34    FORMAT(I6,4F18.4)
35    FORMAT(I6,F18.4,18X,2F18.4)
36    FORMAT(I6,18X,3F18.4)
37    FORMAT(I6,36X,2F18.4)
      RETURN
      END
      SUBROUTINE INTST(NSW)
      COMMON TL(20,10),TT(10,30),TE(20,30),TI(10,30),TB(20),TA(10)
     *JCOL(30,2),JROW(20,2),NOBJ,NPRI,NVAR,NCOL,NROW,INSW,IR,IW
      NSW=0
      IF(INSW.EQ.0) RETURN
      DO 2 NO=1,NOBJ
      IF(JROW(NO,1).NE.2) GO TO 2
      I=TB(NO)
      J=I-2
      DO 1 K=1,3
        G=J+K
      IF(ABS(TB(NO)-G)-.0001) 2,2,1
1     CONTINUE
      GO TO 3
2     CONTINUE
      RETURN
3     NSW=NO
      RETURN
      END
```

Ⅲ. 프로그램 이름 :CPMMEN

1. 目 的

프로젝트 수행에 있어서 각 날짜별로 장비, 인원을 균등하게 분배하여 고용을 하는 측에서 가장 균등하게 장비, 인원을 고용할 수 있게 해 주는 것이다.

(1) 각 node별로 early start, late start, 여유시간을 구해 준다.

(2) 각 node별로 시작 날짜와 끝나는 날짜를 구해 준다.

(3) 날짜별로 필요한 인원 수를 구해 준다.

2. 解 法

(1) The Burgess-Killebrew heuristic Method를 사용한다.

(2) 참고문헌

Roy D.Harris, Michael J. Maggard and William G Lesso, Computer Models in Operations Research, Harper & Row Publishers

3. 構造와 重要變數

(1) Main : 각 node별로 early start, late start, 여유시간을 계산하고, Subroutine NOBODY에서 계산된 최적 행렬로 프로젝트의 일정을 구한다.

(2) Subroution NOBODY : 프로젝트의 일정을 짜기 위한 최적 행렬을 구한다.

• 重要變數

M : node의 총 갯수

N : 최단 시일

A : node와 날짜로 된 행렬(=간트도표)

E : node의 가장 일찍 시작할 수 있는 날짜

B : node의 가장 늦게 시작할 수 있는 날짜

C : node의 가장 일찍 끝날 수 있는 날짜

D : node의 가장 늦게 끝날 수 있는 날짜

PROT : node의 필요한 작업 날짜

S : node의 여유시간

MN : 필요한 인원 수

4. 使用方法

(1) 入 力

카 드	例	入力형태	변수와 내용
1	1 ~ 5	I5	M
2~N+1	1 ~ 5	I5	node
	6 ~30	5I5	선행작업
	31 ~40	I10	총 작업시간
	41 ~50	I10	일간 작업시간
	51 ~55	I 5	인원 수

(2) 出 力

• 입력자료

• early start, late start, 여유시간

• node와 각각의 시작 날짜와 끝나는 날짜

• 날짜와 필요한 인원 수

5. 參考事項

(1) node의 수 N에는 假活動(dummy node)의 수도 포함되어야 한다.

(2) 날짜자료는 선행 작업이 없는 것 부터 차례대로 入力시켜야 한다.

(3) 선행 작업이 5개가 넘을 경우는 사용할 수 없다.

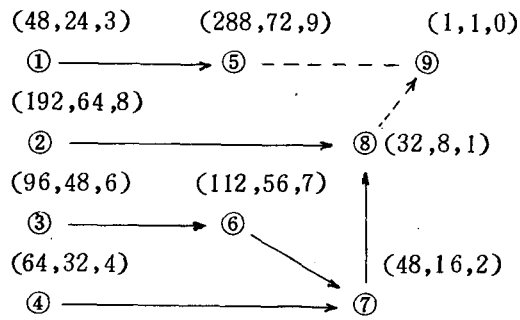
6. 使用例

(1) 問 題

node의 수 (M) = 9

프로젝트의 일정 계획이 다음 그림과 같고, ( )안은 차례로 총 작업시간, 日間작업시간, 필요한 인원 수 이다.

⑨ node는 dummy node이다.



(2) 入 力

카드	자 료								
1	9								
2	1	0	0	0	0	0	48	24	3
3	2	0	0	0	0	0	192	64	8
⋮	⋮						⋮	⋮	⋮
6	5	1	0	0	0	0	288	72	9
⋮	⋮						⋮	⋮	⋮
9	8	2	7	0	0	0	32	8	1
10	9	5	8	0	0	0	1	1	0

(3) 出 力

* EVALUATION OF INPUT DATA *

NODE	PREDECESSORS	TOTAL M/H	NORM DAILY M/H	MEN	EARLY START	LATE START	SLACK
1	0 0 0 0 0	48	24	3	0	5	5
2	0 0 0 0 0	192	64	8	0	4	4
3	0 0 0 0 0	96	48	6	0	0	0
4	0 0 0 0 0	64	32	4	0	2	2
5	1 0 0 0 0	288	72	9	2	7	5
6	3 0 0 0 0	112	56	7	2	2	0
7	4 6 0 0 0	48	16	2	4	4	0
8	2 7 0 0 0	32	8	1	7	7	0
9	5 8 0 0 0	1	1	0	11	11	0

* SCHEDULE OF THE PROJECT *

NODE	START	FINISH
1	5	6
2	5	7
3	1	2
4	1	2
5	8	11
6	3	4
7	5	7
8	8	11
9	0	0

DATE	NUMBER OF MEN
1	10
2	10
3	7
4	7
5	13
6	13
7	10
8	10
9	10
10	10
11	10

```
C                                                                       CMN00020
C     PROGRAMMED BY KIM,JI SUNG                                         CMN00030
C     DIRECTED BY PROF.SOONDAL                                          CMN00040
C     SEOUL NATIONAL UNIVERSITY                                         CMN00050
C                                                                       CMN00060
C     LAST REVISED ON FEB.23,1983, VER.1.1                              CMN00070
C                                                                       CMN00080
      DIMENSION K(50,8),MS(50),IB(50),IE(50),IL(50),MN(50)              CMN00090
      INTEGER E(50),B(50),C(50),D(50),PROT(50)                          CMN00100
      INTEGER A(9,11),SS(50),S(50),SM                                   CMN00110
      OPEN(1,FILE='CPMMENDAT',MAXRECL=80,PAD='YES',BLANK='ZERO')        CMN00120
      OPEN(3,FILE='CPMMENOUT',CARRIAGECONTROL='FORTRAN',STATUS='FRESH') CMN00130
      DATA IR,IW/1,3/                                                   CMN00140
      READ(IR,55) M                                                     CMN00150
   55 FORMAT(I5)                                                        CMN00160
      DO 75 I=1,M                                                       CMN00170
      READ(IR,700) (K(I,J),J=1,8),MS(I)                                 CMN00180
  700 FORMAT(6I5,2I10,I5)                                               CMN00190
   75 CONTINUE                                                          CMN00200
      DO 35 I=1,M                                                       CMN00210
   35 PROT(I)=K(I,7)/K(I,8)                                             CMN00220
      DO 45 I=1,M                                                       CMN00230
      IF(K(I,2).EQ.0) GO TO 10                                          CMN00240
      IF(K(I,3).EQ.0) GO TO 20                                          CMN00250
      L=K(I,2)                                                          CMN00260
      N=K(I,3)                                                          CMN00270
      IF(B(L).GE.B(N)) GO TO 30                                         CMN00280
      E(I)=B(N)                                                         CMN00290
      B(I)=E(I)+PROT(I)                                                 CMN00300
      GO TO 45                                                          CMN00310
   10 E(I)=0                                                            CMN00320
      B(I)=E(I)+PROT(I)                                                 CMN00330
      GO TO 45                                                          CMN00340
   20 J=K(I,2)                                                          CMN00350
      E(I)=PROT(J)                                                      CMN00360
      B(I)=E(I)+PROT(I)                                                 CMN00370
      GO TO 45                                                          CMN00380
   30 E(I)=B(L)                                                         CMN00390
      B(I)=E(I)+PROT(I)                                                 CMN00400
   45 CONTINUE                                                          CMN00410
      C(M)=E(M)                                                         CMN00420
      D(M)=C(M)                                                         CMN00430
      DO 53 I=1,M                                                       CMN00440
            J=M-I+1                                                     CMN00450
      IF(K(J,3).EQ.0) GO TO 32                                          CMN00460
      L=K(J,2)                                                          CMN00470
      N=K(J,3)                                                          CMN00480
      D(L)=C(J)                                                         CMN00490
      C(L)=D(L)-PROT(L)                                                 CMN00500
      D(N)=C(J)                                                         CMN00510
      C(N)=D(N)-PROT(N)                                                 CMN00520
      GO TO 53                                                          CMN00530
   32 IF(K(J,2).EQ.0) GO TO 53                                          CMN00540
      L=K(J,2)                                                          CMN00550
      D(L)=C(J)                                                         CMN00560
      C(L)=D(L)-PROT(L)                                                 CMN00570
   53 CONTINUE                                                          CMN00580
      DO 60 I=1,M                                                       CMN00590
   60 S(I)=C(I)-E(I)                                                    CMN00600
      WRITE(IW,80)                                                      CMN00610
   80 FORMAT(1H1,//,30X,'*EVALUATION OF  INPUT DATA*'//5X,'NODE  PREDECECMN00620
     *SSORS       TOTAL       NORM     MEN     EARLY  LATE  SLACK'/31X,'M/HCMN00630
     *     DAILY M/H         START  START'/)                            CMN00640
      DO 65 I=1,M                                                       CMN00650
   65 WRITE(IW,300) (K(I,J),J=1,8),MS(I),E(I),C(I),S(I)                 CMN00660
  300 FORMAT(5X,I3,2X,5I3,5X,I5,6X,I4,4X,I3,4X,I5,3X,I4,2X,I5/)         CMN00670
      ICRT=C(M)+PROT(M)-1                                               CMN00680
      N=ICRT                                                            CMN00690
      DO 2 I=1,M                                                        CMN00700
      DO 3 J=1,N                                                        CMN00710
    3 A(I,J)=0                                                          CMN00720
```

```
  2 CONTINUE                                                          CMN00730
    DO 4 I=1,M                                                        CMN00740
    N1=E(I)+PROT(I)                                                   CMN00750
    E(I)=E(I)+1                                                       CMN00760
    LN=E(I)                                                           CMN00770
    DO 5 IK=LN,N1                                                     CMN00780
  5 A(I,IK)=MS(I)                                                     CMN00790
  4 CONTINUE                                                          CMN00800
    DO 6 J=1,N                                                        CMN00810
    IS=0                                                              CMN00820
    DO 7 I=1,M                                                        CMN00830
  7 IS=A(I,J)+IS                                                      CMN00840
    SS(J)=IS                                                          CMN00850
  6 CONTINUE                                                          CMN00860
    SM=0                                                              CMN00870
    DO 8 I=1,N                                                        CMN00880
  8 SM=SM+SS(I)*SS(I)                                                 CMN00890
    CALL NOBODY(A,M,N,S,SM,IB)                                        CMN00900
    DO 24 I=1,M                                                       CMN00910
    DO 23 J=1,N                                                       CMN00920
    IF(A(I,J).EQ.0) GO TO 23                                          CMN00930
    IE(I)=J                                                           CMN00940
    GO TO 24                                                          CMN00950
 23 CONTINUE                                                          CMN00960
 24 CONTINUE                                                          CMN00970
    DO 25 I=1,M                                                       CMN00980
 25 IL(I)=IE(I)+PROT(I)-1                                             CMN00990
    DO 510 J=1,N                                                      CMN01000
    IP=0                                                              CMN01010
    DO 540 I=1,M                                                      CMN01020
540 IP=A(I,J)+IP                                                      CMN01030
    MN(J)=IP                                                          CMN01040
510 CONTINUE                                                          CMN01050
    WRITE(IW,61)                                                      CMN01060
 61 FORMAT(1H1,//,30X,"SCHEDULE OF THE PROJECT"//20X,"NODE   START    CMN01070
   * FINISH")                                                         CMN01080
    DO 63 I=1,M                                                       CMN01090
 63 WRITE(IW,62) K(I,1),IE(I),IL(I)                                   CMN01100
 62 FORMAT(/,20X,I3,4X,I5,4X,I6)                                      CMN01110
    WRITE(IW,72)                                                      CMN01120
 72 FORMAT(//,20X,"DATE    NUMBER OF MEN")                            CMN01130
    DO 64 I=1,N                                                       CMN01140
 64 WRITE(IW,78) I,MN(I)                                              CMN01150
 78 FORMAT(/,20X,I4,7X,I5)                                            CMN01160
    STOP                                                              CMN01170
    END                                                               CMN01180
    SUBROUTINE  NOBODY(IA,M,N,IS,IBO,IB)                              CMN01190
    DIMENSION IA(M,N),IS(M),IB(N),IBLOCP(10)                          CMN01200
    DATA IR,IW/1,3/                                                   CMN01210
    IBL=5                                                             CMN01220
    IBLOCP(1)=1                                                       CMN01230
    IBLOCP(2)=5                                                       CMN01240
    IBLOCP(3)=7                                                       CMN01250
    IBLOCP(4)=8                                                       CMN01260
    IBLOCP(5)=9                                                       CMN01270
  9 IF(IBL.EQ.1) RETURN                                               CMN01280
    II=IBLOCP(IBL)                                                    CMN01290
    IBL=IBL-1                                                         CMN01300
    II=II-1                                                           CMN01310
310 MAXS=IS(IBLOCP(IBL))                                              CMN01320
    IPTM=IBLOCP(IBL)                                                  CMN01330
    IM=IBLOCP(IBL)                                                    CMN01340
    DO 17 I=IM,II                                                     CMN01350
    IF(MAXS.GE.IS(I)) GO TO 17                                        CMN01360
    MAXS=IS(I)                                                        CMN01370
    IPTM=I                                                            CMN01380
 17 CONTINUE                                                          CMN01390
    IF(MAXS.EQ.0) GO TO 9                                             CMN01400
410 IF(IS(IPTM).EQ.0) GO TO 310                                       CMN01410
    DO 400 JJ=1,N                                                     CMN01420
```

```
      IB(JJ)=IA(IPTM,JJ)                                          CMN01430
      IA(IPTM,JJ)=0                                               CMN01440
  400 CONTINUE                                                    CMN01450
      NI=N-1                                                      CMN01460
      DO 500 JJ=1,NI                                              CMN01470
      IK=JJ+1                                                     CMN01480
      IA(IPTM,IK)=IB(JJ)                                          CMN01490
  500 CONTINUE                                                    CMN01500
      IRTO=0                                                      CMN01510
      IP=0                                                        CMN01520
      DO 56 JJ=1,N                                                CMN01530
      DO 57 II=1,M                                                CMN01540
      IP=IP+IA(II,JJ)                                             CMN01550
   57 CONTINUE                                                    CMN01560
      IRTO=IP*IP+IRTO                                             CMN01570
      IP=0                                                        CMN01580
   56 CONTINUE                                                    CMN01590
      IF(IRTO.LT.IRO) GO TO 520                                   CMN01600
      IS(IPTM)=0                                                  CMN01610
      DO 530 JJ=1,N                                               CMN01620
      IA(IPTM,JJ)=IB(JJ)                                          CMN01630
  530 CONTINUE                                                    CMN01640
      GO TO 310                                                   CMN01650
  520 IS(IPTM)=IS(IPTM)-1                                         CMN01660
      IRO=IRTO                                                    CMN01670
      GO TO 410                                                   CMN01680
      END                                                         CMN01690
```

Ⅳ. 프로그램 이름 : TRANSV

1. 目 的

일반적인 수송문제에서 비용을 최소로 하는 수송계획을 하는데 그 목적이 있다.

2. 解 法

(1) 공급처와 수요처의 공급량과 수요량을 만족시키는 초기 가능해를 구한 다음, 각 route의 상대 비용을 계산하여 그 값이 음이 되는 route에 가능한 최대 흐름을 이동시키고, 다시 상대비용을 계산하여 그 값이 모두 0 이상이 될 때에 멈추는 primal-dual algorithm을 사용한다. 초기해는 Vogel 방법을 사용한다.

(2) 참고문헌

Harvey M. Wagner, Principle of Management Science, Prentice Hall, 1969, pp 213-231

3. 構造와 重要變數

TRANSV는 8개의 Subroutine으로 되어있고 각각의 기능은 다음과 같다.

(1) Subroutine TRANSV : 자료를 읽어들이고, 선택변수에 의해 초기 가능해를 구하는 Subroutine을 불러서 가능해를 구한 다음 primal-dual algorithm을 수행하여 최적해를 찾는다.

(2) Subroutine VOGEL : 선택변수가 3일때 초기 가능해를 구하는 Subroutine으로 Vogel approximation method를 사용한다.

(3) Subroutine BIG : 입력자료중 가장 큰 값을 갖는 것의 위치를 찾는다.

(4) Subroutine SMALL : 입력자료 중 가장 작은 값의 위치를 찾는다.

(5) Subroutine REDUNT : 초기 가능해에서 흐름이 0보다 큰 base의 갯수가 부족하게 될 때 base를 연결시킬 수 있는 route를 찾아 base에 첨가시킨다.

- **重要變數**

M1 : 공급처의 갯수

N1 : 수요처의 갯수

IC : 각 route의 비용을 의미한다.

IS : 각 공급처의 공급량

ID : 각 수요처의 수요량

IX : 각 route에 흐르는 양

MAXM : 초기 array를 잡기 위한 변수 (MAXM ≧ M1)

MAXN : 초기 array를 잡기 위한 변수 (MAXN ≧ N1)

KBUG : 출력에 관계되는 변수

= 0, 최적해만 출력

≧ 1, 매 회의 결과를 출력

4. 使用方法

(1) 入力

카드例	入力형태	변수와내용
1	2I5	MAXM, MAXN
2	3I4	M1, N1, KBUG
3~P	20I4	IC
p+1~q	20I4	ID
q+1~	20I4	IS

(2) 出力

- 공급량
- 수요량
- 각 route의 비용
- 초기해를 구한 방법
- 최적해
- 중간 단계의 결과 (KBUG ≧ 1때)

5. 참고사항

(1) 모든 입력자료는 I4로 되어 있으므로 이 이상의 자료는 FORMAT을 수정해야 한다.

(2) MAXM, MAXN의 값이 0으로 들어오면 각각 30으로 지정된다.

6. 使用例

(1) 問 題

	1	2	3	4	5	공급
1	16	16	13	22	17	50
2	14	14	13	19	15	60
3	19	19	20	23	40	50
4	40	10	40	11	12	50
수요	30	20	70	30	60	

(2) 入 力

카드	자 료
1	10 10
2	4 5 0
3	16 16 13 22 17 14 14 13 19 15
	19 19 20 23 40 40 10 40 11 12
4	30 20 70 30 60
5	50 60 50 50

(3) 出 力

```
  SUPPLY
S(1) = 50
S(2) = 60
S(3) = 50
S(4) = 50
  DEMAND
D(1) = 30
D(2) = 20
D(3) = 70
D(4) = 30
D(5) = 60
```

```
COST TABLE
 16    16    13    22    17
 14    14    13    19    15
 19    19    20    23    40
 40    10    40    11    12

INITIAL BASIC FEASIBLE SOLUTION
BY VOGEL-APPROXIMATION METHOD

  THIS IS INITIAL FEASIBLE
  SOLUTION FLOW
  0     0    50     0     0
  0     0     0     0    60
 30     0    20     0     0
  0    20     0    30     0
TOTAL COST = 3050

  OPTIMAL SOLUTION
TOTAL COST = 3030
  SUPPLY       FLOW        DEMAND
S(1)  ===   (50)  ===  D(3)
S(2)  ===   (20)  ===  D(3)
S(2)  ===   (40)  ===  D(5)
S(3)  ===   (30)  ===  D(1)
S(3)  ===   (20)  ===  D(2)
S(4)  ===   (30)  ===  D(4)
S(4)  ===   (20)  ===  D(5)
  NUMBER OF ITERATION= 2
```

```
C      PROGRAM TRANSV                                                    TRV00010
C                                                                        TRV00020
C      THIS IS A SIMPLIFIED VERSION OF TRANSI.                           TRV00030
C      ORIGINALLY PROGRAMMED BY PARK,HA YOUNG                            TRV00040
C      MODIFIED BY SO, YEONG SUP                                         TRV00050
G      DIRECTED BY PROF. PARK, SOONDAL                                   TRV00060
C      SEOUL NATIONAL UNIVERSITY                                         TRV00070
C                                                                        TRV00080
C      LAST REVISED ON FEB.21, 1983                                      TRV00090
C                                                                        TRV00100
       INTEGER IC(50,50),IX(50,50),IX1(50,50),                           TRV00110
      1 IC1(50,50),NBIG(2500),IS(50),LCR(50),IS1(50),                    TRV00120
      2 ISUROW(50),ID(50),LCC(50),ID1(50),JCOL(50),                      TRV00130
      3 IN(50),IE(51,51),IEC(51,51),ICIR(100),                           TRV00140
      4 JCIR(100),INDEX(2500,3)                                          TRV00150
       DATA M,N,ITER/5,6,0/                                              TRV00160
       OPEN(5,FILE='TRANSIDAT',MAXRECL=80,PAD='YES',BLANK='ZERO')        TRV00170
       OPEN(6,FILE='TRANSIOUT',CARRIAGECONTROL='FORTRAN')                TRV00180
       READ(M,3000)M1,N1,INIT,KBUG                                       TRV00190
 3000 FORMAT(4I4)                                                        TRV00200
       M3=M1+1                                                           TRV00210
       N3=N1+1                                                           TRV00220
       ICPR=M1+N1-1                                                      TRV00230
       READ(M,100) ((IC(I,J),J=1,N1),I=1,M1)                             TRV00240
       READ(M,100) (ID(J),J=1,N1)                                        TRV00250
  100 FORMAT(20I4)                                                       TRV00260
       READ(M,101)(IS(I),I=1,M1)                                         TRV00270
  101 FORMAT(20I4)                                                       TRV00280
       WRITE(N,300)                                                      TRV00290
  300 FORMAT(1H1,///,12X,' SUPPLY',/)                                    TRV00300
       DO 1 I=1,M1                                                       TRV00310
    1  WRITE(N,303)I,IS(I)                                               TRV00320
  303 FORMAT(10X,' S (',I2,') = ',I4,/)                                  TRV00330
       WRITE(N,306)                                                      TRV00340
  306 FORMAT(1H1,///,12X,' DEMAND ',//)                                  TRV00350
       DO 2 J=1,N1                                                       TRV00360
    2  WRITE(N,309)J,ID(J)                                               TRV00370
  309 FORMAT(10X,' D (',I2,') = ',I4,/)                                  TRV00380
       WRITE(N,312)                                                      TRV00390
  312 FORMAT(1H1,///,12X,' COST TABLE ',/)                               TRV00400
       DO 7 I=1,M1                                                       TRV00410
    7  WRITE(N,315)(IC(I,J),J=1,N1)                                      TRV00420
  315 FORMAT(/10X,20I6)                                                  TRV00430
       CALL VOGEL(N,M1,N1,IC,IS,ID,IX,IS1,ID1,IC1,NBIG,INDEX)            TRV00440
       WRITE(N,314)                                                      TRV00450
  314 FORMAT(1H1,////,10X,' THIS IS INITIAL FEASIBLE SOLUTION FLOW ',/)  TRV00460
       DO 6 I=1,M1                                                       TRV00470
       DO 6 J=1,N1                                                       TRV00480
    6  ITOT=ITOT+IC(I,J)*IX(I,J)                                         TRV00490
       WRITE(N,324)ITOT                                                  TRV00500
  324 FORMAT(/,2X,'TOTAL COST = ',I8)                                    TRV00510
       ICNT=0                                                            TRV00520
       DO 2000 I=1,M3                                                    TRV00530
       DO 2000 J=1,N3                                                    TRV00540
 2000 IEC(I,J)=0                                                         TRV00550
       DO 1994 I=1,M1                                                    TRV00560
       DO 1994 J=1,N1                                                    TRV00570
       IF(IX(I,J).EQ.0) GO TO 1994                                       TRV00580
       IEC(I,J)=1                                                        TRV00590
       ICNT=ICNT+1                                                       TRV00600
 1994 CONTINUE                                                           TRV00610
       IF(ICNT.GE.ICPR) GO TO 2003                                       TRV00620
       CALL REDUNT(N1,M1,IEC,IX,ICNT,JCOL,ISUROW,IN)                     TRV00630
 2003 CONTINUE                                                           TRV00640
       DO 2001 I=1,M1                                                    TRV00650
 2001 LCR(I)=0                                                           TRV00660
       DO 2002 J=1,N1                                                    TRV00670
 2002 LCC(J)=0                                                           TRV00680
       DO 3001 I=1,M3                                                    TRV00690
       IEC(I,N3)=0                                                       TRV00700
 3001 IE(I,N3)=0                                                         TRV00710
```

```
      DO 3002 J=1,N3                                                    TRV00720
      IEC(M3,J)=0                                                       TRV00730
 3002 IE(M3,J)=0                                                        TRV00740
      DO 3003 I=1,M3                                                    TRV00750
      DO 3003 J=1,N3                                                    TRV00760
 3003 IE(I,J)=0                                                         TRV00770
      IEC(M1,N3)=1                                                      TRV00780
      IEC(M3,N3)=1                                                      TRV00790
      DO 2006 J=1,N1                                                    TRV00800
      IF(IEC(M1,J).NE.1) GO TO 2006                                     TRV00810
      IE(M3,J)=IC(M1,J)                                                 TRV00820
      IEC(M3,J)=1                                                       TRV00830
      DO 2004 I=1,M1                                                    TRV00840
      IF(IEC(I,J).NE.1) GO TO 2004                                      TRV00850
      IE(I,N3)=IC(I,J)-IE(M3,J)                                         TRV00860
      IEC(I,N3)=1                                                       TRV00870
 2004 CONTINUE                                                          TRV00880
 2006 CONTINUE                                                          TRV00890
 2507 DO 2010 J=1,N1                                                    TRV00900
      IF(IEC(M3,J).EQ.1) GO TO 2010                                     TRV00910
      I=1                                                               TRV00920
 2512 IF(IEC(I,J).EQ.1.AND.IEC(I,N3).EQ.1) GO TO 2300                   TRV00930
      I=I+1                                                             TRV00940
      IF(I.LE.M1) GO TO 2512                                            TRV00950
      GO TO 2010                                                        TRV00960
 2300 IE(M3,J)=IC(I,J)-IE(I,N3)                                         TRV00970
      IEC(M3,J)=1                                                       TRV00980
      DO 2009 II=1,M1                                                   TRV00990
      IF(IEC(II,J).EQ.1.AND.IEC(II,N3).EQ.0) GO TO 2303                 TRV01000
      GO TO 2009                                                        TRV01010
 2303 IE(II,N3)=IC(II,J)-IE(M3,J)                                       TRV01020
      IEC(II,N3)=1                                                      TRV01030
 2009 CONTINUE                                                          TRV01040
 2010 CONTINUE                                                          TRV01050
      J=1                                                               TRV01060
 2515 IF(IEC(M3,J).NE.1) GO TO 2507                                     TRV01070
      J=J+1                                                             TRV01080
      IF(J.LE.N1) GO TO 2515                                            TRV01090
      DO 2012 I=1,M1                                                    TRV01100
      DO 2012 J=1,N1                                                    TRV01110
      IF(IX(I,J).NE.0) GO TO 2012                                       TRV01120
      IE(I,J)=IE(M3,J)+IE(I,N3)-IC(I,J)                                 TRV01130
 2012 CONTINUE                                                          TRV01140
      DO 2015 I=1,M1                                                    TRV01150
      NTE=I*N1                                                          TRV01160
      NIN=NTE-N1+1                                                      TRV01170
      DO 2015 JJ=NIN,NTE                                                TRV01180
      J=JJ-(I-1)*N1                                                     TRV01190
 2015 NBIG(JJ)=IE(I,J)                                                  TRV01200
      NN=M1*N1                                                          TRV01210
      CALL BIG(NBIG,NN,L)                                               TRV01220
      K=L/N1                                                            TRV01230
      L=L-K*N1                                                          TRV01240
      IF(L.NE.0) GO TO 2539                                             TRV01250
      L=N1                                                              TRV01260
      GO TO 2306                                                        TRV01270
 2539 K=K+1                                                             TRV01280
 2306 IF(IE(K,L).LE.0) GO TO 2650                                       TRV01290
      ITER=ITER+1                                                       TRV01300
      LCR(K)=LCR(K)+1                                                   TRV01310
      LCC(L)=LCC(L)+1                                                   TRV01320
      DO 2016 I=1,M1                                                    TRV01330
      DO 2016 J=1,N1                                                    TRV01340
      IF(IEC(I,J).EQ.0) GO TO 2016                                      TRV01350
      LCR(I)=LCR(I)+1                                                   TRV01360
      LCC(J)=LCC(J)+1                                                   TRV01370
 2016 CONTINUE                                                          TRV01380
      DO 2018 I=1,M1                                                    TRV01390
      DO 2018 J=1,N1                                                    TRV01400
 2018 IX1(I,J)=IEC(I,J)                                                 TRV01410
 2318 DO 2021 I=1,M1                                                    TRV01420
```

```
      IF(LCR(I).NE.1) GO TO 2021                                        TRV01430
      J=1                                                               TRV01440
 2569 IF(IX1(I,J).NE.0) GO TO 2575                                      TRV01450
      J=J+1                                                             TRV01460
      IF(J.LE.N1) GO TO 2569                                            TRV01470
      GO TO 2021                                                        TRV01480
 2575 IX1(I,J)=0                                                        TRV01490
      LCR(I)=0                                                          TRV01500
      LCC(J)=LCC(J)-1                                                   TRV01510
 2021 CONTINUE                                                          TRV01520
 2321 DO 2024 J=1,N1                                                    TRV01530
      IF(LCC(J).NE.1) GO TO 2024                                        TRV01540
      I=1                                                               TRV01550
 2581 IF(IX1(I,J).NE.0) GO TO 2587                                      TRV01560
      I=I+1                                                             TRV01570
      IF(I.LE.M1) GO TO 2581                                            TRV01580
      GO TO 2024                                                        TRV01590
 2587 IX1(I,J)=0                                                        TRV01600
      LCC(J)=0                                                          TRV01610
      LCR(I)=LCR(I)-1                                                   TRV01620
 2024 CONTINUE                                                          TRV01630
      I=1                                                               TRV01640
 2590 IF(LCR(I).NE.0.AND.LCR(I).NE.2) GO TO 2318                        TRV01650
      I=I+1                                                             TRV01660
      IF(I.LE.M1) GO TO 2590                                            TRV01670
      J=1                                                               TRV01680
 2596 IF(LCC(J).NE.0.AND.LCC(J).NE.2) GO TO 2321                        TRV01690
      J=J+1                                                             TRV01700
      IF(J.LE.N1) GO TO 2596                                            TRV01710
      K1=K                                                              TRV01720
      L1=L                                                              TRV01730
      I=1                                                               TRV01740
      J=1                                                               TRV01750
      ICIR(I)=K1                                                        TRV01760
      JCIR(J)=L1                                                        TRV01770
 2323 L1=JCIR(J)-1                                                      TRV01780
 2324 IF(L1.LT.1) GO TO 2608                                            TRV01790
      IF(IX1(K1,L1).NE.0) GO TO 2614                                    TRV01800
      L1=L1-1                                                           TRV01810
      GO TO 2324                                                        TRV01820
 2608 L1=JCIR(J)+1                                                      TRV01830
 2327 IF(IX1(K1,L1)) 2614,2611,2614                                     TRV01840
 2611 L1=L1+1                                                           TRV01850
      GO TO 2327                                                        TRV01860
 2614 J=J+1                                                             TRV01870
      JCIR(J)=L1                                                        TRV01880
      IF(L1.EQ.L) GO TO 2635                                            TRV01890
      K1=ICIR(I)+1                                                      TRV01900
 2330 IF(K1.GT.M1) GO TO 2626                                           TRV01910
      IF(IX1(K1,L1).NE.0) GO TO 2632                                    TRV01920
      K1=K1+1                                                           TRV01930
      GO TO 2330                                                        TRV01940
 2626 K1=ICIR(I)-1                                                      TRV01950
 2333 IF(IX1(K1,L1).NE.0) GO TO 2632                                    TRV01960
      K1=K1-1                                                           TRV01970
      GO TO 2333                                                        TRV01980
 2632 I=I+1                                                             TRV01990
      ICIR(I)=K1                                                        TRV02000
      GO TO 2323                                                        TRV02010
 2635 II=I                                                              TRV02020
      JJ=J                                                              TRV02030
      JM=2                                                              TRV02040
 2336 I1=JM-1                                                           TRV02050
      IJ=JM                                                             TRV02060
      I2=I1                                                             TRV02070
      NJM=JCIR(JM)                                                      TRV02080
      NI1=ICIR(I1)                                                      TRV02090
      NIJ=JCIR(IJ)                                                      TRV02100
      NI2=ICIR(I2)                                                      TRV02110
      JR=JM+1                                                           TRV02120
 2339 IF(JR.GT.JJ) GO TO 2647                                           TRV02130
```

```
      I3=JR-1                                                          TRV02140
      NJR=JCIR(JR)                                                     TRV02150
      NI3=ICIR(I3)                                                     TRV02160
      IF(IX(NI1,NJM).GT.IX(NI3,NJR)) GO TO 2644                        TRV02170
      JR=JR+1                                                          TRV02180
      GO TO 2339                                                       TRV02190
 2644 JM=JM+1                                                          TRV02200
      GO TO 2336                                                       TRV02210
 2647 IAL=IX(NI2,NIJ)                                                  TRV02220
      IEC(K,L)=1                                                       TRV02230
      IEC(NI2,NIJ)=0                                                   TRV02240
      J=1                                                              TRV02250
      NJ=JCIR(J)                                                       TRV02260
      DO 2027 I=1,II                                                   TRV02270
      NI=ICIR(I)                                                       TRV02280
      IX(NI,NJ)=IX(NI,NJ)+IAL                                          TRV02290
      J=J+1                                                            TRV02300
      NJ=JCIR(J)                                                       TRV02310
      IX(NI,NJ)=IX(NI,NJ)-IAL                                          TRV02320
 2027 CONTINUE                                                         TRV02330
      IF(KBUG.LE.0) GO TO 2003                                         TRV02340
      WRITE(N,316) ITER                                                TRV02350
  316 FORMAT(///20X,' THIS IS FEASIBLE SOLUTION FLOW ',                TRV02360
     1/30X,' AT ITERATION ',I4,/)                                      TRV02370
      DO 317 I=1,M1                                                    TRV02380
      WRITE(N,315) (IX(I,J),J=1,N1)                                    TRV02390
  317 CONTINUE                                                         TRV02400
      ITOT=0                                                           TRV02410
      DO 406 I=1,M1                                                    TRV02420
      DO 406 J=1,N1                                                    TRV02430
  406 ITOT=ITOT+IC(I,J)*IX(I,J)                                        TRV02440
      WRITE(N,324) ITOT                                                TRV02450
      GO TO 2003                                                       TRV02460
 2650 WRITE(N,359)                                                     TRV02470
  359 FORMAT(1H1,///,12X,' OPTIMAL SOLUTION ',//)                      TRV02480
      ITOT=0                                                           TRV02490
      DO 2030 I=1,M1                                                   TRV02500
      DO 2030 J=1,N1                                                   TRV02510
 2030 ITOT=ITOT+IC(I,J)*IX(I,J)                                        TRV02520
      WRITE(N,324)ITOT                                                 TRV02530
      WRITE(N,401)                                                     TRV02540
  401 FORMAT(//9X,' SUPPLY ',5X,' FLOW ',6X,' DEMAND ',/)              TRV02550
      DO 2033 I=1,M1                                                   TRV02560
      DO 2033 J=1,N1                                                   TRV02570
      IF(IX(I,J).EQ.0) GO TO 2033                                      TRV02580
      WRITE(N,362) I,IX(I,J),J                                         TRV02590
  362 FORMAT(/,10X,' S (',I2,') === (',I4,') ===  D (',I2,')')         TRV02600
 2033 CONTINUE                                                         TRV02610
      WRITE(N,356)ITER                                                 TRV02620
  356 FORMAT(///,12X,'NUMBER OF ITERATION = ',I4)                      TRV02630
 2342 STOP                                                             TRV02640
      END                                                              TRV02650
      SUBROUTINE BIG(NBIG,NN,L)                                        TRV02660
      INTEGER IMUL                                                     TRV02670
      INTEGER NBIG(2500)                                               TRV02680
      JM=1                                                             TRV02690
   20 L=JM                                                             TRV02700
      JR=JM+1                                                          TRV02710
   30 IF(JR.GT.NN) GO TO 45                                            TRV02720
      IF(NBIG(JM).LT.NBIG(JR)) GO TO 35                                TRV02730
      JR=JR+1                                                          TRV02740
      GO TO 30                                                         TRV02750
   35 JM=JM+1                                                          TRV02760
      GO TO 20                                                         TRV02770
   45 RETURN                                                           TRV02780
      END                                                              TRV02790
      SUBROUTINE SMALL(NBIG,NN,L)                                      TRV02800
      INTEGER IMUL                                                     TRV02810
      INTEGER NBIG(2500)                                               TRV02820
      JM=1                                                             TRV02830
   20 L=JM                                                             TRV02840
```

```
      JR=JM+1                                                          TRV02850
   30 IF(JR.GT.NN) GO TO 45                                            TRV02860
      IF(NBIG(JM).GT.NBIG(JR)) GO TO 35                                TRV02870
      JR=JR+1                                                          TRV02880
      GO TO 30                                                         TRV02890
   35 JM=JM+1                                                          TRV02900
      GO TO 20                                                         TRV02910
   45 RETURN                                                           TRV02920
      END                                                              TRV02930
      SUBROUTINE VOGEL(N,M1,N1,IC,IS,ID,IX,IS1,ID1,IC1,NBIG,INDEX)     TRV02940
C*****INITIAL SOLUTION BY VOGEL APPROXIMATION METHOD*****              TRV02950
      INTEGER MAXM,MAXN,IMUL                                           TRV02960
      INTEGER IC(50,50),IS(50),ID(50),IX(50,50),                       TRV02970
     1 IS1(50),ID1(50),IC1(50,50),NBIG(2500),INDEX(2500,3)             TRV02980
      DO 500 I=1,M1                                                    TRV02990
      DO 500 J=1,N1                                                    TRV03000
      IC1(I,J)=IC(I,J)                                                 TRV03010
  500 IX(I,J)=0                                                        TRV03020
      DO 501 I=1,M1                                                    TRV03030
  501 IS1(I)=IS(I)                                                     TRV03040
      DO 502 J=1,N1                                                    TRV03050
  502 ID1(J)=ID(J)                                                     TRV03060
    1 I=1                                                              TRV03070
    3 IF(IS1(I).NE.0) GO TO 6                                          TRV03080
      I=I+1                                                            TRV03090
      IF(I.LE.M1) GO TO 3                                              TRV03100
      GO TO 9                                                          TRV03110
    6 II=0                                                             TRV03120
      DO 103 I=1,M1                                                    TRV03130
      IF(IS1(I).EQ.0) GO TO 103                                        TRV03140
      II=II+1                                                          TRV03150
      DO 100 J=1,N1                                                    TRV03160
  100 NBIG(J)=IC1(I,J)                                                 TRV03170
      CALL SMALL(NBIG,N1,L)                                            TRV03180
      MIN1=IC(I,L)                                                     TRV03190
      INDEX(II,1)=I                                                    TRV03200
      INDEX(II,2)=L                                                    TRV03210
      NBIG(L)=10000                                                    TRV03220
      CALL SMALL(NBIG,N1,L)                                            TRV03230
      MIN2=IC(I,L)                                                     TRV03240
      INDEX(II,3)=MIN2-MIN1                                            TRV03250
  103 CONTINUE                                                         TRV03260
      DO 109 J=1,N1                                                    TRV03270
      IF(ID1(J).EQ.0) GO TO 109                                        TRV03280
      II=II+1                                                          TRV03290
      DO 106 I=1,M1                                                    TRV03300
  106 NBIG(I)=IC1(I,J)                                                 TRV03310
      CALL SMALL(NBIG,M1,L)                                            TRV03320
      MIN1=IC(L,J)                                                     TRV03330
      INDEX(II,1)=L                                                    TRV03340
      INDEX(II,2)=J                                                    TRV03350
      NBIG(L)=10000                                                    TRV03360
      CALL SMALL(NBIG,M1,L)                                            TRV03370
      MIN2=IC(L,J)                                                     TRV03380
      INDEX(II,3)=MIN2-MIN1                                            TRV03390
  109 CONTINUE                                                         TRV03400
      DO 112 I=1,II                                                    TRV03410
  112 NBIG(I)=INDEX(I,3)                                               TRV03420
      CALL BIG(NBIG,II,L)                                              TRV03430
      K=INDEX(L,1)                                                     TRV03440
      L=INDEX(L,2)                                                     TRV03450
      IF(IS1(K)-ID1(L)) 50,53,56                                       TRV03460
   50 IX(K,L)=IS1(K)                                                   TRV03470
      ID1(L)=ID1(L)-IS1(K)                                             TRV03480
      IS1(K)=0                                                         TRV03490
      DO 115 J=1,N1                                                    TRV03500
  115 IC1(K,J)=10000                                                   TRV03510
      GO TO 1                                                          TRV03520
   53 IX(K,L)=IS1(K)                                                   TRV03530
      IS1(K)=0                                                         TRV03540
      ID1(L)=0                                                         TRV03550
```

```
      DO 118 I=1,M1                                                     TRV03560
  118 IC1(I,L)=10000                                                    TRV03570
      DO 121 J=1,N1                                                     TRV03580
  121 IC1(K,J)=10000                                                    TRV03590
      GO TO 1                                                           TRV03600
   56 IX(K,L)=ID1(L)                                                    TRV03610
      IS1(K)=IS1(K)-ID1(L)                                              TRV03620
      ID1(L)=0                                                          TRV03630
      DO 124 I=1,M1                                                     TRV03640
  124 IC1(I,L)=10000                                                    TRV03650
      GO TO 1                                                           TRV03660
    9 WRITE(N,318)                                                      TRV03670
  318 FORMAT(///,2X,'INITIAL BASIC FEASIBLE SOLUTION',/,2X,'BY VOGEL-APPTRV03680
     *ROXIMATION METHOD',/)                                             TRV03690
      RETURN                                                            TRV03700
      END                                                               TRV03710
      SUBROUTINE REDUNT(N1,M1,IE1,IX,IK,JCOL,ISUROW,IN)                 TRV03720
      INTEGER MAXM,MAXN,MAXM1,MAXN1                                     TRV03730
      INTEGER IE1(51,51),IX(50,50),JCOL(50),                            TRV03740
     1 ISUROW(50),IN(50)                                                TRV03750
      N=3                                                               TRV03760
      NUM=M1+N1-1                                                       TRV03770
      DO 6600 J=1,N1                                                    TRV03780
      JJ=0                                                              TRV03790
      DO 6620 I=1,M1                                                    TRV03800
      IF(IE1(I,J).NE.1) GO TO 6620                                      TRV03810
      JJ=JJ+1                                                           TRV03820
      IROW=I                                                            TRV03830
 6620 CONTINUE                                                          TRV03840
      IF(JJ.GT.1) GO TO 6600                                            TRV03850
      II=0                                                              TRV03860
      DO 1000 JKK=1,N1                                                  TRV03870
      IF(IE1(IROW,JKK).NE.1) GO TO 1000                                 TRV03880
      IF(JKK.EQ.J) GO TO 1000                                           TRV03890
      II=II+1                                                           TRV03900
      JCOL(II)=JKK                                                      TRV03910
 1000 CONTINUE                                                          TRV03920
      DO 6640 IKK=1,II                                                  TRV03930
      IF(II.EQ.0) GO TO 7777                                            TRV03940
      I=JCOL(IKK)                                                       TRV03950
      IN(IKK)=0                                                         TRV03960
      INN=0                                                             TRV03970
      DO 300 J2=1,M1                                                    TRV03980
  300 ISUROW(J2)=0                                                      TRV03990
      DO 6660 J1=1,M1                                                   TRV04000
      IF(IE1(J1,I).NE.1) GO TO 6660                                     TRV04010
      IF(J1.EQ.IROW) GO TO 6660                                         TRV04020
      IN(IKK)=IN(IKK)+1                                                 TRV04030
      INN=IN(IKK)                                                       TRV04040
      ISUROW(INN)=J1                                                    TRV04050
 6660 CONTINUE                                                          TRV04060
      IF(INN.EQ.0) GO TO 6640                                           TRV04070
      DO 6680 IL=1,INN                                                  TRV04080
      ILL=ISUROW(IL)                                                    TRV04090
      IKC=0                                                             TRV04100
      DO 6700 IM=1,N1                                                   TRV04110
      IF(IE1(ILL,IM).NE.1) GO TO 6700                                   TRV04120
      IF(IM.EQ.I) GO TO 6700                                            TRV04130
      IKC=IKC+1                                                         TRV04140
 6700 CONTINUE                                                          TRV04150
      IF(IKC.EQ.0) GO TO 7777                                           TRV04160
      GO TO 6640                                                        TRV04170
 6680 CONTINUE                                                          TRV04180
 6640 CONTINUE                                                          TRV04190
      DO 200 IT=1,II                                                    TRV04200
      IF(IN(IT).NE.0) GO TO 6600                                        TRV04210
  200 CONTINUE                                                          TRV04220
 7777 CONTINUE                                                          TRV04230
      DO 900 IMM=1,N1                                                   TRV04240
      IF(IE1(IROW,IMM).EQ.0) GO TO 17                                   TRV04250
  900 CONTINUE                                                          TRV04260
```

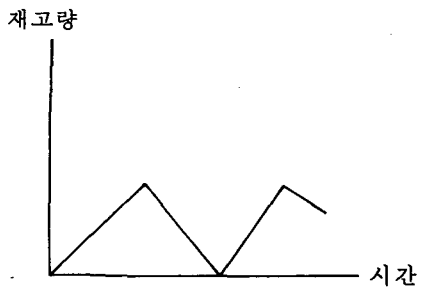
```
  17 IE1(IROW,IMM)=1                                                    TRV04270
     IK=IK+1                                                            TRV04280
     IF(IK.EQ.NUM) GO TO 47                                             TRV04290
6600 CONTINUE                                                           TRV04300
  47 RETURN                                                             TRV04310
     END                                                                TRV04320
```

7. 프로그램 이름 : INVENF

1. 目 的

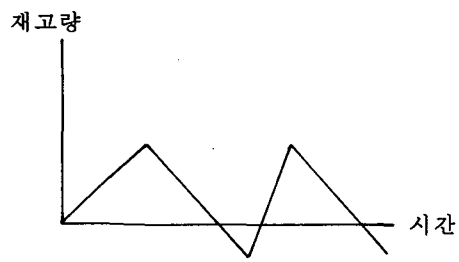
이 프로그램은 OR프로그램集(朴淳遠, 大英社, 1983)의 프로그램 INVENT를 補完하기 위한 것으로써 다음 두가지 형태의 在庫문제를 다룬다.

첫째로 Subroutine TYPE 7은 다음 형태의 재고문제 즉



형태의 문제를 다루고

둘째로 Subroutine TYPE 8은 다음 형태의 재고문제 즉



형태의 재고 문제를 다룬다.

2. 解法

참고문헌 姜錫昊 Operations Research 英志文化社, 1980

3. 構造와 重要變數

INVENF는 2개의 Subroutine TYPE 7과 TYPE 8로 되어 있고 각각의 기능은 다음과 같다.

(1) Subroutine TYPE 7 : 재고고갈을 허용하지 않는 경우의 문제를 푼다.

(2) Subroutine TYPE 8 : 재고고갈을 허용하는 경우의 문제를 푼다.

- **重要變數**

D : 수요량
AQ : 매일 도착량
UQ : 매일 수요량 (AQ ⟩ UQ)
RK : 가동준비비 또는 주문비
A : 단위제품당 재고비용 비율
P : 재고고갈 비용
C : 단위 가격

4. 使用方法

(1) 入 力

카드例	入力형태	변수와 내용
1	3F10.2	RK, A, P
2	F10.2, I5	C,D
3	2I5	AQ, UQ

〈주의〉 D, AQ, DQ는 INTEGER, VARIABLE

(2) 出 力

① 재고고갈을 허용하지 않는 경우
- 최적 주문량
- 총 재고비용
- 연간 주문 횟수

② 재고고갈을 허용하는 경우
- 최적 주문량
- 총 재고비용
- 허용 가능한 재고고갈량

5. 참고사항

INVENF는 하나의 독립된 프로그램으로도 사용할 수 있고, 다른 프로그램의 부프로그램으로도 사용할 수 있다. 이때는 COMMON 문장만 주의하면 된다.

6. 使用例

(1) 재고고갈을 허용하지 않는 경우

① 問 題

주문비 : 200
재고비용 비율 : 0.2
단위 가격 : 35
매일 도착량 : 50
매일 수요량 : 20
재고 고갈비 : 0
수요량 : 500

② 入 力

카드	자	료	
1	200.00	0.20	0.00
2	35.00	500	
3	50 20		

③ 出 力

```
THIS IS INVENTORY MODEL WITH UNIFORM
DELIVERY AND NO BACKORDERING

* INITIAL CONDITIONS

    SETUP COST IS     200.00
    HOLDING COST IS     7.00
    ITEM PRICE IS      35.00
    DEMAND IS          500
    DAILY ARRIVAL IS    50
    DAILY SALES IS      20

* RESULTS

    OPTIMAL ORDER QUANTITY IS     169.03

    TOTAL COST IS     447.21

    NUMBER OF ORDERING IS 2.96
```

(2) 재고고갈을 허용하는 경우

① 問 題

주문비 : 200
재고비용 비율 : 0.2
단위 가격 : 25
수요량 : 500
매일 도착량 : 50
매일 수요량 : 20
재고고갈 비용 : 10

② 入 力

카드	자	료	
1	200.00	0.20	10.00
2	25.00	500	
3	50 20		

③ 出 力

```
THIS IS INVENTORY MODEL WITH UNIFORM
DELIVERY AND BACKORDERING

* INITIAL CONDITIONS

    SETUP COST IS     200.00
    HOLDING COST IS     5.00
    ITEM PRICE IS      25.00
    DEMAND IS          500
    DAILY ARRIVAL IS    50
    DAILY SALES IS      20
    PENALTY COST IS     10.00

* RESULTS

    OPTIMAL ORDER QUANTITY IS     244.95

    TOTAL COST IS     748.45

    ALLOWABLE BACKORDER QUANTITY IS 81.65
```

```
C      PROGRAM INVENF                                                    IVF00010
C                                                                        IVF00020
C      PROGRAMMED BY KIM, TAE HO                                         IVF00030
C      DIRECTED BY PROF. PARK, SOON DAL                                  IVF00040
C      SEOUL NATIONAL UNIVERSITY                                         IVF00050
C                                                                        IVF00060
C      LAST REVISED; DEC.18,1982                                         IVF00070
C                                                                        IVF00080
C************************************************                        IVF00090
C                                                                        IVF00100
C                                                                        IVF00110
C     D; DEMAND QUANTITY                                                 IVF00120
C     AQ; ARRIVAL QUANTITY PER DAY                                       IVF00130
C     UQ; USE OR SALE QUANTITY PER DAY (AQ>UQ)                           IVF00140
C     RK; SETUP COST OR ORDERING COST                                    IVF00150
C     A; INVENTORY COST RATE PER UNIT ITEM                               IVF00160
C     P; PENALTY COST                                                    IVF00170
C     C;ITEM PRICE                                                       IVF00180
C                                                                        IVF00190
C                                                                        IVF00200
C************************************************                        IVF00210
      COMMON IR,IW                                                       IVF00220
      INTEGER D,AQ,UQ                                                    IVF00230
      IR=1                                                               IVF00240
      IW=3                                                               IVF00250
   20 READ(IR,9,END=50)RK,A,P                                            IVF00260
      READ(IR,19) C,D                                                    IVF00270
      READ(IR,29)AQ,UQ                                                   IVF00280
      IF(P.EQ.0.) GO TO 30                                               IVF00290
      CALL TYPE8(RK,A,P,C,D,AQ,UQ)                                       IVF00300
      GO TO 20                                                           IVF00310
   30 CALL TYPE7(RK,A,C,D,AQ,UQ)                                         IVF00320
      GO TO 20                                                           IVF00330
    9 FORMAT(3F10.2)                                                     IVF00340
   19 FORMAT(F10.2,I5)                                                   IVF00350
   29 FORMAT(2I5)                                                        IVF00360
   50 STOP                                                               IVF00370
      END                                                                IVF00380
      SUBROUTINE TYPE7(RK,A,C,D,AQ,UQ)                                   IVF00390
      COMMON IR,IW                                                       IVF00400
      INTEGER D,AQ,UQ                                                    IVF00410
      REAL N                                                             IVF00420
       WRITE(IW,7)                                                       IVF00430
    7 FORMAT(1H1,/////20X,'THIS IS INVENTORY MODEL WITH UNIFORM DELIVERYIVF00440
     * AND NO BACKORDERING')                                            IVF00450
      H=A*C                                                              IVF00460
      WRITE(IW,8) RK,H,C,D,AQ,UQ                                         IVF00470
    8 FORMAT(////30X,'* INITIAL CONDITIONS'//35X,'SETUP COST IS',F10.2   IVF00480
     */35X,'HOLDING COST IS',F10.2/35X,'ITEM PRICE IS',F10.2/35X,        IVF00490
     *'DEMAND IS',5X,I5/35X,'DAILY ARRIVAL IS',I5/35X,                   IVF00500
     *'DAILY SALES IS',I5/////)                                          IVF00510
      Q=SQRT(2*D*RK/((1-UQ/AQ)*H))                                       IVF00520
      TC=SQRT(2*D*RK*(1-UQ/AQ))                                          IVF00530
      N=D/Q                                                              IVF00540
      WRITE(IW,9) Q,TC,N                                                 IVF00550
    9 FORMAT(30X,'* RESULTS'//35X,'OPTIMAL ORDER QUANTITY IS',F10.2//    IVF00560
     *35X,'TOTAL COST IS',F10.2,//35X,'NUMBER OF ORDERING IS',F5.2)      IVF00570
      RETURN                                                             IVF00580
      END                                                                IVF00590
      SUBROUTINE TYPE8(RK,A,P,C,D,AQ,UQ)                                 IVF00600
      COMMON IR,IW                                                       IVF00610
      INTEGER D,AQ,UQ                                                    IVF00620
       WRITE(IW,7)                                                       IVF00630
    7 FORMAT(1H1,/////20X,'THIS IS INVENTORY MODEL WITH UNIFORM DELIVERYIVF00640
     * AND BACKORDERING')                                               IVF00650
      H=A*C                                                              IVF00660
      WRITE(IW,8) RK,H,C,D,AQ,UQ,P                                       IVF00670
    8 FORMAT(////30X,'* INITIAL CONDITIONS'//35X,'SETUP COST IS',F10.2   IVF00680
     */35X,'HOLDING COST IS',F10.2/35X,'ITEM PRICE IS',F10.2/35X,        IVF00690
     *'DEMAND IS',5X,I5/35X,'DAILY ARRIVAL IS',I5/35X,                   IVF00700
     *'DAILY SALES IS',I5/35X,'PENALTY COST IS',F10.2/////)              IVF00710
```

```
      Q=SQRT((2*RK*D/(H*(1-UQ/AQ)))*(H+P)/P)                          IVF00720
      B=Q*(1-UQ/AQ)*(H/(H+P))                                         IVF00730
      TC=RK*D/Q+(H*AQ/(2*Q*(AQ-UQ)))*(Q*((AQ-UQ)/AQ)-B)**2            IVF00740
     &+P*AQ*B**2/(2*Q*(AQ-UQ))                                        IVF00750
      WRITE(IW,9) Q,TC,B                                              IVF00760
    9 FORMAT(30X,'* RESULTS'//35X,'OPTIMAL ORDER QUANTITY IS',F10.2//  IVF00770
     *35X,'TOTAL COST IS',F10.2,//35X,                                IVF00780
     *'ALLOWABLE BACKORDER QUANTITY IS',F10.2)                        IVF00790
      RETURN                                                          IVF00800
      END                                                             IVF00810
```

참고문헌

1. 朴淳達, OR 프로그램集, 大英社, 1983
2. 朴淳達, OR과 電算프로그램, 군사운영
분석학회지 8권 2호, 1982