บรรณานุกรม

ภาษาไทย

หนังสือ

ชัยนนท์ ศรีสุภินานนท์. การควบคุมคุณภาพ. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตธนบุรี, 2525.

เดือน สินธุพันธ์ประทุม. โปรแกรมคอมพิวเตอร์ภาษาโฟแทรน 4. พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : โรงพิมพ์ฝ่าอักษร, 2520.

พรรณี ประเสริฐวงศ์. แรงงานสัมพันธ์และการจัดการอุตสาหกรรม. พิมพ์ครั้งที่ 4. กรุงเทพมหานคร : หอรัตนชัยการพิมพ์, 2524.

ภาณุสันต์ เชื้อหงษ์ทอง. การเขียนโปรแกรมคำสั่งโดยใช้ภาษาเบสิค. เชียงใหม่ : ภาควิชาคณิตศาสตร์ มหาวิทยาลัยเชียงใหม่, 2522.

เอกสารอื่น ๆ

เมธี เอกะสิงห์. "โปรแกรมสำเร็จรูปเพื่อการวิเคราะห์เชิงสถิติสำหรับไมโครคอมพิวเตอร์." เชียงใหม่ : สมาคมสถิติแห่งประเทศไทย, 2527. (อัดสำเนา)

สมเกียรติ เกตุเอี่ยม. "การเปรียบเทียบวิธีประมาณค่าพารามิเตอร์ เพื่อการทดลองการแจกแจงของประชากรที่ให้ค่าสถิติไคสแควร์ต่ำสุด." วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2529.

สมบูรณ์วัลย์ เหมคำศรี. "ขั้นตอนการดำเนินงานและปัญหาในการใช้คอมพิวเตอร์ในงานวิจัย." เชียงใหม่ : สมาคมสถิติแห่งประเทศไทย, 2527. (อัดสำเนา)

อุตสาหกรรม, กระทรวง. "มาตรฐานผลิตภัณฑ์อุตสาหกรรมสับประรดกระป๋อง .." กรุงเทพมหานคร : กระทรวงอุตสาหกรรม, 2522.

————. "การปลูกสับประรด." กรุงเทพมหานคร : กระทรวงอุตสาหกรรม, 2522.

ภาษาต่างประเทศ

หนังสือ

Borland International, Inc. Turbo Basic Owner's Handbook. California: 1987.

บทความ

Blischke, W.R. "Estimating the Parameters of Mixtures of Binomial Distribution." Journal of the American Statistical Association 59 (June 1964): 510-528.

Breakwell, J.V. "Economically Optimum Acceptance Test." Journal of the American Statistical Association 51 (June 1956): 243-256.

Case, Kenneth E., and J. Bert Keats. "On the Selection of a Prior Distribution in Bayesian Acceptance Sampling." Journal of Quality Technology 14 (January 1982): 10-18.

Case, Kenneth E., Bennett, G. Kemble, and Schmidt, J.W. "The Effect of Inspection Error on Average Outgoing Quality." Journal of Quality Technology 7 (January 1975) : 28-33.

Hald, A. Compound and Hypergeometric Distribution and A System of Single Sampling Inspection Plans Based On Prior Distribution and Cost." Techometrics 2 (August 1960):275-340.

Minton, G. "Verification Error in Single Sampling Inspection Plans for Processing Survey Data." Journal of the American Statistical Association 67 (March 1972):46-54.

Smith, B.E. "The Economics of Sampling Inspection." Industrial Quality Control 21 (March 1965) : 453-458.

เอกสารอื่น ๆ

Behrooz Parkhideh. "Estimating the Parameters of Mixed Binomial and Mixed Polya Distributions." Master's thesis, Department of Industrial Engineering and Management, Oklahoma State University, 1980.

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

โปรแกรมการประมาณค่าพารามิเตอร์ของการแจกแจงของผลิตภัณฑ์ยางรถในงานควบคุมคุณภาพ

```
'                    ****************
'                    * MAIN PROGRAM *
'                    ****************
'  ***************************************************************
'  *                                                             *
'  * THIS PROGRAM ESTABLISHES THE INTENT OF THE USER AND CALLS THE *
'  * APPROPRIATE SUBROUTINES TO  (1)ESTIMATE THE PARAMETERS OF THE *
'  * SELECTED PRIOR DISTRIBUTION AND  (2)PRINT THE RESULTS.       *
'  *                                                             *
'  ***************************************************************
DEFDBL A,D,F,I,O,P,Q,S,X,Y,Z
DIM X(10),XMAX(10),XMIN(10),D(20),Q(20)
DIM ATWOB(3,6),ATREB(5,6),AFORB(7,6)
DIM AONEP(2,6),ATWOP(5,6),ATREP(8,6)
DIM NPROB(3000),PROB(3000),NOO(3000)
'  * Read data
'  * Data for ATWOB
FOR RC = 1 TO 6
  FOR RR = 1 TO 3
    READ ATWOB(RR,RC)
  NEXT RR
NEXT RC
DATA .3,.01,.9,.3,.002,.7,.4,.02,.8,.2,.1,.8,.3,.1,.4,.6,.01,.7
'  * Data for ATREB
FOR RC = 1 TO 6
  FOR RR = 1 TO 5
    READ ATREB(RR,RC)
  NEXT RR
NEXT RC
DATA .3,.33,.01,.2,.8,.33,.3,.01,.4,.9,.33,.33,.9,.4,.05,.3,.6,.01,.3,.8
DATA .2,.7,.9,.1,.01,.3,.4,.001,.2,.99
'  * Data for AFORB
FOR RC = 1 TO 6
  FOR RR = 1 TO 7
    READ AFORB(RR,RC)
  NEXT RR
NEXT RC
DATA .25,.2,.3,.001,.1,.3,.9,.25,.2,.3,.01,.1,.2,.8,.2,.3,.4,.001,.2,.4
DATA .9,.2,.1,.4,.9,.4,.2,.0001,.3,.4,.25,.002,.3,.6,.9,.1,.6,.1,.01,.6
DATA .7,.9
'  * Data for AONEP
FOR RC = 1 TO 6
  FOR RR = 1 TO 2
    READ AONEP(RR,RC)
  NEXT RR
NEXT RC
DATA 5,2,1,5,9,7,5,15,10,6,4.5,9.6
'  * Data for ATWOP
FOR RC = 1 TO 6
  FOR RR = 1 TO 5
    READ ATWOP(RR,RC)
  NEXT RR
NEXT RC
DATA .15,1,5,2,12,.15,2,12,1,5,.35,1,5,2,12,.3,3,6,10,8,.3,4,5,2,12,.2
DATA 10,12,3,15
'  * Data for ATREP
```

```
      FOR RC = 1 TO 6
        FOR RR = 1 TO 8
          READ ATREP(RR,RC)
        NEXT RR
      NEXT RC
      DATA .15,.75,1,5,2,12,3,15,.1,.55,1,5,10,7,5,15,.3,.2,1,5,2,12,5,15,.35
      DATA .45,1,5,3,15,9,6,.1,.3,1,5,7,13,9,3,.15,.35,9,1,5,6,5,12
      '  *
      '  * INITIALIZE *
      '  *
      IOUTPT = 0
      ICSOD = 0
      '  *
111   ISAVE = 0
      '  *
      '  * MIXED DISTRIBUTION REPRESENTATION *
      '  *
1     IJIJ = 0
      IIII = 0
      NCOMP = 0
      NCPOL = 0
      '  *
      '  * STARTING SOLUTION VECTOR POINTER *
      '  *
      NSTS = 1
      '  *
      '  * INCRMN IS INCREMENT FOR LIM *
      '  *
      INCRMN=0
      '  *
      '  * PRINT INTRODUCTION *
      '  *
1001  CLS : PRINT : PRINT
      PRINT "     FITTING A PRIOR DISTRIBUTION TO DATA " : PRINT
      PRINT " THIS PROGRAM DETERMINES THE PARAMETERS OF A "
      PRINT "  (1) MIXED BINOMIAL DISTRIBUTION "
      PRINT "  (2) POLYA AND MIXED POLYA DISTRIBUTION " : PRINT
      PRINT " WHICH ONE DO YOU DESIRE?  " : PRINT
      INPUT " ENTER  1 OR 2  (TO STOP THE PROGRAM ENTER 3)   : " ,JCOD
      PRINT
      '  *
      '  * TRANSFER TO APPROPRIATE PROTION
      '  *
      IF JCOD = 3 THEN GOTO 9
      IF JCOD = 1 THEN GOTO 10
      IF JCOD = 2 THEN GOTO 11
      PRINT " YOU DID NOT ENTER   1,2, OR 3 "
      GOTO 1001
9     INPUT " YOU WANT TO STOP THE PROGRAM.    ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000
      GOTO 1001
10    INPUT " YOU WANT TO USE MIXED BINOMIAL PRIOR.    ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 105
      GOTO 1001
      PRINT : PRINT " YOU DO NOT WANT MIXED BINOMIAL    ...NOW TRY."
      GOTO 1001
```

```
105   CLS : PRINT : PRINT
      PRINT " THIS PROGRAM PERMITS YOU TO FIT ONE OF"
      PRINT " THE FOLLOWING MODELS TO YOUR DATA : "
      PRINT "  (1) A TWO COMPONENT MIXED BINOMIAL "
      PRINT "  (2) A THREE COMPONENT MIXED BINOMIAL "
      PRINT "  (3) A FOUR COMPONENT MIXED BINOMIAL "
      PRINT " WHICH DO YOU WANT TO DO ? " : PRINT
1055  INPUT " ENTER 1,2, OR 3  (TO STOP THE PROGRAM ENTER 4)   : " ,NCOMP
      PRINT
      IF NCOMP > 0 AND NCOMP < 4 THEN GOTO 112
      IF NCOMP = 4 THEN GOTO 106
      PRINT " YOU DID NOT ENTER   1,2,3, OR 4 "
      GOTO 1055
106   PRINT
      INPUT " YOU WANT TO STOP THE PROGRAM     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000
      GOTO 1055
112   PNCOMP = NCOMP+1
      PRINT " SO, YOU WANT TO APPROXIMATE THE PARAMETER OF A " ;
      PRINT USING " #  COMPONENT MIXED BINOMIAL."; PNCOMP
      INPUT " ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 116
      PRINT : PRINT USING " YOU DID NOT WANT A # COMPONENTS "; PNCOMP ;
      PRINT "MIXED BINOMIAL    ...TRY AGAIN."
      GOTO 105
      '  *
      '  * SET PARAMETERS TO INDICATE MIXED BINOM & CALCULATE # OF VARIABLES.
      '  *
116   IJIJ = 1
      IIII = NCOMP*2+1
      GOTO 220
      '  * IF POLYA IS CHOOSEN...
11    INPUT " YOU WANT TO USE POLYA.    ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 205
      PRINT : PRINT " YOU DO NOT WANT POLYA    ...TRY AGAIN."
      GOTO 1001
205   CLS : PRINT : PRINT
      PRINT " THIS PROGRAM PERMITS YOU TO FIT ONE OF"
      PRINT " THE FOLLOWING MODELS TO YOUR DATA : "
      PRINT "  (1) A POLYA "
      PRINT "  (2) A TWO COMPONENT MIXED POLYA "
      PRINT "  (3) A THREE COMPONENT MIXED POLYA "
      PRINT " WHICH DO YOU WANT TO DO ? " : PRINT
2055  INPUT " ENTER 1,2, OR 3  (TO STOP THE PROGRAM ENTER 4)   : " ,NCPOL
      PRINT
      IF NCPOL > 0 AND NCPOL < 4 THEN GOTO 211
      IF NCPOL = 4 THEN GOTO 206
      PRINT " YOU DID NOT ENTER   1,2,3, OR 4 " : PRINT
      GOTO 2055
206   PRINT
      INPUT " YOU WANT TO STOP THE PROGRAM.    ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000
      GOTO 205
211   IF NCPOL = 1 THEN GOTO 212
      PRINT " SO, YOU WANT TO APPROXIMATE THE PARAMETER " ;
      PRINT USING "OF A  #  COMPONENT MIXED POLYA."; NCPOL
```

```
      GOTO 214
212   PRINT " SO, YOU WANT TO APPROXIMATE THE PARAMETER OF A POLYA."
214   INPUT "  ( Yes or No )  : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 216
      IF NCPOL = 1 THEN GOTO 215
      PRINT USING " YOU DID NOT WANT A # COMPONENT MIXED POLYA "; NCPOL ;
      PRINT "  ...NOW TRY AGAIN."
      GOTO 2055
215   PRINT " YOU DID NOT WANT A POLYA   ...NOW TRY AGAIN."
      GOTO 2055
      '  *
      '  * SET PARAMETERS TO INDICATE MIXED POLYA & CALCULATE # OF VARIABLES.
      '  *
216   IJIJ = 2
      IIII = (NCPOL-1)*3+2
      '  *
220   IF ISAVE = 1 THEN GOTO 404
      '  *
      CLS : PRINT : PRINT
      PRINT " * ENTER DATA AS REQUIRED BY PROGRAM * " : PRINT
221   INPUT " DO YOU WANT AN EXAMPLE OF HOW TO DO THIS ?     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "N" OR ICODE$ = "n" THEN GOTO 305
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 222 ELSE GOTO 221
222   CLS : PRINT : PRINT
      PRINT " E X A M P L E " : PRINT
      PRINT " SUPPOSE THAT 45 LOTS HAVE BEEN INSPECTED BY TAKING A SAMPLE OF "
      PRINT " SIZE 16 FROM EACH LOT AND OBSERVING OF DEFECTIVE IN EACH SAMPLE."
      PRINT " IF IT IS OBSERVED THAT,OF THESE 45 SAMPLES, 18 SAMPLES HAVE NO "
      PRINT " DEFECTIVES, 21 SAMPLES HAVE 2 DEFECTIVES,AND 6 SAMPLES HAVE 11 "
      PRINT " DEFECTIVES,THEN DATA ARE ENTERED AS BELOW." : PRINT
      PRINT " NUMBER OF LOTS SAMPLED                                  : 45 "
      PRINT " SAMPLE SIZE(NUMBER OBSERVED IN EACH SAMPLE)             : 16 "
      PRINT " NUMBER OF DIFFERENT DEFECTIVE COUNTS OBSERVED           :  3 "
      PRINT " 1-NUMBER OF DEFECTIVES IN SAMPLE                        :  0 "
      PRINT " NUMBER OF SAMPLES IN WHICH THERE WERE  0 DEFECTIVES     : 18 "
      PRINT " 2-NUMBER OF DEFECTIVES IN SAMPLE                        :  2 "
      PRINT " NUMBER OF SAMPLES IN WHICH THERE WERE  2 DEFECTIVES     : 21 "
      PRINT " 3-NUMBER OF DEFECTIVES IN SAMPLE                        : 11 "
      PRINT " NUMBER OF SAMPLES IN WHICH THERE WERE 11 DEFECTIVES     : 16 "
      PRINT : PRINT
      INPUT " PRESS ENTER  TO CONTINUE " ,IWAIT$ : PRINT : PRINT
305   CLS : PRINT : PRINT
      PRINT " ENTER DATA AS REQUIRED BY PROGRAM. " : PRINT
      INPUT " NUMBER OF LOTS SAMPLED                                  :  " , NLOT
      IDEN = IIII+2
      IF NLOT > IDEN THEN GOTO 3053
      PRINT : PRINT " THE DISTRIBUTION IS NOT IDENTIFIABLE "
      PRINT USING " NUMBER OF LOTS SAMPLED SHOULD AT LEAST BE ## "; IDEN
3050  PRINT
      PRINT " WHAT DO YOU WANT TO DO ? "
      PRINT " (1) STOP "
      PRINT " (2) CHANGE NUMBER OF LOTS SAMPLED " : PRINT
      INPUT " ENTER 1 OR 2     : " ,IIRAN
      IF IIRAN <> 1 THEN GOTO 305
      INPUT " YOU WANT TO STOP THE PROGRAM.     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000 ELSE GOTO 3050
```

```
3053  IF ICSOD = 1 THEN GOTO 404
3055  INPUT " SAMPLE SIZE (NUMBER OBSERVED IN EACH SAMPLE)          :  " , NX
      IF NX >= IIII THEN GOTO 390
      PRINT : PRINT " THE DISTRIBUTION IS NOT IDENTIFIABLE "
      PRINT USING " SAMPLE SIZE SHOULD AT LEAST BE ## "; IIII
3056  PRINT
      PRINT " WHAT DO YOU WANT TO DO ? "
      PRINT " (1) STOP "
      PRINT " (2) CHANGE SAMPLE SIZE " : PRINT
      INPUT " ENTER 1 OR 2     : " ,IRAN
      IF IRAN <> 1 THEN GOTO 3055
      INPUT " YOU WANT TO STOP THE PROGRAM.     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000 ELSE GOTO 3056
390   INPUT " NUMBER OF DIFFERENT DEFECTIVE COUNTS OBSERVED         :  " ,NOBSR
      IF NOBSR > IIII AND NOBSR < 3001 THEN GOTO 391
      IF NOBSR > 3000 THEN GOTO 3901
      NOATL = IIII+1
      PRINT " THE DISTRIBUTION IS NOT IDENTIFIABLE "
      PRINT " NUMBER OF DIFFERENT DEFECTIVE COUNTS OBSERVED "
      PRINT USING " SHOULD AT LEAST BE ## "; NOATL
      PRINT
3902  PRINT " WHAT DO YOU WANT TO DO ? "
      PRINT " (1) STOP "
      PRINT " (2) CHANGE NUMBER OF DIFFERENT DEFECTIVE COUNTS OBSERVED " : PRINT
      INPUT " ENTER 1 OR 2     : " ,IRACO
      IF IRACO <> 1 THEN GOTO 390
      INPUT " YOU WANT TO STOP THE PROGRAM.     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000 ELSE GOTO 3902
3901  PRINT : PRINT " THIS CANNOT BE GREATER THAN 3000 " : PRINT : GOTO 3902
      '  *
      '  * INITIALIZE ALL OF DATA POINTS.
      '  *
391   MNX = NX+1
      FOR IJI = 1 TO MNX
        NOO(IJI) = IJI-1
        NPROB(IJI) = 0
      NEXT IJI
      '  *
      '  * READ IN DATA POINTS.
      '  *
      FOR IJI = 1 TO NOBSR
392     PRINT USING " ####-NUMBER OF DEFECTIVES IN SAMPLE                 :  "; IJI ;
        INPUT , NOO(IJI)
        IF NOO(IJI) <= NX AND NOO(IJI) => 0 THEN GOTO 396
        IF NOO(IJI) > NX THEN GOTO 393
        PRINT : PRINT " NUMBER OF DEFECTIVES SHOULD BE A POSITIVE INTEGER " ;
        PRINT "     ...TRY AGAIN. " : PRINT
        GOTO 392
393     PRINT : PRINT " NUMBER OF DEFECTIVES CANNOT EXCEED SAMPLE SIZE "
        PRINT "     ...TRY AGAIN. " : PRINT : GOTO 392
396     PRINT USING " NUMBER OF SAMPLES IN WHICH THERE WERE ### DEFECTIVES  :  "; NOO(IJI) ;
        INPUT , NPROB(IJI)
        PROB(IJI) = NPROB(IJI)/NLOT
        IF NPROB(IJI) => 0 AND NPROB(IJI) <= NLOT THEN GOTO 403
        IF NPROB(IJI) > NLOT THEN GOTO 397
        PRINT : PRINT " NEGATIVE NUMBER IS NOT CORRECT     ...TRY AGIAN. " : PRINT
```

```
      GOTO 396
397   PRINT : PRINT " NUMBER OF DEFECTIVE COUNTS OBSERVED CANNOT EXCEED NUMBER "
      PRINT " OF LOTS SAMPLED. " : PRINT : GOTO 396
403   NEXT IJI
      '  *
      '  * ADDITIONAL CHECK ON DATA ENTERED.
      '  *
      IADNE = 0
      FOR IJI = 1 TO NOBSR
        IADNE = IADNE + NPROB(IJI)
      NEXT IJI
      IF IADNE = NLOT THEN GOTO 404
4032  CLS : PRINT : PRINT
      PRINT "     INCONSISTENT ! " : PRINT
      PRINT USING " SUM OF NUMBERS OF SAMPLES = ####  WHILE "; IADNE ;
      PRINT USING " ####  WAS EXPECTED. "; NLOT
4033  PRINT : PRINT " WHICH DO YOU WANT TO DO ? "
      PRINT " (1) STOP THE PROGRAM "
      PRINT " (2) CHANGE THE NUMBER OF LOTS SAMPLED "
      PRINT " (3) CHANGE OTHER DATA "
4034  INPUT " ENTER 1,2, OR 3    : " ,ICSOD
      IF ICSOD = 2 THEN GOTO 305
      IF ICSOD = 3 THEN GOTO 4065
      IF ICSOD <> 1 THEN GOTO 4034
      INPUT " YOU WANT TO STOP THE PROGRAM     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000 ELSE GOTO 4033
      '  *
      '  * WRITE A LISTING OF DATA.
      '  *
404   ILIST = 0
      ICSOD = 0
      '  * CALCULATE FREQUENCIES AGIAN, TO TAKE CARE OF CHANGS IN
      '  * THE VALUE OF NUMBER OF LOTS SAMPLED.
      FOR IJI = 1 TO NOBSR
        PROB(IJI) = NPROB(IJI)/NLOT
      NEXT IJI
406   CLS : PRINT : PRINT
      INPUT " DO YOU WANT A LISTING OF DATA ?      ( Yes or No )  : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 4065
      IF ICODE$ = "N" OR ICODE$ = "n" THEN GOTO 4061 ELSE GOTO 406
4061  IF ILIST = 1 THEN GOTO 425 ELSE GOTO 4125
4065  CLS : PRINT : PRINT
      PRINT TAB(22);"*** DATA INPUT ***" : PRINT
      PRINT USING " LOTS SAMPLED = #### "; NLOT
      PRINT USING " SAMPLE SIZE  = #### "; NX
      PRINT
      PRINT " OBS. #    # OF DEFECTIVES       # OF SAMPLES      FREQUENCY "
      FOR IJI = 1 TO NOBSR
        PRINT USING " ####--        ####                 "; IJI,NOO(IJI);
        PRINT USING "####              #.######## "; NPROB(IJI),PROB(IJI)
      NEXT IJI
      PRINT
      '  *
      '  * CORRECTION ROUTINE.
      '  *
411   PRINT : PRINT " CHECK YOUR DATA INPUT. "
```

```
      INPUT " PRESS ENTER  WHEN YOU FINISH CHECKING. " ,IWAIT$
4125  PRINT
      INPUT " DO YOU WANT TO CHANGE ANY DATA ?     ( Yes or No )  : " ,IMIST$
      IF IMIST$ = "N" OR IMIST$ = "n" THEN GOTO 422
414   PRINT : PRINT " WHICH OBSERVATION DO YOU WANT TO CHANGE ? "
      INPUT " ENTER OBSERVATION #                                  : " ,IJI
      IF IJI <= NOBSR AND IJI => 0 THEN GOTO 419
      PRINT : PRINT USING " YOU SHOULD ENTER A BETWEEN 0 AND #### "; NOBSR
      GOTO 414
419   PRINT USING " ####-NUMBER OF DEFECTIVES IN SAMPLE                 : "; IJI ;
      INPUT , NOO(IJI)
      IF NOO(IJI) <= NX AND NOO(IJI) => 0 THEN GOTO 421
      IF NOO(IJI) > NX THEN GOTO 420
      PRINT : PRINT "  NUMBER OF DEFECTIVES SHOULD BE A POSITIVE INTEGER "
      PRINT "      ...TRY AGAIN. " : PRINT
      GOTO 419
420   PRINT : PRINT " NUMBER OF DEFECTIVES CANNOT EXCEED SAMPLE SIZE "
      PRINT "      ...TRY AGAIN. " : PRINT
      GOTO 419
421   PRINT " NUMBER OF SAMPLES IN WHICH ";
      PRINT USING "THERE WERE ### DEFECTIVES : "; NOO(IJI) ;
      INPUT , NPROB(IJI)
      IF NPROB(IJI) => 0 AND NPROB(IJI) < NLOT THEN GOTO 404
      IF NPROB(IJI) => NLOT THEN GOTO 4215
      PRINT : PRINT " NEGATIVE NUMBER IS NOT CORRECT    ...TRY AGIAN. " : PRINT
      GOTO 421
4215  PRINT : PRINT " NUMBER OF DEFECTIVE COUNTS OBSERVED CANNOT EXCEED NUMBER " ;
      PRINT " OF LOTS SAMPLED. " : PRINT
      GOTO 421
      '  *
      '  * REPEAT CORRECTION ROUTINE.
      '  *
422   CLS : PRINT : PRINT
      INPUT " SO, ALL OF DATA ARE NOW CORRECT.     ( Yes or No )  : " ,ICORC$
      IF ICORC$ = "N" OR ICORC$ = "n" THEN ILIST = 1 : GOTO 4125
      '  * ILIST IS ONE...
      '  * ADDITIONAL CHECK.
      '  *
      IADNE = 0
      FOR IJI = 1 TO NOBSR
        IADNE = IADNE + NPROB(IJI)
      NEXT IJI
      IF IADNE <> NLOT THEN GOTO 4032
      '  *
      '  * NOW, CALCULATE THE FINAL VALUES OF FREQUENCIES.
      '  *
      FOR IJI = 1 TO NOBSR
        PROB(IJI) = NPROB(IJI)/NLOT
      NEXT IJI
      '  *
      '  * SET CIMENSIONALITY OF OBJECTIVE FUNCTION.
      '  *
425   N = IIII
      '  *
      '  * SET MIN. AND MAX. OF VARIBLES.
      '  *
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      A = .00002
      B = .99997
      '  *
      '  * WRITE A HEADING.
      '  *
4257  CLS : PRINT
      PRINT " PARAMETERS OF YOUR SELECTED DISTRIBUTION ARE : " : PRINT
      '  *
      '  * NEVAL IS USED AS A COUNTER TO BE COMPARED WITH LIM IN SEARCH ROUTINE.
      '  *
      NEVAL = 0
      IF NSTS > 6 THEN GOTO 4541
      IF IJIJ = 2 THEN GOTO 437
      '  *
      '  ****************************************
      '  * M.BINOMIAL * SUGGESTED STARTING POINT. *
      '  ****************************************
      '  * IF ALL OF THE SUGGESTED STARTING POINT HAVE ALREADY BEEN USED,
      '  * THEN ASK USER TO SPECIFY THE STARTING POINT.
      '  *
      FOR JI = 1 TO N
        XMIN(JI) = A
        XMAX(JI) = B
      NEXT JI
      '  *
      IF NCOMP = 1 THEN GOTO 430          '* TWO COMPONENT M.BINOMIAL.
      IF NCOMP = 2 THEN GOTO 432          '* THREE COMPONENT M.BINOMIAL.
      IF NCOMP = 3 THEN GOTO 434          '* FOUR COMPONENT M.BINOMIAL.
      '  *
      '  * TWO COMPONENT M.BINOMIAL.
      '  *
430   FOR JI = 1 TO N
        X(JI) = ATWOB(JI,NSTS)
      NEXT JI
      PRINT "     W(1),F(1), AND F(2) " : PRINT
      PRINT " SUGGESTED STRATING POINT IS : "
      PRINT USING " W(1) = ##.###### "; X(1)
      PRINT USING " F(1) = ##.###### "; X(2)
      PRINT USING " F(2) = ##.###### "; X(3) : PRINT
      LIM = 3000
      TOL = 1E-3
      GOTO 444
      '  *
      '  * THREE COMPONENT M.BINOMIAL.
      '  *
432   FOR JI = 1 TO N
        X(JI) = ATREB(JI,NSTS)
      NEXT JI
      PRINT "     W(1),F(1),W(2),F(2), AND F(3) " : PRINT
      PRINT " SUGGESTED STRATING POINT IS : "
      PRINT USING " W(1) = ##.###### "; X(1)
      PRINT USING " F(1) = ##.###### "; X(3)
      PRINT USING " W(2) = ##.###### "; X(2)
      PRINT USING " F(2) = ##.###### "; X(4)
      PRINT USING " F(3) = ##.###### "; X(5) : PRINT
      LIM = 1200+INCRMN
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      IF NX <= 50 THEN TOL = 1E-3
      IF NX >  50 THEN TOL = 1E-2
      GOTO 444
      '  *
      '  * FOUR COMPONENT M.BINOMIAL.
      '  *
434   FOR JI = 1 TO N
        X(JI) = AFORB(JI,NSTS)
      NEXT JI
      PRINT "     W(1),F(1),W(2),F(2),W(3),F(3), AND F(4) " : PRINT
      PRINT " SUGGESTED STRATING POINT IS : "
      PRINT USING " W(1) = ##.###### "; X(1)
      PRINT USING " F(1) = ##.###### "; X(4)
      PRINT USING " W(2) = ##.###### "; X(2)
      PRINT USING " F(2) = ##.###### "; X(5)
      PRINT USING " W(3) = ##.###### "; X(3)
      PRINT USING " F(3) = ##.###### "; X(6)
      PRINT USING " F(4) = ##.###### "; X(7) : PRINT
      LIM = 900+INCRMN
      IF NX <= 40 THEN TOL = 1E-3
      IF NX > 40 THEN TOL = 1E-2
      GOTO 444
      '  ************************************
      '  * POLYA * SUGGESTED STARTING POINT. *
      '  ************************************
437   FOR JI = 1 TO N
        XMIN(JI) = A
        XMAX(JI) = 72
      NEXT JI
      '  *
    IF NCPOL = 1 THEN GOTO 439          '* SIMPLE POLYA.
    IF NCPOL = 2 THEN GOTO 440          '* TWO COMPONENT MIXED POLYA.
    IF NCPOL = 3 THEN GOTO 442          '* THREE COMPONENT MIXED POLYA.
      '  *
      '  * SIMPLE POLYA.
      '  *
439   FOR JI = 1 TO N
        X(JI) = AONEP(JI,NSTS)
      NEXT JI
      PRINT "   S AND T " : PRINT
      PRINT " SUGGESTED STRATING POINT IS : "
      PRINT USING " S = ##.###### "; X(1)
      PRINT USING " T = ##.###### "; X(2) : PRINT
      LIM = 3000
      TOL = 1E-3
      GOTO 444
      '  *
      '  * TWO COMPONENT MIXED POLYA.
      '  *
440   XMAX(1) = B
      '  *
      FOR JI = 1 TO N
        X(JI) = ATWOP(JI,NSTS)
      NEXT JI
      PRINT "     W(1),S(1),T(1),S(2), AND T(2) " : PRINT
      PRINT " SUGGESTED STRATING POINT IS : "
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      PRINT USING " W(1) = ##.###### "; X(1)
      PRINT USING " S(1) = ##.###### "; X(2)
      PRINT USING " T(1) = ##.###### "; X(3)
      PRINT USING " S(2) = ##.###### "; X(4)
      PRINT USING " T(2) = ##.###### "; X(5) : PRINT
      LIM = 1200+INCRMN
      IF NX <= 20 THEN TOL = 1E-3
      IF NX >  20 THEN TOL = 1E-2
      GOTO 444
      '  *
      '  * THREE COMPONENT MIXED POLYA.
      '  *
442   XMAX(1) = B
      XMAX(2) = B
      '  *
      FOR JI = 1 TO N
        X(JI) = ATREP(JI,NSTS)
      NEXT JI
      PRINT "     W(1),S(1),T(1),W(2),S(2),T(2),S(3), AND T(3) " : PRINT
      PRINT " SUGGESTED STRATING POINT IS : "
      PRINT USING " W(1) = ##.###### "; X(1)
      PRINT USING " S(1) = ##.###### "; X(3)
      PRINT USING " T(1) = ##.###### "; X(4)
      PRINT USING " W(2) = ##.###### "; X(2)
      PRINT USING " S(2) = ##.###### "; X(5)
      PRINT USING " T(2) = ##.###### "; X(6)
      PRINT USING " S(3) = ##.###### "; X(7)
      PRINT USING " T(3) = ##.###### "; X(8) : PRINT
      LIM = 800+INCRMN
      TOL = 1E-2
      '  *
444   INPUT " YOU ACCEPT THIS STARTING POINT.    ( Yes or No )  : " ,IACC$
      IF IACC$ = "Y" OR IACC$ = "y" THEN GOTO 461
      IF IACC$ = "N" OR IACC$ = "n" THEN GOTO 4441 ELSE GOTO 444
4441  IF NSTS =  6 THEN GOTO 4541
4442  CLS : PRINT : PRINT : PRINT " WHICH DO YOU WANT TO DO ? "
      PRINT " (1) SPECIFY STARTING POINT  "
      PRINT " (2) TRY OUR NEXT SUGGESTED STARTING POINT  "
      INPUT " ENTER 1 OR 2    : " ,ISUGG
      IF ISUGG =  1 THEN GOTO 4541
      IF ISUGG =  2 THEN NSTS = NSTS+1 : GOTO 4257
      GOTO 4442
      '  *
4541  CLS : PRINT : PRINT : PRINT " SPECIFY STARTING POINT. "
      '  ********************************
      '  * USER SUGGESTS STARTING POINT *
      '  ********************************
      '  **** LOOP START ****
455   FOR JI = 1 TO N
      '  *
      '  * KK IS SUBSCRIPT OF VARIABLE X
      '  *
        IF IJIJ = 2 THEN GOTO 456
      '  ***********************************************
      '  * M.BINOMIAL * USER SUGGESTS STARTING POINT *
      '  ***********************************************
```

```
      IF NCOMP = 1 THEN GOTO 4551      '* TWO COMPONENT M.BINOMIAL.
      IF NCOMP = 2 THEN GOTO 4552      '* THREE COMPONENT M.BINOMIAL.
      IF NCOMP = 3 THEN GOTO 4553      '* FOUR COMPONENT M.BINOMIAL.
    '  *
    '  * TWO COMPONENT M.BINOMIAL.
    '  *
4551  IF JI = 1 THEN CHAR$ = "W(1)" : KK = 1
      IF JI = 2 THEN CHAR$ = "F(1)" : KK = 2
      IF JI = 3 THEN CHAR$ = "F(2)" : KK = 3
      GOTO 4578
    '  *
    '  * THREE COMPONENT M.BINOMIAL.
    '  *
4552  IF JI = 1 THEN CHAR$ = "W(1)" : KK = 1
      IF JI = 2 THEN CHAR$ = "F(1)" : KK = 3
      IF JI = 3 THEN CHAR$ = "W(2)" : KK = 2
      IF JI = 4 THEN CHAR$ = "F(2)" : KK = 4
      IF JI = 5 THEN CHAR$ = "F(3)" : KK = 5
      GOTO 4578
    '  *
    '  * FOUR COMPONENT M.BINOMIAL.
    '  *
4553  IF JI = 1 THEN CHAR$ = "W(1)" : KK = 1
      IF JI = 2 THEN CHAR$ = "F(1)" : KK = 4
      IF JI = 3 THEN CHAR$ = "W(2)" : KK = 2
      IF JI = 4 THEN CHAR$ = "F(2)" : KK = 5
      IF JI = 5 THEN CHAR$ = "W(3)" : KK = 3
      IF JI = 6 THEN CHAR$ = "F(3)" : KK = 6
      IF JI = 7 THEN CHAR$ = "F(4)" : KK = 7
      GOTO 4578
    '  ****************************************
    '  * POLYA * USER SUGGESTS STARTING POINT. *
    '  ****************************************
456   IF NCPOL = 1 THEN GOTO 4561     '* SIMPLE POLYA.
      IF NCPOL = 2 THEN GOTO 4562     '* TWO COMPONENT MIXED POLYA.
      IF NCPOL = 3 THEN GOTO 4563     '* THREE COMPONENT MIXED POLYA.
    '  *
    '  * SIMPLE POLYA.
    '  *
4561  IF JI = 1 THEN CHAR$ = "S   " : KK = 1
      IF JI = 2 THEN CHAR$ = "T   " : KK = 2
    GOTO 4578
    '  *
    '  * TWO COMPONENT M.POLYA.
    '  *
4562  IF JI = 1 THEN CHAR$ = "W(1)" : KK = 1
      IF JI = 2 THEN CHAR$ = "S(1)" : KK = 2
      IF JI = 3 THEN CHAR$ = "T(1)" : KK = 3
      IF JI = 4 THEN CHAR$ = "S(2)" : KK = 4
      IF JI = 5 THEN CHAR$ = "T(2)" : KK = 5
      GOTO 4578
    '  *
    '  * THREE COMPONENT M.POLYA.
    '  *
4563  IF JI = 1 THEN CHAR$ = "W(1)" : KK = 1
      IF JI = 2 THEN CHAR$ = "S(1)" : KK = 3
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      IF JI = 3 THEN CHAR$ = "T(1)" : KK = 4
      IF JI = 4 THEN CHAR$ = "W(2)" : KK = 2
      IF JI = 5 THEN CHAR$ = "S(2)" : KK = 5
      IF JI = 6 THEN CHAR$ = "T(2)" : KK = 6
      IF JI = 7 THEN CHAR$ = "S(3)" : KK = 7
      IF JI = 8 THEN CHAR$ = "T(3)" : KK = 8
    '  *
4578  PRINT USING " \  \    =  "; CHAR$ ;
    '  * READ VALUE OF THE PARAMETER.
      INPUT ,X(KK)
    '  *
    '  * CHECK THE STRATING POINT BEING ENTERED BY USER.
    '  *
      IF X(KK) < XMAX(KK) AND X(KK) => A THEN GOTO 460
      PRINT : PRINT USING " THIS SHOULD BE A NUMBER BETWEEN ##.###### "; A ;
      PRINT USING "AND ##.###### ";XMAX(KK)
      PRINT "   ...TRY AGIAN. " :PRINT
      GOTO 4578
460   NEXT JI
      GOTO 444
      '  **** END LOOP ****
461   CLS : PRINT : PRINT
      PRINT " LIM : UPPER BOUND ON THE NUMBER OF SEARCHES. "
      PRINT " TOL : LOWER BOUND ON STEP SIZE FOR EXPLORATORY MOVES. " : PRINT
      PRINT USING " LIM = #### AND TOL = #.##### "; LIM,TOL : PRINT
      INPUT " YOU ACCEPT THIS.    ( Yes or No )  : " ,ILIT$
      IF ILIT$ = "Y" OR ILIT$ = "y" THEN GOTO 481
      IF ILIT$ = "N" OR ILIT$ = "n" THEN GOTO 471 ELSE GOTO 461
471   CLS : PRINT : PRINT
      PRINT " ENTER YOUR DESIRED VALUES." : PRINT
      INPUT " LIM  : " ,LIM
      IF LIM => 800 AND LIM <= 3000 THEN GOTO 475
      PRINT : PRINT " LIM SHOULD BE A NUMBER BETWEEN 800 TO 3000 " : PRINT
      GOTO 471
475   INPUT " TOL  : " ,TOL
      IF TOL < .001 OR LIM > .1 THEN GOTO 461
      PRINT : PRINT " TOL SHOULD BE A NUMBER BETWEEN .001 TO .1 " : PRINT
      GOTO 475
      '  *
481   CLS
      LOCATE 8,1
      PRINT TAB(15) "IMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) ":                                               :"
      PRINT TAB(15) "HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM<"
      COLOR 21
      LOCATE 12,25 : PRINT "   Please Wait !!!!!!!!   "
      COLOR 7
      GOSUB SUBPATS    '**************Goto Subroutine PATS
      GOSUB SUBPRINT   '**************Goto Subroutine PRINT
      '  *
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      ISAVE = 0
      '  *
      '  * ASK THE INTENT OF USER.
      '  *
500   PRINT : PRINT " DO YOU WANT TO FIND A BETTER FIT EMPLOYING "
      INPUT " THE SAME MODEL ?     ( Yes or No )  : " ,IBETE$
      '  * IF A BETER FIT IS NOT DESIERD, INITIALIZE THE STARTING VALUES.
      IF IBETE$ = "N" OR IBETE$ = "n" THEN NSTS = 1 : GOTO 520
      IF IBETE$ = "Y" OR IBETE$ = "y" THEN GOTO 501  ELSE GOTO 500
501   CLS : PRINT : PRINT
      PRINT " TO GET A BETTER FIT DO SOME OR ALL OF THE FOLLOWING : "
      PRINT " (1) INCREASE LIM UP TO 3000, IF IT IS NOT THERE ALREADY. "
      PRINT " (2) TRY OTHER STARTING POINTS. "
      PRINT " (3) DECREASE TOL TO .001, IF IT IS NOT THERE ALREADY. "
      IF LIM < 2900 THEN GOTO 508
      PRINT
      PRINT USING " LIM IS ALREADY #### SO TRY A NEW STARTING POINT "; LIM
      '  *
      '  * PRINT THE NEXT STARTING POINT.
      '  *
      NSTS = NSTS+1
      GOTO 4257
      '  *
      '  * INCREASE THE INCREMENTOF LIM
      '  *
508   INCRMN = INCRMN+100
      PRINT : PRINT " WHICH DO YOU WANT TO USE ? "
      PRINT " (1) THE PREVIOUS STARTING POINT. "
      PRINT " (2) A NEW STARTING POINT. "
509   PRINT : INPUT " ENTER 1, OR 2     : " ,IPONS
      IF IPONS = 1 THEN GOTO 461
      IF IPONS = 2 THEN NSTS = NSTS+1 : CLS : GOTO 4257
      GOTO 509
      '  *
      '  * ASK THE INTENT OF THE USER.
      '  *
520   CLS : PRINT : PRINT
      PRINT " WHICH DO YOU WANT TO USE ? "
      PRINT " (1) STOP THE PROGRAM. "
      PRINT " (2) SAVE THE DATA AND FIT ANOTHER MODEL TO IT. "
      PRINT " (3) FIT A MODEL TO OTHER DATA. "
522   PRINT : INPUT " ENTER 1,2, OR 3     : " ,ISTOP
      IF ISTOP  = 2 THEN ISAVE = 1 : GOTO 1
      IF ISTOP  = 3 THEN GOTO 111
      IF ISTOP <> 1 THEN GOTO 520
      INPUT " YOU WANT TO STOP THE PROGRAM.     ( Yes or No ) : " ,ICODE$
      IF ICODE$ = "Y" OR ICODE$ = "y" THEN GOTO 1000
      GOTO 522
      '  *
1000  STOP
      END
      '  *
SUBPATS:
      '  *            ******************************
      '  *            * PATTERN SEARCH SUBROUTINE *
      '  *            ******************************
```

```
'  ************************************************************************
'  * THIS ROUTINE IS BASED LARGELY ON PRIOR WORK BY TAUBERT WHO, IN TURN,  *
'  * RELIED HEAVILY ON A SEARCH ROUTINE BY WEISMAN-WOOD-RIVLIN, WHICH      *
'  * WAS ITSELF AN ADAPTATION OF THE ORIGINAL HOOKE-JEEVES PATTERN SEARCH. *
'  * THE CODE IS DIVIDED INTO 18 SUBSETS CALLED 'BOXES',WITH ENTERANCES    *
'  * AND EXITS FOR EACH BOX EXHAUSTIVELY IDENTIFIED. THIS FORMS A COMPLETE *
'  * BASIS FOR A FLOW CHART, IF THE NEED SHOULD ARISE.                     *
'  ************************************************************************
'  *
'  *     BOX 1
'  *
'  * PURPOSE: INITIALIZE DATA.
'  * ENTRANCES: FROM START OF THE SUBROUTINE AND FROM BOX 18 VIA STATEMENT 6100.
'  * EXITS: TO 6190 IN BOX 5.
'  *
6100  BET = .1
      DEL = .1
      LT = 0
      LSN = 0
      NPF = 0
      ITR = 1
      ALP = 2
      GR = 2.2
      LA = 1
      LT2 = 0
      LT3 = 0
      LT4 = 0
      LT5 = 0
      LT6 = 0
      LT7 = 0
      ID1 = 0
      ID2 = 0
      ID3 = 0
      ID4 = 0
      KOUNT = 0
      V = 0
      OLDV = 0
      NEVOLD = 0
      FOR I = 1 TO N
        Q(I) = X(I)
'  *
'  * NOTE... INITIAL VALUE OF D(K) DEPENDS ON HOW FAR APART THE UPPER
'  * AND LOWER BOUNDRIES ARE.
'  *
        D(I) = DEL*(XMAX(I)-XMIN(I))
      NEXT I
'  *
      GOSUB SUBFCT1
'  *
      SNOLD = SN
      OLDSN = SN
      SC = SN
      SP = SN
      M1 = 1
      M2 = 1
      K = 1
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      KK = 1
      L4 = 1
      GOTO 6190
      '  *
      '  *     BOX 18
      '  *
      '  * PURPOSE: CALL OBJECTIVE FUNCTION.
      '  * ENTRANCES : FROM BOXS 17, 7,AND 14.
      '  * EXITS: TO STMTS 6100 AND 6500, TO 6282 AND 6285 IN BOX 2, TO 6463
      '  * AND 6466 IN BOX 11, AND TO 6580 IN BOX 15, TO 6510 IN BOX 13.
      '  *
6270  GOSUB SUBFCT1
      ON (LA) GOTO 6100,6282,6463,6580,6285,6466,6510,6500
      '  *
      '  *     BOX 2
      '  *
      '  * PURPOSE: EVALUATE FORWARD MOVES.
      '  * ENTRANCES: FROM BOX 18 VIA STMTS 6282, 6285.
      '  * EXITS: TO 6360 IN BOX 6 WHEN FORWARD MOVE FAILS, TO 6300 IN BOX 3
      '  * WHEN MOVE SUCCESS.
      '  * COMMENT: THIS BOX IS REACHED WHEN LA = 2 OR 5.
      '  *
      '  * INCREMENT STATE COUNTER
6282  LT2 = LT2+1
      GOTO 6280
6285  LT5 = LT5+1
6280  IF SN >= SP THEN GOTO 6360
      '  * TEST FOR SUCCESS OR FAILURE(6290, OR 6360)
6290  ON (L4) GOTO 6300,6292
      '  *
      '  * IF THE SEARCH IS IN AN EXPLORATORY MOVE FOLLOWING AN ATTEMPTED
      '  * PATERN MOVE, MULTIPLY THIS SUCCESSFULL FORWARD STEP SIZE BY ALP.
      '  *
6292  D(K) = D(K)*ALP
      '  *
      '  *     BOX 3
      '  *
      '  * PURPOSE: RESET SOME VARIABLES.
      '  * ENTRANCES: FROM BOXES 2 AND 11, BOTH VIA STM 6300.
      '  * EXIT: TO 6305 IN BOX 4.
6300  SP = SN
      '  *
      M2 = 1
      M1 = 1
      MPF = 0
      '  *
      '  *     BOX 4
      '  *
      '  * PURPOSE: DECIDE WHETHER TO TEST FOR A NEW BASE POINT.
      '  * ENTRANCES: FROM BOXES 3 AND 12 VIA STM 6305.
      '  * EXITS: TO 6340 IN BOX 7 WHEN A BASE POINT TEST IS REQUIRED,
      '  * TO 6200 IN BOX 5 OTHERWISE.
      '  * COMMENT: THIS BOX IS THE ONLY ENTRANCE TO THE BASE POINT TEST.
      '  * ALSO NOTE THAT WHEN LT = 0, AS IT NORMALY IS, IT IS ONLY HERE
      '  * THAT KK IS INCREASED AND TESTED.
      '  *
```

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

```
6305  K = K+1
      IF K <= N THEN GOTO 6320
6777  K = 1
6320  IF LT > 0 THEN GOTO 6340
6330  KK = KK+1
      IF KK > N THEN GOTO 6340
      '  *
      '  *     BOX 5
      '  *
      '  * PURPOSE: SET SOME INDICES AND INCREMENT THE VALUE OF X(K) IN
      '  * PREPARATION FOR TESTING A FORWARD MOVE IN THE KTH VARIABLE.
      '  * ENTRANCES: FROM BOX 1 VIA STM 6190, FROM BOXES 4 AND 8 VIA STM
      '  * 6200, FROM BOX 10 VIA STM 6190, FROM BOX 15 VIA STM 6210.
      '  * EXITS: TO 6490 IN BOX 12 WHEN VARIABLE IS FIXED AND CANNOT BE
      '  * IS ADJUSTED HERE TO TRY A FORWARD MOVE.
      '  *
6200  ON (L4) GOTO 6190,6210
6190  L4 = 1
      LA = 2
      GOTO 6220
6210  LA = 5
6220  IF D(K) = 0 THEN GOTO 6490
6225  X(K) = X(K)+D(K)
      GOTO 6230
      '  *
      '  *     BOX 6
      '  *
      '  * PURPOSE: WHEN FORWARD MOVE FAILES THIS BOX IS REACHED AND THE VALUE
      '  * OF X(K) IS SET IN PREPARATION FOR ATEMPTING A REVERSE MOVE.
      '  * ENTRANCES: FROM BOXES 2 AND 17, BOTH VIA STM 6360.
      '  * EXIT: TO 6230 IN BOX 17 (BOUNDRY CHECK).
      '  *
6360  ON (L4) GOTO 6370,6380
6370  LA = 3
      GOTO 6390
6380  LA = 6
6390  X(K) = X(K)-2*D(K)
      GOTO 6230
      '  *
      '  *     BOX 17
      '  *
      '  * PURPOSE: BOUNDRY CHECK IN PREPARATION FOR POSITIVE OR NEGATIVE
      '  * MOVE OF X(K).
      '  * ENTRANCE: FROM BOXES 5 AND 6 VIA STM 6230.
      '  * EXITS: TO 6270 IN BOX 18 WHEN BOUNDRY CHECK IS PASSED,TO 6500 WHEN
      '  * FAILD, TO 6360 IN BOX 6 WHEN POSITIVE MOVE FAILES THE BOUNDRY CHECK,
      '  * TO 6480 IN BOX 2 WHEN NEGATIVE MOVE FAILES THE BOUNDRY CHECK.
      '  * COMMENT: 6270 IMPLIES BOUNDRY WAS NOT VIOLATED.
      '  *
6230  IF X(K) = XMAX(K) THEN GOTO 6270
      IF X(K) > XMAX(K) THEN GOTO 6260
6250  IF XMIN(K) <= X(K) THEN GOTO 6270
6260  ON (LA) GOTO 6500,6360,6480,6500,6360,6480,6500,6500
      '  *
      '  *     BOX 7
      '  *
```

```
      '  * PURPOSE: TEST TO SEE WHETHER A NEW BASE POINT HAS BEEN ESTABLISHED.
      '  * ENTRANCE: FROM BOX 4 VIA STM 6340.
      '  * EXITS: TO 6400 IN BOX 8 WHEN TEST FOR A NEW BASE POINT FAILS,
      '  * TO 6270 IN BOX 18 WHEN NEW BASE POINT IS ESTABLISHED.
      '  *
6340  IF SP+0.0001-SC >= 0 THEN GOTO 6400
      '  *
      '  * NOTE THAT 6350 MEANS THAT A SUCCESSFULLY NEW BASE POINT, THIS BOX IS
      '  * IDENTIFIED. IF L4 = 1, THE FIRST PATTERN MOVE IN THIS SERIES
      '  * IS TO BE ATTEMPTED.
      '  *
6350  ON (L4) GOTO 6352,6353
6352  LA = 7
      M1 = 1
      GOTO 6270
6353  IF NPF < 5 THEN GOTO 6352
      '  *
      '  *     BOX 8
      '  *
      '  * PURPOSE: HAVING FAILED TO ESTABLISH A NEW BASE POINT, THIS BOX IS
      '  * REACHED TO DECIDE WHERE TO GO NEXT.
      '  * ENTRANCE: FROM BOX 7 VIA STM 6400.
      '  * EXITS: TO 6200 IN BOX 5 WHEN ( KK <= N ), AFTER BASE POINT TEST, WHEN
      '  * FAILURE OCCURED AT THE CONCLUSION OF ADJUSTMENTS FOLLOWING AN ATTEMPTED
      '  * PATTERN MOVE, AND TO 6440 IN BOX 10 WHEN IN THE PROCESS OF SEARCHING
      '  * FOR A PATTERN THE VALUE OF M1 IS LESS THAN N+1. TO 6500 IN BOX 16 WHEN
      '  * NO NEW PATTERN IS FOUND AND SEARCH IS BE ESTABLISHED.
      '  *
6400  IF LT <= 0 THEN GOTO 6404
6778  KK = KK+1
6404  IF KK <= N THEN GOTO 6200
6779  ON (L4) GOTO 6429,6410
6429  L4 = 2
      IF M1 > N THEN GOTO 6500 ELSE GOTO 6440
      '  *
      '  *     BOX 9
      '  *
      '  * PURPOSE: RESTORE THE VALUES OF X(1) AFTER UNSUCCESSFUL ATTEMPT TO
      '  * ESTABLISH NEW BASE POINT.
      '  * ENTRANCE: FROM BOX 8 VIA STM 6410.
      '  * EXIT: TO 6440 IN BOX 10.
      '  *
6410  SP = SC
      FOR I = 1 TO N
        X(I) = Q(I)
      NEXT I
      NPF = 0
      '  *
      '  *     BOX 10
      '  *
      '  * PURPOSE; RESET SOME COUNTERS (SEE GLOSSERY FOR DEFINITIONS)
      '  * ENTRANCE: FROM BOX 8 AND 9, BOTH VIA STM 6440.
      '  * EXITS: TO 6190 IN BOX 5.
      '  *
6440  M2 = 1
      M1 = 1
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      KK = 1
      GOTO 6190
      '  *
      '  *     BOX 11
      '  *
      '  * PURPOSE: EVALUATE A REVERSE MOVE (CORRESPONDING TO BOX 2'S FORWARD
      '  * MOVE EVALUATION).
      '  * ENTRANCE: FROM BOX 18 VIA STMS 6463 AND 6466.
      '  * EXITS: TO 6480 IN BOX 12 WHEN REVERSE MOVE FAILES, TO 6300 IN BOX 3
      '  * WHEN REVERSE MOVE SUCCESS.
      '  * COMMENT: NOTE THAT AS USUAL THE COUNTERS LT3 AND LT6 ARE SET
      '  * DEPENDING WHETHER THIS BOX IS RESCHED WITH LA = 3 OR 6.
      '  *
6463  LT3 = LT3+1
      GOTO 6460
6466  LT6 = LT6+1
6460  IF SN >= SP THEN GOTO 6480
      '  *
      '  * SUCCESS IN FUTURE A FORWARD MOVE WILL BE WHAT HAS PERETOFORE BEEN
      '  * A BACKWORD MOVE.
      '  *
6470  D(K) = -D(K)
      GOTO 6300
      '  *
      '  *     BOX 12
      '  *
      '  * PURPOSE: AFTER REVERSE MOVE FAILURE THIS BOX REDUCES STEP SIZE AND
      '  * ATTEMPTS TO TRY AN EXPLORATORY STEP FOR ANOTHER VARIABLE.
      '  * ENTRANCES: FROM BOX 5 VIA STM 6490, FROM BOXES 17 AND 11 VIA STM 6480.
      '  * EXIT: TO 6305 IN BOX 4.
      '  *
6480  X(K) = X(K)+D(K)
      D(K) = D(K)*BET
      DX = ABS(X(K)/D(K)*TOL)
      IF DX = 1 THEN GOTO 6482
      IF DX < 1 THEN GOTO 6484
6481  D(K) = D(K)*DX
      '  *
6482  DX = ABS(1D-30/D(K))
      IF DX >= 1 THEN GOTO 6490
6483  D(K) = D(K)*DX
      GOTO 6490
6484  DX = ABS(1D-30/D(K))
      IF DX = 1 THEN GOTO 6490
      IF DX < 1 THEN GOTO 6492
6485  D(K) = D(K)*DX
6490  M1 = M1+1
6492  ON (L4) GOTO 6495,6493
6495  GOTO 6309
6493  M2 = M2+1
      IF M2 <= N THEN GOTO 6309
6780  M2 = 1
      NPF = NPF+1
6309  GOTO 6305
      '  *
      '  *     BOX 13
```

```
      '  *
      '  * PURPOSE: ADAPTIVE LOGIC TO COMPUTE GR, A MULTIPLICATIVE FACTOR
      '  * THAT GOVERNS THE SIZE OF THE PATTERN MOVE ATTEMPT.
      '  * ENTRANCE: FROM BOX 18 VIA STM 6510.
      '  * EXIT: TO 6888 IN BOX 14.
      '  *
6510  LT7 = LT7+1
      KOUNT = KOUNT+1
      IF (KOUNT MOD 1) <> 0 THEN GOTO 6888
6398  V = (OLDSN - SN) * 100 / (OLDSN * (NEVAL - NEVOLD))
      IF V > 0.6 THEN GOTO 6801
6781  IF V < 0.3 THEN GOTO 6802
6782  IF OLDV < V THEN GOTO 6803 ELSE GOTO 6804
6801  ID4 = ID4+1
      GOTO 6815
6802  ID1 = ID1+1
      D1 = 0.1
      GOTO 6810
6803  ID2 = ID2+1
      GOTO 6815
6804  ID3 = ID3+1
      D1 = 0.5
6810  GR = GR+D1
      IF GR < 3.5 THEN GOTO 6815
6783  GR = 2.2
6815  OLDSN = SN
      OLDV = V
      NEVOLD = NEVAL
      '  *
      '  *     BOX 14
      '  *
      '  * PURPOSE: SET X(1) TO MAKE PATTERN MOVE ATTEMPT.
      '  * ENTRANCE: FROM BOX 13 VIA STM 6888.
      '  * EXITS: TO 6270 IN BOX 18 (ALMOST ALWAYS), TO 6500 IN BOX 16
      '  * WHEN NEVAL VIOLATES LIMIT, CAUSING A RETURN TO THE MAIN ROUTINE.
      '  *
6888  IF (KOUNT MOD ITR) <> 0 THEN GOTO 6700
6700  IF LIM < NEVAL THEN GOTO 6500
6784  KK = 1
      IF LSN <= 0 THEN GOTO 6530
6785  SP = SN
6530  SC = SP
      LA = 4
      FOR I = 1 TO N
        P = Q(I)
        Q(I) = X(I)
        X(I) = P+GR*(X(I)-P)
        IF X(I) < XMAX(I) THEN GOTO 6550
        IF X(I) = XMAX(I) THEN GOTO 6570
6786    X(I) = XMAX(I)
        GOTO 6570
6550    IF XMIN(I) <= X(I) THEN GOTO 6570
6787    X(I) = XMIN(I)
6570  NEXT I
      GOTO 6270
      '  *
```

```
      '  *     BOX 15
      '  *
      '  * PURPOSE: SET CERTAIN COUNTERS PRIOR TO ADJUSTMENTS OF THE PATTERN
      '  * MOVE ATTEMPT.
      '  * ENTRANCE: FROM BOXES 18 VIA STM 6580.
      '  * EXIT: TO 6210 IN BOX 5 ALWAYS.
      '  *
6580  LT4 = LT4+1
      L4 = 2
      SP = SN
      GOTO 6210
      '  *
      '  *     BOX 16
      '  *
      '  * PURPOSE: PRINT OUT FINAL PARAMETERS VALUES AND TERMINATE THE SEARCH.
      '  * ENTRANCES: FROM BOXES 17 AND 18, FROM BOXES 8 AND 14, ALL VIA STM 6500.
      '  * EXIT: RETURN TO THE MAIN ROUTINE.
      '  *
6500  LA = 8
6990  RETURN
      '  *
SUBFCT1 :
      '  *                ***********************************
      '  *                * OBJECTIVE FUNCTION SUBROUTINE *
      '  *                ***********************************
      '  ********************************************************************************
      '  *                                                                              *
      '  * THIS SUBROUTINE CALCULATES THE SUM OF SQUARE OF DIFFERENCES BETWEEN          *
      '  * THE SELECTED PRIOR DISTRIBUTION AND THE DATA POINTS INPUTED BY THE           *
      '  * USER FOR ANY GIVEN VALUES OF THE PARAMETERS OF THE PRIOR DISTRIBUTION.       *
      '  * THESE VALUES OF THE PARAMETERS OF THE PRIOR DISTRIBUTION ARE SUGGESTED       *
      '  * BY THE PATTERN SEARCH SUBROUTINE. THIS SUBROUTINE MAY BE CALLED              *
      '  * HUNDEREDS OF TIMES BY THE PATTERN SEARCH SUBROUTINE AFTER THE PATTERN        *
      '  * SEARCH SUBROUTINE WAS CALLED BY THE MAIN PROGRAM. THE LIMIT ON THE           *
      '  * NUMBER OF CALLS IS LIM.                                                      *
      '  *                                                                              *
      '  ********************************************************************************
      '  *
       DEF FNDLGAMA(XXX)
       ZZ = XXX
       IF XXX > 1D+10 THEN GOTO 8008
8002   IF XXX > 1D-9 THEN GOTO 8004
       FNDLGAMA = -1D+75
       GOTO 8010
8004   ZTERM = 1D+0
       DO
         ZTERM = ZTERM*ZZ
         ZZ = ZZ+1D+0
       LOOP UNTIL ZZ > 18D+0
       ZRZZ = 1D+0/ZZ^2
       FNDLGAMA = (ZZ-.5D+0)*LOG(ZZ) - ZZ + .9189385332046727 - LOG(ZTERM)
       + ((1D+0/ZZ)*(.8333333333333333D-1 - (ZRZZ*(.2777777777777777D-2
       + (ZRZZ*(.7936507936507936D-3 - (ZRZZ*(.5952380952380952D-3)))))))
       GOTO 8010
8008   IF XXX < 1D+70 THEN FNDLGAMA = ZZ*(LOG(ZZ)-1D+0)
8009   FNDLGAMA = 1D+75
```

```
8010      END DEF
      '  *
      '  ************************************************************************
      '  *
      NEVAL = NEVAL+1
      Z = 0
      PNLT = 0
      XX = NX
      IF IJIJ = 1 THEN GOTO 8100
      IF IJIJ = 2 THEN GOTO 8200
      '  ******************
      '  * MIXED BINOMIAL *
      '  ******************
8100  IF NCOMP = 1 THEN GOTO 8110 '* TWO-COMPONENT MIXED BINOMAIL
      IF NCOMP = 2 THEN GOTO 8120 '* THREE-COMPONENT MIXED BINOMAIL
      IF NCOMP = 3 THEN GOTO 8130 '* FOUR-COMPONENT MIXED BINOMAIL
      '  *
      '  * TWO-COMPONENT MIXED BINOMIAL
      '  *
8110  FOR J = 1 TO NOBSR
        XJ = NOO(J)
        ALGRS = FNDLGAMA(XX+1) - FNDLGAMA(XJ+1) - FNDLGAMA(XX-XJ+1)
        PREA = LOG(X(1)) + XJ*LOG(X(2)) + (NX-XJ)*LOG(1-X(2)) + ALGRS
        IF PREA > -79 THEN YA = EXP(PREA) ELSE YA = 0
      '  *
        PREB = LOG(1-X(1)) + XJ*LOG(X(3)) + (NX-XJ)*LOG(1-X(3)) + ALGRS
        IF PREB > -79 THEN YB = EXP(PREB) ELSE YB = 0
      '  *
        Z = Z+((YA+YB)-PROB(J))^2+PNLT
      NEXT J
      GOTO 8300
      '  *
      '  * THREE-COMPONENT MIXED BINOMIAL
      '  *
      '  * COINSIDER PENALTY IF SUM OF THE WEIGHTS IS GREATER THAN ONE
      '  *
8120  SUM = X(1)+X(2)
      IF SUM > 1 THEN PNLT = (SUM-1)*100000
      FOR J = 1 TO NOBSR
        XJ = NOO(J)
        ALGRS = FNDLGAMA(XX+1) - FNDLGAMA(XJ+1) - FNDLGAMA(XX-XJ+1)
        PREA = LOG(X(1)) + XJ*LOG(X(3)) + (NX-XJ)*LOG(1-X(3)) + ALGRS
        IF PREA > -79 THEN YA = EXP(PREA) ELSE YA = 0
      '  *
        PREB = LOG(X(2)) + XJ*LOG(X(4)) + (NX-XJ)*LOG(1-X(4)) + ALGRS
        IF PREB > -79 THEN YB = EXP(PREB) ELSE YB = 0
      '  *
        PREC = XJ*LOG(X(5)) + (NX-XJ)*LOG(1-X(5)) + ALGRS
        Y = 1-X(1)-X(2)
        IF Y > 0 THEN PREC = PREC+LOG(Y)
        IF PREC > -79 THEN YC = EXP(PREC) ELSE YC = 0
      '  *
        Z = Z+((YA+YB+YC)-PROB(J))^2+PNLT
      NEXT J
      GOTO 8300
      '  *
```

```
      '  * FOUR-COMPONENT MIXED BINOMIAL
      '  *
      '  * CONSIDER PENALTY IF SUM OF THE WEIGHTS IS GREATER THAN ONE
      '  *
8130  SUM = X(1)+X(2)+X(3)
      IF SUM > 1 THEN PNLT = (SUM-1)*100000
      FOR J = 1 TO NOBSR
        XJ = NOO(J)
        ALGRS = FNDLGAMA(XX+1) - FNDLGAMA(XJ+1) - FNDLGAMA(XX-XJ+1)
        PREA = LOG(X(1)) + XJ*LOG(X(4)) + (NX-XJ)*LOG(1-X(4)) + ALGRS
        IF PREA > -79 THEN YA = EXP(PREA) ELSE YA = 0
      '  *
        PREB = LOG(X(2)) + XJ*LOG(X(5)) + (NX-XJ)*LOG(1-X(5)) + ALGRS
        IF PREB > -79 THEN YB = EXP(PREB) ELSE YB = 0
      '  *
        PREC = LOG(X(3)) + XJ*LOG(X(6)) + (NX-XJ)*LOG(1-X(6)) + ALGRS
        IF PREC > -79 THEN YC = EXP(PREC) ELSE YC = 0
      '  *
        PRED = XJ*LOG(X(7)) + (NX-XJ)*LOG(1-X(7)) + ALGRS
        Y = 1-X(1)-X(2)-X(3)
        IF Y > 0 THEN PRED = PRED+LOG(Y)
        IF PRED > -79 THEN YD = EXP(PRED) ELSE YD = 0
      '  *
        Z = Z+((YA+YB+YC+YD)-PROB(J))^2+PNLT
      NEXT J
      GOTO 8300
      '  *********
      '  * POLYA *
      '  *********
8200  IF NCOMP = 1 THEN GOTO 8210
      IF NCOMP = 2 THEN GOTO 8220
      IF NCOMP = 3 THEN GOTO 8230
      '  *
      '  * SIMPLE POLYA
      '  *
8210  FOR J = 1 TO NOBSR
        XJ = NOO(J)
        ALGRS = FNDLGAMA(XX+1) - FNDLGAMA(XJ+1) - FNDLGAMA(XX-XJ+1)
        YLOG = ALGRS + FNDLGAMA(X(1)+XJ) + FNDLGAMA(X(2)+NX-XJ)
        + FNDLGAMA(X(1)+X(2)) - FNDLGAMA(X(1)) - FNDLGAMA(X(2))
        - FNDLGAMA(X(1)+X(2)+NX)
        IF YLOG > -79 THEN Y = EXP(YLOG) ELSE Y = 0
      '  *
        Z = Z+(Y-PROB(J))^2
      NEXT J
      GOTO 8300
      '  *
      '  * TWO-COMPONENT MIXED POLYA
      '  *
8220  FOR J = 1 TO NOBSR
        XJ = NOO(J)
        ALGRS = FNDLGAMA(XX+1) - FNDLGAMA(XJ+1) - FNDLGAMA(XX-XJ+1)
        YLOG1 = LOG(X(1)) + ALGRS + FNDLGAMA(X(2)+XJ) + FNDLGAMA(X(3)+NX-XJ)
        + FNDLGAMA(X(2)+X(3)) - FNDLGAMA(X(2)) - FNDLGAMA(X(3))
        - FNDLGAMA(X(2)+X(3)+NX)
        IF YLOG1 > -79 THEN Y1 = EXP(YLOG1) ELSE Y1 = 0
```

```
'  *
   YLOG2 = LOG(1-X(1)) + ALGRS + FNDLGAMA(X(4)+XJ) + FNDLGAMA(X(5)+NX-XJ)
   + FNDLGAMA(X(4)+X(5)) - FNDLGAMA(X(4)) - FNDLGAMA(X(5))
   - FNDLGAMA(X(4)+X(5)+NX)
   IF YLOG2 > -79 THEN Y2 = EXP(YLOG2) ELSE Y2 = 0
'  *
   Z = Z+((Y1+Y2)-PROB(J))^2+PNLT
NEXT J
GOTO 8300
'  *
'  * THREE-COMPONENT MIXED POLYA
'  *
'  * CONSIDER PENALTY IF SUM OF THE WEIGHTS IS GREATER THAN 1
8230 SUM = X(1)+X(2)
IF SUM > 1 THEN PNLT = (SUM-1)*100000
FOR J = 1 TO NOBSR
   XJ = NOO(J)
   ALGRS = FNDLGAMA(XX+1) - FNDLGAMA(XJ+1) - FNDLGAMA(XX-XJ+1)
   YLOG1 = LOG(X(1)) + ALGRS+FNDLGAMA(X(3)+XJ) + FNDLGAMA(X(4)+NX-XJ)
   + FNDLGAMA(X(3)+X(4)) - FNDLGAMA(X(3)) - FNDLGAMA(X(4))
   -FNDLGAMA(X(3)+X(4)+NX)
   IF YLOG1 > -79 THEN Y1 = EXP(YLOG1) ELSE Y1 = 0
'  *
   YLOG2 = LOG(X(2)) + ALGRS + FNDLGAMA(X(5)+XJ) + FNDLGAMA(X(6)+NX-XJ)
   + FNDLGAMA(X(5)+X(6)) - FNDLGAMA(X(5)) - FNDLGAMA(X(6))
   - FNDLGAMA(X(5)+X(6)+NX)
   IF YLOG2 > -79 THEN Y2 = EXP(YLOG2) ELSE Y2 = 0
'  *
   YLOG3 = ALGRS + FNDLGAMA(X(7)+XJ) + FNDLGAMA(X(8)+NX-XJ)
   + FNDLGAMA(X(7)+X(8)) - FNDLGAMA(X(7)) - FNDLGAMA(X(8))
   - FNDLGAMA(X(7)+X(8)+NX)
   YY = 1-X(1)-X(2)
   IF YY > 0 THEN YLOG3 = YLOG3 + LOG(YY)
   IF YLOG3 > -79 THEN Y3 = EXP(YLOG3) ELSE Y3 = 0
'  *
   Z = Z+((Y1+Y2+Y3)-PROB(J))^2+PNLT
NEXT J
'  *
8300
SN = Z
RETURN
'  *
SUBPRINT:
'  *              **********************
'  *              * SUBROUTINE PRINT *
'  *              **********************
'  ************************************************************************
'  *                                                                      *
'  * THIS SUBROUTINE PRINTS THE ESTIMATED PARAMETERS OF THE SELECTED PRIOR *
'  * DISTRIBUTION IN APPROPRIATE FORMAT.                                  *
'  *                                                                      *
'  ************************************************************************
'  *
'  *
BEEP 2
IOUPT = IOUPT+1
```

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

```
      '  *
      '  * WRITE OUTPUT STARS
      '  *
      CLS : PRINT TAB(2) STRING$(71,"*")
      PRINT TAB(30) "OUTPUT #";USING "##";IOUPT
      PRINT TAB(2) STRING$(71,"*")
      '  *
      IF IJIJ = 1 THEN GOTO 9100
      IF IJIJ = 2 THEN GOTO 9200
      '  ******************
      '  * MIXED BINOMIAL *
      '  ******************
9100  PRINT " *MIXED BINOMAIL*MIXED BINOMAIL*MIXED BINOMAIL*MIXED BINOMAIL*MIXED BIN*"
      PRINT " LIM = ";USING "####";LIM;
      PRINT TAB(20) "TOL = ";USING "##.#####";TOL
      PRINT " LOTS SAMPLED = ";USING "#####";NLOT
      PRINT "  SAMPLE SIZE = ";USING "#####";NX
      IF NCOMP = 1 THEN GOTO 9110  '* TWO-COMPONENT MIXED BINOMIAL
      IF NCOMP = 2 THEN GOTO 9120  '* THREE-COMPONENT MIXED BINOMIAL
      IF NCOMP = 3 THEN GOTO 9130  '* FOUR-COMPONENT MIXED BINOMIAL
      '  *
      '  * TWO-COMPONENT MIXED BINOMIAL
      '  *
9110  PRINT " MIXED BINOMIAL PARAMETERS ARE : W(1) = ";USING "##.#######";X(1);
      PRINT TAB(55) "F(1) = ";USING "##.#######";X(2)
      XWLST = 1-X(1)
      PRINT TAB(34) "W(2) = ";USING "##.#######";XWLST;
      PRINT TAB(55) "F(2) = ";USING "##.#######";X(3)
      GOTO 9300
      '  *
      '  * THREE-COMPONENT MIXED BINOMIAL
      '  *
9120  PRINT " MIXED BINOMIAL PARAMETERS ARE : W(1) = ";USING "##.#######";X(1);
      PRINT TAB(55) "F(1) = ";USING "##.#######";X(3)
      PRINT TAB(34) "W(2) = ";USING "##.#######";X(2);
      PRINT TAB(55) "F(2) = ";USING "##.#######";X(4)
      XWLST = 1-X(1)-X(2)
      PRINT TAB(34) "W(3) = ";USING "##.#######";XWLST;
      PRINT TAB(55) "F(3) = ";USING "##.#######";X(5)
      GOTO 9300
      '  *
      '  * FOUR-COMPONENT MIXED BINOMIAL
      '  *
9130  PRINT " MIXED BINOMIAL PARAMETERS ARE : W(1) = ";USING "##.#######";X(1);
      PRINT TAB(55) "F(1) = ";USING "##.#######";X(4)
      PRINT TAB(34) "W(2) = ";USING "##.#######";X(2);
      PRINT TAB(55) "F(2) = ";USING "##.#######";X(5)
      PRINT TAB(34) "W(3) = ";USING "##.#######";X(3);
      PRINT TAB(55) "F(3) = ";USING "##.#######";X(6)
      XWLST = 1-X(1)-X(2)-X(3)
      PRINT TAB(34) "W(4) = ";USING "##.#######";XWLST;
      PRINT TAB(55) "F(4) = ";USING "##.#######";X(7)
      GOTO 9300
      '  *********
      '  * POLYA *
      '  *********
```

```
9200  PRINT " *POLYA*POLYA*POLYA*POLYA*POLYA*POLYA*POLYA*POLYA*POLYA*POLYA*POL*"
      PRINT " LIM = ";USING "####";LIM;
      PRINT TAB(15) "TOL = ";USING "##.#####";TOL
      PRINT " LOTS SAMPLED = ";USING "#####";NLOT
      PRINT "  SAMPLE SIZE = ";USING "#####";NX
      IF NCPOL = 1 THEN GOTO 9210  '* SIMPLE POLYA
      IF NCPOL = 2 THEN GOTO 9220  '* TWO-COMPONENT MIXED POLYA
      IF NCPOL = 3 THEN GOTO 9230  '* THREE-COMPONENT MIXED POLYA
      '  *
      '  * SIMPLE POLYA
      '  *
9210  PRINT " POLYA PARAMETERS ARE : "
      PRINT TAB(7) "S =";USING "###.#######";X(1);
      PRINT TAB(25) "T =";USING "###.#######";X(2)
      GOTO 9300
      '  *
      '  * TWO-COMPONENT MIXED POLYA
      '  *
9220  PRINT " MIXED POLYA PARAMETERS ARE : "
      PRINT TAB(7) "W(1) = ";USING "##.######";X(1);
      PRINT TAB(25) "S(1) =";USING "###.#######";X(2);
      PRINT TAB(44) "T(1) =";USING "###.#######";X(3)
      XWLST = 1-X(1)
      PRINT TAB(7) "W(2) = ";USING "##.######";XWLST;
      PRINT TAB(25) "S(2) =";USING "###.#######";X(4);
      PRINT TAB(44) "T(2) =";USING "###.#######";X(5)
      GOTO 9300
      '  *
      '  * THREE-COMPONENT MIXED POLYA
      '  *
9230  PRINT " MIXED POLYA PARAMETERS ARE : "
      PRINT TAB(7) "W(1) = ";USING "##.######";X(1);
      PRINT TAB(25) "S(1) =";USING "###.#######";X(3);
      PRINT TAB(44) "T(1) =";USING "###.#######";X(4)
      PRINT TAB(7) "W(2) = ";USING "##.######";X(2);
      PRINT TAB(25) "S(2) =";USING "###.#######";X(5);
      PRINT TAB(44) "T(2) =";USING "###.#######";X(6)
      XWLST = 1-X(1)-X(2)
      PRINT TAB(7) "W(3) = ";USING "##.######";XWLST;
      PRINT TAB(25) "S(3) =";USING "###.#######";X(7);
      PRINT TAB(44) "T(3) =";USING "###.#######";X(8)
      '  *
9300  PRINT " SUM OF SQUARES OF DIFFERENCES BETWEEN DATA POINTS"
      PRINT USING "  AND FITTED CURVE = ##.##############& ";SN
      '  *
      PRINT TAB(2) STRING$(71,"*")
      PRINT TAB(2) STRING$(71,"*")
      RETURN
```

ภาคผนวก ย.

บัญชีศัพท์ของตัวแปรต่าง ๆ ที่ใช้ใน Pattern Search Subroutine

ALP : อัลฟ่า ; เป็นแฟคเตอร์ซึ่งเป็น Step size D(I) เพิ่มเมื่อการเคลื่อนไปข้างหน้าเป็นผลสำเร็จ และกำหนดให้ L4 = 2 (ค่านี้จะเริ่มต้นที่ 2.0 และจะใช้ใน Statement ที่ 6292)

BET : เบต้า ; เป็นแฟคเตอร์ผลคูณซึ่งเป็น Step Size สำหรับค่าตัวแปรอิสระที่ลดลง ถ้าการเคลื่อนไปข้างหน้าและเคลื่อนกลับ สำหรับที่ตัวแปรนั้นตกไป

D(I) : ค่าปัจจุบันของ Step Size สำหรับตัวแปรอิสระที่ i

DEL : เดลต้า ; เป็นตัวคูณที่ซึ่งถูกใช้ในการตัดสินใจค่าเริ่มต้นของ D(I) ซึ่งเป็น Step Size

DX : ปริมาณ local ที่ถูกใช้ในการตัดสินใจ Lower Bound บน Step Size ที่ถูกค้นพบ ซึ่งค่า DX นี้ถูกคำนวณและถูกใช้เฉพาะระหว่าง Statement ที่ 6480 กับ 6485 โดยที่ Step Size จะถูกลดแทนที่

D1 : เป็นปริมาณที่ถูกใช้เพิ่มค่าของ GR ใน Adaptive Logic ซึ่งค่า D1 นี้จะ Set ไว้ใน Statement ที่ 6802 + 1 และ 6804 และค่านี้จะถูกใช้ใน Statement ที่ 6810

GR : เป็นแฟคเตอร์ที่ซึ่งเวคเตอร์ Pattern Move เป็นตัวคูณได้รับขนาดที่แท้จริงของ Pattern Move (GR ถูกกำหนดค่าเริ่มต้นที่ 2.2 และถูกปรับขึ้นโดยปกติเพิ่มทีละ 0.1 ระหว่าง Statement ที่ 6510 ถึง 6783) และเมื่อ GR มีค่าในระหว่างโปรแกรมการคำนวณถึง 3.5 จะถูก Set ให้เริ่มใหม่ที่ค่า 2.2 โดยอัตโนมัติ

I : เป็นตัวแปร Highly Localized ที่ถูกใช้เป็นเล่มือนตัวดัชนีใน DO-LOOPS (ดูที่ Statement ที่ 6810, 6420 และ 6786)

ID1, ID2, ID3, ID4 : เป็นตัวนับที่ถูกบันทึกผ่าน Statement ที่ 6804, 6803, 6804 และ 6801 ตามลำดับ

ITR : เป็นตัวอักษรควบคุม Printout ของความสำคัญ ; เริ่มต้นที่ 1 ซึ่งค่านี้จะถูกทดสอบที่ Statement ที่ 6888 กล่าวคือ ถ้าค่านี้ > 1 จะเกิดการยืดลบทิ้งของ Certain Print Line (แต่จะไม่เกิดกรณีนี้ แม้ว่าค่าเริ่มต้นของ ITR > 1 ก็ตาม)

K : เป็น Subscript ที่นิยามของตัวแปรที่กำลังศึกษาอยู่ เช่น X(K), D(K) เป็นต้น

KK : เป็นตัวนับ สำหรับการเพิ่มหนึ่งครั้งในจำนวนของตัวแปรที่ศึกษา ในการทดสอบสุดท้าย สำหรับจุดฐานใหม่ เมื่อ KK ถึง N + 1 การทดสอบ สำหรับจุดฐานใหม่ จะถูกทำซ้ำ (คำอธิบายข้างต้นนี้ประยุกต์ เมื่อ Subroutine ใน Full Exploratory Search Mode ถูกแสดงโดย LT ⩽ 0 นี้เป็น Mode ปกติทั่วไป) ซึ่งค่า KK นี้จะ set ให้ = 1 ใน Statement ที่ 6100 + 41 6440 + 2 และ 6784 ใน Full Exploratory Search Mode (LT ⩽ 0) และค่า KK ถูกเพิ่มขึ้นที่ Statement ที่ 6330 และจะถูกทดสอบที่ Statement ที่ 6330 + 1 และ 6404 ฉะนั้นการตัด Search Mode (LT ⩾ 1) ค่า KK จะถูกเพิ่มที่ Statement ที่ 6778 และถูกทดสอบที่ Statement ที่ 6404

KOUNT : ตัวนับของจำนวนครั้งที่เราป้อน Adaptive Logic ซึ่งการพยายามเตรียมการ Pattern Move (มีค่าเหมือนกับ LT7) ซึ่งจะเพิ่มที่ Statement ที่ 6510 + 1

KPRESS : ตัวอักษรควบคุม Printout ซึ่งใส่ข้อมูลตัวแรกที่อ่านเข้าไปโดย Main Routine KPRESS = 1 จะหยุดได้ประมาณ 99% ของ Output ที่ถูกสร้างโดย Pattern Search Subroutine; KPRESS = 0 จะอนุญาตให้มีรายละเอียดเต็มที่ที่จะเป็น Output ระหว่างการคำนวณของ Pattern Search Subroutine

LA : เป็น Master Monitor ของสถานะ Subroutine ที่จะบอกถึงที่หมายที่จะ GOTO ต่อไป งานแรกเริ่มของ LA นั้น จะเป็นการควบคุมการ เคลื่อนที่ผ่านไปใน Boundary Check และ Objective Function Subroutine และค่าต่อไปนี้ ของ LA จะขึ้นกับจุดมุ่งหมายดังต่อไปนี้

ค่าของ LA	Statement ที่	จุดมุ่งหมายในงาน
1	6100	การเริ่มต้นของ Subroutine
2	6282	Forward Exploratory Move (ปกติ)
3	6463	Reverse Exploratory Move (ปกติ) (การ Forward ล้มเหลว)
4	6580	ประเมินค่าจุดพื้นฐานตาม Pattern Move
5	6285	Forward Exploratory Move (ตาม Pattern Move)
6	6466	Reverse Exploratory Move (ตาม Pattern Move)
7	6510	พยายามปรับ Pattern Move
8	6500	Terminate Search และออกจาก Subroutine

LIM : ประมาณ Upper Bound บนจำนวนของการเรียก Objective Function ขณะเรียก Pattern Search Subroutine (โดยปกติแล้วค่า LIM นี้จะกำหนดไว้ที่ 3000 แล้วแต่ความเหมาะสมในกรณีต่าง ๆ ที่ทำให้ค่านี้แตกต่างกันออกไป)

LSN : กำหนดที่อยู่ใน Statement ที่ 6100 + 3 ของ Subroutine และเก็บค่าไว้ที่นั่น ไม่นำมากล่าวอ้างอิงในการคำนวณยกเว้นใน Statement ที่ 6784 + 1

LT : ควบคุมข้อเลือกระหว่าง Statement full Exploratory Search Mode (LT ≤ 0) และ Trucated Search Mode (LT ≥ 1) ภายใต้ Truncated Mode นั้น Exploratory Search จะถูกหยุดในขณะเดียวกันกับที่ขบวนการเคลื่อนที่ใด ๆ จะพิสูจน์ในจุดฐานสุดท้าย จุดนี้จะเป็นจุดที่ประหยัดและถือว่าเป็นจุดฐานใหม่ และ Pattern Move ถูกกระทำขึ้น ฉะนั้น Exploratory ต่อไป จะเริ่มต้นค้นหาด้วย ตัวแปรถัดไปซึ่งประสบได้โดยความสำเร็จสุดท้าย ภายใต้ Full Exploratory Search Mode ซึ่งเป็น Mode ที่ลงรหัสอีกครั้งตั้งแต่ LT = 0 และ Exploratory Move นั้นถูกกระทำด้วยตัวแปรทั้งหมด N ตัว ก่อนที่ Pattern Move จะถูกพยายามอีกครั้ง ค่า LT นี้จะกำหนดใน Statement ที่ 6100 + 2 และไม่ถูกนำไปใช้เลย นอกจาก Statement ที่ 6320 และ 6400

LT2, LT3, LT4, LT5, LT6, LT7 : จำนวนครั้งของ Subroutine นี้มีถึง สถานะภาพที่ LA = 2* , LA = 3*, LA = 4, LA = 5*, LA = 6* LA = 7 ตามลำดับ

หมายเหตุ * หมายถึง ในเหตุการณ์ที่ย้อนกลับนั้น Forward หรือ Reverse Move สลับเหลว ฉะนั้น Boundary Check ที่เหมาะสมอันหนึ่งของตัวนับนั้นจะสลับเหลวที่ถูกเริ่มขึ้น

L4 : ตัวแปรที่คงสถานะไว้เสมอเท่ากับ 1 หรือ 2 นั่นคือ L4 = 1 กล่าวได้ว่าเราอยู่ในขบวนการของการกระทำ Exploratory ในรูปปกติ เช่น เรากำลังหาสำหรับ Pattern หนึ่ง ๆ ส่วน L4 = 2 นั้นกล่าวได้ว่า เราอยู่ในขบวนการของการกระทำ Exploratory Moveตาม Pattern Move

M1 : บวกหนึ่งจำนวนของตัวแปรอิสระที่เคยเป็นทั้ง Reverse และ Forward ที่สลับเหลวด้วย Step Size ที่ต่ำสุด ตั้งแต่การทดสอบสุดท้ายสำหรับจุดฐานใหม่ การค้นหาจะหมดสิ้น เมื่อ M1 ≥ N + 1 ระหว่างที่ Exploratory Search Move นั่นคือ เมื่อ L4 = 1 และ เรากำลังพยายามที่จะสร้าง Pattern ใหม่ แม้ว่า M1 อาจจะถูกใช้ขณะ ค้นหาปฏิบัติการ Post-Pattern Move Exploratory (เมื่อ L4 = 2) และไม่เคยนำมาใช้ทดสอบอะไรเลย ยกเว้น เมื่อพยายามที่จะสร้างชื่อ Pattern ใหม่ เมื่อ L4 = 1

(ในการ Reprogramming นั้น ตัวแปรนี้ จะถูกลดค่าลงโดย 1 นั่นคือ จุดเริ่มต้นสู่ศูนย์ และถูกทดสอบ สำหรับ $\geq n$ ด้วยพิสูจน์อย่างแจ้งชัดเจน ของการแปลผล ซึ่งในกรณีนี้สามารถประยุกต์สู่ M2 ได้) M1 นี้จะเริ่มต้นใน Statement ที่ 6100+38, 6300+2, 6352+1 และ 6440+1 โดยจะถูกเพิ่มค่าที่ Statement ที่ 6490 และถูกทดสอบที่ Statement ที่ 6429 + 1

M2 : บวกหนึ่งจำนวนของตัวแปรอิสระ ซึ่งเคยเป็นทั้ง Reverse และ Forward สัมเหลว ตั้งแต่ความพยายามสุดท้ายสู่การกระทำ Pattern Move โดยที่ Pattern นี้จะถูกพิจารณาเต็มที่และค้นหาก็จะถูกซ่อมแซม สู่ Exploratory Mode (ค่าของ L4 เปลี่ยนจาก 2 เป็น 1) ถ้าตัวแปรทุกตัวสัมเหลวทั้งหมด Single Pattern Move ก็จะถูกพยายามใหม่อีกครั้ง

N : จำนวนของตัวแปรอิสระนั่นคือ ความมีมิติของที่ว่างที่ต้องการค้นหา

NEVAL : ตัวนับของจำนวนของการตีค่าของ Objective Function FCTI ซึ่งถูกกระทำระหว่าง Particular ที่เรียกว่า Pattern Search Subroutine NEVAL นี้ถูกทำให้เพิ่มด้านในของ Subroutine FCTI; มันไม่ถูกทดสอบทุกครั้ง เมื่อมันถูกทำให้เพิ่มขึ้น และ LIM นี้จะเป็นการประมาณ ค่า Upper Bound บน NEVAL การทดสอบนี้ จะถูกกระทำใน Statement ที่ 7 ที่ Prior จะพยายามสู่ Pattern Move สิ่งนี้จะเป็นสิ่งที่ดีกว่าการทดสอบ ทุก ๆ ครั้งของการเรียก FCTI

NEVOLD : ตัวแปรที่ถูกใช้สู่เก็บค่าที่ได้ก่อนแล้วของ NEVAL และจำนวนผลรวมของการประเมินค่าของ Objective Function ถูกทำต่อไป ฉะนั้น NEVOLD จะถูกกำหนดที่ Statement ที่ 6815 + 2 และถูกใช้ในการคำนวณของค่า V ที่ Statement ที่ 6398

NPF : ตัวนับสำหรับจำนวนของ Pattern Move ที่พยายามจนประสบผลสำเร็จซึ่งถูกทำตาม โดยความสัมเหลวของ Step ทั้งหมดแต่ละขั้นตอนที่พยายามจะปรับ Pattern Move (กำหนดค่าที่ 0 ที่ Statement ที่ 6100 + 5, 6300 +3 6420 + 1 และถูกเพิ่มขึ้นที่ Statement ที่ 6780 + 1 และถูกทดสอบที่ Statement ที่ 6353) ดูคำอธิบายของ M2 ประกอบ

OLDSN : ตัวแปรหนึ่งที่ถูกใช้เก็บค่า Prior ที่แน่นอนของ SN ซึ่งจะถูกกำหนดให้เริ่มต้นที่ Statement ที่ 6180 + 35 โดยตั้งต้นใหม่ที่ Statement ที่ 6815 และใช้ในการคำนวณของค่า V ที่ Statement ที่ 6398 (ภายใน Adaptive Logic)

OLDV : ตัวแปรที่ถูกใช้เก็บค่าก่อนของ V (ถูกกำหนดเริ่มต้นที่ Statementที่ 6815 + 1 และถูกใช้ใน Statement ที่ 6782 สู่การเปรียบเทียบ ค่าเก่าและค่าใหม่ของ V)

P : เป็นตัวแปรที่ Highly Localized Temporary Storage ถูกใช้เฉพาะระหว่าง Statement ที่ 6530 + 3 และ 6530 + 5 โดยที่ Pattern Move ถูกพยายามอยู่

Q(I) : เป็น Storage Matrix สำหรับการเก็บค่าเก่านั่นคือจุดฐานของตัวแปรอิสระ ซึ่งค่าเหล่านี้ ถูกกำหนดเริ่มแรกเท่ากับค่าของ X(I) และถูกปรับปรุงใน Statement ที่ 6530 + 4 ถ้าความพยายามของ Pattern Moveล้มเหลว ลงค่าเก่าของ X(I) จะถูกนำมาใช้ใน Q(I) ใน Statement ที่ 6420

SC : ตัวแปรที่ถูกใช้เก็บค่าของ Objective Function ที่จุดฐานที่ใช้ล่าสุดท้าย ค่า SC นี้ จะถูกเริ่มต้นใช้ใน Statement ที่ 6530 (Prior สู่ทางพยายามของ Pattern Move และถูกทดสอบที่ Statement ที่ 6340 สู่การตัดสินใจ แม้ว่า จุดฐานใหม่ถูกสร้างขึ้นและถูกใช้เสริมค่า SP สู่ค่าเดิมใน Statement ที่ 6410 ถ้าความพยายามของ Pattern Move ล้มเหลว

SN : ค่าที่กลับไปโดย Objective Function, Subroutine FCTI

SNOLD : ตัวแปร Redundant ที่เริ่มต้นที่ Statement ที่ 6100 + 34 และไม่เคยอ้างถึงเลยสำหรับข้อเสนอใด ๆ

SP : ตัวแปรที่ยกเว้นในเหตุการณ์ต่อไปนี้ คือ ตัวแปรที่เท่ากับค่าต่ำสุดถูกนำกลับโดย Objective Function การคาดหวังที่เกิดขึ้น เมื่อ Exploratory Search ปฏิบัติทันทีทันใดหลังจาก Pattern Move ในที่ตั้งของ SP ที่กำหนด (ใน Statement ที่ 6580 + 2) เท่ากับค่าของ Objective Function ที่ผลลัพธ์กลับของ Pettern Move ที่ไม่ได้ปรับปรุงและต่อมายกเลิก

Exploratory Search สู่การปรับ Pattern Move กำลังดำเนินอยู่ ค่า SP นี้ถูกปรับปรุงสู่ผลลัพธ์ต่ำสุดของค่านี้และเป็น Exploratory Move ที่ดีที่สุด หลังจากที่ Exploratory Move ถูกทำให้สมบูรณ์เรียบร้อยแล้วนั้น การที่ค่าของ SP ถูกปรับปรุงนี้จะนำมาเปรียบเทียบกับ SC (Statement ที่ 6340) ค่าของ Objective Function ที่จุดฐานสุดท้าย ถ้าจุดฐานนี้ถูกทำการทดสอบและผ่านด้วยนั้น Pattern Move ที่ปรับแล้วจะถูกประกาศถึงผลสำเร็จและจุดฐานใหม่นี้ จะถูกสร้างตั้งไว้ ถ้าจุดฐานทดสอบแล้วไม่ผ่าน Pattern Move จะถูกประกาศถึงความไม่เป็นผลสำเร็จ ค่า SP นั้นจะถูกเก็บไว้อีกครั้ง สู่ค่าเดิมของมัน (ใน Statement ที่ 6410 และ Local Exploration ถูกเริ่มในเรื่องเกี่ยวกับจุดฐานเดิมในความพยายามที่จะสร้าง Pattern ใหม่

TOL : ปริมาณที่ถูกใช้ใน Statement ที่ 6480 + 2 ในการได้มาซึ่ง Lower Bound บน Step Size สำหรับ Exploratory Moves

V : เปอร์เซ็นต์การพิสูจน์ในค่าของ Objective Function ต่อการเรียกของ Objective Functionซึ่งค่านี้ถูกคำนวณ และถูกใช้ใน Adaptive Logic (ดู Statement ที่ 6398 จนถึง 6782)

X(I) : ตัวแปรอิสระที่ตัดสิน (ของ N : $1 \leq I \leq N$)

XMAX(I) : Upper Bound บนค่าที่ยอมรับของ X(I)

XMIN(I) : Lower Bound บนค่าที่ยอมรับของ X(I)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียน

นายไพโรจน์ มีทอง เกิดเมื่อวันที่ 26 เมษายน พ.ศ. 2503 ที่จังหวัดอุตรดิตถ์ สำเร็จการศึกษาปริญญาตรี วิทยาศาสตรบัณฑิต (สถิติ) จากมหาวิทยาลัยเชียงใหม่ เมื่อปีการศึกษา 2524 และเข้าศึกษาต่อระดับปริญญาโท ในภาควิชาสถิติ บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2527 ปัจจุบัน รับราชการในตำแหน่งนักสถิติระดับ 4 กองนโยบายและแผนงานประมง กรมประมง กระทรวงเกษตรและสหกรณ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย