รายการอ้างอิง

ภาษาไทย

ธีระพร วีระถาวร. การอนุมานเชิงสถิติขั้นกลาง: โครงสร้างและความหมาย กรุงเทพมหานคร: พิทักษ์การพิมพ์, 2531.

สากัณย์ สุวรรณการ. วิธีนอนพาราเมตริกสำหรับการประมาณค่าฟังก์ชันการอยู่รอดในปัญหา 2 ตัวอย่างที่มีค่าสังเกตไม่สมบูรณ์ วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาสถิติ บัณฑิตวิทยาลัย

ภาษาอังกฤษ

Ferguson Thomas S. and Eswar G. Phadia, "Bayesian Nonparametric Estimation Based on Censored Data," The Annals of Statistics 7(1979) : 163-186.

Kaplan EL and Meier P., "Nonparametric Estimation From Incomplete Observations," Journal of the American Statistical Association 53(1958) : 457-481.

Lawless J.F, Statistical Models and Methods for lifetime Data. New York: John Wiley and Sons, 1982.

London, D., Survival Models and Their Estimation. Winsted: ACTEX Publications, Inc., 1988.

Rupert G. Miller, Jr., Survival Analysis. New York: John Wiley and Sons, 1981.

Susarla V. and Ryzin J. Van, "Nonparametric Bayesian Estimation of Survival Curve from Incomplete observations," Journal of the American Statistical Association 71(1976) : 897-902.

Tobias, Paul A., Applied reliability. Van Nostrand Reinhod Company Inc., 1986.

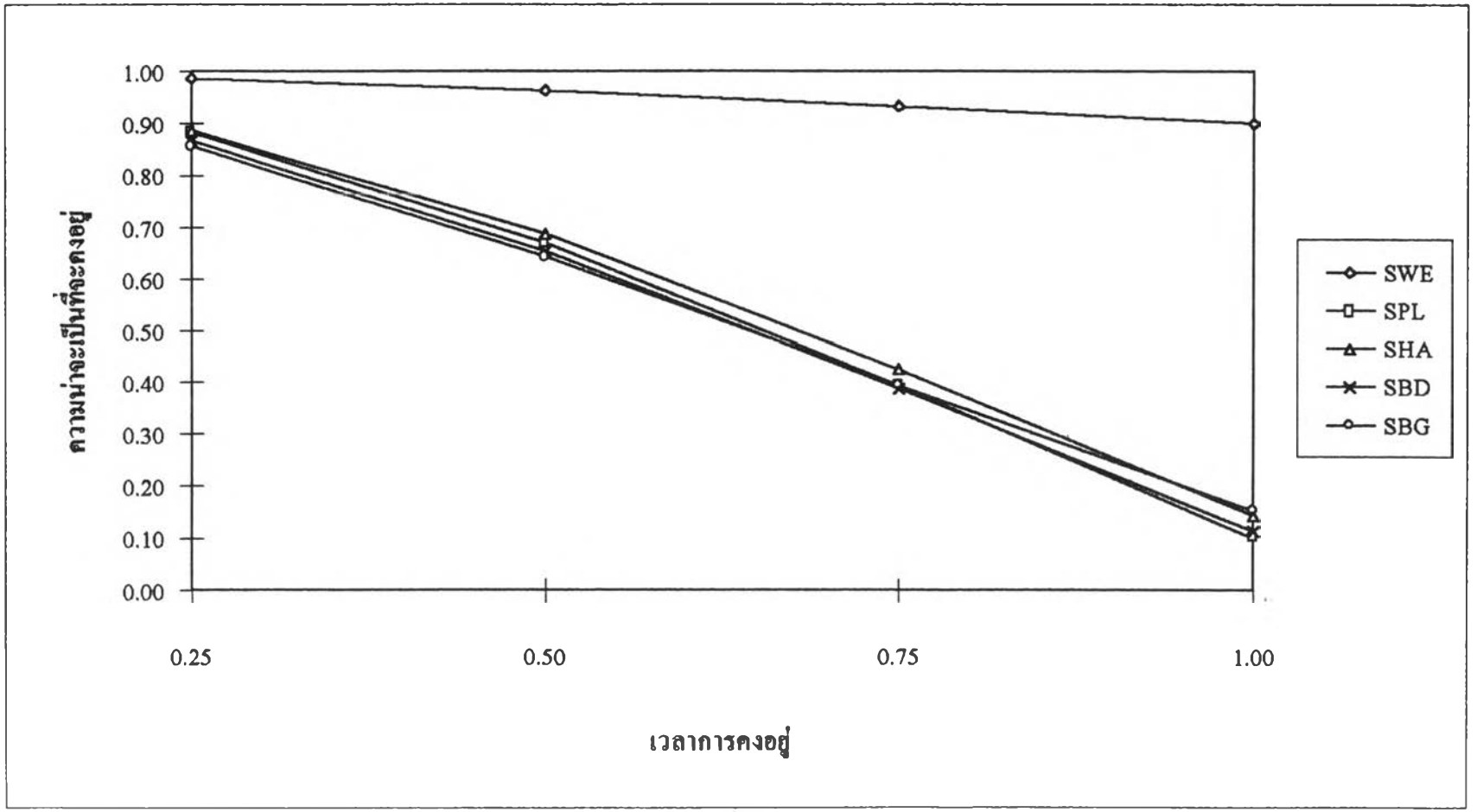
ภาคผนวก

ภาคผนวก ก

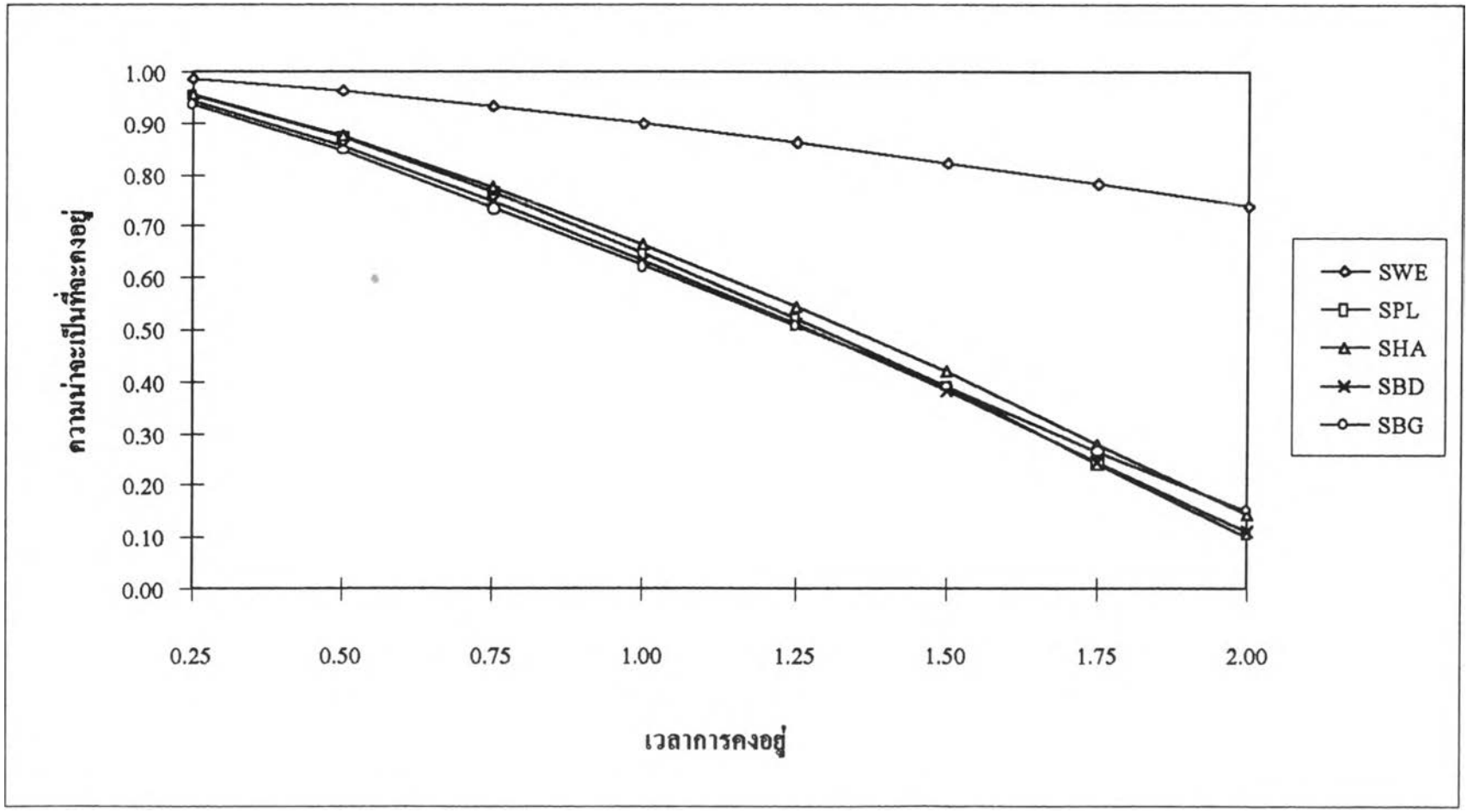
สำหรับการแสดงผลการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีได้แสดงเป็นรูปกราฟ และสามารถอธิบายสัญลักษณ์ต่างๆ ในกราฟได้ดังนี้

Tc	หมายถึง	เวลาที่กำหนดไว้ล่วงหน้า
SWE	หมายถึง	ค่าของฟังก์ชันการคงอยู่จริง ซึ่งมีการแจกแจงแบบไวบูลล์
SLO	หมายถึง	ค่าของฟังก์ชันการคงอยู่จริง ซึ่งมีการแจกแจงแบบลอกนอร์มอล
SPL	หมายถึง	ค่าประมาณฟังก์ชันการคงอยู่ ซึ่งประมาณโดยวิธีพีแอล
SHA	หมายถึง	ค่าประมาณฟังก์ชันการคงอยู่ ซึ่งประมาณโดยวิธีฟังก์ชันภาวะภัย
SBD	หมายถึง	ค่าประมาณฟังก์ชันการคงอยู่ ซึ่งประมาณโดยวิธีเบส์ที่มีการแจกแจงก่อนเป็นกระบวนการดีริชเลต์
SBG	หมายถึง	ค่าประมาณฟังก์ชันการคงอยู่ ซึ่งประมาณโดยวิธีเบส์ที่มีการแจกแจงก่อนเป็นกระบวนการแกมมา

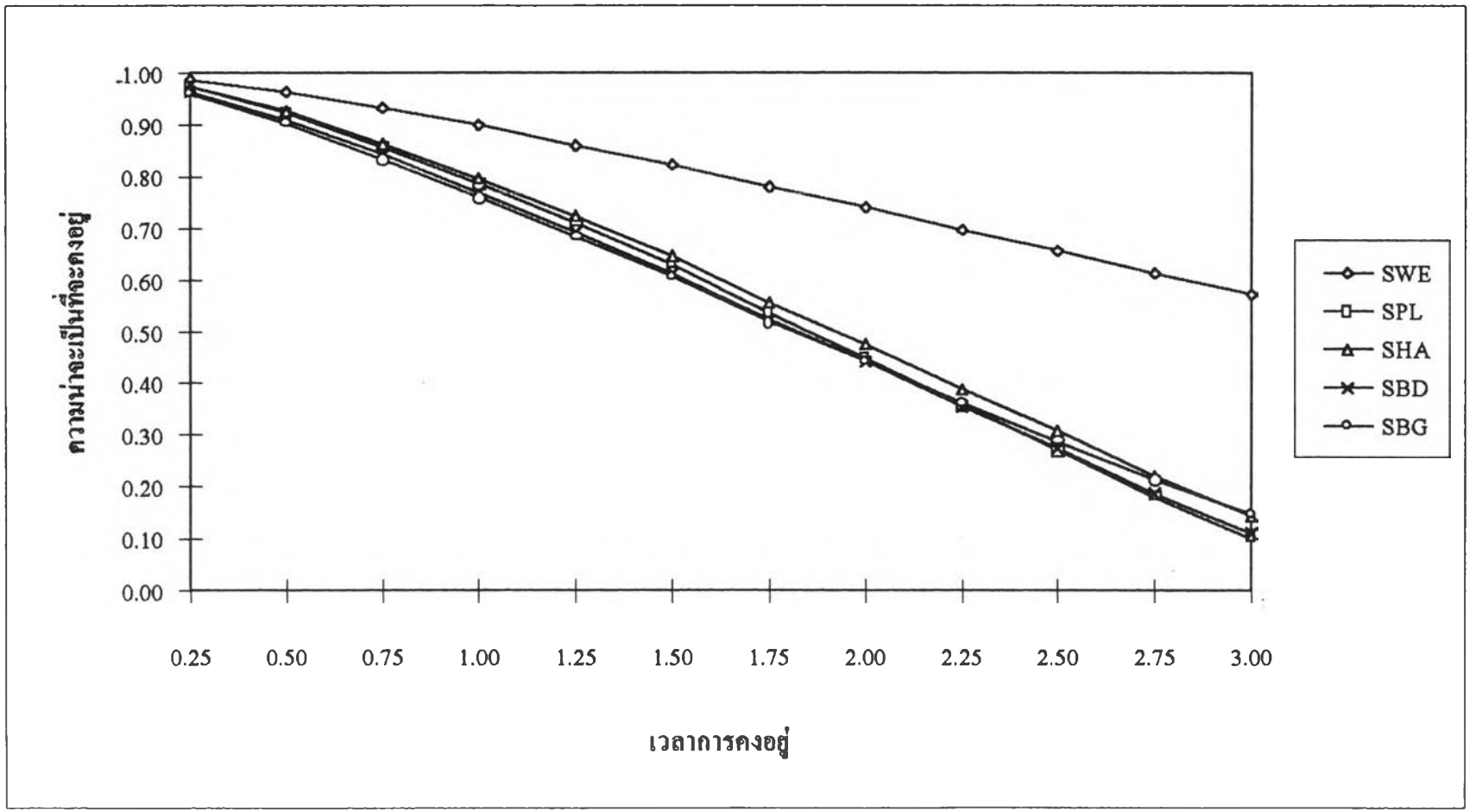
รูปที่ 1 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าน้อยกว่าค่าเฉลี่ยของการแจกแจง 75% (Tc = 1.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



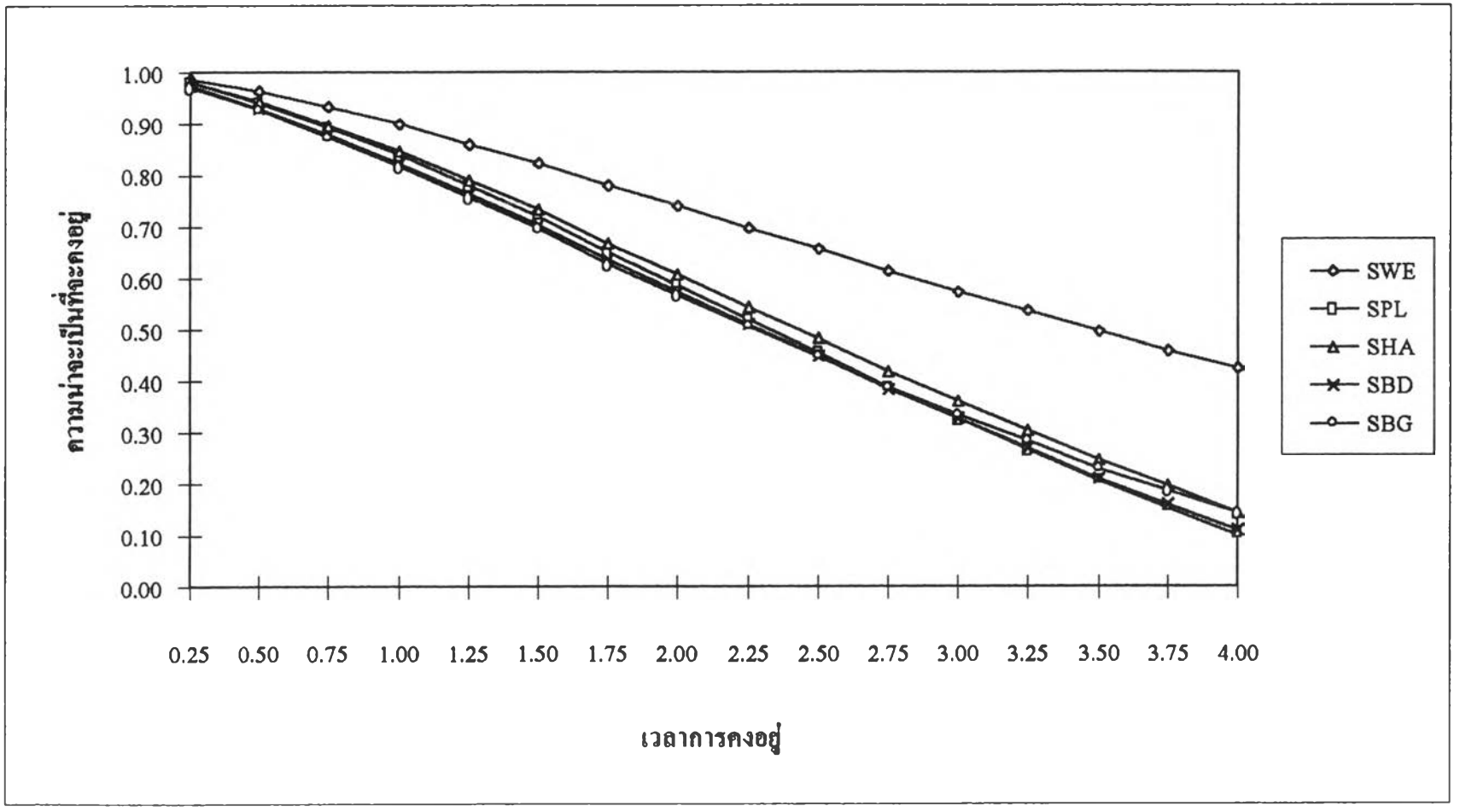
รูปที่ 2 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าน้อยกว่าค่าเฉลี่ยของการแจกแจง 50% (Tc = 2.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



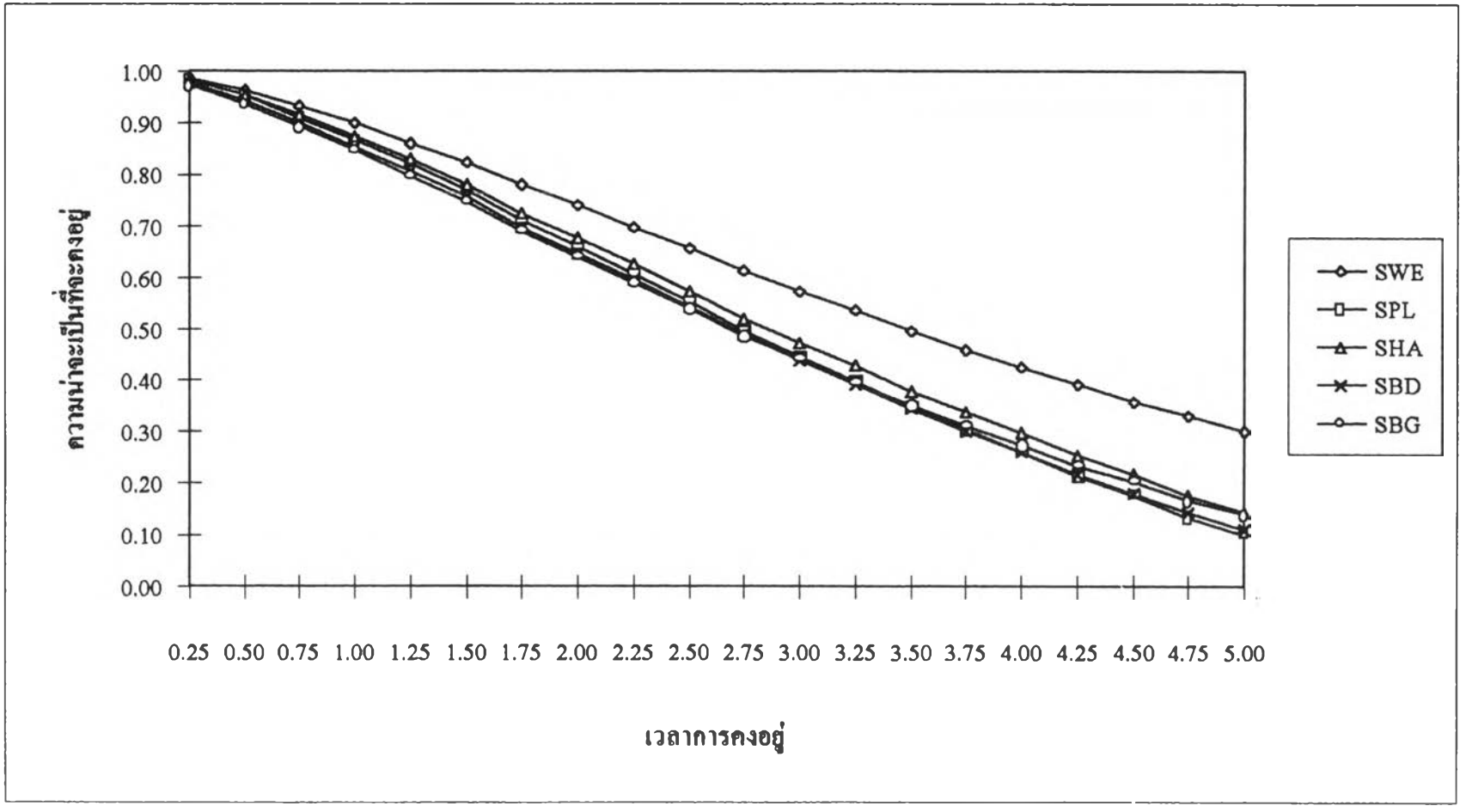
รูปที่ 3 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าน้อยกว่าค่าเฉลี่ยของการแจกแจง 25% (Tc = 3.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



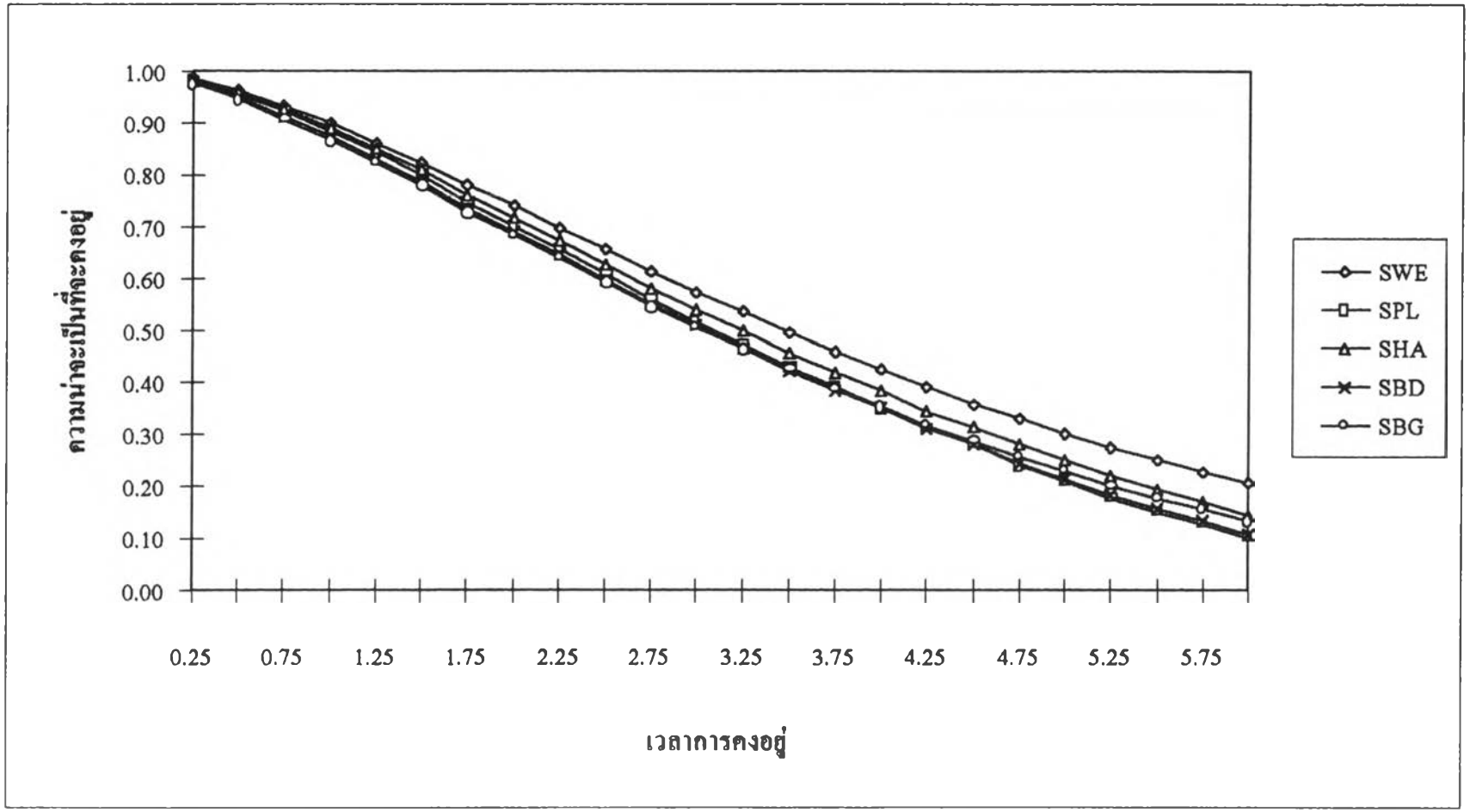
รูปที่ 4 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าเท่ากับค่าเฉลี่ยของการแจกแจง (Tc = 4.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



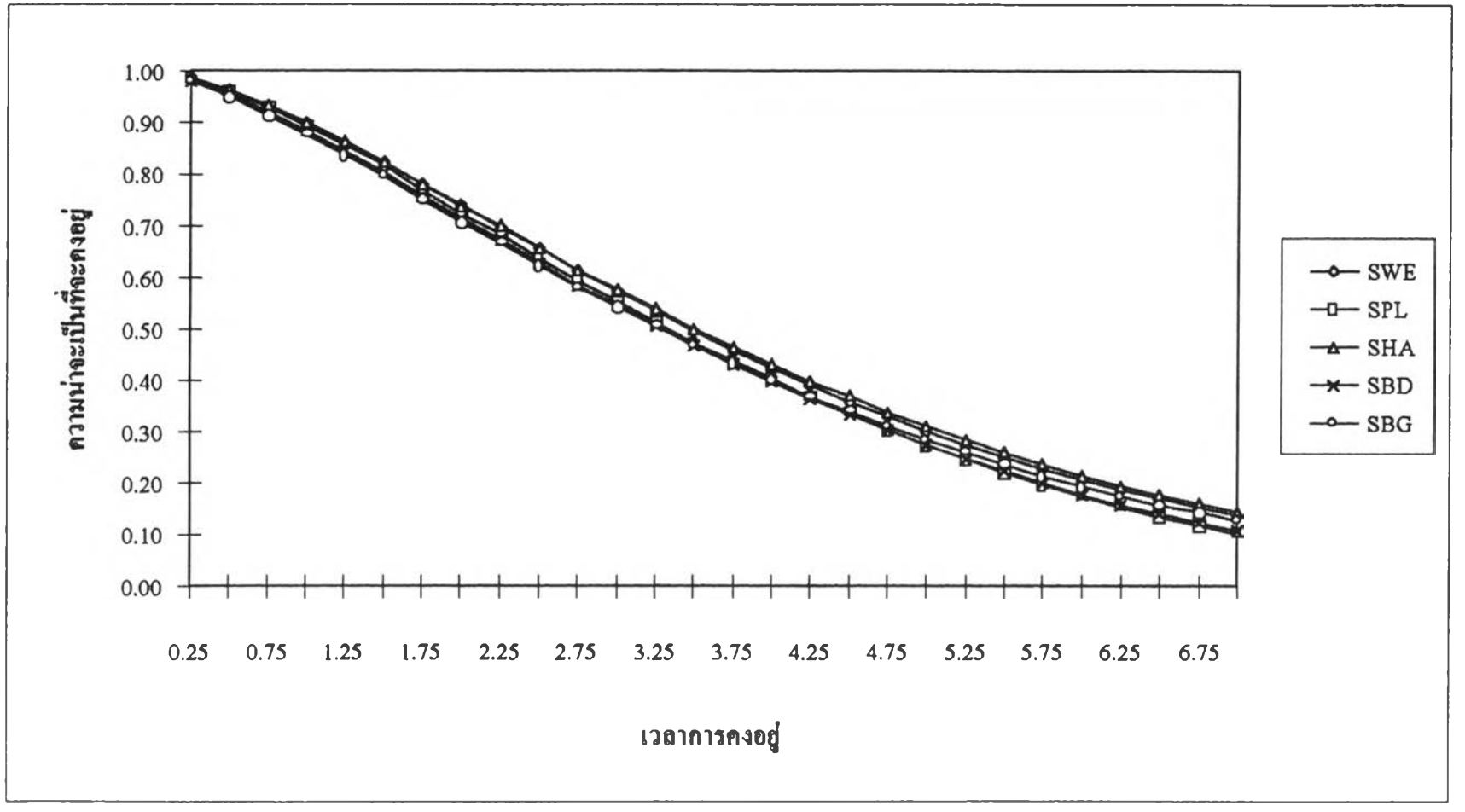
รูปที่ 5 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่ามากกว่าค่าเฉลี่ยของการแจกแจง 25% (Tc = 5.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



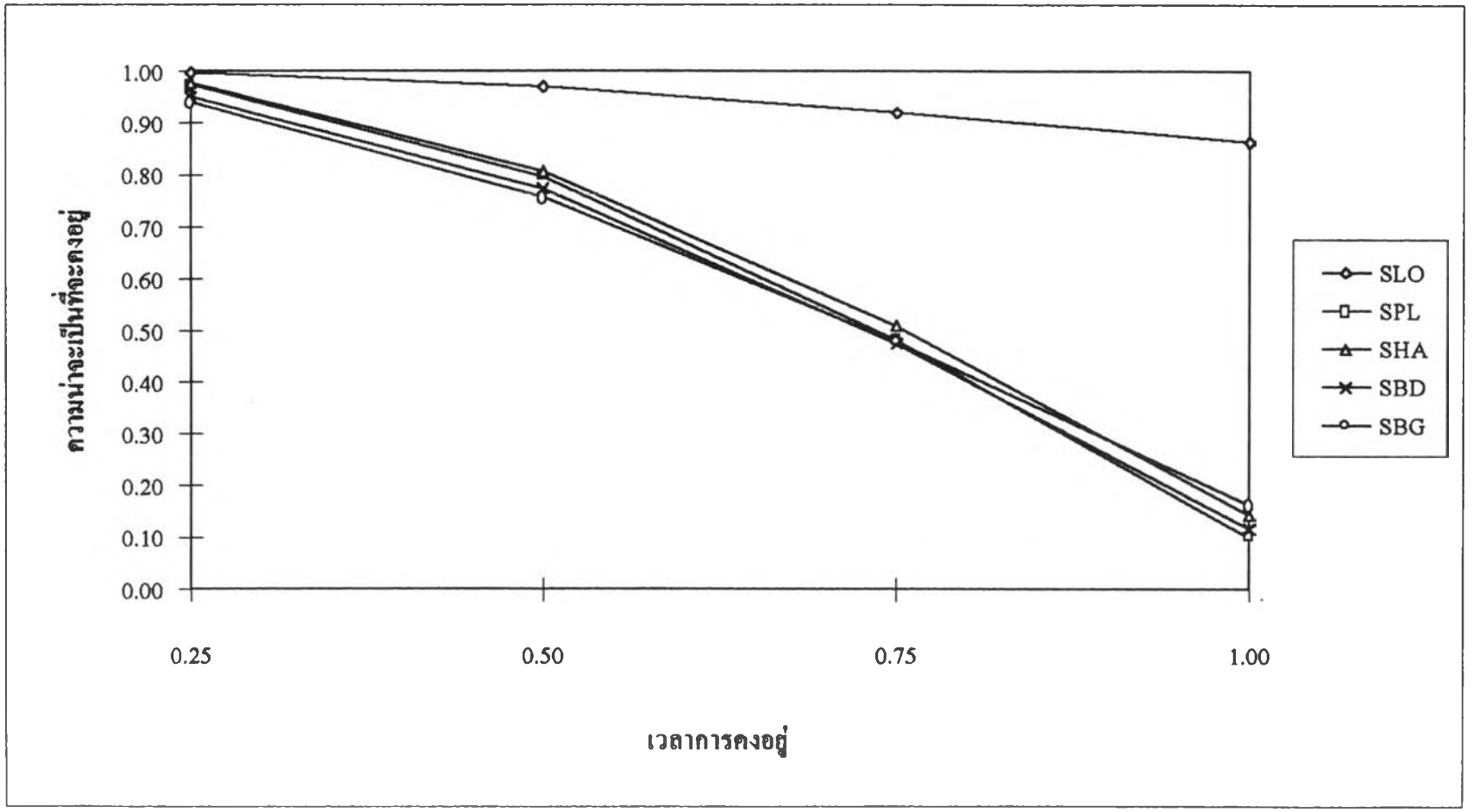
รูปที่ 6 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่ามากกว่าค่าเฉลี่ยของการแจกแจง 50% (Tc = 6.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



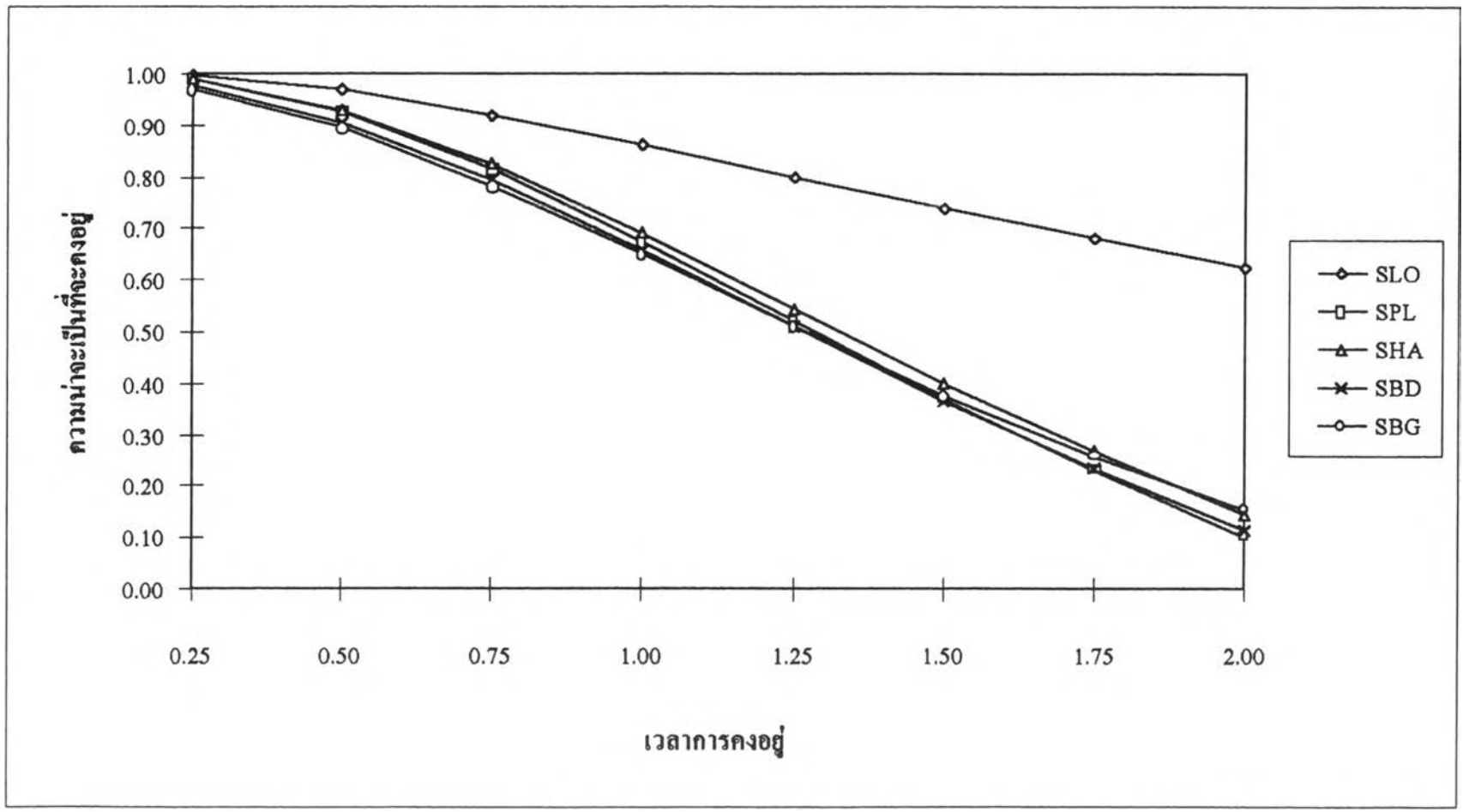
รูปที่ 7 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธี โดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบไวบูลล์เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่ามากกว่าค่าเฉลี่ยของการแจกแจง 75% (Tc = 7.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



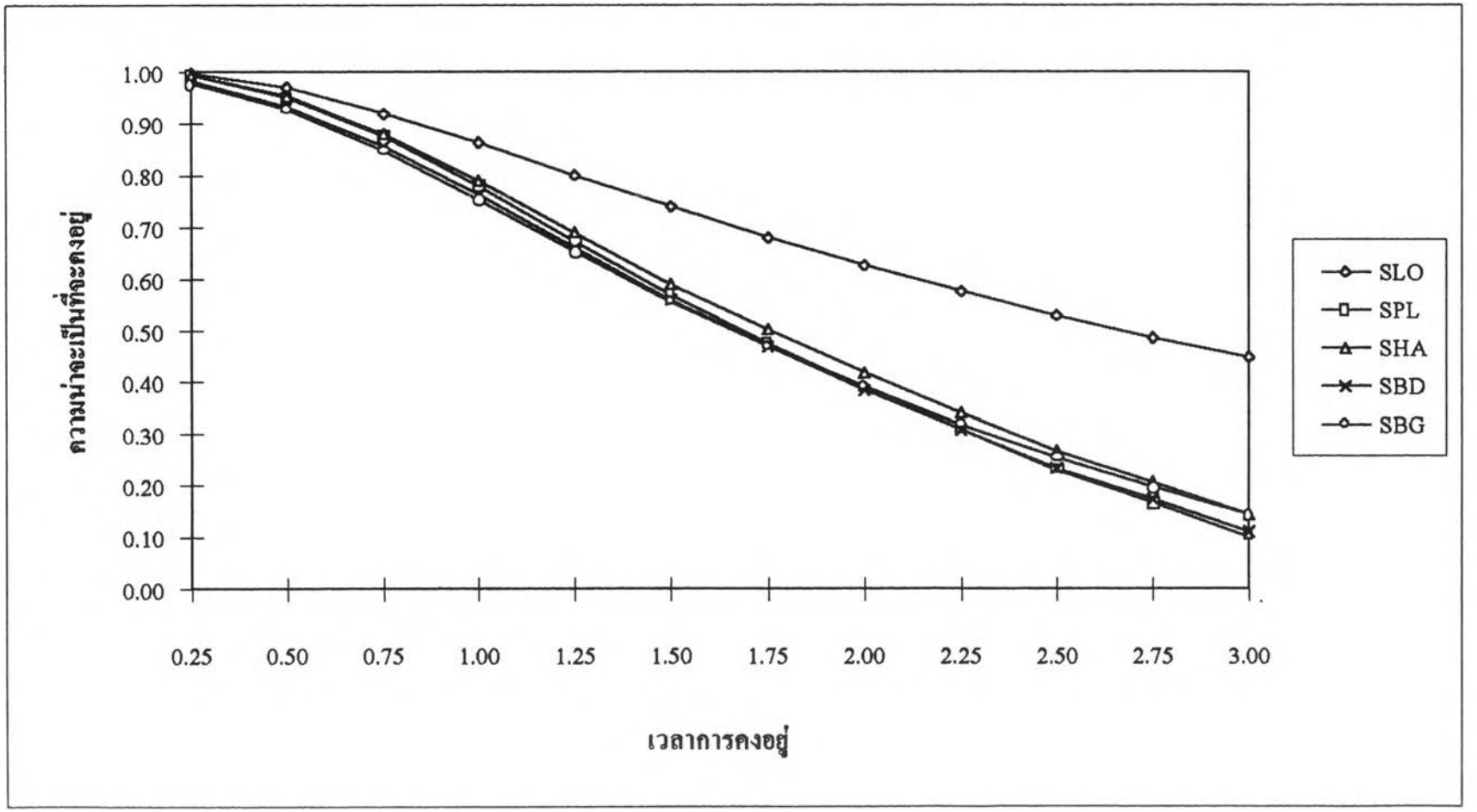
รูปที่ 8 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าน้อยกว่าค่าเฉลี่ยของการแจกแจง 75% (Tc = 1.0) เปอร์เซนต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



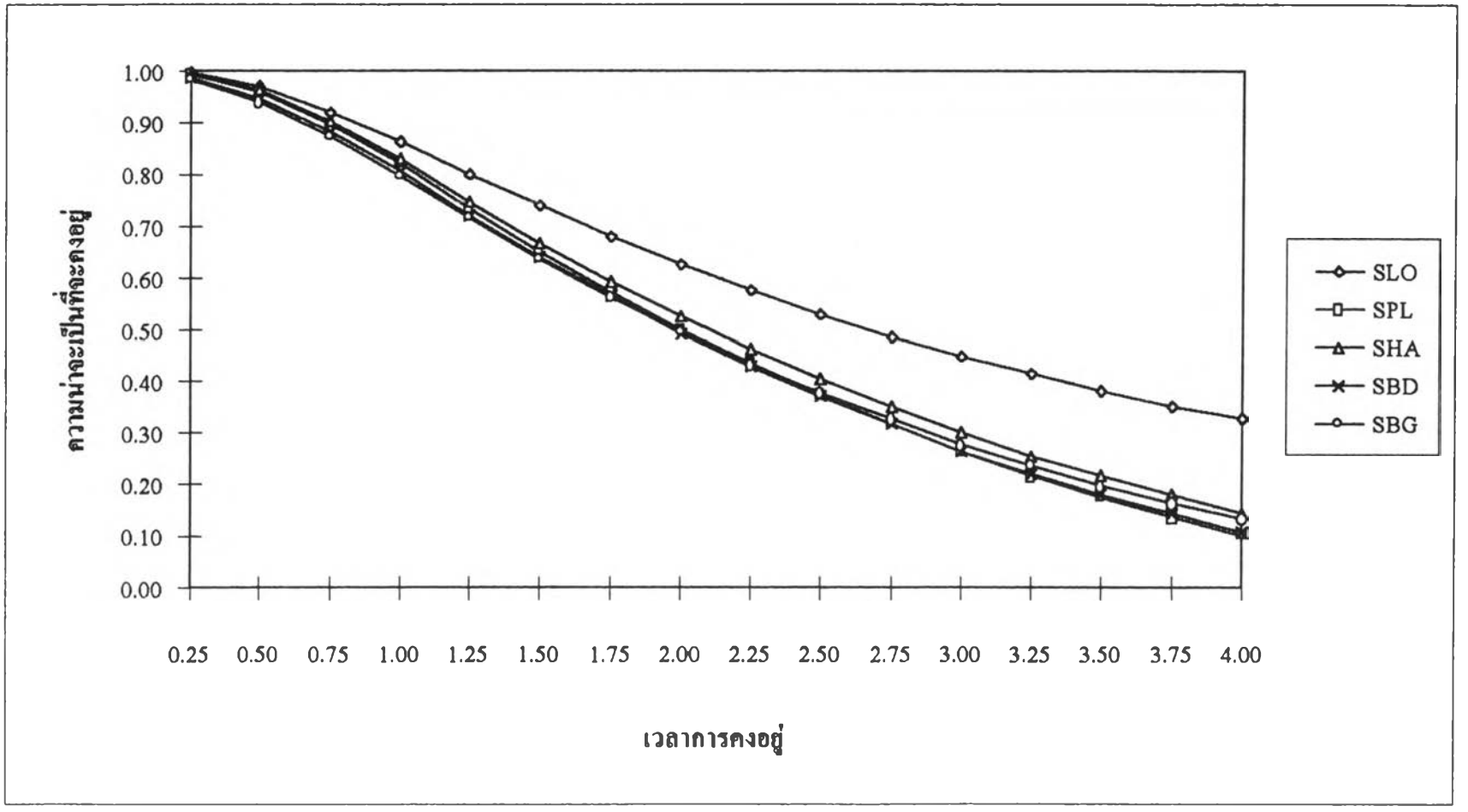
รูปที่ 9 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริง และค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าน้อยกว่าค่าเฉลี่ยของการแจกแจง 75% (Tc = 2.0) เปอร์เซ็นต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



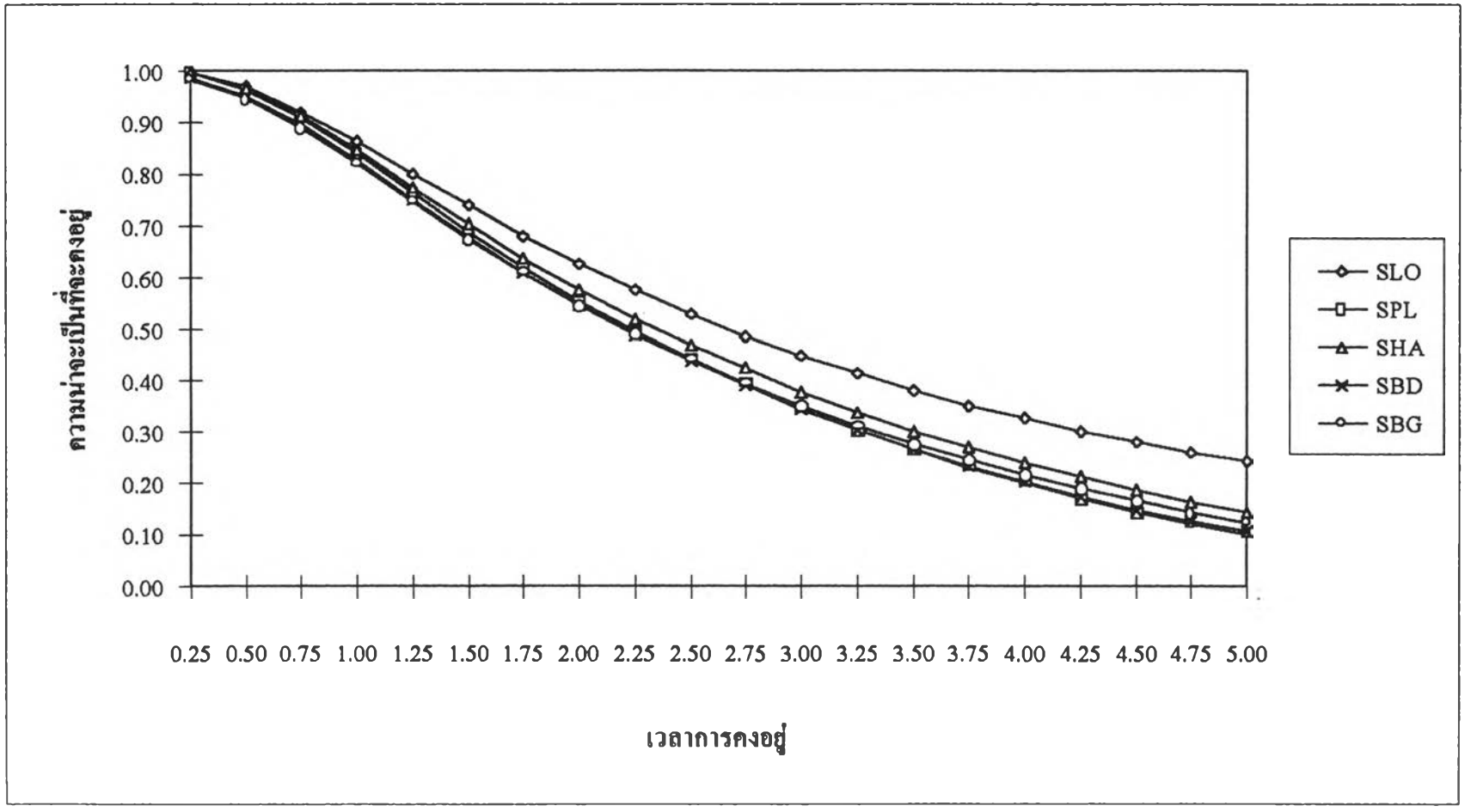
รูปที่ 10 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริงและค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าน้อยกว่าค่าเฉลี่ยของการแจกแจง 25% (Tc = 3.0) เปอร์เซ็นต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



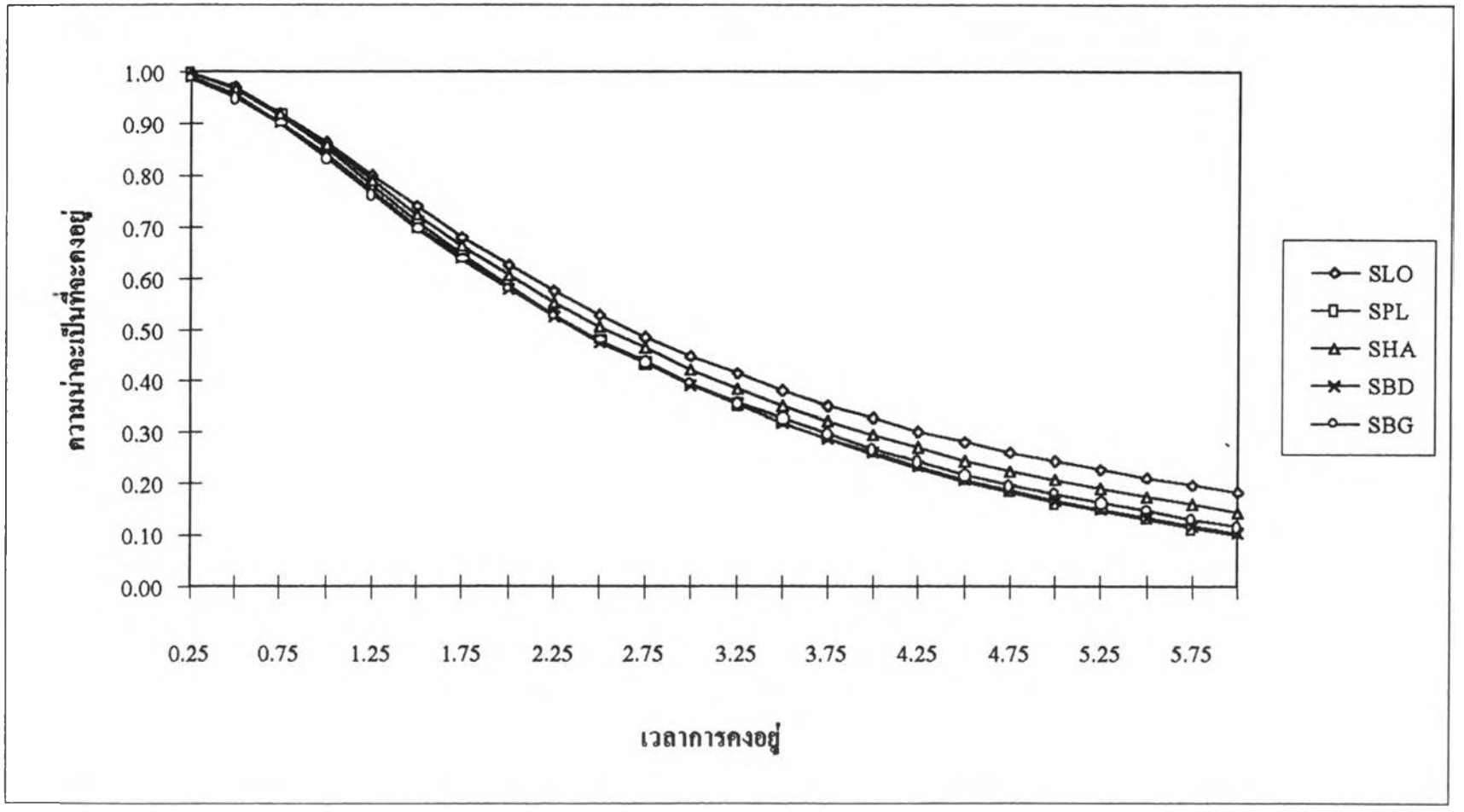
รูปที่ 11 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริงและค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่าเท่ากับค่าเฉลี่ยของการแจกแจง (Tc = 4.0) เปอร์เซ็นต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



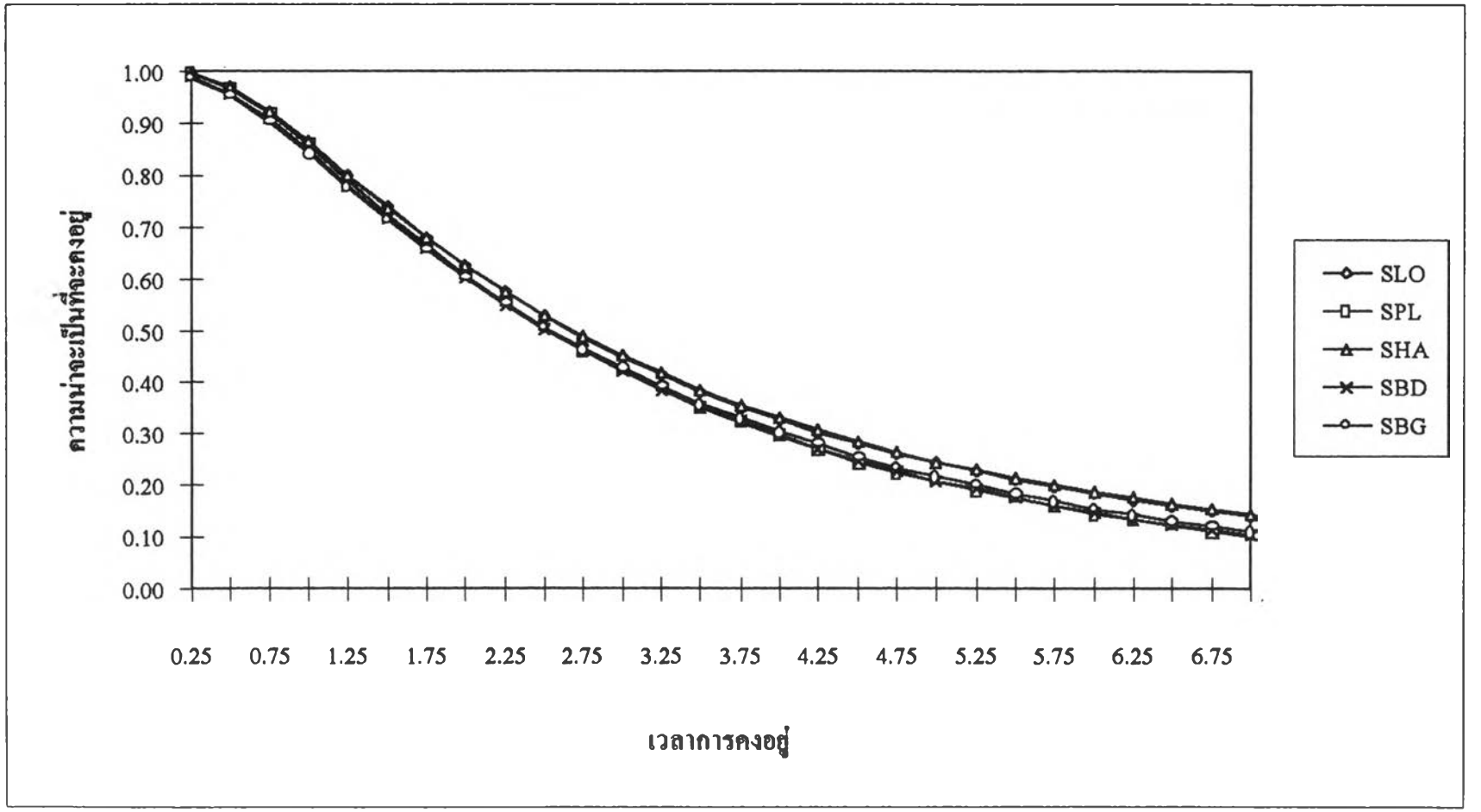
รูปที่ 12 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริงและค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่ามากกว่าค่าเฉลี่ยของการแจกแจง 25% (Tc = 5.0) เปอร์เซ็นต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



รูปที่ 13 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริงและค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่ามากกว่าค่าเฉลี่ยของการแจกแจง 50% (Tc = 6.0) เปอร์เซ็นต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



รูปที่ 14 แสดงการเปรียบเทียบค่าฟังก์ชันการคงอยู่จริงและค่าประมาณฟังก์ชันการคงอยู่ในแต่ละวิธีโดยจำแนกตามเวลาการคงอยู่ ภายใต้การแจกแจงแบบลอกนอร์มอล เมื่อเวลาที่กำหนดไว้ล่วงหน้ามีค่ามากกว่าค่าเฉลี่ยของการแจกแจง 75% (Tc = 7.0) เปอร์เซ็นต์การตัดทิ้งข้อมูลเป็น 10% และขนาดตัวอย่างเป็น 10



ภาคผนวก ข

```
C##################################################################
C       ESTIMATING SURVIVAL FUNCTION FOR RIGHT CENSOR DATA
C##################################################################
      DIMENSION       X(50),IDX(50),Y(50),REPL(50),REHAZ(50),REBAY(50),
     *               SREPL(50),SREHAZ(50),SREBAY(50),SLOG(50),SPL(50),
     *               SHAZ(50),SBAY(50),NPC(5),SBAYL(50),SREBAL(50),
     *               REBAYL(50),SUMSP(50),SUMSH(50),SUMSB(50),XN(50),
     *               SUMSBL(50),AVGSP(50),AVGSH(50),AVGSB(50),
     *               AVGSBL(50),SREBAG(50),SUMSBG(50),SBAYG(50),
     *               REBAYG(50),AVGSBG(50),TC(10),
     *               AVGPL(10,10),AVGHAZ(10,10),AVGBAY(10,10),
     *               AVGBAL(10,10),AVGBAG(10,10)
      COMMON          /SEED/IX /SELECT/KK
      INTEGER         PC
      REAL            NORMAL
      DATA            (NPC(J),J=1,4)/10,20,30,40/
      DATA            (TC(K),K=1,7)/1.00,2.00,3.00,4.00,5.00,6.00,7.00/
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C                     CONSTANT VALUE
C                     ~~~~~~~~~~~~~~
      SIGMA  = 0.9
      ET     = 4.0
      N      = 10
      MPC    = 4
      NTC    = 7
      LO     = 1000
      ALPHAR = 1.0
      TINT   = 0.25
      TSTEP  = 0.25
      IX     = 673
```

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C               HEADLINE & ITS CALCULATION
C               ~~~~~~~~~~~~~~~~~~~~~~~~~~
      ES2  = (SIGMA**2)/2
      ES   = EXP(SIGMA**2)
      DMEAN = ALOG(ET)-ES2
      T50  = EXP(DMEAN)
      ET1  = T50*EXP(ES2)
      VT   = T50*ES*(ES-1)
      WRITE(6,*)
      WRITE(6,5)
    5 FORMAT(21X,'LOGNORMAL DISTRIBUTION')
      WRITE(6,6) DMEAN,SIGMA,ET1,VT
    6 FORMAT('DMEAN = ',F7.4,2X,'SIGMA = ',F7.4,2X,'MEAN = ',F6.4,2X,
     *     'VAR = ',F10.4)
      WRITE(6,7) N,IX,ALPHAR,LO
    7 FORMAT(6X,'N = ',I2,2X,'IX = ',I4,2X,'ALPHAR = ',F4.2,2X,
     *     'LOOP = ',I4)
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C            1ST DO LOOP : FOR VARY TC POINT
C            ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      DO 991 K = 1,NTC
        NP = 0
        IX = 673
        DO 992 TK1 = TINT,TC(K),TSTEP
          NP   = NP + 1
          Y(NP) = TK1
  992   CONTINUE
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C          2ND DO LOOP : FOR VARY PERCENT OF CENSORING
C          ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        DO 200 J = 1,MPC
          PC = NPC(J)
          NC = (N*PC)/100
          NU = N-NC
```

```
      IA = 65539
      KK = 0
      DO 110 LI = 1,NP
        SREPL(LI)  = 0.0
        SREHAZ(LI) = 0.0
        SREBAY(LI) = 0.0
        SREBAL(LI) = 0.0
        SREBAG(LI) = 0.0
        SUMSP(LI)  = 0.0
        SUMSH(LI)  = 0.0
        SUMSB(LI)  = 0.0
        SUMSBL(LI) = 0.0
        SUMSBG(LI) = 0.0
  110   CONTINUE
      SUM  = 0.0
      STMED = 0.0
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C         3RD DO LOOP : FOR SIMULATE & GEN DATA
C         ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
      DO 101 L = 1,LO
        K1  = 0
        K2  = 0
        I   = 0
        SUMT = 0.0
        PDL  = 0.0
        HAZ  = 0.0
        BAY  = 0.0
        DO 30 I1 = 1,N
   60     XN(I1)  = NORMAL(DMEAN,SIGMA)
          X(I1)   = EXP(XN(I1))
          IDX(I1) = JINDEX(X(I1),TC(K))
          IF (IDX(I1) .EQ. 1) THEN
            K1 = K1 + 1
            IF (K1 .LE. NU) THEN
              GOTO 30
```

```
            ELSE
              IF (K2 .LT. NC) THEN
                GOTO 60
              ELSE
                GOTO 31
              ENDIF
            ENDIF
          ELSE
            K2 = K2 + 1
            IF (K2 .LE. NC) THEN
              GOTO 30
            ELSE
              IF (K1 .LT. NU) THEN
                GOTO 60
              ELSE
                GOTO 31
              ENDIF
            ENDIF
          ENDIF
30        CONTINUE
31        DO 32 I1 = 1,N
          IF (X(I1) .EQ. TC(K)) THEN
            I = I + 1
          ENDIF
          SUMT = SUMT + X(I1)
          SUM  = SUM + X(I1)
32        CONTINUE
        POS1  = N/2
        POS2  = POS1 + 1
        TMED  = (X(POS1) + X(POS2))/2
        STMED = STMED + TMED
        CALL RANK(N,X,IDX)
        DO 140 LI = 1,NP
          SLOG(LI) = 0.0
          XS = Y(LI)
```

```
      Z0 = (ALOG(XS/T50)/SIGMA)
      CD = CDFN(Z0)
      SLOG(LI) = 1-CD
      CALL PRODUC(N,X,IDX,XS,PDL)
      SPL(LI) = PDL
      CALL HAZARD(N,X,IDX,XS,TC(K),HAZ)
      SHAZ(LI) = HAZ
      CALL BAYE(N,NU,X,XS,TC(K),SUMT,ALPHAR,BAY)
      SBAY(LI) = BAY
      CALL BAYEL(N,NC,T50,SIGMA,SLOG(LI),X,XS,TC(K),ALPHAR,
     *          BAYL)
      SBAYL(LI) = BAYL
      CALL BAYEG(N,X,XS,TC(K),NU,SUMT,BAYG)
      SBAYG(LI) = BAYG
      SUMSP(LI)  = SUMSP(LI) + SPL(LI)
      SUMSH(LI)  = SUMSH(LI) + SHAZ(LI)
      SUMSB(LI)  = SUMSB(LI) + SBAY(LI)
      SUMSBL(LI) = SUMSBL(LI) + SBAYL(LI)
      SUMSBG(LI) = SUMSBG(LI) + SBAYG(LI)
140   CONTINUE
101   CONTINUE
      SUMPL  = 0.0
      SUMHAZ = 0.0
      SUMBAY = 0.0
      SUMBAL = 0.0
      SUMBAG = 0.0
      AMED   = STMED/LO
      TMEAN  = (SUM/(N*LO))
      XMEAN  = T50*EXP((SIGMA**2)/2)
      WRITE(6,*)
      WRITE(6,291) TMEAN,XMEAN,TC(K)
291   FORMAT(2X,'DTMEAN = ',F5.3,'  XMEAN = ',F5.3,'  TC = ',F5.3)
      WRITE(6,*)
      DO 160 LI = 1,NP
         TT = Y(LI)
```

```
      AVGSP(LI)  = SUMSP(LI)/LO
      AVGSH(LI)  = SUMSH(LI)/LO
      AVGSB(LI)  = SUMSB(LI)/LO
      AVGSBL(LI) = SUMSBL(LI)/LO
      AVGSBG(LI) = SUMSBG(LI)/LO
      CALL RELATE(SLOG(LI),AVGSP(LI),RE)
      REPL(LI) = RE
      CALL RELATE(SLOG(LI),AVGSH(LI),RE)
      REHAZ(LI) = RE
      CALL RELATE(SLOG(LI),AVGSB(LI),RE)
      REBAY(LI) = RE
      CALL RELATE(SLOG(LI),AVGSBL(LI),RE)
      REBAYL(LI) = RE
      CALL RELATE(SLOG(LI),AVGSBG(LI),RE)
      REBAYG(LI) = RE
      SUMPL    = SUMPL + REPL(LI)
      SUMHAZ   = SUMHAZ + REHAZ(LI)
      SUMBAY   = SUMBAY + REBAY(LI)
      SUMBAL   = SUMBAL + REBAYL(LI)
      SUMBAG   = SUMBAG + REBAYG(LI)
160   CONTINUE
      AVGPL(K,J) = SUMPL/NP
      AVGHAZ(K,J) = SUMHAZ/NP
      AVGBAY(K,J) = SUMBAY/NP
      AVGBAL(K,J) = SUMBAL/NP
      AVGBAG(K,J) = SUMBAG/NP
      WRITE(6,*)
      WRITE(6,392) PC
392   FORMAT(2X,'Percent of Censoring = ',I2,' %')
      WRITE(6,193)
193   FORMAT(1X,62('-'))
      WRITE(6,1161)
1161  FORMAT(2X,'T',7X,'SPL',6X,'APEPL',5X,'SHAZ',5X,'APEHAZ',5X,
     *  'SBAYD',4X,'APEBAD',5X,'SBAYG',4X,'APEBAG')
      DO 163 LI = 1,NP
```

```
        TT = Y(LI)
        WRITE(6,1163) TT,AVGSP(LI),REPL(LI),AVGSH(LI),
     *              REHAZ(LI),AVGSB(LI),REBAY(LI),
     *              AVGSBG(LI),REBAYG(LI)
1163      FORMAT(1X,F4.2,4(1X,F9.5,1X,F9.5))
 163    CONTINUE
       WRITE(6,1172)
1172     FORMAT(1X,62('-'))
       WRITE(6,1201) AVGPL(K,J),AVGHAZ(K,J),AVGBAY(K,J),
     *           AVGBAG(K,J),AVGBAL(K,J)
1201     FORMAT(1X,'MAPE',5(10X,F10.5))
   3   FORMAT(1X,62('-'))
 200   CONTINUE
 991 CONTINUE
     WRITE(6,*)
     WRITE(6,*) '          ********* SUMMARY *********'
     DO 199 K = 1,NTC
       Z01 = (ALOG(TC(K)/T50))/SIGMA
       CDD = CDFN(Z01)
       SL  = 1-CDD
       WRITE(6,*)
       WRITE(6,8) SL,TC(K)
   8   FORMAT(1X,'S(TC) = ',F7.5,2X,'TC = ',F5.2)
       WRITE(6,2)
       WRITE(6,1)
   1   FORMAT(3X,'PC',5X,'PL',7X,'HAZ',7X,'BAYD',6X,'BAYG',6X,'BAL')
   2   FORMAT(62('-'))
       DO 210 J = 1,MPC
         PC = NPC(J)
         WRITE(6,1202) PC,AVGPL(K,J),AVGHAZ(K,J),AVGBAY(K,J),
     *              AVGBAG(K,J),AVGBAL(K,J)
1202     FORMAT(3X,I2,5(1X,F9.5))
 210   CONTINUE
       WRITE(6,13)
  13   FORMAT(62('-'))
```

```
 199 CONTINUE
     STOP
     END
C#####################################################################
C              SUBROUTINE RANDOM BARIABLE
C#####################################################################
     SUBROUTINE RAND(IX,IY,YFL)
C    IY = IX*742938285
     IY = IX*65539
     IF (IY) 10,20,20
  10 IY  = IY+2147483647+1
  20 YFL = IY
     YFL = YFL/2147483647
     IX = IY
     RETURN
     END
C#####################################################################
C              GEN DATA : WEIBULL DISTRIBUTION
C#####################################################################
     FUNCTION WEIBUL(LAM,MSHA)
     COMMON   /SEED/IX
     REAL     LAM,MSHA
  10 CALL RAND(IX,IY,YFL)
     IF ((YFL .EQ. 1.0) .OR. (YFL .EQ. 0.0)) GOTO 10
     WEIBUL = (1.0/LAM)*(-1.0*(ALOG(1.0-YFL)))**(1.0/MSHA)
     RETURN
     END
C#####################################################################
C              GEN DATA : NORMAL DISTRIBUTION
C#####################################################################
     FUNCTION NORMAL(DMEAN,SIGMA)
     COMMON   /SEED/IX /SELECT/KK
     REAL     NORMAL
     PI = 3.1415926
     IF (KK .EQ. 1) GOTO 10
```

```
      CALL RAND(IX,IY,YFL)
      RONE = YFL
      CALL RAND(IX,IY,YFL)
      RTWO = YFL
      ZONE = SQRT(-2*ALOG(RONE))*COS(2*PI*RTWO)
      ZTWO = SQRT(-2*ALOG(RONE))*SIN(2*PI*RTWO)
      NORMAL = ZONE*SIGMA + DMEAN
      KK = 1
      RETURN
   10 NORMAL = ZTWO*SIGMA + DMEAN
      KK = 0
      RETURN
      END
C######################################################################
C               SUBROUTINE FOR SORTING DATA
C######################################################################
      SUBROUTINE RANK(N,X,IDX)
      INTEGER I,J,SP,MID,LEFT(20),RIGHT(20),IDX(50)
      REAL X(50),PIVOT
      LEFT(1)  = 1
      RIGHT(1) = N
      SP = 1
   10 IF (LEFT(SP) .LT. RIGHT(SP)) GOTO 20
      SP = SP-1
      GOTO 130
   20 I = LEFT(SP)
      J = RIGHT(SP)
      PIVOT = X(J)
      MID = (I+J)/2
      IF (J-I .LT. 6) GOTO 50
      IF ((PIVOT .GT. X(I)) .AND. (PIVOT .LT. X(MID))) GOTO 50
      IF ((PIVOT .LT. X(I)) .AND. (PIVOT .GT. X(MID))) GOTO 50
      IF ((X(I) .LT. X(MID)) .AND. (X(I) .GT. PIVOT)) GOTO 30
      IF ((X(I) .GT. X(MID)) .AND. (X(I) .LT. PIVOT)) GOTO 30
      CALL SWAP(X(I),X(J),IDX(I),IDX(J))
```

```
      GOTO 40
   30 CALL SWAP(X(I),X(J),IDX(I),IDX(J))
   40 PIVOT = X(J)
   50 IF (I .GE. J) GOTO 110
   60 IF (X(I) .GE. PIVOT) GOTO 70
      I = I+1
      GOTO 60
   70 J = J-1
   80 IF (.NOT.((I .LT. J) .AND. (PIVOT .LT. X(J)))) GOTO 90
      J = J-1
      GOTO 80
   90 IF (I .LT. J) CALL SWAP(X(I),X(J),IDX(I),IDX(J))
  100 GOTO 50
  110 J = RIGHT(SP)
      CALL SWAP(X(I),X(J),IDX(I),IDX(J))
      IF ((I-LEFT(SP)) .GE. (RIGHT(SP)-I)) GOTO 120
      LEFT(SP+1)  = LEFT(SP)
      RIGHT(SP+1) = I-1
      LEFT(SP)    = I+1
      GOTO 125
  120 LEFT(SP+1)  = I+1
      RIGHT(SP+1) = RIGHT(SP)
      RIGHT(SP)   = I-1
  125 SP = SP+1
  130 IF (SP .GT. 0) GOTO 10
      RETURN
      END
C=======================================================================
C              SUBROUTINE SWAP X(I) WITH X(I+1)
C=======================================================================
      SUBROUTINE SWAP(X1,X2,IDX1,IDX2)
        T  = X1
        X1 = X2
        X2 = T
        II = IDX1
```

```
      IDX1 = IDX2
      IDX2 = II
    RETURN
    END
C#####################################################################
C                FUNCTION FOR FIND INDEX
C#####################################################################
    FUNCTION JINDEX(X,TC)
      IF (X .LE. TC) THEN
        JINDEX = 1
      ELSE
        X = TC
        JINDEX = 0
      ENDIF
    RETURN
    END
C#####################################################################
C          FUNCTION FOR FIND S(X) OF WEIBULL DISTRIBUTION
C#####################################################################
    SUBROUTINE SXWEI(XS,SUW,LAM,MSHA)
    REAL LAM,MSHA
      SUW = EXP(-1.0*((LAM*XS)**MSHA))
    RETURN
    END
C#####################################################################
C            FUNCTION FOR FIND VALUE OF STANDARD NORMAL
C#####################################################################
    FUNCTION CDFN(Z0)
    TLZ0 = (1.0/2.5066282746)*EXP((-1.0*(Z0**2)/2.0))
    TLWW = 1.0/(1.0+0.33267*ABS(Z0))
    TLP  = 1.0-TLZ0*(0.4361836*TLWW-0.1201676*(TLWW**2)
   *      +0.937298*(TLWW**3))
    IF (Z0 .GE. 0) THEN
      CDFN = TLP
    ELSE
```

```
      CDFN = 1.0-TLP
    ENDIF
    RETURN
    END
C=======================================================================
C               FUNCTION FOR FIND H(J)
C=======================================================================
    FUNCTION NH(N,L)
      NH = N - L
    RETURN
    END
C=======================================================================
C               FUNCTION FOR FIND ST0
C#######################################################################
    FUNCTION ST0(T,CETA)
      ST0 = EXP((-1)*CETA*T)
    RETURN
    END
C#######################################################################
C               FUNCTION FOR FIND RT
C#######################################################################
    FUNCTION RT(T,DTAU,CETA1)
      RT = ALOG(ST0(T,CETA1))/ALOG(DTAU)
    RETURN
    END
C=======================================================================
C               FUNCTION FOR FIND YG
C#######################################################################
    FUNCTION YG(PLUS,BETA)
      IF (BETA .EQ. 1.0) THEN
        YG = ALOG((PLUS+1)/PLUS)
      ELSE
        YG = 1.0
      ENDIF
    RETURN
```

```
      END
C#####################################################################
C           SUBROUTINE FOR FIND S(X) BY PL ESTIMATOR
C#####################################################################
      SUBROUTINE PRODUC(N,X,IDX,T,PDL)
      DIMENSION  X(50),IDX(50)
      REAL       NK,NK1
        SURMU = 1.0
        DO 20 K = 1,N
          IF (X(K) .LE. T) THEN
            NK  = N-K
            NK1 = N-K+1
            SX  = (NK/NK1)**IDX(K)
            SURMU = SURMU*SX
          ELSE
            GOTO 30
          ENDIF
   20   CONTINUE
   30   PDL = SURMU
        RETURN
        END
C#####################################################################
C      SUBROUTINE FOR FIND S(X) BY HAZARD FUNCTION ESTIMATOR
C#####################################################################
      SUBROUTINE HAZARD(N,X,IDX,T,TC,HAZ)
      DIMENSION  X(50),IDX(50)
      REAL       NK1,HA
        SUMHA = 0.0
        DO 20 K = 1,N
          IF (X(K) .LE. T) THEN
            IF ((X(K) .EQ. T) .AND. (X(K) .EQ. TC)) THEN
              SUMHA = SUMHA
              GOTO 30
            ENDIF
            NK1 = N-K+1
```

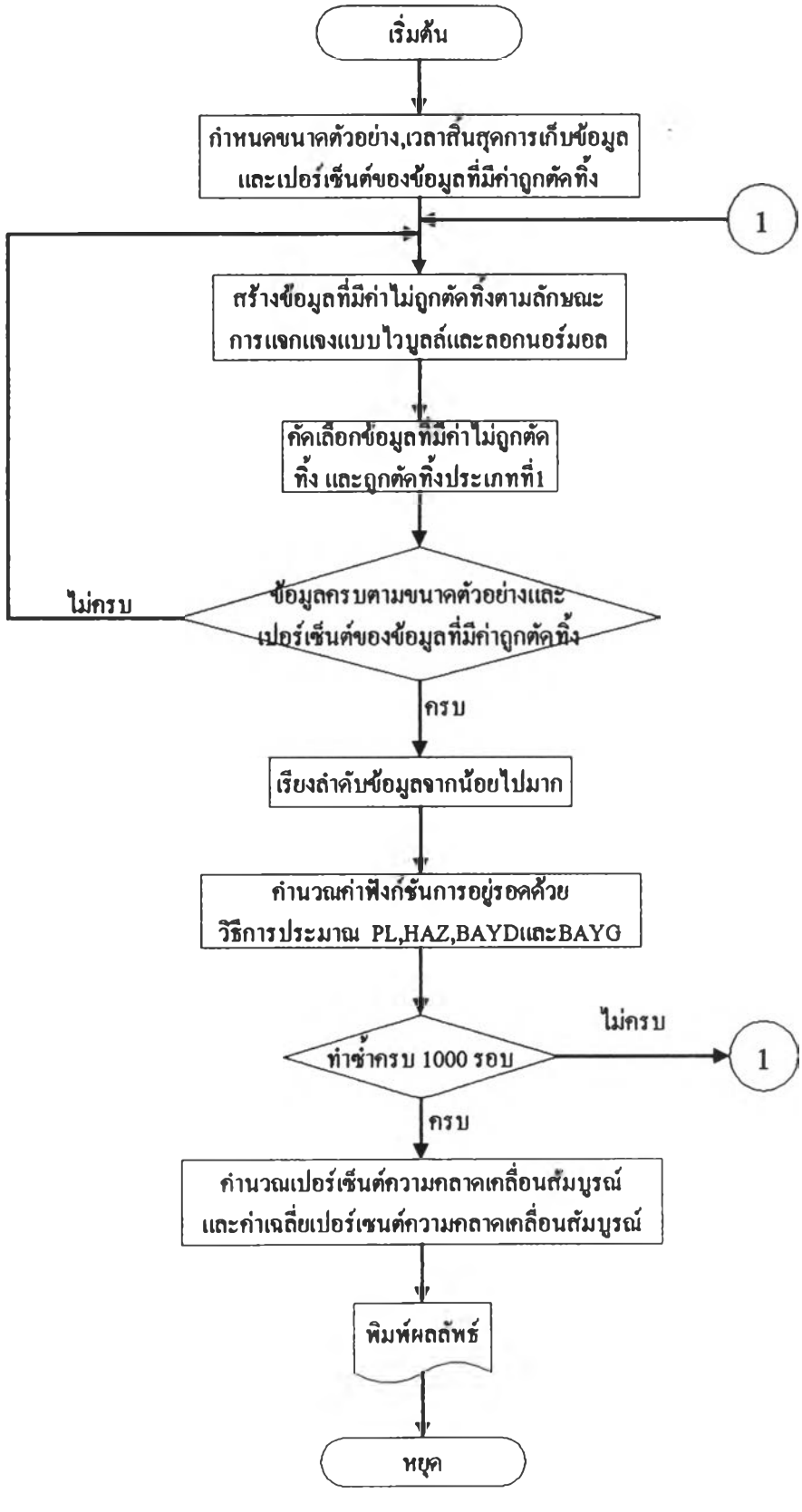
```
          HA  = IDX(K)/NK1
          SUMHA = SUMHA+HA
        ELSE
          GOTO 30
        ENDIF
   20   CONTINUE
   30   HAZ = EXP(-1*SUMHA)
        RETURN
        END
C#####################################################################
C       SUBROUTINE FOR FIND S(X) BY BAYESIAN ESTIMATOR
C#####################################################################
      SUBROUTINE BAYE(N,NU,X,T,TC,SUMT,ALPHAR,BAY)
      DIMENSION  X(50)
        J = 0
        FAC   = 1.0
        CETA  = NU/SUMT
        ALPHAT = EXP((-1)*CETA*T)
        ALPHAF = EXP((-1)*CETA*TC)
        DO 10 I = 1,N
          IF (X(I) .LE. T) THEN
            IF ((X(I) .EQ. T) .AND. (X(I) .EQ. TC)) THEN
              J  = N
              FAC = (ALPHAF+(N-NU))/ALPHAF
            ELSE
              J  = J+1
              FAC = 1.0
            ENDIF
          ELSE
            GOTO 10
          ENDIF
   10   CONTINUE
        BAY = ((ALPHAT+N-J)/(ALPHAR+N))*FAC
        RETURN
        END
```

```
C#####################################################################
C       SUBROUTINE FOR FIND S(X) BY BAYESIAN ESTIMATOR
C=====================================================================
      SUBROUTINE BAYEL(N,NC,T50,SIGMA,SLOG,X,T,TC,ALPHAR,BAYL)
      DIMENSION  X(50)
        J = 0
        FAC   = 1.0
        ZTC = (ALOG(TC/T50))/SIGMA
        CUM = CDFN(ZTC)
        ALPHAF = 1-CUM
        ALPHAT = SLOG
        DO 10 I = 1,N
          IF (X(I) .LE. T) THEN
            IF ((X(I) .EQ. T) .AND. (X(I) .EQ. TC)) THEN
              J   = N
              FAC = ((ALPHAF+NC)/ALPHAF)
            ELSE
              J   = J+1
              FAC = 1.0
            ENDIF
          ELSE
            GOTO 10
          ENDIF
   10   CONTINUE
        BAYL = ((ALPHAT+N-J)/(ALPHAR+N))*FAC
        RETURN
        END
C#####################################################################
C       SUBROUTINE FOR FIND S(X) BY GAMMA-BAYESION ESTIMATOR
C#####################################################################
      SUBROUTINE BAYEG(N,X,T,TC,NU,SUMT,BAYG)
      DIMENSION  X(50)
        J   = 0
        TAU = 1.0
        SURFAC = 1.0
```

```
      DTAU = TAU/(TAU+1)
      CETA1 = NU/SUMT
      DO 11 I = 1,N
        IF (X(I) .LE. T) THEN
          IF(X(I) .EQ. TC) THEN
            J  = NU+1
            NC = N-NU
          ELSE
            J  = J+1
          ENDIF
        ELSE
          GOTO 11
        ENDIF
   11   CONTINUE
      NU1 = NU+1
      IF (J .EQ. NU1) THEN
        NHJ = NH(N,N)
      ELSE
        NHJ = NH(N,J)
      ENDIF
      R   = RT(T,DTAU,CETA1)
      FAC1 = ((NHJ+TAU)/(NHJ+TAU+1))**R
      IF (J .EQ. 0) THEN
        SURFAC = 1.0
        GOTO 13
      ENDIF
      DO 12 K = 1,J
        KK1  = K-1
        NHKK1 = NH(N,KK1)
        IF (K .EQ. NU1) THEN
          NHK = NH(N,N)
        ELSE
          NHK = NH(N,K)
        ENDIF
        FAC21 = (NHKK1+TAU)*(NHK+TAU+1)
```

```
      FAC22 = (NHKK1+TAU+1)*(NHK+TAU)
      RXK   = RT(X(K),DTAU,CETA1)
      FAC2  = (FAC21/FAC22)**RXK
      IF (K .EQ. NU1) THEN
        NHKC  = NH(N,N)
        PLUS1 = NHKC+NC+TAU+1
        PLUS2 = NHKC+NC+TAU
        BETA  = 0.0
        F1 = YG(PLUS1,BETA)
        F2 = YG(PLUS2,BETA)
      ELSE
        NHKN  = NH(N,K)
        PLUS1 = NHKN+TAU+1
        PLUS2 = NHKN+TAU
        BETA  = 1.0
      ENDIF
      FAC3 = YG(PLUS1,BETA)/YG(PLUS2,BETA)
      FAC  = FAC2*FAC3
      SURFAC = SURFAC*FAC
   12 CONTINUE
   13 BAYG = FAC1*SURFAC
      RETURN
      END
C######################################################################
C           SUBROUTINE FOR FIND RELATIVE  ERROR
C######################################################################
    SUBROUTINE RELATE(S,ST,RE)
      DIFF = S-ST
      RE   = (ABS(DIFF)/S)*100
    RETURN
    END
```

แผนผังแสดงขั้นตอนการทำงานของโปรแกรมในการประมาณค่าฟังก์ชันการอยู่รอด



ประวัติผู้เขียน

นางสาว นันทพร อารยะสกุลวงศ์ เกิดวันที่ 27 กันยายน พ.ศ. 2514 สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิชาสถิติ จากคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์ ในปีการศึกษา 2535 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาการประกันภัย ภาควิชาสถิติ คณะพาณิชยศาสตร์และการบัญชี จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2536