AMO - Advanced Modeling and Optimization, Volume 4, Number 3, 2002

RECURSIVE ALGORITHM TO EVALUATE THE FAILURE DENSITY FUNCTION OF A K-OUT-OF –N : G SYSTEM

G. Arulmozhi, Department of Mathematics and Computer Applications,

P.S.G. College of Technology, Coimbatore-641 004, India.

Email id: arul_nithya2000@yahoo.com

ABSTRACT

An expression for probability density function (pdf) of a K – out – of - N : G system is proposed. A simple, fast and memory efficient recursive algorithm computes the pdf of K – out – of - N : G system with unequal component failure probabilities. The special cases of constant failure rate, Weibull failure time distributions are provided for illustrating the results.

Key words : K-out-of-N : G system - Probability density function - Recursive algorithm.

1. INTRODUCTION

A more general K – out – of - N and related systems, have caught the attention of many engineers and researchers because of their general nature, larger applications, high reliability and low cost than parallel and series systems. A common form of redundancy is a K – out – of - N : G system in which at least K out of N components must be good for the system to be functioning. An example of K – out – of - N : G redundancy is cables for bridge where a minimum number of cables are necessary to support the structure. Some methods by Chang Huajian et al (1995), Barlow et al (1984) and Heidtmann (1982) exits in the literature for reliability evaluation of such systems.

Previous methods of calculating the failure probability for K – out – of - N : G systems, both direct methods and recursive methods consider i.i.d components. But in the microwave stations and oil transportation examples of Chao et al (1995) all the stations (pumps) may not have the same transmission transmitting capacity. In real world, it is quite common to have telecommunication systems where some stations can communicate to the adjacent relays only, while others of a higher power are able to transmit information to more than adjacent relay. Cooksey (1975) proposes an algorithm for system reliability of a parallel system.

System pdf is an important, integral feature of planning, design and operation of all engineering systems. With the help of pdf the maximum likelihood estimators can be readily computed for a broad collection of software reliability models. Jain et al (1985) proposed a recursive algorithm for computing system reliability of a K – out – of – N system. Richard C. Bollinger (1985) gives a method for calculating the failure probability function for consecutive K – out – of - N : F systems. Richard C. Bollinger (1982) gives direct computational method for determining the failure probability for i.i.d components.

The objective of this paper is to provide exact equation for failure density of the most general K – out – of - N : G system. The equation is developed by a straightforward analysis. An efficient recursive algorithm is developed for pdf evaluation of statistically independent but not necessarily identically distributed systems. The reduction in the number of terms and multiplication is considerable compared to event space method.

The paper is organized as follows: In section 2 notations and assumptions are given and the system description and equations are given in section 3. In section 4 the proposed recursive function is presented with examples and the special cases of constant failure time, Weibull failure distribution are considered in section 5 and in the end the recursive algorithm and event space method algorithm are presented.

2. NOTATIONS AND ASSUMPTIONS

NOTATIONS

N Total number of components in the system.

K Minimum number of components that must be good for the system to be good.

t length of testing.

$f(K,N)$ pdf of the K out of N system and $f(1,1) = f_1(t)$

$R_i(t), f_i(t)$ reliability function and pdf of the i th component.

$F_i(t)$ unreliability function of the i th component.

$P(i,j)$ Pr[exactly i units are failed out of the first j units in a system consisting of N units] $(i \le j \le N)$

$$P(0,N)=\prod_{j=1}^{N} R_i(t) \quad \text{and} \quad P(N,N)=\prod_{j=1}^{N} F_i(t)$$

$X(i,j)$ Pr[exactly i units are failed out of the last j units in a system consisting of N units] $(i \le j \le N)$

λ_i constant failure rate of the i th component.

ASSUMPTIONS

a) Each unit and the system is either good or failed .

b) If N-K+1 components fail the system fails.

c) Failed components are not replaced.

d) The component failures are mutually statistically independent and not necessarily identical.

3. MATHEMATICAL MODEL

This system under investigation is a K – out - of - N: G standby system. The K out of N system is more general than purely series or parallel systems. Let N components be connected in such a way that the system fails if N-K+1 components fail. In this when an operating component fails, standby component becomes active and at least K out of N components must be good for the system to be good. It is equivalent to a (N-K+1) – out-of – N :F system. There are two main advantages of using the system. It usually has much higher reliability than the series system and is often less expensive than the parallel system.

The pdf of K- out- of-N: G system is the union of $N*\binom{N-1}{K-1}$ mutually exclusive events and each is a product of N terms. An expression is derived first for the pdf of K out of N sytem as follows

a) Probability that any one of N component fail in an interval $t_1 < t < t_2$ ($f_i(t)$ terms in $\binom{N}{1}$ ways)

b) Probability of failure of any of N-K components from N components before the start of an interval $t < t_1$ ($F_i(t)$ terms in $\binom{N-1}{K-1}$ ways) and

probability that the remaining (K-1) component survive during and after an interval $t > t_1$ ($R_i(t)$ terms).

Using a) and b) pdf of general K-out-of-N: G system is

$$f(K,N) = \sum_{l=1}^{N}\left(f_l(t)\sum_{i_1<}\sum_{i_2<}\cdots\sum_{i_{k-2}<}\sum_{i_{k-1}=1}^{N}\left(\begin{array}{l}\prod_{m=1}^{K-1} R_{i_m}(t)\\ 1<i_1<i_2<\ldots i_{K-1}<N\\ i_m \neq l\end{array}\right)\left(\prod_{\substack{j=1\\ j\neq i_1,i_2\ldots i_{k-1},l}}^{N} F_j(t)\right)\right) \quad (1)$$

The pdf of series and parallel systems that are deduced from (1) are given in (2) and (3)

$$f(N,N) = \sum_{i=1}^{N}\left(f_i(t)\prod_{j=1,\, j\neq i}^{N} R_j(t)\right) \quad (2)$$

$$f(1,N) = \sum_{i=1}^{N}\left(f_i(t)\prod_{j=1,\, j\neq i}^{N} F_j(t)\right) \quad (3)$$

If the components are statistically independent and identically distributed the pdf of the most general K-out-of-N :G system deduced from (1) is

$$f(K,N) = N * f(t) * \binom{N-1}{K-1} * (N-1) R(t)^{K-1} F(t)^{N-K} \qquad (4)$$

Equation (1) can further be written in compact form as

$$f(K,N) = \sum_{j=1}^{N-K+2} f(1,j) P(N-K+2-j, N-j)) R_{j+1}(t) + \sum_{j=2}^{N} f_j(t) \sum_{i=0}^{j-2} P(i, j-1) X(N-K-i, N-j)$$

4. ALGORITHM

To improve the computational efficiency a recursive algorithm is proposed for pdf of K out of N system given by (1). The proposed algorithm involves only ($K*(N+1-K)$-1)*3 terms and each is a product of only two terms.

With the following assumptions

$$P(K,N) = 0 \ \ and \ \ f(K,N) = 0 \ \ if \ \ K > N \ \ ; \ f(0,N) = 0 \ \ for\, N \geq 1$$

Equation (1) can be represented by a recursive function as follows

$$f(K,N) = f(K,N-1) F_N(t) + f(K-1,N-1) R_N(t) + f_N(t) P(N-K, N-1) \qquad (5)$$

The last term of equation (4) is given by a recursive function

$$P(m,n) = P(m,n-1) F_n(t) + P(m-1,n-1) R_n(t) \qquad (6)$$

Justification for (5) and (6)

Consider (6)

$$P(m,n) = P(m,n-1) F_n(t) + P(m-1,n-1) R_n(t)$$

In the above $P(m,n-1)$ represents probability of failure of exactly m units out of n-1 units with unit n always failed and $P(m-1,n-1)$ represents probability of failure of exactly m-1 units out of n-1 units with unit n always good . Each unit of the system is either good or failed and so (6) is true.

Now consider (5) and define K1=K-1 and N1=N-1

$$f(K,N) = F_N(t) f(K,N1) + R_N(t) f(K1,N1) + f_N(t) P(N1-K1,N1)$$

$$= F_N(t)\left[\sum_{j=1}^{N1-K+2} f(1,j)P(N1-K+2-j,N1-(j+1))R_{j+1} + \sum_{j=2}^{N1} f_j(t)\sum_{i=0}^{j-2} P(i,j-1)X(N1-K-i,N1-j)\right]$$
$$+ R_N(t)\left[\sum_{j=1}^{N1-K1+2} f(1,j)P(N1-K1+2-j,N1-(j+1))R_{j+1} + \sum_{j=2}^{N1} f_j(t)\sum_{i=0}^{j-2} P(i,j-1)X(N1-K1-i,N1-j)\right]$$
$$+ f_N(t)P(N1-K1,N1)$$

$$= F_N(t)\sum_{j=1}^{N1-K+2} f(1,j)P(N1-K+2-j,N1-(j+1))R_{j+1}(t) +$$
$$R_N(t)\sum_{j=1}^{N1-K+2} f(1,j)P(N1-K1+2-j,N1-(j+1))R_{j+1}(t) + R_N(t) f(1,N-K+2)P(0,K1-2)R_{N-K+1}(t)$$
$$+ F_N(t)\sum_{j=2}^{N1} f_j(t)\sum_{i=0}^{j-2} P(i,j-1)X(N1-K-i,N1-j) + R_N(t)\sum_{j=2}^{N1} f_j(t)\sum_{i=0}^{j-2} P(i,j-1)X(N1-K1-i,N1-j)$$
$$+ f_N(t)P(N1-K1,N1)$$

$$= \sum_{j=1}^{N1-K+2} f(1,j)P(N1-K1+2-j,N-(j+1))R_{j+1}(t) + f(1,N-K+2)P(0,K-2)R_{N-K+1}(t)$$
$$+ \sum_{j=2}^{N1} f_j(t)\sum_{i=0}^{j-2} P(i,j-1)X(N1-K1-i,N-j)$$
$$+ f_N(t)P(N1-K1,N1)$$

$$= \sum_{j=1}^{N-K+2} f(1,j)P(N-K+2-j,N-(j+1))R_{j+1}(t) + \sum_{j=2}^{N} f_j(t)\sum_{i=0}^{j-2} P(i,j-1)X(N-K-i,N-j)$$

ILLUSTRATIVE EXAMPLE

To illustrate equation (1) and the recursive algorithm consider the pdf of 3-out-of -6 system

$$f(3,6) \quad = \quad f_1(t)\begin{bmatrix} R_2(t)R_3(t)F_4(t)F_5(t)F_6(t) + R_2(t)R_4(t)F_3(t)F_5(t)F_6(t) + R_2(t)R_5(t)F_3(t)F_4(t)F_6(t) + \\ R_2(t)R_6(t)F_3(t)F_4(t)F_5(t) + R_3(t)R_4(t)F_2(t)F_5(t)F_6(t) + R_3(t)R_5(t)F_2(t)F_4(t)F_6(t) + \\ R_3(t)R_6(t)F_2(t)F_4(t)F_5(t) + R_4(t)R_5(t)F_3(t)F_4(t)F_6(t) + R_4(t)R_6(t)F_2(t)F_3(t)F_5(t) + \\ + R_5(t)R_6(t)F_2(t)F_3(t)F_4(t) \end{bmatrix}$$

$$..+ ....$$

$$f_6(t)\begin{bmatrix} R_1(t)R_2(t)F_3(t)F_4(t)F_5(t) + R_1(t)R_3(t)F_2(t)F_4(t)F_5(t) + R_1(t)R_4(t)F_2(t)F_3(t)F_5(t) + \\ R_1(t)R_5(t)F_2(t)F_3(t)F_5(t) + R_2(t)R_3(t)F_1(t)F_4(t)F_5(t) + R_2(t)R_4(t)F_1(t)F_3(t)F_5(t) + \\ R_2(t)R_5(t)F_1(t)F_3(t)F_4(t) + R_3(t)R_4(t)F_1(t)F_2(t)F_5(t) + R_3(t)R_5(t)F_1(t)F_2(t)F_4(t) \\ R_4(t)R_5(t)F_1(t)F_2(t)F_3(t) + \\ \end{bmatrix}$$

.

The above 3- out- of-6 : G system is the union of $6*\binom{5}{2}$ mutually exclusive events and each is a product of 6 terms. The proposed algorithm involves only 11*3 terms and each is a product of only two terms as shown below

$$f(3,6) = f(3,5)F_6(t) + f(2,5)R_6(t) + f_6(t)P(3,5)$$

$$f(3,5) = f(3,4)F_5(t) + f(2,4)R_5(t) + f_5(t)P(2,4)$$

$$f(2,5) = f(2,4)F_5(t) + f(1,4)R_4(t) + f_4(t)P(3,4)$$

$$f(3,4) = f(3,3)F_4(t) + f(2,3)R_4(t) + f_4(t)P(1,3)$$

$$f(2,4) = f(2,3)F_4(t) + f(1,3)R_4(t) + f_4(t)P(2,3)$$

$$f(1,4) = f(1,3)F_4(t) + f(0,3)R_4(t) + f_4(t)P(3,3)$$

$$f(3,3) = f(3,2)F_3(t) + f(2,2)R_3(t) + f_3(t)P(0,2)$$

$$f(2,3) = f(2,2)F_3(t) + f(1,2)R_3(t) + f_3(t)P(1,2)$$

$f(1,3) = f(1,2)F_3(t) + f(0,2)R_3(t) + f_3(t)P(2,2)$

$f(2,2) = f(2,1)F_2(t) + f(1,1)R_2(t) + f_2(t)P(0,1)$

$f(1,2) = f(1,1)F_2(t) + f(0,1)R_2(t) + f_2(t)P(1,1)$

And P(3,5) is computed by

$P(3,5) = P(3,4)F_5(t) + P(2,4)R_5(t)$

$P(3,4) = P(3,3)F_4(t) + P(2,3)R_4(t)$

$P(2,4) = P(2,3)F_4(t) + P(1,3)R_4(t)$

$P(2,3) = P(2,2)F_3(t) + P(1,2)R_3(t)$

$P(1,3) = P(1,2)F_3(t) + P(0,2)R_3(t)$

$P(1,2) = P(1,1)F_2(t) + P(0,1)R_2(t)$

To illustrate the pattern and number of terms required to evaluate pdf using this algorithm let us consider three cases

Case 1: K=N-K; Let N=6, K=3

Event space method : $6 * \binom{5}{2}$ terms

Proposed algorithm : 11*3 terms

Case 2 : K < N-K; Let N=23, K=9

Event space method : $23 * \binom{22}{8}$ terms

Proposed algorithm : K*(K+1) + (N-2*K)*K-1 = 134*3 terms

Case 3 K > N-K; Let N=25, K=15

Event space method : $25 * \binom{24}{14}$ terms

Proposed algorithm : (N-K+1)*(N-K+2)+(N-2*(K-N+1))*(N-K+1)-1 = 164*3 terms

and the number of combinations in each case is represented below

K	N
3	6
2 , 3	**5**
1 , 2, 3	4
1, 2, 3	3
1, 2	2

Example for case 1 (K=N-K)

K	N
9	23
8 ,9	22
7 ,8, 9	21
6 ,7, 8, 9	20
5 ,6, 7, 8, 9	19
4 ,5 ,6 ,7, 8 ,9	18
3 ,4, 5, 6, 7, 8, 9	17
2, 3, 4 ,5, 6, 7, 8, 9	16
1, 2, 3, 4, 5, 6, 7, 8, 9	15
1, 2, 3, 4, 5, 6, 7, 8, 9	14
1, 2, 3, 4, 5, 6, 7, 8, 9	13
1, 2, 3, 4, 5, 6, 7, 8, 9	12
1, 2, 3, 4, 5, 6, 7, 8, 9	11
1, 2, 3, 4, 5, 6, 7, 8, 9	10
1, 2, 3, 4, 5 ,6 ,7 ,8 ,9	9
1, 2, 3, 4 ,5, 6, 7, 8	8
1, 2, 3, 4, 5, 6, 7	7
1, 2, 3, 4, 5 .6	6
1, 2, 3, 4, 5	5
1, 2, 3, 4	4
1, 2, 3	3
1, 2	2

Example for case 2 (K<N-K)

K	N
15	25
14 , 15	24
13 , 14 , 15	23
12 , 13, 14, 15	22
11 , 12 , 13 , 14 , 15	21
10 , 11, 12, 1 3, 14 , 15	20
9 , 10 , 11 , 12 , 13 , 14 , 15	19
8 , 9 , 10, 11, 12, 13, 14 , 15	18
7 , 8 , 9 , 10 , 11, 12 , 13 , 14 , 15	17
6 , 7, 8, 9, 10, 11, 12, 13, 14, 15	16
5, 6 , 7 , 8, 9, 10, 11, 12, 13, 14 , 15	15
4 , 5, 6, 7, 8, 9, 10, 11, 12, 13, 14	14
3 , 4, 5, 6, 7, 8, 9, 10 , 11, 12 , 13	13
2 , 3 , 4 , 5 , 6 , 7, 8, 9, 10, 11, 12	12
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11	11
1, 2, 3, 4, 5, 6, 7, 8, 9, 10	10
1, 2, 3, 4, 5, 6, 7, 8, 9	9
1, 2, 3, 4, 5, 6, 7, 8	8
1, 2, 3, 4 , 5, 6. 7	7
1, 2, 3, 4, 5, 6	6
1, 2, 3, 4, 5	5
1, 2, 3, 4	4
1, 2, 3	3
1, 2	2

Example for case 3 (K>N-K)

5. SPECIAL CASES

In this section the results obtained in Section 3 and 4 are illustrated assuming some special cases with respect to the distribution of failure time.

5.1 Constant failure time

The exponential distribution has been widely used in life testing and reliability. So under exponential assumption, failure density function, reliability function and MTTF of some of the systems are discussed.

Consider four component systems with constant failure rates equal to $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ time units. Then from (5) we get

$$f(3,4) = (\lambda_1 + \lambda_2 + \lambda_3) Exp(-(\lambda_1 + \lambda_2 + \lambda_3)t) + (\lambda_1 + \lambda_2 + \lambda_4) Exp(-(\lambda_1 + \lambda_2 + \lambda_4)t) +$$

$$(\lambda_1 + \lambda_3 + \lambda_4) Exp(-(\lambda_1 + \lambda_3 + \lambda_4)t) + (\lambda_2 + \lambda_3 + \lambda_4) Exp(-(\lambda_2 + \lambda_3 + \lambda_4)t)$$

$$-3(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) Exp(-(\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4)t)$$

If constant failure rate and i..i distributed exponential random variables are assumed,

from (4) we get $f(t) = \begin{pmatrix} N-1 \\ K-1 \end{pmatrix} * N * \lambda * \sum_{i=0}^{N-K} \begin{pmatrix} N-K \\ i \end{pmatrix} (-1)^i Exp(-(i+k)\lambda t$ (7)

The system reliability and MTTF using (7) are

$$R(t) = \begin{pmatrix} N-1 \\ K-1 \end{pmatrix} * N * \sum_{i=0}^{N-K} \begin{pmatrix} N-K \\ i \end{pmatrix} (-1)^i \left(\frac{1}{i+k} \right) Exp(-(i+k)\lambda t \qquad (8)$$

$$MTTF == \binom{N-1}{K-1} * \left(\frac{N}{\lambda}\right) * \sum_{i=0}^{N-K} \binom{N-K}{i} (-1)^i \left(\frac{1}{i+k}\right)^2 \qquad (9)$$

Results (7), (8) and (9) coincide with the result derived by Jieyu She et al. (1992) and Grosh (1985) Pdf, reliability function and MTTF of the series and parallel systems can be easily deduced and these results.

4.2 Weibull distribution with parameters α *and* β

Next we consider Weibull distribution with scale parameter α and location parameter β.Using the recursive function pdf and reliability of the system are

$$f(t) = \binom{N-1}{K-1} * N * \alpha * \beta \sum_{i=0}^{N-K} \binom{N-K}{i} (-1)^i t^{\beta-1} Exp(-(i+k)\alpha t^{\beta}$$

$$R(t) = \binom{N-1}{K-1} * N * \sum_{i=0}^{N-K} \binom{N-K}{i} (-1)^i \left(\frac{1}{i+k}\right) Exp(-(i+k)\alpha t^{\beta}$$

Algorithms

1.Recursive method

The computational part is divided into four steps and step 1 is for input values and step 2 is common for both the cases of K < N-K and K > N-K. Step 3a and step 4a are for the case of K < N-K and step 3b and step 4b are for the case of K > N-K.

Step1

Initial stage

a) Enter the values of K, N.
b) If any particular distribution is assumed (exponential, Weibull etc) the reliability and probability density function(pdf) of that distribution is used and reliability

function R_i (t) and hence unreliability function $F_i(t)=1-R_i(t)$ and pdf $f_i(t)$ values are computed(i =1,2,3….N) for a given time period.

Step 2

```
Dofor(j = 2 to min(N-K,K) )
        i=1;  add=0, sum=0;
     Dofor (m=1 to j+1)
```
$$f(i,j) = f(i,j-1)F_j(t) + f(i-1,j-1)R_j(t) + Sub(j-i,j-1)f_j(t)$$
```
        add=add+f(i,j);  sum=sum+f(i,j)
        i++;
     endof for m
  endof for j
```

/* If $K \le N-K$

Step 3a

```
  If( min=K)
sum=sum+(N-2K)*add
```

Step 4a

```
Dofor(j = N-K+1 to N) )
        i=1
     Dofor (m=K to i)
```
$$f(i,j) = f(i,j-1)F_j(t) + f(i-1,j-1)R_j(t) + Sub(j-i,j-1)f_j(t)$$
```
         sum=sum+f(i,j)
        i++
     endof for m

endof for j
```

/* If $K \ge N-K$

Step 3b

```
If min=N-K
S =1
Dofor(j = N-K+1 to K)
        i=S
     Dofor (m=i to j)
```
$$f(i,j) = f(i,j-1)F_j(t) + f(i-1,j-1)R_j(t) + Sub(j-i,j-1)f_j(t)$$
```
         sum=sum+f(i,j)
        i++
     endof for m
     S++
endof for j
```

```
Step 4b
S=l
Dofor(j = K+1 to N )
          i=S;
       Dofor (m=i to k)
          f(i,j) = f(i,j-1)F_j(t) + f(i-1,j-1)R_j(t) + Sub(j-i,j-1)f_j(t)
          sum=sum+f(i,j)
          i++
      endof for m
       S++
endof for j
 Sub(k,n)
   Dofor(j=2 to n)
     i=1
       Dofor(k=1 to j+1)
        P(i,j) = P(i,j-1)F_j(t) + P(i-1,j-1)R_j(t)
         i++
      endof for k
  endof  for j
```

$$f(i,j) = f(i,j-1)F_j(t) + f(i-1,j-1)R_j(t) + Sub(j-i,j-1)f_j(t)$$

$$P(i,j) = P(i,j-1)F_j(t) + P(i-1,j-1)R_j(t)$$

2.E**vent space method**

It consists of 3 procedures other than the initial stage where the input data to the model are given.

Step 1

Initial stage

c) Enter the values of k, N.
d) If any particular distribution is assumed (exponential, Weibull etc) the reliability and probability density function(pdf) of that distribution is used and reliability pp[i] and pdf f [i] values are computed(i =1,2,3….N) for a given time period.
e) Assign k1 = k - 1 , n = N – 1 and r = N-k
f) For i1=1 to N do

```
{
   assign i3=1
      for i2=1 to N do
        {
           if i1!=i2
           {
             p[i3]=pp[i2] and q[i3]=1-p[i3]
               increment i3 by 1
             } /* endof if
        } /* endof for i2
```

i) Pq = 0
ii) send control to the procedure 1 (step 2).
iii) get the product of p[i] values from step 2 as rel.
iv) send control to the procedure 2 (step 3)
v) get the product of (q_i /p_i) combination values as pq.
vi) get pdf = pdf + f[i1]*rel*pq

} /* endof for i1

Step 2

Procedure 1
This procedure is used to compute the rel values using the following recurrence relation

Assgn $X[0]=1$
For m=1 to N do
$X[m]= p[m]*X[m-1] \quad m=1,2,3...N$
Reassign rel= $X[m]$ /* endof procedure 1
Goto step 1
Step 3

Procedure 2
set m=1 and c[0] =0
goto procedure 3 (step 4)
get pq from step 4 and goto step 1 /* end of procedure 2

Step 4

Procedure 3
It uses the recursive function Combination(m) to get all the required combinations and in the start of the function m=1

a) Set c[m] = c[m-1]
b) While c[m]=N-r+m

```
{
   assign pq1 = 1 and increment c[m] by 1
      test if(m<r)
         {
            go to function combination(m+1)
         }/* To call the recursive function repeatedly.
      else
         {
               for j = 1 to r do
               {
               } /* endof for j
```

$$pq1 = pq1 * \left(q_j \big/ p_j \right)$$

```
        pq = pq + pq1
      ) /* endof else
  }/* endof while
```

REFERENCES

Barlow, R.E. Heidtmann, K.D. (1984). Computing k – out – of - n system reliability, IEEE Trans. on Reliability Vol. R-33, Oct, 322 - 323.

Chang Huajian , Shi Yong Chang (1995) . A practical reliability analysis method for engineers, Reliability Engineering and system safety, 47,93-95.

Chao, M.T., Fu, J.C., Koutras, M.V. (1995). Survey of reliability studies of Consecutive - k - out – of – n : F & related systems, IEEE Trans. on Reliability Vol. 44, No. 1, March, 120 – 127.

Cooksey Jr.W., (1975). An algorithm to determine parallel system reliability , IEEE Trans. on Reliability Vol. R-24, Oct, 287.

Grosh, D.L.(1985). A Primer of Reliability theory, John Wiley & sons.

Heidtmann, K.D. (1982). Improved method of inclusion – exclusion applied to k –out –of – n sytem, IEEE Trans. on Reliability Vol. R-31, Apr, 36 - 40.

Jain, S.P. , Krishna Gopal.(1985). Recursive Algorithm for Reliability Evaluation of k – out – n : G system, IEEE Trans. on Reliability Vol. R-34, No. 2 June, 144-146.

Jieyu She, Michael .G, Pecht . (1992). Reliability of a k – out – of – n Warm Standby System, IEEE Trans. on Reliability Vol. 41, No: 1, March, 72 - 75.

Richard C. Bollinger, (1985). Strict consecutive – k – out – of – n : F systems, IEEE Trans. on Reliability Vol. 34, No.1, April, 50 - 52.

Richard C. Bollinger, (1982). Direct computation for consecutive – k – out – of – n : F systems, IEEE Trans. on Reliability Vol. 31, No.15 December, 444 - 446.