Efficient Algorithm for Reliability of a Circular Consecutive-k-out-of-n:F System

Jer-Shyan Wu  
National Chiao-Tung University, Hsinchu  
Rong-Jaye Chen, Member IEEE  
National Chiao-Tung University, Hsinchu

Key Words — Circular consecutive-k-out-of-n:F system, System reliability, Algorithm.

Reader Aids —  
Purpose: Report a new algorithm  
Special math needed for explanations: Probability theory  
Special math needed to use results: Same  
Results useful to: Reliability analysts and theoreticians

Summary & Conclusions — The time complexities of previously published algorithms for circular consecutive-k-out-of-n:F systems are $O(n \cdot k^2)$ and $O(n \cdot k)$. This paper proposes a method to improve upon the original $O(n \cdot k^2)$ algorithm and hence derives an $O(n \cdot k)$ algorithm.

1. INTRODUCTION

A consecutive-k-out-of-n:F system consists of a sequence of $n$ ordered components where each component either functions or fails. The system fails if and only if at least $k$ consecutive components fail. There are two topologies for this system: a line and a circle. The reliability analysis of such systems was first studied by Chiang & Niu [2], and later by Derman, Lieberman, Ross [3], Hwang [4], Shanthikumar [5], Antonopoulou & Papastavridis [1], and Wu & Chen [6]. Hwang [4] proposed a recursive $O(n)$ algorithm for linear consecutive-k-out-of-n:F system and an $O(n \cdot k^2)$ algorithm for circular consecutive-k-out-of-n:F system. Antonopoulou & Papastavridis [1] announced that they had found an $O(n \cdot k)$ recursive algorithm for computing the reliability of a circular system. Wu & Chen [6] also found a new $O(n \cdot k)$ algorithm for such system. This paper proposes an improvement upon the Hwang [4] $O(n \cdot k^2)$ algorithm and introduces a new $O(n \cdot k)$ algorithm.

2. MODEL

Assumptions  
1. Each component, subsystem, and system either functions or fails.  
2. All $n$ component-states are mutually $s$-independent.  
3. Components $1, 2, \ldots, n$ are arranged as to form a circle in that order.  
4. The system or subsystem fails if and only if at least $k$ consecutive components fail.

Notation

- $n$: number of components in a system
- $k$: minimum number of consecutive failed components that causes system failure
- $p_i$, $q_i$: probability that component $i$ [functions, fails]; $q_i + p_i = 1$
- $R_L(i, j)$, $R_C(i, j)$: reliability of [linear, circular] system consisting of components $i, i+1, i+2, \ldots, j$

Other, standard notation is given in "Information for Readers & Authors" at the rear of each issue.

3. IMPROVEMENT

Hwang [4] proposed a recursive $O(n)$ algorithm for linear consecutive-k-out-of-n:F systems and an $O(n \cdot k^2)$ algorithm for circular consecutive-k-out-of-n:F system. The $O(n \cdot k^2)$ algorithm is:

$$
R_C(1,n) = \sum_{s-1+n-l<k} \left\{ \prod_{i=1}^{l-1} q_i \cdot p_i \cdot R_L(s+1, l-1) \cdot p_i \right\} 
\cdot \prod_{i=l+1}^{n} q_i
$$

(1)

Now, we derive the $O(n \cdot k)$ algorithm for circular consecutive-k-out-of-n:F systems by (1):

$$
R_C(1,n) = \sum_{s=1}^{k} \left\{ \prod_{i=1}^{l-1} q_i \cdot p_i \cdot R_L(s+1, l-1) \right\}
\cdot \prod_{i=l+1}^{n} q_i
$$

\[=
\sum_{i=2}^{k+1} \sum_{j=n-k+2+i}^{n-1} \left\{ \prod_{m=1}^{i-2} q_m \cdot p_{i-1} \cdot R_L(i, j) \cdot p_{j+1} \right\}
\cdot \prod_{m=j+2}^{n} q_m \right\}. \quad (2)

In (2), we need to generate:

$$
R_L(2, n-k), \ldots, R_L(2, n-1); \\
R_L(3, n-k+1), \ldots, R_L(3, n-1); \ldots; R_L(k+1, n-1)
$$

0018-9529/92$03.00 ©1992 IEEE
The Hwang [4] recursive $O(n)$ algorithm can be expressed as:

$$R_C(1,n) = R_C(1,n-1) - R_C(1,n-k-1) \cdot p_{n-k}$$

(3)

While computing $R_C(1,n)$, we also get $R_C(1,1), R_C(1,2), \ldots, R_C(1,n-1)$; that is important. Using this property, we can evaluate: $R_C(2,n-k), \ldots, R_C(2,n-1); R_C(3,n-k+1), \ldots, R_C(3,n-1); \ldots; R_C(k+1,n-1)$ with time complexity $O(n-2) + O(n-3) + \ldots + O(n-k-1) = O(n-k)$. And we can compute $\Pi_i^{(j)} = q_j, \ldots, \Pi_i^{(j)} = q_j; \Pi_i^{(j)} = q_j; \ldots, \Pi_i^{(j)} = q_i$ in $O(k)$. Store all derived values in memory. The sum in (2) contains $\frac{1}{2}k(k+1)$ terms, so the time complexity for computing $R_C(1,n)$ is: $O(n-k) + O(k) + O(k^2) = O(n-k)$, for $n > k$.

REFERENCES


Dr. Rong-Jaye Chen; Dept. of Computer Science and Information Engineering; National Chiao-Tung University; Hsinchu 30050 TAIWAN ROC.

Rong-Jaye Chen (M’90) was born in Taiwan in 1952. He received his BS (1977) in Mathematics from National Tsing-Hua University and the PhD (1987) in Computer Science from the University of Wisconsin, Madison. Dr. Chen is an Associate Professor in the Dept. of Computer Science and Information Engineering at National Chiao-Tung University. He is a Member of IEEE. His research interests include reliability theory & algorithms, mathematical programming, and computer networks.

Jer-Shyan Wu; Dept. of Computer Science and Information Engineering; National Chiao-Tung University; Hsinchu 30050 TAIWAN ROC.

Jer-Shyan Wu was born in Taipei, Taiwan in 1967. He received his BS (1989) in Computer Science from National Taiwan University, his MS (1991) in Computer Science from National Chiao-Tung University. His research interests include reliability theory, queueing theory, and algorithms.

Manuscript TR91-195 received 1991 October 31; revised 1992 April 2.

IEEE Log Number 01246