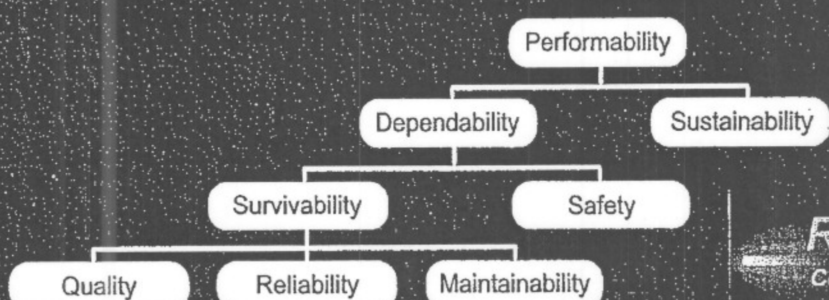
ISSN: 0973-1318

International Journal of Performability Engineering

Volume 4 - Number 1 - January 2008

Special Issue on Reliability and Quality in Design
Guest Editors: Suprasad V. Amari and Hoang Pham, U. S. A.



RAMS Consultants
www.ramsconsultants.org

International Journal of Performance Engineering Vol 4 No. 1, January 2008 pp. 19 – 29

Printed in India

A Method for Reducing the Number of Reliability Formulae in Reliability Expression of Weighted-k-out-of-n System

YOICHI HIGASHIYAMA[1*] and VENTSI RUMCHEV[2]

[1] *Department of Electrical and Electronic Engineering, Ehime University Matsuyama, 790-8577 Japan*

[2] *Department of Mathematics and Statistics, Curtin University of Technology P.O Box U1987. Perth WA 6845 Australia*

(Received on February 14. 2007)

Abstract: A weighted version of the k-out-of-n system is considered In the disjoint products version of reliability analysis of weighted-k-out-of-n systems, it is necessary to determine the order in which the weight of components is to be considered The k-out-of-n :G(F) system consists of n components; each component has its own probability and positive integer weight such that the system is operational (failed) if and only if the total weight of some operational (failure) components is at least k This paper designs a method to compute the reliability in $O(n \cdot k)$ computing time and in $O(n \cdot k)$ memory space The proposed method expresses the system reliability in fewer reliability formulae than those already published

Keywords: *weighted-k-out-of-n system, reliability equation order of components, weight of components reliability formulae used in later step*

1. Introduction

The weighted-k-out-of-n :G(F) system consists of n components, each of which has its own probability and positive integer weight (total system weight = w), such that the system is operational (failed) if and only if the total weight of some operational (failure) components is at least k [6]. The reliability of the weighted-k-out-of-n :G system is the component of the unreliability of a weighted-($w-k+1$)-out-of-n :F system. Without loss of generality, we discuss the weighted-k-out-of-n :G system only. The original k-out-of-n :G system is a special case of the weighted-k-out-of-n :G system wherein the weight of each component is 1 The system model was extended to a two-stage weighted model with components in common [7]. Recently, several different aspects of related problems were investigated [4], [5].

One of the questions that arise when using recursive disjoint products algorithms for reliability of the weighted-k-out-of-n system is the order in which the weight of components

*Corresponding author's email: mountain@dpc ehime-u ac.jp,

should be considered [3]. The system was introduced by Wu and Chen in 1994 [1] They proposed $O(n\ k)$ algorithm to compute the exact system reliability. However, their algorithm does not take any account of the order of components. The number of product terms in their reliability formula is strongly influenced by the order of components.

Higashiyama has pointed out the advantages of an alternative order in the method based on the weight of components [2]. Three types of orders are studied in [2]: (1) random order [1], (2) ascending order, and (3) descending order In ascending order, the components are arranged so that the lower weight has a lower component number That means that the component order is equivalent to the order of the weight of components in the system This order is also called best order. For example, if the weight of component i is less than the weight of component j, then the component number i must be lower than the number j The descending order is opposite of the ascending order and is also called worst order The best order method reduces the computing cost and data processing effort required to generate an optimal factored formula [2].

The method proposed in [2] dramatically reduced the computing cost and data processing effort. However, a lot of reliability formulae unused in later steps are automatically derived in the method. This paper gives an efficient algorithm to generate the reliability formulae only to be used in later steps.

Section 2 describes the notation & assumptions. Section 3 shows an $O(n\ k)$ algorithm by Wu–Chen for the reliability of the weighted–k–out–of–n:G system Section 4 shows a revised algorithm by Higashiyama to generate a factored reliability formula. Section 5 proposes a new algorithm to reduce the number of computing steps.

2. Model

Notation

n number of components in a system

k minimal total weight of all operational (failure) components which makes the system operational (failure)

w_i weight of component i

p_i operational probability of component i

q_i $\triangleq 1.0 - p_i$, failure probability of component i.

R, B, W [random, best, worst] case in which the components of the system are ordered [randomly, the lower weight one has lower number, the higher weight one has lower number].

$R_\Omega(i, j)$ reliability formula of the weighted–j–out–of–i : G_Ω for $\Omega = R, B, W$ case.

$R^N_\Omega(i, j)$ reliability formula of $R_\Omega(i, j)$ for $\Omega = R, B, W$ case in the new method which generates only the reliability formulae that are used in later steps

$M_\Omega(i, j)$ binary random value indicating the state of $R^N_\Omega(i, j)$ for $\Omega = R, B, W$.

Assumptions

A. Each component and the system has binary states, i.e , either operational or failed.

B The components and system are non-repairable.

C All components are statistically independent
D Sensing and switching mechanisms are perfect.
E Each component has a known positive integer weight.
F. Operational probability of each component is known.
G The system is operational if and only if the total weight of operational components is at least k.

3. Wu–Chen (random case) [1]

Wu and Chen [1] have presented an $O(n\ k)$ algorithm to evaluate the reliability of the weighted–k–out–of–n : G_R system.

To derive $R_R(i, j)$, the algorithm needs to construct the table with $R_R(i, j)$, for $i = 0,1,2,\ldots,n$, and $j = 0,1,2,\ldots,k$. Initially,

$$R_R(i,0) = 1.0, \quad \text{for } i = 0,1,2,\ldots,n; \tag{1}$$

$$R_R(0, j) = 0.0, \quad \text{for } j = 1,2,\ldots,k. \tag{2}$$

Furthermore, if $j < 0$, it is obvious that for any i:

$$R_R(i, j) = 1.0 \tag{3}$$

For $i = 1,2,\ldots,n$, and $j = 1,2,\ldots,k$, their algorithm generates each $R_R(i, j)$,

$$R_R(i, j) = \begin{cases} p_i \cdot R(i-1, j-w_i) + q_i \cdot R(i-1, j), & \text{if } j - w_i \geq 0; \\ p_i + q_i \ R(i-1, j), & \text{otherwise.} \end{cases} \tag{4}$$

Now the algorithm for computing $R(n,k)$ is:

1. Using equation (1) and equation (2), construct row 1 and column 1 in the $R_R(i, j)$ table.
2. Using equation (4), construct row 2, row 3, ..., row $(n+1)$ in that order. Hence, $R_R(n,k)$ is eventually derived.

Because the size of the $R_R(i, j)$ table is $(n+1)\cdot(k+1)$, the size of the sequential algorithm needs $O(n\ k)$ running time.

This method has a disadvantage in that the number of terms depends on the order of components. Hereafter it is referred to as random order method.

Consider a weighted–5–out–of–3: G_R system with weights; $w_1 = 2$, $w_2 = 6$, and $w_3 = 4$.

By equation (1), get column #1 wherein,

$$R_R(0,0) = R_R(1,0) = R_R(2,0) = R_R(3,0) = 1.0 \tag{5}$$

and by equation (2), get row #1 wherein,

$$R_R(0,1) = R_R(0,2) = R_R(0,3) = R_R(0,4) = R_R(0,5) = 0.0 \tag{6}$$

Therefore, by equation (4) rows #2, #3, and #4 are derived as follows:

Row #2;

$$\left.\begin{aligned}
R_R(1,1) &= p_1 \cdot R_R(0,-1) + q_1 \cdot R_R(0,1) = p_1 \\
R_R(1,2) &= p_1 \cdot R_R(0,0) + q_1 \cdot R_R(0,2) = p_1 \\
R_R(1,3) &= p_1 \cdot R_R(0,1) + q_1 \cdot R_R(0,3) = 0.0 \\
R_R(1,4) &= p_1 \cdot R_R(0,2) + q_1 \cdot R_R(0,4) = 0.0 \\
R_R(1,5) &= p_1 \cdot R_R(0,3) + q_1 \cdot R_R(0,5) = 0.0
\end{aligned}\right\} \quad (7)$$

Row #3;

$$\left.\begin{aligned}
R_R(2,1) &= p_2 \cdot R_R(1,-5) + q_2 \cdot R_R(1,1) = p_2 + q_2 p_1 \\
R_R(2,2) &= p_2 \cdot R_R(1,-4) + q_2 \cdot R_R(1,2) = p_2 + q_2 p_1 \\
R_R(2,3) &= p_2 \cdot R_R(1,-3) + q_2 \cdot R_R(1,3) = p_2 \\
R_R(2,4) &= p_2 \cdot R_R(1,-2) + q_2 \cdot R_R(1,4) = p_2 \\
R_R(2,5) &= p_2 \cdot R_R(1,-1) + q_2 \cdot R_R(1,5) = p_2
\end{aligned}\right\} \quad (8)$$

Row #4;

$$\left.\begin{aligned}
R_R(3,1) &= p_3 \cdot R_R(2,-3) + q_3 \cdot R_R(2,1) \\
&= p_3 + q_3 \cdot (p_2 + q_2 p_1) = p_3 + q_3 p_2 + q_3 q_2 p_1 \\
R_R(3,2) &= p_3 \cdot R_R(2,-2) + q_3 \cdot R_R(2,2) \\
&= p_3 + q_3 \cdot (p_2 + q_2 p_1) = p_3 + q_3 p_2 + q_3 q_2 p_1 \\
R_R(3,3) &= p_3 \cdot R_R(2,-1) + q_3 \cdot R_R(2,3) = p_3 + q_3 p_2 \\
R_R(3,4) &= p_3 \cdot R_R(2,0) + q_3 \cdot R_R(2,4) = p_3 + q_3 p_2 \\
R_R(3,5) &= p_3 \cdot R_R(2,1) + q_3 \cdot R_R(2,5) \\
&= p_3 \cdot (p_2 + q_2 p_1) + q_3 p_2 = p_3 p_2 + p_3 q_2 p_1 + q_3 p_2
\end{aligned}\right\} \quad (9)$$

4. Higashiyama method [2]

4.1 Best case

This section presents the best order of components so that the lower weight component has lower component number. After reordering of the components, the same procedure as in [1] can be used to compute the system reliability Hereafter it is referred to as best order method.

Therefore, consider the reliability formula for the reordered weighted–5–out–of–3: G_B system with weights; $w_1 = 2$, $w_2 = 4$, and $w_3 = 6$.

By equation (1), get column #1 wherein,

$$R_B(0,0) = R_B(1,0) = R_B(2,0) = R_B(3,0) = 1.0 \quad (10)$$

and by equation (2), get row #1 wherein,

$$R_B(0,1) = R_B(0,2) = R_B(0,3) = R_B(0,4) = R_B(0,5) = 0.0 \quad (11)$$

Therefore, by equation (4) rows #2, #3, and #4 are derived as follows:

Row #2;

$$\left.\begin{aligned} R_B(1,1) &= p_1 \cdot R_B(0,-1) + q_1 \cdot R_B(0,1) = p_1 \\ R_B(1,2) &= p_1 \cdot R_B(0,0) + q_1 \cdot R_B(0,2) = p_1 \\ R_B(1,3) &= p_1 \cdot R_B(0,1) + q_1 \cdot R_B(0,3) = 0.0 \\ R_B(1,4) &= p_1 \cdot R_B(0,2) + q_1 \cdot R_B(0,4) = 0.0 \\ R_B(1,5) &= p_1 \cdot R_B(0,3) + q_1 \cdot R_B(0,5) = 0.0 \end{aligned}\right\} \quad (12)$$

Row #3;

$$\left.\begin{aligned} R_B(2,1) &= p_2 \cdot R_B(1,-3) + q_2 \cdot R_B(1,1) = p_2 + q_2 p_1 \\ R_B(2,2) &= p_2 \cdot R_B(1,-2) + q_2 \cdot R_B(1,2) = p_2 + q_2 p_1 \\ R_B(2,3) &= p_2 \cdot R_B(1,-1) + q_2 \cdot R_B(1,3) = p_2 \\ R_B(2,4) &= p_2 \cdot R_B(1,0) + q_2 \cdot R_B(1,4) = p_2 \\ R_B(2,5) &= p_2 \cdot R_B(1,1) + q_2 \cdot R_B(1,5) = p_2 p_1 \end{aligned}\right\} \quad (13)$$

Row #4;

$$R_B(3,5) = p_3 \cdot R_B(2,-1) + q_3 \cdot R_B(2,5) = p_3 + q_3 p_2 p_1 \quad (14)$$

The final result $R_B(3,5)$ is only generated from reliabilities $R_B(2,-1)$ and $R_B(2,5)$, so it is not necessary to calculate $R_B(3,1)$, $R_B(3,2)$, ..., $R_B(3,4)$

4.2 Worst case

This section presents the worst order of components so that the higher weight one has lower component number. After reordering of the components, the same procedure as in [1] can be used to compute the system reliability Hereafter it is referred to as worst order method.

Consider the reliability formula for the reordered weighted–5–out–of–3: G_W system with weights; $w_1 = 6$, $w_2 = 4$, and $w_3 = 2$

By equation (1), get column #1 wherein,

$$R_W(0,0) = R_W(1,0) = R_W(2,0) = R_W(3,0) = 1.0 \quad (15)$$

and by equation (2), get row #1 wherein,

$$R_W(0,1) = R_W(0,2) = R_W(0,3) = R_W(0,4) = R_W(0,5) = 0.0 \quad (16)$$

Therefore, by equation (4) rows #2, #3, and #4 are derived as follows:

Row #2;

$$\left.\begin{aligned} R_W(1,1) &= p_1 \cdot R_W(0,-5) + q_1 \cdot R_W(0,1) = p_1 \\ R_W(1,2) &= p_1 \cdot R_W(0,-4) + q_1 \cdot R_W(0,2) = p_1 \\ R_W(1,3) &= p_1 \cdot R_W(0,-3) + q_1 \cdot R_W(0,3) = p_1 \\ R_W(1,4) &= p_1 \cdot R_W(0,-2) + q_1 \cdot R_W(0,4) = p_1 \\ R_W(1,5) &= p_1 \cdot R_W(0,-1) + q_1 \cdot R_W(0,5) = p_1 \end{aligned}\right\} \quad (17)$$

Row #3;

$$\left.\begin{aligned}
R_W(2,1) &= p_2 \cdot R_W(1,-3) + q_2 \cdot R_W(1,1) = p_1 + q_2 p_1 \\
R_W(2,2) &= p_2 \cdot R_W(1,-2) + q_2 \cdot R_W(1,2) = p_2 + q_2 p_1 \\
R_W(2,3) &= p_2 \cdot R_W(1,-1) + q_2 \cdot R_W(1,3) = p_2 + q_2 p_1 \\
R_W(2,4) &= p_2 \cdot R_W(1,0) + q_2 \cdot R_W(1,4) = p_2 + q_2 p_1 \\
R_W(2,5) &= p_2 \cdot R_W(1,1) + q_2 \cdot R_W(1,5) = p_2 p_1 + q_2 p_1
\end{aligned}\right\} \quad (18)$$

Row #4;

$$\begin{aligned}
R_W(3,5) &= p_3 \cdot R_W(2,3) + q_3 \cdot R_W(2,5) = p_3 \cdot (p_2 + q_2 p_1) + q_3 \cdot (p_2 p_1 + q_2 p_1) \\
&= p_3 p_2 + p_3 q_2 p_1 + q_3 p_2 p_1 + q_3 q_2 p_1 \qquad (19)
\end{aligned}$$

In the same manner to best case, the final result $R_W(3,5)$ is only generated from reliabilities $R_W(2,3)$ and $R_W(2,5)$, so it is not necessary to calculate $R_W(3,1)$, $R_W(3,2)$, ..., $R_W(3,4)$

4.3. Comparisons between three results

A. Using the component numbers in the weighted–5–out–of–3: G_B system, $R_R(3,5)$ (interchange component numbers 2 and 3) and $R_W(3,5)$ (interchange component numbers 1 and 3) can be rewritten as, respectively;

$$R_R(3,5) = p_3 p_2 + q_3 p_2 p_1 + p_3 q_2 = p_3 \cdot (p_2 + q_2) + q_3 p_2 p_1 = p_3 + q_3 p_2 p_1 = R_B(3,5) \qquad (20)$$

$$\begin{aligned}
R_W(3,5) &= p_2 p_1 + p_3 q_2 p_1 + p_3 p_2 q_1 + p_3 q_2 q_1 \\
&= q_3 p_2 p_1 + p_3 p_2 p_1 + p_3 q_2 p_1 + p_3 p_2 q_1 + p_3 q_2 q_1 \\
&= p_3 \cdot \{(p_2 + q_2) \cdot p_1 + (p_2 + q_2) \cdot q_1\} + q_3 p_2 p_1 = p_3 + q_3 p_2 p_1 = R_B(3,5) \qquad (21)
\end{aligned}$$

B. Best order method generates only 2 product terms and 4 variables, and requires 1 addition (+–operator) and 2 multiplications (×–operator)

C. Random order method generates 3 product terms and 7 variables, and requires 2 additions and 4 multiplications.

D. Worst order method generates 4 product terms and 11 variables, and requires 3 additions and 7 multiplications

5. Proposed method

The method proposed in [2] dramatically reduced the computing cost and data processing effort. However, a lot of reliability formulae unused in later step are automatically derived in the method. For Example, Best case in the section 4.1 derives $R_B(3,5)$ as a final result. The final result is only derived from three reliability formulae, $R_B(2,5)$, $R_B(1,1)$, and $R_B(1,5)$. Each of formulae without three ones are not used to generate the final result, then these formulae do not need to generate the final result. This section gives an efficient algorithm to generate the formulae only to be used in later steps.

5.1 Algorithm

The Algorithm: Generate reliability formulae only used in later steps is based on the definition of the system structure function, which is given in *Notation* of Section 2. Step 1 generates the matrix, M, (i, j) position of which corresponds to a reliability formula, $R_B(i, j)$. Each digit 1 of M means the formula to be derived. Each digit 0 of M means the formula not to be derived. The format of the algorithm makes it easy to implement in a high-level programming language like Fortran, Pascal, or C.

Algorithm: Generate reliability formulae only used in later steps

input: $n, k, w_1 \sim w_n, p_1 \sim p_n$;

common: $n, k, w_1 \sim w_n, p_1 \sim p_n, M, R; q_i = 1.0 - p_i$;

Step 1

initial clear: $M[1 \le i \le n, 1 \le j \le k] := 0$; $M[n,k] := M[n-1,k] := 1$;

if $k - w_n > 0$ then $M[n-1, k-w_n] := 1$; **end if;**

for $i := n-1$ step -1 **until** 2 **do**

for $j := 1$ **until** k **do**

if $M[i, j] = 1$ then $M[i-1, j] := 1$;

if $j - w_i > 0$ then $M[i-1, j-w_i] := 1$; **end if; end if; end for; end for;**

Step 2

initial clear: $R[0 \le i \le n, j \le 0] := 1.0$; $R[0, 1 \le j \le k] := 0.0$;

for $i := 1$ **until** n **do**

for $j := 1$ **until** k **do**

if $M[i, j] = 1$ **then** $R[i, j] := p_i \cdot R[i-1, j-w_i] + q_i \ R[i-1, j]$;

end if; end for; end for;

Return

5.2 Examples

Consider the weighted-5-out-of-3: G system with weights; $w_1 = 2$, $w_2 = 6$, and $w_3 = 4$. For each case (R, B, W), the Algorithm generates the reliability formulae below for each case about the example system. The proposed method only derives the reliability formulae to get the final result, then each of formula numbers corresponds to the formula number to be added in the section 3 and 4

5.2.1 Random case

After executing of Step 1 in the Algorithm, the matrix, $M_R[\,]$, is;

$$M_R = \begin{bmatrix} 10001 \\ 10001 \\ 00001 \end{bmatrix}$$

By virtue of M_R, Step 2 generates the reliability formulae as follows;

$R_R^N(1,1) = p_1 \cdot R_R^N(0,-1) + q_1 \cdot R_R^N(0,1) = p_1$ — 1st row in equation (7)

$R_R^N(1,5) = p_1 \cdot R_R^N(0,3) + q_1 \cdot R_R^N(0,5) = 0.0$ — 5th row in equation (7)

$R_R^N(2,1) = p_2 \cdot R_R^N(1,-5) + q_2 \cdot R_R^N(1,1) = p_2 + q_2 p_1$ — 1st row in equation (8)

$R_R^N(2,5) = p_2 \cdot R_R^N(1,-1) + q_2 \cdot R_R^N(1,5) = p_2$ — 5th row in equation (8)

Finally the algorithm derives the final result as follows;

$R_R^N(3,5) = p_3 \cdot R_R^N(2,1) + q_3 \cdot R_R^N(2,5) = p_3 \cdot (p_2 + q_2 p_1) + q_3 p_2 = p_3 p_2 + p_3 q_2 p_1 + q_3 p_2$

5th row in equation (9)

5.2.2 Best case

After executing of Step 1, the matrix is;

$$M_B = \begin{bmatrix} 10001 \\ 00001 \\ 00001 \end{bmatrix}$$

By virtue of M_B, Step 2 derives the formulae as follows;

$R_B^N(1,1) = p_1 \cdot R_B^N(0,-1) + q_1 \cdot R_B^N(0,1) = p_1$ — 1st row in equation (12)

$R_B^N(1,5) = p_1 \cdot R_B^N(0,3) + q_1 \cdot R_B^N(0,5) = 0.0$ — 5th row in equation (12)

$R_B^N(2,5) = p_2 \cdot R_B^N(1,1) + q_2 \cdot R_B^N(1,5) = p_2 p_1$ — 5th row in equation (13)

$R_B^N(3,5) = p_3 \cdot R_B^N(2,-1) + q_3 \cdot R_B^N(2,5) = p_3 + q_3 p_2 p_1$ — (14)

5.2.3 Worst case

The M_W and R_W^N are derived as follows;

$$M_W = \begin{bmatrix} 10101 \\ 00101 \\ 00001 \end{bmatrix}$$

$R_W^N(1,1) = p_1 \cdot R_W(0,-5) + q_1 \cdot R_W(0,1) = p_1$ — 1st row in equation (17)

$R_W^N(1,3) = p_1 \cdot R_W^N(0,-3) + q_1 \cdot R_W^N(0,3) = p_1$ — 3rd row in equation (17)

$R_W^N(1,5) = p_1 \cdot R_W^N(0,-1) + q_1 \cdot R_W^N(0,5) = p_1$ — 5th row in equation (17)

$R_W^N(2,3) = p_2 \cdot R_W^N(1,-1) + q_2 \cdot R_W^N(1,3) = p_2 + q_2 p_1$ — 3rd row in equation (18)

$R_W^N(2,5) = p_2 \cdot R_W^N(1,1) + q_2 \cdot R_W^N(1,5) = p_2 p_1 + q_2 p_1$ — 5th row in equation (18)

$R_W^N(3,5) = p_3 \cdot R_W^N(2,3) + q_3 \cdot R_W^N(2,5) = p_3 \cdot (p_2 + q_2 p_1) + q_3 \cdot (p_2 p_1 + q_2 p_1)$

$= p_3 p_2 + p_3 q_2 p_1 + q_3 p_2 p_1 + q_3 q_2 p_1$ — (19)

5.2.3 Comparisons

The proposed algorithm can generate three types of the final reliability formula above, $R_R^N(3,5)$, $R_B^N(3,5)$, or $R_W^N(3,5)$, for each case.

A. For the random case, the proposed algorithm needs 5 reliability formulae to get the final reliability formula and 6 reliability formulae are omitted.

B. For the best case, the proposed algorithm needs 4 reliability formulae to get the final reliability formula and 7 reliability formulae are omitted.

C. For the worst case, the proposed algorithm needs 6 reliability formulae to get the final reliability formula and 5 reliability formulae are omitted

5.3 Large Examples

Given n and k, both methods proposed [1] and [2] need basically to calculate $(n \times k)$ reliability formulae to get the final result. The proposed algorithm is hard to estimate the number of the reliability formulae, $R(i, j)$, to get the final result The execution time of the proposed algorithm strongly depends on the number of reliability formulae In this section, we present sample numerical results, Table 1, 2 and 3, obtained by applying the Step 1 in the Algorithm to some examples In table 1 and 2, α (β) denotes the number of the reliability formulae to be used (unused) to get the final result

Table 1: The number of used (unused) reliability formulae ($n = 20$)

k	α			β		
	Random	Best	Worst	Random	Best	Worst
30	427	236	507	173	364	93
40	597	396	687	203	404	113
50	767	576	867	233	424	133

Each system in Table 1 and 2 has same parameters, $n = 20$, and $k = 30, 40, \text{and } 50$, however each weight of components is not equal to each other. The number of formulae depends on the value of k and the weight given on the each component.

Table 2: The number of used (unused) reliability formulae ($n = 20$)

k	α			β		
	Random	Best	Worst	Random	Best	Worst
30	423	252	453	177	348	147
40	419	278	604	381	522	196
50	585	334	721	415	666	275

Table 3: The number of used (unused) reliability formulae ($n = 40$)

k	α			β		
	Random	Best	Worst	Random	Best	Worst
50	1756	968	1865	244	1032	135
60	1905	1027	2245	495	1379	155
70	2490	1285	2587	310	1515	213

Table 1, 2, and 3 denote that the number of the used reliability formulae in the best case is dramatically reduced as compared with the worst case. Especially, α of the best case in Table 3 is nearly equal to a half of α of the worst case.

6. Conclusions

In the disjoint products version of the reliability analysis of weighted–k–out– of–n systems, it is necessary to determine the order in which the weight of components is to be considered. This paper designs a method to compute the reliability in $O(n \cdot k)$ computing time and in $O(n \cdot k)$ memory space. The proposed method expresses the system reliability in fewer reliability formulae than those already published.

Acknowledgment

We are grateful to editor, Suprasad Amari, and an anonymous referee for constructive comments and suggestions which have significantly improved this paper.

References

[1] J.S. Wu and R.J. Chen, *An algorithm for computing the reliability of weighted–k–out–of–n systems, IEEE Transactions on Reliability*, Vol.43, no.2, pp327–329, 1990.

[2] Y. Higashiyama, "A factored reliability formula for weighted–k–out–of–n system", *Asia–Pacific Journal of Operational Research*, Vol.18, pp61–66, 2001.

[3] J.A. Buzacott, "The ordering of terms in cut– based recursive disjoint products", *IEEE Transactions on Reliability*, Vol.R–32, no. 5, pp472– 474, 1983

[4] G. Levitin and A.Lisnianski, Reliability *optimization for weighted voting system, Reliability Engineering and System Safety*, vol. 71, no. 2, pp.131-138, 2001

[5] M. Xie and H. Pham, *Modeling the reliability of threshold weighted voting systems, Reliability Engineering and System Safety*, vol. 87, no. 1, pp.53-63, 2005

[6] W. Kuo and M. J. Zuo, Optimal *Reliability Modeling*, Chapter 7, John Wiley & Son, pp.258-264, 2003.

[7] Y. Chen and Q. Yang, *Reliability of two-stage weighted-k-out-of-n systems with components in common, IEEE Transactions on Reliability*, vol. 54, no. 3, pp.431-440, 2005

Yoichi Higashiyama received his B.E. and M.E. degrees in Electrical Engineering both from Ehime University (Matsuyama, Japan) in 1971 and 1973, respectively, and his Dr. Eng. degree in Electronic Engineering from Osaka University (Osaka, Japan) in 1993. In 2003 he was a visiting professor at the Department of Mathematics and Statistics, Curtin University of Technology (Perth, Australia). Dr Higashiyama is now an Associate Professor of Circuit System Engineering in the Department of Electrical and Electronic Engineering, Ehime University. His research interests include applications of graph theory, reliability of communication systems, and VLSI design automations.

Ventsi Rumchev received his M.Sc. degree in Electrical and Electronic Engineering and his Ph.D. degree in Applied Mathematics both from St. Petersburg Technical University (Saint Petersburg, Russia), and his D.Sc. degree from the Technical University of Sofia, Bulgaria. He worked for the Faculty of Applied Mathematics and Informatics in the Technical University of Sofia for a number of years

Dr. Rumchev has held Visiting Professor positions with Coventry University and Brunel University, and Visiting Research Fellow positions with Imperial College and Coventry University, all in the UK.

In 1995 Rumchev joined the School of Mathematics and Statistics in Curtin University of Technology, Perth, Australia, and is now a Senior Lecture in Department of Mathematics and Statistics.

Dr Rumchev is an author and co-author of 6 books and monographs and has published over 160 research papers in archived journals and conference proceedings. He has an extensive consultancy in mathematical modeling, optimization-based modeling and design, applied system analysis, decision-support systems for manpower planning, and others. His current research interests include positive system theory and its applications, constrained linear systems, and the applied aspects of optimization and optimal control.