New algorithm for computing exact reliability formula of weighted–k–out–of–n system using SDP method

Yoichi Higashiyama[1], Cai Xu[1] and Ventsi Rumchev[2]

[1] Department of Electrical and Electronic Engineering
Ehime University
Matsuyama, 790-8577 Japan
e-mails: mountain@dpc.ehime–u.ac.jp (Y-H)
Cai.Xu@akitsu.ee.ehime-u.ac.jp (C-X)

[2] Department of Mathematics and Statistics
Curtin University of Technology
P.O. Box U1987, Perth, WA 6845 Australia
e-mail: rumchevv@curtin.edu.au

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 new method to compute the reliability formulas using SDP method. The proposed method expresses the system reliability in fewer reliability formulas than those already published.

Keywords: Weighted–k–out–of–n system; Reliability formulas; SDP method; Weight of components.

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 should be considered [3]. The system was introduced by Wu and Chen in 1994 [1]. They proposed $O(n\cdot k)$ 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 formulas unused in later steps are automatically derived in the method. This paper gives an efficient algorithm to generate the reliability formulas only to be used in later steps.

A relatively new but popular method for obtaining a reliability formula for coherent system is the Abraham SDP method [9]. This paper aims to apply the SDP method to the weighted–k–out–of–n :G systems. The algorithm described in this paper gives the disjoint product terms.

Section 2 describes the notation & assumptions. Section 3 shows an $O(n \cdot k)$ 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 gives Higashiyama's recent algorithm to reduce the number of computing steps. Section 6 proposes new algorithm using SDP method to derive the reliability formula.

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$_\Omega$ 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 formulas 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 $O(n \cdot k)$ 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,

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

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

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 $R_R(i,j)$,

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

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 $O(n \cdot k)$ 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;

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

Row #3;

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

Row #4;

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

4. Higashiyama method -1[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;

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

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 (interchange component numbers 1 and 3) can be rewritten as, respectively;

$$\begin{aligned}
R_R(3,5) &= p_3 p_2 + q_3 p_2 p_1 + p_3 q_2 = p_3 \cdot (p_2 + q_2) + q_3 p_2 p_1 \\
&= p_3 + q_3 p_2 p_1 = R_B(3,5)
\end{aligned} \quad (20)$$

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

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. Higashiyama method-2 [8]

The method proposed in [2] dramatically reduced the computing cost and data processing effort. However, a lot of reliability formulas 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 formulas, $R_B(2,5)$, $R_B(1,1)$, and $R_B(1,5)$. Each of formulas without three ones are not used to generate the final result, then these formulas do not need to generate the final result. This section gives an efficient algorithm to generate the formulas only to be used in later steps.

5.1 Algorithm

The **Algorithm: Generate reliability formulas 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 formulas 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 $R[i,j] := p_i \cdot R[i-1, j-w_i] + q_i \cdot R[i-1, j];$

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 formulas below for each case about the example system. The proposed method only derives the reliability formulas 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, $M_R[\cdot]$, is;

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

By virtue of M_R, **Step 2** generates the reliability formulas 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;

$$\begin{aligned} R_R^N(3,5) &= p_3 \cdot R_R^N(2,1) + q_3 \cdot R_R^N(2,5) \\ &= p_3 \cdot (p_2 + q_2 p_1) + q_3 p_2 = p_3 p_2 + p_3 q_2 p_1 + q_3 p_2 \end{aligned}$$

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 formulas 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)

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

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)

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

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 formulas to get the final reliability formula and 6 reliability formulas are omitted.

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

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

6 New algorithm using SDP method

6.1 SDP method

In the reliability problem of weighted– k –out– of– n :G(F) system, the system is operational (failed) if the total weight of some operational components is at least k . Therefore, the reliability formula of weighted system would be calculated if all the minimal sets were evaluated. Let A_i denote the product of Boolean variables corresponding to $i-\text{th}$ minimal set. Then the reliability formula is derived as:

$$F = A_1 + A_2 + \ldots + A_m \tag{20}$$

where m is the number of minimal sets. If the minimal set denotes the operational set, A_i is the product of Boolean variables, x 's, corresponding to $i-\text{th}$ operational set: otherwise, A_i is the product of Boolean variables, $\bar{x}$'s, corresponding to $i-\text{th}$ failed set.

If the terms of the reliability formula are disjoint, then the reliability formula and the numerical formula are one–to–one identical with one another. As is well known, some of A_i ($i = 1, 2, \ldots, m$) in equation (19) are not disjoint each other. This means that if one substitutes the numerical values into the reliability formula, the system reliability can not be computed.

To make A_i 's disjoint, F is transformed as:

$$F = A_1 + \bar{A}_1 A_2 + \bar{A}_1 \bar{A}_2 A_3 + \ldots + \bar{A}_1 \bar{A}_2 \ldots \bar{A}_j \ldots \bar{A}_{i-1} A_i + \ldots + \bar{A}_1 \bar{A}_2 \ldots \bar{A}_{m-1} A_m \tag{21}$$

Consider the following $i-\text{th}$ function F_i in equation (21):

$$F_i = \bar{A}_1 \bar{A}_2 \ldots \bar{A}_j \ldots \bar{A}_{i-1} A_i \, . \tag{22}$$

Let B_{ji} be the set of variables which exist in A_j and which do not exist in A_i , and A_{ji} be the product of x_k 's, $x_k \in B_{ji}$. Then equation (22) can be rewritten as:

$$F_i = \bar{A}_{1i} \bar{A}_{2i} \ldots \bar{A}_{ji} \ldots \bar{A}_{i-1,i} A_i \tag{23}$$

Furthermore, if any two product terms A_{ki} , A_{ji} in equation (23) is satisfied with $B_{ki} \subset B_{ji}$, then $\bar{A}_{ji}$ is dropped from equation (23). Let A'_{ji} ($j = 1, 2, \ldots, b$) be the un–dropped product terms. Then F_i can be rewritten as:

$$F_i = \bar{A}'_{1i} \bar{A}'_{2i} \ldots \bar{A}_{bi} A_i \tag{24}$$

Let A has fixed 2–valued indicators $x_1 x_2 \ldots x_k$, then

$$\bar{A} = \bar{x}_1 + x_1 \bar{x}_2 + x_1 x_2 \bar{x}_3 + \ldots + x_1 x_2 \ldots x_{k-1} \bar{x}_k \tag{25}$$

Referring to the formula of equation (25), equation (24) is transformed into sum of disjoint terms.

6.2 Basic idea

In this section we will briefly expose, by means of an example, the basic idea in the method to be proposed.

Consider the random case weighted–5–out–of–3 system with weights $w_1 = 2$, $w_2 = 6$, $w_3 = 4$. Let S_i be a minimal set of weighted components whose total weight is more than or equal to k . There are 2 sets, which are ordered so that the smallest one are first, that is, in order of the number of elements in the events.

$S_1 = \{w_2\}$

$S_2 = \{w_1, w_3\}$

Let x_i be Boolean variable of $i-\text{th}$ component, w_i , in the weighted– k –out– of– n system. The product of Boolean variables, A_i , corresponds to the $i-\text{th}$ set, S_i , as follows:

$A_1 = x_2$

$A_2 = x_1 x_3$

Then the reliability formula, F , is derived as:

$$F = A_1 + A_2 = x_2 + x_1 x_3 \tag{26}$$

And using equations (21)~(25), the reliability formula, F , is transformed by Boolean algebra as flows:

$$F = A_1 + \bar{A}_1 A_2 = x_2 + \bar{x}_2 x_1 x_3 \tag{27}$$

The number of indicators and terms in equation (27) is equal to the number of probabilities and terms in equation (14). And the final result is independent of the order of weighted components.

Next we consider another example, weighted–7–out– of–4 system with weights, $w_1 = 2$, $w_2 = 3$, $w_3 = 4$, $w_4 = 5$. There are 4 minimal sets as follows:

$S_1 = \{w_2, w_3\}$

$S_2 = \{w_1, w_5\}$

$S_3 = \{w_2, w_5\}$

$S_4 = \{w_3, w_5\}$

Then the product of Boolean variables are correspond to minimal sets:

$A_1 = x_2 x_3$

$A_2 = x_1 x_5$

$A_3 = x_2 x_5$

$A_4 = x_3 x_5$

The reliability formula, F , is derived as:

$$F = A_1 + A_2 + A_3 + A_4 = A_1 + \bar{A}_1 A_2 + \bar{A}_1 \bar{A}_2 A_3 + \bar{A}_1 \bar{A}_2 \bar{A}_3 A_4 = F_1 + F_2 + F_3 + F_4 \tag{28}$$

Consider F_2, F_3, and F_4 in equation (28):

$$F_2 = (\overline{x_1 x_5}) \cdot x_2 x_5 = \bar{x}_1 \cdot x_2 x_5 \tag{29}$$

$$F_3 = (\overline{x_1 x_5})(\overline{x_2 x_5}) \cdot x_3 x_5 = \bar{x}_1 \bar{x}_2 \cdot x_3 x_5 \tag{30}$$

$$F_3 = (\overline{x_1 x_5})(\overline{x_2 x_5})(\overline{x_3 x_5}) \cdot x_2 x_3 = \bar{x}_5 \cdot x_2 x_3 \tag{31}$$

Therefore, the final result, F is

$$\begin{aligned} F &= A_1 + A_2 + A_3 + A_4 = A_1 + F_2 + F_3 + F_4 \\ &= x_1x_5 + \bar{x}_1x_2x_5 + \bar{x}_1\bar{x}_2x_3x_5 + \bar{x}_5x_2x_3 \end{aligned} \tag{32}$$

7. Conclusions

In the old version of the reliability analysis of weighted– k –out– of– n systems, it was necessary to determine the order in which the weight of components is to be considered. This paper designs an algorithm using SDP method to compute the reliability formula. The proposed method expresses the system reliability in fewer reliability formulas than those already published. The number of disjoint product terms in the reliability formula is independent of the order of weighted components.

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.

[8] Y. Higashiyama and V. Rumchev, "A method for reducing the number of reliability fprmulae in reliability expression of weighted-k-out-of-n system", *International Journal of Performance Engineering*, vol. 4, no. 1, pp.19-29, 2008.

[9] J. Abraham, "An improved algorithm for network reliability", *IEEE Transactions on Reliability* , vol. R-28, no. 1, pp.58-61, 1979.