Abstract: A k-to-l-out-of-n: G system is one in which neither less than k nor more than l units out of n units are to be good for the system to be good. It's a very general system; as when l = n, it is a k-out-of-n: G system; when l = k = n, it is a series system; when l = n and k = 1, it is a parallel system; and when l = k, a system in which exactly k successes are required for the system to be successful. An efficient method of computing the reliability of such systems is presented. The efficiency results because of appropriately combining the terms in the expanded form of the reliability expression. Recursive relations are then developed. This principle is applied to computing the all-terminal (global) reliability also. Here too, it results in least possible computations in finding the reliability.

Key-Words: k-out-of-n: G system reliability, k-to-l-out-of-n: G system reliability, all-terminal reliability.

1. Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>Edge or unit reliability</td>
</tr>
<tr>
<td>Q</td>
<td>Edge or unit unreliability = 1-P</td>
</tr>
<tr>
<td>F(i,n)</td>
<td>Probability that exactly i units out of n units are in failed state</td>
</tr>
<tr>
<td>S(i,n)</td>
<td>Probability that exactly i units out of n units are in good state</td>
</tr>
<tr>
<td>ne</td>
<td>Number of edges</td>
</tr>
<tr>
<td>nn</td>
<td>Number of nodes</td>
</tr>
<tr>
<td>e</td>
<td>Edge</td>
</tr>
<tr>
<td>Pr</td>
<td>Probability</td>
</tr>
<tr>
<td>R</td>
<td>Reliability</td>
</tr>
<tr>
<td>(□)</td>
<td>Failed state</td>
</tr>
</tbody>
</table>

2. Introduction

A ‘k-out-of-n:G’ system, denoted as (k,n) system is a system in which at least k of its n (all) units must function properly for the system to be successful. The examples are- a suspension bridge supported by 5 steel ropes out of which a minimum of say, any 3 must be healthy if the bridge is to sustain the load successfully. Similarly, in a 100-strand cable , say, at least 70 must be in good state for the cable to carry the rated current safely. In a communication system , at least 2 transmitters out of its 3 transmitters may be required to function properly for the transmission system to be successful. In an 8-cylinder engine, say, if 4 or more than 4 cylinders are firing then the automobile can be driven; if only three or less are firing , the automobile fails.

A (k,n) system thus has redundancy in it, but it is only partial; and hence is also designated as- a partially redundant system. A (1,n) system is a parallel system; and a (n,n) system- a series system.

A ‘k-out-of-n:F’ system is similarly a system wherein a minimum of k units should have failed for the system to fail. Thus a ‘k-out-of-n:G’ system is equivalent to ‘n-k+1-out-of-n:F’ system.

3. General Assumptions

1. The actual system has been adequately represented by a logic graph of nodes and edges.
2. The edges in series, or the edges in parallel have been reduced to an equivalent single edge.
3. Each unit, and the system have only two mutually exclusive states, viz., good and bad.
4. All nodes are perfect, i.e., the reliability of each node is 1.
5. The states of all the units are mutually independent.
6. The reliability of each unit or edge is known and constant.
7. There is no repair.

4. The Union Law Application
Consider a (3,5) system comprising 5 independent units. The possible sets of 3 at a time out of 5 are \{1,2,3\}, \{1,2,4\}, \{1,2,5\}, \{1,3,4\}, \{1,3,5\}, \{1,4,5\}, \{2,3,4\}, \{2,3,5\}, \{2,4,5\}, and \{3,4,5\}. These are 10 sets; and in general, the number of such sets would be \(\binom{n}{k}\). Any one of these sets should survive for the (k,n) system to survive. Union law of probability is thus to be applied to find the reliability expression. We note that these sets are not mutually exclusive. Applying the said law to these sets, the reliability expression can be obtained. The resulting number of terms is \(\sum_{i=0}^{n} \binom{n}{i}\); an enormous figure, \((1024^{2}-1)!\).

In the expression that is thus obtained, there are a large number of canceling terms, and generating such canceling terms can be avoided. Then the total number of such terms reduces merely to

\[\sum_{i=k}^{n} \binom{n}{i}\]. In the (3,5) system, this number is just 16. An alternative approach is the event space approach; wherein we generate mutually exclusive states, and pick up the terms, which correspond to success states of the system under consideration. For instance, for a 3-unit system, there are 8 possible mutually exclusive states - \{1 2 3\}, \{1 2 3\}, \{1 2 3\}, \{2 3\}, \{1 2 3\}, \{1 2 3\}, \{1 2 3\}, and for a (2,3) system, the states which have 2 or more than 2 good units correspond to success of the system, i.e., the first 4 states. The reliability expression then is \(P_{1}P_{2}P_{3} + Q_{1}P_{2}P_{3} + P_{1}Q_{2}P_{3} + P_{1}P_{2}Q_{3}\).

I would like to explain the principle, and the development of the reliability expression of (k,n) systems, which generates the least possible number of terms and hence computationally very efficient. Large (k,n) systems can thus be handled with only a little computational effort.

5. The Principle of Combining the Terms

The terms having the same multiplicands and differing in one failure only can be combined by using the Boolean identity

\[AB + \overline{A}B = \overline{A}\]

In place of the simple event A, there may be any compound event. Such successive combinations lead to an expression that has fewer terms, and each term having fewer multiplicands than in the original expression. This expression is presentable in recursive form; and the corresponding algorithm to evaluate the reliability can be developed to implement on the computer.

6. Developing the Compact Reliability expression

Reliability expression of a (3,3) system can simply be written as

\[R(3,3) = P_{1}P_{2}P_{3}\]

Reliability expression of a (3,4) system can be written as

\[R(3,4) = P_{1}P_{2}P_{3}P_{4} + Q_{1}P_{2}P_{3}P_{4} + P_{1}Q_{2}P_{3}P_{4} + P_{1}P_{2}Q_{3}P_{4}\]

\[= P_{1}P_{2}P_{3} + P_{4}(Q_{1}P_{2}P_{3} + P_{1}Q_{2}P_{3} + P_{1}P_{2}Q_{3})\]

\[= R(3,3) + P_{4}F(1,3)\]

Similarly,

\[R(3,5) = R(3,4) + P_{5}F(2,4)\]

We can generalize the observation as

\[R(k,n) = R(k,n-1) + P_{n}F(n-k,n-1)\]

The other way of putting it is

\[R(k, k+i) = R(k, k + i-1) + P(k+i)F(i, k+i-1)\]

6.1 Alternative Forms

The above expressions can be reduced to the following forms:
For instance,
\[ R(3,4) = R(3,3) + P1P2P3P4 \left( \frac{Q1}{P1} + \frac{Q2}{P2} + \frac{Q3}{P3} \right) \]
\[ = R(3,3) + S(4,4) \left( \frac{Q1'}{P1'} + \frac{Q2'}{P2'} + \frac{Q3'}{P3'} \right) \]

Also,
\[ R(3,4) = R(3,3) + Q1Q2Q3Q4 \left( \frac{P4}{Q4} \left( \frac{P1P2}{Q1Q2} + \frac{P1P3}{Q1Q3} + \frac{P2P3}{Q2Q3} \right) \right) \]
\[ = R(3,3) + F(4,4) \left( P4'(P1'P2' + P1'P3' + P2'P3') \right) \]

The generalized expression is
\[ R(k,n) = R(k,n-1) + S(n,n) \left[ \text{Sum of combinations of } Q' \text{ taken } (n-k) \text{ at a time out of first } (n-1) \text{ units} \right] \]
\[ R(k,n) = R(k,n-1) + F(n,n) Pn' \left[ \text{Sum of combinations of } P' \text{ taken } (k-1) \text{ at a time out of first } (n-1) \text{ units} \right] \]

If (n-k) is less than (k-1), the former should be used, otherwise the latter. The relations found above are recursive in form.

It is seen that in all the expressions the maximum number of terms
\[ = kC0 + C1 + \ldots + C_{k+1} + \ldots + nC_{n-k} = nCk \]

For example, for a (3,5) system, this number
\[ = 2C0 + 3C1 + 4C2 = 1 + 3 + 6 = 10 = 5C3 \]

For a (4,7) system, the simple event space approach generates 64 terms, each term involving 6 multiplications.

The simplifications proposed above generate only 35 terms and require only 76 multiplications. Moreover, the relations presented are recursive in nature and hence are suited for large systems as well.

Identical Units-
\[ R(k,n) = \sum_{i=k}^{n} nC_i P^i Q^{(n-1)} \]
\[ R(k-1,n) = \sum_{i=k-1}^{n} nC_i P^i Q^{(n-1)} \]

A recursive relation in this case is written as
\[ R(k,n) = R(k-1,n) - nC_{k-1} P^{(k-1)} Q^{(n-k+1)} \]

MTTF relation has also recursivity as
\[ T(k,n) = T(k-1,n) - 1/\lambda (k-1) \]
Where \( \lambda \) and \( \lambda \) are related as \( \lambda = e^{(-2t)} \)

On comparing with a parallel system, and a series system; a (k,n) system has MTTF shorter than that of the parallel system, and longer than that of the series system.

7. k-to-l-out-of-n Systems

A k-to-l-out-of-n (henceforth denoted by (k,l,n)) system is one in which neither less than k nor more than l units out of n units are to be good, if the system is to function successfully. This system is even more general than a (k,n) system. When l equals n, it becomes a (k,n) system; when k = 1; l = n, it is a parallel system; when k = l = n, a series system; and when k = l, a system wherein exactly k successes are required.

Generally, in any given system, it is considered good if all its constituent units are good. In the (k,n) system as already stated, the number of successful units must be anything between k and n, both inclusive. But there are situations, where due to the limitation of some member(s) of the complete system, we cannot afford to allow all n units to function simultaneously. In a powerhouse, we may have seven generating units and five to six units may be enough to feed the required power into network. If all the seven units are operating, the boilers may not be able to cope. It is, therefore, a (5-6,7) system.

A (k-l,n) system, moreover, is a non-coherent system. Generally, for any system, if the reliability of its constituent units improves, we expect system reliability also improves. It is found that this is not true for a (k-l,n) system. It is established that for the iid units system, there exists an optimal value of unit reliability for given values of k, l and n in order to achieve maximum system reliability.

Developing the Reliability Expression
\[ R(k-l,n) = \sum_{i=k}^{n} S(i,n) = \sum_{i=n-l}^{n} F(i,n) \]

If k and l are much less than n, former is preferred, and if k and l are close to n then latter is more suitable because the number of combinations to be generated is reduced.

On expanding the above expression and combining the terms as in (k,n) systems, the final expressions are as given below:
\( R(k-l,n) = s(k,m) + \sum_{i=0}^{n-m-1} p_{m+i} S(k,1,m+i) + \sum_{i=0}^{n-m-1} q_{m+i} S(k+1,1,m+i) \)

\( R(k-l,n) = F(j,m) + \sum_{i=0}^{j} p_{j} F(j+1+i,m+i) + \sum_{i=0}^{j} q_{j} F(j-1,m+i) \)

where \( j = n-l, m = n-(l-k) \).

Various other forms, as in the case of \((k,n)\) system, can be obtained here also.

For iid units, the expression becomes

\[ R(k-l,n) = \sum_{i=k}^{n} C_{i} P^{i} Q^{n-i} \]

Non-Coherence:

The table below gives the values of system reliability for various cases with iid units. It can be seen that for a given set of values of \( k, l, \) and \( p \), there is an optimum value of \( n \); and for given values of \( k, l \) and \( n \), there is an optimum value of unit reliability \( p \).

<table>
<thead>
<tr>
<th>S. No</th>
<th>System description</th>
<th>Value of system reliability for</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( P_{0.5} )</td>
<td>( P_{0.6} )</td>
</tr>
<tr>
<td>1</td>
<td>( k = 5, l = 8, n = 10 )</td>
<td>0.611</td>
</tr>
<tr>
<td>2</td>
<td>( k = 5, l = 9, n = 12 )</td>
<td>0.786</td>
</tr>
<tr>
<td>3</td>
<td>( k = 10, l = 12, n = 15 )</td>
<td>0.147</td>
</tr>
</tbody>
</table>

8. Global Reliability

All–terminal reliability of a network is the probability that all its nodes are connected. This is also called global reliability. The global reliability is, therefore, the probability of existence of a minimal set of up-state edges such that all the nodes of the network are connected. Such a minimal set of edges is known as spanning tree of the network. A network, depending upon its size, has several spanning trees and the number of these trees grows exponentially with the size of the network.

A spanning tree is said to be in up state if all its edges are in up state. The g-reliability is the probability of union of events representing up state of all the spanning trees. These events are generally not disjoint; as such, determining g-reliability involves transforming these events into disjoint events. Application of the union law gives rise to large number of terms, viz., \( 2^{b} - 1 \), where \( b \) is the number of terms in the union. This is, therefore, impracticable to use.

8.1 Substitution Method

Starting from a spanning tree of the network, all other spanning trees can be developed from it by successive replacement of edges one-by-one.

Also, we observe that the probability of the union of events representing up state of two spanning trees which differ in only one edge is given by the sum of probability of up-state of one spanning tree and the product of probability of up-state of the second spanning tree with the probability of failure of the edge in the first spanning tree but not in the second spanning tree. For example, if edges \( e_1, e_2 \) and \( e_3 \) constitute one spanning tree and edges \( e_2, e_3 \) and \( e_4 \) the other, then

\[
Pr \{ e_1e_2e_3 \cup e_2e_3e_4 \} = Pr \{ e_1e_2e_3 \} + Pr \{ e_2e_3e_4 \} - Pr \{ e_1e_2e_3e_4 \}
\]

\[
= Pr \{ e_1e_2e_3 \} + Pr \{ e_2e_3e_4 \}Pr \{ e_1 \} [ As Pr\{ \widehat{e_1} \} = 1 - Pr\{ e_1 \} ]
\]

It could also be written as

\[
Pr \{ e_1e_2e_3 \cup e_2e_3e_4 \} = Pr \{ e_1e_2e_3 \} + Pr \{ e_1e_2e_3 \Pr \{ e_4 \} \}
\]

The two terms in each of the above equations are disjoint with each other.

8.2 Disjoint Tree Generation

Having selected the base tree, further trees in disjoint form are developed in different stages as given below:

Stage 1-

Substitute edges of base tree one by one, by edges in the cotree of the base tree to form new trees. The new edge is put on the extreme right of the tree elements being used. Substitution of edges is done serially. While substituting a particular edge, there may be more than one substitution possibilities; but only one substitution is to be made. To make the trees obtained disjoint with each other, the equation explained above is applied. For
example, if edge 1 is being substituted, edge 1 is retained with a bar over it, meaning thereby that while writing probability expression, probability of failure of edge 1 is to be written where 1 occurs. All trees generated at this stage will have one complemented edge.

**Stage 2-**

Substitute, now, edges one by one which are to the right of barred edge in each of the previous stage (stage 1) trees. One should, of course, retain trees only. The substituted edge is also retained with a bar over it as in Stage 1. all trees generated at this stage have two complemented edges.

**Stage 3-**

Stage 2 trees are to be used for further trees generation at this stage. Now edges are to be substituted which are to the right of rightmost barred edge in each of the above trees. The trees generated at this stage have three complemented edges. The process of substitution is continued in the same manner till all trees which contain \((n_e - n_n +1)\) complemented edges are obtained. Thus there will be \((n_e - n_n +1)\) stages. All the trees would have been obtained by the end of stage \((n_e - n_n +1)\).

The g-reliability expression is obtained by interpreting each edge in the trees as corresponding probability term on one-to-one basis and summing all the tree contributions.

**Example:**

The g-reliability of the bridge network shown below is to be determined.

A tree of edges 1,3,4 which is a star-tree is selected as base tree. For brevity e1,e2,.... are being represented by numerals 1,2,....

**Stage 1** Replacement of edges 1,3,4 in tree 1 3 4 out of edges 2, 5:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Edge complemented</th>
<th>New edge</th>
<th>New tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>1</td>
<td>2</td>
<td>1342</td>
</tr>
<tr>
<td>2.</td>
<td>3</td>
<td>2</td>
<td>1 3 4 2</td>
</tr>
<tr>
<td>3.</td>
<td>4</td>
<td>5</td>
<td>1 3 4 5</td>
</tr>
</tbody>
</table>

**Stage 2**

(a) Replacement of edges 3,4,2, in tree 1 3 4 2 by edge 5:

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Edge complemented</th>
<th>New edge</th>
<th>New tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>3</td>
<td>5</td>
<td>1 3 4 2 5</td>
</tr>
<tr>
<td>2.</td>
<td>4</td>
<td>5</td>
<td>1 3 4 2 5</td>
</tr>
</tbody>
</table>

(b) Replacement of edges 4, 2 in tree 1 3 4 2 by edge 5

<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Edge complemented</th>
<th>New edge</th>
<th>New tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>4</td>
<td>5</td>
<td>1 3 4 2 5</td>
</tr>
<tr>
<td>2.</td>
<td>2</td>
<td>5</td>
<td>1 3 4 2 5</td>
</tr>
</tbody>
</table>

(c) Replacement of edge 5 in tree 1 3 4 5 by edge 2:

This does not result into a tree.

At the end of stage 2 all such trees which have \((5 - 4 + 1) = 2\) complemented edges have been obtained. The process of tree generation is, therefore, over. The g-reliability can be expressed as-

\[
g = P_{1} P_{3} P_{4} + Q_{1} P_{3} P_{4} P_{2} + P_{1} Q_{3} P_{4} P_{2} + P_{1} P_{3} Q_{4} P_{5} + Q_{1} Q_{3} P_{4} P_{2} P_{5} + Q_{1} P_{3} Q_{4} P_{2} P_{5} + P_{1} Q_{3} Q_{4} P_{2} P_{5} + P_{1} Q_{3} P_{4} Q_{2} P_{5}.
\]

**8.3 Incorporating Back Tracking**

Instead of generating trees in stages wherein at each stage one edge is complemented in the tree under
consideration, its all permissible edges are complemented in sequence and new edges are appended. For example, in the tree 1 3 6 5; edged 1; 1,3; 1,3,6 are complemented to generate 1 3 6 5 2, 1 3 6 5 2 4, 1 3 6 5 2 4 7. Of course, 1 3 6 5 2 4 7 is not a tree and hence dropped. Initial spanning tree of the graph is considered. The edges not in this tree are put in the link set {L} and a forward step is taken by assuming the first edge of the initial tree to have failed and adding an edge from {L} to complete the tree. This addition will give rise to either a new AST(Appended Spanning Tree) or a circuit. If a circuit results the link tried is put back in {L} and next link is tried.

Operation is repeated till an AST is obtained. This is the second AST. The link used up in forming this AST is deleted from {L}. Then the edge next to the edge already assumed to have failed is assumed next to have failed and again an AST is developed. This operation continues till the required number of edges have been assumed to have failed, or {L} gets empty, or an AST does not result from addition of any of the links in {L}. Then one performs backward step and generates another set of AST’s by assuming the last assumed failed edge of the most recently generated AST to have not failed but the next to it having failed, and putting back the last link into {L}. Such forward and backward steps continue and end when no backward movement is possible. At this stage all AST’s have been generated.

9. Conclusions

A general type of system, viz., a k-to-l-out-of-n: G system has been discussed. Reliability expressions are obtained for various sub cases of this system. The expressions are reduced by appropriately combining the terms so that we finally get an expression which requires least possible storage and computations. Some simple examples have been included.

**References**


