International Journal of Performability Engineering, Vol. 3, No. 2, April 2007, pp. 243-256

Printed in India

Irredundant Subset Cut Generation to Compute Capacity Related Reliability

S. K. CHATURVEDI*

Reliability Engineering Centre
Indian Institute of Technology Kharagpur
Kharagpur-721302 (WB) India

(Received on February 28, 2006)

Abstract: This paper proposes an efficient cut set approach to enumerate the subsets cut groups (SCG) from apriori knowledge of minimal cutsets of a network. These SCG block a given amount of flow through the network despite the network being *(s, t)* connected and are being used to evaluate *capacity related reliability* (CRR) by employing any *sum of disjoint product* approach.

For each minimal cutset and a given flow requirement, the proposed method enumerates the non-redundant SCG from a cut-matrix, which is constructed from the minimal cutsets information. Two simple equations are proposed to check the validity of a cut for being (i) valid SCG (as a whole) or (ii) used to enumerate its subsets from a certain order onwards to form valid SCG. Thus, an enumeration scheme to enumerate subsets of certain order from a given set is also proposed.

The proposed approach is illustrated with an example and implemented using MATLAB. Several examples are solved and their results are also provided along with a comparison with recent algorithms.

Key Words: Minimal cut sets, network reliability, subset cut, CRR, flow networks

1. Introduction

A general communication system can be modeled as a probabilistic graph G (V, E), which consists of a set of *V* nodes and a set *E* of links, directed or undirected depending upon the corresponding communication channel being simplex (duplex). Various measures for the reliability index of a communication network have been proposed in the literature [1]. The most common quantitative index in reliability analysis of such system is '*s-t reliability'* which, *is defined as the probability of successful communication between a specified pairs of nodes*, viz., source and terminal. However, the assumption that the links capacities are large enough to sustain the transmission of any size is unrealistic and economically unjustifiable in the design of communication networks as the link capacity is a function of cost and definitely limited. Similarly, the most used index in capacity analysis finds the maximum possible flow capability of the network. The capacity assignment is carried out thereafter ignoring the failure probability of the links that is again an infeasible and invalid assumption [2].

* Communicating author's email: skcrec@hijli.iitkgp.ernet.in

Lee [3], Misra [4], Aggrawal et al. [5] may be credited for their pioneering works in modifying the definition of reliability for such networks as *the probability of successfully transmitting the required amount of information/flow between a specified pair of nodes.* In other words, the network would be in a failed state if it cannot transmit the desired amount of information (W_{min}) despite being *'s-t'* connected. Reference [6] calls such performance index as *capacity related reliability* (CRR) that can also be used for other networks such as power distribution network, transportation network or a water supply network.

Obtaining the reliability expression and quantitative assessment of network reliability for such networks with variable link capacities has attracted much attention in the recent past. Literature survey indicates that the evaluation-methodologies can be distinctively put into the following categories, i.e., (i) General Approaches [3, 5, 7-10] (ii) Minimal Path sets and Cut sets Based Approaches [6, 15] (iii) Minimal Path sets Based Approaches [4-5, 11-14] (iv) Minimal Cut sets Based Approaches [18-20]

Most of the general methods suffer from the disadvantages of either being exhaustive and cumbersome as pointed out in [19, 20] and are impractical even for a moderate size network. The approaches in (ii)-(iv) need a priori knowledge of minimal pathsets or minimal cutsets or knowledge of both. The main thrusts in these methods have been on the efficient enumeration and subsequent flow-capacity evaluation of the success/failure sub networks. From this information, they try to enumerate the success (failure) terms and obtain the disjoint sets of these terms by employing well-established *Sum-of-Disjoint-Product* (SDP) [21-24] techniques thereafter, which are suitable for reliability evaluation of large networks, i.e., CRR evaluation is a two-step procedure. The greatest advantage of this approach is that the disjoint terms have a one-to-one correspondence with the reliability (unreliability) expression. That is why these approaches find a niche among the researchers working in this area. The drawbacks and unsuitability of approaches (ii-ii) have also been well stated in [19].

As mentioned in [18], in most practical systems, the number of cutsets is much smaller than the number of pathsets. It is therefore advisable to address the CRR evaluation based on minimal cutsets rather than minimal pathsets. Besides, it is easier to handle to a single minimal cutset at a time rather than to handle two or more pathsets at a time to form CP. Aggrawal et al. [18] perform the CRR calculation by enumerating all the possible non-redundant subset of a given minimal cutset, which would be capable of blocking the desired transmission. Such a subsets they referred as *valid cut groups* and are obtained by using a *cutset matrix* whose rows correspond to various minimal cutsets and columns indicate the links contained in that particular cutset. The non-zero entries in the matrix are replaced by the respective link's flow-capacity contained in the cut. However, it does not provide any cut-group generating scheme and also suffers from the disadvantage of generating large number of redundant terms and thus suitable for small size networks where 1-link valid cuts exits. For large networks with/without 1-link cut, it has to generate all possible combinations of subsets for all cuts.

Recently, Soh and Rai [19] proposed two cutset-based techniques, viz., A1 and A2, which generate the non-redundant valid cut groups in polynomial time in order of number of minimal cutsets of the network, by obtaining *valid cuts*, referred as *subset cut* (SC), by utilizing links with capacity less than desired W_{min} and contained in each cutset. Later, Soh et al [20] further improved this method by providing a new scheme named subset cut enumerations (SCE) to reduce the enumerations and internal redundant SC. These methods, however, require lots of bookkeeping and data processing such as keeping a separate list of cuts meeting the criterion as in theorem 2 [19], removal of redundant cuts as per theorem 3 [19] etc… to reduce the redundancies to a certain extent. Besides, they enumerate many subsets and external and internal redundant SC as the W_{min} approaches closer to maximum capacity of the network.

From the survey of literature, the author has observed the following:

- All algorithms so far suffer from the disadvantage of generating redundant terms (external, internal or both) to a larger to a smaller extent; thereby suffering from the overheads of redundancy removal.
- In pathsets, cutsets or both, based approaches, it has been advised that the minimal pathsets or cutsets should be put in increasing order of cardinality (number of terms in a given path or cut set) and within the same cardinality, lexicographic ordering is being preferred whereas other possibilities have not been explored yet.
- The major issues in CRR problem are lying in efficient enumeration of non-redundant SCG from minimal cut sets or CP from minimal path sets and their carrying or blocking capacity evaluation thereafter. (First Step)
- Most of the approaches existing in the literature are either minimal pathsets or cutsets based, which utilize the apparent advantages of SDP approaches to deduce the CRR expression after obtaining the SCG or CP. (Second Step)

In the present paper, the author proposes an algorithm to generate the irredundant SCG (first step) that can be fed as input to any SDP based reliability evaluation algorithm to obtain the CRR. The generation of such terms requires apriori knowledge of minimal cutsets arranged in order of increasing order of their flow blocking capacity and within the same value of blocking-capacity, a lexicographic ordering. The ordering scheme not only helps in reducing effort in enumerations but also help eliminates the internal/external redundancies through simple validity checks by proposing two equations. Besides, it proposes a subset-generating scheme from a certain order onwards to reduce number of enumerations, which in turn are tested for valid SCG qualifications through the equations proposed. Unlike other approaches, the final product of the algorithm presented in this paper, is non-redundant subsets of minimal cutsets; thus devoid of any redundancy overheads at the end. The algorithm is presented with an illustrative example.

Acronym and Notation

SCG	Subset Cut Group or Cut Group
SDP	Sum of Disjoint Product
CP	Composite Path
A	Cutset Matrix
C	A Cutset or a Subset of Cut
C_{max}	Maximum Flow Capacity of the Network
W_{min}	Desired Flow through Network
CA	Cut-capacity Column Vector
N_s	Current Capacity of Network
nC	Number of Minimal Cutsets
i, j	Index
l	Network-Link.
L	Number of Network Links

2. Development of Algorithm

At the outset, we have the following information of a network, viz.,'s-*t'* nodes, minimal cutsets, link-capacities, and C_{max}. We form the cutset matrix, *A* and its cut-capacity vector, *CA* [18]. In the following sections, author presents some of the preliminaries forming the building blocks of the proposed algorithm by utilizing the above information.

2.1 Preliminaries

For a given W_{min}, a minimal cut of the network may itself be a SCG or its subsets would form SCG. A SCG would either be valid, i.e., will block the flow, W_{min} or redundant (external/internal) or cannot block the flow at all (invalid). The equation that is being proposed to compute the capacity on removal of some links contained in a cutest (or SCG) and removing external redundancy as well is based on the following idea:

For any i^{th} minimal cut set, ordered in their respective flow-capacity and lexicography, a SCG of this cutset would either keep the maximum carrying capacity of the network intact or it would decrease the capacity to a certain lower level, lesser than the maximum carrying capacity of the network, i.e., to the current flow-capacity of the network, N_s, which, can be computed by,

$$N_s = min\ [CX], \tag{1}$$

Where, $CX = CA_j - C_k \ \forall j \le i$, CA_j = *Capacity of j^{th} element of CA* and

C_k = *Sum of capacities of links contained in k^{th} SCG(or cut) of certain order.*

Eq. (1) not only provides the exact network flow capacity on removing a set of links from the network whose capacity-sum is in C_k but also helps in identifying external only or both external and internal redundant SCG. However, working on several examples, it does fail to locate the existence of only internal redundant SCG. The following equation overcomes this problem:

A SCG is said to be internally *redundant, if any link contained in it has its capacity value < Δ, where $\Delta = W_{min} - N_s$* (2)

The value of Δ provides the margin by which the network capacity can be improved through the reinsertions of link(s) from a SCG (note that the SCG is nothing but link(s) taken out from the network). And if any reinsertion of link(s) of this set cannot improve the capacity of the network up to the desired level (W_{min}), it implies that this link(s) in this set is internally redundant.

We explain the use of Eq. (1) and (2) by considering some cases in the following sections:

2.2 Checking a Minimal Cut for SCG or Subset Enumeration

We utilize Eq. (1) and (2) to ascertain whether the cut itself is a SCG or its subsets would form SCG. Obviously, any i^{th} minimal cut would produce a current network flow capacity, N_s, = 0 (<W_{min}) occurring only at i^{th} position in Eq. (1). Thus, $\Delta = W_{min}$. Now, if no link in the cut has capacity < Δ would mean that the cut itself is a SCG. However, if the cut has some link(s) capacities ≤ Δ , then it would imply that there could be a certain sets of links of this cut, capable of blocking a flow of W_{min}. Thus, subsets of this cut will have to be formed to determine those SCG. Other situations could be:

(i) When $N_s \neq 0$ ($<W_{min}$) but occur only at i^{th} position.
This situation occurs when we have already identified and removed the first order SCG from a cut. However, this situation can be dealt with in a similar manner as is done for the cut itself explained in the above paragraph.

(ii)When $N_s \neq 0$ ($<W_{min}$) occurs at i^{th} and at j^{th} position(s) or only at j^{th} position(s) ($j < i$)
This situation also occurs when we have already identified and removed the first order SCG from a cut_i *($i > 1$)* and/or some of the SCGs are going to be externally redundant. Besides, $N_s<W_{min}$, may occur at more than one positions. In this scenario, subsets of the cut will have to be formed.

2.3 Determining the Initial Order of Subsets to Enumerate

Once it is established that the SCG of the cut will have to be formed, the next task is to determine what order of subsets of the cut to be enumerated to reduce the number of enumerations and validity checks? The situation arises when a cut has some link(s) capacity $\leq \Delta$ or situation (ii) stated above. This is dealt with in the following manner:

For an i^{th} cut:

(i) Arrange the capacities of links contained in the cut in decreasing order.
(ii) Calculate, $M = CA_i\text{-}W_{min}$.
(iii) Determine the minimum number of links needed to provide capacity value $> M$, by summing their individual capacities.

The number of links so determined would be the initial order of SCGs to be enumerated, which would be checked for valid/invalid/redundant SCG. The remaining SCG, if any, are then carried over for generating next higher order SCG. The following illustrations are used to explain the above points.

Consider network shown in Fig. 1. The capacities of different links are shown in (.). Table 1 shows the minimal cut sets arranged with their respective cut-capacity and lexicography..

Example 1: Consider the 3[rd] cut with its link capacities shown in brackets, {1 (10), 2(9)}, CA_3 = 19. Let W_{min} = 6. Applying Eq. (1), N_s = min [15, 18, **0**] = 0, occurs at position, i =3. From Eq. (2), Δ = 6. Since there is no link with capacity $<\Delta$, {1, 2} is itself an irredundant SCG.

Consider 4[th] cut, {9(9), 10(5), 11(6)}, CA_4 = 20. Applying Eq. (1) and (2) provide, N_s = min [15, 12, 19, **0**] = 0 occurs at position i = 4 and Δ = 6. Since there are links with capacity $\leq \Delta$, SCG will have to be formed.

Example 2 case (i): Consider 12[th] cut, after removing its first order SCG {1}, i.e., {3(7), 5(5), 8(8), 11(6)}, CA_{12} = 36 and W_{min} = 10. Applying Eq. (1) yields, N_s = 4 at i = 2 (≠ 12). So, M = 26 and link capacity are arranged in decreasing order as [8, 7, 6, 5]. All four links are failed to provide capacity > 26 and so no SCG generation is performed.

Consider 5[th] cut, {4(4), 7 (5), 10(5), 11(6)}. For W_{min} = 10, N_s = min [11, **4**, 19, **9**, **0**] = 0 and $N_s < W_{min}$ occurs at three positions, viz., at [2, 4, 5]. Therefore, after calculating M = 10 and arranging the capacities in decreasing order [6, 5, 5, 4], we find that the SCG of minimum order two are to be generated and if required, then higher order.

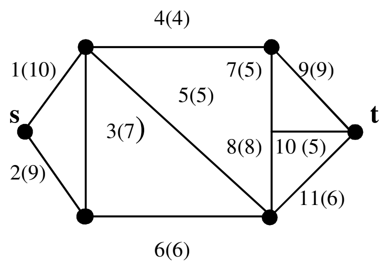


Fig. 1: 6 Node, 11 Link Network, Cmax = 15 units

Table 1: Minimal Cut sets for the Network in Fig. 1

Cut set	CA_i
C_1 = {4, 5, 6}	15
C_2 = {4, 8, 11}	18
C_3 = {1, 2}	19
C_4 = {9, 10, 11}	20
C_5 = {4, 7, 10, 11}	20
C_6 = {1, 3, 6}	23
C_7 = {2, 3, 4, 5}	25
C_8 = {5, 6, 7, 9}	25
C_9 = {7, 8, 9, 11}	28
C_{10} = {5, 6, 8, 9, 10}	33
C_{11} = {2, 3, 5, 7, 9}	35
C_{12} = {1, 3, 5, 8, 11}	36
C_{13} = {1, 3, 5, 7, 10, 11}	38
C_{14} = {2, 3, 5, 8, 9, 10}	43

For W_{min} = 6, there are two positions at which N_s <W_{min} and M = 14. Thus, SCG of third order only are to be generated. Clearly, this method greatly reduces the number of subsets generation.

2.4 Efficient Enumeration of SCG of a Minimal Cut

Once we establish the order of enumeration, we can generate SCGs of a particular order in the following manner. Let us represent a 6^{th} order cut with a set of ordered numbers, S_6 = {1, 2, 3, 4, 5, 6} to represent the position of a link in the cut. Let order of subsets generation required is third. Taking the last three terms of S_6 (equal to the order of enumeration) provides a term = {4, 5, 6}. From this term, we generate all the other terms by noting that the first term in this set could decrease up to 1, second up to 2 and third up to 3. In other words, the last term of third order in the list would be {1, 2, 3}. The complete procedure for the enumerating scheme can be obtained from the author or one may utilize routines available in MALAB.

Clearly, if all SCG of a particular order are valid or redundant then there is no need of generating higher order SCG. However, the SCG of previous order, which are survived after the validity-check operation, can be used as seed to generate the SCG of next higher order. Author has utilized an approach for generating higher order subsets from lower order subsets reported in [25].

2.5 External or External and Internal Redundancy Removal

In Eq. (1), for i^{th} cutset, if $N_s < W_{min}$, occurs at position(s) lesser than i^{th} position, it implies that some SCG have already been encountered in the SCG of some earlier cut set and already been passed through validity checks. Thus we generate SCG of a particular order of this i^{th} cut. On these SCG, we reapply Eq. (1) and check, if $N_s < W_{min}$, occurs at a position(s) lesser than i^{th}, if it happens then the SCG would be externally redundant.

Example 3: Reconsidering $i = 11^{th}$ minimal cutset *{2, 3, 5, 7, 9}* of the network shown in Fig. 1 and consider the following cases:

Case (i): Let us consider one of its SCG *{2, 3, 5, 7} for* $W_{min}=10$. Eq. (1) for this combination would be: $N_s = min\ [10, 18, 10, 20, 15, 16, \mathbf{4}, 15, 23, 28, \mathbf{9}] = 4$ units and

$N_s < W_{min}$ occurs at 7th and 11th positions rather than only at 11th position, (< i , i.e., 7 < 11). This implies that although *{2, 3, 5 7}* is a SCG but it is a superset of SCG {2, 3} generated by 7th minimal cutset, *{2, 3, 4, 5},* earlier and is therefore externally redundant. In fact, for the 7th minimal cutset, Eq (1) for the subset *{2, 3}* is *min [15, 18, 10, 20, 20, 16,* ***9****] = 9*, and minimum occurs at 7th position, which provides *{2, 3}* as a valid SCG and this SCG is not used further to generate its third order SCG. Likewise, {2, 5, 7, 9} would be detected as a superset of a valid SCG {5, 7, 9} produced by 8th cut set earlier.

Case (ii): Consider *5th* minimal cutset, *{4, 7, 10, 11},* of the same network wherein after the test on this cut, it is found that SCG of order two onwards are required to be generated.

Let us consider the subset (SCG), *{4, 7}*, for which Eq. (1) yields, *min [11, 14, 19, 20, 11] = 11* > W_{min}. (Not a valid SCG but possibly adding one or more link of the cut to this SCG might give a valid SCG. Thus it is be taken to generate next higher order combination)

However, for its next order combination, *{4, 7, 10}* and *{4, 7, 11},* Eq. (1) yields:

min [11, 15, 19, 15, ***6****] = 6* (a valid SCG), and
min [11, ***8****, 19, 14,* ***5****] = 5* (A redundant SCG.). In this case the current network flow capacity on removing links *{4, 7, 11}* would be 5 units (can *be* verified visually). However, $N_s < W_{min}$ has already occurred at 2nd position. Therefore, this SCG is redundant. Basically, *{4, 7, 11}⊇{4, 11}* or *{7, 11},* which are valid and non-redundant SCG. In fact, it is a case of both external for *{4, 11}* and internal redundant for *{7, 11}* SCG detected by Eq. (1) but can also be removed using Eq. (2). Similar, situation can occur on a tie between the minimums, i.e., for a SCG, *{10, 11}, min [15, 12, 19,* ***9****,* ***9****] = 9.* However, the set is not a valid SCG as $N_s < C_s$ has occurred at 4th position earlier.

2.6 Internal Redundancy Removal

Let us again illustrate it through a case.

Case (iii): Consider the network of Fig. 2 in 19]. The maximum network flow capacity is 10 units and let the desired network capacity (W_{min}) be 4 units.

Consider SCG, *{1, 2, 7, 11}* of the third minimal cutset, *{1, 2, 7, 11, 15}* in the order of its blocking capacity with respective capacities of links as *{3, 1, 4, 3, 1}.* Applying Eq. (1) yields, N_s *= min [10, 8, 1] = 1* < 4 units, which appears to be a non-redundant SCG. However, capacity of link '2' < Δ (= 3) and even if link '2' is reinserted in the network could only raise the network capacity to 2 units, still less than W_{min}. So link 2's presence or absence does not matter for a specified W_{min} = 4. In fact, *{1, 2, 7, 11}⊇{1, 7, 11},* and *{1, 7, 11}* has already been detected as an irredundant SCG in an earlier iteration implying *{1, 2, 7, 11}* is an internally redundant SCG.

In the following section, the author presents the algorithm for the determination of non-redundant SCG from minimal cutsets for a given W_{min}. The inputs to the algorithm are number of links, W_{min}, link-capacities and minimal cut sets.

3. Algorithm

Steps:

1. Cut Matrix: Formulate cut matrix, *A*, wherein rows corresponds to the various minimal cutsets and columns indicate the links contained in that particular cutset. Besides, all the non-zero entries in a particular row (indicating the presence of

links in that cutset) correspond to the capacities of the individual links contained in the cutset, i.e.,

$$A_{ij} = \begin{cases} \boldsymbol{C_j}\textbf{; if j}^{\textbf{th}}\textbf{ branch having capacity C}_{\textbf{j}}\textbf{ contained in i}^{\textbf{th}}\textbf{ cutsets.} \\ \textbf{0; otherwise.} \end{cases}$$

2. Cut-Capacity Vector: Generate a column vector, *CA* (of order *nC*), which has it i^{th} element as the sum of all the non-zero entries in the i^{th} row of cut matrix, *A*, i.e.,

$$CA_i = \sum_j C_{ij} \forall i = 1, 2 \ldots \text{nC}$$

3. Generating and Validating SCG
   a. *First Order Minimal SCG*
   Scan the cut matrix column-wise. Locate the first non-zero entries, in each column and compute, $N_s = min\ [CA_i\text{-}A_{ij}]$
   If $N_s < W_{min}$ then link l_{ij} is a valid SCG. Make all column entries zero corresponding to link, l_{ij}.

   b. *Higher Order Minimal SCG Generation and Redundancy Elimination*

   Select the rows sequentially (i = 1, 2…nC), which have more than one non-zero entries. Find the corresponding links to form a SCG_i. Apply Eq. (1) on this SCG_i to determine N_s and $N_s<W_{min}$ at how many position(s).

   (i) $N_s<W_{min}$ occurs only at i^{th} position. Determine if any link capacity in $SCG_i<\Delta$. If No, store SCG_i and go to (b) to process next cut.
   If Yes then go to (iii).

   (ii) if $N_s<W_{min}$ occurs at less than at i^{th} position only or more than one positions. Go to (iii)

   (iii) Determine the order of SCG to enumerate and generate SCG of this order. If no order of SCG can be found valid, go to (b) to process next cut. Otherwise,

   (iv) For each SCG, compute,
   $N_s = min\ [CX]$, nPos = $[CX]<W_{min}$
   $\Delta = W_{min} - N_s$
   Where, $CX = CA_j - C_k\ \forall j \le i$, and
   CA_j = *Capacity of* j^{th} *element of CA*
   C_k = *Sum of capacities of links contained in* k^{th} *SCG of certain order.*

   (v) External and Internal Redundancy Check

   External
   If $N_s < W_{min}$ and occurred at position *i*, check for internal redundancy.
   If N_s has occurred at position (s) < *i*, then remove it from the list of combinations from further consideration.
   Internal
   If any link capacity in $SCG<\Delta$, then it is an internal redundant SCG; remove it from further consideration.

Store and remove the qualified and redundant SCG.
For remaining SCG, if any, check, If order of SCG < order of the cut, then generate next higher order SCG and repeat the step from *3 b (iv)*. Else, repeat from step *3(b)* for next cut.

Illustration

Consider the network shown in Fig. 1 with its minimal cutsets and cut-capacities as shown in Table 1. The capacity of the network, C_{max}, is 15 units. Let W_{min} = 10 units, we apply each step of the algorithm on this network. The Cut-matrix, *A*, Cut-capacity vector and steps involved are shown side-by-side in Table 2.

Table 2: Algorithmic Steps to Solve Illustrative Example

Step #1.	**#2.**
$A = \begin{bmatrix} 0&0&0&4&5&6&0&0&0&0&0 \\ 0&0&0&4&0&0&0&8&0&0&6 \\ 10&9&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&9&5&6 \\ 0&0&0&4&0&0&5&0&0&5&6 \\ 10&0&7&0&0&6&0&0&0&0&0 \\ 0&9&7&4&5&0&0&0&0&0&0 \\ 0&0&0&0&5&6&5&0&9&0&0 \\ 0&0&0&0&0&0&5&8&9&0&6 \\ 0&0&0&0&5&6&0&8&9&5&0 \\ 0&9&7&0&5&0&5&0&9&0&0 \\ 10&0&7&0&5&0&0&8&0&0&6 \\ 10&0&7&0&5&0&5&0&0&5&6 \\ 0&9&7&0&5&0&0&8&9&5&0 \end{bmatrix}$	$CA = \begin{bmatrix} 15 \\ 18 \\ 19 \\ 20 \\ 20 \\ 23 \\ 25 \\ 25 \\ 28 \\ 33 \\ 35 \\ 36 \\ 38 \\ 43 \end{bmatrix}$
#3. a Only links, 1 and 6, gives network-flow-capacity = 9, i.e., less than W_{min}. Thus, these two links form single order SCG, viz., {1}, {6}. Replacing all non-zero entries in column 1 and 6 to zero, the new cut matrix is shown on the right side. **# 3. b** **Row #1: SCG {4, 5}** N_s = min [6] = 6 < 10 → {4, 5} at position 1. (Entire SCG check). Δ = 4, No link capacity < Δ → a valid SCG. **Row #2: SCG {4, 8, 11}** N_s = min [11, 0] at position 2. *Δ=10. Link(s) have capacity < Δ and M= 8, so second order SCG to be generated.*	$A = \begin{bmatrix} 0&0&0&4&5&0&0&0&0&0&0 \\ 0&0&0&4&0&0&0&8&0&0&6 \\ 0&9&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&9&5&6 \\ 0&0&0&4&0&0&5&0&0&5&6 \\ 0&0&7&0&0&0&0&0&0&0&0 \\ 0&9&7&4&5&0&0&0&0&0&0 \\ 0&0&0&0&5&0&5&0&9&0&0 \\ 0&0&0&0&0&0&5&8&9&0&6 \\ 0&0&0&0&5&0&0&8&9&5&0 \\ 0&9&7&0&5&0&5&0&9&0&0 \\ 0&0&7&0&5&0&0&8&0&0&6 \\ 0&0&7&0&5&0&5&0&0&5&6 \\ 0&9&7&0&5&0&0&8&9&5&0 \end{bmatrix}$
SCG= {{4, 8}, {4, 11}, {8, 11}} min [11, 6] = 6 <10 → {4, 8} valid SCG. min [11, 8] = 8 < 10 → {4, 11} valid SCG. min [15, 4] = 4 < 10 → {8, 11} valid SCG. No further SCG. **Row #3: No SCG.** **Row #4: {{9, 10}, {9, 11}, {10, 11}}** → all valid SCG.	**Row #5: SCG {4, 7, 10, 11}** N_s = min [11, **8**, 19, **9**, **0**] = 0, and $Ns < W_{min}$ occurs at three position, [2, 4, 5]. *M* =10, Second order SCG are to be generated. SCG are **{{4, 7}, {4, 10}, {4, 11}, {7, 10}, {7, 11}, {10, 11}** {4, 11}→ min [11, **8**, 19, 14, 10} (redundant) *{7, 11}*→ min [15, 12, 19, 14, **9**] (valid) {10, 11} → min [15, 12, 19, **9**, **9**] (redundant)

SCG {{4, 7}, {4, 10}, {7, 10}} used to form next order. {4, 7, 10} → min [11, 14, 19, 15, **6**], *Δ=4*. No link capacity < *Δ* → a valid SCG. {4, 7, 11} → min [15, 11, **8**, 19, 14, **5**] (redundant), Similarly, {4, 10, 11} and {7, 10, 11} (redundant)	**Row #6: No SCG.** **Row #7:** {2, 3} (second and third order SCG) **Row # 8:** {5, 7, 9} (No SCG generation) **Row # 9:** {7, 8, 9} (Third order generation) **Row #10-11:** Generates all redundant 4th order SCG. **Row 12-13**: None. **Row 14:** Generates all redundant 5th order SCG.

Therefore, for W_{min} = 10 units, the non-redundant valid SCG are: {{1}, {6}, {4, 5}, {4, 8}, {4,11}, {8, 11}, {9, 10}, {9, 11}, {10, 11}, {7, 11}, {2, 3}, {4, 7, 10}, {5, 7, 9}, {7, 8, 9} out of 39 SCG generated by the algorithm. The program output for W_{min} = 6 is provided in Appendix I.

4. Comparison, Experimental Results and Discussion

Since, the key issue in CRR problem lying in generation of valid SCG from subsets of cuts, we make the performance comparison with reference to this with the recent approaches reported in [19, 20] based on -how the valid SCG are generated and from how many subsets? To differentiate between the proposed approach and approaches in [19, 20], the following comparative statements can be made:

1. The proposed approach takes an entire cut and tests whether it is a valid SCG or its subsets of a particular order onwards to be enumerated. However, A1 and SCE generate the subsets from the links (defined as *small links*), if any.
2. The difference between the generation schemes used in [19, 20] lies in generating subsets from lower to higher (A1) and higher to lower orders (SCE). Both the schemes thus suffer from overheads of extracting small links from a cut, a test for generating/not generating higher (lower) order SC, supersets extraction and removal of redundant subsets.
3. In proposed approach, obtaining subsets is a single-step process utilizing Eq. (1) and (2). The subsets themselves are SCG containing valid/invalid/redundant SCG. In [19, 20], SCG are obtained by the *set-theoretic difference* operation performed on the cut by each of its subsets. In the process many internal/external redundant terms gets generated. Besides, the internal redundancies are removed from SC (SCG) to obtain MSC at the time of processing i^{th} cut. When MSC for all cuts have been generated, the external redundancies are removed to obtain NMSC (valid SGC).
4. Algorithms in [19, 20] proposed a theorem (T3) to remove the redundant cut sets and to reduce the number of subsets generations. However, there are situations where T3 does not provide any benefits [19, Network of Fig. 7] and for various values of W_{min} [19, Network of Fig. 9]. There is no way apriori to ascertain whether T3 would provide benefits or not, thus there remains an overheads of applying T3.

The proposed algorithm is simple and has been implemented in MATLAB 7.01 on PIV machine under Windows 2000 environment. The algorithm is applied on several networks and the results provided by the implementation were exactly the same as reported in [19, 20]. These results, empirically, show the correctness of our approach.

To compare the performance of the algorithm, author provides comparative results of some of the networks, which are treated as complex in recently reported algorithms [19, 20]. A graphical representation of experimental results of the number of subsets enumerated by various algorithms (Columns 2, 3 and 5 of Table 3-4) with varying W_{min} is shown in Fig. 2 and 3 for visualizing the efficiency of the proposed algorithm. The complex networks are shown in Fig. 4 and 5. The cut sets of these networks are 214 and 7376, respectively. The results for various W_{min} are tabulated in Tables 3-4. Column four of the Tables shows whether T3 provides the benefit or not.

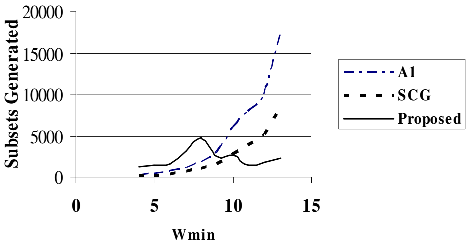


Fig. 2: Comparison of Subsets Generated for Fig. 4

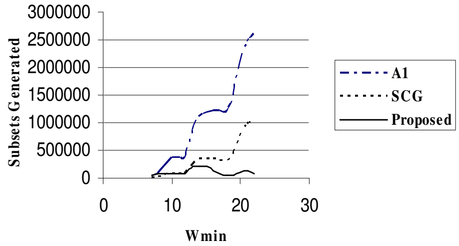


Fig. 3: Comparison of Subsets Generated for Fig. 5

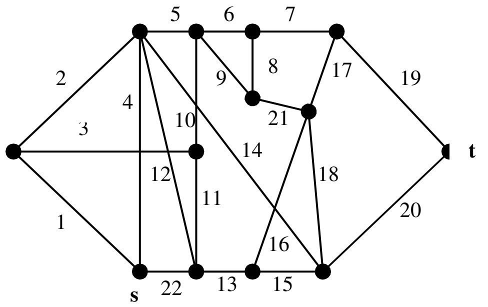
The results shown in bold in Table 3-4 are worth to note.

1. If the desired flow remaining less than the minimum capacity branch in the network, all the minimal cut sets would be valid cut groups. For these cases, there would not be any necessity of generating any subsets.
2. At certain points onwards, algorithms A1 and SCE both generate more subsets of cuts even after taking the benefits of T3 (Table 3) in comparison to the proposed algorithm. Thus, T3 become overheads after a certain point onwards.
3. Wherever T3 starts providing no benefit, the proposed algorithm generates much less number of subsets in comparison to both the algorithms A1 and SCE with T3 application becoming redundant (Table 4).
4. As the network complexity increases, T3 does not provide much benefit. The proposed algorithm starts outperforming both A1 and SCE at an earlier stage and in a greater way (Table 4).
5. From the foregoing points (4) and (5), it can be concluded that whenever network complexity increases or wherever T3 is not applicable, the proposed algorithm is expected to perform better than algorithms of [19, 20]. In fact, we applied our approach to the network of Fig. 7 [19] where no such cuts existed to take the benefit of T3 for all values of W_{min}. For all cases, the proposed algorithm worked superior to A1, which substantiates author's contention to a certain extent, as the results from SCE for the same network were not available in [20].

Summarily, the proposed method is definitely efficient than the method proposed in [18, 19, 20] as it substantially reduces the number of subsets generations, removes the internal/external redundancies simultaneously rather than its removal after generating all cut groups.

Acknowledgement

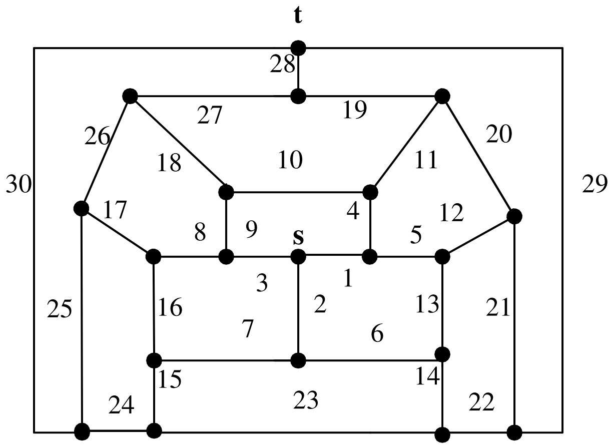
The author would like to thank the anonymous referees for their valued comments and suggestions to enhance the quality and readability of this paper. The author would also like to thank the SRIC, IIT, Kharagpur, for providing the grant to support this research.



1 (Link/Cap) 22

5 8 10 3 6 11 4 3 7 8 12 14 15 13 9 7 9 12 13 16 4 5

Fig. 4: 13 Node, 22 Link Network.



1 2 … …. (Link/Cap) … … 30

6 7 9 8 8 6 7 8 9 6 8 7 6 9 8 7 6 8 9 7 9 7 8 9 8 9 9 8 9 7

Fig. 5: 20 Node, 30 Link Network

Table 3: Results and Comparison for the Network in Fig. 4

W_{min}	Subsets by				Valid SCG
	A1 [19]	SCE [20]	T3/B [19, 20]	Proposed	
1-3	0	0	-	0	214
4	213	175	Y/Y	1246	230
5	388	245	Y/Y	1480	218
6	731	390	Y/Y	1637	294
7	1085	620	Y/Y	3127	298
8	1842	1044	Y/Y	4753	287
9	**2986**	1555	Y/Y	**2411**	190
10	**5971**	**2756**	Y/Y	**2667**	246
11	**7880**	**3900**	Y/Y	**1359**	171
12	**10060**	**5039**	Y/Y	**1707**	143
13	**17354**	**8462**	Y/Y	**2201**	173

Table 4: Results and Comparison for the Network in Fig. 5

W_{min}	Subsets by				Valid SCG
	A1 [19]	SCE [20]	T3/B [19, 20]	Proposed	
1-6	0	0	-	0	7376
7	16925	10196	Y/Y	62381	7644
8	68978	25395	Y/Y	70226	8855
9	**198032**	48906	Y/Y	**70374**	7067
10-12	**378642**	**70374**	Y/Y	**70374**	4962
13	**864170**	**244292**	Y/N	**223371**	4675
14	**1114879**	**331881**	Y/N	**214568**	5278
15	**1185575**	**346776**	Y/N	**222601**	4794
16	**1197592**	**333725**	Y/N	**131241**	2184
17	**1198806**	**321063**	Y/N	**71647**	1199
18	**1198816**	**311826**	Y/N	**51634**	782
19	**1513832**	**474411**	Y/N	**65367**	624
20	**2075477**	**769937**	Y/N	**96971**	647
21	**2466397**	**967941**	Y/N	**124814**	773
22	**2599331**	**1003599**	Y/N	**71171**	479

Appendix I

Program Sample Output

DATA READ

Number of Links: 11
The link capacities as:
10.00 9.00 7.00 4.00 5.00 6.00 5.00 8.00 9.00 5.00 6.00
The desired net capacity: 6.00 units
Number of minimal cut sets as per data read: 14
The DATA PROCESSED

The maximum net capacity: 15.00
The number of minCut for a net capacity of 6.00 is = 17
Number of subsets formed to determine number of minimal cutsets for a Desired capacity of 6.00 is = 52

SCG	Capacity (Ns)
4 6	5.00
5 6	4.00
8 11	4.00
1 2	0.00
9 11	5.00
4 7 11	5.00
4 10 11	5.00
7 10 11	4.00
1 3 6	0.00
2 3 4	5.00
2 3 5	4.00
6 7 9	5.00
6 8 9 10	5.00
2 3 7 9	5.00
1 3 5 7 11	5.00
1 3 5 10 11	5.00
2 3 8 9 10	5.00

References

[1]. Colbourn C. J., *The Combinatronics of Network Reliability,* Oxford University Press, NY, 1987.

[2]. Deo Narsingh, *Graph Theory with Applications to Engineering and Computer Science,* Prentice-Hall Series in Automatic Computation. Prentice-Hall, Englewood Cliffs, 1982**.**

[3]. Lee S. H., *Reliability Evaluation of a Flow Network*, IEEE Transaction on Reliability, Vol. 29, pp. 24-26, 1980.

[4]. Misra K. B. and P. Prasad, *Comment on: Reliability Evaluation of a Flow Network,* IEEE Transaction on Reliability, Vol. 31, pp. 174-176, 1982.

[5]. Aggrawal K. K., Y. C. Chopra Y. C. and J. S. Bajwa, *Capacity Consideration in Reliability Analysis of Communication Systems,* IEEE Transaction on Reliability, Vol. 31, pp. 177-181, 1982.

[6]. Rai S. and S. Soh, *A Computer Approach for Reliability Evaluation of Telecommunication Networks with Heterogeneous Link-Capacities*, IEEE Transaction on Reliability, Vol. 40, pp. 441-451, 1991.

[7]. Rueger W. J., *Reliability Analysis of Network with Capacity Constraints and Failure at Branches and Nodes*, IEEE Transaction on Reliability, Vol. R-35, pp. 523-528, 1986.

[8]. Jane C. C. and Y. Yuan*, A Sum of Disjoint Product Algorithm for Reliability Evaluation of Flow Networks,* European Journal of Operations Research, Vol. 131, pp. 664-675, 2001.

[9]. Rai S. Rai, A. Kumar and E. V. Prasad, *Computing Performance Index of a Computer Network*, Reliability Engineering, Vol. 16, pp. 153-161, 1986.

[10]. Qiu L. Y. and C. H. Zhong, *A New Algorithm for Reliability Evaluation of a Telecommunication Network with Link-Capacities*, Microelectronics and Reliability, Vol. 34, pp. 1943-1946, 1994

[11]. K. K. Aggrawal, “Integration of Reliability and Capacity in Performance Evaluation of Telecommunication Networks”, IEEE Transaction on Reliability, Vol. R-34, No. 2, 184-186, pp. 184-186, 1985.

[12]. Aggrawal K. K., *A Fast Algorithm for the Performance Index of a Telecommunication Network*, IEEE Transaction on Reliability, Vol. R-37, pp. 65-69, 1988.

[13]. Varshney P. K., A. R. Joshi and P. L Chang, *Reliability Modeling and Performance Evaluation of Variable Link Capacity Networks*, IEEE Transaction on Reliability, Vol. 43, No. 3, pp. 378-382, 1994.

[14]. Lee S. M. and D. H. Park, *An Efficient Method for Evaluating Network Reliability with Variable Link-Capacities,* IEEE Transaction on Reliability, Vol. 50, No. 4, pp. 374-379, 2001.

[15]. Lee S. M., C. H. Lee and D. H. Park, *Sequential Capacity Determination of Sub Networks in Network Performance Analysis,* IEEE Transaction on Reliability, Vol. 53, No. 4, pp. 481-486, 2004.

[16]. Schanzer R., *Comment on: Reliability Modeling and Performance Evaluation of Variable Link Capacity Networks,* IEEE Transaction on Reliability, Vol. 44, No. 4, pp. 620-621, 1995.

[17]. Kyandoghere K., *Comment on: Reliability Modeling and Performance Evaluation of Variable Link Capacity Networks,* IEEE Transaction on Reliability, Vol. 47, No. 1, pp. 44-45, 1998.

[18]. Aggrawal K. K., Y. C. Chopra and J. S. Bajwa, *Modification of Cutsets for Reliability Evaluation of Computer Systems*, Microelectronics and Reliability, Vol. 22, No. 3, pp. 1337-340, 1982.

[19]. Soh S. and S. Rai , *An Efficient Cutset Approach for Evaluating Communication Network Reliability with Heterogeneous Link-Capacities*, IEEE Transaction on Reliability, Vol. 50, No. 4, pp. 374-379, 2005

[20]. S. Soh, Lim K. Y. and Rai S., *Evaluating Communication Network Reliability with Heterogeneous Link Capacities using Subset Enumeration*, International Journal of Performability Engineering, Vol. 2, No. 1, pp. 3-17, 2006

[21]. Chaturvedi S. K. and K. B. Misra, *An Efficient Multi-variable Inversion Algorithm for Reliability Evaluation of Complex System Using Path Sets,* International Journal of Reliability, Quality and Safety Engineering, Vol. 9, No. 3, pp. 237-259, 2002.

[22]. Chaturvedi S. K. and K. B Misra, *A Hybrid Method to Evaluate Reliability of Complex Networks*, International Journal of Quality & Reliability Management, Vol 19. No. 8, pp. 1098-1112, 2002.

[23]. Luo T. and S. Rai , *An Improved Algorithm for Coherent System Reliability*, IEEE Transaction on Reliability, Vol. 47, No. 1, 73-78, 1998

[24]. Soh. S. and S. Rai., *CAREL: Computer Aided Reliability Evaluator for Distributed Computer Networks,* IEEE Transaction on Parallel and Distributed System, Vol. 2, No. 2, pp. 199-213, 1991.

[25]. Jasmon G. S. and K. W. Foong, *A Method for Evaluating all the Minimal Cuts of Graph*, IEEE Transaction on Reliability, Vol. R-36, No. 5, pp. 539-545, 1987.

S. K. Chaturvedi is Assistant Professor in Reliability Engineering Centre, Indian Institute of Technology, Kharagpur (India). He did his bachelor's and master's degree from Indian Institute of Technology, Roorkee. He received his Ph. D. degree from Reliability Engineering Centre, IIT, Kharagpur (India) in year 2003. He has research interest in the area of network reliability, life-data analysis, and optimization and has published a few papers in international and national journals. He is Assistant Editor to IJPE.