Raport Badawczy

Research Report

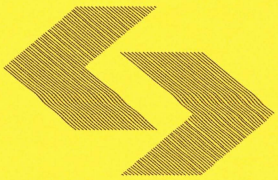
RB/8/2017

A new efficient method of enumerating all min-d-cut-sets in a flow network

J. Malinowski

Instytut Badań Systemowych
Polska Akademia Nauk

Systems Research Institute
Polish Academy of Sciences



POLSKA AKADEMIA NAUK

Instytut Badań Systemowych

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Zakładu zgłaszający pracę:
Prof. dr hab. inż. Olgierd Hryniewicz

Warszawa 2017

Abstract

A new solution to the problem of finding all minimal d-cut-sets in a flow network with one source and one sink node is presented. A d-cut-set is a set of components (nodes and/or links) whose removal or failure causes the maximal flow in the network (Φ_{max}) to fall below level d, while Φ_{max} in the fully operational network is greater or equal to d. A d-cut-set is minimal if no its subset is a d-cut-set. According to the developed method, min-d-cut-sets are generated from min-τ-cut sets, where a τ-cut-set is a set of components whose removal or failure causes the source and sink to be topologically disconnected. A τ-cut-set is minimal if no its subset is a τ-cut-set. The method consists of two algorithms; the first one is applied if $\Phi(C)<2d$, the second – if $\Phi(C)\geq 2d$, where C is the min-τ-cut-set from which min-d-cut-sets are generated, and $\Phi(C)$ denotes the total flow capacity of C's components. This distinction results in quick generation of min-d-cut-sets without first having to find many non-d-cut-sets or non-minimal d-cut-sets. In comparison to the known methods of min-d-cut sets generation, the one presented herein is highly efficient, due to a number of technical improvements which include applying different algorithms for the cases $\Phi(C)<2d$ and $\Phi(C)\geq 2d$, and an effective method of checking if the found min-d-cut set is a redundant one.

1. INTRODUCTION

In this paper two efficient algorithms are proposed for the purpose of enumerating all minimal d-cut-sets in a flow network with one source and one sink node. Although this topic has been pursued by several authors (see [2], [5] and [10]), there is still room for some essential improvements that will be presented herein.

Let D be a set of components (links and/or nodes) in a flow network with one source and one sink node. D will be called a d-cut-set (the author's own term) if the failure of all components in D causes the maximum flow in the network to fall below value d. Clearly, in order that such a definition make sense, the maximum flow, provided that the network is fully operational (all its components are in operation), is assumed to be greater than d. Further, D will be called a minimal-d-cut-set (abbreviated to min-d-cut-set) if no subset of D is a d-cut-set. It should be noted that min-d-cut-sets can be named differently throughout the literature, e.g. with the term "subset cuts" in [2] and [5], but the former term seems to be more self-explanatory.

For the theoretical considerations conducted in the next section we will need the well-known Ford-Fulkerson theorem stating that the maximum flow from the source to the sink node is equal to the smallest of the capacities of the minimal topological cut-sets (see [6]). A topological cut-set (shortened to τ-cut-set) is a set of links whose failure results in topological disconnection between the source and the sink node. A minimal τ-cut-set (shortened to min-τ-cut-set) has the property that no its subset is a τ-cut set. The capacity of a τ-cut-set is the sum of all its links' capacities. The author's own second term – τ-cut-set – is used herein to distinguish it from a d-cut set defined above.

As mentioned in the beginning, the presented method consists of two algorithms. Both of them use min-τ-cut-sets as the initial data, just as the earlier developed algorithms do. Putting it shortly, a number of min-d-cut sets is generated from each min-τ-cut-set, and all min-d-cut-sets are found in the process. It is assumed that all min-τ-cut-sets are known; the problem of finding them is a long studied one and there exist multiple algorithms for that

purpose, presented e.g. in [1], [7], and [11]. The first algorithm is applied if $\Phi(C)<2d$, while the second – if $\Phi(C)\geq 2d$, where C denotes a min-τ-cut-set from which min-d-cut-sets are currently generated, and $\Phi(C)$ denotes the total flow capacity of C's components. This distinction has not been made in the relevant works of other authors, and handling the cases $\Phi(C)<2d$ and $\Phi(C)\geq 2d$ differently accelerates the enumeration procedure significantly.

The list of all (or some) min-d-cut-sets can be used for a number of practical purposes. First of all, it is necessary for computing the network reliability which in this context is defined as the probability that the maximum flow is greater or equal to d. However, other uses are also possible, e.g. for drainage or traffic system control.

The paper is organized as follows. In section 2 a general description of the newly developed method is given, along with a comparison to recent results published in [2] and [5]. In sections 3 and 4 the algorithms for the cases $\Phi(C)<2d$ and $\Phi(C)\geq 2d$ respectively are presented. They are illustrated by their application to an example network, and each step is analyzed in detail. Each of the sections 2-4 begins with several lemmas which together constitute the method's mathematical background. In section 6 a number of conclusive remarks is given with some hints regarding the possible extensions of the method to networks modeled by (partly) directed graphs or networks with multiple sources and/or sinks.

2. NOTATION

$G=(V,E)$ – a network defined as a graph G with the set of nodes V and the set of links E

x_i – the state of the i-th component in $V \cup E$, $i=1,\ldots,|V \cup E|$; $x_i=1$ if the respective component is in working condition, $x_i=0$ if it is failed

s,t – the source and sink nodes

$\Gamma_{s,t}^{G}(x)$ – the s-t connectivity function of G, i.e. $\Gamma_{s,t}^{G}(x)=1$ if there is a path from s to t, otherwise $\Gamma_{s,t}^{G}(x)=0$

τ-cut-set – a set of links whose failure results in topological disconnection between the source and the sink node

min-τ-cut-set – a τ-cut-set whose no subset is a τ-cut-set

m – the number of all min-τ-cut-sets in G

$C_1,\ldots,C_m$ – all min-τ-cut-sets in G

$\varphi(c)$ – the flow capacity of a component c, $c \in E \cup V$

Large capacity component – a component c such that $\varphi(c) \geq d$

$\Phi(C)$ – the total flow capacity of all components in a set C, $C \subseteq E \cup V$, i.e. $\Phi(C)=\sum_{c \in C} \varphi(c)$

$\Phi_{max}(G)$ – the maximal flow from s to t through G

$G \backslash D$ – a subgraph of G obtained by removing all the components of D, $D \subset V \cup E$

d-cut-set – a set D, $D \subset V \cup E$, such that $\Phi_{max}(G \backslash D)<d$, i.e. D is a set of components such that their removal (or failure) causes the maximal flow through G to be smaller than d, where $d \leq \Phi_{max}(G)$

minimal-d-cut-set – a d-cut-set D such that no subset of D is a d-cut set, i.e. $\Phi_{max}(G \backslash D') \geq d$ for each $D' \subset D$

m-d-c-s candidate relative to C_k – a subset D of C_k such that $\Phi(C_k \backslash D) < d$ and $\Phi(C_k \backslash D') \geq d$ for each $D' \subset D$

A(C,D) – the set of all elements in C\D that precede the last element in D, e.g. if C={1,...,10} and D={2,4,6}, then A(C,D)={1,3,5}

B(C,D) – the set of all elements in C\D that succeed the last element in D, e.g. if D and C are defined as above then B(C,D)={7,...,10}

$\mu(C,D)$ – the element of C\D with the smallest flow capacity

Preliminary assumptions:

For convenience, we will equate the components with their indices throughout the whole paper, i.e. the components c_1, c_2, etc. of G will be referred to as 1,2, etc.

We assume that C_k , $k \in \{1,...,m\}$, are ordered according to the increasing flow capacity, hence from the Ford-Fulkerson theorem we have: $\Phi_{max}(G) = \Phi(C_1) \leq ... \leq \Phi(C_m)$. Such an ordering is required by the proposed method.

3. GENERAL DESCRIPTION OF THE PROPOSED METHOD

Before giving a general description of our method, we will set up its theoretical basis in the form of 8 following lemmas.

Lemma 3.1 (used in the proofs of Lemmas 3.2, 3.4, and 3.5)

If $D \subset V \cup E$, then all min-τ-cut-sets of G\D are obtained by removing all redundant sets in the family $(C_1 \backslash D),...,(C_m \backslash D)$.

Remark: A set is redundant in a family of sets if it contains or is equal to another set in this family.

Proof: Let us first assume that D fulfills the following condition: $C_k \setminus D \neq \varnothing$ for $k \in \{1,\ldots,m\}$, i.e. none of the sets $C_1,\ldots,C_m$ is included in D. It is a well-known fact from the reliability theory that $C_1,\ldots,C_m$ are all the s-t min-cut-sets of G if and only if

$$\Gamma_{s,t}^{G}(x) = \Lambda_{k=1}^{m} \mathrm{V}_{i \in C_k} x_i \qquad (1)$$

Clearly, we obtain the s-t connectivity function of G\D by setting x_i to 0 in $\Gamma_{s,t}{}^{G}$ for $i \in D$. We thus have:

$$\Gamma_{s,t}^{G\setminus D}(x) = \Lambda_{k=1}^{m} \mathrm{V}_{i \in C_k \setminus D} x_i = \Lambda_{k \in K_D} \mathrm{V}_{i \in C_k \setminus D} x_i \qquad (2)$$

where K_D is obtained from $\{1,\ldots,m\}$ by removing each k such that $C_k\setminus D$ is redundant in $(C_1\setminus D),\ldots,(C_m\setminus D)$. This ends the first part of the proof.

If $C_k \subset D$ for a certain $k \in \{1,\ldots,m\}$, then there is no path from s to t in G\D, because all components of C_k are removed, thus G\D has no min-τ-cut-sets, or, in other words, the family of min-τ-cut-sets of G\D consists of the empty set alone. In turn, $C_k\setminus D = \varnothing \subseteq C_j\setminus D$ for each j, $j \neq k$, i.e. each $C_j\setminus D$ is redundant in regard to $C_k\setminus D$ being the only (empty) min-τ-cut-set of G\D. Thus the whole proof is completed.

Lemma 3.2 (used in the proof of Lemma 3.3)

If D is a d-cut-set, then there exists a min-τ-cut-set C_k, $k \in \{1,\ldots,m\}$, such that $C_k \cap D$ is also a d-cut-set.

Proof: By assumption, D is a d-cut-set, thus we have:

$$d > \Phi_{max}(G\backslash D) = \min [\Phi(C_k\backslash D): k\in K_D] = \min [\Phi(C_k\backslash D): k\in\{1,...,m\}] \quad (3)$$

The first of the above equalities is obtained from Lemma 1 and the Ford-Fulkerson theorem. The second one follows from the fact that $\Phi(C_j\backslash D) \leq \Phi(C_k\backslash D)$ if $(C_j\backslash D) \subseteq (C_k\backslash D)$, i.e. the sets redundant in $C_1\backslash D,...,C_m\backslash D$ are irrelevant for determining the minimum. Let

$$k^*(D) = \arg\min_{k\in\{1,...,m\}} \Phi(C_k\backslash D) \quad (4)$$

i.e. $k^*(D)$ is one of those k for which $\Phi(C_k\backslash D)$ attains its minimum over $k\in\{1,...,m\}$. From (3), (4), and the equality $A\backslash B = A \setminus (A\cap B)$, where A and B are arbitrary sets, it follows that

$$\Phi_{max}(G\backslash D) = \Phi(C_{k^*(D)} \setminus D) = \Phi[C_{k^*(D)} \setminus (C_{k^*(D)} \cap D)] \quad (5)$$

Since $C_{k^*(D)} \cap D \subseteq D$, we have:

$$\Phi(C_k\backslash D) \leq \Phi[C_k \setminus (C_{k^*(D)} \cap D)],\ k\in\{1,...,m\}, \quad (6)$$

hence

$$\min [\ \Phi(C_k\backslash D): k\in\{1,...,m\}\ ] \leq \min [\ \Phi[C_k \setminus (C_{k^*(D)} \cap D)]: k\in\{1,...,m\}\ ] \quad (7)$$

Using (3) and (5) we obtain:

$$\Phi_{max}(G\backslash D) = \min [\ \Phi(C_k\backslash D): k\in\{1,...,m\}\ ] \quad (8)$$

and

$$\min [\Phi[C_k \setminus (C_{k^*(D)} \cap D)]: k=1,...,m] \leq \Phi[C_{k^*(D)} \setminus (C_{k^*(D)} \cap D)] = \Phi_{max}(G\backslash D) \quad (9)$$

From (8) and (9) we conclude that the inequality (7) can be replaced with equality which, in view of (3) and Lemma 3.1, implies that $d > \Phi_{max}(G\backslash D) = \Phi_{max}[G \setminus (C_{k^*(D)} \cap D)]$, hence $C_{k^*(D)}$ is the sought min-τ-cut-set that intersected with D yields the d-cut-set $C_{k^*(D)} \cap D$.

Lemma 3.3 (states that each m-d-c-s is a subset of certain m-τ-c-s)

If D is a min-d-cut-set, then there exists a min-τ-cut-set C_k, $k\in\{1,\ldots,m\}$, such that $D\subseteq C_k$.

Proof: D is a d-cut-set, hence, by Lemma 3.2, $C_{k^*(D)} \cap D$ is a d-cut-set too, k*(D) being defined by (4). Clearly, $C_{k^*(D)} \cap D \subseteq D$, but since D is a min-d-cut-set, $C_{k^*(D)} \cap D$ cannot be a d-cut-set smaller than D, thus $C_{k^*(D)} \cap D = D$. It readily follows that $D \subseteq C_{k^*(D)}$.

Lemma 3.4 (used in the proof of Lemma 3.5)

If $D \subset C_k$ for a certain $k\in\{1,\ldots,m\}$, then $C_k\backslash D$ is a min-τ-cut-set in $G\backslash D$.

Proof: Since $C_1,\ldots,C_m$ are min-τ-cut-sets in G, none of them includes another, thus $C_j\backslash C_k \neq \emptyset$, $j\neq k$. Let us choose arbitrary $j\neq k$ and $c\in C_j\backslash C_k$. Clearly, $c\in C_j\backslash D$, because $C_j\backslash C_k \subset C_j\backslash D$. In turn, $c\notin C_k\backslash D$, because $c\notin C_k$. As a result, $C_j\backslash D \not\subset C_k\backslash D$, which means that $C_k\backslash D$ is not redundant in the family $C_1\backslash D,\ldots,C_m\backslash D$, hence, by virtue of Lemma 3.1, $C_k\backslash D$ is a min-τ-cut-set in $G\backslash D$.

Lemma 3.5 (the key lemma for the presented method)

Let $D\subset C_k$, where $k\geq 1$. If the following assumptions hold:

1. $\Phi(C_k\backslash D)<d$
2. $\Phi(C_k\backslash D')\geq d$ for each $D'\subset D$
3. $\Phi(C_j\backslash D)\geq d$ for each $j<k$, $k\geq 2$,

then D is a min-d-cut-set, non-redundant in regard to any min-d-cut-set found earlier. Let us note that if D fulfils assumptions 1 and 2, then D is a m-d-c-s candidate relative to C_k.

Proof: By Lemma 3.4, $C_k \backslash D$ is a min-τ-cut-set in $G \backslash D$ (obtained from G by removing all components of D), hence, by virtue of Ford-Fulkerson theorem, the flow capacity of $C_k \backslash D$ is greater or equal to the maximum flow through $G \backslash D$. Thus, by assumption 1, we have:

$$\Phi_{max}(G \backslash D) \leq \Phi(C_k \backslash D) < d, \tag{10}$$

i.e. D is a d-cut-set in G. It thus remains to prove that $\Phi_{max}(G \backslash D') \geq d$ for each $D' \subset D$. If $D' \subset D$, then Lemma 3.1, the Ford-Fulkerson theorem, and the argument used to explain (3), yield:

$$\Phi_{max}(G \backslash D') = \min [\Phi(C_j \backslash D') : j=1,\ldots,m] \tag{11}$$

Since $D' \subset D$, by assumption 3 we have:

$$\Phi(C_j \backslash D') \geq \Phi(C_j \backslash D) \geq d, \; j<k \tag{12}$$

From assumption 2 it follows that

$$\Phi(C_k \backslash D') \geq d \tag{13}$$

Finally, as the flow capacities of C_j increase in j, for $j>k$ we have:

$$\Phi(C_j \backslash D') = \Phi(C_j) - \Phi(C_j \cap D') \geq \Phi(C_k) - \Phi(D') = \Phi(C_k \backslash D') \geq d \tag{14}$$

The above 4 inequalities imply that $\Phi_{max}(G \backslash D') \geq d$, hence, in view of (10), D fulfills both criteria to be a min-d-cut set in G.

Lemma 3.6 (facilitates the verification of assumption 2 in Lemma 3.5)

Let $D \subset C_k$ and let $d_{min}^{(D)}$ be the element of D with the lowest flow capacity, i.e. $\varphi[d_{min}^{(D)}] = \min[\varphi(d) : d \in D]$. Then $\Phi[C_k \setminus (D \setminus \{ d_{min}^{(D)} \})] \leq \Phi(C_k \backslash D')$ for each $D' \subset D$, $D' \neq \varnothing$.

Proof: the lemma follows from the fact that $\Phi(D') \leq \Phi(D \setminus \{ d_{min}^{(D)} \})$ for each $D' \subset D$, $D' \neq \varnothing$.

Corollary: $\Phi(C_k \backslash D') \geq d$ for each $D' \subset D$ if and only if $\Phi[C_k \setminus (D \setminus \{d_{min}^{(D)}\})] \geq d$. Thus, the second assumption 2 in lemma 3.5 can be replaced with "$\Phi[C_k \setminus (D \setminus \{d_{min}^{(D)}\})] \geq d$".

Lemma 3.7 (accelerates the verification of assumption 3 in Lemma 3.5)

If $\Phi(C_j \backslash C_k) \geq d$, where $j<k\leq m$, then $\Phi(C_j \backslash D) \geq d$ for each $D \subseteq C_k$.

Proof: If $D \subseteq C_k$, then $\Phi(C_j \backslash D) \geq \Phi(C_j \backslash C_k)$, and the lemma follows from the assumption that $\Phi(C_j \backslash C_k) \geq d$.

Corollary: If the lemma's assumption holds and D is a m-d-c-s candidate relative to C_k, then the check if $\Phi(C_j \backslash D) \geq d$ can be skipped, because it would give a positive result. Thus, assumption 3 in Lemma 3.5 needs to be verified only for j such that $\Phi(C_j \backslash C_k) < d$.

Lemma 3.8

Let C_k^* be a set composed of all "large capacity" components in C_k. If $C' \subseteq C_k^*$ is a m-d-c-s obtained from a certain C_j, $j<k$, then each m-d-c-s candidate obtained from C_k is redundant in regard to C'.

Proof: If D is a m-d-c-s candidate obtained from C_k, then $C^* \subseteq D$. Otherwise, there would exist at least one $c \in C^* \backslash D$, and, since $C^* \subset C_k$, we would have:

$$\Phi(C_k \backslash D) \geq \Phi(C^* \backslash D) \geq \varphi(c) \geq d \qquad (15)$$

Clearly, in view of (15), D would not be a m-d-c-s candidate. Since $C' \subseteq C^*$, it holds that $C' \subseteq D$, hence D is redundant in regard to C'. This ends the proof.

Remark: Lemma 3.8 allows to easily ascertain whether each m-d-c-s candidate relative to C_k is redundant in regard to one relative to some C_j, $j<k$. Clearly, no m-d-c-s is to be generated from such a C_k.

Now we will present in outline the two algorithms whose detailed pseudo-codes can be found in the two following sections. We will also briefly explain why there are two algorithms instead of one.

From Lemmas 3.3 and 3.5 we conclude that, in order to find all the min-d-cut-sets of G, we have to find all the subsets of each C_k, $k=1,\ldots,m$, satisfying the three assumptions of Lemma 3.5. Actually, min-d-cut-sets are generated from each successive C_k, $k=1,\ldots,m$, by means of one of two algorithms: Alg. 1 if $\Phi(C_k)<2d$, or Alg. 2 if $\Phi(C_k)\geq 2d$. Alg. 1 arranges the components of C_k according to the decreasing flow capacity, then it generates successive subsets of C_k and checks them for being min-d-cut-sets. Alg. 2 arranges the components of C_k according to the increasing flow capacity, then it generates successive subsets of C_k and checks their complements for being min-d-cut-sets. The following subset generation policy is used: first, the successive one-element subsets of C_k are generated (empty set is augmented with the successive elements of $B(C_k, \emptyset)$); second, each (eligible for augmentation) j-element subset C of C_k is augmented with the successive elements of $B(C_k, C)$, $j=1,\ldots,k-1$. For better understanding, let us apply the above policy to the set $C_1=\{2,3,4,5\}$ (see Fig. 1) with the assumption that each less-than-4-element subset C of C_1, such that $B(C_1,C)\neq\emptyset$, is eligible for augmentation. The following subsets are generated in the successive steps:

step 1 – {2}, {3}, {4}, {5}

step 2 – {2,3}, {2,4}, {2,5}; note that $B(C_1,\{2\})=\{3,4,5\}$

step 3 – {3,4}, {3,5}; note that $B(C_1,\{3\})=\{4,5\}$

step 4 – {4,5}; note that $B(C_1,\{4\})=\{5\}$

step 5 – {2,3,4}, {2,3,5}; note that $B(C_1,\{2,3\})=\{4,5\}$

step 6 – {2,4,5}

step 7 – {3,4,5}

step 8 – {2,3,4,5}.

This well-known procedure generates all subsets of a finite set, and each subset is generated only once. However, in the proposed algorithm, many obtained subsets are not eligible for further augmentation, thus the total number of generated subsets of C_k is substantially smaller than $2^{|C_k|}-1$ which is the number of all non-empty subsets of C_k.

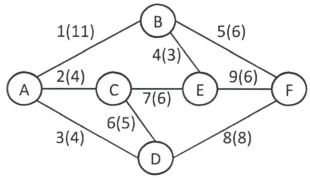


Fig. 1. An example network system

Below, all min-τ-cut-sets of the network in Fig. 1 are listed, ordered according to the increasing flow capacity.

$C_1=\{5,2,3,4\}$, $\Phi_1=17$, $C_2=\{1,2,3\}$, $\Phi_2=19$, $C_3=\{5,9,8\}$, $\Phi_3=20$; $C_4=\{3,6,5,9\}$, $\Phi_4=21$, $C_5=\{4,5,7,8\}$, $\Phi_5=23$, $C_6=\{4,3,6,5,7\}$, $\Phi_6=24$, $C_7=\{7,8,1\}$, $\Phi_7=25$, $C_8=\{3,6,7,1\}$, $\Phi_8=26$, $C_9=\{4,2,6,5,8\}$, $\Phi_9=26$, $C_{10}=\{2,3,5,7,9\}$, $\Phi_{10}=26$, $C_{11}=\{4,9,8,1\}$, $\Phi_{11}=28$, $C_{12}=\{2,6,8,1\}$, $\Phi_{12}=28$, $C_{13}=\{4,3,6,9,1\}$, $\Phi_{13}=29$.

Let C be a currently generated subset of C_k, and let D=C (Alg. 1) or D=C_k\C (Alg. 2). The check whether D is a min-d-cut-set is done as follows. First it is checked if $\Phi(C_k \setminus D)<d$ (assumption 1 in Lemma 3.5). Let us note that if this check is positive then D is a d-cut-set in G (see 1-st part of Lemma's 3.5 proof). In case $\Phi(C_k \setminus D)<d$, it is checked whether $\Phi[C_k \setminus (D \setminus \{d_{min}^{(D)}\})] \geq d$ (see corollary to Lemma 3.6). If the last inequality holds, D is marked as a so-called min-d-cut-set candidate, i.e. in order to state whether D is a min-d-cut-set it has yet to be verified whether assumption 3 in Lemma 3.5 holds. It is important that once D turns out to be a min-d-cut-set candidate, then C is not eligible for further augmentations, because d-cut-sets larger than D=C are not min-d-cut-sets (Alg. 1), while subsets of D=C_k\C are not even d-cut-sets (Alg. 2).

The different handling of the two cases ($\Phi(C_k)<2d$ and $\Phi(C_k)\geq 2d$) allows for generating min-d-cut-set candidates, without generating any non-minimal d-cut-sets (Alg. 1), or for quicker generation of min-d-cut-set candidates, without first having to generate many non-d-cut-sets (Alg. 2). Other advantages of this distinction will be given in sections 4 and 5.

The above outlined method is an improvement of the procedures presented in [2] and [5], also in the context of internal and external redundancy – the concepts used extensively by the authors of the aforementioned papers. A d-cut-set D' is internally redundant w.r.t. to a d-cut-set D, if $D \subset D'$ and both D and D' are generated from the same C_k, $k \in \{1,\ldots,m\}$. Clearly, D' cannot be a min-d-cut set. In turn, a d-cut-set D' generated from C_k, $k \geq 2$, is externally redundant w.r.t. a d-cut-set D, if $D \subset D'$, and D is generated from certain C_j, $j<k$. Obviously, D' cannot be a min-d-cut set. Let us note that assumption 3 in Lemma 3.5 is a criterion used for checking the absence of external redundancy. Indeed, let us suppose that assumption 3 does

not hold, i.e. D is a d-cut-set obtained from C_k and $\Phi(C_j \setminus D) < d$ for certain $j<k$. Then, since $C_j \setminus D = C_j \setminus (C_j \cap D)$ and $C_j \cap D \subseteq D$, it holds that $\Phi[C_j \setminus (C_j \cap D)] < d$ and D is externally redundant w.r.t. $C_j \cap D$ obtained from C_j, hence D is not a min-d-cut set or D is a min. Let us also note that, in view of assumption 3 in Lemma 3.5, it is only required to compare a d-cut-set obtained from C_k with C_j, $j<k$, but not with C_j, $j>k$, in order to state whether it is a min-d-cut-set. By applying Lemma 3.7 the enumeration technique used in this paper reduces (as compared to [2]) the time needed for external redundancy check. Also, this technique avoids generating internally redundant d-cut-sets (Alg. 1), or allows for instantaneous internal redundancy check (Alg. 2).

4. MIN-D-CUT-SETS ENUMERATION FOR $\Phi(C_k) < 2d$

If $\Phi(C_k) < 2d$, then the logic of the algorithm requires that the components of C_k be ordered according to the decreasing flow capacity. The algorithm for this case is based on the following four lemmas.

Lemma 4.1

Let C be a subset of C_k such that

1. $\Phi(C_k \setminus C) \geq d$
2. $\Phi[C_k \setminus (C \cup \{b\})] < d$ for a certain $b \in B(C_k, C)$

$C \cup \{b\}$ is then a min-d-cut-set candidate relative to C_k. The second assumption says that the failure of all components in $C \cup \{b\}$ causes the capacity of C_k to fall below value d. The lemma is also valid for $C = \varnothing$, in which case {b} is a one-element min-d-cut-set candidate.

Proof: From the definition we conclude that D is a m-d-c-s candidate relative to C_k if $\Phi(C_k \setminus D) < d$ and $\Phi(C_k \setminus D') \geq d$ for each $D' = D \setminus \{c\}$, $c \in D$. Let $D = C \cup \{b\}$. By assumption 2, $\Phi(C_k \setminus D) < d$. Further, since $b \in B(C_k, C)$, the ordering of components in C_k yields that $\varphi(c) \geq \varphi(b)$ for each $c \in D$, hence $\Phi[\, C_k \setminus (D \setminus \{c\})\,] \geq \Phi[\, C_k \setminus (D \setminus \{b\})\,] = \Phi(C_k \setminus C) \geq d$, where the last inequality is assumption 1. Thus D fulfills both conditions to be a min-d-cut-set candidate.

Remark: Lemma 4.1 provides a criterion for stating whether $C \cup \{b\}$, where C is a subset of C_k, is a m-d-c-s candidate relative to C_k. Let us note that the lemma's assumptions are equivalent to the first two assumptions of Lemma 3.5, where $D = C \cup \{b\}$.

Lemma 4.2

Let D be a subset of C_k such that $\Phi[A(C_k, D)] \geq d$. It then holds that $\Phi[C_k \setminus (D \cup C)] \geq d$ for each $C \subseteq B(C_k, D)$, i.e. no augmentation of D is a min-d-cut-set candidate relative to C_k.

Proof: If $C \subseteq B(C_k, D)$ then $A(C_k, D) \subseteq C_k \setminus (D \cup C)$, because $A(C_k, D) \subseteq C_k$, $A(C_k, D) \cap D = \varnothing$, and $A(C_k, D) \cap C \subseteq A(C_k, D) \cap B(C_k, D) = \varnothing$. In consequence, $\Phi[C_k \setminus (D \cup C)] \geq \Phi[A(C_k, D)] \geq d$, which was to be proved.

Remark: Lemma 4.2 provides a simple way to check whether a subset of C_k is non-augmentable, thus allowing to reduce the number of generated subsets.

Lemma 4.3

Let b_1, b_2,... be the consecutive elements of $B(C_k, C)$. Then $\Phi[\, C_k \setminus (C \cup \{b_i\})\,]$ is a non-decreasing sequence with respect to $i \geq 1$.

Proof: the lemma is obvious, since $\varphi(b_i)$, $i \geq 1$, is a non-increasing sequence.

Corollary: if $\Phi[\, C_k \setminus (C \cup \{b_i\})\,] \geq d$ then $\Phi[\, C_k \setminus (C \cup \{b_j\})\,] \geq d$ for $j > i$. This property allows not to compute the latter capacities. This lemma is not referenced in Alg. 1, and its role is further explained in the example following Alg. 1.

Lemma 4.4

If $\Phi(C_k) < 2d$ and $\varphi(c^*) \geq d$ for certain $c^* \in C_k$ then $\{c^*\}$ is the only min-d-cut-set candidate relative to C_k.

Proof: From the lemma's assumptions we have:

$$\Phi(C_k \setminus \{c^*\}) = \Phi(C_k) - \Phi(\{c^*\}) < 2d - d = d \tag{16}$$

and, since $\varnothing$ is the only subset of $\{c^*\}$, it holds that

$$\Phi(C_k \setminus \varnothing) = \Phi(C_k) \geq \Phi(C_1) = \Phi_{max}(G) \geq d \tag{17}$$

It thus follows that $\{c^*\}$ is a min-d-cut-set candidate relative to C_k. Let us suppose that D is a m-d-c-s candidate relative to C_k, and $D \neq C$. Since $\varphi(c^*) \geq d$, D must include $\{c^*\}$, otherwise $\Phi(C_k \setminus D) \geq \varphi(c^*) \geq d$ would hold. In turn, (16) implies that for $D' = \{c^*\} \subset D$ we have:

$$\Phi(C_k \setminus D') < d \tag{18}$$

thus D does not fulfill the second criterion to be a m-d-c-s candidate. This ends the proof.

Based on the Lemmas 4.1, 4.2, 4.4, and the general rules formulated in section 3, the following algorithm is constructed.

Algorithm 1

For k=1 to max[k: $\Phi(C_k)<2d$]

Check if $\Phi(\{c_{1,k}\}) \geq d$, where $c_{1,k}$ is the first element of C_k. If so, mark $\{c_1\}$ as m-d-c-s candidate (lemma 4.4), check it for ER*, and pass to the next k

For j=0 to $|C_k| - 1$ do

If all j-element sets generated so far from C_k are marked, pass to the next k;

Else

For each unmarked j-element set C generated so far from C_k do

For the successive $b \in B(C_k, C)$ do

$D \leftarrow C \cup \{b\}$ (augment C with {b});

In case "$\Phi(C_k \setminus D) < d$" do

mark D as m-d-c-s candidate (lemma 4.1), check it for ER, and pass to the next b or to next j-element C (if b is the last element of C_k);

In case "$\Phi(C_k \setminus D) \geq d$" do

If b is the last element of C_k, mark D as non-augmentable ($B(C_k, D)=\varnothing$, thus D cannot be augmented) and pass to next j-element C;

If $\Phi[A(C_k, D)] \geq d$, mark D as non-augmentable (lemma 4.2) and pass to next j-element C;

* ER denotes external redundancy; if ER does not occur, i.e. $\Phi(C_j \backslash D) \geq d$ for each j such that $j<k$ and $\Phi(C_j \backslash C_k)<d$ (see Lemma 3.7), then D is a m-d-c-s non-redundant in regard to any earlier found m-d-c-s (see Lemma 3.5)

It should be noted that no internally redundant d-cut-sets are generated by Alg. 1, because its logic and Lemma 4.1 yield that if the obtained set D is a d-cut-set, then D is a m-d-c-s candidate. Although Alg. 1 can generate non-d-cut-sets (the case $\Phi(C_k \setminus D) \geq d$), their number is small due to the condition $\Phi(C_k)<2d$.

Let us now trace the flow of Algorithm 1 for the example network in Fig. 1. Let d=10. There are two min-τ-cut-sets with capacities lower than 2d, i.e. $C_1=\{5,2,3,4\}$ and $C_2=\{1,2,3\}$ with $\Phi(C_1)=17$ and $\Phi(C_2)=19$. The results of the successive operations are presented in the tables below – one table for each k. The markings used in the 4-th column have the following meanings:

* – D is a m-d-c-s candidate

– D is non-augmentable because $\Phi[A(C_k, D)] \geq d$ (see Lemma 4.2)

| – D is non-augmentable, because b is the last component in C_k. This marking is only used if * or # does not apply

Cells with non-computed values are filled with crosses. Starting from k=2, at the top of each table the values of $\Phi(C_j \backslash C_k)$, $j<k$, are listed to indicate those j for which assumption 3 of Lemma 3.5 needs to be verified in order to check if D is externally redundant.

k=1:

j	C	b	D=C∪{b}	$\Phi(C_k \setminus D)$	$\Phi[A(C_k, D)]$
0	∅	5	{5}	11	0
0	∅	2	{2}	× (Lemma 4.3)	6
0	∅	3	{3}# (Lemma 4.2)	× (Lemma 4.3)	10
1	{5}	2	{5,2}*	7	×
1	{5}	3	{5,3}*	7	×
1	{5}	4	{5,4}*	8	×
1	{2}	3	{2,3}*	9	×
1	{2}	4	{2,4}|	10	×

Remarks:

1. As follows from Lemma 4.3, if $\Phi[\, C_k \setminus (C \cup \{b\})\,] \geq d$ for a certain $b \in B(C_k, C)$, then $\Phi[\, C_k \setminus (C \cup \{b^+\})\,] \geq d$ for each b^+ that succeeds b in $B(C_k, C)$, thus there is no need to compute $\Phi[\, C_k \setminus (C \cup \{b^+\})\,]$, because we only need to know if it's greater or equal to d.

2. As follows from Algorithm 1, $\Phi[A(C_k, D)]$ is only computed if $\Phi(C_k \setminus D) \geq d$

k=2: $\Phi(C_1 \setminus C_2) = 9$ (ER to be verified for j=1)

j	C	b	D=C∪{b}	$\Phi(C_k \setminus D)$	$\Phi[A(C_k, D)]$
0	∅	1	{1}*	8	×

Remark: according to Lemma 4.4, {1} is the only m-d-c-s candidate obtained from C_2

5. MIN-D-CUT-SETS ENUMERATION FOR $\Phi(C_k) \geq 2d$

If $\Phi(C_k) \geq 2d$, then it is required that the components of C_k be ordered according to the increasing flow capacity. The algorithm for this case is based on the following four lemmas.

Lemma 5.1

Let C be a subset of C_k such that

1. $\Phi(C) < d$,

2. $\Phi[C \cup \mu(C_k, C)] \geq d$,

then $(C_k \setminus C)$ is a m-d-c-s candidate relative to C_k.

Proof: Let $D = (C_k \setminus C)$ and $c \in D$. It follows from the definition of μ that

$$\varphi(c) \geq \varphi[\mu(C_k, C)]. \tag{19}$$

We have:

$$\Phi[C_k \setminus (D \setminus \{c\})] = \Phi[(C_k \setminus D) \cup \{c\}] = \Phi[C \cup \{c\}] \geq \Phi[C \cup \mu(C_k, D)] \geq d, \tag{20}$$

where the first inequality in (20) is a consequence of (19), and the second one is the rewritten assumption 2. In view of (20) and assumption 1 which can be written as $\Phi(C_k \setminus D) < d$, D is a m-d-c-s candidate relative to C_k.

Remark: Lemma 5.1 provides a criterion for stating whether $C_k \setminus C$, where C is a subset of C_k, is a m-d-c-s candidate relative to C_k. Let us note that the lemma's assumptions are equivalent to the first two assumptions of Lemma 3.5, where $D = C_k \setminus C$.

Lemma 5.2

If C is a j-element subset of C_k, j>1, $\Phi(C)\geq d$, and C is composed of consecutive elements of C_k, then $\Phi(C')\geq d$ for each j-element C' generated subsequently to C.

Proof: Let $C=\{c_1,...,c_j\}$ where $c_1,...,c_j$ are consecutive components of C_k, and $C'=\{c_1',...,c_j'\}$ be a j-element set generated subsequently to D. First, it will be proved by induction that $c_1'\geq c_1$. This fact is obvious for j=1, in which case $c_1'>c_1$, and let us assume that it holds for a certain $j\geq 1$. Let $C^+=\{c_1^+,...,c_{j+1}^+\}$ be a (j+1)-element subset of C_k composed of its consecutive components. Clearly, C^+ is the first (j+1)-element set obtained by augmenting $\{c_1^+,...,c_j^+\}$, and, according to the subset generating policy, each subsequent (j+1)-element set $C^{+\prime}$ is obtained by augmenting either $\{c_1^+,...,c_j^+\}$ or $\{c_1^{+\prime},...,c_j^{+\prime}\}$ – another j-element set generated subsequently to $\{c_1^+,...,c_j^+\}$. Thus, the first element of $C^{+\prime}$ is equal either to c_1^+ or to $c_1^{+\prime}$. The induction assumption yields that $c_1^{+\prime}\geq c_1^+$, hence the first element of $C^{+\prime}$ is greater or equal to c_1^+, which means that the fact to be proved holds for j+1, and, in consequence, for any $j\geq 1$.

Now we can pass to the proper proof. Since the components of C and C' are ordered according to increasing flow throughputs, we have:

$$\varphi(c_1)\leq ...\leq \varphi(c_j) \qquad (21)$$

and, in view of the above proved fact:

$$\varphi(c_1)\leq \varphi(c_1')\leq ...\leq \varphi(c_j') \qquad (22)$$

As $c_1,...,c_j$ are consecutive components of C_k, from (21) and (22) it follows that $\varphi(c_1')\geq \varphi(c_1),..., \varphi(c_j')\geq \varphi(c_j)$, hence $\Phi(C')\geq\Phi(C)\geq d$, q.e.d.

Remark: It follows immediately that each j-element C' generated subsequently to C is not a m-d-c-s candidate, thus C is non-augmentable.

Lemma 5.3

If $B(C_k,C)=\{b_1, b_2,\dots\}$, where $\varphi(b_1)\leq\varphi(b_2)\leq \dots$, and $\Phi(C\cup\{b_i\})\geq d$, then $\Phi(C\cup\{b_j\})\geq d$ for $j>i$.

Proof: The lemma is obvious, because $\varphi(b_j)\geq\varphi(b_i)$ for $j>j$.

Remark: It follows immediately that $C\cup\{b_j\}$, $j>i$, are not m-d-c-s candidates, thus $C\cup\{b_i\}$ is non-augmentable.

Lemma 5.4

Let $B(C_k,C)=\{b_1, b_2,\dots\}$, where b_i, $i\geq1$, are ordered as in C_k. Then $\Phi[\, C\cup\{b_i\} \cup\mu(C_k, C\cup\{b_i\})\,]$ is non-decreasing with respect to $i \geq 1$.

Proof: If $A(C_k,C) = \varnothing$ then we have:

$$\mu(C_k, C\cup\{b_1\}) = b_2 \text{ and } \mu(C_k, C\cup\{b_i\}) = b_1,\ i \geq 2. \tag{23}$$

It thus follows that

$$\Phi[C\cup b_1 \cup\mu(C_k,C\cup\{b_1\})] = \Phi(C) + \varphi(b_1) + \varphi(b_2),$$

$$\Phi[C\cup b_2 \cup\mu(C_k,C\cup\{b_2\})] = \Phi(C) + \varphi(b_2) + \varphi(b_1),$$

$$\Phi[C\cup b_i \cup\mu(C_k,C\cup\{b_i\})] = \Phi(C) + \varphi(b_i) + \varphi(b_1),\ i \geq3. \tag{24}$$

Since $\varphi(b_i)$ is non-decreasing in i, $i\geq1$, the lemma's thesis follows from (24).

If, in turn, $A(C_k, C) \neq \varnothing$ then

$$\mu(C_k, C\cup\{b_i\}) = a_1,\ i \geq1, \tag{25}$$

where a_1 is the first element of $A(C_k, C)$. In consequence

$$\Phi[C\cup b_i \cup\mu(C_k, C\cup\{b_i\})] = \Phi(C) + \varphi(b_i) + \varphi(a_1),\ i \geq1. \tag{26}$$

Thus the thesis also holds for $A(C_k, C) \neq \varnothing$.

Remark: Lemma 5.4 allows not to compute $\Phi[\, C^+ \cup \mu(C_k, C^+) \,]$ (in order to check, as per Lemma 5.1, whether $C_k \backslash C^+$ is a m-d-c-s candidate), if $C^+ = C \cup \{b^+\}$, where b^+ succeeds the first $b \in B(C_k, C)$ for which $\Phi[\, C \cup \{b\} \cup \mu(C_k, C \cup \{b\}) \,] \geq d$. This lemma is not referenced in Alg. 2, and its role is further explained in the example following Alg. 2.

Based on the Lemmas 5.1, 5.2, 5.3, and the general rules formulated in section 3 the following algorithm is constructed.

Algorithm 2

For k=min [k: $\Phi(C_k) \geq 2d$] to m do

  For j=0 to $|C_k| - 1$ do

    If each j-element subset C of C_k is marked, pass to the next k;

    For each j-element unmarked subset C of C_k do *

      For each $b \in B(C_k, C)$, where {b} is unmarked and $\varphi(b)<d$, do *

        $C^+ \leftarrow C \cup \{b\}$ (augment C with {b});

        In case "$\Phi(C^+)<d$" do

          If $\Phi[C^+ \cup \mu(C_k, C^+)] \geq d$, mark C^+ as non-augmentable, mark $C_k \setminus C^+$ as m-d-c-s candidate, and check it for ER; **

        In case "$\Phi(C^+) \geq d$" do

          If C^+ is composed of consecutive elements of C_k, mark C^+ as non-augmentable and pass to the next j; ***

          Mark C^+ as non-augmentable and pass to the next j-element C; ****

Remarks to commands marked with asterisks:

* adding any b to a marked set C, or adding b such that $\varphi(b) \geq d$ to any C, yields C^+ such that $\Phi(C^+) \geq d$, i.e. $C_k \setminus C^+$ is not a m-d-c-s candidate

** see Lemma 5.1

*** the capacities of all (j+1)-elements subsets of C_k generated subsequently to C^+ would be greater or equal to d (see Lemma 5.2)

**** the capacities of subsequent augmentations of C would be greater or equal to d (see Lemma 5.3)

It should be noted that Alg. 2 generates as few as possible non-d-cut-sets (the case $\Phi(C^+) \geq d$), and although it can generate internally redundant d-cut-sets (it happens if $\Phi(C^+) < d$ and $\Phi[C^+ \cup \mu(C_k, C^+)] < d$), their number is small due to the condition $\Phi(C_k) \geq 2d$. Furthermore, the IR check is done instantaneously (it is positive if $\Phi[C^+ \cup \mu(C_k, C^+)] < d$) and requires no comparison with the previously found d-cut-sets.

To illustrate how Algorithm 2 operates, we will apply it to the network in Fig.1. The results of the successive operations are presented in tables, one table for each k=3,...,m. The algorithm starts with k=3, because this is the first k for which $\Phi(C_k) \geq 2d$. The markings used in the 4-th and 5-th column have the following meanings:

* – $C_k \setminus C^+$ is a m-d-c-s candidate

** – $C_k \setminus C^+$ is a redundant m-d-c-s candidate

– C^+ fulfills the assumptions of Lemma 5.2 or 5.3

| – b is the last component of C_k (this marking is only used if * or # does not apply)

Cells with non-computed values are filled with crosses. Starting from k=3, at the top of each table the values of $\Phi(C_j \setminus C_k)$, $j<k$, are listed to indicate those j for which assumption 3 of Lemma 3.5 needs to be verified in order to check if $C_k \setminus C^+$ is externally redundant.

k=3: $\Phi(C_1 \setminus C_3) = 11$, $\Phi(C_2 \setminus C_3) = 19$ (ER verification not needed)

j	C	b	C^+	$C_k \setminus C^+$	$\Phi(C^+)$	$\Phi[C^+ \cup \mu(C_k, C^+)]$
0	∅	5	{5}*	{9,8}*	6	12
0	∅	9	{9}*	{5,8}*	6	12
0	∅	8	{8}*	{5,9}*	8	14

k=4; $\Phi(C_1 \setminus C_4) = 7$, $\Phi(C_2 \setminus C_4) = 15$, $\Phi(C_3 \setminus C_4) = 8$ (verify ER for j=1,3)

j	C	b	C^+	$C_k \setminus C^+$	$\Phi(C^+)$	$\Phi[C^+ \cup \mu(C_k, C^+)]$
0	∅	3	{3}		4	9
0	∅	6	{6}		5	9
0	∅	5	{5}*	{3,6,9}*	6	10
0	∅	9	{9}*	{3,6,5}*	6	× (Lemma 5.4)
1	{3}	6	{3,6}*	{5,9}**	9	15

Remark: As follows from Lemma 5.4, if $\Phi[\, C^+ \cup \mu(C_k, C^+) \,] \geq d$, where $C^+ = C \cup \{b\}$, then $\Phi[\, C \cup \{b^+\} \cup \mu(C_k, C \cup \{b^+\}) \,] \geq d$, where b^+ succeeds b in $B(C_k, C)$. Thus, there is no need to compute $\Phi[\, C_k \setminus (C \cup \{b^+\}) \,]$, because we only need to know if it's greater or equal to d.

k=5; $\Phi(C_1 \setminus C_5) = 8$, $\Phi(C_2 \setminus C_5) = 19$, $\Phi(C_3 \setminus C_5) = 6$, $\Phi(C_4 \setminus C_5) = 15$ (verify ER for j=1,3)

j	C	b	C^+	$C_k \setminus C^+$	$\Phi(C^+)$	$\Phi[C^+ \cup \mu(C_k, C^+)]$
0	$\varnothing$	4	{4}		3	9
0	$\varnothing$	5	{5}		6	9
0	$\varnothing$	7	{7}		6	9
0	$\varnothing$	8	{8}*	{4,5,7}**	8	11
1	{4}	5	{4,5}*	{7,8}*	9	15
1	{4}	7	{4,7}*	{5,8}**	9	× (Lemma 5.4)
1	{5}	7	{5,7}# (Lemma 5.2)		12; pass to j=2	×

k=6; $\Phi(C_1\backslash C_6)=4$, $\Phi(C_2\backslash C_6)=15$, $\Phi(C_3\backslash C_6)=14$, $\Phi(C_4\backslash C_6)=6$, $\Phi(C_5\backslash C_6)=8$ (verify ER for j=1,4,5)

j	C	b	C^+	$C_k \setminus C^+$	$\Phi(C^+)$	$\Phi[C^+\cup\mu(C_k, C^+)]$
0	∅	4	{4}		3	7
0	∅	3	{3}		4	7
0	∅	6	{6}		5	8
0	∅	5	{5}		6	9
0	∅	7	{7}		6	9
1	{4}	3	{4,3}*	{6,5,7}*	7	12
1	{4}	6	{4,6}*	{3,5,7}**	8	× (Lemma 5.4)
1	{4}	5	{4,5}*	{3,6,7}*	9	× (Lemma 5.4)
1	{4}	7	{4,7}*	{3,6,5}**	9	× (Lemma 5.4)
1	{3}	6	{3,6}*	{4,5,7}**	9	12
1	{3}	5	{3,5}# (Lemma 5.3)		10; pass to next C	×
1	{6}	5	{6,5}# (Lemma 5.2)		11; pass to j=2	×

k=9; $\Phi(C_1\backslash C_9)=4$, $\Phi(C_2\backslash C_9)=15$, $\Phi(C_3\backslash C_9)=6$, $\Phi(C_4\backslash C_9)=10$, $\Phi(C_5\backslash C_9)=6$, $\Phi(C_6\backslash C_9)=10$, $\Phi(C_7\backslash C_9)=17$, $\Phi(C_8\backslash C_9)=21$ (verify ER for j=1,3,5)

j	C	b	C^+	$C_k \setminus C^+$	$\Phi(C^+)$	$\Phi[C^+ \cup \mu(C_k, C^+)]$
0	∅	4	{4}		3	7
0	∅	2	{2}		4	7
0	∅	6	{6}		5	8
0	∅	5	{5}		6	9
0	∅	8	{8}		8	9
1	{4}	2	{4,2}*	{6,5,8}**	7	12
1	{4}	6	{4,6}*	{2,5,8}**	8	× (Lemma 5.4)
1	{4}	5	{4,5}*	{2,6,8}*	9	× (Lemma 5.4)
1	{4}	8	{4,8}# (Lemma 5.3)		11	×
1	{2}	6	{2,6}*	{4,5,8}**	9	12
1	{2}	5	{2,5}# (Lemma 5.3)		10; pass to next C	×
1	{6}	5	{6,5}# (Lemma 5.2)		11; pass to j=2	×

k=10; $\Phi(C_1 \backslash C_{10})=3$, $\Phi(C_2 \backslash C_{10})=11$, $\Phi(C_3 \backslash C_{10})=8$, $\Phi(C_4 \backslash C_{10})=5$, $\Phi(C_5 \backslash C_{10})=11$, $\Phi(C_6 \backslash C_{10})=8$, $\Phi(C_7 \backslash C_{10})=19$, $\Phi(C_8 \backslash C_{10})=16$, $\Phi(C_9 \backslash C_{10})=16$ (ER to be verified for j=1,3,4,6)

j	C	b	C^+	$C_k \setminus C^+$	$\Phi(C^+)$	$\Phi[C^+ \cup \mu(C_k, C^+)]$
0	∅	2	{2}		4	8
0	∅	3	{3}		4	8
0	∅	5	{5}*	{2,3,7,9}**	6	10
0	∅	7	{7}*	{2,3,5,9}**	6	× (Lemma 5.4)
0	∅	9	{9}*	{2,3,5,7}**	6	× (Lemma 5.4)
1	{2}	3	{2,3}*	{5,7,9}**	8	14

In the previous section we found that {1}, composed of one large capacity link, is a m-d-c-s. Thus, on the basis of Lemma 3.8, all m-d-c-s candidates generated from C_7, C_8, C_{11}, C_{12}, C_{13}, each of which includes {1}, would be redundant.

Summing up, the following min-d-cut-sets (d=10) in the network from Fig.1 have been found: {5,2}, {5,3}, {5,4}, {2,3}, {1}, {9,8}, {5,8}, {5,9}, {3,6,9}, {3,6,5}, {7,8}, {6,5,7}, {3,6,7}, {2,6,8}. They are listed in the order in which they have been generated.

6. CONCLUSION

A new efficient algorithm for enumerating all the min-d-cut-sets in a flow network was presented. It has several features that make it competitive in comparison with the analogous algorithms to be found in the relevant literature. First, when generating min-d-cut-sets from the min-τ-cut-set C_k, it makes a distinction between the cases $\Phi(C_k)<2d$ and $\Phi(C_k)\geq 2d$, which significantly accelerates the search procedure. This is achieved by generating a small number

of non-d-cut-sets (Alg. 1 and Alg. 2), and by either avoiding the generation of internally redundant d-cut-sets (Alg. 1) or by generating only a small number of them (Alg. 2). To be more precise, if Alg. 2 instead of Alg. 1 were used for C_k such that $\Phi(C_k)<2d$, then many internally redundant d-cut-sets would be generated before a min-d-cut-set would be found. In turn, if Alg. 1 instead of Alg. 2 were used for C_k such that $\Phi(C_k)\geq 2d$, then many non-d-cut-sets would be generated before a min-d-cut-set would be found.

Second important feature is a fast way of checking the external redundancy, based on Lemma 3.7. If D a m-d-c-s candidate obtained from C_k, then D is not compared with all min-d-cut-sets obtained from C_j, $j<k$, but only with the min-τ-cut-sets C_j, $j<k$. This is particularly effective when the number of already found min-d-cut-sets exceeds k. The third advantage of the presented method is that it does not unnecessarily compute the flow capacities of some generated sets, as follows from lemmas 4.3 and 5.4. The fourth feature is the possibility of quickly determining the only m-d-c-s candidate in C_k (lemma 4.4) or ascertaining that all candidates in C_k are externally redundant (lemma 3.8).

For comparison with the earlier results of other authors, the newly developed method was applied to the network structure used as an example in [2] and [5], and it needed significantly less basic operations in order to find the min-d-cut-sets of that network.

As mentioned in the introduction, the first and foremost application of a min-d-cut-set enumeration method is to compute the reliability of a flow network, which is defined as the probability that the maximal flow in the network is greater or equal to d. If all min-d-cut-sets are known, then the SDP (sum of disjoint products) method of constructing an expression for

the network reliability is considered to be highly efficient (see [4], [12]). The starting point of this method is the Boolean sum of products each of which corresponds to one min-d-cut-set. Quite recently, the author of this paper devised a method competitive to SDP, which he named "Tree of Boolean Expressions" and whose details can be found in [8].

Although the provided example suggests that the proposed method can only be used for networks with undirected links and failure-free nodes, this is not the case. Let us note that the initial data used by the method are all the min-τ-cut-sets of the considered network, and there exists algorithms that find min-τ-cut-sets also in networks with failing nodes and directed links (see [7], [9]). Moreover, the method can be generalized to the multi-source and/or multi-sink case, but this would require appropriate redefining of min-τ-cut-sets, which will be a topic of further research. It should be noted that a similar issue has been addressed in [3].

Bibliography

[1] U. Abel and R. Bicker, "Determination of All Minimal Cut-Sets between a Vertex Pair in an Undirected Graph", IEEE Transactions on Reliability, vol. R-31, pp. 167-171, 1982.

[2] S. Chakraborty and N.K. Goyal, "Irredundant Subset Cut Enumeration for Reliability Evaluation of Flow Networks," *IEEE Trans. Rel.*, vol. 64, no. 4, pp. 1194-1202, 2015.

[3] S. Chakraborty and N.K. Goyal, "An Efficient Reliability Evaluation Approach for Networks with Simultaneous Multiple-Node-Pair Flow Requirements," *Quality and Reliability Engineering International*, 2016.

[4] E. Chatelet et al., "An optimal procedure to generate sums of disjoint products," *Reliability Engineering and System Safety*, vol. 65, no. 3, pp. 289–294, 1999.

[5] S.K. Chaturvedi, "Irredundant Subset Cut Generation to Compute Capacity Related Reliability," *International Journal of Performability Engineering*, vol. 3, no. 2, pp. 243-256, 2007.

[6] T.H. Cormen et al., "Introduction to algorithms," MIT Press, London 2009.

[7] J. Malinowski, "A new efficient algorithm generating all minimal s-t cut sets in a graph-modeled network," in *Proc. Int. Conf. of Numerical Analysis and Applied Mathematics 2015*, AIP Conference Proceedings, pp. 480030-1 – 480030-4, 2016.

[8] J. Malinowski, "A fast tree-scanning algorithm finding a compact expression for the structure function of a system with known minimal path(cut) sets," in *Risk, Reliability and Safety: Innovating Theory and Practice, Proc. of the 26-th European Safety and Reliability Conference, ESREL 2016*, pp. 1375-1379.

[9] B. Singh, "Enumeration of node cut sets for an s-t network", *Microelectronics Reliability*, vol. 34, pp. 559-561, 1994.

[10] S. Soh and S. Rai, "An efficient cutset approach for evaluating communication-network reliability with heterogeneous link-capacities," *IEEE Trans. Rel.*, vol. 54, no. 1, pp. 133–144, 2005.

[11] W.C. Yeh, "A simple algorithm to search for all MCs in networks", *European Journal of Operational Research*, vol. 174, pp. 1694-1705, 2006.

[12] W.C. Yeh, "An improved sum-of-disjoint-products technique for the symbolic network reliability analysis with known minimal paths," *Reliability Engineering and System Safety*, vol. 92, no. 2, pp. 260–268, 2007.