Some Results on Generalized Connectivity With Applications to Topological Design of Fault-Tolerant Multicomputers

A. Duksu Oh
Department of Mathematics
St. Mary's College of Maryland
St. Mary's City, MD 20686

Hyeong-Ah Choi
Department of Electrical Engineering & Computer Science
George Washington University
Washington, DC 20052

Abdol-Hossein Esfahanian
Department of Computer Science
Michigan State University
East Lansing, MI 48824

Abstract. The connectivity of a graph $G(V, E)$ is the least cardinality $|S|$ of a vertex set S such that $G - S$ is either disconnected or trivial. This notion of connectivity has been generalized in two ways: one by imposing some graphical property on the set S, and the other by imposing some graphical property on the components of the graph $G - S$. In this paper, we study some instances of each of the above generalizations.

First, we prove that the problem of finding the least cardinality $|S|$ such that the graph $G - S$ is disconnected and one of the following properties (i) - (iii) is satisfied, is NP-had: (i) given a set of forbidden pairs of vertices, the set S does not contain a forbidden pair of vertices; (ii) the set S does not contain the neighborhood of any vertex in G; (iii) no component of $G - S$ is trivial. We then show that the problem satisfying property (ii) or (iii) has a polynomial-time solution if G is a k-tree. Applications of the above generalizations and the implications of our results to the topolodcal design of fault-tolerant networks are discussed.

1. Introduction

Designing fault-tolerant topologies for multicomputer systems (whose underlying interconnection can be modeled by a graph) has increasingly become the concern of many researchers due to the need for high performance and reliable systems [Haye76, LoFu87]. Fault-tolerance at the *topological level* implies that the types of faults to be tolerated are processor and/or link failures, and a multicomputer (MC) is said to be *fault-tolerant* if it can remain *functional* in presence of failures.

One functionality criterion that has received much attention considers a MC functional as long as there is a nonfaulty communication path between each pair of nonfaulty processors [KuRe80, Boes86, PrRe82]. In other words, the underlying topology of the MC should remain *connected* in the presence of certain failures. A major consideration in the study of this model has been the choice of the *fault* *patterns*, that is, the ways that processors and/or links are perceived to fail. Given the nature of failures, *measures* have been proposed to *quantify* the extent of fault-tolerance of topologies for MCs [Boes86, ZaEl88].

Researchers have mostly used graph theoretic concepts to develop deterministic measures of fault-tolerance. The *edge-* and *vertex-connectivity* have been the prime candidates for deterministic measures of fault-tolerance [Tane81, PrRe82, DuHw88]. The *vertex-connectivity*, $\kappa(G)$, of graph G is defined as the least cardinality $|S|$ of a set $S \subset V$ such that $G - S$ is either disconnected or *trivial* (i.e., a single-vertex graph). The *edge-connectivity*, $\lambda(G)$, is defined similarly with S being a set of edges. Connectivities are among the most extensively studied graph *invariants*, partly due to their many applications. A recent study of connectivities can be found in [Oell86].

Using connectivities as measures of fault-tolerance implies that if graph G is the underlying topology of a MC, then that MC can tolerate $\kappa(G) - 1$ processor failures or $\lambda(G) - 1$ link failures. These parameters, however, have some deficiencies, two of which are examined here. First, the parameters do not differentiate between the different types of disconnected graphs that result from removing $\kappa(G)$ disconnecting vertices or $\lambda(G)$ disconnecting edges. This implies that the *severity of the damage* to the system caused by processor or link failures is unaccounted for in these parameters which, consequently, renders them inaccurate for some applications [Boes86]. To compensate for this shortcoming, one can make use of several generalized measures of connectedness that have appeared in the graph theory literature [Watk70, Chva73, Boes86, BaES87]. These measures include the *atomic number*, the *toughness*, the *mean connectivity*, and the *integrity* of a graph. For many applications, these parameters, in conjunction with connectivities, can provide improved measures of fault-tolerance for MCs. Computational aspects of some of these parameters have also been studied [ClEF87].

To see the second deficiency in using connectivities in the above context, we note that in using these parameters it is tacitly assumed that any subset of processors (or links) can potentially fail. To compute $\kappa(G)$, for example, one finds the minimum cardinality $|S|$ of a set S of vertices (processors) such that $G - S$ is disconnected; the likelihood of the corresponding processors failing at the same time is not accounted for in this computation. The parameter $\lambda(G)$ is computed similarly. Therefore, these measurements are inaccurate for MCs in which some subsets of system components (processors or links) do not fail simultaneously. Such sets are called *forbidden faulty-sets*. The notion of forbidden faulty-sets can arise in many contexts. Let graph $G(V, E)$ be the underlying topology of a MC. Also, let $S \subset V$ be any subset of V such that $|S| = \kappa(G)$, and there are $\binom{|V|}{\kappa}$ distinct such subsets. By definition, there exists a set S such that $G - S$ is disconnected (to avoid trivial cases, it is assumed that G is not a *complete* graph), and we will refer to such a set as a *minimum cut*. Now, if the number of minimum cuts in G is very small compared to $\binom{|V|}{\kappa}$, one may look "beyond" vertex-connectivity and let each minimum cut be a forbidden faulty-set, and then ask for the minimum number of vertices whose removal will disconnect G. An example of such graphs is the n-cube, Q_n, which is the underlying topology of hypercube multicomputers. Each minimum cut in Q_n is of size n, and among the $\binom{2^n}{n}$ possible n-subsets, exactly 2^n are minimum cuts, which is a small percentage especially when n is large [Esfa88]. Some work has been done in designing graphs with as few minimum cuts as possible [HaAm73].

The above discussion was the motivation behind the connectivity generalization introduced by Esfahanian and Hakimi [EsHa88]. They generalized the notion of connectivity by imposing some graphical property on the set being removed. Formally, the *r-connectivity* (read as restricted connectivity) is defined as follows. Let $G(V, E)$ be a graph (or a directed graph) and ρ be a given graphical property. Then the *r-connectivity* is the cardinality $|S|$ such that:

(a) the set S has property ρ,
(b) the graph $G - S$ is disconnected,
(c) no set S' satisfying (a) and (b) above has fewer elements than set S.

In general, the set S can contain both vertices and edges. However, when S is a subset of $V(G)$, the corresponding r-connectivity will be referred to as *r-vertex-connectivity*. Similarly, the term *r-edge-connectivity* will be used when S is a subset of $E(G)$.

In this paper, some instances of r-connectivity are studied. In particular, the *r-vertex-connectivities* are studied for certain graphical properties ρ.

2. Problem Formulation

In this section, we formulate a list of r-connectivity related problems.

Problem 1:

Instance: A connected graph $G(V, E)$ and set $M = \{X_1, X_2, \ldots, X_m\}$ where each X_i is a subset of $V(G)$, and an integer α.

Question: Is there a set $S \subset V(G)$ such that:

(a) for each i, we have $|S \cap X_i| < |X_i|$,
(b) the graph $G - S$ is disconnected,
(c) $|S| = \alpha$?

Problem 2:

Instance: A connected graph $G(V, E)$ and set $M = \{A(v) \mid$ for each vertex $v \in V(G)$, the set $A(v)$ consists of all the vertices which are adjacent to v in $G\}$, and an integer α.

Question: Is there a set $S \subset V(G)$ such that:

(a) for each $v \in V(G)$, we have $|S \cap A(v)| < |A(v)|$,
(b) the graph $G - S$ is disconnected,
(c) $|S| = \alpha$?

Problem 3:

Instance: A connected graph $G(V, E)$ and set $M = \{X \subset V \mid$ for any $v \in V$, if $A(v)$ is a subset of X then v is a member of $X\}$, and an integer α.

Question: Is there a set $S \subset V(G)$ such that:

(a) the set S is an element of M,
(b) the graph $G - S$ is disconnected,
(c) $|S| = \alpha$?

3. NP-Completeness Results

In this section, we discuss the complexities of Problems 1 - 3. We begin with Problem 1. We observe that problem 1 can be answered in polynomial-time when each $|X_i| = 1$. However, in proving Theorem 1, we show that it is NP-complete to determine whether there exists a subset S of $V(G)$ that satisfies only conditions (a) and (b) if $|X_i| > 1$. This makes the existence of any polynomial-time approximation algorithm unlikely unless $P = NP$.

Theorem 1. *Problem 1 is NP-complete, even if G is a bipartite series-parallel graph with the maximum degree 3 and each $|X_i| \leq 2$.*

Proof: Since it is easy to see that the above problem is in NP, our proof will focus on showing a polynomial transformation from the following NP-complete problem.

X3C Problem:

Instance: Set Y with $|Y| = 3q$ and a collection C of 3-element subsets of Y such that no element of Y occurs in more than *three* subsets.

Question: Is there a *cover* for Y, that is, does C contain a subcollection $C' \subseteq C$ such that every element of Y occurs in exactly one member of C'?

Let $Y = \{y_1, y_2, \ldots, y_{3q}\}$ and $C = \{c_1, c_2, \ldots, c_l\}$ be an instance of the X3C problem. We set $D_i = \{d_i^j \mid y_i \text{ occurs in } c_j\}$ for $1 \leq i \leq 3q$, and $D = \cup_{i=1}^{3q} D_i$. Observe that $|D_i| \leq 3$ for each i, $1 \leq i \leq 3q$, and so set $D_i = \{d_i^{i_1} \mid |D_i| = 1\} \cup \{d_i^{i_1}, d_i^{i_2} \mid |D_i| = 2\} \cup \{d_i^{i_1}, d_i^{i_2}, d_i^{i_3} \mid |D_i| = 3\}$, where $i_1 < i_2 < i_3$. Set $W = \cup_{i=1}^{3q} (\{w_i^1 \mid |D_i| = 2\} \cup \{w_i^1, w_i^2 \mid |D_i| = 1\})$, and choose an integer p such that $2^{p-1} < 3q \leq 2^p$. Let H and F denote full binary trees with leaf node sets $\{h_1, h_2, \ldots, h_{2^p}\}$ and $\{f_1, f_2, \ldots, f_{2^p}\}$ and the root nodes s and t, respectively. We now define a graph $G(V, E)$ as follows. The vertex set of G is

$$V(G) = V(H) \cup D \cup W \cup Y \cup V(F)$$

and the edge set of G is

$$
\begin{aligned}
E(G) &= E(H) \cup \{(h_i, d_i^{i_1}) \mid 1 \le i \le 3q\} \\
&\cup \Big(\cup_{1 \le i \le 3q} (\{(d_i^{i_1}, d_i^{i_2}), (d_i^{i_2}, d_i^{i_3})(d_i^{i_3}, y_i) \mid |D_i| = 3\} \\
&\cup \{(d_i^{i_1}, d_i^{i_2}), (d_i^{i_2}, w_i^1), (w_i^1, y_i) \mid |D_i| = 2\} \\
&\cup \{(d_i^{i_1}, w_i^1), (w_i^1, w_i^2), (w_i^2, y_i) \mid |D_i| = 1\}) \Big) \\
&\cup \{(y_i, f_i) \mid 1 \le i \le 3q\} \cup E(F).
\end{aligned}
$$

Note that the graph G is a bipartite series-parallel graph with the maximum degree 3. As an example, let $Y = \{y_1, y_2, y_3, y_4, y_5, y_6\}$ and $C = \{c_1, c_2, c_3\}$, where $c_1 = \{y_1, y_2, y_3\}$, $c_2 = \{y_1, y_2, y_4\}$, $c_3 = \{y_4, y_5, y_6\}$. Figure 1 shows the graph $G(V, E)$ for this example.

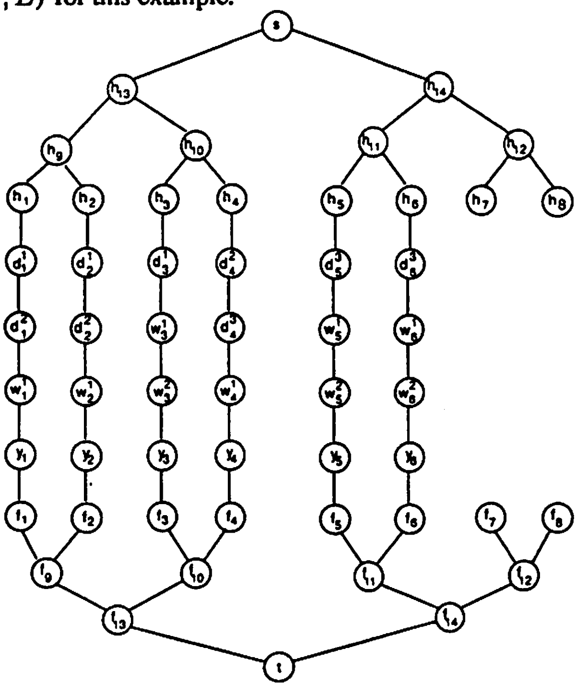

Figure 1. Construction of $G(V, E)$ from $c_1 = \{y_1, y_2, y_3\}$, $c_2 = \{y_1, y_2, y_4\}$ and $c_3 = \{y_4, y_5, y_6\}$.

The construction of our instance is complete by setting

$$
\begin{aligned}
M = &\{\{v\} \mid v \in V(H) \cup V(F) \cup W \cup Y\} \cup \Big(\cup_{i=1}^{3q} \big\{ \{d_i^j, d_i^{j'}\} \mid j \ne j' \big\} \Big) \\
&\cup \big\{ \{d_i^j, d_{i'}^{j'}\} \mid d_i^j, d_{i'}^{j'} \in D, i \ne i', j \ne j' \text{ and } c_j \cap c_{j'} \ne \emptyset \big\}
\end{aligned}
$$

In Figure 1, $M = \{\{s\},\{h_1\},\{h_2\},\ldots,\{h_{14}\},\{t\},\{f_1\},\{f_2\},\ldots,\{f_{14}\}, \{w_1^1\},\{w_2^1\},\{w_3^1\},\{w_3^2\},\{w_4^1\},\{w_5^1\},\{w_5^2\},\{w_6^1\},\{w_6^2\},\{y_1\},\ldots,\{y_6\}, \{d_1^1,d_2^2\},\{d_1^1,d_4^2\},\{d_1^2,d_2^1\},\{d_1^2,d_3^1\},\{d_1^2,d_4^3\},\{d_1^2,d_5^3\},\{d_1^2,d_6^3\},\{d_2^1,d_4^2\}, \{d_2^2,d_3^1\},\{d_2^2,d_4^3\},\{d_2^2,d_5^3\},\{d_2^2,d_6^3\},\{d_3^1,d_4^2\},\{d_4^2,d_5^3\},\{d_4^2,d_6^3\}\}$.

We next proceed to prove that there exists an exact cover C' for Y if and only if there exists a vertex cut S which satisfies the conditions (a) and (b).

Suppose there exists an exact cover C' for Y. Then, $|C'| = q$ and each element of Y occurs in exactly one member of C'. Further, observe that if y_i and $y_{i'}(i \neq i')$ occur in $c_j \in C'$ and $c_{j'} \in C'$, respectively, then $\{d_i^j, d_{i'}^{j'}\} \notin M$. This implies that $S = \{d_i^j | y_i \text{ occurs in } c_j \in C'\}$ is the desired vertex cut. Conversely, suppose that there exists a set S that satisfies the conditions (a) and (b). Then, $S \subseteq D$. Note that for all i, $1 \leq i \leq 3q$, there is a path from s to t going through vertices in D_i. This gives for $1 \leq i \leq 3q$ a lower bound for $|S \cap D_i|$, that is, $|S \cap D_i| \geq 1$. On the other hand, for any pair of vertices d_i^j and $d_i^{j'}$ in D_i with $j \neq j'$, we have $\{d_i^j, d_i^{j'}\} \in M$ and thus $|S \cap D_i| \leq 1$. Hence, $|S \cap D_i| = 1$ for all i, $1 \leq i \leq 3q$. Consequently, if $d_i^j, d_{i'}^{j'} \in S$ and $j \neq j'$, then since $\{d_i^j, d_{i'}^{j'}\} \notin M$, we have $c_j \cap c_{j'} = \emptyset$. This establishes the existence of an exact cover C' for Y. In particular, $C' = \{c_j \in C | d_i^j \in S\}$. This completes the proof of Theorem 1. ∎

In order to show the NP-completeness of the other two problems, we next describe the following known NP-complete problem, which will be used in both transformations.

One-In-Three 3SAT Problem

Instance: Set U of variables, collection C of clauses over U such that each clause $c \in C$ has $|c| = 3$.

Question: Is there a truth assignment for U such that each clause in C has exactly one true literal?

Since both Problems 2 and 3 belong to NP, again our proofs will focus on the polynomial time transformations.

Theorem 2. *Problem 2 is NP-complete.*

Proof: Let $U_0 = \{u_1, u_2, \ldots, u_{n_0}\}$ and $C_0 = \{c_1, c_2, \ldots, c_{m_0}\}$ be an instance of the One-In-Three 3SAT problem. We first construct an instance of the One-In-Three 3SAT problem U and C from U_0 and C_0 such that for each $u_i \in U$, literal u_i appears in at least one clause of C and literal $\overline{u}_i$ also appears in at least one clause of C.

We initially set $U = U_0$ and $C = C_0$. Suppose there exists $u_k \in U_0$ such that only one of the literals $u_k, \overline{u}_k$ appears in some clause of C_0. We then add to C four new clauses $d_1 = \{u_k, \overline{u}_k, a\}$, $d_2 = \{\overline{a}, a, b\}$, $d_3 = \{\overline{b}, \overline{c}, a\}$, and $d_4 = \{c, b, a\}$. In the mean time, new variables a, b, and c are added to U. It is now observed that assigning the false value to a and b and the true value to c makes that each of the clauses $d_1 - d_4$ has exactly one true literal. Therefore, there exists a truth assignment for U_0 such that each clause in C_0 has exactly one true literal if and only if there exists a truth assignment for U such that each clause in C has exactly one true literal. By repeating the above process, we obtain $U = \{u_1, u_2, \ldots, u_n\}$ and $C = \{c_1, c_2, \ldots, c_m\}$ such that for each $u_i \in U$, literal u_i appears in at least one clause of C and literal $\overline{u}_i$ also appears in at least one clause of C. Define an order "<" on $U \cup \{\overline{u} | u \in U\}$ by $u_1 < \overline{u}_1 < u_2 < \overline{u}_2 < \cdots < u_n < \overline{u}_n$ so that three literals of each clause of C are ordered.

We next construct a graph $G(V, E)$ from U and C as follows. The vertex set of G is $V(G) = X \cup Y \cup D \cup Z \cup A \cup B \cup \Delta$, where

$$X = \{c_1^i, c_2^i, c_3^i | 1 \leq i \leq m\},$$
$$Y = \{d_{12}^i, d_{23}^i, d_{31}^i | 1 \leq i \leq m\},$$
$$D = \{u_i, \overline{u}_i | 1 \leq i \leq n\},$$
$$Z = \{z_i | 1 \leq i \leq n\},$$
$$A = \{a_i | 1 \leq i \leq m\},$$
$$B = \{b_i | 1 \leq i \leq n\}, \text{ and}$$
$$\Delta = \{\delta_i^i, \delta_2^i, \delta_3^i | 1 \leq i \leq m\}.$$

The edge set of G is $E(G) = H \cup F \cup W \cup T \cup Q \cup \pi \cup R \cup L$, where

$H = \{c_1^i, c_2^i), (c_2^i, c_3^i), (c_3^i, c_1^i), (d_{12}^i, c_1^i), (d_{12}^i, c_2^i), (d_{23}^i, c_2^i), (d_{23}^i, c_3^i), (d_{31}^i, c_3^i), (d_{31}^i, c_1^i) | 1 \leq i \leq m\}$,

$F = \{(c_j^i, c_k^{i+1}) | 1 \leq i \leq m-1, 1 \leq j, k \leq 3\}$,

$W = \{(u_i, \overline{u}_i), (u_i, z_i), (\overline{u}_i, z_i) | 1 \leq i \leq n\}$,

$T = \{(u_i, u_{i+1}), (\overline{u}_i, u_{i+1}), (u_i, \overline{u}_{i+1}), (\overline{u}_i, \overline{u}_{i+1}) | 1 \leq i \leq n-1\}$,

$Q = \{(c_a^j, u_i) |$ the ath literal of clause c_j is u_i, for $1 \leq j \leq m$, $1 \leq a \leq 3$, $1 \leq i \leq n\} \cup \{(c_a^j, \overline{u}_i) |$ the ath literal of clause c_j is $\overline{u}_i$, for $1 \leq j \leq m$, $1 \leq a \leq 3, 1 \leq i \leq n\}$,

$\pi = \{(\delta_a^i, c_a^i) | 1 \leq i \leq m, 1 \leq a \leq 3\} \cup \{(\delta_a^i, w) | 1 \leq i \leq m, 1 \leq a \leq 3, (c_a^i, w) \in Q\}$.

$R = \{(u_i, a_j) |$ literal u_i appears in clause c_j for $1 \leq i \leq n$ and $1 \leq j \leq m\} \cup \{(\overline{u}_i, a_j) |$ literal $\overline{u}_i$ appears in clause c_j for $1 \leq i \leq n$ and $1 \leq j \leq m\}$, and

$L = \{(b_i, c_a^j) | j$ is the smallest integer such that the ath literal of clause c_j is u_i, $1 \leq i \leq n\} \cup \{(b_i, c_{a'}^{j'}) | j'$ is the smallest integer such that the a'th literal of clause $c_{j'}$ is $\overline{u}_i$, $1 \leq i \leq n\}$.

As an example, consider $U = \{u_1, u_2, u_3\}$ and $C = \{c_1, c_2, c_3\}$, where $c_1 = \{u_1, u_2, u_3\}$, $c_2 = \{\overline{u}_1, \overline{u}_2, \overline{u}_3\}$, $c_3 = \{\overline{u}_1, u_2, \overline{u}_3\}$. The graph G constructed from U and C is shown in Figure 2.

We next prove that there exists a truth assignment for U such that exactly one literal of each clause is true if and only if there exists a subset $S \subset V(G)$ such that conditions (a) - (c) are satisfied, where $\alpha = m + n$.

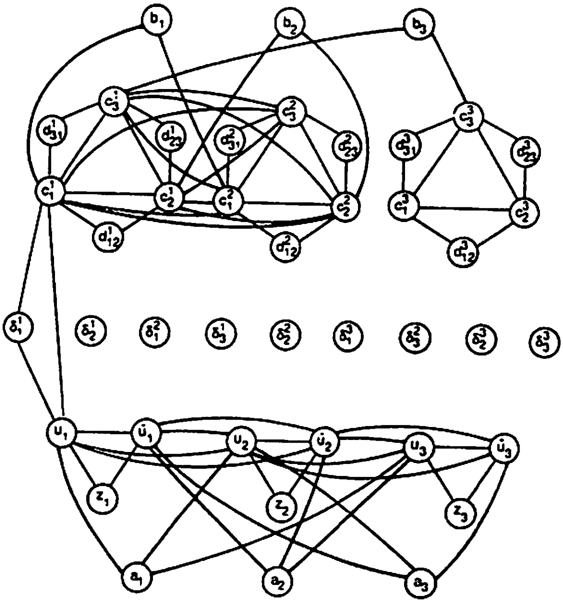

Figure 2. Construction of $G(V, E)$ from $c_1 = \{u_1, u_2, u_3\}, c_2 = \{\overline{u}_1, \overline{u}_2, u_3\}$ and $c_3 = \{\overline{u}_1, u_2, \overline{u}_3\}$. $E(G)$ consists of the edges shown, and edges $\{(c_j^2, c_k^3) \mid 1 \leq j, k \leq 3\} \cup \{(\delta_2^1, c_2^1), (\delta_2^1, u_2), (c_2^1, u_2), (\delta_1^2, c_1^2), (\delta_1^2, \overline{u}_1), (c_1^2, \overline{u}_1), (\delta_3^1, c_3^1), (\delta_3^1, u_3), (c_3^1, u_3), (\delta_2^2, c_2^2), (\delta_2^2, \overline{u}_2), (c_2^2, \overline{u}_2), (\delta_1^3, c_1^3), (\delta_1^3, \overline{u}_1), (c_1^3, \overline{u}_1), (\delta_3^2, c_3^2), (\delta_3^2, u_3), (c_3^2, u_3), (\delta_2^3, c_2^3), (\delta_2^3, u_2), (c_2^3, u_2), (\delta_3^3, c_3^3), (\delta_3^3, \overline{u}_3), (c_3^3, \overline{u}_3)\}$.

Suppose there exists such a truth assignment. Let $S = \{c_\alpha^i \mid$ the αth literal of clause c_i is true, $1 \leq i \leq m\} \cup \{\overline{u}_i \mid$ variable u_i is true, $1 \leq i \leq n\} \cup \{u_i \mid$ variable u_i is false, $1 \leq i \leq n\}$. Clearly, $G - S$ is disconnected, there exists no vertex in G whose neighbors are all in S, and $|S| = m + n$. Thus, S is a desired subset. In Figure 2, $S = \{c_1^1, c_2^2, c_3^3, \overline{u}_1, u_2, u_3\}$ when u_1 is true and u_2, u_3 are false.

Conversely, assume that there exists a subset $S \subset V(G)$ such that $|S| = m + n$ and conditions (a) and (b) are satisfied. We then make the following observations: (i) at most one vertex in $\{c_1^i, c_2^i, c_3^i\}$ for $1 \leq i \leq m$ belongs to S; (ii) at most one vertex in $\{u_i, \overline{u}_i\}$ for $1 \leq i \leq n$ belongs to S, since otherwise vertex z_i has all its neighbors in S.

The observation (i) together with the existence of edge set F implies that all the vertices in $X - S$ belong to one component of $G - S$. Similarly, the observation (ii) together with the existence of edge set T implies that all the vertices in $D - S$ belong to one component of $G - S$. We thus conclude that vertices in $X - S$ and vertices in $D - S$ belong to different components of $G - S$ and that for each edge $(u, v) \in Q$, either $u \in S$ or $v \in S$, but $\{u, v\} \not\subset S$ because of the existence of the edge set π. Therefore, if $\{c_1^j, c_2^j, c_3^j\} \cap S = \emptyset$ for some j, then the three vertices in D corresponding to literals c_1^j, c_2^j, and c_3^j must be a subset of S. However, those vertices are neighbors of a_j, a contradiction. Hence, we must have that

$|\{c_1^j, c_2^j, c_3^j\} \cap S| = 1$ for all $1 \le j \le m$.

Now, assume that neither u_i nor $\overline{u}_i$ is in S for some i. Then, all the vertices in X which are adjacent to u_i or $\overline{u}_i$ must be in S. However, the existence of vertex b_i prohibits it since the neighbors of b_i cannot be all in S. Thus, we have that $|\{u_i, \overline{u}_i\} \cap S| = 1$ for all $1 \le i \le n$. Consequently, the existence of a desired truth assignment for U is established. In particular, u_i is true if and only if $u_i \notin S$. This completes the proof of Theorem 2. ∎

The formulation of Problem 2 was motivated by the question that if a graph G is the underlying topology of a MC, then what is the least number of processor failures required to render the system unfunctional (i.e., disconnected) provided that for each processor p in the system all its neighboring processors do not fail at the same time? It should also be stated that the r-edge-connectivity version of the above problem can be solved in polynomial time [EsHa88].

Theorem 3. *Problem 3 is NP-complete.*

Proof: The transformation is again from the One-In-Three 3SAT problem. The construction of a graph $G'(V, E)$ from the given instance U and C is very similar to that of $G(V, E)$ in the proof of Theorem 2. Thus, we only describe the necessary modifications as follows. The vertex set of G' is $V(G') = V(G) \cup (\cup_{j=1}^{t}(Y_j \cup Z_j \cup A_j \cup B_j \cup \Delta_j))$, where $t = m + n - 2$ and

$$\begin{aligned}
Y_j &= \{d_{12}^{ij}, d_{23}^{ij}, d_{31}^{ij} \mid 1 \le i \le m\},\\
Z_j &= \{z_i^j \mid 1 \le i \le n\},\\
A_j &= \{a_i^j \mid 1 \le i \le m\},\\
B_j &= \{b_i^j \mid 1 \le i \le n\}, \text{ and}\\
\Delta_j &= \{\delta_1^{ij}, \delta_2^{ij}, \delta_3^{ij} \mid 1 \le i \le m\}.
\end{aligned}$$

The edge set of G' is $E(G') = E(G) \cup E_Y \cup E_Z \cup E_A \cup E_B \cup E_\Delta$, where $t = m + n - 2$ and

$$\begin{aligned}
E_Y &= \cup_{i=1}^{m}\{(d_{12}^{ij}, u), (d_{23}^{ij}, v), (d_{31}^{ij}, w) \mid (d_{12}^{i}, u), (d_{23}^{i}, v), (d_{31}^{i}, w) \in E(G),\\
&\quad 1 \le j \le t\},\\
E_Z &= \cup_{i=1}^{n}\{(z_i^j, v) \mid (z_i, v) \in E(G), 1 \le j \le t\},\\
E_A &= \cup_{i=1}^{m}\{(a_i^j, v) \mid (a_i, v) \in E(G), 1 \le j \le t\},\\
E_B &= \cup_{i=1}^{n}\{(b_i^j, v) \mid (b_i, v) \in E(G), 1 \le j \le t\}, \text{ and}\\
E_\Delta &= \cup_{i=1}^{m}\{(\delta_1^{ij}, u), (\delta_2^{ij}, v), (\delta_3^{ij}, w) \mid (\delta_1^{i}, u), (\delta_2^{i}, v), (\delta_3^{i}, w) \in E(G),\\
&\quad 1 \le j \le t\},
\end{aligned}$$

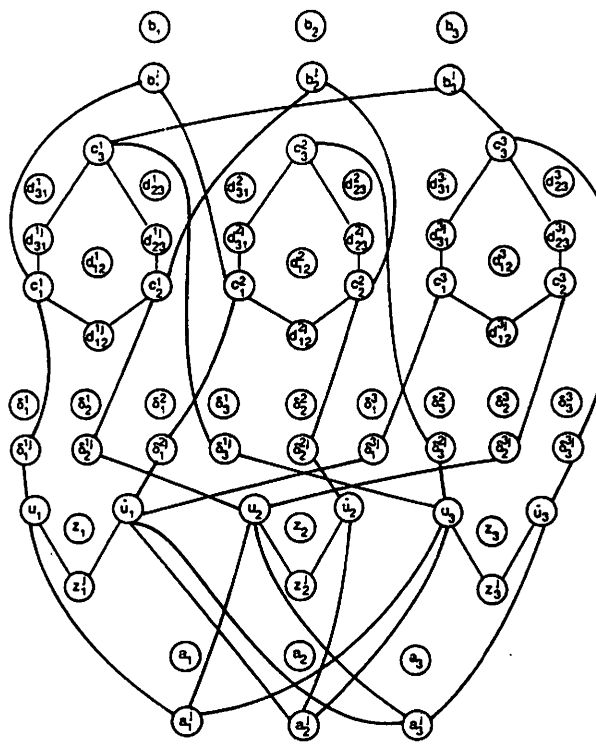

Figure 3. Construction of $G'(V,E)$ from $c_1 = \{u_1, u_2, u_3\}$, $c_2 = \{\overline{u}_1, \overline{u}_2, u_3\}$ and $c_3 = \{\overline{u}_1, u_2, \overline{u}_3\}$. The vertex set of G' is $V(G') = V(G) \cup \Psi$, where $V(G)$ is the vertex set of G shown in Figure 2 and $\Psi = \{d_{12}^{ij}, d_{23}^{ij}, d_{31}^{ij}, z_i^j, a_i^j, b_i^j, \delta_1^{ij}, \delta_2^{ij}, \delta_3^{ij} | 1 \leq i \leq 3, 1 \leq j \leq 4\}$. In the figure, d_{31}^{1j} represents four vertices $d_{31}^{11}, d_{31}^{12}, d_{31}^{13}, d_{31}^{14}$, each of which has neighboring vertices c_3^1, c_1^1. Similarly, each vertex in $\Psi - \{d_{31}^{1j}\}$ represents four vertices which have the same neighboring vertices. The edge set $E(G)$ shown in Figure 2 is the remaining edges in $E(G')$.

Figure 3 shows the graph $G'(V,E)$ constructed from the same example of U and C in Figure 2 by following the above modifications. We set again $\alpha = m + n$.

In order to prove the correctness of the transformation, we first note that the necessary part is same as before, and to complete the proof we make the following observations. If both u_i and $\overline{u}_i$ belong to S for some i, then all the $t+1$ vertices $z_i, z_i^1, z_i^2, \ldots, z_i^t$ must belong to S. Since $|S| = m + n$, this cannot happen. In fact, exactly one of u_i and $\overline{u}_i$ must be in S from the same reasoning as before. Similarly, exactly one of c_1^j, c_2^j, and c_3^j must be in S for all $1 \leq j \leq m$. Now, the rest of the proof follows. ∎

4. Polynomial-Time Special Cases

In this section, we show that Problems 2 and 3 can be solved in polynomial-time if graph G is a k-tree. Further, we give efficient algorithms that find the corresponding disconnecting sets S. We proceed with some definitions and terminology.

A complete graph on k vertices is called a k-clique. For a vertex v in a graph $G(V, E)$, the *degree* of v in G will be denoted by $\deg_G(v)$. The set of all vertices that are adjacent to v in G will be denoted by $A_G(v)$. The graph *induced* by a set $X \subset V$ will be denoted by $[X]$. A vertex v is a *k-leaf* of G if $\deg_G(v) = k$ and $[A_G(v)]$ is a k-clique. The set of all k-leaves of G will be denoted by $L(G)$.

A set $S \subset V$ is called a *vertex-cut* (or simply a *cut*) if $G - S$ is disconnected (note that by this definition, a complete graph has no cut). A cut S is a *minimum-cut* if for any other cut X we have $|S| \leq |X|$. A cut S is called a *restricted-cut* if for no vertex v in G we have $A(v) \subseteq S$. A restricted-cut S is a *minimum restricted-cut* if for any other restricted-cut X we have $|S| \leq |X|$. A cut S is called a *conditional-cut* if either S is a restricted-cut, or S also contains every vertex v for which $A_G(v)$ is a subset of S. That is, for any conditional cut S of G, there exists no isolated vertex in $G - S$. A conditional-cut S is a *minimum conditional-cut* if for any other conditional-cut X we have $|S| \leq |X|$. A cut S having a property ρ (e.g., restricted, conditional) is *minimal* if there is no $X \subset S$ with property ρ.

A k-tree is defined recursively as follows. A graph $G(V, E)$ is a *k-tree* if G is either a k-clique, or G contains a k-leaf v such that $G - \{v\}$ is a k-tree. Note that if $G(V, E)$ is a k-tree then $\deg_G(v) \geq k$ for every vertex $v \in V$, and $\deg_G(v) = k$ only if $v \in L(G)$.

Lemma 1. *Let $G(V,E)$ be a k-tree with $|V| > k + 1$. Then $|L(G)| \geq 2$. Further, no two vertices belonging to $L(G)$ are adjacent in G.*

Proof: Suppose that two k-leaves v_1 and v_2 of G are adjacent. Then, $\deg_G(v_2) = k$ implies that $\deg_{G-\{v_1\}}(v_2) = k - 1$. But, $G - \{v_1\}$ is a k-tree, and so $\deg_{G-\{v_1\}}(v_2) \geq k$. Hence, no two k-leaves of G are adjacent.

Next, we show by induction on $|V(G)|$ that G contains at least two k-leaves. This is clear when $|V(G)| = k + 2$, and assume it is true for $|V(G)| = k + i$. Let $|V(G)| = k + i + 1$ and let v be a k-leaf of G. Let B be the set of k-leaves of $G - \{v\}$. Since $|B| \geq 2$ by the inductive hypothesis, and $[A_G(v)]$ must form a k-clique, we have that $B \not\subseteq A_G(v)$. Thus, $|B - A_G(v)| \geq 1$. Note that each $w \in B - A_G(v)$ is a k-leaf of G. Hence, G has at least two k-leaves. ∎

Theorem 4. *Let $G(V, E)$ be a k-tree with $|V(G)| > k + 1$. Then for any cut S in G there exists a cut X with $|X| = k$ such that $X \subseteq S$.*

Proof: Let S be a cut of G. We use induction on $|V(G)|$ to prove the theorem. If $|V(G)| = k+2$, the result is trivial. Suppose $|V(G)| > k+2$. We may assume that $S \cap L(G) = \emptyset$, since if $w \in S \cap L(G)$ then $S - \{w\}$ is a cut of G. Choose a vertex $v \in L(G)$ and let C be the component of $G - S$ containing v. If $|C| = 1$ then $A_G(v) \subseteq S$, and so $A_G(v)$ is a desired cut of G. Next, let $|C| > 1$. Note that S is a cut of $G - \{v\}$ and $C - \{v\}$ is a component of $(G - \{v\}) - S$. By the inductive hypothesis, S contains a cut S_0 of $G - \{v\}$ with size k. It follows that S_0 is a desired cut of G. ∎

Corollary 1. *Let $G(V, E)$ be a k-tree with $|V(G)| > k+1$. Then $\kappa(G) = k$.*

Theorem 5. *Let G be a k-tree with $|V(G)| > k+1$, and let S be a minimum cut of G. Then $[S]$ is a k-clique. Further, each component of $G - S$ contains at least one k-leaf of G.*

Proof: Let v be a k-leaf of G, and observe that $v \notin S$. Let C be the component of $G - S$ containing v. Suppose $|C| = 1$. Then, since $|S| = k$ and $\deg_G(v) = k$, we see that $A_G(v) = S$ and thus $[S]$ is a k-clique. Now, assume that $|C| > 1$. We then reduce G to a k-tree $H \subseteq G$ where a k-leaf of H is isolated in $H - S$ by successively deleting a k-leaf from the reduced graph. We thus conclude that $[S]$ is a k-clique.

We will use induction on $|V(G)|$ to show the second part of the theorem. If $|V(G)| = k+2$ then it clearly holds. Assume it is true for $|V(G)| = k+i$. Let $|V(G)| = k+i+1$, and let $C_1, C_2, \ldots, C_t$ be all components of $G - S$. Note that if there exists i, $1 \leq i \leq t$, such that $|C_i| = 1 (C_i = \{v_i\})$, then we see that v_i is a k-leaf of G.

Case 1. $t \geq 3$: Let v be a k-leaf of G, and we may assume that $v \in C_1$. Then, each $C_i (i \geq 2)$ is a component of $(G - \{v\}) - S$ and so, by the inductive hypothesis, each C_i contains a k-leaf w_i of $G - \{v\}$. Since $w_i \notin A_G(v)$ by Lemma 1, we conclude that w_i is a k-leaf of G.

Case 2. $t = 2$: We may assume that $|C_1| > 1$ and C_1 contains a k-leaf v of G, since G has at least 2 k-leaves by Lemma 1. By the same arguments as in Case 1, we conclude that C_2 also contains a k-leaf of G.

This completes the proof of Theorem 5. ∎

4.1 Restricted-Cuts of k-trees

Theorem 6. *Let G be a k-tree. If there exists a restricted cut T of G, then T contains a restricted cut S of G such that $[S]$ is a k-clique, and for any such S, $|C \cap \{v \in V(G) | S \subseteq A_G(v)\}| = 1$ for each component C of $G - S$.*

Proof: Let T be a restricted cut of G. Let S be a restricted cut of G such that $S \subseteq T$ and $|S|$ is minimum. Note that $|S| \geq k$, since any restricted cut is a cut and $\kappa(G) = k$ by Corollary 1. If $A_G(x) \cap C = \emptyset$, for an $x \in S$ and a component C of $G - S$, then $S - \{x\}$ is also a restricted cut of G. Thus, we have $A_G(x) \cap C \neq \emptyset$ for each $x \in S$ and each component C of $G - S$. In particular, S does not contain a k-leaf of G.

Now, we show that $|S| = k$, it will follow then from Theorem 5 that $[S]$ is a k-clique. For a k-leaf v of G, we denote by C_v the component of $G - S$ containing v.

Case 1: $|C_v| = 2$ for each k-leaf v of G: Take two k-leaves v and w of G, and let $C_v = \{v, \alpha\}$, and $C_w = \{w, \beta\}$. Note that $(v, \alpha) \in E(G)$, and for any $x \in S$, if $(v, x) \in E(G)$ then $(\alpha, x) \in E(G)$, since $[A_G(v)]$ is a k-clique. Thus, each vertex in S is adjacent to α and similarly, to β. Note that α and β are not adjacent in G. Now, suppose $|S| > k$. Then, G can be reduced to $S \cup \{\alpha, \beta\}$ by successively removing $|V(G)| - (|S| + 2)$ vertices of degree k in the reduced graphs, respectively. Further, S contains a k-leaf of the k-tree $S \cup \{\alpha, \beta\}$, which implies that α and β are adjacent, a contradiction. Hence, $|S| = k$.

Case 2: $|C_v| > 2$ for a k-leaf v of G: Observe that S is a restricted cut of k-tree $G - \{v\}$, where $C_v - \{v\}$ is a component of $(G - \{v\}) - S$. We claim that S does not properly contain a restricted cut of $G - \{v\}$. Let $S_0 \subseteq S$ be a restricted cut of $G - \{v\}$. Then, it follows from the fact that v is a k-leaf of G that S_0 is also a restricted cut of G. The choice of S thus yields $S_0 = S$ as desired. Thus, we can reduce G to a k-tree $G_1 \subset G$ such that (i) S is a restricted cut of G_1, (ii) each component of $G_1 - S$ consists of exactly two vertices, and (iii) S does not properly contain a restricted cut of G_1. Now, G_1 with a restricted cut S satisfies the conditions for Case 1, hence $|S| = k$.

Next, we show $|C \cap \{v \in V(G) | S \subseteq A_G(v)\}| = 1$ for each component C of $G - S$. Let C be a component of $G - S$. Note that C contains an x of degree k by Theorem 5 and $C - \{x\}$ is a component of $(G - \{x\}) - S$. So, C can be reduced to a single vertex v_0 by successively removing $(|C| - 1)$ vertices of degree k from C. Since $S \subseteq A_G(v_0)$, we have $|C \cap \{v \in V(G) | S \subseteq A_G(v)\}| \geq 1$. Now, suppose that C contains two vertices v and w such that $S \subseteq A_G(v)$ and $S \subseteq A_G(w)$. Then, we reduce C to $\{v, w\}$ by successively removing $(|C| - 2)$ vertices of degree k from C, and let G^* be the resulting k-tree. Since $\{v, w\}$ is a component of $G^* - S$, Theorem 5 yields that $\deg_{G^*}(v) = k$ or $\deg_{G^*}(w) = k$. This is a contradiction since v is adjacent to w, and hence the above inequality is actually equality. This completes the proof of Theorem 6. ∎

Remark: Theorem 6 implies that for any k-tree G, either there exists a restricted cut of G with size k or no restricted cut of G exists.

The following algorithm determines whether or not a restricted-cut of a given

k-tree G exists, and finds one whose cardinality is k if a restricted-cut of G exists.

Algorithm Restricted-Cut(G)
Input: a k-tree $G(V, E)$
Output: a restricted cut S of G with size k if G has a restricted cut, or $S = \emptyset$ otherwise

begin
1. Compute $L = \{v \in V(G) \mid \deg_G(v) = k\}$
2. Let $G_1 = G$; $F = \emptyset$
3. **while** ($L \neq \emptyset$ and $|V(G_1)| \geq k + 2$) **do**
4. Select an arbitrary vertex $v \in L$ and let $L \leftarrow L - \{v\}$
5. $F \leftarrow F \cup \{A_{G_1}(v)\}$
6. $G_1 \leftarrow G_1 - \{v\}$
7. Let $T = \{w \in V(G_1) \mid w \notin L, \deg_{G_1}(w) = k$, and $A_{G_1}(w) \in F\}$
8. $L \leftarrow L \cup T$
9. **endwhile**
10. **if** $|V(G_1)| \geq k + 2$
11. **then** **begin**
12. Select an arbitrary vertex $z \in V(G_1)$ such that $\deg_{G_1}(z) = k$;
13. $S \leftarrow A_{G_1}(z)$
14. **end**
15. **else** $S \leftarrow \emptyset$ **endif**
16. **return**(S)

end.

Theorem 7.
Algorithm Restricted-Cut(G) is correct and runs in $O(|V(G)| \log |V(G)|)$ *time.*

Proof: We first prove that a restricted cut of G is produced by the algorithm if and only if there exists a restricted cut of G. Assume that a restricted cut S of G is produced by the algorithm. Let G_1 be the resulting graph in line 10 of the algorithm. So, $S = A_{G_1}(z)$ for a k-leaf z of G_1. We show that S is a restricted cut of G. From the algorithm, we see that $A_G(v) \not\subset S$ for each $v \in V(G) - V(G_1)$. Each isolated vertex w in $G_1 - S$ satisfies $\deg_{G_1}(w) = k$ and $\deg_G(w) > k$, so w is not isolated in $G - S$ and thus $A_G(w) \not\subset S$. We conclude that $A_G(v) \not\subset S$ for each $v \in V(G)$. Now, we need only show that S is a cut of G. Since G_1 is a k-tree with $|V(G_1)| \geq k + 2$, $G_1 - S$ has at least two components C and D. It follows from the algorithm that for each $x \in C$ and each $y \in D$, there exists no path from x to y in G. Hence, $G - S$ has at least two components and thus S is a cut of G.

Conversely, assume that there exists a restricted cut of G. Then it contains a restricted cut T such that $[T]$ is a k-clique by Theorem 6. Let C_i, for $1 \leq i \leq t$, be the components of $G-T$. Then, by Theorem 6, there exists exactly one $v_i \in C_i$ such that $T \subseteq A_G(v_i)$, for each $1 \leq i \leq t$. Let G' be the subgraph of G induced by $T \cup \{v_i | 1 \leq i \leq t\}$. Clearly, G' is a k-tree and $|V(G')| = k+t \geq k+2$. Now, let G_1 be the resulting graph in line 10 of the algorithm. Then, $G' \subseteq G_1$ since any vertex in $V(G')$ remains undeleted by the algorithm. Hence, a restricted cut of G with size k is produced by the algorithm.

To show the time complexity of the algorithm, it is observed that the *while* loop in lines 3 - 9 is executed $O(|V(G)|)$ times and the bottleneck on each iteration is to check whether $A_{G_1}(w) \in F$ in line 7. Line 7 can be done in $O(\log |V(G)|)$ time using a data structure for UNION-FIND operations. This would imply the desired result. ∎

4.2 Conditional-Cuts of k-trees

It is observed that given a graph G, if S is a restricted cut of G, then S is a conditional cut of G. However, the converse does not always hold. Therefore, this observation together with Theorem 6 implies that given a k-tree G, there exists a restricted cut of G if and only if there exists a conditional cut S of G such that $|S| = k$.

Theorem 8. *Let G be a k-tree. Then, there exists a conditional cut of G if and only if $|V(G) - L(G)| \geq k+2$.*

Proof: Assume a conditional cut S of G. Then, since no two k-leaves of G are adjacent, each component of $G - S$ contains at least one non-leaf vertex. On the other hand, $S - L(G)$ is a cut of the k-tree $G - L(G)$ and so $|S - L(G)| \geq k$ by Theorem 4. We conclude that G contains at least $k+2$ non-leaf vertices.

Conversely, assume $|V(G) - L(G)| \geq k+2$, and let $G' = G - L(G)$. Choose a k-leaf x of the k-tree G' and let $S' = A_{G'}(x)$. Since $|V(G')| \geq k+2$, $G' - S'$ has at least two components, say C' and D', and let C and D be two components of $G - S'$ such that $C' \subseteq C$ and $D' \subseteq D$. Note that $|C| > 1$, $|D| > 1$, and $C \cap D = \emptyset$. Since $x \in V(G)$ is isolated in $G - S'$ if and only if $x \in L(G)$ and $A_G(x) = S'$, we conclude that $S' \cup \{x \in L(G) | A_G(x) = S'\}$ forms a conditional cut of G. This completes the proof. ∎

Lemma 2. *Let G be a k-tree with $|V(G) - L(G)| \geq k+2$. Let $G^* \subseteq G - L(G)$ be a k-tree with $|V(G^*)| \geq k+2$ obtained from $G - L(G)$ by successively removing zero or more vertices of degree k. Let $v \in L(G^*)$ and $W = \{x \in L(G) | A_G(x) = A_{G^*}(v)\}$. Then*

(a) *$A_{G^*}(v)$ is a cut of G.*
(b) *Any conditional cut of G containing $A_{G^*}(v)$ also contains W.*
(c) *$A_{G^*}(v) \cup W$ is a minimal conditional cut of G.*
(d) *$A_{G^*}(v)$ is a restricted cut of G if and only if $W = \emptyset$.*

Proof: (a) is immediate from the construction of G^* and the choice of v. For any conditional cut S of G containing $A_{G^*}(v)$, if $x \in W - S$ then $G - S$ isolates x, and thus $W \subset S$ and (b) is proved. Since $w \in W$ if and only if w is isolated in $G - A_{G^*}(v)$, we see that $A_{G^*}(v) \cup W$ is a conditional cut of G. To show the minimality of $A_{G^*}(v) \cup W$, let $S_0 \subseteq A_{G^*}(v) \cup W$ be a conditional cut of G. Since $S_0 - L(G)$ is a cut of the k-tree $G - L(G)$, $|S_0 - L(G)| \geq k$ by Theorem 4 and so $S_0 - L(G) = A_{G^*}(v)$. Now, it follows from (b) that $S_0 = A_{G^*}(v) \cup W$. The proof of (c) is now complete, and (d) follows from (c) since $|A_{G^*}(v)| = k$.

∎

Lemma 3. *Let S be a cut of a k-tree of G. Then, G can be reduced to a k-tree G^* by successively removing zero or more vertices of degree k such that $|V(G^*)| \geq k+2$ and $A_{G^*}(v) \subseteq S$ for some $v \in L(G^*)$.*

Proof: We proceed by induction on $|V(G)|$. If $|V(G)| = k + 2$, the result is trivial. Suppose $|V(G)| > k + 2$. We may assume that S is a minimal cut of G, so that $S \cap L(G) = \emptyset$. Now, let $w \in L(G)$ and denote by C the component of $G - S$ containing w. If $|C| = 1$, then we are done. If $|C| > 1$, we apply the inductive hypothesis to the cut S of the k-tree $G - \{w\}$, and the result follows since $w \in L(G)$. ∎

Suppose now that S is a conditional cut of a k-tree G. Then since $S - L(G)$ is a cut of the k-tree $G - L(G)$, Lemma 3 applied to the cut $S - L(G)$ of $G - L(G)$ implies that $G - L(G)$ can be reduced to a k-tree $G^* \subseteq G - L(G)$ by the successively removing process, where $|V(G^*)| \geq k+2$ and $A_{G^*}(v) \subseteq S - L(G)$ for some $v \in L(G^*)$.

Next, suppose that S is a minimal conditional cut of G. Then, by Lemma 2 (b), $A_{G^*}(v) \cup W \subseteq S$ where $W = \{x \in L(G) | A_G(x) = A_{G^*}(v)\}$. Now, Lemma 2 (c) and the minimality of S yield that $A_{G^*}(v) \cup W = S$, hence we have $S - L(G) = A_{G^*}(v)$ and $S \cap L(G) = W$. Note that $A_{G^*}(v)$ is now uniquely determined by S; we will denote the $A_{G^*}(v)$ by $K(S)$. Finally, note that if S is also a restricted cut of G then $W = \emptyset$; the converse also holds by Lemma 2(d). The above observations establish Lemma 4.

Lemma 4. *Let G be a k-tree with $|V(G) - L(G)| \geq k + 2$, and let S be a minimal conditional cut of G. Then,*

(a) $S \cap L(G) = \{x \in L(G) | A_G(x) = K(S)\}$,
(b) $S - L(G) = K(S)$, *and*
(c) *S is a restricted cut of G if and only if $S \cap L(G) = \emptyset$.*

We obtain from Lemmas 2 - 4 the following theorem.

Theorem 9. *Let G be a k-tree with no restricted cuts. Suppose $|V(G) - L(G)| \geq k + 2$, and let $S \subset V(G)$. Then, S is a minimal conditional cut of G if and only if $S = A_G(v) \cup \{x \in L(G) | A_G(x) = A_G(v)\}$ for some $v \in L(G)$ such that*

$A_G(v)$ is a cut of $G - L(G)$. (Note that v itself is included in S since v can be x.)

Proof: The necessity is immediate by Lemma 4. The sufficiency is proven by applying Lemma 3 to the cut $A_G(v)$ of $G - L(G)$ and then by Lemma 2 (c). ∎

Now, we obtain the following algorithm that determines whether or not a conditional cut of a given k-tree G exists, and finds one whose cardinality is minimum if a conditional cut of G exists.

Algorithm Conditional-Cut(G)
Input: a k-tree $G(V, E)$
Output: a conditional cut T of G whose cardinality is minimum if a conditional cut of G exists, or $T = \emptyset$ otherwise

begin
1. Compute $L = \{v \in V(G) | \deg_G(v) = k\}$
2. **if** $|V(G) - L| < k + 2$ **then** $T \leftarrow \emptyset$; goto 16 **endif**
3. Call *Restricted-Cut*(G)
4. **if** the output S of the *Restricted-Cut*(G) is not empty
5.   **then** $T \leftarrow S$; **goto 16 endif**
6. $M \leftarrow \{v \in L | A_G(v)$ is a cut of $G - L\}$
7. $T \leftarrow V(G)$
8. **while** $M \neq \emptyset$ **do**
9.   Let v be a vertex in M
10. $D \leftarrow A_G(v)$ and $W \leftarrow \{x \in M | A_G(x) = D\}$
11. $Q \leftarrow D \cup W$
12.   **if** $|W| = 1$ **then** $T \leftarrow Q$; **goto 16 endif**
13.   **if** $|Q| < |T|$ **then** $T \leftarrow Q$ **endif**
14. $M \leftarrow M - W$
15. **endwhile**
16. **return**(T)
end.

Theorem 10. *Algorithm Conditional-Cut(G) is correct and runs in $O(|V(G)|^2)$ time.*

Proof: The correctness of the algorithm is immediate by Theorem 9. To analyze the time complexity, note that line 3 can be done in $O(|V(G)| \log |V(G)|)$ time by Theorem 7. It is also observed that $|L| = O(|V(G)|)$, and for each vertex $v \in L$, checking whether or not $A_G(v)$ is a cut of $G-L$ can be done in $O(|V(G)|)$ time. Thus, line 6 can be done in $O(|V(G)|^2)$ time. The *while* loop in lines 8 -15 is executed at most $|V(G)|$ times, and on each iteration, line 10 can be done in $O(|V(G)|)$ time. This would imply the desired result. ∎

References

[Boes86] F.T. Boesch, *Synthesis of reliable networks - A survey*, IEEE Trans. on Reliability R-35, No. 3, 240–246. August (1986).

[DuHw88] D.Z. Du and F.K. Hwang, *Generalized de Bruijn Digraphs*, NETWORKS 18 (1988), 27–38.

[Esfa88] A.-H. Esfahanian, *Generalized Measures of Fault-Tolerance with Application to n-Cube Networks*, IEEE Transactions on Computers 38, 1586–1591. November (1989).

[EsHa88] A.-H. Esfahanian and S.L. Hakimi, *On Computing a Conditional Edge-connectivity of a Graph*, Journal of Information Processing Letters 27, 195–199. April (1988).

[Haye76] J.P. Hayes, *A Graph Model for Fault-Tolerant Computing Systems*, IEEE Trans. on Comput. 25, No. 9, 875–884. September (1976).

[KuRe80] J.G. Kuhl and S.M. Reddy, *Distributed Fault-Tolerance for Large Multiprocessor Systems*, Proc. 7th Annu. Symp. Comput. Architecture, 23–30. May (1980).

[LoFu87] M.B. Lowrie and W.K. Fuchs, *Reconfigurable Tree Architectures using Subtree Oriented Fault Tolerance*, IEEE Trans. on Comput. 36, 1172–1183. October (1987).

[Oell86] O.R. Oellermann, *Generalized Connectivity in Graphs*, PhD. Dissertation. Dept. of Mathematics, Western Michigan University, Kalamazoo, MI, August (1986).

[PrRe82] D.K. Pradhan and S.M. Reddy, *A Fault-Tolerant Communication Architecture, for Distributed Systems*, IEEE Trans. on Comput. C-31, No. 9, 863–869. September (1982).

[Tane81] A.S. Tanenbaum, "Computer Networks", Prentice Hall, 1981.

[ZaEl88] M. Zaki and M.M. Elboraey, *Analysis of Reliability Models for Interconnection of MIMD Systems*, The Cormputer Journal 31 (1988), 304–312.