Algorithms for the Computation of Communication Network Vulnerability Indexes

Luis Petingi, María Urquhart
Investigación Operativa *
Instituto de Computación, PeDeCiBa Informática
Facultad de Ingenería, Montevideo, Uruguay

March 29, 1996

Abstract

A comunication network may be represented as a graph G, where the nodes of the network (hosts, packet switches) and its communication links are modelled as vertices and edges of the graph respectively. The *vulnerability* of a communication network is defined as the measurement of the global strength of its underlying graph. The purpose of this paper is to present two vulnerability indexes associated with a graph and the corresponding algorithms, of polynomial time complexity, for computing these functions.

The first vulnerability index we propose is $t(G)$, the *number of spanning trees* of a graph G. A model extensively used to determine the reliability of a network is the one in which the edges of its underlying graph G survives with equal but independent probability ρ (nodes are perfect). The reliability $R(G, \rho)$ is then defined as the probability that the surviving edges induce a spanning connected subgraph of G. Let $\Omega(n, e)$ represent the class of graphs with n nodes and e edges. Given two graphs G and G' which belong to the same class $\Omega(n, e)$, if $t(G) > t(G')$ then there exist a small value $\rho_0 > 0$ such that for $0 < \rho \leq \rho_0, R(G, \rho) > R(G', \rho)$. Thus the number of spanning trees of a graph represents a good measurement of vulnerability.

The second vulnerability index of a graph G, here called *edge-toughness* $\eta(G)$ tell us that in order to split a graph G into $k + \omega(G)$ components, where $\omega(G)$ represents the number of connected components of G, we must then remove at least $k\eta(G)$ edges from G, thus $\eta(G)$ measures how tough it is to break up G.

1 Introduction

The design a good communication newtwork must take into account notions such as reliability and vulnerability. The reliability of a network is a probabilistic model in which its

*This paper is sponsored by CONICYT

components (hosts, packets switches, comunications lines, etc) are subject to random failures (defective components, natural accidents, etc). In choosing a topology for a communication network, it is desirable to minimize the impact of this failures by selecting, among the competing topologies, those which will minimize the likelihood of the network loosing its ability to route messages from one node to any other node.
On the other hand, a topology is said to be less vulnerable than a competing topology, if more components than its competitor must be subject to failures, in order to alter certain graph theoretical properties (i.e. number of connected components).

Even though in this paper our primary concern is with vulnerability, we establish a link between the concept of reliability and the concept of vulnerability, and that is the number of spanning trees of a graph. Given that the edges of a graph G are subject to independent but equal probabilities of survival ρ, let $R(G, \rho)$ denote the probability that the surviving edges induce a spanning connected subgraph of G. Given two graphs $G, G' \in \Omega(n, e)$, if $t(G) > t(G')$ then there exist $\rho_0 > 0$ such that for any $0 < \rho \leq \rho_0$, $R(G, \rho) > R(G', \rho)$. Thus a graph with a large number of spanning trees is less likely to be disconnected given that its edges are subject to failures. Thus the number of spanning trees represent a good measurement of vulnerability. Moreover the number of spanning trees of a graph G can be obtained by computing the determinant of a graph related matrix known as the *nodal admittance* matrix, thus can be computed in polynomial time.

The second vulnerability index of a graph G, here called *edge-toughness* $\eta(G)$, was originally introduced by Gusfield [8]. The definition of $\eta(G)$ tell us that in order to split a graph G into $k + \omega(G)$ components, where $\omega(G)$ represents the number of connected components of G, we must then remove at least $k\eta(G)$ edges from G, thus $\eta(G)$ measures how tough it is to break up G.

In [24] we had proposed an index of vulnerability:

$$Vul(G) = \frac{D + CE(c) + CV(c')}{t + c.c'} \tag{1}$$

where:

- t: is the girth of the graph G.
- c: is the size of the the minimum edge disconnecting set of the graph.
- c': is the size of the the minimum point disconnecting set of the graph.
- D : is the diameter of the graph.
- $CE(c)$: is the maximum, taken over all pair of points (i, j), of the number of edge (i, j)-cuts of size c.
- $CV(c')$: is the maximum, taken over all pair of points (i, j), of the number of vertex (i, j)-cuts of size c'.

The edge connectivity c can be computed in polynomial time by application of network flow techniques [1, page 131].

Unfortunately, at the present, the existence of a polynomial time algorithm for the evaluation of $CE(c)$ is unknown. Given two vertices i, and j, let $c(i,j)$ represent the size of the minimum edge (i,j)-cut. Moreover $c(i,j) \geq c$ for all (i,j), thus in the determination of $CE(c)$ we are only concerned with those pair of vertices (i,j) such that $c(i,j) = c$. Even though $c(i,j)$ is calculated in polynomial time, Ball and Provan [6] showed that calculating the number of (i,j)-cuts of size $c(i,j)$ is #P. Thus the complexity for the determination of $CE(c)$ is most likely #P.

Computation of both the number of spanning trees and edge-toughness of a graph G are of polynomial-time complexity thus these indexes are more useful from a practical point of view.

2 Spanning Trees

A model extensively used to determine the reliability of a network is the one in which the edges of its underlying graph G, survive with equal but independent probability ρ (nodes are perfect). The reliability $R(G,\rho)$ is then defined as the probability that the surviving edges induce a spanning connected subgraph of G. It is easily seen that $R(G,\rho)$ is given by the polynomial in ρ:

$$\sum_{j=n-1}^{e} S_j \rho^j (1-\rho)^{e-j} \tag{2}$$

where S_j is the number of spanning connected subgraphs of G with j edges and e is the number of edges of G. Clearly S_{n-1} is the number of spanning trees of G.

The number of spanning trees of a graph G, $t(G)$, can be used as a measure of vulnerability of G, since intuitively, among competing graphs with the same number of vertices and edges, one with a large number of spanning trees is probably better suited to withstand failures of its edges. This intuitive idea can be made more precise using (2) by noting that, if $G, G' \in \Omega(n,e)$ and $t(G) \geq t(G')$ then there is some $\rho_0 > 0$ such that $R(G,\rho) > R(G',\rho)$ for all ρ with $0 < \rho \leq \rho_0$, where $\Omega(n,e)$ are the class of non-isomorphic graphs with n nodes and e edges.

A graph G is said to be *t-optimal* iff $t(G) \geq t(G'), \forall G' \in \Omega(n,e)$. Even though to determine t-optimal graphs for arbitrary (n,e) have resisted the assault of many mathematicians, some results are known for particular values of n and e. For literature concerning this subject refer to [17, 11, 13, 14].

2.1 Number of Spanning Trees

In this section we derive a formula for determining the number of spanning trees of a connected graph G. This formula is based on a theorem implicating a graph-theoretical matrix called the *incidence matrix* and also on a result in matrix theory, known as the Binet-Cauchy theorem.

Consider a multigraph G with n vertices and e edges, and let $\vec{G}$ represent an orientation, i.e. a directed multigraph obtained by assigning a direction to each edge of G. The incidence matrix $S(\vec{G}) = [s_{ij}]$ of $\vec{G}$ has n rows, one for each vertex, and e columns, one for each edge and the element s_{ij} of $S(\vec{G})$ is defined as follows:

$$s_{ij} = \begin{cases} 1 & : \text{ if the jth edge is incident on the ith vertex and oriented away from it} \\ -1 & : \text{ if the jth edge is incident on the ith vertex and oriented towards it} \\ 0 & : \text{ if the jth edge is not incident on the ith vertex} \end{cases}$$

From the previous definition we can conclude that each column of $S(\vec{G})$ has exactly two nonzero entries, one +1 and one -1. Thus we can obtain any row of $S(\vec{G})$ from the previous $n-1$ rows. Therefore the rows of $S(\vec{G})$ are linearly dependent.

Futhermore if we delete a row v (representing the vertex v of $\vec{G}$) the resulting matrix $S_v(\vec{G})$ is called a *reduced incidence matrix* .

The following two lemmas conclude with a corollary of significant importance.

Lemma 1 *[3, page 125] If T is a tree then $|S_v(\vec{T})| = \pm 1$.*

Lemma 2 *[3, page 126] If the subgraph of G consisting of the edges $e_1, e_2, \ldots, e_k$ contain a cycle, the corresponding columns of $S_v(\vec{G})$ are linearly dependent.*

Corollary 1 *[3, page 126] A square submatrix of $S_v(\vec{G})$ has determinant ± 1 if the corresponding edges form a tree in G and 0 otherwise.*

If we denote the submatrix of A obtained by crossing out all rows but $i_1, i_2, \ldots, i_k$ and all columns except $j_1, j_2, \ldots, j_l$ by $A\binom{i_1, i_2, \ldots, i_k}{j_1, j_2, \ldots, j_l}$ it follows that the number of spanning trees is

$$t(G) = \sum_{j_1 < j_2 < \ldots < j_{n-1}} \left| S_v(\vec{G}) \begin{pmatrix} i_1, i_2, \ldots, i_{n-1} \\ j_1, j_2, \ldots, j_{n-1} \end{pmatrix} \right|^2 \tag{3}$$

In order to write this result in a more compact fashion we require the following result:

Theorem 1 *(Binet-Cauchy) Suppose that A and B are a $p \times q$ and a $q \times p$ matrices respectively ($p \leq q$), then*

$$|AB| = \sum_{j_1 < j_2 < \ldots < j_p} \left| A \begin{pmatrix} i_1, i_2, \ldots, i_p \\ j_1, j_2, \ldots, j_p \end{pmatrix} \right| \left| B \begin{pmatrix} j_1, j_2, \ldots, j_p \\ i_1, i_2, \ldots, i_p \end{pmatrix} \right| \tag{4}$$

Applying this result to $A = S_v(\vec{G})$ and $B = S_v(\vec{G})^T$ we obtain:

$$|S_v(\vec{G}) S_v(\vec{G})^T| = \sum_{j_1 < j_2 < \ldots < j_{n-1}} \left| S_v(\vec{G}) \begin{pmatrix} i_1, i_2, \ldots, i_{n-1} \\ j_1, j_2, \ldots, j_{n-1} \end{pmatrix} \right| \left| S_v(\vec{G})^T \begin{pmatrix} j_1, j_2, \ldots, j_{n-1} \\ i_1, i_2, \ldots, i_{n-1} \end{pmatrix} \right| \tag{5}$$

$$= \sum_{j_1 < j_2 < \ldots < j_{n-1}} \left| S_v(\vec{G}) \begin{pmatrix} i_1, i_2, \ldots, i_{n-1} \\ j_1, j_2, \ldots, j_{n-1} \end{pmatrix} \right|^2 = t(G) \tag{6}$$

Next we examine $S_v(\vec{G})S_v(\vec{G})^T$. First note that

$$(S(\vec{G})S(\vec{G})^T)_{ii} = \sum_{j=1}^{e}(S_{ij}(\vec{G}))^2 = D_G(i) \tag{7}$$

where $D_G(i)$ is the degree of vertex i, and

$$(S(\vec{G})S(\vec{G})^T)_{ik} = \sum_{j=1}^{e} S_{ij}(\vec{G})S_{kj}(\vec{G}) = -\Delta_{ik} \tag{8}$$

where Δ_{ik} = number of edges between vertex i and vertex k.
Given a multigraph G, let us define a square matrix $H(G)$ as follows:

$$H(G)_{ij} = \begin{cases} D_G(i) & : \quad i = j \\ -\Delta_{ij} & : \quad i \neq j \end{cases}$$

where $H(G)$ is called the *nodal admittance matrix* of G.

It can be easily shown that $S_v(\vec{G})S_v(\vec{G})^T = H_v(G)$ where $H_v(G)$ is obtained from $H(G)$ by eliminating row and column v, thus from (5) and (6) we get

$$t(G) = |H_v(G)|, \quad \text{for any} \quad v \in V(G) \tag{9}$$

where $V(G)$ denotes the vertex set of G.

Moreover [13] $H(G)$ is symmetric, positive semi-definite by noting that $\bar{x}H(G)\bar{x}^T = ||\bar{x}S(\vec{G})||^2 \geq 0$, and singular because $H(G)\bar{1}^T = \bar{0}$. Therefore the eigenvalues of $H(G)$ are real, non-negative and at least one is 0. These eigenvalues can be written as

$$\lambda_0 = 0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq \ldots \leq \lambda_{n-1} \tag{10}$$

We required yet another formula for $t(G)$ utilizing the spectrum of the nodal admittance matrix of G.
Let $P_{H(G)}$ denote the characteristic polynomial of $H(G)$. Then

$$P_{H(G)}(\lambda) = \lambda \prod_{i=1}^{n-1}(\lambda - \lambda_i) = |\lambda I_n - H(G)| \tag{11}$$

By differentiating the left side of (11) and evaluating for $\lambda = 0$ we obtain

$$P'_{H(G)}(0) = (-1)^{n-1} \prod_{i=1}^{n-1} \lambda_i \tag{12}$$

and by differentiating the right side of (11) and evaluating for $\lambda = 0$ we get

$$P'_{H(G)}(0) = n(-1)^{n-1}|H_v(G)| \tag{13}$$

From (12) and (13) we conclude that

$$t(G) = \frac{1}{n} \prod_{i=1}^{n-1} \lambda_i \tag{14}$$

Thus the number of spanning trees of a multigraph G can be obtained from the eigenvalues of $H(G)$.

2.2 Algorithmic implementation for the determination of the number of spanning trees (ST algorithm)

Given a matrix A and transformation invertible matrix P let $P^{-1}AP$ be the *similarity transform* . It can be easily shown that similarity transformations have the same spectrum as the original matrix.

If A is a $n \times n$ real, symmetric matrix, there exist a orthogonal matrix Z such that there exists a similarity transformation

$$Z^T AZ = \mathrm{diag}(\lambda_0, \lambda_1, \ldots, \lambda_{n-1}) \tag{15}$$

For any matrix A consider the sequence of similarity transformations

$$A \to P_1^{-1}AP_1 \to P_2^{-1}P_1^{-1}AP_1P_2 \to \ldots \to P_k^{-1}P_{k-1}^{-1}\ldots P_1^{-1}AP_1P_2\ldots P_k \tag{16}$$

where in each iteration the eigenvalues are preserved.

Given that A is a symmetric matrix, the *Jacobi Transformation* [3, page 339] technique constructs the individual $P_i^{\prime}s$ with the purpose of zeroing specific non-diagonal matrix elements. For each transformation (plane rotation) a non-diagonal matrix element is annihilated. Since by obtaining a triangular matrix the diagonal elements are eigenvalues of the original matrix, it's sufficient to zero the superdiagonal or the subdiagonal elements.

The Jacobi plane rotation P_{pq} matrix is of the form

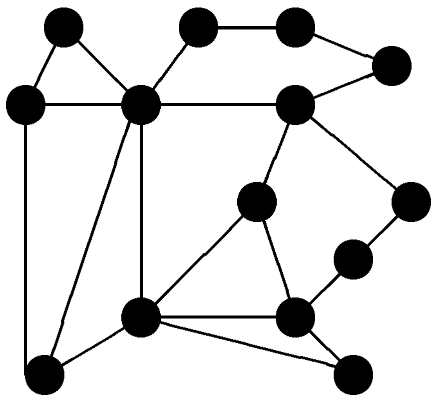
$$P_{pq} = \begin{pmatrix} 1 & & & & & & \\ & \ldots & & & & & \\ & & c & \ldots & s & & \\ & & \vdots & 1 & \vdots & & \\ & & -s & \ldots & c & & \\ & & & & & \ldots & \\ & & & & & & 1 \end{pmatrix} \tag{17}$$

The element $P_{pq}\{ii\} = 1$ for $i \neq p$ and $i \neq q$, $P_{pq}\{ii\} = c$ for $i = p$ or $i = q$. The non-diagonal elements are 0 except for $P_{pq}\{pq\} = s$ and $P_{pq}\{qp\} = -s$. Moreover P_{pq} is orthogonal given that $s^2 + c^2 = 1$. Then s and c represent the cosine and sine of a rotation angle θ. For each plane rotation $P_{pq}^T AP_{pq}$, and using the symmetry of A, θ will be determine in order to annihilate (within machine precision) a superdiagonal (or subdiagonal) element a_{pq} [3, page 339].

In [3, page 347] the Jacobi Transformation algorithm was implemented in FORTRAN IV, where given a $n \times n$ symmetric matrix A, a $n \times n$ triangular matrix which diagonal elements are the eigenvalues of A, is obtained in $O(n^2)$ steps. We then converted this algorithm into C++ classes, where $H(G)$ is the input to the algorithm and obtaining $\bar{\lambda}(H(G)) = (\lambda_0, \lambda_1, \ldots, \lambda_{n-1})$ and $t(G)$ (see (14)) as a result. This was done within a frame of the HEIDI tool [23].

2.3 Numerical results

In this section we present computational results obtained from the above mentioned algorithm. The model under consideration is a subset of the Montevideo's optical fiber telephone



G G'

Figure 1: G) Montevideo's network - G') Alternative network

network, and is composed of 14 nodes and 21 links (see Figure 1). This network was designed by the national telecomunications company (ANTEL) by application of heuristic rules.

In [24] a methodology based on the *Local Optimum Search* heuristic for network vulnerability improvement (reduction) is presented. The vulnerability index here under consideration, $Vul(G)$, was previously introduced in Section I. Given an original network $G = (V, E) \in \Omega(n, e)$, and an associated edge-weight function $W(G) = \{w(e_1), w(e_2), \ldots, w(e_{|E|})\}$ an alternative network $G' = (V, E') \in \Omega(n, e)$, edge-weight function $W(G') = \{w(e'_1), w(e'_2), \ldots, w(e'_{|E'|})\}$, with $Vul(G') < Vul(G)$ is obtained. Consider the sequence of *local search* vulnerability reducing applications: $G \to G_1 \to G_2 \ldots \to G'$. G_i is obtained from G_{i-1} by replacing two not adjacent edges $e_1 = (a, b)$ and $e_2 = (c, d)$ by edges $e_1{}^* = (a, d)$ and $e_2{}^* = (b, c)$ (*X-change*) given that the sum of the weights of new edges is constrained by a function of the sum of the weights of the replaced edges (e.g. euclidean coordinates).

In Figure 1, a vulnerability improved network G' is shown after the application of this algorithm to the Montevideo's network labeled here as G.

In Table 1, the resulting spectrum (rounded to two decimal places) and the number of spanning trees for G and G' are shown, by application of previously defined ST algorithm. Furthermore $t(G') > t(G)$, suggesting the inclusion of this index within a frame of a local search heuristic such as *X-change* for vulnerability improvement.

Table 1: **Spectrum and number of spanning trees**

λ	0	1	2	3	4	5	6	7	8	9	10	11	12	13	$t(G)$
G	0	.53	.66	1.14	1.71	2.22	2.5	2.6	3.28	4.08	4.73	5.25	5.83	7.48	10280
G'	0	.51	.8	1.17	1.67	1.97	2.36	2.84	3.32	3.9	5.00	5.22	5.8	7.43	10984

3 Edge-Toughness

In this section we introduce another vulnerability index known as *edge-toughness*.

Let $\omega(G)$ represent the number of connected components of a graph G. For any multigraph $G = (V(G), E(G))$, we define edge-toughness $\eta(G)$ by

$$\eta(G) = \min_{S \subseteq E(G)} \frac{|S|}{\omega(G - S) - \omega(G)} \tag{18}$$

over all subsets $S \subseteq E(G)$ such as $\omega(G - S) > \omega(G)$. This function was originally introduced by Gusfield [8]. If we are interested in splitting a graph G into $k + \omega(G)$ components, the definition of $\eta(G)$ tell us that is necessary to remove at least $k\eta(G)$ edges, i.e. how tough is it is to break up G.

One of the immediate consequences of the definition of $\eta(G)$ is that

$$\eta(G) \leq \frac{|E(G)|}{n - 1} \tag{19}$$

where $n = |V(G)|$.

Even though $\eta(G)$ is as function on the edge cuts of G, it is intrinsically related to the spanning trees of G. In 1961, Nash-Williams [15] and Tutte [4] independently proved the following:

Theorem 2 *For any non-trivial connected graph G, the maximum number of edge-disjoint spanning trees of G is $\lfloor \eta(G) \rfloor$.*

Furthermore, Cunningham [18, page 551] found a generalization of Theorem 2, implicating the spanning trees of a graph:

Theorem 3 *For any nontrivial graph G and any natural numbers s and t, $\eta(G) \geq s/t$ if and only if G has a family F of s spanning trees, known as a* t-packing*, such that each edge of G lies in at most t trees of F.*

The edge-connectivity of a graph $G = (V(G), E(G))$, $\lambda(G)$, is the size of the minimum edge-cut. Catlin [19] has shown that edge-toughness allows a deeper characterization of the edge-connectivity:

Theorem 4 *For any non-trivial graph $G = (V(G), E(G))$, let $k \leq |E(G)|$ be an integer, and let E_k be the collection of all k-elements subsets of $E(G)$. Then*

(a) $\lambda(G) \geq 2k$ if and only if for all $E' \subseteq E_k, \eta(G - E') \geq k$; and

(b) $\lambda(G) \geq 2k + 1$ if and only if for all $E' \subseteq E_k, \eta(G - E') > k$.

3.1 Edge-toughness properties

One of the relevant characteristics of this vulnerability index it that mantains certain properties when minors of a graph are under consideration.
Let H be a subgraph of G. We define G/H the graph obtained from G identifying each of the end-points of each of the edges of H and deleting the possible resulting self-loops. This is reflected in the following lemma:

Lemma 3 *[20, page 104] If G has a subgraph H such that $\eta(H) > \eta(G)$, then $\eta(G) = \eta(G/H)$.*

This lemma concludes in the following corollary:

Corollary 2 *[20, page 104] Let G be a connected graph with blocks $B_1, B_2, \ldots, B_r$. Then $\eta(G) = min_i \quad \eta(B_i)$.*

A graph G is said to be η-reduced if G contains no subgraph H with $\eta(H) > \eta(G)$. Thus if G is not η-reduced there exist a sequence of contractions of subgraphs H with $\eta(H) > \eta(G)$ to a η-reduced graph G_0; by Lemma 2, $\eta(G) = \eta(G_0)$ where G_0 is called an η-reduced graph of G.

In [15] and [20] graphs with $\eta(G) = \frac{|E(G)|}{n-1}$ (see (19)) were characterized:

Theorem 5 *The sets of graphs G satisfying $\eta(G) = \frac{|E(G)|}{n-1}$ includes:*

(a) Plane triangulations;

(b) For any natural number k, the edge-disjoint union of k nontrivial spanning trees;

(c) edge-transitive graphs.

3.2 Generalization to matroids

Cunningham [18] generalized this function to matroids and also provided a polynomial time algorithm for computing $\eta(G)$ for any non-nontrivial graph G.

Let M be a matroid on S with rank function ρ. Define

$$\eta(M) = \min_{T \subset S} \frac{|S - T|}{\rho(S) - \rho(T)}, \tag{20}$$

where the minimum is taken over subsets T of S such that $\rho(T) < \rho(S)$. Also (19) can be easily generalized to matroids as follows:

$$\eta(M) \leq \frac{|S|}{\rho(S)}. \tag{21}$$

The following theorem due to Catlin, Hobbs , and Lai [20], reinforces this generalization.

Theorem 6 *If matroid M is the cycle matroid of a graph G, then $\eta(M) = \eta(G)$.*

We define a *t-packing* of a matroid M to be a family F of bases such that each $e \in S$ lies in at most t members of F. Let $\eta_t(M)$ be the largest cardinality of a t-packing of M.

Theorem 7 *[18, 20] For any natural numbers s and t, $\eta(M) \geq s/t$ if and only if there exist a t-packing of M containing s bases.*

We have the following corollaries to this theorem.

Corollary 3 *[19] Let M be a matroid on S and let t be a natural number. Then $\eta_t(M) = \lfloor t\eta(M) \rfloor$.*

Corollary 4 *Let M be a matroid. Then $\eta(M) = \max_{1 \leq i \leq \rho(M)}\{\eta_i(M)/i\}$.*

3.3 Algorithm for the determination of $\eta(G)$ and numerical results

An algorithm for computing $\eta(G)$ for graph G appears in Cunningham [18]. However this algorithm seems to be limited to graphs or graphic matroids. Hobbs [22, page 94] presented a polynomial time algorithm for matroids.

Given matroids $M_1, M_2, \ldots, M_k$, Edmonds [5] provided an algorithm for partitioning the elements of a matroid into disjoints subsets $X_1, X_2, \ldots, X_k$, such that X_i is independent in M_i for each i. Later Knuth [12] modified and extended Edmonds' algorithm. Hoob modified Knuth's algorithm to compute $\eta(M)$ for a matroid M.

Let M be a matroid on S with $|S| = m$, and where $k, n_1, n_2, \ldots, n_k$ and v are nonnegative integers that are determined by the algorithm. Here k is the ordinary label of the presently increasing independent set, and $n_1, n2, \ldots, n_k$ are the sizes of the independent sets founded sor far, and v is the number of bases among the independent sets found. In Step 4 of the algorithm, M_t is obtained from M by replacing every element of M, and S_t is the sets of elements of M_t.

The algorithm for the determination of $\eta(M)$ of a matroid M (see Figure 2) is easily translated into a graph algorithm by considering the cycle matroid M of a graph $G = (V(G), E(G))$ (i.e. an independent set is a subset $X \subseteq E(G)$ such that doesn't contain any cycles). By letting $S = E(G)$ and noting that for a connected graph the largest independent sets (bases) of the cycle matroid correspond to the spanning trees of G. Thus $\rho(S) = n - 1$. Moreover by Theorem 6, $\eta(G) = \eta(M)$.

This algorithm [22, page 101] is $O(m^3(\rho(S)^4)$ in its number of test of independence, or equivalently $O(|E(G)|^3|V(G)|^4)$ in test for cycles in the case of the of cycle matroid of G. The detection of cycles was accomplished by application of a modified *Depth First Search* [1, page 53] (see step 3b). Moreover *Breadth First Search* was applied in the determination of shortests paths [1, page 12](Step 3c). The edge-toughness algorithm was implemented in C++, with Incidence Matrix $A(G)$ as input to the algorithm and $\eta(G)$ as a result. This was done within the frame of the HEIDI tool [23].

The edge-thougness algorithm shows $\eta(G) = 1.3333$ where G is the underlying graph of the Montevideo's Network (see Figure 1). In the class of graphs $\Omega(14, 21)$, by (19), the upper bound is $1.62 = 21/13$ and it is achieved by the graph showed in Figure 3.

1. Set $t = 1, M_1 = M$, and $S_1 = S$.
2. Set $k = 1$, and $v = 0$.
3. (a) Start with $X_i = \emptyset$ for all i.
   (b) Construct a digraph D_t on $S_t \cup \{O_1, \ldots, O_k\}$ as follows: if $x \in S_t$ is not in X_j and if $X_j \cup \{x\}$ is independent in $M_t, 1 \leq j \leq k$, put arc (x, O_j) in D_t. If $x \in S_t - X_j$ and if $X_j \cup \{x\}$ is dependent, let C be the unique circuit in $X_j \cup \{x\}$. Put (x, y) in D_t for each $y \neq x$ in C.
   (c) Attempt to find a $y \in S_t - \bigcup X_i$ such that there is a dipath of minimal length from y to O_k in D_t.
   (d) If such a path $y = x_0, x_1, x_2, \ldots, x_p, O_k$ exists, for each $j = 1, 2, \ldots, p$, successively let i_j be the subscript such that $x_j \in X_{i_j}$ and reform X_{i_j} as $(X_{i_j} \cup \{x_{j-1}\}) - x_j$. Last reform X_k as $X_k \cup \{x_p\}$. Go to $3b$.
   (e) If not such y exists, X_k is as large as possible. Set $n_k = |X_k|$. If $|X_k| = \rho(S)$, increase v by 1. If $S_t - \bigcup X_i \neq \emptyset$, increase k by 1 and go to $3b$.
4. Now $S_t - \bigcup X_i = \emptyset$; set $r_t = v$ and $s_t = k$. If $r_t = s_t$ then $\eta(M) = r_t/t$;stop. Otherwise, if $t < (n-1)$, increase t by one, form M_t and S_t, and go to Step 2;
5. Then $\eta(M) = max\{r_i/i\}$.

Figure 2: Algorithm for the determination of $\eta(M)$ of a matroid M

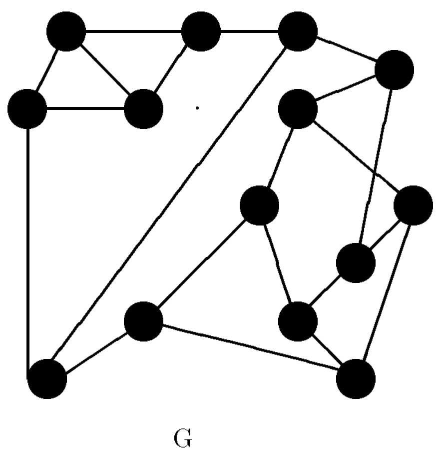


Figure 3: G - Graph with optimal edge-toughness in $\Omega(14, 21)$

4 Conclusions

In this paper we proposed two vulnerability indexes and algorithms of polynomial time complexity for their computation. Although, given a graph G, the computation of $t(G)$ is more efficient than $\eta(G)$ time complexitywise, its utilization within a frame of a heuristic technique (ej. local optimal search) was justified when we consider graphs with the same number of nodes and edges. This restriction could be overlooked in the case of $\eta(G)$.

References

[1] S. Even (1974), *Graph Algorithms*, Computer Science Press, Maryland.

[2] M.N.S. Swamy, K.Thulasiram (1981), *Graphs, Networks, and Algorithms*, John Wiley and Sons.

[3] W. Press, B.P. Flannery, S.A. Teukolsky, W. T. Vetterling (1986), *Numerical Recipes*, Cambridge University Press, Cambridge.

[4] W.T. Tutte (1961), *On the problem of decomposing a graph into n connected factors*, J. London Math. Soc. 36, 221-230.

[5] J. Edmonds (1967), *Matroids Partition*, Lectures Notes in Appl. Math. 11 (Mathematics of the Decision Sciences), 335-346.

[6] Proban and Ball (1983), *The complexity of counting cuts and of computing that a graph is connected*, SIAM J. Computing 12, 777-788.

[7] Boesch, Li and Suffel (1991), *On the exsitence of uniformly most reliable networks*, Networks 21, 181-194.

[8] D. Gusfield (1983), *Connectivity and edge-disjoint spanning trees*, Information Processing Letters 16, 87-89.

[9] C. Chang (1981), *Maximizing the total number of spanning trees in a graph: two related problems in graph theory and optimization design theory*, Journal Combinatorial (B) 31, 240-248".

[10] Cvetovic, Doob, Sachs (1980), *Spectra of Graphs*, Academic Press, Matematics 87.

[11] A. Kelmans and V. Chelnokov (1974), *A certain polynomial of a graph and graphs with an extremal number of trees*, Journal of Combinatorial Theory (B) 16, 197-214.

[12] D. E. Knuth (1973), *Matroid Partitioning*, Stanford University STAN-CS-73-342, 1-12.

[13] L. Petingi, F. Boesch, C. Suffel, *On the characterization of graphs with maximum number of spanning trees*, Discrete Mathematics, to appear.

[14] J. Rodriguez, L. Petingi, *Graphs with maximum number of spanning trees and optimally reliable graphs*, Combinatorial Optimization, to appear.

[15] C. St. J. A. Nash-Williams (1961), *Edge-disjoint spanning trees of finite graphs*, J. London Math Soc. 445-450.

[16] H. Narayanan and M. N. Vartak (1980), *On molecular and atomic matroids*, Combinatorics and Graph Theory (Calcutta), 358-364, Lectures Notes in mathematics 885, Springer-Verlag, Berlin (1981).

[17] D. Shier (1974), *Maximizing the number of spanning trees in a graph with n nodes and m edges*, J. Res. Nat. Bureau Standards 78B, 193-196.

[18] W. H. Cunningham (1985), *Optimal attack and reinforcement of a network*,J. Assoc. Comp. Mach. 549-561.

[19] P. A. Catlin, *the reduction of graphs families closed under construction*, Preprint.

[20] P. A. Catlin, A. M. Hobbs, H. Lai *Matroids Unions, edge-toughness, and fractional arboricity*, Preprint.

[21] F. Harary (1971), *Graph Theory*, Addison-Wesley, Reading, Mass.

[22] A. M. Hoobs (1989), *Computing Edge-Toughness and Fractional Arboricity*, Contemporary Mathematics, Graphs and Algorithms, American Mathematical Society 89.

[23] H. Cancela, M. E. Urquhart, G. Rubino (1992), *Optimización en el diseño de redes de comunicación.*, Proceedings of the *XIII CILAMCE*, Porto Alegre, Brasil.

[24] H. Cancela, M. Urquhart, G. Rubino (1993), *Heidi- Una herramienta para el diseño de Redes de Comunicación.*, Technical Report INCO 93-01, Pedeciba Informática, Facultad de Ingeniería, Uruguay.