Finding the Most Vital Edges with Respect to the Number of Spanning Trees

Fu-Shang P. Tsen
National Chiao Tung University, Hsinchu
Ting-Yi Sung
Academia Sinica, Taipei
Men-Yang Lin
National Taichung Institute of Commerce, Taichung
Lih-Hsing Hsu
National Chiao Tung University, Hsinchu
Wendy Myrvold
University of Victoria, Victoria

Key Words — **Reliability optimization, number of spanning trees, most vital edges, matrix operation**

Reader Aids —
General purpose: Sensitivity analysis of network reliability
Special math needed for derivations: Matrix operations, graph theory, and associated jargon
Special math needed to use results: Same
Results useful to: Reliability analysts, network designers

Summary & Conclusions — **A *most vital edge* of a graph (w.r.t. the spanning trees) is an edge whose deletion most drastically decreases the number of spanning trees. We present an algorithm for determining the most vital edges based on Kirchoff's matrix-tree theorem whose asymptotic time-complexity can be reduced to that of the fastest matrix multiplication routine, currently $O(n^{2.376})$. The foundation for this approach is a more general algorithm for directed graphs for counting the rooted spanning arborescences containing each of the arcs of a digraph. A network can be modeled as a probabilistic graph. Under one such model proposed by Kel'mans, the all-terminal network reliability, maximizing the number of spanning trees is critical to maximizing reliability when edges are very unreliable. For this model, the most vital edges characterize the locations where an improvement of the reliability of the link most improves the reliability of the network.**

1. INTRODUCTION

Any elementary linear algebra text can be used as a reference for the linear algebra in this paper, *eg*, [19]. See [3] for any graph theoretic questions.

1.1 Definitions

An undirected graph $G = (V,E)$ consists of a set V of n vertices and a set E of m edges, where each edge is an unordered pair of vertices from V. Graph $H = (V',E')$ is a *subgraph* of G if $V' \subseteq V$ and $E' \subseteq E$. A *spanning* subgraph is any subgraph of G satisfying $V' = V$. A *spanning tree* of G is a connected spanning subgraph of G that contains no cycles.

A *most vital* edge in G is an edge whose deletion results in a maximum decrease in the number of spanning trees. This paper presents an algorithm for determining the most vital edges of a graph (w.r.t. spanning trees). The same approach can be used to pinpoint locations where an added edge most increases the number of spanning trees. Refs [7,10-12] present algorithms for determining the most vital edges for some other graph parameters.

1.2 Algorithm Background

The cornerstone of the algorithm is the famous Matrix-tree theorem of Kirchoff[1] which expresses the number of spanning trees of a graph in terms of the determinant of a matrix of order n. A naive approach for pinpointing the most vital edges is to count the spanning trees of each of the m graphs created by removing each of the m edges of G in turn; this approach involves m determinant computations. We develop a more sophisticated approach by first considering the more general question of counting the rooted spanning arborescences containing each arc of a directed graph. The time complexity of our algorithm reduces to that of matrix multiplication[2]; or equivalently, the amount of work done by the entire algorithm is asymptotically equivalent to the time it takes to do just one determinant computation as required by the naive approach.

This algorithm has applications in network reliability. The *all-terminal* network reliability model is defined as:

- Vertices represent sites, and edges represent links between the sites. The vertices are perfectly reliable, but edges operate *s*-independently with the same probability, p. The network is operational if the underlying probabilistic graph is connected. ◄

Kel'mans [14] first proposed this model of reliability. Colbourn's monograph [5] is an excellent survey of the work on this problem.

1.3 Purpose

Since computing the all-terminal reliability of a network appears to be intractable (the problem is *#P*-complete), approximation schemes have been proposed [5: chapter 5]. When the edges are very unreliable, maximizing the number of spanning trees is critical to maximizing reliability. Maximizing the number of spanning trees over all graphs with the same numbers of edges & vertices does not guarantee a network that is most reliable for all values of p [17].

[1]See theorem 2 — which can also be induced as a corollary of Tutte [21].
[2]The fastest algorithm for this so far is $O(n^{2.376})$ [6].

Other authors have concentrated on the problem of determining the graphs on n vertices and m edges with the maximum number of spanning trees [2,4,13,15,18,20,22]. However in many applications, the graph topology is already given. For this situation, determining the most vital edges of a network highlights the locations where an improvement to link reliability most improves reliability.

Standard notation is given in "Information for Readers & Authors" at the rear of each issue.

2. COUNTING ARBORESCENCES

To get an improved algorithm, it helps to consider the more general problem for directed graphs. In a directed graph (digraph), a *rooted spanning arborescence* or *rooted spanning out-tree* (or simply *arborescence*) is a spanning subdigraph which has no directed cycles, and in which one vertex r distinguished as the root can reach each other vertex by directed paths. The digraph D *associated with* an undirected graph G is constructed by replacing each edge (u,v) of G with the arcs (u,v) & (v,u). Lemma 1 is well-known; *eg*, it follows from the intuitive proof of Kirchoff's theorem [8: theorem 2.5, p 53].

Lemma 1. Let G be an undirected graph and D the digraph associated with G. There is a one-to-one correspondence between the spanning trees of G and the spanning arborescences rooted at any one of the vertices of D. ◄

Furthermore, from the one-to-one correspondence stated in lemma 1, it follows that the number of spanning trees of an undirected graph G containing a particular edge (u,v) equals 'the number of arborescences in the associated digraph containing arc (u,v)' plus 'the number of them containing (v,u)'. For this reason, we restrict our attention to the problem of determining the number of arborescences containing a fixed arc in a digraph. The digraphs can have *multiple* arcs (more than one arc between a pair of vertices). *Loops* (arcs whose endpoints are the same vertex) can be safely erased since they do not contribute to any arborescences.

We now present theorem 1, a classical result for counting spanning rooted arborescences of a digraph. The *Kirchoff* matrix (*in-degree* matrix) K associated with a digraph D is defined as:

- the diagonal entry k_{ii} is the in-degree of vertex i; the off-diagonal entry k_{ij} is $-n_{ij}$ where n_{ij} is the number of arcs entering vertex i from vertex j.

Theorem 1 [16]. The number of spanning arborescences rooted at r in a digraph D equals $\det(K_{rr})$, where K_{rr} is the Kirchoff matrix of D with the row and the column corresponding to vertex r deleted. ◄

To avoid treating the arcs which are incident to the root vertex r of D as a special case, we augment the original digraph D with a new vertex u, along with a new arc (u,r). There is a one-to-one correspondence between the arborescences of D and those of the augmented digraph. Correspondingly, the *augmented* Kirchoff matrix A for a rooted digraph D is defined as:

- the Kirchoff matrix K with one added to the diagonal entry for the root vertex (or equivalently, the matrix of the augmented digraph with 'the row and the column corresponding to the new root' removed).

Corollary 1. The number of spanning arborescences of D rooted at a vertex r equals the determinant of the corresponding augmented Kirchoff matrix. ◄

Because exactly one arc enters each non-root vertex in an arborescence, there is a one-to-one correspondence between the arborescences of D which contain arc a and those of $D \circ a$ which is obtained from D by removing all arcs, except a, which enter the terminus of a. Thus, the number of spanning trees containing an arc can be computed by evaluating the determinant of the augmented Kirchoff matrix for $D \circ a$. However, if we first compute the inverse of the augmented Kirchoff matrix, the same result can be found in constant time.

Lemma 2. The number of arborescences rooted at vertex r in digraph D can be computed in $O(1)$ time for any arc $a=(i,j)$, given the augmented Kirchoff matrix A, $\det(A)$, and A^{-1}. ◄

3. THE ALGORITHM

The algorithm finds the most vital arcs (or edges).

Algorithm 1

RELIABILITY_MVE

1. Let A be the augmented Kirchoff matrix (as defined in section 2). Construct:

$\mathrm{adj}(A) = [b_{ij}] = \det(A) \cdot A^{-1}$.

2. For each arc (i,j), the number of arborescences containing this arc is $b_{jj} - b_{ij}$. (Alternatively, for the undirected case, the number of trees containing edge (i,j) is:

$b_{jj} - b_{ij} + b_{ii} - b_{ji}$).

Select the arcs (edges) for which this is maximized. ◄

Step 1 can be completed in $O(n^3)$ time by using Gaussian elimination. It can be accelerated to $O(n^{2.376})$ by using matrix multiplication [6], or in general to the speed of a fastest matrix multiplication routine. All the edges can be processed, as in step 2, in $O(n^2)$ time. Thus, the limiting factor is the complexity of matrix multiplication.

Because of the structure of the augmented Kirchoff matrix, it is never necessary to permute rows or columns to avoid zero pivots when applying Gaussian elimination to compute the inverse. This can be used in a practical implementation. The theoretical implication of this is that a very fast matrix multiplication routine, *eg*, $O(n^2)$, implies that the inverse of the augmented Kirchoff matrix can be found in the same time; this is not true for the general inversion problem where row and/or column permutations might be required [1: chapter 6].

We could alternatively define a most vital edge as an edge $e^* \notin E$ whose insertion into the graph results in the largest increase in the number of spanning trees. The algorithm is exactly as algorithm 1, except that step 2 is applied for each non-edge.

ACKNOWLEDGMENT

This work was partially supported by the National Science Council, Taiwan - R.O.C, and by NSERC.

APPENDIX

Proof of Lemma 2

The *classical adjoint* matrix of a matrix A, adj(A) = $[b_{ij}]$, is det(A) $\cdot A^{-1}$. The value of b_{ij} is $(-1)^{i+j} \cdot \det(A_{ji})$, where A_{ji} is the matrix obtained from A by deleting row j and column i. The determinant of $A = [\alpha_{ij}]$ is $\Sigma_{j=1}^{n} \alpha_{ij} \cdot (-1)^{i+j} \cdot \det(A_{ij})$ — this is *expansion by cofactors across row i*. Thus, the determinant of A can be computed by cross-multiplying row i of A and column i of the matrix adj(A). The crucial observation is that column i of adj(A) contains the cofactors of A obtained by deleting row i of A (along with one of the columns). Hence any change to row i of A has no effect on the column i of adj(A).

e_i is a vector which is 1 in position i but is 0 otherwise. The augmented Kirchoff matrix of $D \circ a$ is obtained from A by changing row j to $(e_j - e_i)$. Thus, from the above observations, the number of arborescences containing a is $b_{jj} - b_{ij}$. Given the information that we have, this can be computed in $O(1)$ time. *Q.E.D.*

REFERENCES

[1] A.V. Aho, J.E. Hopcroft, J.D. Ullman, *The Design and Analysis of Computer Algorithms*, 1974; Addison-Wesley.

[2] F.T. Boesch, Z.R. Bogdanowicz, "The number of spanning trees in a prism", *Int'l J. Computer Mathematics*, vol 21, num 3/4, 1987, pp 229-243.

[3] J.A. Bondy, U.S.R. Murty, *Graph Theory with Applications*, 1980; North-Holland.

[4] C.-S. Cheng, "Maximizing the total number of spanning trees in a graph: two related problems in graph theory and optimum design theory", *J. Combinatorial Theory*, series B, vol 31, 1981 Oct, pp 240-248.

[5] C.J. Colbourn, *The Combinatorics of Network Reliability*, 1987; Oxford University Press.

[6] D. Coppersmith, S. Winograd, "Matrix multiplication via arithmetic progressions", *Proc. 19th Ann. ACM Symp. Theory of Computing*, 1987, pp 1-6.

[7] H.W. Corley, D.Y. Sha, "Most vital links and nodes in weighted networks", *Operations Research Letters*, vol 1, 1982 Sep, pp 157-160.

[8] A. Gibbons, *Algorithmic Graph Theory*, 1985; Cambridge University Press.

[9] P.E. Gill, W. Murray, M.H. Wright, *Numerical Linear Algebra and Optimization* vol 1, 1991; Addison-Wesley.

[10] Hsu, Jan, Lee, Hung, Chern, "Finding the most vital edge with respect to minimum spanning tree in weighted graphs", *Information Processing Letters*, vol 39, 1991 Sep, pp 277-281.

[11] C.N. Hung, L.H. Hsu, T.Y. Sung, "The most vital edges of a matching in a bipartite graph", *Networks*, vol 23, 1993 Jul, pp 309-313.

[12] R.H. Jan, L.H. Hsu, Y.Y. Lee, "The most vital edges with respect to the number of spanning trees in 2-terminal series-parallel graphs", *BIT*, vol 32, num 3, 1992, pp 403-412.

[13] A.K. Kel'mans, V. M. Chelnokov, "A certain polynomial of a graph and graphs with an extremal number of trees", *J. Combinatorial Theory*, series B, vol 16, 1974 Apr, pp 197-214.

[14] A.K. Kel'mans, "Connectivity of probabilistic networks", *Automation and Remote Control*, vol 29, 1967, pp 444-460.

[15] A.K. Kel'mans, "Comparison of graphs by their number of trees", *Discrete Mathematics*, vol 16, 1976, pp 241-261.

[16] G. Kirchoff, Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird, *Ann. Phys. Chemistry*, vol 72, 1847, pp 497%508, 1847. [Translated in: "On the solution of equations obtained from the investigation of the linear distribution of galvanic currents", *IRE Trans. Circuit Theory*, vol CT-5, 1958, pp 4-7.]

[17] W.J. Myrvold, K.H. Cheung, L.B. Page, J.E. Perry, "Uniformly-most reliable networks do not always exist", *Networks*, vol 21, 1991 Jul, pp 417-419.

[18] D.R. Shier, "Maximizing the number of spanning trees in a graph with n nodes and m edges", *J. Res. National Bureau of Standards*, vol 78B, 1974, pp 193-196.

[19] G. Strang, *Linear Algebra and Its Applications* (3rd ed), 1988; Harcourt Brace Jovanovich.

[20] S.S. Tseng, L.R. Wang, "Maximizing the number of spanning trees of networks based on cycle basis representation", *Int'l J. Computer Mathematics*, vol 28, num 1, 1989, pp 47-56.

[21] W.T. Tutte, "Lectures on matroids", *J. Res. National Bureau of Standards*, vol 69B, 1965, pp 1-47.

[22] J.F. Wang, C.S. Yang, "On the number of trees of circulant graphs", *Int'l J. Computer Mathematics*, vol 16, num 3/4, 1984, pp 229-241.

AUTHORS

Dr. Fu-Shang P. Tsen; Dept. of Applied Mathematics; National Chiao Tung University; Hsinchu, TAIWAN - R.O.C.
e-mail: fsptsen@twnctu01.bitnet

Fu-Shang P. Tsen is an Associate Professor at the Dept. of Applied Mathematics, National Chiao Tung University. He received his PhD from Brown University. His research interests include differential equations and numerical analysis.

Dr. Ting-Yi Sung; Inst. of Information Science; Academia Sinica; Taipei, TAIWAN - R.O.C.

Ting-Yi Sung is an Associate Research Fellow at the Institute of Information Science, Academia Sinica. She received the PhD from New York University. Her research interests include mathematical programming, graph algorithms, and interconnection networks. Her papers have appeared in *Mathematical Programming*, *Networks*, *etc*.

Men-Yang Lin; Dept. of EDP; National Taichung Institute of Commerce; Taichung, TAIWAN - R.O.C.

Men-Yang Lin is a Lecturer at the Department of Electronic Data Processing, National Taichung Institute of Commerce. He received his MS from National Chung Hsing University.

Dr. Lih-Hsing Hsu; Dept. of Computer & Information Science; National Chiao Tung University; Hsinchu, TAIWAN - R.O.C.
e-mail: lhhsu@cc.nctu.edu.tw

Lih-Hsing Hsu is a Professor at the Department of Computer & Information Science, National Chiao Tung University. He received his PhD from State University of New York at Stony Brook. His research interests include graph algorithms, interconnection networks, and VLSI. His papers have appeared in *Discrete Mathematics*, *Networks*, *etc*.