Improving Reliability Bounds in Computer Networks

Timothy B. Brecht and Charles J. Colbourn
Computer Communications Networks Group, Department of Computer Science, University of Waterloo, Waterloo, Ontario, N2L 3G1, Canada

The probability that a computer network is operational in an environment of statistically independent link failures has been widely studied. Three natural problems arise, when all nodes are to be connected (all-terminal reliability), when two nodes are to communicate (2-terminal reliability), and when k specified nodes are to communicate (k-terminal reliability); the latter case includes the first two. Each of these reliability measures is NP-hard to compute, and thus efficiently computable reliability bounds are of significant interest. To date, the all-terminal and 2-terminal cases have been treated separately, and few results apply to the k-terminal case. In this paper, we develop a simple strategy to obtain k-terminal reliability bounds. In the process, we demonstrate improvements on the previous best bounds for all-terminal, k-terminal, and 2-terminal reliability. Computational experience with these new bounds is reported, by comparing the new lower bounds to existing lower bounds.

1. INTRODUCTION

A computer network is often modelled as a *probabilistic graph* $G = (V,E)$; V is a set of n nodes representing computer centers, and E is a set of e edges representing bidirectional communications links. In this model, we assume that nodes are perfectly reliable, but that edges fail statistically independently with known probability. This model has been widely used [12,16,19]. The ability of a computer network to withstand random failures is a key component of its reliability. Hence, the *all-terminal reliability* of such a network is the probability that all nodes can communicate with one another. Similarly, the *2-terminal* (*k-terminal*) *reliability* is the probability that 2 (resp., k) specified nodes can communicate. We refer the reader to [4] for standard graph-theoretic terminology.

Exact computations for these reliability measures are almost certainly intractable, since computing any of the three is #P-complete [3,18]. One major method used to cope with this intractability is the development of upper and lower bounds on these reliability measures. Naturally, we are most interested in bounds which

NETWORKS, Vol. 16 (1986) 369–380

CCC 0028-3045/86/040369-12$04.00

are efficiently computable. Most of the bounds mentioned below apply only when all edge operation probabilities are equal; however, the bounds we develop do not require this assumption.

In the context of all-terminal reliability, Jacobs [10] developed simple bounds. These were later improved by Frank and Van Slyke [19] using a theorem of Kruskal [13] and Katona [11], which were improved further by Ball and Provan [2]. A substantially different approach was employed by Lomonosov and Polesskii [15]; their bounds are typically looser than the Ball–Provan bounds, but do yield occasional improvements [7]. Colbourn and Harms [7] describe computational comparisons of these bounds, and further describe a method using linear programming to combine these bounds (indeed, any set of basic bounds) to form a uniform bound that is at least as good as each basic bound.

In the context of 2-terminal reliability, the Kruskal–Katona bounds once again apply; however, the Ball–Provan bounds do not. Brecht and Colbourn [5] develop bounds based on edge-disjoint paths, and compare these bounds to the Kruskal–Katona bounds. Although there are many bounds for 2-terminal reliability, to our knowledge no others can be efficiently computed. Finally, in the context of k-terminal reliability, one can modify the Kruskal–Katona bounds to apply by using an efficiently computed underestimate for the size of a minimum cardinality pathset.

There are a number of different strategies used in obtaining bounds. The two basic strategies are subgraph counting (Kruskal–Katona and Ball–Provan, for example) and edge-disjoint subgraphs (Lomonosov–Polesskii and Brecht–Colbourn, for example). Neither strategy has (thus far) produced bounds which uniformly improve on the other strategy; this suggests the importance of combining basic bounds to obtain better bounds. A linear programming formulation can be used to combine basic all-terminal bounds to obtain an improved uniform all-terminal bound, and to combine basic 2-terminal bounds to obtain a 2-terminal bound [7]; this method only applies when edge operation probabilities are all equal. We develop a new method for combining bounds here even when edge probabilities differ, by exploring a simple relationship between 2-terminal bounds and k-terminal bounds. Despite its simplicity, it enables us to obtain substantial improvements, not only on existing k-terminal bounds, but also on all known 2-terminal and all-terminal bounds!

2. THE BOUNDING STRATEGY

Any method for computing bounds on 2-terminal reliability can be used to compute bounds $l(x,y) \leq p(x,y) \leq u(x,y)$, where $p(x,y)$ is the probability that there is an operating path from x to y. By computing $l(x,y)$ and $u(x,y)$ for all pairs x,y of nodes, we obtain a completely connected network in which each edge $e = (x,y)$ has success probability between $l(x,y)$ and $u(x,y)$. One might consider then applying standard bounding methods to this network; however, it is essential to note that these computed edge probabilities are *not* statistically independent, and hence all of the bounds mentioned thus far do not apply. Nevertheless, there are bounds which apply in this situation.

The general framework is as follows. For a probabilistic network $G = (V,E)$, each edge e is operational with probability p_e, where $l_e \le p_e \le u_e$; we are given l_e and u_e, but not p_e. Moreover, no information about statistical dependencies is given (and none can be assumed). We are to compute lower and upper bounds on the probability that G is operational. The meaning of "operational" here is intentionally left unspecified; later, we consider each of all-terminal, k-terminal, and 2-terminal operations. Notice that this model does apply to the completely connected network computed above.

For this model, Hailperin [9] developed a linear programming formulation; we use K^* to denote the set of a minimal cutsets of G, and P^* to denote the set of all minimal pathsets of G. Zemel [20] and Assous [1] employed Hailperin's model to observe that

Theorem 2.1. For a probabilistic network $G = (V,E)$, given only l_e and u_e for each edge e so that $l_e \le p_e \le u_e$, the best upper bound on reliability is given by

$$\beta = \min\left(1, \min_{C \in K^*} \sum_{j \in C} u_j\right)$$

The best lower bound is given by

$$\alpha = \max\left(0, 1 - \min_{S \in P^*} \sum_{j \in S} (1 - l_j)\right)$$

In other words, to compute an upper bound, we need only find a minimum weight cutset using the $\{u_i\}$ as weights; for a lower bound, we need only find a minimum weight pathset using $1 - l_i$ as weights.

Our general strategy is now easily described. A complete graph is created using 2-terminal reliability bounds which assume statistical independence. Theorem 2.1 is then applied to find lower and upper bounds for the operational probability of this graph.

3. THE APPLICATION TO 2-TERMINAL BOUNDS

In applying Hailperin's model to 2-terminal bounds from node s to node t, minimal pathsets are just s,t-paths, and minimal cutsets are minimal s,t-cuts. Hence to compute a lower bound, we need only compute a minimum weight s,t-path, and for an upper bound to compute a minimum weight, s,t-cut. Both problems have efficient solutions [14], and hence we can efficiently compute the bounds given by Theorem 2.1.

An implementation of the computation outlined here has been done, and yields improvements (which are substantial in some cases) on existing efficiently computable 2-terminal bounds. To illustrate this, we applied the method to compute lower bounds for a 5×5 "grid" network, depicted in Figure 3.1. In this and subsequent examples, each edge is assumed to be operational with the same

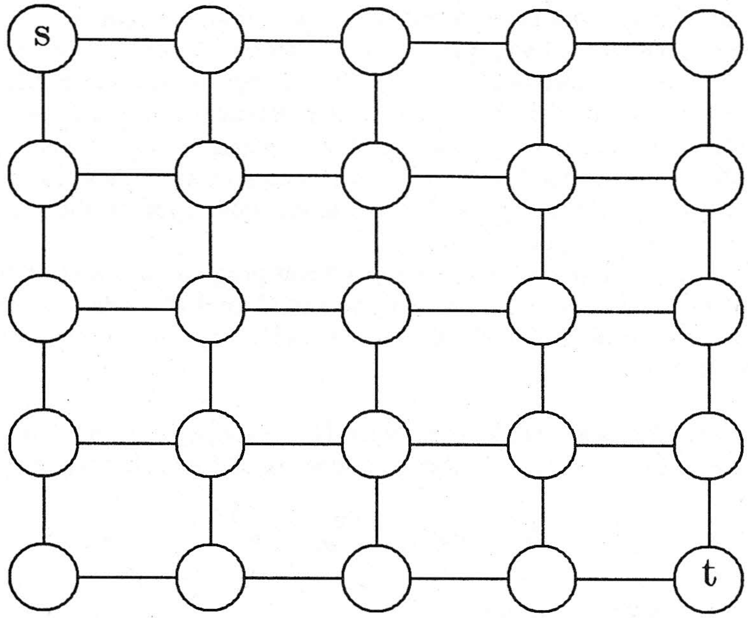

TABLE I. Two-terminal bounds (5 × 5 grid).

p	Kruskal–Katona	Mincost (edp)	$2t- > 2t$
0.80	0.260232	0.307397	0.599683
0.82	0.302680	0.367043	0.681382
0.84	0.350886	0.434309	0.753712
0.86	0.405724	0.508905	0.816302
0.88	0.468394	0.589932	0.870003
0.90	0.540558	0.675632	0.915261
0.91	0.580852	0.719368	0.933676
0.92	0.624326	0.763044	0.949442
0.93	0.671211	0.806032	0.962702
0.94	0.721587	0.847564	0.973620
0.95	0.775207	0.886714	0.982369
0.96	0.831195	0.922376	0.989136
0.97	0.887532	0.953233	0.994108
0.98	0.940221	0.977728	0.997469
0.99	0.981954	0.994032	0.999386

probability p. The results of this computation are given in Table I, where the new bound is compared to the previously known Kruskal–Katona and edge-disjoint path bounds described in [5]. It is worth noting here that on occasion, the Ball–Provan *all-terminal* bound improves on the Kruskal–Katona 2-terminal bound, and on the edge-disjoint path 2-terminal bounds; every all-terminal lower bound is also a 2-terminal lower bound. We have *not* included the Ball–Provan bound in the 2-terminal cases, however. A similar computation on the 25-node ladder depicted in Figure 3.2 is reported in Table II. It is somewhat remarkable

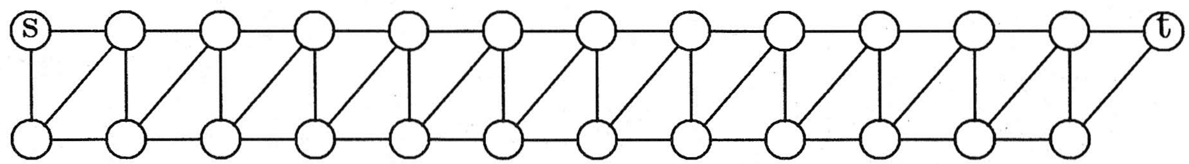

TABLE II. Two-terminal bounds (25 node ladder).

p	Kruskal–Katona	Mincost (edp)	$2t- > 2t$
0.75	0.031682	0.054681	0.054681
0.80	0.068803	0.119917	0.430912
0.82	0.092654	0.161200	0.558991
0.84	0.124041	0.214282	0.669269
0.86	0.165305	0.281396	0.761945
0.88	0.219694	0.364529	0.837486
0.90	0.291856	0.464826	0.896659
0.91	0.336579	0.521297	0.920440
0.92	0.388392	0.581555	0.940574
0.93	0.448415	0.644934	0.957259
0.94	0.517724	0.710375	0.970720
0.95	0.597041	0.776313	0.981207
0.96	0.686113	0.840514	0.989003
0.97	0.782518	0.899895	0.994420
0.98	0.879474	0.950274	0.997801
0.99	0.961964	0.986085	0.999524

that such significant improvements to 2-terminal bounds are obtained by the appropriate application of Theorem 2.1. One possible explanation is that Theorem 2.1 exploits "local structure" of the graph, whereas the edge-disjoint path bounds exploit only the "global" structure of the number and length of s,t-paths.

It is also useful to note that the 2-terminal bounds obtained in this way could, in turn, be used to construct another set of probability bounds for the complete graph and the process repeated. However, this can be rendered unnecessary by a simplification of the method, using the following easy observation:

Lemma 3.1 (the triangle inequality). $1 - l(x,z) \geq (1 - l(x,y)) + (1 - l(y,z))$.

Proof. This follows directly from Theorem 2.1 applied to the 2-terminal case.

With Lemma 3.1 in mind, we can preprocess the complete graph using an all-pairs shortest path algorithm to increase any lower bounds which fail to satisfy the triangle inequality. Once this is done, the edge (x,y) always forms a minimum weight x,y-path.

Of course, Lemma 3.1 can be applied to improve the bounds in the complete graph directly; in the 2-terminal case, this obviates the need for Theorem 2.1. Even in the other cases, we can apply Lemma 3.1 to improve the bounds on the edges of the complete graph prior to applying Theorem 2.1.

4. THE APPLICATION TO ALL-TERMINAL RELIABILITY

In the all-terminal case, minimal pathsets are spanning trees, and minimal cutsets are minimal network cuts; once again, a minimum weight spanning tree and a minimum weight network cut can be found efficiently [14]. Hence we can compute the bounds from Theorem 2.1 efficiently in this case.

At the outset, it is worth remarking that this appears to be the first time that 2-terminal bounds have been used to compute all-terminal bounds. One might expect that since more sophisticated bounds (e.g., Ball–Provan) are available in the all-terminal case, the simple observation of Theorem 2.1 would not provide an improvement. However, considering the 25-node ladder again, we find an improvement over the Ball–Provan bounds (see Table III). Similar improvements are found for the 5×5 grid graph (see Table IV).

Once again, the apparent explanation is that the appropriate application of Theorem 2.1 allows us to exploit local structure. It is particularly interesting to observe improvements in the all-terminal case, since most of the work on efficiently computable bounds has concentrated here.

5. THE APPLICATION TO k-TERMINAL RELIABILITY

Many bounding techniques have been developed, but most cannot be effectively applied to the "hardest" of the three problems, k-terminal reliability. To apply Theorem 2.1 directly, observe that a pathset is a Steiner tree, and a cutset is a

TABLE III. All-terminal bounds (25 node ladder).

p	Ball–Provan	$2t-$ > all-term
0.80	0.152922	0.285760
0.82	0.205936	0.457070
0.84	0.274281	0.600709
0.86	0.359969	0.718239
0.88	0.463470	0.811492
0.90	0.582322	0.882580
0.91	0.645523	0.910550
0.92	0.709467	0.933900
0.93	0.772323	0.952978
0.94	0.831873	0.968147
0.95	0.885609	0.979793
0.96	0.930922	0.988319
0.97	0.965470	0.994149
0.98	0.987767	0.997726
0.99	0.998113	0.999515

Steiner cut. Minimum weight Steiner cuts can be found in polynomial time using network flows [14], but it is NP-hard to find a minimum weight Steiner tree [8], even when the triangle inequality is satisfied. At first, this seems to preclude applying our general strategy. The difficulty can be largely circumvented using heuristic algorithms for Steiner trees. Theorem 2.1 states that when w is the weight of a minimal Steiner tree, the k-terminal reliability is at least $\max(0,1 - w)$. Suppose we compute any Steiner tree, of weight w'. Since necessarily $w' \geq w$, $\max(0,1 - w') \leq max(0,1 - w)$, and hence $1 - w'$ is a lower bound on the k-terminal reliability.

The development of a good lower bound hinges on the accuracy with which we can compute Steiner trees. We consider two strategies. The first strategy was

TABLE IV. All-terminal bounds (5 × 5 grid).

p	Ball–Provan	$2t-$ > all-term
0.80	0.205198	0.0
0.82	0.260718	0.000992
0.84	0.329549	0.252695
0.86	0.413092	0.463491
0.88	0.511367	0.634428
0.90	0.621856	0.767480
0.91	0.679862	0.820687
0.92	0.738132	0.865614
0.93	0.795066	0.902788
0.94	0.848744	0.932792
0.95	0.897007	0.956261
0.96	0.937626	0.973873
0.97	0.968607	0.986337
0.98	0.988698	0.994373
0.99	0.998170	0.998700

TABLE V. k-Terminal bounds (5×5 grid $k = 4$).

p	Ball–Provan	Spanning	Steiner
0.80	0.205198	0.0	0.204861
0.82	0.260718	0.175133	0.376270
0.84	0.329549	0.353877	0.511763
0.86	0.413093	0.510520	0.639370
0.88	0.511367	0.647677	0.736298
0.90	0.621856	0.766342	0.831804
0.91	0.679862	0.815376	0.868273
0.92	0.738132	0.857851	0.899990
0.93	0.795066	0.894044	0.924814
0.94	0.848744	0.924266	0.947472
0.95	0.897007	0.948853	0.964568
0.96	0.937626	0.968168	0.978325
0.97	0.968607	0.982579	0.988234
0.98	0.988698	0.992459	0.994937
0.99	0.998170	0.998160	0.998772

chosen for its simplicity. We compute a minimum *spanning* tree on just the target nodes of the network. The result is a Steiner tree. Our second strategy is a more sophisticated heuristic method due to Wong [17]. Our only adaptation to Wong's method is to excise degree 2 nodes which are not target nodes. The triangle inequality (Lemma 3.1) ensures that we need no degree 2 nodes in the Steiner tree; a degree 2 node with neighbors x and y is therefore deleted and replaced by the edge (x,y). Applying these two heuristics to the 5×5 grid graph from Figure 3.1 shows that the extra computational effort in Wong's method does yield substantial improvements (see Table V). In this example, the four corner nodes are the $k = 4$ required nodes. Another example is given by the same graph using the $k = 16$ outside nodes as required (see Table VI), and a third is the 1979 Arpanet (see Table VII and Figure 6.1).

TABLE VI. k-Terminal bounds (5×5 grid $k = 16$).

p	Ball–Provan	Spanning	Steiner
0.80	0.205198	0.0	0.0
0.82	0.260718	0.0	0.031148
0.84	0.329549	0.167288	0.242797
0.86	0.413093	0.382956	0.453222
0.88	0.511367	0.566819	0.628150
0.90	0.621856	0.716641	0.759554
0.91	0.679862	0.778748	0.811307
0.92	0.738132	0.832416	0.861959
0.93	0.795066	0.877809	0.903084
0.94	0.848744	0.916306	0.931489
0.95	0.897007	0.946466	0.956346
0.96	0.937626	0.968722	0.973547
0.97	0.968607	0.984104	0.986282
0.98	0.988698	0.993693	0.994362
0.99	0.998170	0.998613	0.998698

TABLE VII. k-Terminal (Arpanet $k = \{3,5,55,56\}$).

p	Ball–Provan	Spanning	Steiner
0.960	0.642423	0.946886	0.948431
0.970	0.779466	0.973140	0.973994
0.980	0.901378	0.989747	0.990072
0.990	0.979896	0.997964	0.997964
0.999	0.999922	0.999985	0.999985

It is difficult to establish a standard for comparison in these tables, as no powerful k-terminal bounds appear to be available. One *could* employ the Kruskal–Katona bounds here, but since no efficient algorithm for Steiner trees is likely to exist, one would be forced to underestimate the number of edges in a minimal Steiner tree. The underestimate $k - 1$ could be used in general, but leads to poor bounds. Therefore, in our tables of lower bounds here, we have compared against the Ball–Provan all-terminal lower bound. The justification for this is immediate: any all-terminal lower bound is a k-terminal lower bound for any set of any k nodes.

6. PRACTICAL ISSUES

We have thus far illustrated improvements on simple small networks. One concern is that the method may not be able to handle larger networks, despite its polynomial running time. For a 100-node network, the complete graph con-

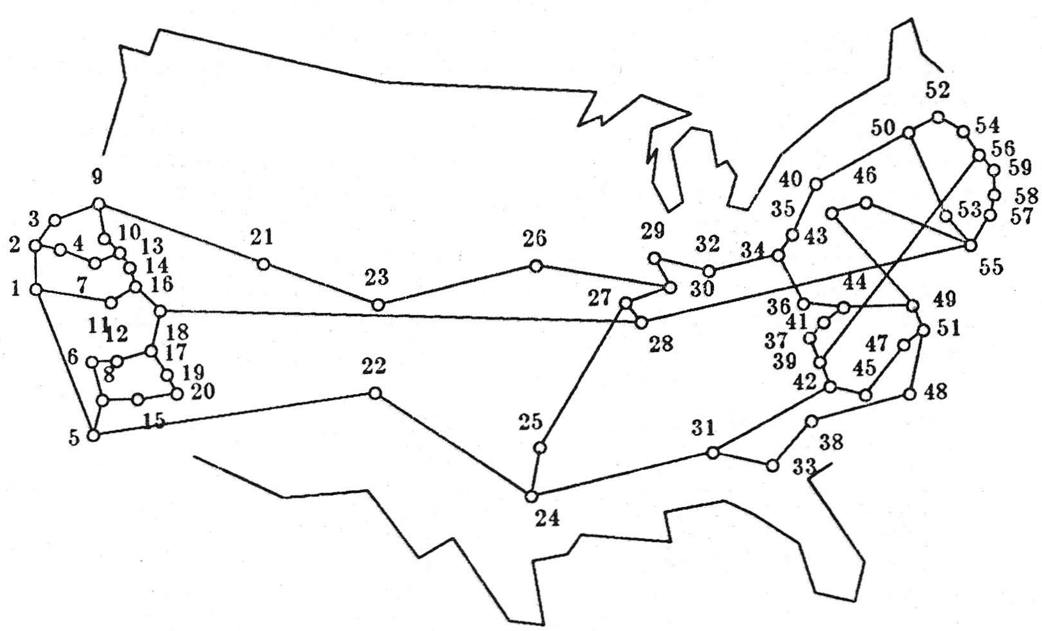

FIG. 6.1 The 1979 Arpanet.

TABLE VIII. Two-terminal bounds (5 × 5 grid) using neighbors.

p	Kruskal–Katona	Mincost (edp)	$2t- > 2t$	Neighbors
0.75	0.175208	0.190203	0.358514	0.234690
0.80	0.260232	0.307397	0.599683	0.544598
0.82	0.302680	0.367043	0.681382	0.645411
0.84	0.350886	0.434309	0.753712	0.732517
0.86	0.405724	0.508905	0.816302	0.805921
0.88	0.468394	0.589932	0.870003	0.865915
0.90	0.540558	0.675632	0.915261	0.913116
0.91	0.580852	0.719368	0.933676	0.932205
0.92	0.624326	0.763044	0.949442	0.948480
0.93	0.671211	0.806032	0.962702	0.962111
0.94	0.721587	0.847564	0.973629	0.973284
0.95	0.775207	0.886714	0.982369	0.982199
0.96	0.831195	0.922376	0.989136	0.989062
0.97	0.887532	0.953233	0.994108	0.994083
0.98	0.940221	0.977728	0.997469	0.997463
0.99	0.981954	0.994032	0.999386	0.999385

structed has 4900 edges; this implies 4900 computations of 2-terminal bounds, which may be unacceptably large. Therefore, let us note that one need not use the same method to compute the reliability bounds for each pair; it is perfectly acceptable to employ a lower bound of 0. If bounds are computed for sufficiently many edges, the lower bounds for the "0 edges" will be modified by the application of Lemma 3.1. We considered another possibility for large networks: computing bounds only for neighboring nodes. For sparse large networks, this approach saves dramatically on the number of basic bounds to compute. The effect of this restriction to the 2-terminal bounds on the 5 × 5 grid graph are given in Table VIII. The results suggest that computational effort can be saved at the expense of loosening the bound.

It is also reasonable to ask whether improvements arise in practical situations. For this reason, we considered the 1979 Arpanet, depicted in Figure 6.1. Improvements to 2-terminal lower bounds via Lemma 3.1 are common here; an example is from node ISI22 (node 5) to CCA (node 56), with improvements given in Table IX. Moreover, k-terminal lower bounds often improve on the Ball–Provan lower bound. An improvement to the all-terminal lower bound is not obtained. However, the lower bound obtained is competitive with the Ball–Provan bound (see Table X).

TABLE IX. Two-terminal bounds (Arpanet s = ISI22 t = CCA).

p	Kruskal–Katona	Mincost (edp)	$2t - > 2t$
0.960	0.835288	0.960673	0.973129
0.970	0.898877	0.980780	0.987606
0.980	0.955207	0.993388	0.995981
0.990	0.991414	0.999038	0.999455
0.999	0.999987	0.999999	0.999999

TABLE X. All-terminal bounds (Arpanet).

p	Ball–Provan	$2t- >$ all-term
0.960	0.642423	0.483473
0.970	0.779466	0.712135
0.980	0.901378	0.874147
0.990	0.979895	0.969321
0.999	0.999922	0.999703

7. CONCLUDING REMARKS

The combination of 2-terminal bounds with statistical independence and bounds with statistical dependence is remarkably fruitful. Efficiently computable bounds for all-terminal, k-terminal, and 2-terminal reliability result which occasionally improve on the best previously known bounds. Moreover, techniques which give 2-terminal bounds can now be effectively used to improve all-terminal bounds, partially putting an end to the division in reliability investigations. Finally, the method proposed yields substantial bounds for k-terminal reliability, especially as p approaches unity.

A major area for further investigation here is to attempt to exploit the extension of Hailperin's model when limited information concerning statistical dependencies is available. When information regarding pairs of edges is available, the so-called second order bounds result [1,6]. The extension of our work here to incorporate second order information may provide some improvement. Another area for future research is the combination of the method proposed here with the linear programming combination in [7].

We thank Steve Lui for providing a Pascal implementation of Wong's algorithm, and Nancy Ross for help with bounds with statistical dependence. Research of the second author is supported by NSERC Canada under grant A0579.

References

[1] J. Y. Assous, Bounds on terminal reliability. Preprint (1985).
[2] M. O. Ball and J. S. Provan, Bounds on the reliability polynomial for shellable independence systems. *SIAM J. Alg. Disc. Meth.* **3** (1982) 166–181.
[3] M. O. Ball and J. S. Provan, The complexity of counting cuts and of computing the probability that a graph is connected. *SIAM J. Computing* **12** (1983) 777–788.
[4] J. A. Bondy and U. S. R. Murty, *Graph Theory with Applications,* Macmillan, London, 1976.
[5] T. B. Brecht and C. J. Colbourn, Lower bounds for two-terminal network reliability. CCNG Report E-127, University of Waterloo (1985).
[6] E. H. Carrasco and C. J. Colbourn, Reliability bounds for networks with statistical dependence. Proc. INFOCOM86, Miami (1986) 290–292.
[7] C. J. Colbourn and D. D. Harms, Bounding all-terminal reliability in computer networks. CCNG Report E-123, University of Waterloo, 1985.
[8] M. R. Garey and D. S. Johnson, *Computers and Intractability*. Freeman, San Francisco (1979).
[9] T. Hailperin, Best possible inequalities for the probability of a logical function of events. *Amer. Math. Monthly* **72** (1965) 343–359.

[10] I. M. Jacobs, Connectivity in probabilistic graphs. Technical Report 356, ERL, MIT, 1959.
[11] G. Katona, A theorem of finite sets. In *Theory of Graphs,* P. Erdös, G. Katona, Eds., Akademia Kiado, Budapest (1966) pp. 187–207.
[12] A. K. Kel'mans, Connectivity of probabilistic networks. *Automat. Remote Control* **29** (1967) 444–460.
[13] J. B. Kruskal, The number of simplices in a complex. In *Mathematical Optimization Techniques,* R. Bellman, Ed., UC Press, Berkeley (1963), pp. 251–278.
[14] E. Lawler, *Combinatorial Optimization: Networks and Matroids.* Holt, Rinehart, Winston, New York (1976).
[15] M. V. Lomonosov and V. P. Polesskii, Lower bound of network reliability. *Prob. Inf. Trans.* **8** (1972) 118–123.
[16] E. F. Moore and C. E. Shannon, Reliable circuits from less reliable relays. *J. Franklin Inst.* **262** (1956) 191–208, 281–297.
[17] R. T. Wong, A dual ascent approach for Steiner tree problems on a directed graph. *Math. Program.* **28** (1984) 271–287.
[18] L. G. Valiant, The complexity of enumeration and reliability problems. *SIAM J. Comput.* **8** (1979) 410–421.
[19] R. M. Van Slyke and H. Frank, Network reliability analysis: part I. *Networks* **1** (1972) 279–290.
[20] E. Zemel, Polynomial algorithms for estimating network reliability. *Networks* **12** (1982) 439–452.

Received January 3, 1986.
Accepted May, 1986.