

## The Delta-Wye Approximation Procedure for Two-terminal Reliability

Manoj K. Chari<sup>1</sup>      Thomas A. Feo<sup>2</sup>  
J. Scott Provan<sup>3</sup>

UNC/OR TR92-18

October 5, 1992

### Abstract

The *Delta-Wye Approximation Procedure (DWAP)* is a procedure for estimating the two-terminal reliability of an undirected planar network  $G = (V, E)$  by reducing the network to a single edge via a sequence of local graph transformations. It combines the probability equations of Lehman — whose solutions provide bounds and approximations of two-terminal reliability for the individual transformations — with the *Delta-Wye Reduction Algorithm* of the second two authors — which performs the corresponding graph reduction in  $O(|V|^2)$  time. A computational study is made comparing the DWAP to one of the best currently known methods for approximating two-terminal reliability, and it is shown that the DWAP produces approximations that are between 10 and 80 times as accurate.

(Figures not included.)

<sup>1</sup>Department of Mathematics, Louisiana State University, Baton Rouge, Louisiana. Work partially supported by LEQSF grant no. (92-94)-RD-A-09 from the Louisiana Board of Regents.

<sup>2</sup>Operations Research Group, Department of Mechanical Engineering, University of Texas at Austin, Austin, Texas

<sup>3</sup>Department of Operations Research, University of North Carolina, Chapel Hill, North Carolina. Work partially supported NSF Grant CCR-9200572.

**Key words:** reliability, connectedness, approximation, planar, network

# 1 Introduction

Of considerable concern to designers of communications, transportation, and electrical/VLSI networks is the *reliability* of the network. The various components (vertices and edges) of a network are typically subject to degradation or failure, due to hardware/software faults, physical damage, or congestion. These occur unpredictably, and can render the component unable to carry the required load. The network designer or user is concerned with the ability of the network to maintain communications or travel between various pairs of vertices in the network in the face of this random failure.

When one considers assessing reliability of a network subject to throughput, transmission speed, congestion, and timing constraints one is dealing with the *performability analysis* problem, and the complexity of the problem usually requires simulating the network performance to get any reasonable assessment of reliability. This is cumbersome and somewhat unsatisfactory from an analysis point of view, and the results are themselves subject to error due to statistical variance in the simulation. A more effective approach would be to produce *bounds* on the required reliability measure, so that one is guaranteed a minimum (or maximum) reliability for the network, subject of course to the accuracy of the input data.

To apply powerful methodology to a complex performability analysis problem, it is usually necessary to make one or both of the following restrictions: *(i)* the performance measure of interest must be a fairly simple one, and *(ii)* the class of problem instances considered must have a sufficient amount of structure for the appropriate machinery to be used. It is important to choose these restrictions in such a way as to capture a measure of significance to the network designer while still being able solve problem instances similar to ones with which the designer is likely to be faced. In this paper we choose the *two-terminal reliability* of a network, that is, the probability that a specified pair of vertices can be connected by a path when edges of the network fail at random. This measure has generally been considered the central one underlying many performability analysis problems. We also restrict our attention to *planar* network instances, that is, instances where the network is laid out in such a way that no two edges cross. This is frequently the case with transportation and local communications networks, and the planarity structure is found extensively in global communications and VLSI networks as well.

This paper presents the *delta-wye approximation technique* for approximating two-terminal reliability in a planar network, by iteratively simplifying the network while maintaining the most accurate possible approximation of the reliability after each simplification. This method is both remarkably simple and at the same time extremely accurate — in particular, the computational studies in this paper show accuracy improvements of *ten to eighty times* those of the best previous approximation techniques. It thus stands as one of the most promising tools in the area of performability analysis.

The two-terminal reliability problem (TTRP) is described by instance  $(G, \mathbf{p})$ , where  $G = (V, E, s, t)$  is a *two-terminal undirected graph* with *vertex set*  $V = \{v_1, \dots, v_n\}$ , *edge set*  $E = \{e_1, \dots, e_m\}$ , and *terminal vertices*  $s$  and  $t$ ; and  $\mathbf{p} = (p_1, \dots, p_m)$  is the set of *edge operating probabilities* with  $0 < p_j < 1$ ,  $j = 1, \dots, m$ . The vertices of  $G$  are assumed to be perfectly reliable and each edge  $e_j$  of  $G$  operates independently with probability  $p_j$ ,  $j = 1, \dots, m$ . The measure of interest for this instance is the *two-terminal reliability* (or  *$(s, t)$ -connectedness reliability*)  $R(G, \mathbf{p})$ , which is the probability that the operating edges of  $G$  admit a path between  $s$  and  $t$ .

Two-terminal reliability has been a notoriously difficult measure to compute exactly for general graphs. This should not be surprising, as TTRP is known to be in the computationally intractable class of *#P-complete* problems [VAL79]. The most efficient worst-case algorithm for TTRP [PRO84] can require as much as order  $3^n$  computations. Planar graphs have a slightly better worst-case algorithm (order  $2^{\sqrt{n}}$  computations [BIE86]) although TTRP on planar graphs is still *#P-complete* [PRO86]. There are a small number of classes of graph for which two-terminal reliability can be computed in polynomial time, the best-known being *series-parallel* graphs [WAL83], [SAT85]. Since computation of exact solutions is intractable except for these very special cases, attention has turned to approximation schemes. The most successful of these to date appear to be variants of *path-* and *cut-packing* schemes, and more general *inclusion-exclusion* schemes which were empirically studied by Brecht and Colbourn [BRE 88] and Colbourn [COL88] and Shier [SHI91] (for a complete account of the schemes, see [COL87] Ch.6.)

The purpose of this paper is to investigate an approximation scheme for TTRP on planar graphs which relies on the use of a *delta-wye reduction* for  $G$ . Specifically, suppose we are given instance  $(G, \mathbf{p})$  for TTRP, with  $G$  a planar graph and  $s$  and  $t$  arbitrary vertices of  $G$ . An example is given in Figure 3a (the labels will be explained in Section 3). We will allow  $G$  to

have *parallel edges*, that is, edges with the same pair of endpoints. We will assume, however, that  $G$  does not have any *irrelevant 2-component*, that is, any subgraph of  $G$  that is attached to the remainder of  $G$  by a single vertex and that does not contain  $s$  or  $t$  except possibly as the attachment vertex. Irrelevant 2-components can be detected in linear time, and their removal will have no effect on the reliability of  $G$ . The combinatorial framework of our approximation algorithm is a procedure to reduce  $G$  to the single edge  $(s, t)$ , using the following set of four transformations:

- T1. *Series reduction*: if  $e = (u, v)$  and  $f = (u, w)$  are edges with  $u$  a nonterminal of degree two, then  $e, f$  and  $u$  are replaced by the single edge  $(v, w)$ .
- T2. *Parallel reduction*: if  $e$  and  $f$  are distinct edges with the same endpoints  $u$  and  $v$ , then  $e$  and  $f$  are replaced by the single edge  $(u, v)$ .
- T3. *Delta-wye transformation*: if  $e = (v, w)$ ,  $f = (w, z)$  and  $g = (z, v)$  are edges, then  $e, f$  and  $g$  are replaced by the three edges  $e' = (u, v)$ ,  $f' = (u, w)$  and  $g' = (u, z)$ , where  $u$  is nonterminal and has no other edge incident to it.
- T4. *Wye-delta transformation*: the inverse of T3.

These transformations are illustrated in Figure 1. Note that transformations T1 and T2 decrease the number of edges in  $G$  (hence the label “reduction”) whereas in transformations T3 and T4 the number of edges in  $G$  remains the same.

A sequence of these transformations which, when performed on  $G$ , reduces  $G$  to a single edge  $(s, t)$  is called a *delta-wye reduction sequence* for  $G$ . An example of a delta-wye reduction sequence is given in Figure 3b. The existence of delta-wye reduction sequence for any arbitrary two-terminal planar graph was conjectured by Akers [AKE60] and Lehman [LEH62], and later proved by Epifanov [EPI66]. Epifanov’s proof is quite difficult and does not lend itself to an algorithm for the reduction. Specific algorithms have been given by Feo [FEO85] and Truemper [Tru89], but these are difficult to implement efficiently. Feo and Provan [FEO92] give a very simple  $O(|V|^2)$  reduction algorithm, and it is this algorithm which is the basis for our study.

The primary questions addressed in this paper are the following:

- How can a delta-wye reduction sequence such as one given by the Feo and Provan algorithm be used to approximate  $R(G, \mathbf{p})$ ?
- How accurate are these approximations in practice ?

It is well-known that for the transformations T1 and T2, it is possible to assign probabilities to the edges in the transformed subgraph in such a way that its reliability is identical to that of the original graph. As indicated in the next section, it is impossible in general to construct reliability maintaining transformations associated with the delta-wye and wye-delta transformations — which is to be expected since the problem is known to be #P-complete. Lehman [LEH63], however, presents three sets of reliability *approximating* transformations for T3 and T4, one of which is guaranteed to *underapproximate* the actual reliability, one of which is guaranteed to *overapproximate* the actual reliability, and one of which lies between these two. Lehman goes on to show that these are the “best” possible underapproximation, overapproximation, or intermediate approximation possible in a local sense.

In this paper an approximation scheme is presented based on the Feo-Provan delta-wye reduction algorithm and the reliability approximating transformations of Lehman. The scheme performs a delta-wye reduction of  $G$ , while at each stage maintaining an edge probability vector which corresponds to either a lower bound, upper bound, or best approximation to the true reliability when evaluated on the graph at that stage. As each graph transformation in the reduction sequence is performed, the edge probability vector is transformed according to the appropriate probability assignments, namely:

- the reliability maintaining transformation if a T1 or T2 transformation is involved;
- the appropriate underapproximation, overapproximation, or intermediate approximation of Lehman if a T3 or T4 transformation is involved.

This means that at each stage of the reduction algorithm the lower bound (upper bound) edge probability are guaranteed to produce a lower bound (upper bound) on the two-terminal reliability of the network, while the best approximation will lie between the two. When  $G$  is reduced to the single edge  $(s, t)$  then its reliability is simply the the probability of that edge, and hence

the appropriate upper bound, lower bound, or intermediate approximation can be read off immediately.

In Section 2 of the paper we outline, on an implementational level, the analysis of Lehman. In Section 3, we give the Delta-Wye Reduction Procedure of [FEO92] and the incorporation of Lehman’s approximations into this algorithm. Section 4 gives a computational study of the resulting bounds, comparing them to some of the best known bounds for two-terminal reliability. These studies indicate that the given scheme produces approximations that are dramatically better than the best known polynomial-time approximation techniques.

## 2 Lehman’s approximation scheme for delta-wye transformations

The graph transformations T1–T4 given above are examples of *local replacements*, that is, the current graph  $G$  is transformed into graph  $G'$  by removing a subgraph  $S$  of some specified type from  $G$  and replacing it with a subgraph  $S'$  of specified replacement type. The associated reliability transformations we consider will likewise be local, in that the only changes in the probability vector  $\mathbf{p}$  occur only on the edges of  $S'$ . The associated two-terminal reliability  $R(G', \mathbf{p}')$  of this instance should be as close to  $R(G, \mathbf{p})$  as possible, in a sense that is independent of the values of the nonreplaced edges. That is, it should minimize the *worst-case error* in the resulting probability over all possible probability values of the nonreplaced edges, as given by

$$\epsilon(\mathbf{p}', \mathbf{p}) = \max\{|R(G', \rho') - R(G, \rho)|\}$$

the maximum being taken over all pairs of edge probability vectors  $\rho$  and  $\rho'$  with  $\rho_e = \mathbf{p}_e, e \in S, \rho'_e = \mathbf{p}'_e, e \in S'$ , and  $\rho_e = \rho'_e$  on the remaining edges.<sup>1</sup> The goal is choose the edge probabilities for edges of the replacement subgraph so as to minimize  $\epsilon(\mathbf{p}', \mathbf{p})$ .

---

<sup>1</sup>Lehman’s analysis assumes nothing about the structure of  $G$  outside of  $S$ , as it must apply in every instance in which a T3 or T4 transformation occurs. As a technical matter, then, this maximum is also taken over all possible graphs  $G \setminus S$ . Unless  $S$  is situated very close to  $s$  or  $t$ , however, the structure of the remaining graph will be sufficient so that the above definition applies.

We now look at the probability assignments that go with the four transformations given in Figure 1. The series and parallel replacements have well-known associated edge probability assignments, which are given in Figure 2. These assignments comprise *exact* reliability transformations, since when the replacement edge is given the probability indicated, we have  $R(G', \mathbf{p}') = R(G, \mathbf{p})$ , i.e.  $\epsilon(\mathbf{p}', \mathbf{p}) = 0$ . For the delta-wye and wye-delta replacements it is known that except in very special cases there can exist *no* assignment of edge probabilities making either of these transformations exact. Lehman [LEH63] has performed a thorough analysis of just how good such a replacement can be, in terms of the measure  $\epsilon(\mathbf{p}', \mathbf{p})$ . The purpose of this section is to give a practical summary of Lehman's analysis.

## 2.1 The open- and short-circuit equations

The two sets of equations which play a key role in Lehman's analysis are called the *open-circuit* and *short-circuit* equations. Referring to Figure 2, they relate the edge probabilities  $(p_e^d, p_f^d, p_g^d)$  of the delta to the edge probabilities  $(p_{e'}^y, p_{f'}^y, p_{g'}^y)$  of the wye.

**open circuit equations (O):**

$$\begin{aligned} p_{f'}^y p_{g'}^y &= p_e^d + p_f^d p_g^d - p_e^d p_f^d p_g^d \\ p_{e'}^y p_{g'}^y &= p_f^d + p_e^d p_g^d - p_e^d p_f^d p_g^d \\ p_{e'}^y p_{f'}^y &= p_g^d + p_e^d p_f^d - p_e^d p_f^d p_g^d. \end{aligned}$$

**short circuit equations (S):**

$$\begin{aligned} p_{e'}^y (p_{f'}^y + p_{g'}^y - p_{f'}^y p_{g'}^y) &= p_f^d + p_g^d - p_f^d p_g^d \\ p_{f'}^y (p_{e'}^y + p_{g'}^y - p_{e'}^y p_{g'}^y) &= p_e^d + p_g^d - p_e^d p_g^d \\ p_{g'}^y (p_{e'}^y + p_{f'}^y - p_{e'}^y p_{f'}^y) &= p_e^d + p_f^d - p_e^d p_f^d. \end{aligned}$$

The probabilistic interpretation of these two sets of equations is as follows: system (O) equates the probabilities that each specified pair among the vertices  $v$ ,  $w$ , and  $z$  are connected through the subgraph, and system (S) equates the probabilities that a specified vertex of  $v$ ,  $w$  or  $z$  is connected to *at least* one other vertex through the subgraph. These six equations are sufficient to completely specify the probabilistic behavior of one subgraph from

that of the other. Unfortunately they are inconsistent in general, and so this probabilistic behavior can only be approximated. We devote the remainder of the section to giving the approximation analysis of Lehman. The readers are referred to [LEH63] for proofs.

## 2.2 Solving the open-and short-circuit equations

- (A) For (O) or (S) taken separately, if the set of probability values for either  $p_{e'}^y, p_{f'}^y, p_{g'}^y$  or  $p_e^d, p_f^d, p_g^d$  is specified, then the other set of probability values is determined *uniquely*.
- (B) The values of  $p_{e'}^y, p_{f'}^y, p_{g'}^y$  are determined from those of  $p_e^d, p_f^d, p_g^d$  through (O) by the following formulae:

$$p_{e'}^y = \sqrt{r_f r_g / r_e}, \quad p_{f'}^y = \sqrt{r_e r_g / r_f}, \quad p_{g'}^y = \sqrt{r_e r_f / r_g},$$

where  $r_e, r_f, r_g$  are the values of the right-hand sides of (O), respectively.

- (C) The values of  $p_{e'}^y, p_{f'}^y, p_{g'}^y$  are determined from those of  $p_e^d, p_f^d, p_g^d$  through (S) by finding the unique root  $0 < z < 1$  in the following cubic equations:

$$p_{e'}^y : \kappa \lambda_e z^3 - (\kappa^2 \mu_f \mu_g + (1 + \kappa) \lambda_e) z^2 + \kappa (\lambda_g \mu_g - \lambda_f \mu_f) z + \lambda_f \lambda_g = 0$$

$$p_{f'}^y : \kappa \lambda_f z^3 - (\kappa^2 \mu_e \mu_g + (1 + \kappa) \lambda_f) z^2 + \kappa (\lambda_e \mu_e - \lambda_g \mu_g) z + \lambda_e \lambda_g = 0$$

$$p_{g'}^y : \kappa \lambda_g z^3 - (\kappa^2 \mu_e \mu_f + (1 + \kappa) \lambda_g) z^2 + \kappa (\lambda_f \mu_f - \lambda_e \mu_e) z + \lambda_e \lambda_f = 0$$

where

$$\begin{aligned} \lambda_e &= p_e^d + (1 - p_e^d) p_f^d p_g^d + \kappa p_e^d (1 - p_f^d) (1 - p_g^d) \\ \lambda_f &= p_f^d + (1 - p_f^d) p_e^d p_g^d + \kappa p_f^d (1 - p_e^d) (1 - p_g^d) \\ \lambda_g &= p_g^d + (1 - p_g^d) p_e^d p_f^d + \kappa p_g^d (1 - p_e^d) (1 - p_f^d) \\ \mu_e &= (1 - p_e^d) (p_f^d - p_g^d) \\ \mu_f &= (1 - p_f^d) (p_g^d - p_e^d) \\ \mu_g &= (1 - p_g^d) (p_e^d - p_f^d) \\ \kappa &= 1 \end{aligned}$$

(D) The values of  $p_e^d, p_f^d, p_g^d$  are determined from those of  $p_{e'}^y, p_{f'}^y, p_{g'}^y$  through (O) and (S), respectively, as follows:

- (i) substitute the variables  $q_e^d, q_f^d, q_g^d, q_{e'}^y, q_{f'}^y, q_{g'}^y$  for  $p_{e'}^y, p_{f'}^y, p_{g'}^y, p_e^d, p_f^d, p_g^d$ , respectively, in the *opposite* set of equations (S) and (O), respectively;
- (ii) using the values  $q_{e'}^y = 1 - p_{e'}^y, q_{f'}^y = 1 - p_{f'}^y$  and  $q_{g'}^y = 1 - p_{g'}^y$ , determine  $q_e^d, q_f^d$  and  $q_g^d$  via either (B) or (C);
- (iii) set  $p_e^d = 1 - q_e^d, p_f^d = 1 - q_f^d, p_g^d = 1 - q_g^d$ .

(Note that this is simply the “dualization” of the problem solved in (C).)

### 2.3 Approximation analysis

Let  $G'$  be obtained from  $G$  by making a delta-wye or wye-delta transformation, and let  $\mathbf{p}^O$  and  $\mathbf{p}^S$  be obtained from  $\mathbf{p}$  by using the solutions of (O) and (S), respectively, as the subgraph replacement probabilities. Let  $\mathbf{p}'$  be any set of edge probabilities for  $G'$  which agrees with  $\mathbf{p}$  outside of the replacement subgraph.

- (E) The extreme values of  $R(G', \mathbf{p}')$  occur when  $\mathbf{p}' = \mathbf{p}^O$  and  $\mathbf{p}' = \mathbf{p}^S$ .
- (F) For a delta-wye transformation, the *sign* of the error  $R(G', \mathbf{p}') - R(G, \mathbf{p})$  is the same as that of  $\delta^d = p_e^d p_f^d p_g^d - (p_e^d + p_f^d + p_g^d) + 1$  when  $\mathbf{p}' = \mathbf{p}^O$ , and opposite that of  $\delta^d$  when  $\mathbf{p}' = \mathbf{p}^S$ . Thus  $\mathbf{p}^O$  and  $\mathbf{p}^S$  provide *upper* and *lower* bounds for  $R(G, \mathbf{p})$ , respectively, when  $\delta^d$  is positive, and *lower* and *upper* bounds, respectively, when  $\delta^d$  is negative.
- (G) For a wye-delta transformation, the sign of  $R(G', \mathbf{p}') - R(G, \mathbf{p})$  is the same as that of  $\delta^y = p_{e'}^y p_{f'}^y p_{g'}^y - (p_{e'}^y p_{f'}^y + p_{e'}^y p_{g'}^y + p_{f'}^y p_{g'}^y) + 1$  when  $\mathbf{p}' = \mathbf{p}^O$ , and the opposite of  $\delta^y$  when  $\mathbf{p}' = \mathbf{p}^S$ . The assignment of lower or upper bounds is, therefore, as above.
- (H) For the transformations given in (F) or (G), the appropriate error  $\epsilon(\mathbf{p}^O, \mathbf{p})$  or  $\epsilon(\mathbf{p}^S, \mathbf{p})$  corresponding to the lower (upper) bound is the smallest possible for any  $\mathbf{p}'$  guaranteed to produce a lower (upper) bound for  $R(G, \mathbf{p})$ , and is bounded above by .0089.

- (I) When the approximation is not required to be an upper or lower bound, the minimum worst case error  $\epsilon(\mathbf{p}', \mathbf{p})$  for a delta-wye transformation occurs when  $p_{e'}^y, p_{f'}^y$ , and  $p_{g'}^y$ , are assigned to be the unique roots  $0 < z < 1$  of the cubic equations given in (C), with  $\kappa$  now set to  $\frac{1}{2}$ . The resulting error here is  $\epsilon(\mathbf{p}', \mathbf{p}) < .0037$ ., but the sign of the error  $R(G, \mathbf{p}') - R(G, \mathbf{p})$  cannot be determined independently of the structure of the rest of the graph, and hence it cannot be determined whether  $R(G', \mathbf{p}')$  is a lower or upper bound on  $R(G, \mathbf{p})$ . It is always the case, however, that  $R(G', \mathbf{p}')$  is between  $R(G', \mathbf{p}^O)$  and  $R(G', \mathbf{p}^S)$ .
- (J) For a wye-delta transformation, the associated minimum worst-case error is achieved by substituting  $1 - p_{e'}^y, 1 - p_{f'}^y, 1 - p_{g'}^y$ , for  $p_e^d, p_f^d$ , and  $p_g^d$  in defining the cubic equations given in (I), finding the appropriate roots  $z_e, z_f$  and  $z_g$ , respectively, and then setting  $p_e^d = 1 - z_e, p_f^d = 1 - z_f$ , and  $p_g^d = 1 - z_g$ . Again,  $\epsilon(\mathbf{p}', \mathbf{p}) < .0037$ . (Again, this is simply the “dualization” of the problem solved in (I).)

Note that while the “local” error value  $\epsilon(\mathbf{p}', \mathbf{p})$  was used in Lehman’s worst-case analysis, the actual error from any of the T3-T4 transformations given above will depend upon the probabilistic structure of the remainder of the network. Thus the optimizing values of  $\mathbf{p}'$  given above may not minimize the actual error (and cannot be expected to, as they depend only on local information). The actual error, however, is always bounded above by  $\epsilon(\mathbf{p}', \mathbf{p})$ , regardless of the structure of the rest of the graph. Further, the given values .0089 and .0037 are (approximately) the smallest possible, in that there exist probability values that can be assigned to the edges of the delta or wye for which the appropriate best approximation actually attains an error of (about) .0089 or .0037.

We proceed to use the appropriate values found in (F) and (G) for the lower and upper bounds in our delta-wye approximation procedure, and the values found in (I) and (J) as the best estimates. We note that the computation of values for the cubic equations given in (C) or (I) could be solved in closed form by classical techniques, although the precise closed form will depend upon the values of the original edge probabilities. In practice, a more efficient method is to simply apply Newton’s method starting anywhere in the interval (0,1).

### 3 The delta-wye reduction of planar graphs

In this section we outline the Delta-Wye Reduction Procedure given in [FEO92], which reduces a two-terminal planar graph to the single edge  $(s, t)$  by means of a polynomial number of transformations of type T1–T4. By applying the appropriate edge reliability assignments given in Section 2 for each transformation, we will have a valid approximation scheme for the two-terminal reliability problem.

We assume that  $G$  has been laid out on the plane with no crossing edges, and more specifically, that the appropriate data structures have been produced which can, in constant time per operation, (i) determine the two faces of  $G$  incident to a given edge, (ii) determine the degree (number of adjacent edges) of any face or vertex, and (iii) traverse the edges incident to a given face or vertex in clockwise order. (The noncrossing layout of  $G$  can be constructed in linear time using [HOP74], and the appropriate data structures can be produced in linear time using [WHI90].) The Delta-Wye Reduction Procedure consists of two phases, a *labeling phase* and a *reduction phase*. The labeling phase assigns to each edge and vertex of  $G$  a label indicating in a certain sense how far that edge or vertex lies from the terminal  $s$ . It consists of the following iterative procedure:

### Labeling Phase

Assign the label 0 to  $s$ , and declare all other vertices and edges unlabeled.

Set  $l = 0$  [ $l =$  current level]

*do while* there are unlabeled vertices or edges:

1. To each unlabeled edge incident to a labeled vertex, assign the label  $l + 1$ .
2. To each unlabeled edge sharing a face with a labeled vertex, assign the label  $l + 2$ .
3. To each unlabeled vertex incident to a newly labeled edge, assign the label  $l + 2$ .
4. Set  $l := l + 2$ .

*end do while*

The example graph given in Figure 3a illustrates the labeling. With the associated data structure given with the plane layout, the labeling phase requires  $O(|V|)$  steps.

The reduction phase amounts to the identification and processing of a special class of the transformations T1–T4, called *positive transformations*, as given in Figure 4. These transformations depend precisely on the labels of the associated edges, and result in a change of labels of edges of the transformed subgraph. It is easy to see that resultant labeling is consistent with one obtained if the labeling algorithm were applied directly to the new graph.

The procedure can thus be given as follows:

### Reduction Phase

*do while*  $G$  is not the single edge  $(s, t)$

Find and perform a positive transformation as given in Figure 4, with the priority that T1 and T2 transformations are always chosen before T3 and T4 transformations.

*end do while*

Figure 3b gives a possible positive reduction sequence with respect to the labeling given in Figure 3a. It should be noted that the terminal  $t$  can appear as a *pendant* (degree 1) vertex lying inside one of the regions required to be empty in a positive transformation. In this case the pendant edge should be moved to another region so that the transformation can proceed as given.

In [FEO92] the following two facts are established:

1. If  $G$  is not the single edge  $(s, t)$ , there always exists a positive transformation which can be applied to  $G$ .
2. The reduction phase reduces  $G$  to a single edge after at most  $3|V|^2$  such transformations.

As it is given in [FEO92] the Delta-Wye Reduction Procedure actually uses two additional transformations, one that removes a pendant edge whose degree 1 vertex is a nonterminal, and the other that removes a loop (edge having identical endpoints). The following lemma alleviates the necessity of using these reductions.

**Lemma 1** *Let  $G$  be a two-terminal planar graph with no irrelevant 2-components. Then the reduction phase given above never creates a loop or nonterminal pendant edge.*

**Proof** Since  $G$  has no irrelevant 2-components, then it initially can have no loops or nonterminal pendant edges. We now prove that when transformations are performed using the priorities given in the Reduction Procedure, there will be no irrelevant 2-components created, and hence no loops or nonterminal pendant edges created. Proceeding by contradiction, consider the first transformation which creates an irrelevant 2-component, and let  $G'$  be the graph occurring immediately before this transformation. Now the only possible transformation for  $G'$  which can create a new irrelevant 2-component is the delta-wye transformation T3, and the only new irrelevant component that can be created is an nonterminal pendant edge attached to the vertex  $u$  of the wye. But this means that the original delta had to have had a degree 2 nonterminal vertex, and so the reduction phase would have chosen this T1 reduction first. Thus such a transformation can never occur, and the lemma follows.  $\square$

The data structures of the plane layout can also be used to identify and perform positive transformations in constant time. This is done by keeping track of the set of potential positive transformations at each step of the graph reduction. Since the degree of each face and vertex is part of the input data of the planar layout, it is easy to run through the initial labeled graph and test each vertex and face of degree 2 or 3 for admitting one of the four positive transformations. If so, the candidate is put into one of two stacks holding, respectively, the candidate T1 and T2 transformations and the candidate T3 and T4 transformations, and the stacks are queried according to the priority given by the reduction procedure. When a transformation is performed, the (at most six) adjacent regions and vertices affected by the transformation are updated, and if a new face or vertex of degree 2 or 3 is detected, then the labeling is checked to see if the associated transformation is a positive transformation. If so, the candidate transformation is added to the appropriate stack. It may be that some stack items no longer correspond to valid transformations, due to the intervening modifications which have been made to the graph. In this case the transformation is simply discarded when it comes to the top of the stack. Note that since a performed transformation can affect at most six other faces and vertices, it can affect at most six stack elements. Thus the number of discarded transformations is no more than six times the total number of transformations (and in practice is nowhere near this number). It follows that the entire delta-wye-reduction algorithm can be implemented in time  $O(|V|^2)$  as promised.

It is now a straightforward matter to combine the approximation technique of Section 2 with the Delta-Wye Reduction Procedure. Letting  $(G^0, \mathbf{p}^0)$  be the initial instance of TTRP, we apply the reduction phase to  $G^0$ . At each stage of the procedure we maintain the current graph  $G$ , together with three associated edge-probability vectors  $\mathbf{p}^l$ ,  $\mathbf{p}^u$  and  $\mathbf{p}^*$ , for which the value of  $R(G, \mathbf{p})$  at each of these three vectors is a *lower bound*, *upper bound* and *best estimate* for  $R(G^0, \mathbf{p}^0)$ , respectively. As transformations are performed on  $G$ , the three associated probability vectors are transformed accordingly, until  $G$  is reduced to a single edge, at which point the upper bound, lower bound, and best estimate for  $R(G^0, \mathbf{p}^0)$  are simply  $p_{st}^u$ ,  $p_{st}^l$ , and  $p_{st}^*$ , respectively. The complete procedure is outlined below. The computations of the upper and lower bounds on the example graph of Figure 3, with  $p_{uv}^0 = .5$  for all edges, are illustrated in Figures 5a and 5b.

### Delta-Wye Approximation Procedure

**Input:** TTRP instance  $(G^0, \mathbf{p}^0)$ .

Apply the labeling phase to  $G^0$ .

Set  $G = G^0, \mathbf{p}^l = \mathbf{p}^u = \mathbf{p}^* = \mathbf{p}^0$

*do while*  $G$  is not the single edge  $(s, t)$ .

1. Identify positive transformation  $T$  on  $G$  obtained in the reduction phase and modify  $G$  accordingly.
2. If  $T$  is a series or parallel transformation, then make the associated exact edge-probability replacements in each of  $\mathbf{p}^l, \mathbf{p}^u$  and  $\mathbf{p}^*$  as given in Figure 2.
3. If  $T$  is a delta-wye or wye-delta transformation, then:
  - a. Using (F) or (G) in Section 2.3, find the edge probabilities on the replacement subgraph which give the *lower bound* for  $R(G, \mathbf{p}^l)$ , and make the appropriate replacement in  $\mathbf{p}^l$ .
  - b. Again using (F) and (G) in Section 2.3, find the edge probabilities for the replacement subgraph which give the *upper bound* for  $R(G, \mathbf{p}^u)$ , and make the appropriate replace in  $\mathbf{p}^u$ .
  - c. Using (I) or (J) in Section 2.3, find the edge probabilities which give the *minimum worst-case error* on  $R(G, \mathbf{p}^*)$ , and make the appropriate replacement in  $\mathbf{p}^*$ .

*end do while*

**Output:** lower bound  $p_{st}^l$ , upper bound  $p_{st}^u$  and a best estimate  $p_{st}^*$

**Main result** *The Delta-Wye Approximation Procedure has a running time of  $O(|V|^2)$  and gives valid lower and upper bounds for  $R(G^0, \mathbf{p}^0)$ . All three output values are the best possible respective values for the given sequence of transformations, in the sense that for each transformation the smallest error of  $\epsilon(\mathbf{p}, \mathbf{p}')$  is obtained.*

### Applying the DWAP to nonplanar graphs

Although the DWAP depends critically on the planar structure of the graph, it can be modified to apply to “nearly planar” graphs to give a partial delta-wye reduction of the graph. The procedure is as follows:

#### Nonplanar Delta-Wye Approximation Heuristic

1. Find a layout of  $G$  in the plane having a small number of crossing edges.
2. Remove these edges, declaring the endpoints to be terminals, so that they are not allowed to be removed in a T1 or T4 transformation.
3. Apply the DWAP to the resulting planar graph with the modification that whenever two endpoints of a removed edge appear on the same face of some intermediate graph, the edge is put back into the graph with its original edge probability.

It is easy to see that Nonplanar Delta-Wye Approximation Heuristic will find and perform valid transformations as does the Delta Wye Approximation Procedure, and hence if  $G$  is reduced to a single edge then the resulting probabilities are valid lower bounds, upper bounds, and best estimates for  $R(G, \mathbf{p})$ . (An important example of this given in Section 4.) The Nonplanar Delta-Wye Approximation Heuristic may, of course, fail to find any transformations at some point before  $G$  becomes a single edge (one such graph can be constructed from the four-terminal example given in [FEO92]).

The reliability approximation technique given here can in fact be applied to any delta-wye reduction sequence, whether or not it is obtained from the Delta-Wye Reduction Procedure. Thus its efficiency could be improved by finding a more efficient method of delta-wye reducing a planar graph. What is more, there are *nonplanar* graphs which are nevertheless delta-wye reducible

(for example, the complete bipartite graph on six points). Thus there could exist efficient methods for delta-wye reducing a larger class of graphs than that of planar graphs, and hence the Delta-Wye Approximation Procedure could also be applied to this larger class. This applies as well to the Nonplanar Delta-Wye Approximation Procedure, if a delta-wye reduction sequence could be found which avoided the extra created terminals whenever possible. A recent algorithm of Gitler [GIT91] states that three terminal planar graphs are delta-wye reducible to the complete graph on three terminals. Thus the Nonplanar Delta-Wye Approximation Heuristic can be successfully applied to any graph that can be made planar by removing an edge adjacent to one of the two terminals (so that exactly one extra terminal is created). Even when it is not possible to perform a delta-wye reduction directly as stated above, the reduction could be used in conjunction with other methods such as a factoring to complete the approximation procedure. In view of the impressive computational results given in this paper, such an extension would be quite significant.

## 4 Computational Results

This section gives results of a computational study of the procedures described in Section 3. All computations were performed on an IBM-PS/2-286 machine with a 387 math coprocessor. The major portion of the study was performed on a special class of grid graphs. Specifically, for positive integers  $m$  and  $n$  define the  $m \times n$  grid graph  $G_{m,n}$  as follows. The vertices of  $G_{m,n}$  are represented by the rectilinear grid of points  $(i, j)$ ,  $i = 1, \dots, n - 1$ ,  $j = 1, \dots, m$ , together with the two additional terminal points  $s, t$ . The edges of  $G_{m,n}$  consist of the edges between each pair of vertically and horizontally adjacent grid points, together with the set of edges from  $s$  to each of the points  $(1, j)$ ,  $j = 1, \dots, m$  and from each of the points  $(n - 1, j)$  to  $t$ ,  $j = 1, \dots, m$ . Figure 6a shows the graph  $G_{4,4}$ . The graphs  $G_{m,n}$  represent a collection planar graphs which become intractable to all of the known exact algorithms for TTRP's as  $m$  and  $n$  grow large. Another valuable feature of these graphs is that when  $m = n$ , they are *self-dual*, which means in this context that  $R(G, (0.5, 0.5, \dots, 0.5)) = 0.5$ . Since midrange estimates of  $R(G, \mathbf{p})$  are particularly poor for all approximation schemes (see e.g. [COL88]), this provides a good benchmark for testing the accuracy of the approximation in

a worst-case scenario.

Two other specific graphs were tested for historical purposes. The graph  $G_{DO}$  is the edge graph of the dodecahedron, as shown in Figure 6b. It was first studied in [FIS86] and has shown up frequently as a test case for reliability approximations. Our approximation for  $G_{DO}$  can be compared with exact results obtained by Shier [SHI88]. The other example is the popular ARPANET, denoted by  $G_{AR}$  and shown in Figure 6c. This nonplanar graph has typically been used to compute all-terminal reliability (see [COL87]), but it provides a good example to demonstrate the application of the nonplanar procedure given at the end of the last section. Specifically, consider the computation of  $(s, t)$ -connectedness reliability between points DTI and CMU in  $G_{AR}$ . Removal of any edge between these two points, say the one between DTI and ANL, turns out to cause the remaining graph to be planar. It is therefore possible to make the point ANL a third terminal and the delta-wye reduction procedure will reduce the resulting graph to the 3-edge graph consisting of edges (DTI,CMU), (CMU,WPA), and (WPA,ANL). Since the vertices DTI and ANL are clearly on the same face of this graph, then the edge between them can be added back into the graph, and the final three series reductions will reduce the graph to a single edge, for which the  $(s, t)$ -connectedness reliability estimates for  $G_{AR}$  can be computed.

## 4.1 Equal Probabilities

A major portion of our testing was carried out for the case of equal operating probability  $p$  on all edges. This allows for a good measure of overall quality of estimates, in that we can consider  $R(G, \mathbf{p})$  as a function of the single variable  $p$ , which we will henceforth denote simply by  $R(G, p)$ . We can then compare graph plots of the various estimator functions of  $R(G, p)$  as  $p$  ranges from 0 to 1. The quality of a pair  $R^l(G, p)$  and  $R^u(G, p)$  of lower and upper bounding functions for  $R(G, p)$  can be compared using the  $p$ -norm measure defined

$$\int_0^1 [R^u(G, p) - R^l(G, p)] dp.$$

We compared our bounds to the a set of bounds obtained by two path- and cut-based approximations for two-terminal reliability. For any subset  $P_1, \dots, P_q$  of  $(s, t)$ -paths and  $C_1, \dots, C_r$  of  $(s, t)$ -paths, if  $\mathcal{P}_j$  is the event that

all edges of  $P_j$  operate and  $\mathcal{C}_i$  is the event that all edges of  $C_i$  fail, then

$$Pr\{\cup_{i=j}^q \mathcal{P}_j\} \leq R(G, p) \leq 1 - Pr\{\cup_{i=1}^r \mathcal{C}_i\}$$

In particular, if the  $P_j$ 's are edge-disjoint, each of cardinality  $\rho_j$ , and the  $C_i$ 's are edge-disjoint, each of cardinality  $\gamma_i$  then we have the *edge packing* bounds studied in [BRE88] and [COL88]:

$$1 - \prod_{j=1}^q (1 - p^{\rho_j}) \leq R(G, p) \leq \prod_{i=1}^r [1 - (1 - p)^{\gamma_i}] \quad (*)$$

Despite their apparent simplicity, the edge-packing bounds have been shown to be one of the most effective bounding technique available for the two terminal reliability problem.

We compared these edge-packing bounds to our bounds for the the dodecahedron graph ( $G_{DO}$ ), the ARPANET ( $G_{AR}$ ), and the  $4 \times 4$  grid graph ( $G_{4,4}$ ). The packing bounds for  $G_{DO}$  used the three disjoint (shortest) paths and five disjoint  $(s, t)$ -cutsets, which are indicated in Figure 6b by thickened edges and dashes, respectively. The bounds for this case are

$$1 - (1 - p^5)^3 \leq R(G_{DO}, p) \leq [1 - (1 - p)^3]^2 [1 - (1 - p)^6]^3.$$

The packing bounds for  $G_{AR}$  used the three paths of length 3, 7, and 17 indicated in Figure 6c by thickened edges, and by the three 3-edge  $(s, t)$ -cuts of edges adjacent to DTI, edges adjacent to CMU, and edges one-removed from CMU. The bounds for this case are

$$1 - (1 - p^3)(1 - p^8)(1 - p^{17}) \leq R(G_{AR}, p) \leq [1 - (1 - p)^3]^3.$$

The packing bounds for  $G_{4,4}$  used the four horizontal paths and the four vertical cuts, all of length 4. The bounds for this case are

$$1 - (1 - p^4)^4 \leq R(G_{4,4}, p) \leq [1 - (1 - p)^4]^4.$$

These packing collections offer the best possible edge packing bounds for most reasonable probability assignments. The edge-packing bounds were computed and compared to the delta-wye bounds, using the  $p$ -norm, and the results are given in Table 1. The delta-wye bounds were 20 to 80 times tighter than the edge packing bounds for these three graphs.

We also compared the delta-wye bounds for the  $m \times n$  grid graphs to a stronger set of path- and cut-based bounds — which we will refer to as *lattice bounds* — introduced by Shier [SHI88]. In particular, we applied (\*) to the following collection of paths and cuts for  $G_{m,n}$ :

$\mathcal{Q}_{m,n}$ : The  $m$  horizontal paths  $P_1, \dots, P_m$ , together with the  $(m-1)(n-1)$  paths  $P_{m+1}, \dots, P_{m+(m-1)(n-1)}$  formed, for  $j = 1, \dots, m-1$ ,  $i = 1, \dots, n-1$  by combining the leftmost  $i$  edges of  $P_j$  with the rightmost  $n-i$  edges of  $P_{j+1}$ , together with the edge  $((i, j), (i, j+1))$ .

$\mathcal{R}_{m,n}$ : The  $n$  vertical cuts  $C_1, \dots, C_n$ , together with the  $(n-1)(m-1)$  cuts  $C_{n+1}, \dots, C_{n+(n-1)(m-1)}$  formed, for  $i = 1, \dots, n-1$ ,  $j = 1, \dots, m-1$  by combining the top  $j$  edges of  $C_i$  with the bottom  $m-j$  edges of  $C_{i+1}$ , together with the edge  $((i, j), (i+1, j))$ . (The elements of  $\mathcal{R}_{m,n}$  are in fact “dual” to the elements of  $\mathcal{Q}_{n,m}$ ).

The collections  $\mathcal{Q}_{m,n}$  and  $\mathcal{R}_{m,n}$  include maximal collections of disjoint paths and cuts such as those given for  $G_{4,4}$ , and so (\*) will necessarily give tighter bounds than the edge-packing bounds. We first indicate how to compute the left-hand value in (\*). Note that no two paths in  $\mathcal{Q}_{m,n}$  cross, so that for any pair  $P_i$  and  $P_j$  we have either that  $P_i$  lies “below”  $P_j$  — which we denote by  $P_i \prec P_j$  — or vice versa. Thus if we define  $F_i$  to be the event that the path  $P_i$  is the “lowest” operating path in  $\mathcal{Q}_{m,n}$ , then the events  $F_i$  are well-defined and clearly mutually exclusive. Hence if we set  $L(m, n; p) = \sum_j Pr\{F_j\}$ , we have that

$$L(m, n; p) = Pr\{\cup_j F_j\} = Pr\{\cup_j E_j\}$$

and thus  $L(m, n; p)$  provides the correct left-hand-side value for (\*). To compute the  $Pr\{F_j\}$  terms in  $L(m, n; p)$ , we use the special structure of this collection of paths to produce the following recursive formula (for the case of equal operating probabilities) whose derivation can be found in [SHI88].

$$Pr\{F_j\} = Pr\{E_j\} - \sum_{P_i \prec P_j} Pr\{F_i\} p^{|P_j \setminus P_i|}$$

Thus the  $Pr\{F_j\}$  values can be computed in increasing order of  $\prec$ , and so  $L(m, n; p)$  can be computed in polynomial time. The upper bound for (\*) can be shown by a completely analogous “dual” argument to be equal to  $1 - L(n, m; 1-p)$ , and so (\*) can be written as

$$L(m, n; p) \leq R(G_{m,n}, p) \leq 1 - L(n, m; 1-p)$$

The lattice bounds for seven grid graphs were computed, and their  $p$ -norms were compared to those derived for the delta-wye approximation. (This was done by computing 50 point evaluations the bounds, and averaging the differences between the respective upper and lower bound values. For the delta-wye algorithm the the point evaluations were computed simultaneously in one pass of the reduction routine.) The results are shown in Table 2. As is to be expected, the  $p$ -norms of the approximations are directly related to the number of delta-wye reductions that are required by the reduction algorithm. It is evident, however that the delta-wye bounds again offer a dramatic improvement over the lattice bounds, with  $p$ -norms 10–15 times smaller. Figure 7a–7c shows the plots of the reliability bounds for the examples given in Figures 6a–6c, and the associated  $p$ -norm values are given in Table 1. Figure 7d shows an almost perfect fit of the delta-wye best estimate to the exact values of the reliability for the graph  $G_{DO}$ . Figure 7e shows plots of the three bounds on  $G_{15,15}$ , which are also remarkably accurate considering the size of the graph.

Table 3 compares the point-values of the delta-wye upper bound, lower bound, and best estimate to those of using the lattice bounds on 4 self-dual grid graphs at the point  $p = 0.5$ . (Recall that the exact reliability value for these examples is always 0.5.) Again notice the dramatic improvement. It is interesting to note that while the delta-wye upper and lower bounds deteriorate as the graphs get large, the best estimates, surprisingly, appear to converge to the correct value.

## 4.2 Unequal probabilities

We also computed the delta-wye bound for  $G_{DO}$ ,  $G_{5,5}$  and  $G_{5,6}$  using individually and randomly generated probabilities on each edge, chosen within a variety of ranges. We compared our output with exact values which were computed on a Macintosh Quadra 660 using a factoring algorithm coded PASCAL by Page and Perry [PAGE88]. We thank Lavon Page for providing us with a copy of the source code and his permission to include the computations obtained with it. Table 4 shows the computational results for the three graphs. We also list in each case the spread (the difference between the upper and lower bounds) and the error in the point estimate, and compare the two. Again, the estimates were remarkably accurate — with errors between .00001 and .00025 and spreads between .0001 and .04.

It is important to point out that each reliability evaluation using the factoring algorithm required roughly 30 seconds for the dodecahedron, 15 minutes for the  $5 \times 5$  grid graph, and *6 hours* for the  $5 \times 6$  grid graph (hence the small number of examples shown). Clearly, the factoring algorithm or any other algorithm would not be able to compute the two-terminal reliability for graphs much larger than our examples.

Our empirical results suggest that the spread in the bounds for a particular graph varies significantly over varying edge probabilities even when the overall reliabilities are similar and the same set of transformations is used. (This can be seen by comparing the second set of estimates for  $G_{5,5}$  in Table 4 with the estimates for this graph in Table 3.) Several empirical results seem to emerge, however. One is that for the same combinatorial framework and two different edges probability vectors, the vector with smaller variation over the edges tends to result in greater spread. Further, amongst the instances with equal probabilities, the maximum spread tends to occur around the fixed point of the polynomial  $R(G, p)$ . This is consistent with what has been observed empirically with other bounding methods as well (See [COL88]). Thus Table 3 gives upper and lower bounds for what is probably the worst case error scenario for the self-dual grids. The dominant factor in determining the accuracy in the delta-wye bounds, however, seems to be the number of delta-wye transformations made on the graph rather than either the topology of the graph or the precise distribution of edge probabilities.

### 4.3 Conclusion

The delta-wye approximation procedure given in this paper has proven to be an easy-to-implement and remarkably accurate estimator for two-terminal reliability, one that is better by an order of magnitude than the best known estimators. It is particularly interesting to compare the overall effectiveness of the estimators to the errors indicated for the individual T3 and T4 transformations. Lehman's analysis gives a worst-case bound on the individual errors as approximately .0089 for the upper and lower bounds, and .0037 for the best estimate; in the actuality the average per-transformation error seems to fall between .003 and .000006 for the bounds and between .0002 and .0000005 for the best estimate. Further these per transformation errors decrease as the number of transformations grows. Not surprisingly, the analysis shows that the accumulated error between the upper and lower bounds in-

creases as the number of transformations grows, reflecting the additive effect of the errors in the individual transformations. Table 3, however, shows the striking result that the cumulative error of the *best estimates* seem to actually *decrease* as the graphs get larger, indicating more of an averaging or canceling effect of the cumulative transformations. We know of no explanation for this unusual effect.

We end the paper by giving some more complex and realistic performability analysis problems to which one might apply an analogous delta-wye approximation technique.

**nonplanar graphs** (as discussed in Section 3): This might include adding transformations to the four discussed here that will allow the reduction of graphs which are not necessarily planar, and developing the reliability transformations necessary to yield the associated approximation analysis.

**k-terminal reliability**: Here the problem input includes a given set  $K$  of terminals, and the network operates if all pairs of vertices in  $K$  are connected by operating paths.

**directed networks**: The networks discussed in this paper are required to have two-way edges, and there does not appear to be an immediate way to extend this analysis to the case of one-way or mixed edges.

**stochastic shortest path reliability**: Here each edge is given a *length* and there is a specified minimum “delay” tolerance  $L$  for the network. The network operates if vertices  $s$  and  $t$  are connected by an operating path of length at most  $L$ .

**stochastic network flow reliability**: Here each edge is given a *capacity*, and there is a specified minimum “throughput”  $F$  for the network. The network operates if the operation arcs admit a flow of value at least  $F$  between vertices  $s$  and  $t$ . This and the stochastic shortest path problem are particularly interesting, since their nonstochastic versions can be solved exactly by the delta-wye reduction technique (see [FEO92]).

In most performability problems, including the ones given above, it is fairly simple to find some kind of reliability transformations that will give bounds/approximations to the associated performance probability; the key problem is

to choose those transformations that will give the most accurate estimates. Due to the increased complexity of problems like those given above, it is likely that the estimates will be less accurate, as well as more difficult to analyze, than for the two-terminal reliability problem. It is important to note, however, that the general purpose performability analysis bounding techniques currently used provide fairly poor estimates, or can only be used effectively for very small instances. In view of the remarkable success of the delta-wye reduction technique when applied to two-terminal reliability, we think that it is well worthwhile to study this technique for more general performability analysis problems.

## References

- [AKE60] Akers, S. B. (1960). The use of wye-delta transformations in network simplifications, *Operations Research* **8**, 311–323.
- [BIE86] Bienstock, D. (1986). An algorithm for reliability analysis of planar graphs, *Networks* **16**, 411-422.
- [BRE88] Brecht, T. B. and C. J. Colbourn (1988). Lower bounds for two-terminal network reliability, *Discrete Applied Mathematics* **72**, 49–61.
- [CHA92] Chari, M. K. and J. S. Provan (1994). Calculating  $K$ -connectedness reliability using Steiner bounds, Tech. Rep. 92/17, Operations Research Department, University of North Carolina, Chapel Hill.
- [COL87] Colbourn, C. J. (1987). *The Combinatorics of Network Reliability*, Oxford University Press.
- [COL88] Colbourn, C. J. (1988). Edge packings of graphs and network reliability, *Discrete Mathematics* **72**, 49-61.
- [EPI66] Epifanov, G. V. (1966). Reduction of a plane graph to an edge by a star-triangle transformation, *Doklady* **166**, 13-17.
- [FEO85] Feo, T. A. (1985), Efficient reduction of planar networks for solving certain combinatorial problems, Ph.D. dissertation, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA.

- [FEO92] Feo, T. A. and J. S. Provan (1993). Delta-wye transformations and the efficient reduction of two-terminal planar graphs, *Operations Research* **41**, 572–582.
- [GIT91] Gitler, I. Delta-wye reducibility of graphs, Ph.D thesis, Dept. of Combinatorics and Optimization, University of Waterloo.
- [HOP74] Hopcroft, J. E. and R. E. Tarjan (1974). Efficient planarity testing, *Journal of the ACM* **21**, 549–568.
- [LEH63] Lehman A. (1963). Wye-delta transformations in probabilistic network, *Journal of SIAM* **11**, 773–805.
- [PAGE88] Page , L.B. and J.E. Perry, A practical implementation of the factoring algorithm for network reliability,(1988) *IEEE transactions on reliability*, **37** 259–267.
- [PRO84] Provan, J. S. and M. O. Ball (1984). Computing network reliability in time polynomial in the number of cuts *Operations Research* **15**, 694–702.
- [PRO86] Provan J. S. (1986). The complexity of reliability computations in planar and acrylic graphs, *SIAM Journal on Computing* **15**, 694–702.
- [SAT85] Satyararayana, A. and R. K. Wood (1985). A linear-time algorithm for computing K-terminal reliability in series-parallel graphs, *SIAM Journal on Computing* **14**, 818–832.
- [SHI88] Shier, D. R. (1988). Algebraic Aspects of Computing Network Reliability, in *Applications of Discrete Mathematics* R. D. Ringeisen and F. S. Roberts, eds., SIAM Publications, Philadelphia.
- [SHI91] Shier, D. R. (1991), *Network Reliability and Algebraic Structures*, Oxford Science Publications, Oxford University Press, New York.
- [TRU89] Truemper, K. (1989). On the delta-wye reduction for planar graphs, *Journal of Graph Theory* **13**, 141–148.
- [VAL79] Valiant, L. G. (1979). The complexity of enumeration and reliability problems, *SIAM Journal on Computing* **8**, 410–421.

- [VAN72] Van Slyke, R. M. and H. Frank (1972). Network Reliability Analysis: Part I, *Networks* **1**, 279–290.
- [WAL83] Wald, J. A. and C. J. Colbourn (1983). Steiner trees in probabilistic networks, *Microelectronics and Reliability* **23**, 837–840.
- [WHI90] Whited, D. E., D. R. Shier, and J. P. Jarvis (1990). Reliability computations for planar networks, *ORSA Journal on Computing* **2**, 46–60.

graph	$p$ -norm for edge-packing bounds	$p$ -norm for delta-wye bounds
$G_{4,4}$	.2355	.0125
$G_{DO}$	.3057	.0156
$G_{AR}$	.2968	.00362

Table 1. Comparison of delta-wye and edge-packing bounds for example graphs

grid	# nodes	# edges	# transf. required	$p$ -norm for lattice bounds	$p$ -norm for d-y bounds	time in seconds
$G_{5,5}$	22	41	56	.1797	.0166	4
$G_{5,10}$	47	86	156	.2510	.0221	7
$G_{10,10}$	92	181	555	.3657	.0289	20
$G_{15,10}$	131	276	1000	.4182	.0330	30
$G_{15,15}$	212	421	1999	.4840	.0335	60
$G_{20,20}$	382	761	4865	.5640	.0401	180

Table 2. Comparison of delta-wye and lattice bounds for grid graphs

grid	lattice lower bound	Delta-wye estimates			lattice upper bound
		lower bound	best estimate	upper bound	
$G_{5,5}$	.2481	.479	.5025	.527	.7518
$G_{10,10}$	.03182	.439	.5018	.563	.9681
$G_{15,15}$	.0022	.400	.5011	.595	.9978
$G_{20,20}$	.000126	.364	.5002	.625	.999874

Table 3. Comparison of delta-wye and lattice estimates for self-dual grid graphs at  $p = .5$

graph	lower	point	exact	upper	error	spread	error/spread
$G_{D0}$	.014400	.016362	.016309	.018751	.000053	.004351	.0122
	.025663	.028188	.028953	.031604	.000765	.005941	.1288
	.033343	.035530	.035662	.038153	.000132	.004810	.0274
	.134574	.141700	.144207	.151480	.002507	.016906	.1483
	.279837	.296042	.296412	.317333	.000370	.037496	.0099
	.858571	.866867	.868005	.874144	.001138	.015573	.0731
	.872214	.882894	.882040	.891847	.000854	.019633	.0435
	.908861	.916908	.916548	.923799	.000360	.014938	.0241
	.921707	.926560	.928246	.931047	.001686	.009340	.1805
	.961580	.965756	.965895	.969399	.000139	.007819	.0178
	.997218	.997688	.997685	.998050	.000003	.000832	.0036
	.997843	.998089	.998119	.998294	.000030	.000451	.0665
.999640	.999729	.999739	.999795	.000010	.000155	.0645	
$G_{5,5}$	.260993	.268991	.270445	.278251	.001454	.017258	.0843
	.518669	.533199	.534743	.547908	.001544	.029239	.0528
	.647162	.665644	.664985	.683737	.000659	.036575	.0180
	.957197	.961919	.963355	.966559	.001436	.009362	.1534
	.982125	.984391	.984743	.986359	.000352	.004234	.0831
	.984506	.986472	.987299	.988320	.000827	.003814	.2168
	.999753	.999812	.999848	.999866	.000036	.000113	.3186
	.999759	.999813	.999851	.999863	.000038	.000104	.3654
$G_{5,6}$	.251207	.266163	.268609	.284164	.002446	.032957	.0742
	.327675	.343522	.341277	.359500	.002245	.031825	.0705
	.598724	.611945	.612894	.625923	.000949	.027199	.0349

Table 4 : Reliability estimates on three graphs using randomly generated probability values