Polynomial Algorithms for Estimating Network Reliability

Eitan Zemel
MEDS Department, School of Management, Northwestern University,
Evanston, Illinois 60201

We consider the problem of calculating the best possible bounds on the reliability of a system given limited information about the joint density function of its components. We show that a polynomial algorithm for this problem exists iff such an algorithm exists for a certain related problem of minimizing a linear objective function over a clutter. We give numerous examples of network as well as other problems for which the algorithm runs in polynomial time. We also use our construction to prove NP-hardness for others.

1. INTRODUCTION

In numerous cases of practical interest one wishes to calculate the reliability of a given system using an underlying distribution function which is only incompletely specified. Most frequently, the missing information about this function consists of the numeric values of some of the higher moments. For instance, one may have available the individual probabilities of failure of the various components but not those of the simultaneous failure of pairs of components, triples, quadruples, etc. Such instances of incomplete information may be due to two types of obstacles. In the first place, it may be difficult to obtain or calculate the probability of a joint failure for a given subset of components. Second, even when each one of these probabilities can be easily obtained, their number is exponential and thus it is practically impossible to completely specify them all. In view of these obstacles, it is not surprising that a significant fraction of the literature on reliability theory is based on the assumption that the individual components of the system are independent with respect to failure. Obviously, under such an assumption, both types of difficulties do not arise and the function is completely specified in terms of very few and easily obtained parameters. Needless to say, however, in many practical situations the assumption of independence is questionable at best. For instance, it is obvious that the various components and subsystems of a nuclear power plant are not independent of each other with respect to failure.

Suppose we are given a system of this type with an incompletely specified distribution function. Typically, there is a large choice of concrete distribution functions which are consistent with our partial specification. In general, each one of these functions yields a different numeric value for the reliability of the system. Obviously when
we consider all possible choices of this function, the associated values of the reliability trace a certain interval \([a, b] \subseteq [0, 1]\). This interval, then, is the shapely possible specification of the reliability that can be hoped for, given the available information. Our main interest in this paper is in the algorithmic aspects associated with the computation of this interval.

It is obvious that the usefulness of the interval \([a, b]\) as a measure of the actual reliability \(r\) is very crucially dependent on the size of this interval. We do not attempt to assess the size of this interval in this paper, this topic being currently under study by simulation. However, it is clear that the size of the interval can be reduced by tightening the specification of the distribution function. Although this topic too is not pursued in this article, the reader should have no difficulty observing how additional information can be incorporated into the framework developed here. Furthermore, as a by-product of the calculations yielding \(a\) or \(b\), the algorithm also identifies those parameters whose specification can lead to a substantial tightening of the bounds. These issues too are a subject of current study and are addressed in the concluding section.

The structure of this paper is as follows. In Sec. II we give some definitions and preliminary results including a linear programming formulation of the problem, due to Hailperin [9]. The section is concluded by an example which illustrates the concepts used. In Sec. III we discuss our algorithm, which is based on the work of Grötschel, Lovász, and Schrijver [8], Padberg and Rao, [14], Karp and Papadimitriou [12], and Kachian [10]. In Sec. IV we discuss the algorithm \(vis \, \, \, vis\) numerous special cases. Finally, we conclude the paper in Sec. V.

II. DEFINITIONS AND PRELIMINARIES

Consider a system \(G\) which contains a set of \(n\) components \(e_1, \ldots, e_n\). Let \(x_1, \ldots, x_n\) be 0-1 random variables which represent the on-off status of these components, and let \(X = (x_i), i = 1, \ldots, n\). We refer to \(X\) as the state of the system \(G\). Denote by \(F(X)\) the joint probability distribution over the possible states of the system.

For each particular state vector \(X\), the system as a whole may be either on or off. Let \(\Phi(\cdot)\) be a function whose domain is the set of possible state vectors and whose range is the set \([0, 1]\). We say that the system \(G\) is "on" under state \(X\) iff \(\Phi(X) = 1\). The function \(\Phi\) is often referred to as the structure function of \(G\). In most practical applications \(\Phi\) satisfies the following properties:

\[
\Phi(0, \ldots, 0) = 0, \quad \Phi(1, \ldots, 1) = 1, \quad Y \supseteq X \implies \Phi(Y) \geq \Phi(X); \quad (1)
\]

we restrict our attention in this paper to structure functions \(\Phi\) satisfying (1).

Given a structure function \(\Phi\) and a probability distribution function \(F\), we can define the reliability of the system

\[
r = \text{Prob}\{\Phi(X) = 1\}
\]

For most practical structure functions the calculations of \(r\) is extremely tedious even if \(F\) itself is simple (e.g., under the assumption that the random variables \(x_i\) are mutually independent). For example, let \(G\) correspond to an undirected network and let the
components be the edges. Let \( s \) and \( t \) be two distinct nodes of \( G \) and consider the structure function

\[
\Phi(X) = 1 \text{ iff } X \text{ contains a path between } s \text{ and } t.
\]

It is known that the problem of calculating \( r \) is NP-hard [21]. Various other results of this type are shown in [1], [16], [18], and [19].

In this article we analyze the problem of computing \( r \) when the function \( F \) is not completely specified. In particular, we assume that the only information available on this function is in the form of the individual bounds:

\[
a_i \leq p_i \equiv \text{Prob}[x_i = 1] \leq b_i, \quad i = 1, \ldots, n, \tag{2}
\]

for a given set of constants, \( 0 \leq a_i \leq b_i \leq 1, i = 1, \ldots, n \). It is apparent that the relations (2) do not, in general, completely specify the function \( F \). Consequently, the reliability \( r \) is not well defined. What we seek, then, is the best that can be hoped for under the circumstances, namely to calculate the best possible upper and lower bounds on \( r \) consistent with the relations (2). We denote these bounds by \( \beta \) and \( \alpha \), respectively. As shall be revealed shortly, the calculation of these bounds may in some cases be relatively easy and could be accomplished in polynomial time. In other cases, however, the task of calculating \( \alpha \) or \( \beta \) may turn out to be NP-hard.

It will be convenient to describe state vectors as subsets of \( N = \{1, \ldots, n\} \). For a state vector \( X \) let

\[
S_X = \{ j \in N : x_j = 1 \},
\]

and conversely, for a set \( S \subseteq N \) let

\[
X^S = (x_i^S) i \in N, \quad \text{with } x_i^S = 1 \text{ iff } i \in S.
\]

Let

\[
F = \{ S \subseteq N : \Phi(X^S) = 1 \};
\]

then \( r \) can be alternatively expressed as

\[
r = \text{Prob} \left( \bigcup_{S \in F} \left[ X = X^S \right] \right) \tag{3}
\]

We note that properties (1) of the structure function \( \Phi \) translate into the following properties of the family \( F \):

\[
\phi \notin F, \quad N \in F, \quad S \in F \implies S' \in F \quad \text{for every } S' \supseteq S. \tag{4}
\]

The problem of estimating expressions like (3) given conditions of type (2) is old, and was first discussed by Boole [2]. Hailperin [9] has shown that the best upper
bound for such an expression can be obtained by solving the linear program

\[
\beta = \max \sum_{S \in F} y_S 
\]

subject to

\[
\sum_{S : i \in S} y_S \leq b_i, \quad i \in N, \\
\sum_{S : i \in S} y_S \geq a_i, \quad i \in N, \\
\sum_{S \subseteq N} y_S \leq 1, \\
y_S \geq 0, \quad S \subseteq N.
\]

Similarly, the best lower bound on \( r, \alpha \), can be found by solving the minimization problem over the same set of constraints and using the same objective function. We note that the linear program in question involves an exponential number of variables. Thus any method which relies on explicitly writing down this set of constraints cannot yield a polynomial algorithm for this problem. Nevertheless, the problem contains enough structure to allow implicit handling of the set of variables. Such an approach gives rise to a polynomial algorithm for \( \alpha \) and \( \beta \) for various interesting functions \( \Phi \). In other cases, the technique yields an easy proof for the NP-hardness of the problem.

Our approach to solving the linear program (5) is via its dual. Let \( H = \{ S : s \notin F \} \) where \( \bar{S} \) represents the complement of \( S \) in \( N \). Denote by \( F^* \) and \( H^* \) the sets of minimal elements in \( F \) and \( H \). \( F^* \) and \( H^* \) are often referred to as the sets of paths and cuts of the system \( G \), respectively (see Examples 1–4 below). These sets are known to play a crucial role in reliability calculation (see, for instance, the recent article by Ball [1]). It is straightforward to demonstrate that \( H^* \) and \( F^* \) constitute a blocking pair of clutters [5]; i.e., each one is a collection of subsets none of which contains the other, and they can be defined from each other via the relation \( H^* = \{ S : x^S x_T^T \geq 1 \text{ for every } T \in F^* \} \), and \( S \) is minimal with respect to this property and conversely, for \( F^* \).

Using \( F^* \) and \( H^* \) we obtain the following:

**Theorem 1.** Let \( \Phi \) satisfy properties (4); then

\[
\beta = \min \sum_{i=1}^{n} u_i b_i + w
\]

subject to

\[
\sum_{i \in S} u_i + w \geq 1 \quad \text{for every } S \in F^*, \quad u, w \geq 0,
\]

and

\[
\alpha = 1 - \min \left( \sum_{i=1}^{n} u_i (1 - a_i) + w \right)
\]
subject to

\[ \sum_{i \in S} u_i + w \geq 1 \quad \text{for every } S \in H^*, \quad u, w \geq 0. \]

**Proof.** Take the linear programming dual of (5)

\[ \beta = \min \sum_{i=1}^{n} u_i b_i - \sum_{i=1}^{n} v_i a_i + w \]

subject to

\[ \sum_{i \in S} u_i - \sum_{i \in S} v_i + w \geq 1: S \in F, \]

\[ \sum_{i \in S} u_i - \sum_{i \in S} v_i + w \geq 0: S \not\in F, \quad u, v, w \geq 0. \]  \hspace{1cm} (8)

We first note that there exists an optimal solution to (8) where for each pair \(u_i, v_i\) at most one member is positive. For otherwise we can decrease both \(u_i\) and \(v_i\) by \(\min(u_i, v_i)\). This will leave the solution feasible without increasing the objective value (since \(b_i \geq a_i\)). Next we prove that there exists an optimal solution with \(v_i = 0, i = 1, \ldots, n\). Let \((u, v, w)\) be an optimal solution which does not satisfy this property but for which \(v_i u_i = 0, i = 1, \ldots, n\). Let \(N_1 \subseteq N\) be the set of indices for which \(v_i > 0\). Let \(\bar{v} = \sum_{i \in N_1} v_i\). Consider the inequality which corresponds to \(N_1\). Irrespective of whether \(N_1 \subseteq F\) or not, we can conclude that

\[ \sum_{i \in N_1} -v_i + w \geq 0, \]

i.e., \(\bar{v} \leq w\). Consider the point \((u, 0, w - \bar{v})\). Since \(a_i \leq b_i, i = 1, \ldots, n\), the objective function of this point is not worse than that of \((u, v, w)\). Next, we demonstrate that this point is feasible for (7).

Assume, in the negative, that the constraint associated with some set \(S\) is violated. Without loss of generality we can assume that \(S \supseteq N_1\) since otherwise we can add to \(S\) the missing elements of \(N_1\) without changing the left-hand side of the constraints but with possible increase in the right-hand side. But then, for \(S\) which contains \(N_1\), the left-hand side calculated with respect to \((u, v, w)\) is the same as the left-hand side calculated with respect to our new point \((u, 0, w - \bar{v})\). Thus we have contradicted the assumption that the constraint associated with \(S\) is violated. The proof is completed by noting that the constraints which correspond to sets \(S \not\in F\), and to sets \(S\) in \(F\) which are not minimal there, are redundant and that the expression for \(\alpha\) follows from that of \(\beta\) by considering the complementary events \(\Phi(X) = 0\) and \(x_i = 0, i = 1, \ldots, n\).

Q.E.D.

The pair of linear programs (6) and (7) contain \(n\) variables each and as many constraints as there are elements in \(F^*\) and \(H^*\), respectively. Obviously, it is not practical to solve these problems by standard linear programming techniques since the number
of elements in $F^*$ and $H^*$ may be very large. However, in the following section we show how these difficulties can be overcome. But first we consider a simple example.

**Example 1.** Consider the case of a parallel network as depicted in Figure 1 below and let $\Phi = 1$ if and only if there is a path connecting $s$ and $t$. Such a network is an extreme case of a system with a large number of "paths" and few "cuts." In fact $F^* = \{\{1\}, \{2\}, \ldots, \{n\}\}$ and $H^* = \{1, 2, 3, \ldots, n\}$. Thus the linear programs defining $\beta$ and $\alpha$, respectively, are given by

$$\beta = \min \sum u_i b_i + w, \quad u_i + w \geq 1, \quad i = 1, \ldots, n, \quad u_i, w \geq 0;$$

and

$$\alpha = 1 - \min \left( \sum_{i=1}^{n} u_i(1 - a_i) + w \right), \quad \sum_{i=1}^{n} u_i + w \geq 1, \quad u_i, w \geq 0.$$ 

Now let $a_i = b_i = p$ and assume $\Sigma_{i=1}^{n} b_i = np \ll 1$. The solutions for the two linear programs (6) and (7) can be found easily by inspection. For the first, we can take $u_i = 1, i = 1, \ldots, n$, yielding a $\beta$ value of $np$. For the second, the basic optimal solutions are of the form $u_i = 1$ for some $i \in \{1, \ldots, n\}, u_j = 0, j \neq i$, and the value of $\alpha$ is $p$. Thus $p \leq r \leq np$. The reader may note that if all edges are assumed independent, $r = 1 - (1 - p)^n$, which is in fact inside the interval $[p, np]$. It is interesting to note that the optimal solutions to (6) and (7) also reveal the functions $F$ which yield the upper and lower bounds. For the first we find that the basic variables in formulation (5) [tight constraints in (6)] correspond to the sets $\{\{i\}\}, i = 1, \ldots, 5$, i.e., to the set of paths of $G$. Thus, the maximal reliability of the system, is obtained when the events $x_i, i = 1, \ldots, 5$ are mutually disjoint (the correlation between each two being $-1$). On the other hand, the unique nonzero variable in the minimization version of problem (5) corresponds to the set $S = \{1, \ldots, n\}$. Thus the minimum reliability is obtained if the variables $x_i, i = 1, \ldots, S$ are completely identical, i.e., if the correlation between each two is $+1$. Of course, the situation is completely reversed the arcs in the system $G$ in series rather parallel.

![FIG. 1.](image-url)
III. THE ALGORITHM

Our algorithm for problem (6) and (7) relies heavily on the recent work of Grötschel, Lovász, and Schrijver [8] (see also Padberg and Rao [14] and Karp and Papadimitriou [12]), which is in turn a generalization of the polynomial algorithm for linear programming put forward by Kachian [10].

We recall the essentials of Kachian’s method. At the kth iteration we have an ellipsoid $E_k$ which is known to contain an optimal solution to our problem. Denote the center of this ellipsoid by $x_k$. Assume that $x_k$ is not feasible for our problem. Then any constraint which is violated by $x_k$ can be used to split $E_k$ into two halves such that the optimal solution (which is known to be within $E_k$) lies, in fact, in one of these halves. Alternatively, if $x_k$ is feasible for our problem, we can achieve a similar halving of $E_k$ using the objective row. The iteration is concluded by enclosing the half-ellipsoid we wish to keep by another ellipsoid $E_{k+1}$ whose volume is smaller than that of $E_k$. One can show that in a polynomial number of steps we obtain an ellipsoid whose center is “close” enough to the optimal solution. The procedure is terminated by a process of “rounding” the entries of this center, using, say, continued fractions.

Kachian’s algorithm, in the form stated, relies on the possibility of checking, in polynomial time, whether or not $x_k$ is feasible. In the negative case we also must be able to specify a violated constraint, i.e., to separate $x_k$ from the feasible set. This relation between the optimization problem and the “separation problem,” is the central theme in Grötschel, Lovász, and Schrijver’s paper. The following theorem, which suffices for the purposes of our discussion here, is a consequence of the more general theorems of [8]. Similar theorems appear in [12] and [14].

For a rational vector $x \in \mathbb{R}^n$ let $T(x)$ be the smallest positive integer such that every component of $x$ can be expressed as the ratio of two integers bounded in magnitude by $T(x)$. Consider the two problems

\begin{enumerate}
    \item[(P1)] $\max cx$ subject to $Ax \leq b$;
    \item[(P2)] Given $y \in \mathbb{R}^n$, either show $Ay \leq b$ or find a violated constraint $a_i y > b_i$.
\end{enumerate}

Assume that (i) the entries of $c$, $A$, and $b$ are integers whose magnitude is bounded by $T$; (ii) the feasible set of (P1) is full dimensional; (iii) there exists a point $a_0$ in the interior of the feasible set of (P1) such that each entry of $a_0$ can be expressed as the ratio of two integers whose magnitude is bounded by $T$; and (iv) $y$ satisfies the same numeric properties as $a_0$.

**Theorem 2.** Assume that (i)–(iv) are satisfied. Then there exists an algorithm for (P1) which is polynomial in $n$ and $\log T$ if and only if such an algorithm exists for (P2).

Applying Theorem 2 to the linear programs (6) and (7) of Theorem 1, we get the main result of this paper:

**Theorem 3.** Let $\Phi$ satisfy properties (1).

(i) There exists an algorithm for calculating $\beta$ which is polynomial in $[n, \log T(b)]$ iff there exists an algorithm for the problem

$$
\min_{S \subseteq F^*} cX^S
$$
which is polynomial in \([n, \log T(c)]\) for every non-negative vector \(c\).

(ii) Same as (a) with \(\alpha\) replacing \(\beta\), \(H^*\) replacing \(F^*\), and \(\log T(1 - \alpha)\) replacing \(\log T(b)\).

**Proof.** (i) We first consider the polyhedron of Theorem 2. Obviously this polyhedron is full dimensional and contains an interior point as required (e.g., \(u_i = 2, i = 1, \ldots, n, w = 2\)). Furthermore, its constraints matrix is made out of zeros and ones only. Thus it satisfies the stipulations of Theorem 2 for any \(T \geq T(b)\). It follows that there exists an algorithm for \(\beta\) which is polynomial in \([n, \log T]\) if and only if for every vector \(u\) with \(T(u) \leq T\) and for every rational \(w\) which can be expressed as the ratio of two integers bounded by \(T\), we can decide in polynomial time whether or not \((u, w)\) satisfies all the constraints of this polyhedron.

Assume that a polynomial algorithm for \(\min_c X^S : S \in F^*\) does exist. Then running this algorithm with \(u\) replacing \(c\) we can obviously decide whether \((u, w)\) satisfies all the constraints and in the negative case find a violated constraint (namely the one which yields the minimum in the minimization problem). Thus we have demonstrated the existence of an algorithm for \(\beta\) which is polynomial in \([n, \log T(b)]\). Conversely, suppose that such an algorithm for \(\beta\) exists. Then we know by Theorem 2 that a polynomial algorithm exists for deciding, for each \(w\) and cost vector \(c\), whether or not \(\min_c X^S : S \in F^*\) is greater than or equal to \(w\). From the stipulations on the vector \(c\) it follows that the optimal value of this problem, \(w^*\), is a rational number satisfying

\[ w^* \in \left[0, \sum_{i=1}^{n} c_i\right] \]

and furthermore, \(w^* = p/q\), with \(p \geq 0, q > 0, \max\{p, q\} \leq T(c)^n\). Thus \(w^*\) can be found in \(O(\log T(c)^n) = O(\alpha \log T(c))\) applications of this algorithm [14], [16], i.e., in polynomial time as asserted.

Q.E.D.

Before examining specific applications of Theorem 3, we consider a corollary of this theorem. Let \(\Phi_1, \ldots, \Phi_k\) be a given set of structure functions defined on the same system \(G\). For \(i = 1, \ldots, k\) let \(\alpha_i\) be the lower bound on \(r\) defined with respect to \(\Phi_i\). Similarly define \(\beta_i, F_i, F^*_i, H^*_i, i = 1, \ldots, k\).

**Corollary 3.1.** (i) Assume there exist polynomial \([n, \log T(b)]\) algorithms for calculating \(\beta_i, i = 1, \ldots, k\). Then there exists a polynomial \([n, k, \log T(b)]\) algorithm for calculating \(\beta\) for the structure function \(\Phi\) satisfying

\[ \Phi(X) = \max_{i=1, \ldots, k} \Phi_i(X) \]

(ii) Assume there exist polynomial algorithms for calculating \(\alpha_i, i = 1, \ldots, k\). Then there exists a polynomial algorithm for calculating \(\alpha\) for the function

\[ \Phi(X) = \min_{i=1, \ldots, k} \Phi_i(X). \]
Proof. We note that $\Phi(X) = \max_{i=1,...,k} \Phi_i(X)$ implies $F = \bigcup_{i=1,...,k} F_i$ and $F^* \subseteq \bigcup_{i=1,...,k} F^*_i$. Thus we can minimize $c X$ over $F^*$ by minimizing over each of the $F^*_i$ and then taking the grand minimum. Similarly, $\Phi(x) = \min_{i=1,...,k} \Phi_i(x)$ implies that $H^* \subseteq \bigcup_{i=1,...,k} H^*_i$.

Q.E.D.

IV. EXAMPLES

We now consider some specific structure functions $\Phi$ and examine them vis à vis the stipulations of Theorem 3. In cases (a)-(h) below, the random variables $x_i$, $i = 1, \ldots, n$ correspond to the on-off status of the edges of the graph in question. The results utilized in cases (a)-(d) and (f)-(h) are well known and the algorithms available for these problems are all extremely fast and practical. For background and details the reader is referred to Lawler's book [13].

(a) Let $G$ be a direct graph, $s$ and $t$ two specific nodes. Assume that $G$ is considered "on" if the graph induced by state vector $X$ is such that there exists at least one directed path from $s$ to $t$. We note that

$$F^* = \{\text{set of all s-t paths in } G\}$$

$$H^* = \{\text{set of all s-t cuts in } G\}$$

since a polynomial algorithm exists for minimizing a non-negative linear function both on $F^*$ and on $H^*$, we can conclude that there exists a polynomial algorithm for $\alpha$ and for $\beta$ for this case.

(b) Consider a graph as in (a) but with undirected edges. Again we let $\Phi(X) = 1$ if the graph induced by $X$ contains an (undirected) path between two specific nodes $s$ and $t$. Note that $F^*$ and $H^*$ in this case are the undirected versions of the sets $F^*$ and $H^*$ discussed in (a). It is well known that the optimization problems over both $H^*$ and $F^*$ can be solved in polynomial time. Thus we have a polynomial algorithm for $\alpha$ and $\beta$ in this case too.

(c) Let $G$ be a directed network, $s$ a specific node. Let $\Phi(X) = 1$ if the graph induced by $X$ contains a directed path from $s$ to every other node of $G$. We note that $F^*$ is the set of arborescences of $G$, rooted at $s$. Thus there exists a polynomial algorithm for calculating $\beta$ [5]. Also, by applying Corollary 3.1 to case (a) considered earlier, we know that there exists a polynomial algorithm for calculating $\alpha$ for this case.

(d) Let $G$ be an undirected network, and assume that $G$ is "on" if in the graph induced by $X$ there exists a path connecting every pair of nodes of $G$. There exists a polynomial algorithm for calculating $\beta$ since $F^*$ in this case is the set of spanning trees of $G$. A polynomial algorithm for $\alpha$ can be obtained by applying Corollary 3.1 to the problem discussed in (b).

(e) To contrast the four cases considered earlier, consider again an undirected network $G$ and assume that $G$ is "on" if the graph induced by $X$ contains a Hamiltonian tour of $G$. It follows that $F^*$ is the set of Hamiltonian tours in $G$ and $H^*$ is the blocker clutter of this set. Note that the problem of deciding whether $F^*$ is not empty for a given graph is NP-hard. Thus the optimization problem over either $F^*$ or $H^*$ is NP-hard and so are, then, the problems of calculating $\alpha$ and $\beta$. 
(f) (network reliability [7]) To note the distinction between calculating $\alpha$ and $\beta$ we consider an undirected network $G$. Let $v_1, \ldots, v_k$ be a specific subset of the nodes of $G$. Consider the structure function

$$\Phi(X) = 1 \text{ iff the graph induced by } X \text{ contains a path}$$

between every pair of nodes $v_i, v_j$ $1 \leq i < j \leq k$.

Then $F^*$ is the set of Steiner trees defined on $G$ with respect to node set $v_1, \ldots, v_k$. It is well known [17] that minimization of a linear function over this clutter is NP-complete. On the other hand, the problem of calculating $\alpha$ can be accomplished in polynomial time by applying Corollary 3.1 to the problem discussed in (b).

(g) We note a certain converse to the problem discussed in (f). Consider again an undirected graph $G$ with a specified set of nodes $v_1, \ldots, v_k$. Let $\Phi(X) = 1$ if the graph induced by $X$ contains a path between at least one pair of nodes $v_i, v_j$, $1 \leq i < j \leq k$. Then $F^*$ can be described as a union of sets of paths in $G$ (between every pair of nodes $v_i, v_j$, $1 \leq i < j \leq k$) and thus one can calculate $\beta$ in polynomial time. On the other hand $H^*$ can be described as the intersection of sets of cuts on $G$. The status of the optimization problem over $H^*$ is unknown.

(h) Let $G$ be an undirected graph. We say that $G$ is on if the graph induced by $X$ contains a perfect matching; i.e., we can use the “on” edges of $G$ to pair the nodes of $G$ in such a way that each node is paired to exactly one other node. Obviously, the set $F^*$ correspond to the set of perfect matchings of $G$, and minimization over this set can be achieved in polynomial time [3]. The status of the minimization problem over the blocker set of $F^*$ is unknown. However, if $G$ is bipartite, $H^*$ is known [6], and a polynomial algorithm exists for minimizing over this set. Thus, for such graphs, we can calculate $\alpha$ in polynomial time. These assertions are valid for graphs $G$ which do not contain a perfect matching if we replace “perfect” by “maximum-cardinality” matching in the definition of $F$.

(i) (network survivability [7]) Let $G = (V, E)$ be an undirected graph such that both its edges and its nodes are subject to failure. Let $V \cup E = K = \{k_1, \ldots, k_n\}$. Then the components of the state vector $X$ correspond to the elements of $K$. We say that the network survives [i.e., $\Phi(X) = 1$] if for every edge $i \in E$, $e = (u, v)$, at least one of the triple $e, u, v$ is “on.”

For each edge $e = (v, u) \in E$, let $i(e), j(e), k(e)$ be indices such that $k_i(e) = u$, $k_j(e) = v$, $k_h(e) = e$ and let $X^e$ be the vector $x^e_i = 1$ if $i = i(e), j(e), h(e)$.

It is easy to verify that

$$H^* = \{x^e\}_{e \in E}.$$  

Minimizing over $H^*$ is trivial, and so $\alpha$ can be calculated polynomially. On the other hand, the set $F^*$ corresponds to the set of minimal covers (of edges) by edges and nodes. Minimizing a linear function over $F^*$ is NP-hard since it is a more general problem than the node covering (of edges) problem [11]. Thus calculating $\beta$ here is NP-hard.

The last two examples considered refer to a general system, not necessarily associated with a network.
polynomial algorithms 449

(j) Consider a system with components \( e_1, \ldots, e_n \) and let \( K \) be a matroid over \( N = \{ i, \ldots, n \} \). Assume that \( \Phi(x) = 1 \) if the vector \( x \) contains a basis of \( K \). Then

\[
F^* = \{ \text{set of bases of } K \},
\]

\[
H^* = \{ \text{set of co-circuits of } K \}.
\]

Minimization over \( F^* \) can be accomplished in polynomial time by using the greedy algorithm [4]. Thus \( \beta \) can be calculated in polynomial time for this problem. The status of the minimization problem over \( H^* \) is unknown.

(k) Let \( e_1, \ldots, e_n \) be a set of elements each associated with a value \( c_j, j = 1, \ldots, n \). Consider a system which is "on" if the value of the elements which are "on" is at least \( b \). Then

\[
F = \left( \{ S : \sum_{j \in S} c_j \geq b \} \right),
\]

\[
H = \left( \{ S : \sum_{j \in S} c_j \geq \sum_{j \in N} c_j - b \} \right),
\]

with \( F^*, H^* \) being the minimal elements in \( F \) and \( H \). Both minimization problems in this case are NP-complete. However, if \( c_j = 1, j = 1, \ldots, n \), and \( 1 \leq b \leq n \), we get a system which is on if at least \( b \) components out of \( n \) are on. Thus

\[
F^* = \{ S : |S| = b \},
\]

\[
H^* = \{ S : |S| = n - b \}.
\]

Both \( F^* \) and \( H^* \) are known to correspond to the set of bases of especially simple matroids. Thus, by the reasoning of the previous case, \( \alpha \) and \( \beta \) can be solved in polynomial time. An obvious application of this case is the one of estimating the probability of a favorable outcome in a voting system with two possible outcomes where (i) all voters count equally, (ii) there is a minimal number of votes \( b \) required for acceptance, and (iii) we only know bounds on the individual probability of voting for and against this outcome.

(1) Consider a graph with a node set \( V \). Consider two sets of subsets of \( V \), \( \Sigma \), and \( \Delta \). \( \Sigma \) can be thought of as a set of centers located in \( G \). \( \Delta \), on the other hand, corresponds to a set of points where demand exists for the services offered by these service centers. Let \( \Sigma = \{ V_l \}_{l \in N}, \Delta = \{ V_j \}_{j \in K} \) for \( N \) and \( K \) two index sets, not necessarily distinct. Assume that the supply points in \( \Sigma \) are not reliable and let \( X = (x_l)_{l \in N} \) be the state vector in their on-off status. Assume that the system is "on" if every demand point in \( y_j \in \Delta \) is at a distance of no more than \( r_j \) units from a service center \( v_l \in \Sigma \) which is operating, i.e., such that \( x_l = 1 \). Let \( F \) be the set of subsets of \( N \) which satisfy the required property. To show that \( \alpha \) can be found in polynomial time we note that \( H \) can be written as

\[
H = \bigcup_{j \in K} H(j),
\]
where $H(j)$ is the set of subsets of $N$ such that $N \backslash H(j)$ contains only elements of $N$ which are at a distance of more than $r$ units away from $v_j$. Thus $H^*(j)$ contains a unique subset, namely the one which contains all the elements of $N$ which are at a distance of $r_j$ or less from $v_j$. Thus minimization over $H^*$ is trivial. The problem of calculating $\beta$ is a little more involved. It is shown in [20] that optimizing a linear function over $F$ can be done in polynomial time provided the graph in question is a tree. On the other hand, the problem is NP-hard for the general case. Thus $\beta$ can be found in polynomial time on tree networks but its calculation on a general graph is NP-hard.

V. CONCLUDING REMARKS

This paper has presented an alternative framework for assessing network reliability. It is natural to compare the merits and disadvantages of this framework vis-à-vis those of the traditional approach based on the assumption of independent components, or at least on a distribution function which is completely specified.

From a theoretical point of view, the advantages of the new method are obvious. We first note that the assumptions underlying this model are few and reasonable and that the number of parameters needed to specify a given system is essentially minimal. This is in sharp contrast with a model which is based on a complete specification of the distribution function in which case either the assumptions are rather extreme (independence) or else the amount of data required is enormous. Also, from the point of view of computational complexity, the algorithm presented here is polynomially bounded for many interesting structure functions while the computation of the reliability under the assumption of independence is NP-hard for virtually all structure functions which occur in practice. Of course, the obvious disadvantage of our approach is that it yields an interval $[\alpha, \beta]$ rather than a unique numeric value.

The issues raised in the previous paragraph can be assessed from a practical viewpoint as well. Below we elaborate on some of these issues in detail.

The polynomial time complexity achieved in this paper is based on two factors. The first is a polynomial time algorithm for the “separation” or “sub” problem (such as the shortest-path problem, the minimal-spanning-tree problem, etc.). Second, we have the polynomial bound of [8] on the complexity of Kachian’s procedures which serves as a “master” algorithm. We note that while most of the algorithms for the subproblems mentioned in the various examples are extremely efficient and practical, the practicality of Kachian’s method is as yet to be proven. However, we can easily replace the master algorithm proposed here by the simplex algorithm, using the column generation technique on formulation (5) directly. The reader should have no difficulty realizing that the column generation phase [in which we look for a column (a subset) to enter the basis] is exactly identical to the subproblems of Theorem 3 one has to handle in our model. Hence, when one switches to a simplex-based column generation technique, the efficiency of the routines for the subproblems is not affected. It is not known whether a polynomial bound can be obtained for the master problem using a simplex-type algorithm, but the efficiency and reliability of such algorithms in practice are well known. We are currently implementing a software package which is based on these ideas. As a part of our investigation we also wish to shed some more light on another problem which is of major practical importance,
namely the size of the interval \([a, b]\). This question is intimately related to the question of how to use additional information about the function \(F\), in case such information becomes available. The answer to this question follows directly from the work of Hailperin [9], and is in fact quite straightforward. Each additional parameter specifying the function \(F\) can be accommodated by an additional constraint in the formulation (5), and thus to an additional column in (6) or (7). For instance, if we wish to impose lower and upper bounds (say \(a_{ij}\) and \(b_{ij}\)) on the probability that components \(i\) and \(j\) will fail simultaneously, all we have to do is add the constraints

\[ a_{ij} \leq \sum_{S: i \in S} y_S \leq b_{ij} \]

The effect of such constraints is to add a quadratic term to the objective function of the subproblem. Thus, given such parameters, our subproblems for the various cases correspond to the quadratic shortest-path problem, quadratic minimum-spanning-tree problem, etc. Obviously, as additional parameters of the function \(F\) are specified, the interval \([a, b]\) will shrink in size, eventually collapsing to a single point. A practical approach to the problem is probably to start with a set of parameters which can be easily obtained, and then add additional parameters as required until the size of \([a, b]\) is acceptable. Standard LP sensitivity analysis considerations can serve as an excellent guide as to which additional parameters should be most useful at a given iteration of this process.

I wish to thank both referees for many thoughtful remarks and suggestions. This work was supported in part by NSF Grant ENG-7902806 and by the J. L. Kellogg Center for Advanced Studies in Managerial Economics.

References

[16] J. S. Proven and M. O. Ball, The complexity of counting cuts of computing the probability that a graph is connected. University of Maryland College of Business Administration, W.P. MS S 81-0002.