TECHNICAL RESEARCH REPORT

SYSTEMS RESEARCH CENTER

Supported by the National Science Foundation Engineering Research Center Program (NSFD CD 8803012), Industry and the University

Network Reliability

by M.O. Ball, C.J. Colbourn, and J.S. Provan

TR 92-74

Network Reliability

Michael O. Ball
Univeristy of Maryland

Charles J. Colbourn
University of Waterloo

J. Scott Provan
University of North Carolina

June 23, 1992

Abstract

This paper provides a detailed review of the state of the art in the field of network reliability analysis. The primary model treated is a stochastic network in which arcs fail randomly and independently with known failure probabilities. The inputs to the basic network reliability analysis problem consist of the network and a failure probability for each arc in the network. The output is some measure of the reliability of the network. The reliability measures treated most extensively in this paper are: the two terminal measure, the probability that there exists a path between two specified nodes; the all-terminal measure, the probability that the network is connected and the k-terminal measure, the probability that a specified node subset, K, is connected. In all cases the results concerning each problem's computational complexity, exact algorithms, analytic bounds and Monte Carlo methods are covered. The paper also treats more complex reliability measures including performability measures and stochastic shortest path, max flow and PERT problems. A discussion is provided on applications and using the techniques covered in practice.

Network Reliability

Michael O. Ball
College of Business and Management
and
Systems Research Center
University of Maryland
College Park, Maryland

Charles J. Colbourn
Department of Combinatorics and Optimization
University of Waterloo
Waterloo, Ontario, Canada

J. Scott Provan
Department of Operations Research
University of North Carolina
Chapel Hill, North Carolina

June 22, 1992

Contents

1 Motivation

Network reliability encompasses a range of issues related to the design and analysis of networks which are subject to the random failure of their components. Relatively simple, and yet quite general, network models can represent a variety of applied problem environments. Network classes for which the models we cover are particularly appropriate include data communications networks, voice communications networks, transportation networks, computer architectures, electrical power networks and command and control systems.

The advent of the digital computer led to significant reliability modeling efforts [296]. Early computer memories were made up of large numbers of individual components such as relays or vacuum tubes. Computer systems which failed whenever a single component failed were extremely unreliable, since the probability of at least one component out of thousands failing is quite high, even if the component failure probability is low.

Much initial work in high reliability systems concentrated on systems whose failure could cause massive damage or loss of human life. Examples include aircraft and spacecraft systems, nuclear reactor control systems and defense command and control systems. More recently, it has been recognized that very high reliability systems make economic sense in a wide range of industries. Examples include telecommunications networks, banking systems, credit verification systems and order entry systems.

The ultimate objective of research in the area of network reliability is to give design engineers procedures to enhance their ability to design networks for which reliability is an important consideration. Ideally, one would like to generate network design models and algorithms which take as input the characteristics of network components as well as network design criteria, and produce as output an "optimal" network design. Since explicit expressions for the reliability of a network are very complex, typical design models use *surrogates* in place of explicit reliability expressions. For example, in Chapter 15 of this volume, Grötschel and Monma address network design problems where the surrogate used in network connectivity. In this chapter we treat the network reliability analysis problem, which is the problem of evaluating a measure of the reliability of a network. Analysis models are typically used in conjunction with network design procedures. For example, once a network design is produced using the techniques described in Chapter 15, models we describe might be used to determine the value of the network's reliability. If the reliability value is not satisfactory then the design model might be resolved with different design criteria. Alternatively, a designer might manually adjust the design. After a modified design is generated by one of the aforementioned techniques, the value of the network's reliability would be recomputed to determine if it is satisfactory. This process might iterate several times.

1.1 Application Areas

We now describe some specific application settings.

1.1.1 Backbone Level of Packet Switched Networks

Packet switched networks were first developed in the 1960's to allow sharing of high speed communications circuits among many data communications users [157, 159]. Since the traffic associated with individual users tended to be bursty in nature, traffic on individual circuits could be dynamically allocated over time to a variety of users. ARPANET was the first major packet switched network. Much of the research on network reliability in the early 1970s and beyond was motivated by ARPANET. Figure 1 depicts the 1979 version of ARPANET. Most of the reliability measures used for ARPANET are "connectivity" measures. That is, they define the network as operating as long as the network is connected or, in the case of specific user communities, as long as a specified subset of nodes is connected. Such measures are justified since ARPANET employed dynamic routing so that traffic could be rerouted around failed links as long as the network remained connected. However, even though traffic could be rerouted, congestion could occur and delays could increase due to the decrease in overall network capacity.

When one compares ARPANET with the backbone networks of commercial packet switched networks in use in the 1980s, such as Telenet and Tymnet, it is clear that these networks are much denser than ARPANET. As a result the probability of network disconnection is much lower. However, the increased link density is primarily motivated by larger traffic loads. The implication is that capacity and congestion issues must be taken more explicitly into account in defining reliability measures. To address this concern, some recent research has involved the definition and calculation of so-called performability measures (see for example [265, 352, 432]). Rather than defining the network as operating as long as it is connected, performability measures define the network as operating as long as its performance, possibly measured in terms of average delay, satisfies certain criteria.

1.1.2 Backbone Level of Circuit Switched Networks

By far the largest telecommunications networks in existence today are the circuit switched networks that make up the world's public telephone systems. In circuit switched networks, a communications channel is dedicated to a pair of users for the length of their call. As overall network capacity is reduced due to component failures the number of communications channels that the network can support is reduced. Thus, users are adversely affected in that it becomes more likely that when a call is attempted no circuit is available. This phenomenon is known as call blocking. This is to be contrasted with packet switched networks where the affect of failures is increased transmission delay. Of course, in either case, if the network becomes disconnected then it becomes impossible for certain pairs of users to communicate. Some of the earliest work in network reliability involved modeling of circuit switched networks [259] where network links are defined to be failed if they are blocked. Connectivity based measures were then used in conjunction with this failure definition. More recently network performability measures have been defined [354]. In this case network performance is defined in terms of blocking rather than delay.

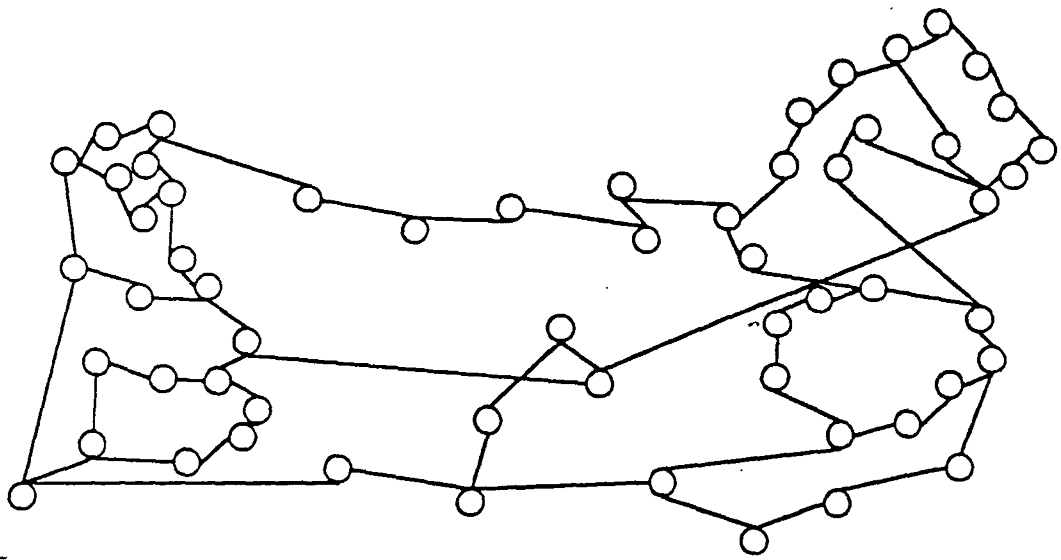

Figure 1: The 1979 ArpaNet

1.1.3 Interconnection Networks

A special case of circuit-switched networks arises in the design of interconnection networks for connecting parallel processors and memories in parallel computer architectures. Connectivity-based models are used both for failures due to congestion and to component wearout. Lee's pioneering work in telephone switching [259] anticipated the extensive use of connectivity-based measures for general interconnection networks [13, 210]. The measures have been particularly important in designing redundancy into interconnection networks [52, 53, 58, 237, 410]; the surrogate for overall system performance here is the average connectivity of an input to an output [95].

1.1.4 Local Voice Fiber Networks

A recently developed technology that is transforming the world's telecommunications networks is fiber optics (see [151] for example). Fiber optic communications channels transmit communications signals via light waves traveling over glass fibers. The principal advantage of this communications medium over traditional cables is a significant increase in transmission capacity. In addition there are certain performance advantages in terms of signal quality, particularly relative to terrestrial microwave radio systems. Because of these very significant advantages most public telephone systems are rapidly replacing their existing transmission networks with networks based on fiber optics. However, it has quickly become apparent that there are major reliability concerns that must be addressed. In particular, due to the extremely high capacity of fiber optic circuits, the resultant fiber optic networks tend to be much sparser than traditional networks. The net effect is that previously reliability could be ignored in designing large scale networks since the networks tended to be very dense and, consequently, naturally had acceptable levels of reliability. Now, if reliability is not explicitly considered in network design, networks can result for which single link failures can cause major disruptions. It is this phenomenon that has motivated much of the work described in Chapter 15.

Fiber optic circuits have redundant channels and rerouting capability built in. In addition, as has been mentioned, they are very sparse. As a result it is felt that connectivity based measures are appropriate for quantifying their reliability.

1.1.5 Fault Tolerant Computer and Switch Architectures

Fault tolerant computer systems are computers that can continue operating in the presence of the failure of one or more of their components. Such computers were used in the 1970s as the basis for telecommunications switching equipment. They are widely used today in a variety of applications. More recently parallel computer architectures have been developed. Such computers have multiple components of the same type for the purpose of increasing the overall throughput of the computer system. However, parallel architectures also naturally have superior reliability characteristics. Typically, these fault tolerant and parallel computer systems are modeled as networks for the purpose of reliability analysis. Whereas much of the work in network reliability analysis motivated by telecommunications networks has concentrated on algorithms for analyzing general network topologies, most network reliability work motivated by computer architectures has concentrated on designing and analyzing the highly structured networks associated with particular computer architectures. Typically connectivity based measures are used. However, especially in the case of parallel computers containing a large number of processors, reliability measures that take into account capacity considerations are of interest.

1.1.6 Other Applications

The richness of network models have led to their use in modeling several other reliability applications. In [102] network reliability model is used to model random spread of fire. In this context, once a fire has established itself in a room or building there is a possibility that it spreads through a barrier (wall) to an adjacent room or building. A network model is employed in which arc failure probability is interpreted as the probability that the fire spreads from a compartment through a wall to an adjacent compartment.

Sanso and Soumis [352] discuss the application of network reliability models to several application settings. A major theme of this paper is to stress the importance of routing in all of the application settings. In particular, in all cases analyzed, the network supports a diverse set of users and each user's traffic follows one or more routes through the network. The implication is that reliability can only be accurately evaluated if routing considerations are incorporated into the reliability measure. To accomplish this, it is necessary to consider performability measures. One of the more interesting applications areas discussed is urban transportation networks. In this context, incidents, such as highway accidents, cause the failure of network nodes and arcs. Although it is rare that urban transportation networks become disconnected, it is quite common for node and link failures to cause major congestion.

Finally, many of the reliability tools developed for connectivity-based measures of network performance generalize to quite different reliability problems in scheduling and assignment problems; see [97, 103, 200].

1.1.7 Causes of Failures

In most classical reliability analysis, failure mechanisms and the causes of failure are relatively well understood. For example, in electronic systems long term wear would result from continual exposure to heat. Such wear randomly causes failure over the range of exposed components. Reliability analysis typically involves the study of these random processes and characterization of associated failure distributions. Although some failure mechanisms associated with network reliability applications have these characteristics many of the most important do not. For example, many well-publicized failures associated with fiber optic networks have been caused by natural disasters such as fires or human error such as the severing of a communications line by a back-hoe operator. As a result it is difficult to model failure mechanisms in order to come up with failure rates. Typically, component failure rates are estimated based on historical data.

1.2 Basic Definitions

Due both to the inability to model failure mechanisms and the inherent difficulty of computing network reliability, time independent, discrete probability models are typically employed in network reliability analysis. In the most commonly studied model to which we devote most of our attention, network components (nodes and arcs) can take on one of two states: operative or failed. The state of a component is a random event that is independent of the states of other components. The reliability analysis problem is: given the probabilities that each component is operative, compute a measure of network reliability. We treat some generalizations of this model. In particular, we look at models in which components can take on one of several state values or models in which a quantity is associated with the operative state. The state values typically either correspond to distances or capacities. The simple two state model is sufficient for the consideration of connectivity measures, but when more complex measures are considered, such as performability measures, more complex component states must be considered.

In the two state model, the component's probability of operation or, simply, reliability, could have one of several possible interpretations. The most common interpretations are

1. the component's availability
2. the component's reliability.

Generally, throughout this chapter, we use the term reliability to mean the probability that a component or system operates. Here we discuss a more specific definition. Availability is used in the context of repairable systems. In these settings, components alternate between being in the operative state and being failed and under repair. The component's *availability* is defined as the probability that at a random point in time the component is operating. The component's availability is typically estimated by estimating both the mean time to failure and the mean time to repair. An estimate of availability is:

$$\frac{\text{mean time to failure}}{\text{mean time to failure} + \text{mean time to repair}}.$$

The definition of component reliability does not involve considerations of repair. Rather, a length of time t is specified and the *reliability* of a component is defined to be the probability that the component does not fail within time t. Other interpretations of a component's probability of operation are possible. For example in [259] the probability that a circuit is not blocked is used as the probability that the corresponding arc operates. Of course, the interpretation of the component level reliabilities in turn determine the appropriate interpretation of the network reliability measures calculated. In the remainder of this paper we simply refer to the probability of operation or reliability and are not specific about the interpretation.

Our starting point is a network $G = (V, E)$, where V is a set of nodes or vertices and E is a set of undirected edges or a set of directed arcs. When studying connectivity models for each $e \in E$ we define p_e as $\Pr[e \text{ operates}]$, the reliability of e. When studying simple flow (shortest path) models, we also associate a capacity c_e (distance d_e) with each $e \in E$. We interpret p_e as the probability the e

operates and has capacity c_e (distance d_e) and $1-p_e$ as the probability that e fails and has capacity 0 (distance equal to infinity). When studying multistate flow (shortest path) models we associate a capacity distribution $\{c_{e,i}, p_{e,i}\}$ (distance distribution $\{d_{e,i}, p_{e,i}\}$) each $e \in E$. We interpret $p_{e,i}$ as the probability that e takes has capacity $c_{e,i}$ (distance $d_{e,i}$).

In studying network reliability it is sometimes more convenient to view a more general context for reliability analysis, coherent binary systems. A *stochastic binary system* (SBS) represents a system that fails randomly as a function of the random failure of its components. Each component in its component set, T, can take on either of two states: operative or failed. The structure of the system is represented by a function $\psi(S)$ defined for each $S \subseteq T$ by

$$\psi(S) = \begin{cases} 1 \text{ if when } S \text{ operates and } T-S \text{ fails, the system operates} \\ 0 \text{ if when } S \text{ operates and } T-S \text{ fails, the system fails.} \end{cases}$$

An SBS is *coherent* if $\psi(T) = 1$, $\psi(\emptyset) = 0$ and $\psi(S') \geq \psi(S)$ for any $S' \supset S$. The third property implies that the failure of any component can only have a detrimental effect on the operation of the system. The computational problem of interest is to compute:

$$\text{Rel(SBS,p)} = \Pr[\psi(S) = 1 \text{ where } S \text{ is the set of operative components}],$$

given some representation of $\psi()$. At times we consider reliability problems where $p_e = p$ for all e in which case we replace p by p in the above notation. For any stochastic coherent binary system (SCBS), define a *pathset* as a set of component whose operation implies system operation, and a *minpath* as a minimal pathset; similarly, define a cutset to be a set of components whose failure implies system failure, and a *mincut* to be a minimal cutset.

1.3 Network Reliability Measures

Network reliability measures that we study are either the probability of certain random events or the expected value of certain random variables that depend on the structure of the network, the distances or capacities associated with the members of E and their associated occurrence probabilities. The majority of the research in network reliability as well as the majority of this paper are devoted to the *k-terminal measure*. A set of nodes K and a node $s \in K (k = |K|)$ are given. Given a network G and arc reliabilities p, the k-terminal reliability measure is defined as

$$Rel(G, s, K, \text{p}) = \Pr[\text{there exist operating paths from } s \text{ to each node in } K].$$

Two important special cases of the measure are the *two terminal measure* for which $|K| = 2$ and the *all terminal measure* for which $K = V$. The two terminal and all terminal measures are denoted by $Rel_2(G, s, t, \text{p})$ and $Rel_A(G, s, \text{p})$ respectively. We call the node s the source node and the nodes in $K \setminus \{s\}$ the terminals. Other connectivity measures have been analyzed (see for example [26, 89]). The details are omitted here, not because they are unimportant, but because their coverage would not provide substantial additional insight.

Performability measures evaluate a network's reliability relative to some performance criterion. Several performance criteria have been considered. For example, for packet switched networks a commonly used criterion is average message or packet delay. Such criteria can be viewed as random variables in that they depend on the set of operative arcs or alternatively on each arc's capacity or distance which are random variables. If Φ is the criterion random variable then two classes of performability measures are commonly considered:

- $\Pr[\Phi \geq \alpha]$ or $\Pr[\Phi \leq \alpha]$, the probability that a threshold is met; and
- $\mathrm{Ex}[\Phi]$, the expected value of the criterion random variable.

Given two terminals s and t and an arc capacity (length) distribution, define Φ_{FLOW} as the value of a max (s,t)-flow and Φ_{PATH} as the value of a shortest (s,t)-path. We discuss the following performability measures:

$$\begin{aligned} FT(G,s,t,\{c_{e,i},p_{e,i}\},f_{\text{thresh}}) &= \Pr[\Phi_{FLOW} \geq f_{\text{thresh}}], \\ ST(G,s,t,\{d_{e,i},p_{e,i}\},l_{\text{thresh}}) &= \Pr[\Phi_{PATH} \leq l_{\text{thresh}}], \\ FE(G,s,t,\{c_{e,i},p_{e,i}\}) &= \mathrm{Ex}[\Phi_{FLOW}], \\ SE(G,s,t,\{d_{e,i},p_{e,i}\}) &= \mathrm{Ex}[\Phi_{PATH}]. \end{aligned}$$

For these measures, s is also referred to as the source node and t as the terminal. We also discuss work which produces complete distributions of the value of the max flow or shortest path.

2 Computational Complexity and Relationships among Problems

We start by discussing the differences between directed and undirected problems and the impact of node failures in sections 2.1 and 2.2 respectively. We then address issues of computational complexity in the remaining sections.

2.1 Directed vs. Undirected Networks

The general technique, illustrated in Figure 2, of replacing an undirected link with the two corresponding anti-symmetric directed links applies quite generally to network reliability problems. Specifically,

Theorem 2.1 *Given an undirected graph, $G = (V,E)$ and edge failure probabilities p, edge length distribution $\{l_{e,i},p_{e,i}\}$ or edge capacity distribution $\{c_{e,i},p_{e,i}\}$, if the directed graph $G' = (V,A)$ is generated by applying the transformation given in Figure 2 to all edges and if each directed arc reliability, p_a, length distribution $\{l_{a,i},p_{a,i}\}$ or capacity distribution $\{c_{a,i},p_{a,i}\}$ as appropriate, is set equal*

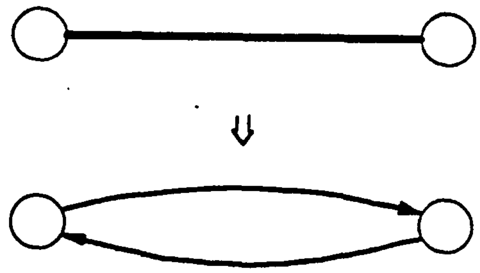

Figure 2: Transforming Undirected Graphs to Digraphs

to the reliability or distribution of the corresponding undirected edge, then

$$\begin{aligned}
Rel(G,s,K,p) &= Rel(G',s,K,\{p_a\}). \\
FT(G,s,t,\{c_{e,i}\},\{p_{e,i}\},f_{thresh}) &= FT(G',s,t,\{c_{a,i}\},\{p_{a,i}\},f_{thresh}) \\
ST(G,s,t,\{d_{e,i}\},\{p_{e,i}\},l_{thresh}) &= FT(G',s,t,\{d_{a,i}\},\{p_{a,i}\},l_{thresh}) \\
FE(G,s,t,\{c_{e,i}\},\{p_{e,i}\}) &= FT(G',s,t,\{c_{a,i}\},\{p_{a,i}\}) \\
SE(G,s,t,\{d_{e,i}\},\{p_{e,i}\}) &= FT(G',s,t,\{d_{a,i}\},\{p_{a,i}\})
\end{aligned}$$

This transformation is similar to transformations used in network flows. It is interesting and slightly surprising that it applies in this context since effectively, Theorem 2.1 allows us to treat the states of the anti-parallel pair of arcs as independent random variables when in fact they are not independent. For the proof of this result in the case of connectivity, see Nakazawa [301] and Ball [26], in the case of shortest paths see Hagstrom [182] and in the case of flows see Hagstrom [185]. This result does not necessarily hold in the context of more complex performability measures.

2.2 Node Failures

In many applications, nodes as well as arcs can fail. Consequently, one is led to consider models that can handle both node and arc failures. Fortunately, in the case of directed networks, using the transformation illustrated in Figure 3, a problem with unreliable nodes and arcs can be transformed into a problem with only unreliable arcs and perfectly reliable nodes. The transformation applies to the measures $Rel()$, $FT()$, $ST()$, $FE()$ and $SE()$ where in each case the arc that replaces the nodes inherits the characteristics of the corresponding node. When carrying out the transformation for a terminal i, the replacement node i_1 should not be a terminal and replacement node i_2 should

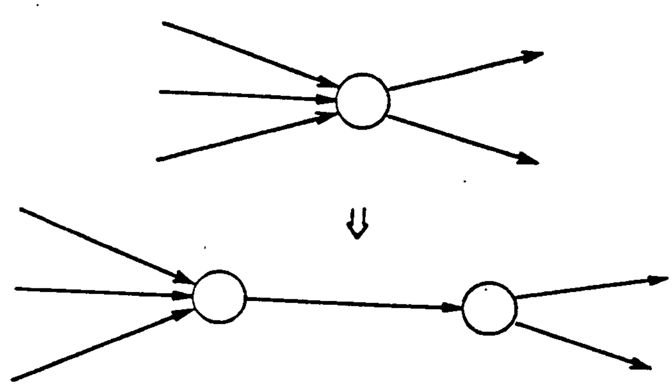

Figure 3: Replacing an Unreliable Node

be a terminal and when carrying out this transformation for a source node i, the replacement node i_1 should be the source and replacement node i_2 should not be the source node. See Ball [26] or Colbourn [88] for a general discussion of this transformation.

Theorem 2.1 and the above discusssion indicate that, from a practical standpoint, one would prefer codes for directed network reliability analysis over codes for undirected reliability analysis. By properly preparing input data, directed network codes can be used to analyze directed and undirected problems and problems with and without node failures.

2.3 An Introduction to the Complexity of Reliability Analysis

The computational problems most often studied by computer scientists and others interested in algorithms are recognition problems, such as determining if a graph contains a Hamiltonian cycle, and optimization problems, such as finding a minimum cost traveling salesman tour. Reliability analysis problems are fundamentally different. They compute a value that depends on the structure of a network as well as related data. Consequently, the analysis of their complexity involves concepts related to, but different from, the machinery used to analyze recognition and optimization problems: the classes P, NP and NP-Complete.

In order to most easily relate reliability analysis problems to more familiar combinatorial problems we consider the special case of the reliability analysis problem that arises when all individual component reliabilities are equal, i.e. $p_i = p$ for all components i. In this case, $Rel(SBS, p)$ can written as a polynomial in p with the following form:

$$Rel(SBS, p) = \sum_{i=0}^{m} F_i p^{m-i}(1-p)^i$$

This polynomial is the ***reliability polynomial.*** The associated computational problem, which we call the ***functional reliability analysis problem,*** takes as input a representation of an SBS and produces as output the vector $\{F_i\}$.

The general term in the reliability polynomial, $F_i p^{m-i}(1-p)^i$, is the probability that exactly $m-i$ components operate and the system operates. Thus, we can interpret F_i as:

$$F_i = |\{S : |S| = i \text{ and } \psi(T-S) = 1\}|.$$

We can see that the problem of determining each of the coefficients F_i is a counting problem. Whereas the output of the Hamiltonian cycle recognition problem is "yes" if the input graph contains a Hamiltonian cycle, and "no" if the graph does not, the output of the Hamiltonian cycle counting problem is the number of distinct Hamiltonian cycles contained in the graph. NP and NP-Complete are classes of recognition problems. The corresponding classes of counting problems are #P and #P-Complete. It is clear that any counting problem is at least as hard as the corresponding recognition problem. For example, if one knows the number of Hamiltonian cycles in a graph then one can immediately answer the question: "Is the number of Hamiltonian cycles greater than zero?". Thus, the counting versions of any NP-Complete problems are often trivially #P-Complete. On the other hand there are certain recognition problems solvable in polynomial time whose corresponding counting problems are #P-Complete. For example, the problem of determining whether a bipartite graph contains a perfect matching is polynomially solvable but the problem of determining the number of perfect matchings in a bipartite graph is #P-complete [408]. To make the presentation simpler, we do not delve further into detailed complexity issues but rather simply indicate whether problems are NP-hard or polynomial.

Many practical applications require the use of models with unequal component reliabilities. For the case of unequal component reliabilities, where all probabilities are rational numbers, we define the rational reliability analysis problem as follows. The input consists of a representation of an SBS and, for each component i a pair of integers a_i, b_i. The output is a pair of integers a, b where $a/b = Rel(SBS, \{a_i/b_i\})$.

We start by establishing that if a particular functional reliability analysis is NP-hard then the corresponding rational reliability analysis problem is NP-hard.

Proposition 2.2 *For any rational reliability analysis problem, r-Rel, and its corresponding functional reliability analysis problem, f-Rel, f-Rel can be reduced in polynomial time to r-Rel.*

Proof: An instance of f-Rel consists of a representation of an SBS. The required output is the set of coefficients $\{F_i\}$ of the reliability polynomial. To transform f-Rel to r-Rel we select $m+1$ rational probabilities $0 < p_0 < p_1 < \cdots < p_m < 1$. For $j = 0, 1, \cdots, m$, we denote by $r_j = Rel(SBS, p_j)$, the solution to the corresponding rational reliability analysis where all component reliabilities are set equal to p_j. We now can set up the following system of equations:

$$\sum_{i=0}^{m} F_i p_j^{m-i}(1-p_j)^i = r_j \quad \text{for } j = 0, 1, \cdots, m.$$

Having solved $m + 1$ rational reliability analysis problems, the p_j's and the r_j's are known. We have a system of $m + 1$ linear equations in $m + 1$ unknowns, the F_i's. The coefficient matrix has the Vandemonde property and consequently is non-singular so that the F_i's can be efficiently determined. □

We now investigate more carefully the structure of the reliability polynomial for SCBSs. Given an SCBS, we define:

$$\begin{aligned} m &= \text{number of components in the system,} \\ c &= \text{cardinality of a minimum cardinality cutset,} \\ n_c &= \text{number of minimum cardinality cutsets,} \\ \ell &= \text{cardinality of a minimum cardinality pathset,} \\ n_\ell &= \text{number of minimum cardinality pathsets.} \end{aligned}$$

It can immediately be seen that the coefficients of the reliability polynomial have the following properties:

$$\begin{aligned} 0 \leq F_i \leq \tbinom{n}{i} \quad &\text{for} \quad i = 0, 1, \cdots, m, \\ F_i = \tbinom{n}{i} \quad &\text{for} \quad i < c, \\ F_i = \tbinom{n}{i} - n_c \quad &\text{for} \quad i = c, \\ F_i = n_\ell \quad &\text{for} \quad i = m - \ell, \\ F_i = 0 \quad &\text{for} \quad i > m - \ell. \end{aligned}$$

These properties imply that by computing the reliability polynomial we immediately determine important properties of the SCBS. For example, by examining the reliability polynomial we can determine the size of a minimum cardinality cutset. Thus, if the minimum cardinality cutset recognition problem is NP-hard then computing the reliability polynomial is NP-hard. This line of reasoning leads to the following result.

Theorem 2.3 *For any SCBS if any one of the following five conditions hold then the functional and rational reliability analysis problems are NP-hard.*

1. *The minimum cardinality pathset recognition problem is NP-hard.*
2. *The minimum cardinality pathset counting problem is NP-hard.*
3. *The minimum cardinality cutset recognition problem is NP-hard.*
4. *The minimum cardinality cutset counting problem is NP-hard.*
5. *The problem of determining a general coefficient of the reliability polynomial is NP-hard.*

Proof: The result for the functional reliability analysis follows since the outputs of the five problems can be immediately deduced from the reliability polynomial. The result for the rational problem then follows from Proposition 2.2. □

We now use the framework just established to investigate network reliability problems.

2.4 The Complexity of Network Reliability Analysis

We now present the results concerning the complexity of network reliability analysis problems for the following problem classes: k-terminal, 2-terminal and all-terminal.

2.4.1 k-terminal

A minimum cardinality pathset for the k-terminal measure is a minimum cardinality Steiner tree. It is well-known [223] that the associated recognition problem in NP-hard for both directed and undirected networks so Theorem 2.3 implies that the associated functional and rational reliability analysis problems are NP-hard. Valiant [408] gives an alternate proof of this result by showing that computing $SN(K) = \sum F_i = |\{S : S$ is a subgraph that contains a path to each node in $K\}|$ is NP-hard. Here K is the set of terminals.

2.4.2 Two-terminal

The minimum cardinality pathset and cutset recognition problems associated with the 2-terminal measure are the shortest path and minimum cut problems respectively. Polynomial algorithms are known for both of these problems [295, 154]. Valiant [408] first showed that the 2-terminal reliability analysis problems were NP-hard. His reduction, which we now describe, is a good illustration of the proof techniques used in this area. The proof given below reduces the problem of computing $SN(K)$ to the 2-terminal rational reliability analysis problem:

> Given a graph $G = (N, A)$, a source node s and a set of terminal nodes K, construct G' by adding a node t and arcs (u, t) for each $u \in K$. We assign a failure probability of $1 - p$ to each (u, t) and a failure probability of $\frac{1}{2}$ to all arcs in the original network. Note that if all arcs in A have failure probability equal to $\frac{1}{2}$ then all random states of the arcs in A have probability $\frac{1}{2}^{|A|}$. If we define A_i as the number of subgraphs in G in which s is connected to exactly i members of K we now have: $\Pr[Rel(G'; s, t)] = \sum_i \sum_{S \subseteq K, |S|=i} \Pr[$there exist operating paths from s to S, but to no other nodes in $K - S$ and (u, t) operates for all $u \in S] = 2^{-|E|} \sum A_i(1 - p^i)$. By evaluating this reliability for $|K|$ different values of p we can set up a system of $|K|$ equations in $|K|$ unknowns, the A_i. The A_i can then be determined. The reduction is now complete since $A_{|K|} = SN(K)$. This reduction shows that the rational problem is NP-hard. A slight extension also shows that the functional problem is NP-hard.

Provan and Ball [330] give an alternate proof of this result by showing the problem of determining the number of minimum cardinality (s, t)-cuts is NP-hard.

2.4.3 All-terminal

For the directed all-terminal measure, the minimum cardinality pathset and cutset problems are the minimum cardinality spanning arborescence and minimum cardinality s-directed cut problems

	k-Term.	2-Term.	All Term. Dir.	All Term. Undir.
min. card. pathset rec.	! [223]	* [295]	*	*
min. card. cutset rec.	* [136, 154]	* [136, 154]	* [136, 154]	* [136, 154]
min. card. pathset count.	! [223]	* [29]	* [238]	* [238]
min. card. cutset count.	! [330]	! [330]	! [29]	* [29, 51]
gen. term. count.	! [223]	! [408]	! [330]	! [330]

Table 1: Complexity Results Related to k-Terminal, 2-Terminal and All-Terminal Reliability Analysis (* $\Rightarrow$ polynomial, ! $\Rightarrow$ NP-hard)

respectively. Both of these are polynomially solvable [121]. Provan and Ball [330] showed that the problem of counting minimum cardinality s-directed cuts is NP-hard, which in turn implies that the associated reliability analysis problems are NP-hard. For the undirected case, the minimum cardinality pathset and cutset recognition and counting problems are all polynomially solvable. However, Provan and Ball [330] showed that the problem of computing a general term in the reliability polynomial is NP-hard, implying that the undirected reliability analysis problems are NP-hard.

Table 1 summarizes the known complexity results for the five problem classes listed in Theorem 2.1 for the k-terminal, all-terminal and 2-terminal problems.

In light of these negative results, much research has been aimed at the analysis of structured networks. The widest class of networks known to be solvable in polynomial time involve series-parallel graphs and certain generalizations. Section 3 treats the solvable cases in more detail. Recent research has addressed the complexity of reliability analysis over structured networks, specifically directed acyclic networks and planar networks. In [328], Provan shows that the undirected two-terminal reliability problem remains NP-hard over planar networks having node degrees bounded by 3 and the directed 2-terminal reliability analysis problems remain NP-hard over acyclic planar networks having node degrees bounded by 3. Vertigan [414, 415] has recently shown that directed and undirected all-terminal reliability analysis problems are NP-hard when restricted to planar networks. There is a simple formula for directed all-terminal reliability analysis problem of acyclic networks [29].

The results of this section indicate that polynomial algorithms are only likely to exist for network reliability problems restricted to small classes of networks. Due to this fact a large amount of research has been devoted to the study of network reliability bounds and Monte Carlo approaches, the subjects of Sections 4 and 5, respectively.

3 Exact Computation of Reliability

In this section, we examine exact algorithms for computing reliability measures. We have seen that for general networks, all of the reliability measures of interest here are #P-complete. For this reason, we explore two main directions: exponential time exact algorithms for general networks, and polynomial time exact algorithms for restricted classes of networks.

Both directions rely on a simple but important observation: there exist graph transformations that leave the values of various reliability measures unchanged, and these can often be used to simplify the network used in the exact computation of reliability. Our first topic is such simplifying transformations.

3.1 Transformations and Reductions

An edge or arc that appears in no minpath is *irrelevant*: the operation or failure of the network is not affected by the operation or failure of such an irrelevant edge. The easiest simplifying transformation is the *deletion of irrelevant edges.* By definition, the transformation is reliability–preserving. Now for the transformation to be of practical use, we must be able to apply it efficiently (in polynomial time in the size of the network). For all-, k-, and two-terminal reliability, loops are always irrelevant. For k- and two-terminal reliability, so also is any edge having an endpoint in a 2-connected component containing no terminal; such edges can be found easily and deleted. For the directed reliability problems, the identification of irrelevant arcs is by no means an easy problem. Provan and Kulkarni [333] have shown that determining whether an arc is irrelevant for s,t-connectedness is NP-hard, although the general undirected problem admits an efficient solution.

We focus on the undirected problems here. An edge or arc that appears in *every* minpath is *mandatory.* After irrelevant edges have been deleted, any bridge (edge cutset of size one) that remains is mandatory. Let $G = (V, E)$ with terminal set $K \subseteq V$, and bridge $e \in E$ with operation probability p_e. The *contraction* $G \cdot e$ of an edge $e = \{x, y\}$ in G is obtained by removing e, identifying x and y and making the resulting node a terminal whenever $K \cap \{x, y\} \neq \emptyset$. The reliability of G, $Rel(G)$, satisfies $Rel(G) = p_e Rel(G \cdot e)$ when e is a mandatory edge. Thus the mapping from G to $G \cdot e$ is a reliability-preserving transformation *with multiplicative factor* p_e.

Two edges e, f having the same endpoints are in *parallel.* Since any minpath contains at most one of the two, and interchanging e and f is a natural bijection between the minimal pathsets containing one and the minpaths containing the other, the replacement of e and f by a single edge g having $p_g = 1 - (1 - p_e)(1 - p_f)$ is reliability-preserving. This is a *parallel reduction.* The notion of parallel reductions can be generalized when e and f are "substitutes"; see [190].

Two edges $e = \{x, y\}$ and $f = \{y, z\}$ are in *series* when y is a node of degree 2. In this case, any mincut contains at most of e or f, and interchanging e and f is a natural bijection between the mincuts containing e and those containing f. Thus a reliability-preserving transformation is obtained by removing the node y and the edges e, f, and adding the edge $g = \{x, z\}$ with $p_g = p_e p_f$ *provided* that y is not a terminal vertex. This is a *series reduction.* More generally, when two edges are "complements", similar reductions can be applied; see [190].

When a degree two *terminal* node is present, one cannot apply a series reduction. However, when x, y and z are all terminals, the same structural replacement is reliability-preserving with factor $1-(1-p_e)(1-p_f)$ if the new edge g is given probability $p_g = p_e p_f/(1-(1-p_e)(1-p_f))$. This is a ***degree-2 reduction***. There remain cases when y is a terminal, but at least one of x or z is not. Generalizations of the series and degree-2 reductions, the ***polygon-to-chain reductions***, are available [341, 429].

In essence, each of the simplifications thus far can be viewed as the replacement of some subnetwork by a subnetwork that has equivalent reliability characteristics, or characteristics that scale the original reliability measure by a fixed amount. With this in mind, consider a network $G = (V, E)$ with terminal set $K \subseteq V$. An induced subnetwork $H = (W, F)$ of G is ***s-attached*** if there is a set $A \subseteq W$ with $|A| \leq s$, for which every edge of G with one endpoint in $V \setminus W$ and one endpoint in W has an endpoint in A — in other words, only the nodes in A attach the subnetwork to the remainder of the network. A general class of simplifications arises by examining s-attached subgraphs for small s, and replacing each by a simpler s-attached subgraph.

A 1-attached subnetwork is connected to the rest of the network at a single cutnode. If the 1-attached subnetwork contains no terminal, all edges in it are irrelevant. On the other hand, if both the subnetwork and the remainder of the network contain terminals, the cutnode itself may be treated as a terminal since it is connected to the terminals in any minpath. So add the cutnode as a terminal. Then the network can be split into two subnetworks H and $G \setminus (W - A)$, and the reliability measure is the product of the measures for these two. This generalizes the notion of transformation to one that partitions the network into two or more subnetworks.

For 2-attached subnetworks, we view the replacement of the subnetwork as the determination of an equivalent edge. If H is a subnetwork attached at $\{x, y\}$, and H contains no terminals, we can determine the two-terminal reliability of H from x to y, and replace H by an edge $\{x, y\}$ whose operation probability is the two–terminal reliability found. When H contains terminals, the situation is more complicated, as it no longer suffices to know whether x can reach y; one must also know whether all of the internal terminals can reach x or y or both. Nevertheless, by permitting an edge to carry a number of probability values, rather than just an operation probability, transformations have been developed [418, 419]. The number of values that must be maintained here is independent of the size of the network, but grows exponentially with the number of attachment nodes; see section 3.2. A number of specific methods for employing the reduction of 2-attached and 3-attached subnetworks have been examined [12, 20, 165, 183, 184, 375]; Rosenthal [347, 348] was apparently the first to develop a general framework for these transformations, and for generalizations to k-attached subnetworks.

3.2 Efficient Algorithms for Restricted Classes

Our goal first and foremost is to obtain polynomial time algorithms for calculating reliability measures whenever possible. In view of the complexity results, we cannot hope at the present time to obtain efficient methods for networks in general. However, we can expect to treat restricted classes of networks efficiently.

From §3.1, we have a large collection of reliability-preserving reductions that, at least in their simplest forms, can be applied in polynomial time. Any set of reductions succeeds in reducing some (typically small) class of networks to a single vertex, and thus when the reductions can be applied efficiently, to an efficient algorithm for this class. For example, the elimination of irrelevant edges and the contraction of mandatory edges together give an algorithm for k-terminal reliability of trees. A better example is obtained by using also series and parallel reductions. Then the two-terminal reliability of series-parallel networks can be calculated. This result dates back at least to Lee [259]. When instead one adds degree-2 and parallel reductions, an all-terminal reliability algorithm for series-parallel graphs is immediate using characterization theorems for series-parallel graphs [118, 418].

For k-terminal reliability of series-parallel graphs, two linear time algorithms exist. Satyanarayana and Wood [362] employed series, parallel and degree-2 reductions, along with a type of 2-attached subnetwork reductions called the polygon-to-chain reductions, to reduce an arbitrary series-parallel network with terminals to a single vertex. Agrawal and Satyanarayana [11, 12] extended this to the directed reliability measures. By employing in addition the reduction of certain 3-attached networks, Politof and Satyanarayana [319, 320] extended these linear time algorithms to larger subclasses of the planar networks.

Wald and Colbourn [418, 419] obtained a linear time algorithm for k-terminal reliability by a different method. They generalized the notion of a transformation to permit each edge to carry a fixed finite number of reliability values, and developed a scheme for replacing an arbitrary 2-attached subnetwork by an edge having six associated values. When two subnetworks are attached at s nodes, the reliability of their union can be determined completely from the values on each of the subnetworks. This can in fact be accomplished by a dynamic programming algorithm whose running time is linear in the number of vertices, but exponential in the number of attachment vertices. Wald and Colbourn's method exploits the fact that series-parallel graphs can be recursively decomposed at vertex cutsets of size two — that is, series-parallel graphs have tree-width two. The extension of the linear time algorithm to any fixed tree-width is almost immediate [20, 285], *except* for the difficulty of recognizing networks of a given tree-width. El-Mallah and Colbourn [129] establish an algorithm for certain networks of tree-width three, and observe that the algorithms of Politof and Satyanarayana [319, 320] can be seen as algorithms on networks of small tree-width.

Obtaining efficient algorithms by restricting the tree-width accounts for the majority of efficient algorithms in the literature. It is important to note that planar graphs do not have tree-width independent of the number of vertices, although tree-width is bounded by $O(\sqrt{n})$ for an n-vertex planar network. This underlies an algorithm due to Bienstock [43] for planar networks that improves on general exact algorithms, but remains exponential time.

Beyond graphs with fixed tree-width, little is known for undirected reliability measures. Gilbert [176] developed an elegant recursive method for computing all-terminal reliability of complete networks when all edges operate with the same probability. The method requires linear time. It generalizes in a natural way to k-terminal reliability [88], and to complete bipartite [88] and related networks [67]. The method depends essentially on the observation that the number of nonisomorphic induced subgraphs is bounded by a polynomial in the number of vertices. A dynamic programming method then need only examine a polynomial number of subnetworks.

Turning to directed reliability measures, one important algorithm stands out. Ball and Provan [29] develop a linear time algorithm for computing reachability of acyclic directed networks. The algorithm is based on the observation that such a network is operational (for reachability) if and only if every non-root node has at least one of its incoming arcs operational.

There has recently been an extensive investigation of efficiently solvable classes of ***nodal*** reliability problems. Nodal two-terminal reliability (without edge failures) admits efficient algorithms for permutation graphs and interval graphs [3]. From this one can obtain efficient algorithms for two-terminal reliability with edge failures when the network has a line graph that is an interval or permutation graph, using a transformation from edge failure problems to node failure problems in [2].

The classes for which efficient exact algorithms are known are quite sparse, and do not appear to be those in which we expect to find most practical problems. Nevertheless, the presence of such exact algorithms even for sparse classes can be used to accelerate exact algorithms for larger classes that simplify the network in some way; and they have applications in computing bounds.

3.3 State-based Methods

When reliability-preserving transformations fail to reduce the network into a restricted class for which an efficient exact method is known, we are forced to resort to potentially exponential time methods. The first main class of these exact methods examines the possible states of the network.

A *state* of a network $G = (V, E)$ is a subset $S \subseteq E$ of operational edges. The conceptually simplest exact algorithm is *complete state enumeration*. Let $\mathcal{O}$ be the set of all operational states. Then

$$Rel(G) = \sum_{S \in \mathcal{O}} \prod_{e \in S} p_e \prod_{e \notin S} (1 - p_e).$$

By generating all states, and determining which is operational, the reliability is "easily" (but not efficiently) computed.

Of course, large groups of the states are easily seen to be operational without listing them out one by one. Hence an immediate improvement is obtained by generating the states in a more clever way. A basic ingredient in this is the Factoring Theorem:

$$Rel(G) = p_e Rel(G \cdot e) + (1 - p_e) Rel(G - e)$$

for any edge e of G. *Factoring*, also called *pivotal decomposition*, was explicitly introduced in Moskowitz [297] and Mine [289]. Factoring carried out until the networks produced have no edges is just complete state enumeration. However, some simple observations result in improvements. When $G - e$ is failed, any sequence of contractions and deletions results in a failed network, and hence there is no need to factor $G - e$. Moreover, although we may be unable to simplify G with a reliability-preserving transformation, we may well be able to simplify $G \cdot e$ or $G - e$.

Factoring with elimination of irrelevant edges, contraction of mandatory edges, and series, parallel and degree-2 reductions forms the basis of many exact algorithms in the literature [340, 341, 356, 428, 430]. Satyanarayana and Chang [356] analyzed the number of times a factoring step is required in an optimal factoring strategy (see also [429, 208]). For complete graphs, complete state enumeration examines $2^{\binom{n}{2}}$ states, while a factoring algorithm using series and parallel reductions examines only $(n-1)!$.

We state the factoring method more explicitly here:

```
procedure factor ( graph G );
        apply reliability-preserving transformations to G that
                delete irrelevant edges
                contract mandatory edges
                apply series reductions
                apply parallel reductions
                apply degree-2 reductions
                apply other reductions such as polygon-to-chain
            maintaining G with each edge having the probability
                resulting from the sequence of reductions, and also
                maintaining a multiplicative factor mult that
                results from the reductions

        if G has only one terminal remaining, return(mult)
        else
                select an edge e of G
                return(mult*(factor(G·e)+factor(G-e)))
end
```

Further improvements are possible by partitioning the graph into its biconnected or triconnected components at each step [430].

3.4 Path- and Cut-based Methods

Once basic reductions are done, complete state enumeration could be applied to finish the computation. Unless the reductions have succeeded in dramatically reducing the size of the graph, however, this remains impractical. It may nevertheless be possible to generate all minpaths of the network, and hence a method employing just the minpaths is in order. Suppose then that the minpaths $P_1, ..., P_h$ of G have been listed. Let E_i be the event that all edges in minpath P_i are operational. Then the reliability is just the probability that one (or more) of the events $\{E_i\}$ occurs. Unfortunately, the $\{E_i\}$ are not disjoint events, and hence we cannot simply sum their probabilities of occurrence. To be specific, $\Pr[E_1$ or $E_2]$ is $\Pr[E_1] + \Pr[E_2]$ - $\Pr[E_1$ and $E_2]$. Now $Rel(G) = \Pr[E_1$ or E_2 or $\ldots$ or $E_s]$, and hence

$$Rel(G) = \sum_{j=1}^{h} (-1)^{j+1} \sum_{I \subseteq \{1,\ldots,h\}, |I|=j} \Pr[E_I]. \quad (1)$$

where E_I is the event that all paths P_i with $i \in I$ are operational. This is a standard inclusion-exclusion expansion.

The algorithmic consequences of this formulation are immediate. Having a list of minpaths, one computes the probability of each subset of the minpaths occurring. To compute the reliability, one need only evaluate the above sum. In doing so, observe that an odd number of minpaths contributes positively to the sum, while an even number contributes negatively. This essentially reduces our problem to the production of the set of all minpaths. This algorithm has been suggested by a number of authors; see, for example, [165, 236, 259, 266, 293]. A naive implementation of this approach is, in fact, worse than complete state enumeration. The number of pathsets, h, may be exponential in n, and hence just the minpath generation requires exponential time. However, the more serious defect is that generating all subsets of minpaths in the naive manner takes 2^h time, which leaves us with a *doubly* exponential time algorithm for the reliability.

With a little care, this doubly exponential behaviour can be avoided. Every subset of the minpaths corresponds to a subgraph whose edge set is the union of the edge sets of the minpaths. With this in mind, an *i-formation* of a subgraph is a set of i minpaths whose union is the subgraph. A formation is *odd* when i is odd, *even* when i is even. A graph having a formation is a *K-subgraph*. Every odd formation of the subgraph contributes positively to the reliability, and every even formation contributes negatively. The *signed domination* of G with terminal set K of vertices, $sdom(G, K)$, is the number of odd formations of G minus the number of even formations of G. The *domination* $dom(G, K)$ is the absolute value of the signed domination. We usually write $sdom(G)$ and $dom(G)$ with the terminal set K understood. With these definitions, Satyanarayana and Prabhakar [360] simplified the expression for the reliability substantially:

$$Rel(G) = \sum_{H \subseteq G} sdom(H)\Pr[H],$$

where H varies over all states of G. This simplification is substantial, as it entails only the generation of all states rather than the generation of all subsets of the minpaths. However, some effort is still required if we are to improve on complete state enumeration. In particular, we now require the signed domination of each state.

In each of the directed reliability problems, Satyanarayana and his colleagues [355, 357, 358, 360, 425] completely determined the signed domination of each state. We outline the derivation in the reachability case here [357, 358].

The first goal is to determine which states have signed domination zero, and can therefore be ignored in the reliability expression. With this in mind, a state (subgraph) is *relevant* whenever it contains no irrelevant arcs. A subgraph containing irrelevant arcs has no formations whatsoever, and hence has signed domination zero. Thus we restrict our attention to relevant subgraphs. Among the relevant subgraphs, many have signed domination zero: precisely the cyclic subgraphs (subgraphs with some directed cycle) [360]. Moreover, an acyclic relevant digraph with m arcs and n nodes has signed domination $sdom(G) = (-1)^{m-n+1}$.

This study of domination in directed reliability problems is a remarkably clever application of combinatorial arguments. A method which naively requires doubly exponential time has been reduced to requiring the generation of the acyclic relevant graphs, and a trivial calculation for each. In practice, this affords a substantial improvement on complete state enumeration. Nevertheless, the number of acyclic subdigraphs is potentially exponential in n. Hence, despite a very significant reduction in computational effort, a very large computational task remains.

The use of signed domination in undirected problems arises in quite a different way. In undirected problems, cyclic relevant graphs have nonzero domination. Thus inclusion-exclusion algorithms using minpaths would require algorithms to compute the signed domination. However, the current algorithm to compute the signed domination of a single graph is the *same* as the algorithm which computes the reliability recursively using factoring. In fact, optimal factoring strategies using factoring with series and parallel reductions employ a number of factoring steps *equal* to the domination [356].

Let us once again suppose that we have an enumeration $P_1, ..., P_h$ of the minpaths, and let E_i be the event that all edges/arcs in minpath P_i are operational. As we have remarked, the events $\{E_i\}$ are not disjoint. We examine the strategy of forming a set of disjoint events. Let $\bar{E}_i$ denote the complement of event E_i. Now define the event $D_1 = E_1$, and in general, $D_i = \bar{E}_1 \cap \bar{E}_2 \cap \ldots \cap \bar{E}_i - 1 \cap E_i$. The events D_i are disjoint, and hence are often called "disjoint product" events. Moreover, $Rel(G) = \sum_{i=1}^{h} \Pr[D_i]$. In employing this approach, one must obtain a formula for $\Pr[D_i]$ in terms of the states of the edges/arcs. Each event E_i can be written as a boolean expression which is the product of the states of the edges/arcs in minpath P_i. Hence D_i can also be written as a boolean expression. For this reason, algorithms using disjoint products are sometimes called "boolean algebra" methods.

There is a wide variety of boolean algebra methods. The pioneering paper here is by Fratta and Montanari [163]. Subsequent improvements have employed two basic ideas [4, 8, 9, 27, 30, 116, 273, 274, 314, 335, 403]. Firstly, observe that the expression for event D_i is a complex boolean expression, involving complements of events E_i and not just edge states and complements of edge states. Evaluation of D_i requires simplification of the boolean expression to one which involves only edge states and their complements. Most methods are primarily concerned with making this simplification efficient, and with producing resulting expressions which are as small as possible. Secondly, in order to make simplification easier, most methods employ some simple strategy for reordering the minpaths prior to defining the events $\{D_i\}$. The particular events defined depend heavily on the ordering of the minpaths chosen. A typical heuristic here is to sort the minpaths, placing minpaths with the fewest edges/arcs first. Despite these heuristic improvements, there is no known polynomial bound in general for the length of the simplified boolean expression for $Rel(G)$ in terms of the number of minpaths. Provan [329] develops a general theory for the size of the boolean expression in special cases.

In the case of all-terminal reliability and reachability, however, such a polynomial bound is achievable using an algorithm of Ball and Nemhauser [27] (see also [30, 32]) to produce a boolean formula describing disjoint events based on minpaths in which the number of terms equals the number of minpaths.

Colbourn and Pulleyblank [104] give an algorithm for ordering the minpaths in k-terminal reliability problems so that the number of terms does not exceed the number of spanning trees. However, this may exceed the number of minpaths by an exponential factor; see also Chari [81]. Ball and Provan

[30, 32] treat the optimal ordering of minpaths in a general setting.

Our goal to this point has been to compute directly the probability of obtaining a pathset. An indirect way to do this is to compute instead the probability of obtaining a cutset. The reliability is then one minus the cutset probability. Let us suppose that we have an enumeration of mincuts. Let $C_1, ..., C_g$ be the mincuts, and let E_i be the event that all edges in mincut C_i fail. Once again, we can apply the strategy of inclusion-exclusion [214, 307] or the strategy of disjoint products [25, 34, 196, 334]. The advantage of this approach computationally is that the number of mincuts is often much smaller than the number of minpaths. In fact, many methods generate both minpaths and mincuts, and proceed with the smaller collection of the two (see, for example, [116]). Although one may typically prefer working with mincuts because of the smaller number, no current theory analogous to domination accounts for which states are relevant. This is a serious drawback to approaches based on cutsets.

A recent algorithm due to Provan and Ball [331] determines two-terminal reliability in time which is polynomial in the number of mincuts. Their algorithm has much the same flavor as a dynamic programming strategy suggested by Buzacott [74, 75, 76, 77]. Buzacott's algorithm applies more generally than the Provan-Ball strategy, but does not perform nearly as well when the number of mincuts is small. Ball and Provan [31] develop a common extension of both methods.

Every algorithm mentioned here requires exponential time in the worst case, whether it enumerates states, minpaths, or mincuts. A complete graph on n nodes, for example, has $2^n - 1$ mincuts, n^{n-2} spanning trees, and $2^{\binom{n}{2}}$ states. If exponential algorithms are the best one can hope for, it is reasonable to consider algorithms that explore a *relatively* small number of states. Out of the many algorithms mentioned here, three of the methods are especially noteworthy. Methods based on domination ensure that only relevant states are examined, and thereby improve on almost all other inclusion-exclusion methods. Methods based on factoring (for the undirected case) also generate only relevant states. The Satyanarayana-Chang approach obtains the best possible computation time using series and parallel reductions. In fact, in the all-terminal case, since the number of spanning trees exceeds the domination, the algorithm improves on all methods based on minpaths. Finally, methods based on disjoint products, although difficult to analyze in general, give two important algorithms: the Ball-Nemhauser algorithm to compute all-terminal reliability in time polynomial in the number of minpaths, and the Provan-Ball algorithm which computes two-terminal reliability in time polynomial in the number of mincuts.

This useful device of measuring complexity in terms of the number of minpaths, mincuts, or relevant states enables one to see that the methods singled out are indeed improvements on the vast remainder of reliability algorithms.

4 Bounds on Network Reliability

Essentially all reliability problems of interest are #P-complete, and hence the fact that the exact algorithms described are quite inefficient comes as no surprise. Nevertheless, in assessing the reliability of a network, it is imperative that the assessment can be completed in a "reasonable" amount of time.

The conflicting desires for fast computation and for great accuracy have led to a varied collection of methods for estimating reliability measures.

Two main themes arise: the ***estimation*** of reliability by Monte Carlo sampling techniques, and the ***bounding*** of reliability. In the first, the goal is to obtain an accurate estimate of a reliability measure by examining a small fraction of the states, chosen randomly. This leads to a point estimate of the reliability measure, along with confidence intervals for the estimate. Bounding is different, both in technique and in result. Current techniques for bounding attempt to find combinatorial or algebraic structure in the reliability problem, permitting the deduction of structural information upon examination of a small fraction of the states. Unlike Monte Carlo methods, the states examined are not chosen randomly. The goal of bounding is to produce ***absolute*** upper and lower bounds on the reliability measure.

It is perhaps misleading to draw a line between Monte Carlo methods and bounding techniques, since a number of the Monte Carlo methods employ bounding as a vehicle to limit the size of the sample space. In this section, we explore bounding methods, leaving Monte Carlo techniques for Section 5. We first examine the case when all edges operate with the same known probability, independently. We then examine the case when edges operate independently with arbitrary (but still known) probabilities.

4.1 Equal Edge Failure Probabilities

In this section, we treat bounds that are valid when every edge has the same operation probability p; in this case, as we have seen, reliability can be expressed as a polynomial in p. A subgraph with operational edges $E' \subseteq E$ now arises with probability $p^{|E'|}(1-p)^{|E-E'|}$. Consequently, the probability of obtaining a subgraph depends only on the number of edges it contains. Then let N_i denote the number of operational subgraphs with i edges. The probability of network operation, denoted $Rel(G,p)$ or simply $Rel(p)$ is then

$$Rel(p) = \sum_{i=0}^{m} N_i p^i (1-p)^{m-i}.$$

Thus the probability is a polynomial in p, which we saw before, called the *reliability polynomial.* This formulation is in terms of pathsets. Another formulation is obtained by examining cutsets. Letting C_i be the number of i-edge cutsets (leaving $m-i$ operational edges),

$$Rel(p) = 1 - \sum_{i=0}^{m} C_i (1-p)^i p^{m-i}.$$

Still another formulation, and probably the most common, is obtained by examining complements of pathsets. Let F_i denote the number of sets of i edges for which the $m-i$ remaining edges form a pathset. Then

$$Rel(p) = \sum_{i=0}^{m} F_i (1-p)^i p^{m-i}.$$

4.1.1 Basic Observations

The first goal in introducing the reliability polynomial is to obtain a compact encoding of reliability information to compare candidate topologies. Moore and Shannon [296] pioneered this approach in their study of electrical relays. Moskowitz [297] and Mine [289] employed a simple strategy for computing reliability polynomials in certain two-terminal problems on series-parallel networks. The application to computer networks, and the reliability polynomials introduced here, were studied by Kel'mans [228]. Kel'mans was apparently the first to make a fundamental observation about comparing two networks via their reliability polynomial. He proved that for two graphs G and H, their reliability polynomials may "cross"; that is, one may be more reliable for one value of p, while the other is more reliable for another value of p. Kel'mans [231, 232, 298] proved that for a given number of vertices and edges, in certain cases there is no graph that is most reliable for all edge operation probabilities. Thus reliability is more than simple graph parameters; it is truly a function of link reliabilities.

4.1.2 Computing Some Coefficients Exactly

In section 2.3, we saw that being able to compute the size ℓ of a minimum cardinality pathset and the size c of a minimum cardinality cutset enables us to determine a number of coefficients exactly. When ℓ is efficiently computable, further information can often be obtained by computing N_ℓ. In the k-terminal problem, this is truly hopeless; one would have to count minimal Steiner trees, a #P-complete problem. In the other two cases, however, efficient algorithms exist.

In the all-terminal problem, N_l is the number of spanning trees. Kirchoff [238] in 1847 developed an elegant method for counting spanning trees; it is developed in a computationally useful form in [63]. In the two-terminal problem, N_l is the number of shortest s,t-paths. Ball and Provan [29] developed an efficient strategy for computing this number. Brecht and Colbourn [59] establish that for any fixed $k \geq 0$, pathsets of size $l+k$ in the two-terminal problem can be counted efficiently.

An efficient algorithm to compute c enables us to determine a number of additional coefficients exactly as well. This problem is tractable in each of the three cases of interest, using the same method in each case. Menger's theorem [286] guarantees that the minimum s,t-cut has size c exactly when the maximum number of edge-disjoint s,t-paths is c. This problem is easily recast as a network flow problem, with all edge capacities equal to 1.

Once again, having computed c it would be valuable to compute C_c, the number of minimum cardinality cutsets. Provan and Ball have shown that in the two-terminal case, computing just this coefficient is #P-complete [330]; since the k-terminal problem includes the two-terminal problem, computing C_c in either of these problems is apparently intractable. However, in the all-terminal problem, Ball and Provan [28] devised a method for computing C_c efficiently. Lomonosov and Polesskii [278] and Bixby [51] have shown that every n-node graph G has $C_c(G) \leq \binom{n}{2}$. In addition, observe that for any i and any edge e of G, $C_i(G) = C_i(G \cdot e) + C_{i-1}(G - e)$. These two facts were used by Ramanathan and Colbourn [338] to develop an efficient method for counting cutsets of size $c+k$ for any fixed $k \geq 0$.

4.1.3 Simple Bounds

The computation of many of the coefficients leaves us with many coefficients about which we have said nothing. Kel'mans [227, 228] observed that when p is close to zero, for all-terminal reliability we have

$$Rel_A(p) \approx N_{n-1}p^{n-1}(1-p)^{m-n+1}$$

and when p is close to 1,

$$Rel_A(p) \approx 1 - C_c p^{m-c}(1-p)^c.$$

In the setting of the reliability polynomials introduced, a precise statement of the Kel'mans approximations is valid for all p:

$$N_{n-1}p^{n-1}(1-p)^{m-n+1} \leq Rel_A(p) \leq 1 - C_c p^{m-c}(1-p)^c.$$

Thus the Kel'mans approximations can be viewed as absolute bounds on the reliability polynomial. The essential observation here is that for extreme values of p, either the term involving N_{n-1} or the term involving C_c dominates the remaining terms. Another observation is inherent in the Kel'mans approach. We know that $N_i + C_{m-i} = \binom{m}{i}$, and hence we have $0 \leq N_i, C_i \leq \binom{m}{i}$.

This observation leads us to a simple set of bounds first formulated by Jacobs [211], and improved to this current form by Van Slyke and Frank [412]:

$$Rel_A(p) \geq N_{n-1}p^{n-1}(1-p)^{m-n+1} + N_{m-c}p^{m-c}(1-p)^c + \sum_{i=m-c+1}^{m} \binom{m}{i} p^i(1-p)^{m-i}.$$

$$Rel_A(p) \leq N_{n-1}p^{n-1}(1-p)^{m-n+1} + \sum_{i=n}^{m} \binom{m}{i} p^i(1-p)^{m-i}.$$

In the lower bound, each "unknown" N_i is approximated by zero; the known coefficients are for $i < n-1$ (zero), $i = n-1$ (the number of spanning trees), $i = m-c$ (the $m-c$-edge subgraphs whose complement is *not* a minimum cardinality cutset), and $i > m-c$ (all possible i-edge subgraphs). In the upper bound, the unknown N_i are approximated by $\binom{m}{i}$. The extension to two-terminal reliability is straightforward; simply substitute l for $n-1$ throughout. The extension to k-terminal reliability is complicated by the difficulty of computing l. A lower bound is nevertheless obtained by underestimating N_l as 0; an upper bound is obtained merely by using an underestimate for l itself.

These bounds are *extremely* weak, and provide useful information only when p is very near 0 or 1.

4.1.4 Coherence

Bounding the unknown N_i depends very heavily on knowledge of the combinatorial structure of the collection of operational subgraphs. Most reliability problems of interest to us have the property of *coherence.* With this in mind, consider the set $\mathcal{D} = \{D_1, D_2, D_3, ..., D_r\}$ in which each D_i is a set of edges for which $E - D_i$ is operational. The set $\mathcal{D}$ has a set for each operational subgraph, in which the edges *not* in the operational subgraph are listed. Defining $\mathcal{D}$ in this way, we produce a hereditary family of sets (or *complex*) called the *$\mathcal{F}$-complex* (that is, if $S \in \mathcal{D}$ and $S' \subseteq S$, then $S' \in \mathcal{D}$). The fact that the family of sets produced is hereditary is precisely the property of coherence. It is also no coincidence that F_i, defined earlier, is precisely the number of sets of cardinality i in $\mathcal{D}$. In fact, the reliability polynomial for a hereditary family $\mathcal{D}$ is completely prescribed by its *F-vector* (F_0, F_1, ..., F_d) where $d = m - l$.

The property of coherence also suggests a particularly appropriate way of viewing the family $\mathcal{D}$, as a partial order whose relation is set inclusion.

The key to using coherence in obtaining bounds is the following. Consider all of the i-sets in a hereditary family $\mathcal{D}$. Since the family is hereditary, there must be a number of $i-1$-sets contained in the collection of i-sets; the minimum number of such induced $i-1$-sets is a lower bound on F_{i-1}. The minimization of F_{i-1} as a function of F_i is a well-known problem in extremal set theory, apparently first studied by Sperner [386]. He proved that $F_{i-1} \geq \frac{i}{m-i+1} F_i$.

Birnbaum, Esary, and Saunders [50] also prove this result, and observe that it has obvious consequences to the coefficients in the reliability polynomial. Bauer, Boesch, Suffel, and Tindell [38] observe that this has a very simple interpretation: the fraction of operational subgraphs with i edges over all subgraphs with i edges is nondecreasing as i increases. They also observe that Sperner's theorem can be used at little computational effort to improve the simple bounds. We assume that l, c, F_{m-l}, and F_c are available. Then

$$Rel(p) \geq \sum_{i=0}^{c-1} \binom{m}{i} p^{m-i}(1-p)^i + F_c p^{m-c}(1-p)^c + \sum_{i=c+1}^{m-l} F_{m-l} \frac{\binom{m}{i}}{\binom{m}{m-l}} p^{m-i}(1-p)^i.$$

$$Rel(p) \leq \sum_{i=0}^{c-1} \binom{m}{i} p^{m-i}(1-p)^i + \sum_{i=c}^{m-l-1} F_c \frac{\binom{m}{i}}{\binom{m}{c}} p^{m-i}(1-p)^i + F_{m-l} p^l (1-p)^{m-l}.$$

This bounding technique applies to any coherent system, and affords significant improvements on the simple bounds [88].

One method to improve on these bounds is to improve on Sperner's theorem. The best possible result in this direction was obtained by Kruskal [244] and independently by Katona [225]. Simplified proofs of this key result have been given by Daykin [108] and Frankl [161]. The Kruskal-Katona theorem places a lower bound $F_i^{i-1/i}$ on F_{i-1} given F_i; alternatively, it places an upper bound $F_{i-1}^{i/i-1}$ on F_i given F_{i-1}. The form of the bound is of little importance here, except to note that $x^{j/i}$ can be efficiently calculated, and that whenever $x \geq y$, $x^{j/i} \geq y^{j/i}$. Van Slyke and Frank [412] used the Kruskal-Katona theorem to bound individual coefficients in the reliability polynomial. Recall that F_c is $\binom{m}{c} - C_c$. For all terminal reliability, we can therefore compute F_c exactly; in the remaining two cases, we cannot hope to compute F_c, but we can easily compute F_{c-1}. In general, let us suppose that we can compute a sequence of coefficients F_0, F_1, ..., F_s efficiently. Then the Kruskal-Katona theorem gives us an upper bound on F_{s+1}. Then given an upper bound on F_{s+1}, we proceed in the same way to obtain upper bounds on F_{s+2}, F_{s+3} and so on.

Lower bounds are obtained symmetrically. We compute some sequence of coefficients $F_{m-l}, F_{m-l+1}, \ldots, F_m$ efficiently. For all-terminal and two-terminal reliability, F_{m-l} is the number of spanning trees and shortest paths, respectively. In the k-terminal problem, we can take $l = k - 1$ (for example) in order to compute this sequence. In any event let $d = m - l$. Knowing F_d, the Kruskal-Katona theorem gives a lower bound on F_{d-1}, namely $F_d^{d-1/d}$. This application of the Kruskal-Katona theorem, first done by Van Slyke and Frank [412], gives us the *Kruskal-Katona* bounds.

4.1.5 Shellability

The Kruskal-Katona theorem is best possible for hereditary families of sets. We therefore have no hope of improving on the Kruskal-Katona bounds without additional information. Such additional information could come in a number of ways. One would be efficient algorithms for computing (or even bounding more tightly) one or more of the unknown F_i. Another would be to observe that the particular hereditary family which arises has some special combinatorial structure. This latter approach is promising, because although complements of pathsets in coherent systems produce a hereditary family, not all hereditary families arise in this way.

In fact, the $\mathcal{F}$-complex in an all-terminal reliability problem is a matroid, the *cographic matroid* of the graph. For now, we restrict our attention to the all-terminal problem. No progress appears to have been made on characterizing F-vectors of cographic matroids, and so one might ask what the F-vector of a matroid can be in general. Even on this problem, no progress has been made directly. However, we can identify a class of hereditary systems that are intermediate between matroids and hereditary systems in general, and results *are* available here.

Provan and Billera [332] prove a powerful result about the structure of matroids, which (together with later results) constrains their F-vectors; they observe that matroids are "shellable" complexes. The importance of the Provan-Billera result in our reliability investigations is that they suggest the possibility of exploiting shellability to improve on the Kruskal-Katona bounds. Of course, this requires that we obtain structure theorems for shellable systems. An *interval* $[L, U]$ is a family of subsets $\{S : L \subseteq S \subseteq U\}$. An interval partition of a complex is a collection of disjoint intervals for which every set in the complex belongs to precisely one interval. A complex is *partitionable* if it has an interval partition $[L_i, U_i]$, $1 \le i \le J$ with U_i a base for all i. Shellable complexes are all partitionable.

Ball and Nemhauser [27] developed the application of the partition property to reliability. Consider a shellable complex with b bases; let $\{[L_i, U_i]|1 \le i \le b\}$ be an interval partition for this complex. $[L_i, U_i]$ is a compact encoding of all sets in this interval; the probability that any one of these sets arises is then $p^{m-|U_i|}(1-p)^{|L_i|}$. In other words, $|L_i|$ edges must fail, and $m - |U_i|$ edges must operate; the state of the remaining edges is of no consequence. Every U_i is a base in the complex; hence the cardinality of each U_i is the same, the rank d of a base. However, the ranks of the L_i are not all identical; we therefore define $H_i = |\{L_j : 1 \le j \le b, |L_j| = i\}|$. This gives rise to an *H-vector* $(H_0, \ldots, H_d)$. The coefficient H_i counts intervals in the partition whose lower set has rank i.

This gives yet another form of the reliability polynomial:

$$Rel(p) = p^l \sum_{i=0}^{d} H_i(1-p)^i.$$

Here, l is the cardinality of a minimum cardinality pathset (spanning tree), and $d = m - l$ is then the rank of a base. More concretely, in an n-vertex m-edge graph, $l = n - 1$ and $d = m - n + 1$.

Naturally, any information about the H-vector also provides information about the reliability polynomial. However, to place the H-vector in appropriate context, it is worthwhile examining the relation between the H-vector and the F-vector for a shellable complex. The H-vector for any complex can be defined directly in terms of the F-vector (see, for example, [392]). In the partitionable case, however, the correspondence is easily seen combinatorially.

Consider the sets of rank k in the complex. These are counted by F_k. Now any interval $[L_i, U_i]$ accounts for certain of these sets. Let r be the rank of L_i. If $r > k$, the interval accounts for 0 of the sets in F_k; however, if $r \leq k$, it accounts for $\binom{d-r}{k-r}$ of the sets. Hence, we find that $F_k = \sum_{r=0}^{k} H_r \binom{d-r}{k-r}$. Equating the F-vector and H-vector forms of the reliability polynomial gives an expression for H_i in terms of the F-vector, namely:

$$H_k = \sum_{r=0}^{k} F_r(-1)^{k-r} \binom{d-r}{k-r}.$$

This expression allows us to efficiently compute $H_0, ..., H_s$ from $F_0, ..., F_s$. Another obvious, but useful, fact is that $F_d = \sum_{i=0}^{d} H_i$.

Following pioneering research of Macaulay [282], Stanley [47, 390, 391, 392, 394] has studied H-vectors in an algebraic context, as "Hilbert functions of graded algebras." Stanley obtained a lower bound $H_i^{<i-1/i>}$ on H_{i-1} given H_i that is tight for shellable complexes in general; this in turn gives an upper bound $H_{i-1}^{<i/i-1>}$ on H_i given H_{i-1}. For our purposes, three things are important. First of all, for $k \geq j \geq i$, $x^{<k/i>} = x^{<j/i><k/j>}$. Secondly, given x, j and i we can compute $x^{<j/i>}$ efficiently. Thirdly, whenever $x \geq y$, $x^{<j/i>} \geq y^{<j/i>}$.

Stanley's theorem can be used to obtain efficiently computable bounds on the reliability polynomial. Given a prefix $(F_0, ..., F_s)$ of the F-vector, we can efficiently compute a prefix $(H_0, ..., H_s)$ of the H-vector.

Knowing this prefix, we obtain some straightforward bounds; these apply to shellable systems in general, but we present them here in the all-terminal case.

$$Rel(p) \geq p^{n-1} \sum_{i=0}^{s} H_i(1-p)^i.$$

$$Rel(p) \leq p^{n-1} \left[\sum_{i=0}^{s} H_i(1-p)^i + \sum_{i=s+1}^{d} H_s^{<i/s>}(1-p)^i \right].$$

This exploits information about the size of the minimum cardinality cutset and, where available, the number of such cutsets. This simple formulation ignores a substantial piece of information, the number of spanning trees. This is introduced by recalling that $F_d = \sum_{i=0}^{d} H_i$. Ball and Provan [28, 29] develop bounds that incorporate this additional information; they suggest a very useful pictorial tool for thinking about the problem. Associate with each H_i a "bucket." Now suppose we have F_d "balls." Our task is to place all of the balls into buckets, so that the number of balls in the ith bucket, n_i, satisfies $n_i \leq n_{i-1}^{<i/i-1>}$.

How do we distribute the balls so as to maximize or minimize the reliability polynomial? These distributions, when found, give an upper and a lower bound on the reliability polynomial. Consider carefully the sum in the reliability polynomial: $\sum_{i=0}^{d} H_i(1-p)^i$. Since $0 < p < 1$, the sum is larger when the lower order coefficients are larger. In fact, for two H-vectors $(H_0, ..., H_d)$ and $(J_0, ..., J_d)$, whenever $\sum_{j=0}^{i} H_j \geq \sum_{j=0}^{i} J_j$ for all i, the reliability polynomial for the H_i dominates the reliability polynomial for the J_i.

This last simple observation suggests the technique for obtaining bounds. In the pictorial model, an upper bound is obtained by placing balls in the leftmost possible buckets (with buckets $0, ..., d$ from left to right); symmetrically, a lower bound is obtained by placing balls in the rightmost possible buckets. We are not totally without constraints in making these placements, as we know in advance the contents of buckets $0, ..., s$.

With this picture in mind, we give a more precise description. We produce coefficients $\overline{H}_i$ for an upper bound polynomial, and $\underline{H}_i$ for a lower bound polynomial, using the prefix $(H_0, ..., H_s)$ and F_d. The steps are:

1. For $i = 0, ..., s$, set $\underline{H}_i = H_i = \overline{H}_i$.
2. For $i = s+1, s+2, ..., d$, set

$$\underline{H}_i = min \left[r : \sum_{j=0}^{i-1} \underline{H}_j + \sum_{j=i}^{d} r^{<j/i>} \geq F_d \right].$$

$$\overline{H}_i = max \left[r : r \leq \overline{H}_{i-1}^{<i/i-1>} \text{ and } \sum_{j=0}^{i-1} \overline{H}_j + r \leq F_d \right].$$

An explanation in plain text is in order. In each bound, we determine the number of balls in each bucket from 0 to d in turn; as we remarked, the contents of buckets $0, ..., s$ are known. For subsequent buckets, the upper bound is determined as follows. The number of balls which can go in the current bucket is bounded by Stanley's theorem, and is also bounded by the fact that there is a fixed number of balls remaining to be distributed. If there are more balls remaining than we can place in the current bucket, we place as many as we can. If all can be placed in the current bucket, we do so; in this case, all balls have been distributed and the remaining buckets are empty. The lower bound is determined by placing as few balls as possible.

The method leads to a very powerful set of bounds, the *Ball-Provan* bounds:

$$Rel(p) \geq p^{n-1} \sum_{i=0}^{d} \underline{H}_i (1-p)^i.$$

$$Rel(p) \leq p^{n-1} \sum_{i=0}^{d} \overline{H}_i (1-p)^i.$$

Unlike the Kruskal-Katona bounds, in the case of the Ball-Provan bounds it is not generally the case that $\underline{H}_i \leq H_i \leq \overline{H}_i$. Brown, Colbourn and Devitt [67] observe that a number of simple network transformations can be used to determine bounds $L_i \leq H_i \leq U_i$ efficiently. Incorporating these coefficient bounds on the H-vector into the Ball-Provan process can result in substantial improvements.

4.1.6 Polyhedral Complexes and Matroid Ports

The Ball-Provan bounds as developed here apply to all-terminal reliability and to reachability. For reachability, Provan [326] observes that the $\mathcal{F}$-complex is a "polyhedral complex", and uses a theorem of Billera and Lee [48] to obtain efficiently computable bounds on reliability that are tight for polyhedral complexes.

The matroid structure of the all-terminal problem and the polyhedral structure of reachability both lead to dramatic improvements over the Kruskal-Katona bounds for general coherent reliability problems. Building on a structure theorem of Colbourn and Pulleyblank [104], Chari [81] characterized two-terminal complexes in terms of "matroid ports", and generalized these to develop some remarkable structure theorems about $\mathcal{F}$-complexes from k-terminal problems. For an n-vertex connected graph having k terminals, and edge set E, $|E| = m$, let $\mathcal{F}$ be its $\mathcal{F}$-complex. The blocking complex $\mathcal{F}^*$ of $\mathcal{F}$ is $\{E \setminus S : S \in 2^E \setminus \mathcal{F}\}$. The F-vector $(F_0, \ldots, F_m)$ of $\mathcal{F}$ and the F-vector $(F_0^*, \ldots, F_m^*)$ of $\mathcal{F}^*$ satisfy $F_i + F_{m-i}^* = \binom{m}{i}$ for $0 \leq i \leq m$. Chari [81] shows that the subcomplex $\mathcal{F}^{(m-n+1)}$ obtained by removing all sets from $\mathcal{F}$ of cardinality exceeding $m-n+1$, is a shellable complex. Hence, given bounds on the single coefficient F_{m-n+1}, the Ball-Provan strategy can be applied to this k-terminal problem in order to bound $(F_0, \ldots, F_{m-n+1})$. What about the remaining coefficients? Chari further proves that $\mathcal{F}^{*(n-2)}$, obtained from $\mathcal{F}^*$ by removing all sets of cardinality exceeding $n-2$, is also shellable. Hence the Ball-Provan bounds can be applied again to bound $(F_0^*, \ldots, F_{n-2}^*)$, or equivalently to bound $(F_m, \ldots, F_{m-n+2})$.

All of the approaches developed for equal edge failure probabilities to date examine extremal results for complexes that are more general than those actually arising in reliability problems on graphs. It remains a very active area of research to determine least or most reliable graphs, rather than complexes, given the values of some specified graph parameters. Even the characterization of least reliable graphs for specified numbers of vertices and edges remains unresolved, however [57].

4.1.7 The Standard Form

We consider yet another form of the reliability polynomial. Until this point, the underlying framework has been the notion of state enumeration, in which either operational or failed states are enumerated. Many reliability algorithms do not operate in this manner, but instead use path or cut enumeration.

We have seen earlier that one of the more useful exact algorithms employs the theory of domination. We return briefly to this theory, to develop an interpretation of coefficients in the reliability polynomial.

Extending the notion of domination, the *i-parity* $P_i(G)$ is defined as follows. Let S_i be all i-edge subgraphs of G. Then $P_i(G) = \sum_{H \in S_i} sdom(H, K)$. Satyanarayana and Khalil [359] have established that $P_i(G) = P_{i-1}(G \cdot e) + P_i(G - e) - P_{i-1}(G - e)$.

In obtaining reliability via pathsets, every formation is considered exactly once. Letting $\{G_1, ..., G_t\}$ denote all K-subgraphs of G, $Rel_k(G) = \sum_{i=1}^{t} sdom(G_i, K)\Pr[G_i]$. Hence when all edge probabilities are equal, we obtain another form of the reliability polynomial, the *standard* form:

$$Rel(p) = \sum_{i=0}^{m} P_i p^i.$$

In other words, the parities are precisely the coefficients of this reliability polynomial, the *P-vector.*

The characterization of P-vectors arising from reliability polynomials has not been widely studied. Two remarks are of interest here. First, Satyanarayana and Chang [356] establish that the coefficients in the P-vector for all-terminal reliability alternate in sign. Second, Brown and Colbourn [64] conjecture that the P-vector is log concave; that is, a P-vector for an all-terminal reliability polynomial satisfies $P_i^2 \geq P_{i-1}P_{i+1}$ for every i. They prove a partial result in support of this conjecture. Little else is currently known about the P-vector. One can easily see that $P_i = 0$ for $i < n-1$, and that P_{n-1} is the number of spanning trees of the network. However, essentially nothing is known about the remaining coefficients in the P-vector (except, of course, those relations inherited by equivalence with the H-vector and F-vector).

4.2 Arbitrary Edge Failure Probabilities

When edges fail with different probabilities, the $\mathcal{F}$-complex contains all of the information about reliability, given the operation probability of each edge. However, the F-vector and the reliability polynomial are no longer applicable. This has led to a number of techniques for using the network structure to obtain bounds. We explore the major techniques here, referring the interested reader to [88] for proofs and further discussion.

4.2.1 Edge-packing

Let $G = (V, E)$ be a graph (or digraph or multigraph). An *edge-partition* of G into k graphs $G_1, ..., G_k$ with $G_i = (V, E_i)$, where the edge set E is partitioned into k classes $E_1, ..., E_k$. An *edge-packing* of G by k graphs $G_1, ..., G_k$ is obtained by partitioning the edge-set E into $k+1$ classes $E_1, ..., E_k, U$ and defining $G_i = (V, E_i)$. The main observation here is straightforward:

Lemma 4.1 *If G has an edge-packing by k graphs $G_1, ..., G_k$, and Rel is any coherent reliability measure,*

$$Rel(G) \geq 1 - \prod_{i=1}^{k} (1 - Rel(G_i)).$$

An inequality results in Lemma 4.1 because there are operational states of G in which no G_i is operational. Some notes are in order on the effective use of Lemma 4.1 in obtaining lower bounds on reliability. Consider an edge-packing of G by $G_1, ..., G_k$. If any G_i is non-operational, coherence ensures that $Rel(G_i) = 0$; in this event, the inclusion of G_i in the edge-packing does not affect the bound, and G_i can be omitted. Thus we need only be concerned with edge-packings by *operational* subgraphs.

Our goal is to obtain efficiently computable bounds; hence, it is necessary that we compute (or at least bound) $Rel(G_i)$ for each G_i. One solution to this, suggested by Polesskii [316], is to edge-pack G with minpaths. The reliability of a minpath is easily computed. This suggests a solution in which we edge-pack G with as many minpaths as possible, and then apply Lemma 4.1; this basic strategy has been explored extensively.

It is essential to note that, while subgraph counting bounds require that edges have the same operation probability, no such assumption is needed here; one need only compute the probability of a minpath as the product of the edge operation probabilities over edges in the minpath. With this in mind, one might modify our edge-packing problem to require packing by the most reliable minpaths rather than by the largest number of minpaths.

It is also worth noting that any edge-packing by operational subgraphs $G_1, ..., G_k$ for which $Rel(G_i)$ is easily computed provides an efficiently computable lower bound. This leads to problems such as edge-packing by series-parallel graphs, or by partial k-trees for fixed k. This latter approach seems not to have been studied in the literature; hence, we concentrate on edge-packing by minpaths.

Polesskii [316] pioneered the use of edge-packing lower bounds, in the all-terminal reliability problem. Here an edge-packing by minpaths is a set of edge-disjoint spanning trees. Using a theorem of Tutte [405] and Nash-Williams [302], Polesskii observed that a c-edge-connected n-vertex graph has at least $\lfloor \frac{c}{2} \rfloor$ edge-disjoint spanning trees; hence when all edge operation probabilities are the same value p, the all-terminal reliability of the graph is at least $1-(1-p^{n-1})^{\lfloor c/2 \rfloor}$. When edge probabilities are not all the same, Polesskii's bound extends in a natural way. Using Edmonds's matroid partition algorithm [120, 122], a maximum cardinality set of edge-disjoint spanning trees, or its minimum cost analogue [84], can be found in polynomial time. Applying Lemma 4.1 then yields a lower bound on all-terminal reliability. Naturally, to obtain the best possible bound from Lemma 4.1, one wants not only a large number of edge-disjoint minpaths, but also minpaths that are themselves reliable. Edmonds's algorithm need not yield a set of spanning trees giving the best edge-packing bound using minpaths. In fact, the complexity of finding the set of spanning trees leading to the best edge-packing bound remains open.

Edge-packing as a general technique was pursued much later. Brecht and Colbourn [59] and Litvak and Ushakov [226, 270] independently developed edge-packing lower bounds for two-terminal reliability. For two-terminal reliability, minpaths are just s,t-paths. Menger's theorem [112, 286] asserts that the maximum number of edge-disjoint s,t-paths is the cardinality of a minimum s,t-cut. Thus using network flow techniques, a maximum edge-packing can be found [154, 124]. Here the problem of finding the best edge-packing, even when all edge operation probabilities are equal, is complicated by the fact that minpaths exhibit great variation in cardinality. In fact, Raman [336] has shown that finding the best edge-packing by s,t-paths is NP-hard. For this reason, heuristics have been examined to find "good" edge-packings. Brecht and Colbourn [59] examine the use of minimum cost network flow routines [166, 396] using edge cost $-\ln p_i$ on an edge of probability p_i, and report improvements over (general) edge-packings of maximum cardinality.

Turning to k-terminal reliability, the situation is not as satisfactory. Here a minpath is a subtree in which each leaf is a terminal, i.e. a *Steiner tree*. Colbourn [90] showed that determining the maximum number of Steiner trees in an edge-packing is NP-hard. No heuristics for finding "good" edge-packings by Steiner trees appear to have been studied. For directed networks, edge-packing (or more properly, arc-packing) bounds can be obtained using directed s,t-paths found by network flow techniques (for s,t-connectedness), and by using arc-disjoint rooted spanning arborescences (directed rooted spanning trees) found by Edmonds's branchings algorithm [123, 171, 281] (for reachability). See Ramanathan and Colbourn [337] for a discussion of the reachability bounds.

Until this point, we have examined lower bounds based on edge-packings by minpaths. Let us now turn to upper bounds. Not surprisingly, Lemma 4.1 has a "dual" form for upper bounds, interchanging the role of pathsets and cutsets:

Lemma 4.2 *Let $G = (V, E)$ be a graph (or digraph or multigraph). Let Rel be a coherent reliability measure. Let $C_1, \ldots, C_s$ be an edge-packing of G by cutsets. Then*

$$Rel(G) \leq \prod_{i=1}^{s} \left(1 - \prod_{e \in C_i} (1 - p_e)\right)$$

where p_e is the operation probability of edge e.

Lemma 4.2 gives an upper bound, since the failure of any cut in the edge-packing causes G to fail, but the failure of G can occur even when no cutset in the packing is failed.

Brecht and Colbourn [59] and Litvak and Ushakov [270] first studied edge-packing upper bounds for the two-terminal reliability problem. A theorem of Robacker [344, 168, 169] gives the necessary dual to Menger's theorem: the maximum number of edge-disjoint s,t-cuts is the length of a shortest s,t-path. Finding a maximum set of edge-disjoint cuts is straightforward — simply label each node with its distance from s. If t gets label ℓ, form cutset C_i containing all edges between vertices labeled $i-1$ and vertices labeled i, for $1 \leq i \leq \ell$. The result is ℓ edge-disjoint s,t-cuts. Finding a "good" set of cuts for the edge-packing upper bound appears to be more difficult than for the lower bound. Recently, Wagner [417] gave a polynomial time algorithm for finding a minimum cost set of edge-disjoint s,t-cutsets of maximum cardinality. Nel and Strayer [306] report that, while using Wagner's mincost algorithm improves in general upon the bounds from edge-packings found by the labeling method above, it is often not competitive with a simple greedy algorithm that repeatedly takes the least reliable cut disjoint from those chosen thus far.

Turning to upper bounds on all- and k-terminal reliability using edge-packings by mincuts, we encounter a major difficulty: even for all-terminal reliability, finding a maximum packing by mincuts is NP-hard [90]. Thus it is particularly surprising that by directing the reliability problems, we *are* able to find a maximum arc-packing by cutsets for the reachability problem using an efficient algorithm of Fulkerson's [121, 170]. Thus an all-terminal reliability upper bound can be obtained by using the *arc*-packing bound for reachability.

Two potential methods to improve the edge-packing strategy stand out. The first is to consider packings by more reliable subgraphs; the second is to extend the sets of pathsets and cutsets being examined to permit some edge intersection (thereby losing the independence of the sets of edges in the packing). We treat the second extension, which has been explored more extensively, to the next subsection. For the first, little work appears to have been done. Using the efficient exact algorithm for reachability of acyclic rooted directed graphs, Ramanathan and Colbourn [337] obtained improvements in reachability upper bounds, and also in all-terminal upper bounds. However, the use of edge-packings by nonminpaths or cutsets has not proceeded far, in part because of the scarcity of exact algorithms for restricted classes, and in part because of the difficulty of finding suitable edge-packings.

4.2.2 Noncrossing Cuts

The use of edge-disjoint pathsets and cutsets until this point is motivated primarily by the necessity to compute the probability that one of the pathsets operates (as in Lemma 4.1) or that one of the cutsets fail (as in Lemma 4.2). Lomonosov and Polesskii [277] devised a method that permits cutsets to share edges, while retaining an efficient method for computing the probability that one of the cutsets fails. For a graph $G = (V, E)$, a partition (A, B) of V forms a cutset, containing all edges having one end in A and the other in B. Two such cutsets (A, B) and $(\hat{A}, \hat{B})$ are *noncrossing* if at least one of $A \cap B$, $A \cap \hat{B}$, $\hat{A} \cap B$ and $\hat{A} \cap \hat{B}$ is empty. A collection of cuts is *noncrossing*, or *laminar*, if every two cutsets in the collection are noncrossing. In an n-node graph with k terminals, a set of noncrossing cutsets contains at most $n - 1 + k - 2 \leq 2n - 3$ noncrossing cuts [102].

A *cut basis* of an n-vertex graph is a set of $n - 1$ cuts $C_1, \ldots, C_{n-1}$ for which every cut can be written as the modulo 2 sum of these $n - 1$ cuts. Gomory and Hu [177] give an algorithm for finding a cut basis $C_1, \ldots, C_{n-1}$ in which $\sum_{i=1}^{n-1} |C_i|$ is minimum; moreover, their cut basis is a set of noncrossing cuts. Lomonosov and Polesskii [277] showed that for any cut basis $C_1, \ldots, C_{n-1}$, the all-terminal reliability satisfies

$$Rel(G) \leq \prod_{i=1}^{n-1} \left(1 - \prod_{e \in C_i} (1 - p_e)\right).$$

The use of cut bases for the k-terminal problem has been studied by Polesskii [318], generalizing the method outlined here. The restriction to a basis, however, limits the number of cuts that can be employed to one fewer than the number of terminals. A more general extension is obtained by permitting the use of sets of noncrossing cuts. Shanthikumar [364] used noncrossing ("consecutive") cuts in obtaining a two-terminal upper bound. This has been extended to k-terminal reliability (actually to s, T-connectedness) in [102]. The bound is obtained by establishing that the probability that none of the noncrossing cuts fail agrees with the k-terminal *nodal* reliability of a special type of graph, a *directed path graph*. A simple dynamic programming strategy then produces the bound in polynomial time.

Bounds using noncrossing cuts extend the edge-packing strategies essentially by considering a larger set of cuts, but still a polynomial number of them.

4.2.3 Transformation and Graph Approximation

We have thus far seen two methods for extending the edge-packing strategy: packing with non-minpaths or cutsets, and relaxing the edge-disjointness requirement. In this subsection, we examine a third extension that is perhaps less immediate than the previous two.

We have seen that transformations can be used to "simplify" a network, in order to reduce the time required in exact algorithms. Such transformations *preserve* the value of the reliability measure. Other transformations on networks may have the property that they guarantee *not to increase* the reliability measure; these *D-transformations* preserve lower bounds on the reliability measure (that is, computing a lower bound *after* applying such a transformation gives a lower bound on the reliability of the network *before* the transformation). Similarly, *I-transformations* guarantee *not to decrease* the reliability measure, and hence preserve upper bounds.

A trivial D-transformation is deleting an edge or arc in a network; it follows from coherence and statistical independence that the reliability measure cannot increase upon such a deletion. Similarly, the operation of *splitting* a node x into two nodes x_1 and x_2, and replacing each edge $\{y, x\}$ by *either* $\{y, x_1\}$ *or* $\{y, x_2\}$, we cannot increase the reliability. These trivial transformations have remarkable consequences. AboElFotoh and Colbourn [1] observe that the edge-packing lower bound for two-terminal reliability can be obtained by using just edge deletion and node splitting (delete all edges not on any path in the packing, and split non-terminals as necessary that are on more than one path of the packing). The result of these transformations is a parallel combination of s, t-paths — a very simple series–parallel graph. The edge-packing upper bound for two-terminal reliability is similar, using the I-transformation that identifies two nodes [1]. The use of transformations to obtain the two-terminal edge-packing bounds permits one to stop the transformation process "early". Once the network has been transformed into a series–parallel network, for example, the reliability can be calculated exactly in polynomial time and there is no need for further transformations. AboElFotoh and Colbourn [1] remark that the approach is very sensitive to the order and location in which the transformations are applied, and suggest some detailed heuristics for the transformations introduced so far.

Lomonosov [275] simplified the presentation of the Lomonosov–Polesskii upper bound that uses cut bases. He introduced an I-transformation, which we call the *Lomonosov join*. Let x, y, z be three nodes and let $\{x, y\}$ be an edge. The Lomonosov join removes edge $\{x, y\}$ and adds the edges $\{\{x, z\}, \{y, z\}\}$ each with the same operation probability as the deleted edge. Lomonosov proved that when x, y, z are all terminals, this cannot decrease the reliability, and Colbourn [92] showed that we only require that z is a terminal. This leads to upper bounds for all-terminal [67] and k-terminal [92] reliability. The use of transformations also permits a better bound than the Lomonosov-Polesskii bound to be obtained, by applying transformations only until the network is series-parallel.

A further I-transformation was studied by Lomonosov and Polesskii [279]. Given an arbitrary graph G, treat every nonadjacent pair of vertices as being connected by an edge of failure probability one; then G is essentially a complete graph. For any two vertices x, y, consider the adjacencies of x and y with the remaining vertices $\{v_1, \ldots, v_{n-2}\}$. Suppose that $\{x, v_i\}$ has failure probability q_i and that $\{y, v_i\}$ has failure probability ${q'}_i$. A transformation of the probabilities is carried out by setting the failure probabilities for both $\{x, v_i\}$ and $\{y, v_i\}$ to $\sqrt{q_i {q'}_i}$, for all $1 \leq i \leq n-2$. Lomonosov and Polesskii [279] show that this is an I-transformation, and by repeated application that the most reliable graph $G = (V, E)$ with $\prod_{e \in E} q_e = \theta$ on n vertices is the complete graph with each edge having failure probability $\theta^{-\binom{n}{2}}$.

A number of other transformations have been studied for their applications in reliability bounds [57, 67, 99, 317]. One particular application of transformations is in the analysis of blocking probabilities of "channel graphs" [78]. Transformations for channel graphs apply to s, t-connectedness as well [209, 242]. Leggett [261, 262] was one of the first to use a transformation-based approach in developing a bound, but his bounds are in error [198].

One pair of transformations, the delta-wye and wye-delta transformations, merit special mention. A delta (or Δ) in a network is a set of three edges $\{\{x, y\}, \{x, z\}, \{y, z\}\}$ forming a triangle, and a wye (or Y) is a set of three edges $\{\{x, w\}, \{y, w\}, \{z, w\}\}$ for which w is incident only to these three edges. Wye-delta and delta-wye transformations are just the replacement of one configuration by the other. In 1962, Lehman [263] provided two methods for determining probabilities on the edges of the wye (delta) given the edge probabilities on the delta (wye, respectively). His two methods are *not* exact in general, but rather he showed that one of the transformations is an I-transformation and the other is a D-transformation, provided the central node of the wye is not a terminal. Surprisingly, which of the two transformations is the I-transformation depends on the numerical probabilities. Thus the wye-delta and delta-wye transformations seem to differ from the earlier transformations mentioned, as there does not appear to be a simple combinatorial justification for the transformations.

Epifanov [131] subsequently showed that every planar network can be reduced to a single edge by repeated applications of wye-delta, delta-wye, series, parallel and degree-1 transformations; see also [138, 402]. This leads to remarkably accurate bounds for two-terminal reliability for planar networks [82]. See Traldi [400], Litvak [268] and Politof [319] for other delta-wye transformations.

Perhaps the most powerful aspect of developing bounds by composing simple transformations is the potential ease of extension to more complicated reliability measures. For example, whenever edge deletion and vertex splitting are I-transformations for a specified reliability or performance measure, we have efficiently computable edge-packing bounds; see [79, 270, 271] for some examples. If in addition the measure can be calculated exactly for series-parallel networks in polynomial time, we have efficient series-parallel approximations. Colbourn and Litvak [99] discuss more general measures of performance using this transformational approach. Using the Lomonosov join, the bounds for static measures of reliability discussed here can be extended in part to time-dependent reliability measures [100].

4.2.4 Miscellaneous Bounds

There are a number of further bounding techniques that have been explored which do not admit an easy classification as "edge-packing" or transformation-based bounds. Among efficiently computable bounds, the most notable is the ***k-cycle bound*** introduced by Lomonosov and Polesskii [278] and sharpened by Lomonosov [275]. Using a random graph model introduced by Erdös and Rényi [132, 133], Lomonosov [275] examined a graph evolution process. Suppose that at time 0 each edge is absent, but has an exponentially distributed time of arrival in the graph. What is the first time at which the graph becomes connected? Lomonosov established an equivalence between this graph evolution process and the static evaluation of all-terminal reliability, and by examining the expected time at which a transition is made from a network state with ℓ components to a state with $\ell - 1$ components (for $\ell = n, \ldots, 2$), he established a lower bound on all-terminal reliability. See [88, 275] for details.

Classical bounds due to Bonferroni (see [324, 368]) can be obtained using the inclusion-exclusion formula (1) of section 3.4. By truncating the sum after $\ell < h$ terms, an upper bound is obtained when ℓ is odd, and a lower bound is obtained when ℓ is even. The Bonferroni bounds require knowledge of all minpaths, an exponential quantity. Two–terminal bounds have been developed by Prékopa, Boros and Lih [324] that use "binomial moments" to improve upon the Bonferroni bounds.

Bounds have also been studied in the case that statistical dependence cannot be assumed. Hailperin [192] develops a general linear programming formulation for reliability measures when the worst possible dependencies are permitted. Efficient implementations of Hailperin's method have been developed for two-terminal reliability by Zemel [435] and Assous [21], and for all-terminal reliability by Carrasco and Colbourn [80]. Under worst case assumptions about statistical dependencies, however, the bounds appear to have little or no practical import unless the information about dependencies specified is substantial.

Finally, there is an extensive literature on bounds that require exponential time in the worst case to compute. We have focussed on efficient methods, so do not give a complete survey of exponential time methods here. Undoubtedly the most influential method is due to Esary and Proschan [134]. They observed that if one examines *all* minpaths in the network, and computes the probability that at least one path fails ***under the assumption that paths are independent***, one obtains an upper bound on the reliability. This is a remarkable contrast to the edge-packing strategy, where the same technique was applied to a subset of all paths, but a *lower* bound was obtained. Esary and Proschan [134] also prove the dual statement to obtain a lower bound from all cuts. At the present time, no algorithm is known to compute the Esary-Proschan bounds, or to improve upon them, in polynomial time, except for upper bounds on s,t-connectedness [94]. This is not to say, however, that they are typically more accurate than the efficiently computable bounds; our experience suggests the contrary.

A further recent direction to obtain bounds is to examine a limited subset of all states, and to compute a bound based upon the states examined. By concentrating on most probable states, one expects that a small fraction of all states need to be examined in order to see most of the contribution to the reliability. Shier [367] gives an excellent introduction to this subject; see also [257, 371, 431, 432]. While accuracy/time tradeoffs are observed empirically here, there appears to be no guarantee that a prescribed accuracy can be achieved in polynomial time.

Along the same lines, Nel and Colbourn [304] observe that one can apply factoring a limited number of times, and apply any or all of the bounding techniques discussed earlier. If the number of edges factored on is bounded by $\log n$, where n is the number of vertices, the process remains polynomial in time — but one expects improved accuracy. Of course, the notions of most probable states and efficiently computable bounds can be combined; see [304] for some steps in this direction.

4.2.5 Postoptimization on Bounds

So far we have discussed basic strategies for obtaining bounds. Even the thumbnail description of each cannot fail to convince one that there is great variety in the available bounds. It is natural to combine the bounds to obtain better, or more general, bounds. The preferred way is to find a general theory in which a number of bounds are unified. Failing that, one wants at least to deduce whatever information about reliability is possible from the many different bounds provided. For example, if one knows the probability of reaching u from s, and independently the probability of reaching t from u, what can be said about the probability of reaching t from s? Using a remarkable theorem of Ahlswede and Daykin [14], Brecht and Colbourn [60, 61] develop methods for improving lower bounds. They observe that if an network G is connected for terminal set K_1 with probability p_1, connected for terminal set K_2 with probability p_2, and $K_1 \cap K_2 \neq \emptyset$, then G is connected for terminal set $K_1 \cup K_2$ with probability at least $p_1 p_2$. This gives a multiplicative triangle inequality for two-terminal reliability, that Brecht and Colbourn [61] found to be effective in improving upon two-terminal reliability bounds that were computed by other methods (edge-packing in particular). The key here is the *post*optimization, techniques to improve upon arbitrary bounds.

A somewhat analogous method for upper bounds, called *renormalization*, has been studied recently [199]. For two-terminal reliability, the probability x_e that one terminal, s, cannot reach the other terminal, t, through a specified edge e is bounded above by the probability that edge e itself fails plus the probability that e operates times the computed probability x_f for every edge f incident to e. Two-terminal upper bounds can be used to determine initial estimates of the probability x_e for each edge e. Then each inequality may force the reduction of some x_e value. Renormalization obtains an upper bound by underestimating the effect of intersections of s,t-paths, and by examining s,t-walks rather than just s,t-paths. For s,t-connectedness of acyclic directed graphs, the lack of directed cycles ensures that all s,t-walks are s,t-paths; in this case, renormalization is a polynomial time method that guarantees an improvement on the Esary-Proschan upper bound [94]. Renormalization is essentially a postoptimization strategy, but can be used by itself commencing with initial overestimates on x_e of the edge failure probability of e.

One final postoptimization strategy appears to apply only when all edge operation probabilities are equal. Nevertheless, we outline the idea here. Colbourn and Harms [98] observe that if one evaluates the polynomial $\sum_{i=0}^{d} F_i p^{m-i}(1-p)^i$ at a fixed value for p, one obtains a linear combination of $F_0, \ldots, F_d$. Hence, if one knows an upper or lower bound on the value of the reliability polynomial at a specified value of p, one obtains a linear constraint. Thus any bounding method whatsoever, when applied to a network with all edge operation probabilities equal, yields a linear constraint of this form. All of the linear inequalities so produced are met simultaneously, and hence one can combine bounds of all different sorts using linear programming. If all the basic bounds used are efficiently computable, one may use a polynomial time algorithm for linear programming [222, 235] to retain polynomial running time overall. Colbourn and Harms [98] note that the linear programming bound so obtained occasionally improves upon all of the basic bounds used to supply constraints.

5 Monte Carlo methods

Due to the extreme intractability of exact computation of the various reliability measures covered in this paper, and to the inability of polynomial-time bounding algorithms to provide very tight bounds on these measures, it is necessary to turn to *simulation* techniques in order to obtain accurate estimates. This, of course, comes at a price: the estimates obtained have a certain degree of uncertainty. Nevertheless, this price is typically well justified by the superior results given by simulation methods over deterministic techniques. Due to the relatively simple structure of these problems it is natural to use the powerful and well-studied *Monte Carlo* method of simulating the stochastic behavior of the system. The first use of Monte Carlo methods in network reliability seems to have occurred in the context of percolation problems by Frisch, Hammersley and Welsh [164], with early work in [109, 193], [194], and [264]. Most of the significant current techniques, however, were developed within the past decade or so.

5.1 Naive sampling

We first establish some notation to be used throughout the section. Let $(\Phi, \boldsymbol{p})$ be an instance of a particular reliability

problem with $\boldsymbol{p} = (p_1, \ldots, p_m)$ the vector of component operating probabilities. Let $\boldsymbol{q} = (q_1, \ldots, q_m) = (1-p_1, \ldots, 1-p_m)$ be the vector o probabilities, and denote by $P(x)$ the probability that a particular state vector $\boldsymbol{x}$ appears, that is,

$$P(x) = \prod_{x_e=1} p_e \prod_{x_e=0} q_e.$$

We are interested in obtaining an estimate $\hat{R}$ for the true system reliability $R = \Pr[\Phi = 1]$

The naive method of sampling is fairly straightforward. A sample of K vectors $\boldsymbol{x}^k = (x_1^k, \ldots, x_m^k)$, $k = 1, \ldots, k$ is drawn from the distribution P, by drawing mK independent samples $\hat{U}_{kj}$, $k = 1, \ldots, K$, $j = 1, \ldots, m$ from a uniform random number generator and then setting

$$x_j^k = \begin{cases} 1 & \hat{U}_{kj} \le p_j \\ 0 & \hat{U}_{kj} > p_j \end{cases} \qquad k = 1, \ldots, K, \ j = 1, \ldots, m.$$

Let $\hat{K}$ be the number of vectors $\boldsymbol{x}^k$ for which $\Phi(\boldsymbol{x}^k) = 1$. Then an unbiased estimator for R is $\hat{R} = \hat{K}/K$, and its variance is bounded above by $R(1-R)/K$. Reduction of this variance can be obtained by a number of standard Monte Carlo sampling techniques, such as *antithetic* and *control variates* and *conditional, importance*, and *stratified sampling.* Since these techniques belong more in the area of probability theory than network theory, we refer the reader to a Monte Carlo text such as [194] for their treatment.

We do wish to review some of the major techniques which have been applied to network reliability problems. An excellent treatment of the first four of these schemes is found in Fishman [141], and we refer the reader to that paper for further details.

5.2 Dagger Sampling

Dagger sampling was developed by Kumamoto *et al.* [255], and can be thought of as an "m-dimensional" extension of antithetic sampling. The idea — which is common to several Monte Carlo techniques — is to "spread out" the individual edge failures in such a way that repeats of sample states is minimized. The procedure is given below.

Dagger Sampling Method

1. Let $(N_e : e \in E)$ be a vector of integers chosen in direct proportion to the (rational) q_e's.
2. Choose sample size K^* so that for each edge e the sequence of K^* replications can be broken into exactly N_e *subblocks* of size K^*/N_e.
3. For each edge e, choose at random exactly one replication in each of the N_e replication subblocks for that edge in which that edge fails. This gives a failure pattern for the K^* replications in which the frequency of failures of each edge is exactly in proportion to the average failure rate of that edge.
4. Make a final pass through the K^* replications, computing the average proportion of replications corresponding to system operation. This is an unbiased estimator of R.

5.3 Sequential Construction/Destruction

The Sequential Construction/Destruction Method of Easton and Wong [119] — and later improved by Fishman [141] and Elperin *et al.* [130] — is based on considering an *ordering* of the edges of the graph. The edges begin as all failed, and then edges are successively "repaired" — i.e. caused to operate — one by one in the specified ordering, until the system becomes operational. The reliability estimate is then a function of how long it takes for the system to become operational. This can result in better estimates than could be obtained by the naive method.

The sample space for the sequential construction method consists of a *pair* (x, π), where x is a state vector for the system, and $\pi = (\pi(1), \ldots, \pi(m))$ is a *permutation* of the edge indices of E such that for some index k we have

$$x_{\pi(1)} = \ldots = x_{\pi(k)} = 1, \quad x_{\pi(k+1)}, \ldots, x_{\pi(m)} = 0.$$

If the state vector x is chosen according to the prescribed state probabilities, and the permutation π is chosen independently and uniformly over all matching permutations, then the probability of a particular pair (x, π) occurring is

$$\begin{aligned} \rho(x,\pi) &= \frac{P(x)}{k!(m-k)!} \\ &= \frac{1}{m!}\binom{m}{k}P(x) \end{aligned}$$

where k is the number of operating elements in x. The sequential construction method samples a permutation $\hat{\pi}$, and considers simultaneously the collection $\mathcal{P}_{\hat{\pi}}$ of *all* possible state pairs (x, π) with $\pi = \hat{\pi}$ and x consistent with $\hat{\pi}$ according to the above criterion. The sample reliability value $\hat{R}$ for this set is then the *conditional* probability of system operation with respect to $\mathcal{P}_{\hat{\pi}}$, that is, the ratio of the sum probabilities of the pairs $(x, \pi) \in \mathcal{P}_{\hat{\pi}}$ for which $\Phi(x) = 1$ divided by the probability of $\mathcal{P}_{\hat{\pi}}$. The details are given below.

Sequential Construction Method

1. Choose a sample permutation $\hat{\pi} = (\hat{\pi}(1), \ldots, \hat{\pi}(m))$ over the set of permutations of $\{1, \ldots, m\}$. Define the vectors $x^{(k)}$, $k = 1, \ldots, m$ by

$$x^{(k)}_{\hat{\pi}(1)} = \ldots = x^{(k)}_{\hat{\pi}(k)} = 1, \qquad x^{(k)}_{\hat{\pi}(k+1)} = \ldots = x^{(k)}_{\hat{\pi}(m)} = 0.$$

2. Determine the first index $r = 0, \ldots, m$ for which $\Phi(x^{(r)}) = 1$.

3. The contribution $\hat{R}$ to the estimator of R is now

$$\begin{aligned} \hat{R} &= \frac{\sum_{k=1}^{m} \Phi(x^{(k)})\rho(x^{(k)}, \hat{\pi})}{\sum_{k=1}^{m} \rho(x^{(k)}, \hat{\pi})} \\ &= \frac{\sum_{k=r}^{m} \binom{m}{k} P(x^{(k)})}{\sum_{k=1}^{m} \binom{m}{k} P(x^{(k)})} \end{aligned}$$

4. Accumulate the set of $\hat{R}$ values, and divide by the number of sample permutations chosen. This is an unbiased estimator of R.

The estimator obtained for each sample permutation chosen has smaller variance than that obtained in one sample of the naive algorithm. The main computational effort occurs in Step 2, and depends critically on how fast one can update the value of $\Phi(x^{(k)})$, that is, how easily one can determine system operation as the edges are repaired one by one. Notice, however, that the edge repair needs to be performed only until the point at which the system operates for (assuming coherence of the system) further edge repairs do not change the operational state of the system. Thus the amount of work done may be considerably less than the order of m. In the case of connectivity reliability, moreover, Fishman [141] has shown that the determination of the index r can be done almost as easily as the determination of $\Phi(x)$ for one value of x, so that the sequential samples come at about the same cost as a single sample in the naive method. Finally, with equal edge failures we have the added advantage that the denominator in the expression in Step 4 above is always 1, and so an extra computational step can be saved.

One can develop a sequential *destruction* method analogous to the sequential construction method given above by starting with all components *operating* and sequentially "destroying" edges until the system fails. This may be advantageous in the situation where the system tends to fail after relatively few edges fail, so that fewer destruction iterations are performed than construction iterations in the reverse process.

5.4 Sampling Using Bounds

This is a powerful hybrid of the classical importance sampling and control variate schemes in Monte Carlo. It was first used to solve network reliability problems by Van Slyke and Frank [412], and expanded upon by Kumamoto *et al.* [254] and later Fishman [141, 142, 145]. It can in principle be applied to any reliability problem where the system function Φ has associated with it a *lower bounding* function Φ^L and and *upper bounding* function Φ^U having the properties

- $\Phi^L(x) \leq \Phi(x) \leq \Phi^U(x)$ for every state vector x
- For $k = 0, \ldots, m$ and any assignment $\hat{x}^{(k)} = (\hat{x}_1, \ldots, \hat{x}_k)$ of values for the first k components of x, the values
$$R_k^L(x^{(k)}) \equiv \Pr[\Phi^L = 1 |\, x_1 = \hat{x}_1, \ldots, x_k = \hat{x}_k]$$
and
$$R_u^U(x^{(k)}) \equiv \Pr[\Phi^U = 1 |\, x_1 = \hat{x}_1, \ldots, x_k = \hat{x}_k]$$
can be computed in polynomial time.

The values $R_0^L = \Pr[\Phi^L = 1]$ and $R_0^U = \Pr[\Phi^U = 1]$ are the unconditional operating probabilities for the bounding functions Φ^U and Φ^L, and typically have the most straightforward reliability computation algorithms. For connectivity measures on undirected graphs, however, the values of R_k^L and R_k^U can be obtained by computing the R_0^L and R_0^U values on the graph obtained by *deleting* all edges e_k for which $\hat{x}_k = 0$, and *contracting* all edges e_k for which $\hat{x}_k = 1$. Thus computation of these values is usually no more difficult than computing the unconditional reliabilities.

The values R_0^L and $1 - R_0^U$ represent easily computable measures of events in which the structure function Φ has known values. What bounds-based sampling does is to draw samples from the *remaining* space
$$\mathbf{X} = \{x \in \{0,1\}^E : \Phi^L(x) = 0,\ \Phi^U(x) = 1\}$$
in proportion to their probability in the original space. The (known) probability from the unsampled space — that is, where $\Phi^L(x) = 1$ or $\Phi^U(x) = 0$ — is then factored back into the conditional

reliability estimate obtained for sampled space $\mathbf{X}$ to obtain an estimate for the required measure R. The improvement obtained is then in direct proportion to the fraction of the original probability which is left to sample in $\mathbf{X}$. The associated Monte Carlo scheme is given below.

Bounds-based Sampling Method

1. Take samples $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$ from the space $\mathbf{X}$ by successively drawing, for $k = 1, \ldots, m$, the component state $\hat{x}_k$ with operating probability

$$\begin{aligned}\hat{p}_{e_k} &= \Pr[x_k = 1 |\, x_1 = \hat{x}_1, \ldots, x_{k-1} = \hat{x}_{k-1} \textit{ and } \Phi^U(x) = 1,\ \Phi^L(x) = 0] \\ &= \left[\frac{R_k^U(x^{(k-1)}, 1) - R_k^L(x^{(k-1)}, 1)}{R_{k-1}^U(x^{(k-1)}) - R_{k-1}^L(x^{(k-1)})}\right] p_{e_k}\end{aligned}$$

2. Compute the proportion $\hat{R}$ of those samples for which $\Phi(x) = 1$. The number $R_0^L + \hat{R}(R_0^U - R_0^L)$ is now an unbiased estimator of R

A simple example of this scheme, investigated in in [141, 412], uses the fact that for any state vector x, at least ρ elements must be operating in order for the system to operate, and at least γ elements must be failed in order for the system to fail, where ρ is the minimum cardinality of a pathset for the system and γ is a minimum cardinality cutset for the system. Then we define $\Phi^U(x)$ to be equal to equal to 1 when at least ρ elements of x are operating and 0 otherwise, and we define $\Phi^L(x)$ to be equal to equal to 0 when at least γ elements of x are failed, and 1 otherwise. The evaluations of R^L and R^U are then k-out-of-m reliability problems, which are known to have efficient probability computation algorithms, and the sample space $\mathbf{X}$ is just the space of state vectors x having at least ρ — but not more that $m - \gamma$ — elements operating. In [141] the disjoint pathset-cutset bounds given in parts 4.2.1 are exploited to give a stronger bounded Monte Carlo sampling scheme. Here if $C_1, \ldots, C_r$ are disjoint cuts and $P_1, \ldots, P_s$ are the disjoint paths, then

$$\begin{aligned}\Phi^L(x) &= \begin{cases} 0 \text{ if } \textit{any} \text{ of the cutsets } C_1, \ldots, C_r \text{ fail} \\ 1 \text{ otherwise}\end{cases} \\ \Phi^U(x) &= \begin{cases} 1 \text{ if } \textit{any} \text{ of the pathsets } P_1, \ldots, P_s \text{ operate} \\ 0 \text{ otherwise}\end{cases}\end{aligned}$$

The values R_0^L and R_0^U can be computed as in part 4.2.1, and this extends to the computation of $R_k^L(x^{(k)})$ and $R_k^U(x^{(k)})$ as well.

5.5 The Coverage Method

The accuracy of the Monte Carlo schemes given above are generally measured by the *variance* of the estimator $\hat{R}$. The variance estimate in each case turns out to be roughly of the form α/K, where K

is the number of samples and α is some constant which depends on R and on the type of sampling being done. Thus a rough analytic comparison between these estimators can be made based on the relative values of α. The actual variance depends on many other factors, and with the rate of decrease linear in K the time to sample becomes critical. Thus comparisons really need to be made based on an empirical basis.

The coverage method developed by Karp and Luby [224], is based on a more demanding criterion of effectiveness of a Monte Carlo scheme, and thus is able to obtain a correspondingly stronger statement concerning the behavior of their method. Specifically, let ϵ and δ be positive scalars. Suppose we are interested in computing some reliability measure value R and let $\hat{R}$ be the outcome from some Monte Carlo scheme for estimating R. Then the estimate $\hat{R}$ is an ***ϵ-δ approximation*** for R if

$$\Pr[\frac{|R - \hat{R}|}{R} > \epsilon] \leq \delta.$$

A Monte Carlo scheme is called a ***fully polynomial randomized approximation scheme*** (*FPRAS*) if in addition, the time to obtain the estimate $\hat{R}$ is of the order of ϵ^{-1}, $\log(\delta^{-1})$, and the size of the problem instance. In rough terms, a *FPRAS* is an algorithm which efficiently produces an estimate of R whose percentage error can be guaranteed to be sufficiently small with high probability.

The Karp-Luby Monte Carlo scheme is actually a hybrid/variant of the importance and stratified sampling methods, and makes use of the ***mincuts*** of the system to improve on the naive sampling scheme. To be consistent with the Karp-Luby paper we consider the computation of $R = \Pr[\Phi = 0]$, i.e. the probability of system failure, although one can develop an analogous scheme from the viewpoint of system operation as well. The idea is to embed the set F of failure events into a ***universal weighted space*** $(\mathcal{U}, w)$ — where w is a nonnegative weight function on the elements of $\mathcal{U}$ — which satisfies the following criteria:

- $w(F) = \Pr(F) = R$
- $w(\mathcal{U})$ is efficiently — i.e., polynomial-time — computable; further, samples can be efficiently drawn from $\mathcal{U}$ with probability proportional to their weight
- It can be efficiently recognized when an element in $\mathcal{U}$ is also in F
- $w(\mathcal{U})/w(F)$ is bounded above by some value M for all instances in the problem class

It is clear then that if any sample is drawn from $\mathcal{U}$, and the estimate $\hat{R}$ is produced by multiplying the proportion of this sample which is contained in F by $w(\mathcal{U})$, then $\hat{R}$ is an unbiased estimator of R. In [224] it is further established that for any positive scalars ϵ and δ, if the sample size is at least $M \ln(2/\delta)4.5/\epsilon^2$, then the resulting estimator $\hat{R}$ is an ϵ-δ approximation for R. In other words, this sampling scheme is a *FPRAS*.

We now describe the coverage method as it applies to the (s,t)-connectedness reliability problem, although the same techniques can be applied in a wide range of situations. Let (G, s, t, p) be an instance of the (s,t)-connectedness reliability problem, and let $\mathcal{C}$ be the collection of minimal (s,t)-cuts for G. Define the universal weighted space $\mathcal{U}$ to consist of the pairs (x, C) with x a state vector, $C \in \mathcal{C}$, and $x_e = 0$ for all $e \in C$. The weight assigned to each pair (x, C) is simply $P(x)$. Now each failure state x of the system appears in the elements of $\mathcal{U}$ as many times as the number of mincuts on which x fails; in order to embed F in $\mathcal{U}$, it is necessary to assign to each x a ***unique*** $C \in \mathcal{C}$. In the (s,t)-connectedness problem this is done by finding the set of elements which can be reached from s by a path of operating edges (with respect to x) and setting $C \equiv C(x)$ to be the set of edges from X to $V \setminus X$. The elements of F now appear in $\mathcal{U}$ as (x, C) such that $C = C(x)$, and an element of $\mathcal{U}$ can be determined in linear time to correspond to an element of F by checking the condition $C = C(x)$. The coverage method for the (s,t)-connectedness problem is given below.

Coverage Method

1. Determine the collection $\mathcal{C}$ of (s,t)-cutsets of G. For each $C \in \mathcal{C}$ compute $w(C) = \prod_{e \in C} q_e$ = the total weight of all elements of $\mathcal{U}$ with second component equal to C, and then compute $w(\mathcal{U}) = \sum_{C \in \mathcal{C}} w(C)$.

2. Draw elements (x, C) from $\mathcal{U}$ in proportion to their weights by first drawing a C from $\mathcal{C}$ with probability $w(C)/w(\mathcal{U})$ and then drawing x by setting $x_e = 0$, $e \in C$, and assigning the other components of x according to their original component probabilities.

3. Compute the proportion $\hat{K}$ of times that a sample (x, C) has $C = C(x)$. Then $\hat{R} = \hat{K} w(\mathcal{U})$ is an unbiased estimator for R.

The above scheme is ***not*** a *FPRAS*, for two reasons. First, it is necessary to enumerate the entire set $\mathcal{C}$ of mincuts, and the cardinality of this set generally grows exponentially in the size of the problem (in fact [331] gives a method of computing R exactly from this list). Second, the boundedness condition for $w(\mathcal{U})/w(F)$ is not satisfied for general instances of the problem. Karp and Luby, however, go on to modify the above procedure for the class if (s,t)-connectedness reliability problems where the graph G is planar and undirected, has its facial boundaries bounded in cardinality and sum probability, and satisfies the condition that $\prod_{e \in E}(1 + q_e)$ is bounded above by some fixed M. We do not go into the details here; the general idea is to expand $\mathcal{C}$ to include cuts which are "almost minimal", in such a way that the associated space $\mathcal{U}$ defined above satisfies the required properties with respect to F. The planarity of G is then employed to allow elements of the expanded space $\mathcal{U}$ to be sampled efficiency and with the correct probabilities, so that the modified scheme becomes an FPRAS for (s,t)-connectedness reliability.

5.6 Estimating the Coefficients of the Reliability Polynomial

One problem with the methods given thus far is that they only estimate the reliability for a single vector $\boldsymbol{p}$ of probabilities. Of greater interest, frequently, is some estimate of the ***functional form*** of the reliability polynomial. This makes the most sense in the case when the edge failure probabilities are all the same probability p, so that the system reliability can be written — as indicated in Section 2 — in one of two polynomial forms:

$$\begin{aligned} Rel(p) &= \sum_{i=0}^{m} F_i \, p^{m-i}(1-p)^i \\ &= p^l \sum_{i=0}^{m} H_i (1-p)^i \end{aligned}$$

In this case a more useful Monte Carlo scheme would be one that estimated each of the coefficients F_i or H_i, for then one could use these estimates to derive an estimate of reliability for any desired value of p.

Two papers have dealt specifically with computing the coefficients of the reliability polynomial. The work of Van Slyke and Frank [412] and Fishman [143] concerns the F_i-coefficients for k-terminal reliability. Van Slyke and Frank uses standard stratified sampling to estimate the F_i values, by sampling separately states having exactly i operating components. Fishman improves this by extending the sequential construction method given in part 7.3. He actually estimates the values

$$\begin{aligned} \mu_i &= \frac{F_i}{\binom{m}{i}} \\ &= \Pr[\text{the system operates } \textit{given } i \text{ elements fail}] \end{aligned}$$

by noting that an unbiased estimator for the differences $\mu_i - \mu_{i-1}$ is simply the proportion of times that the index r obtained in Step 2 of the Sequential Construction Method is equal to i. An unbiased estimator for μ_k — and hence F_k — can then obtained by summing the appropriate difference estimators.

Nel and Colbourn [305] investigate the all-terminal reliability problem, and provide a scheme for estimating the H_i coefficients for this problem. Since the sum of the H_i coefficients is equal to the number of *spanning trees* in the graph G — as opposed to the number of *connected sets* of G, which is the sum of the F_i coefficients — then the number of states contributing to the estimators of the H_i coefficients is much smaller than those which need to be sampled to estimate the F_i coefficients. Referring to part 4.1.5, let $\mathcal{U} = \{[L_i, U_i] | \, i = 1, \ldots, b\}$ be any shelling of the $\mathcal{F}$-complex of G. From the definition of H_i as the number of L_j's of cardinality i, it follows that for any uniform sampling of intervals $[L_j, U_j]$ in $\mathcal{U}$, the proportion of L_j's of cardinality i is an unbiased estimator of H_i. Nel and Colbourn go on to give a technique for sampling uniformly from the collection of intervals of a "canonical" shelling of the $\mathcal{F}$-complex of G, based on a uniform sampling of spanning trees in G [16, 62, 93].

Describing the reliability function when general edge failures are present is problematic, since the polynomial form itself has an exponentially large number of terms. Fishman [145], however, develops a method for partially describing the reliability function by giving the system reliability as a function of a small number of component reliabilities. In particular, suppose that we are interested in knowing the reliability R as a function of the operating probabilities $p_1, \ldots, p_k$ of a chosen set of k edges $e_1, \ldots, e_k$, given *specific* operating probabilities $\tilde{p}_{k+1}, \ldots, \tilde{p}_m$ for the remaining edges $e_{k+1}, \ldots, e_m$. Then we compute the "coefficients" of the partial description of the reliability by performing a variant of stratified sampling (or conditional sampling, depending on the viewpoint). The procedure is as follows: for ***each state vector*** $\hat{x}^{(k)} = (\hat{x}_1, \ldots, \hat{x}_k)$ on the edges $e_1, \ldots, e_k$, sample the ***strata*** of states where edge $x_i = \hat{x}_i$, $i = 1, \ldots, k$ and the remaining edges operate according to their given probabilities. We then compute an estimate $\hat{R}(\hat{x}^{(k)})$ for the associated reliability. When edges operate independently, the strata sampling is fairly straightforward, and can frequently make use of the other improvement schemes given earlier in this section. An estimate for the required functional form for R can now be written in the form

$$\hat{R} = \sum_{\hat{x}^{(k)} \in \{0,1\}^k} \prod_{i:\, \hat{x}_i = 1} p_i \prod_{i:\, \hat{x}_i = 0} q_i \, \hat{R}(\hat{x}^{(k)})$$

As well as its descriptive value, this functional form is useful in measuring the "criticality" of the edges on which the function is defined, by testing the derivative effects on the function of changing a particular component reliability. Although criticality measures have drawn a significant amount of attention in reliability theory, their treatment is beyond the scope of this chapter.

It is apparent from our discussion here that Monte Carlo methods have been explored largely independently of the development of bounds; however, we emphasize that bounds and Monte Carlo approaches appear to operate most effectively when used in conjunction.

6 Performability Analysis and Stochastic Flows and Path Lengths

The previous four chapters have been concerned with connectivity measures. In the context of communications networks, the underlying assumption of these measures is that as long as a path exists between a pair of nodes then satisfactory communication can take place between those nodes. In many practical problem settings this is not the case. Specifically, issues such as delay, path-length, capacity, and the like can be of vital importance: the network must not just be connected, but it must function at an acceptable performance level. This viewpoint has led to research on performability measures. To study measures of this type additional information, such as arc lengths and arc capacities, are associated with the network components. In addition, it is possible that information representing the load on the system must be specified, e.g. a set of origin-destination traffic requirements. In general, the assessment of such information changes the nature of the reliability problem from a binary "operate-fail" type of probabilistic statement to one involving multiple system or component states. In many cases this simply results in a more complex variant of the binary-state problem, but it also includes problems involving average behavior and/or continuous state and system variables, which require substantially different solution techniques. We refer to this more general type of reliability problem as a multistate problem and intend it to include performability measures as well as other measures.

The general format for the multistate network problems considered in this paper is as follows: We are given a network $G = (V, E)$, together with a set of *random variables* $\{X_e : e \in E\}$ associated with the edges of the network. The value assigned to an edge random variable represents a parameter such as path length, capacity, or delay. In most of this section we will assume that each edge random variable can take q discrete states. In many situations, the case where $q = 2$ (two state systems) provides a realistic model. Here, it is common to refer to a "good" and a "bad" state, where in the "good" state the arc operates with a specified capacity, length, etc. and in the "bad" state the arc fails and has zero capacity, infinite length, etc. (Random variables may also be assigned to vertices of the network to represent demand, throughput, or delays at the vertex itself. We will not touch upon those models here, except to mention that they can often be modelled as problems with stochastic edge parameters.) Corresponding to any vector $\bar{x} = (X_e : e \in E)$ of assignments for the edge parameter random variables the system itself is given a value $\Phi(\bar{x})$, which represents some measure of system performance. Thus, the system state value is also be represented by a random variable, whose distribution is some complex function of the distribution of the individual parameter values. The goal of a multistate system evaluation problem is to compute, or estimate, some characteristic of the random variable representing the system state. This could range from a complete description of the system state distribution, to the probability that a certain threshold of system performance has been attained, or to the mean, variance, or selected moments of the system state distribution.

In the context of performability analysis we interpret $\Phi(\bar{x})$ as a measure of network performance when the network is in state $\bar{x}$. Typical performance measures include expected lost call traffic for circuit switched networks and expected packet or message delay for packet switched networks. To evaluate these measures it is necessary to invoke some variant of a multi-commodity flow algorithm. The related performability measures are the expected value of Φ and the probability that Φ is greater than or equal to or less than or equal to some threshold.

We start this chapter by describing the most probable states method, a simple general purpose method for computing upper and lower bounds on almost any reliability measure. This technique is frequently used for performability analysis. The remaining subsections give a more in-depth analysis of the work in three areas: shortest paths, maximum flows and PERT networks. We view these problems as the simplest instances within three important classes of multistate reliability problems. Our point of view is to illustrate how the extensive work on connectivity measures can be adapted to the multistate context. We also feel that this work can serve as the basis for analysis of more complex multistate measures.

6.1 The Most Probable States Method

The most probable states method is a bounding procedure that can be applied to very general classes of multistate problem [265, 257, 431, 432]. The only requirement is an efficient method for evaluating Φ. We describe its application to the performability measure $\Pr[\Phi \leq \alpha]$. The application to $E[\Phi]$ follows in a similar manner. Suppose that the network states are ordered $\bar{x}_1, \ldots, \bar{x}_s$ where $s = q^n$, such that $\Pr[\bar{x}_1] \geq \Pr[\bar{x}_2] \ldots$ The most probable states method is based on enumerating states according to this order. The importance of using this order is that the process can be terminated early with a good bound. Define $lp_\Phi(k)$ and $up_\Phi(k)$ to be any lower and upper bounds, respectively, on $\sum_{\Phi(\bar{x})\leq\alpha}\prod_{j=k}^{s}\Pr[\bar{x}_j]$. The upper and lower bounds typically used here are easily computable and, in fact in most cases, are trivial bounds that are independent of k. For 2-state systems if p_e is the probability of the "good" state for edge e and q_e the probability of the "bad" state for edge e, typical assumptions are that $p_e \geq q_e$ and as a result $\Phi(\bar{x}_1) \geq \Phi(\bar{x}_i \geq \Phi(\bar{x}_{2^n})$ so that we can set $lp_\Phi(k) = \Phi(\bar{x}_{2^n})$ and $up_\Phi(k) = \Phi(\bar{x}_1)$ for all k. The most probable states bounds are defined by,

$$\begin{aligned} LP_\Phi &= \sum_{k=1}^{\bar{k}} \Phi(\bar{x}_k)\Pr[\bar{x}_k] + (1 - \sum_{k=1}^{\bar{k}} \Pr[\bar{x}_k])lp_\Phi(\bar{k}+1) \\ UP_\Phi &= \sum_{k=1}^{\bar{k}} \Phi(\bar{x}_k)\Pr[\bar{x}_k] + (1 - \sum_{k=1}^{\bar{k}} \Pr[\bar{x}_k])up_\Phi(\bar{k}+1) \end{aligned}$$

Here, $\bar{k}$ can be defined dynamically based on some stopping criteria. The most typical criteria is to require that the difference between the upper and lower bounds be within some tolerance. Lower and upper bounds for the expected value measures can be defined in a similar way.

Yang and Kubat [431] describe a method for enumerating states, i.e. $\bar{x}_i$, in order of decreasing probability for 2-state systems in O(n) time per state. Specifically, they maintain a partial binary enumeration tree where each node represents the assignment of a "good" or "bad" state to each edge in some set S. The branching step is to choose an edge j, not in S, and create two new nodes, one with j assigned the "good" state and one with j assigned the "bad" state. At each iteration of the algorithm a new leaf node is created with corresponds to a (complete) network state $\bar{x}_i$. In order to generate these leaf nodes in the correct order two values are associated with each node in the enumeration tree: the probability of the highest probability leaf node, not yet enumerated, in the left sub-tree rooted at the node and the corresponding value for the right sub-tree. These values allow the algorithm to choose the appropriate leaf node to generate at each iteration in O(n) time.

Sanso and Soumis [352] suggest that rather than most probable states it is many time more appropriate to enumerate the "most important" states. The motivation is that in some situations certain lower probability states, which might not otherwise be enumerated, could significantly affect performance measures. Such states might correspond to situations in which the system exhibits extremely poor performance. Specifically, $\Pr[\bar{x}]$ might be relatively small but $\Phi(\bar{x})$ could be very large or very small. This is particularly relevant when computing bounds on $E[\Phi]$.

6.2 Elementary Properties of Multistate Systems

The remainder of this section will be devoted to an analysis of the following three multistate problems.

Shortest path

Input: Graph $G = (V, E)$, vertices s and t

Random parameter: L_e = length of edge e

System value: Φ_{PATH} = length of shortest path from s to t

Maximum flow

Input: Directed graph $G = (V, E)$ with vertices s and t

Random parameter: C_e = the capacity of edge e

System value: Φ_{FLOW} = the maximum (s,t)-flow in G

PERT network performance

Input: Directed acyclic graph $G = (V, E)$ with source vertex s and sink vertex t

Random parameter: T_e = time to complete task associated with edge e

System value: Φ_{PERT} = minimum time to complete the project, where the project starts at point s, ends at point t, and where no task can be started from vertex v until all tasks to vertex v are completed. Equivalently, Φ_{PERT} = length of *longest* path — with T_e representing edge lengths — from s to t.

More elaborate models could include both cost and capacity information (random or deterministic) and include a variety of measures of system operation. Some examples are stochastic min-cost flows, maximum matchings, or minimum spanning trees. The main concentration of this section is on the above three models.

Investigations of stochastic path and flow problems began about the same time as those of binary-state reliability problems. The PERT problem was probably the first of these problems to draw significant attention, and has certainly been the most popular of the stochastic network problems. An excellent account of the current state of computational methods in PERT optimization can be found in [127], and an extensive bibliography on the subject can be found in [7] The problem was first introduced in [283] in the context of project evaluations; early work on stochastic PERT problems also appears in [83, 167, 201]. The first analysis of stochastic shortest path problems was probably in [155], and early work concerning stochastic flow problems appears in [117, 158].

The three problems given above all have special cases which correspond precisely to the (s,t)-connectedness reliability measure $Rel_2(G, s, t, \mathrm{p})$ given in part 1.3. Specifically, each edge parameter takes on values of 0 or 1 corresponding to the operational state of the edge in the binary problem. The (s,t)-connectedness reliability function then has the following interpretations:

Shortest path: If edge failure corresponds to $L_e = 1$ and edge operation corresponds to $L_e = 0$, then $Rel_2(G, s, t, \mathrm{p}) = \Pr[\Phi_{PATH} = 0]$

Maximum flow: If edge failure corresponds to $C_e = 0$ and edge operation corresponds to $C_e = 1$, then $Rel_2(G, s, t, \mathrm{p}) = \Pr[\Phi_{FLOW} \geq 1]$

PERT network performance: If edge failure corresponds to $T_e = 0$, edge operation corresponds to $T_e = 1$, *and* every (s,t)-path in G has the same length n, then $Rel_2(G, s, t, \mathrm{p}) = \Pr[\Phi_{PERT} = n]$

It follows that these problems are #P-complete for any class of graphs for which the associated (s,t)-connectedness reliability problem is #P-complete (which includes those satisfying the additional property given for the PERT correspondence [330]). Similar arguments show that the computation of $E[\Phi]$ is also #P-complete for these same classes of graphs.

The type of distribution allowed for the edge random variables is of critical concern in the computational efficiency of the techniques discussed here, and hence it is necessary to outline the computational efficiency of computing and manipulating distributions for network problems. The following two operations play a major role in most of the computational schemes for multistate network reliability, and the computational difficulty of performing these operations is of primary importance.

- the *convolution* of two independent random variables X_1 and X_2, that is, the distribution of their sum $X_1 + X_2$;
- the *max (min)* of two independent random variables X_1 and X_2, that is, the distribution of $\max(X_1, X_2)$ or $\min(X_1, X_2)$.

A class of edge distributions for a multistate network problem must typically satisfy one or more of the following three criteria, depending on the type of analysis being performed on the associated network problem:

1. The computation of a given cdf value of an element in the class must be able to be performed to a given number of digits accuracy in time polynomial in the number of digits and the size of the input describing the distribution.
2. Given a set of distributions in the class and a sequence of k successive min/max/convolution operations starting with these distributions, the distribution resulting from this sequence of operations must also be in the class, and further, it must be possible to find the description of the resulting distribution in time polynomial in the size of the input describing the original distributions.
3. The expected value, variance, or more generally any specified moment must be computable (in terms of digits of accuracy) in polynomial time.

Typically, it is assumed that the random variables take on *discrete* distributions, in particular, ones having a finite number of values. Although the computation of a single convolution or max/min distribution is elementary, the computation of the distribution for a series of k of these operations is known to be #P-complete, even when each of the original variables has only two values. What is necessary to insure efficient computation of convolution and max/min distributions is that the random variables take on the consecutive values $\{1, 2, \ldots, x_q\}$ (or more generally consecutive multiples of some common denominator) on every edge of the graph, for some fixed q. Hagstrom [187] has in fact shown that in many cases multistate edge distributions such as this can be efficiently reduced to two-state distributions with edge "operation" probabilities all equal to 1/2.

There are two classes of infinite-valued distributions which are among the most general known to satisfy (1)–(3) above. The first is discrete-valued, and can be described as "mixtures of geometric" distributions, having pdf's of the form

$$f(x) = \sum_{i=1}^{q} \sum_{j=1}^{r} a_{ij} \binom{x+i-1}{i-1} (1-p^j)^x p^j i \quad x = 0, 1, \ldots$$

for $0 < p < 1$ and appropriately chosen values of a_{ij}. There is also a class of continuous distributions which satisfy the required properties. These distributions can be described as "mixtures of Erlang" distributions (also known as *Coxian* distributions [106, 350]). They are the continuous analogy to the "mixture of geometric" class described above, and have cdf's of the form

$$F(t) = \sum_{i=1}^{q} \sum_{j=1}^{r} a_{ij} t^i e^{-jt} \quad 0 \le t < \infty$$

for appropriately chosen values of a_{ij}.

We now give a sketch of the major evaluation and bounding techniques for multistate network problems. In most cases, it turns out that the same technique apply to two or more of the above problems. As well, most of the techniques have binary-state versions which have been presented in Sections 3 & 4. Thus we organize the discussion by technique rather than by subject, and follow, when possible, the format given for the binary version of the problem. For most of the discussion we concentrate on the evaluation of $\Pr[\Phi \ge \alpha]$ for the particular system value function Φ and specific system value α of interest.

6.3 Transformations and Reductions

One of the "reductions" popular in multistate problems that does not have an analogue in binary-state problems is to transform the multistate problem into one in which each arc has an "failed" state — where it is essentially deleted from the network — and an "operating" state — where it takes on a single length, capacity, or completion time. Although it generally does not provide any improvement in complexity of the associated reliability computation algorithm, it does allow conceptually easier application of factoring and other enumeration methods. The method for the PERT problem is given in [186], and for the shortest path problem in [290]. Specifically, if edge random variable X_e takes on values $x_1 \le x_2 < \ldots < x_q$ with $\Pr[X_e = x_i] = p_i$, $i = 1, \ldots, q$, then the edge e can be replaced by

Shortest path: q parallel edges, the i^{th} edge having length x_i and operating probability $p_i(1 - \sum_{j=1}^{i-1} p_j)^{-1}$

Maximum flow: q series edges, the i^{th} edge having capacity x_i and operating probability $p_i(1 - \sum_{j=1}^{i-1} p_j)^{-1}$

PERT network performance: q parallel edges, the i^{th} edge having completion time x_i and operating probability $p_i(1 - \sum_{j=i+1}^{q} p_j)^{-1}$

The deletion of irrelevant edges covered in part 3.1 applies as well to the three problems given above, since the state of an edge which does not lie on a minimal (s,t)-path affect neither path length, flow, or project completion time. Mandatory edges can likewise be contracted in the maximum flow problem. In particular, if mandatory edge e is contracted for instance graph G when computing $\Pr[\Phi_{FLOW} \geq \alpha]$, then the multiplicative factor applied to the problem on the contracted graph $G \cdot e$ is $\Pr[C_e \geq \alpha]$. In the shortest path and PERT problems mandatory edges cannot be immediately contracted, since the value of the edge affects the length of the shortest or longest path in the contracted graph. They do, however, induce three 1-attached subnetworks, and hence can be handled as indicated in part 6.3.

Series and parallel reductions have powerful analogues in path and flow problems ([284] for PERT). To summarize the use of these reductions, let e and f be two edges which are either in series or in parallel, and let g be the edge which replaces these two edges in the series or parallel reduction.

Shortest Path: For a *series* reduction, L_g is the *convolution* of L_e and L_f. For a *parallel* reduction, L_g is the *minimum* of L_e and L_f.

Maximum Flow: For a *series* reduction, C_g is the *minimum* of C_e and C_f. For a *parallel* reduction, C_g is the *convolution* of C_e and C_f.

PERT network performance: For a *series* reduction, T_g is the *convolution* of T_e and T_f. For a *parallel* reduction, T_g is the *maximum* of T_e and T_f.

More complicated subnetwork reductions have been considered for the PERT problem in [201, 342], and [343]. The 1- and 2- attached subnetworks treated in part 3.1 also have multistate analogues [125, 376]. First, let H be a 1-attached subnetwork of G, with r the attachment point, and let Φ^H and $\Phi^{\bar{H}}$ be the system value functions for the appropriate problem when applied to the subnetworks H and $G \setminus H$ with terminals s, v and v, t, respectively. Then the system value function Φ^G satisfies

Shortest Path and PERT network performance: Φ^G is the *convolution* of Φ^H and $\Phi^{\bar{H}}$

Maximum Flow: Φ^G is the *minimum* of Φ^H and $\Phi^{\bar{H}}$

Second, let H be a 2-attached subnetwork of G, with attachment points x and y, and let Φ^{xy} and Φ^{yx} be the appropriate system value function for the subgraph H when oriented from x to y and from y to x, respectively. Then the system reliability function of G is the same as that obtained by replacing the subgraph H by the two edges (x,y) and (y,x) having edge random parameters distributed as Φ^{xy} and Φ^{yx}, respectively.

Another reduction— discussed for PERT and reliability problems in [128], but applicable as well to shortest path and maximum flow problems — is called the *node reduction* [128]. It is actually an arc contraction, the arc having the property that it is either the only out-arc of its tail or the only in-arc of its head. The essential feature of this contraction is that it does not introduce any spurious paths, as could occur when an arbitrary arc is contracted in a directed graph. Thus the associated problem can be reduced to k subproblems on networks with one less edge and node, where k is the number of states taken on by the contracted edge. This is covered in more detail in the next part.

6.4 Efficient Algorithms for Restricted Classes

The evaluation of Φ for series-parallel graphs can be accomplished in the same manner as it is done for the (s,t)-connectedness problem, with series and parallel reductions performed as indicated above. The complexity is $O(Rn)$, where R is the worst-case complexity of performing a max/min or convolution at any time in the algorithm. Thus the complexity of these algorithms depends critically upon the time to perform a the series and parallel operations. For the three types of distributions given at the beginning of the section, R is linear in nq (in the finite case) or nqr (in the two infinite cases). It is generally believed that polynomial algorithms exist as well for graphs with bounded tree-width (see part 3.2) although this has not been treated specifically.

Another interesting class of stochastic network reliability problems which have efficient solution algorithms were observed by Nádas for PERT problems [299], who called them "complete tracking" problems. They are characterized by edge random variables of the form $X_e = a_e Z + b_e$, where a_e and b_e are edge parameters and Z is a common random variable. For PERT problems it turns out the resulting system reliability $\Pr\Phi \leq \alpha$ is equal to the maximum of the values

$$w(P) = \frac{\sum_{e \in P} b_e - t}{\sum_{e \in P} a_e}$$

taken over all (s,t)-paths P in G. Computing this maximum can be done in polynomial time by solving a modification of the "minimal cost-to-time ratio cycle problem" (see for example [258] pp. 94–97). The associated problem for shortest paths involves minimizing $w(P)$ over all (s,t)-paths P — and that for the maximum flow problem involves minimizing $w(C)$ over all (s,t)-*cuts* C — and likewise can be solved in polynomial time.

6.5 State-based Methods

Enumerative methods for computing multistate system reliability are necessarily restricted to problems having a finite number of edge states for each arc. Specifically, let each edge e_j have associated random parameter X_j taking on values $1, 2, \ldots, q$, with probabilities $p_{ij} = \Pr[X_j = x_j]$, $x_j = 1, \ldots, q$, and let the system value function Φ take on values $1, \ldots, K$. Then the two classic stochastic measures, namely the cdf of Φ evaluated at particular level $\alpha \in \{1, \ldots, K\}$, and the mean of Φ, can be written:

$$\Pr[\Phi \leq \alpha\,] \;=\; \sum_{\substack{(x_1, \ldots, x_n) \in \{1, \ldots, q\}^n \\ \Phi(x_1, \ldots, x_n) \leq \alpha}} \prod_{j=1}^{n} p_{x_j}$$

$$E[\Phi] = \sum_{(x_1,\ldots,x_n)\in\{1,\ldots,q\}^n} \prod_{j=1}^{n} p_{x_j}$$

the number of terms in the above two measures can be on the order of $q^{|E|}$, and hence these problems become intractable on a considerably smaller scale than even those of the binary state reliability measures. It is worth noting (and this was developed for the maximum flow problem in [384]) that when there are only a *fixed* number of random arcs, the enumeration can be performed in polynomial time.

The factoring method given in part 3.3 has an analogous, if somewhat cumbersome, form here. Namely, if for "pivotal edge" e and edge parameter value x, define $G_{e,x}$ to be the network having arc e fixed at value x. Then the Factoring Theorem for the above two measures becomes

$$\begin{aligned} \Pr[\Phi \le \alpha] &= \sum \Pr[\Phi_{G_{e,x}} \le \alpha] \\ E[\Phi] &= \sum E[\Phi_{G_{e,x}}] \end{aligned} \tag{2}$$

The factoring theorem was first applied to stochastic network problems in [117] (in a somewhat disguised form), and has been applied effectively to PERT and shortest path problems in [139], [128, 187, 191], and to the maximum flow problem in [260]. It can be combined elegantly with series-parallel reductions by means of the node reduction method given in part 6.3. The technique is given in [127], and is based on the simple observation that in any acyclic graph without parallel edges, there is always at least one node reduction which can be performed, say by contracting edge e one of whose endpoints v has only e as an in (out) edge. Let $e_1, \ldots, e_k$ be the other edges incident with vertex v. Then the system $G_{e,x}$ given above is simply $G \cdot e$, with the variables associated with $e_1, \ldots, e_k$ modified as follows:

Shortest Path: L_{e_i} is *shifted* by an amount x, that is, $L'_{e_i} = L_{e_i}$+x, $i = 1, \ldots, k$.

Maximum Flow: C_{e_i} is *capped* at value x, that is, $C'_{e_i} = \max\{C_{e_i}, x\}$, $i = 1, \ldots, k$.

PERT network performance: T_{e_i} is likewise shifted by an amount x, i=1,...,k.

The distributions of these modified random variables are easily computed for finite-state distributions, for example by treating them as convolutions or maximums having one of the random variables one-valued. Thus equation (2) reduces the particular problem on network G to k subproblems — where k is the number of values taken by arc e — all on the same network $G \cdot e$ but with different distributions on the edges $e_1, \ldots, e_k$. The complexity of computing the cdf value or mean in this case depends critically on the *number* of such node reductions which must be performed, in tandem with performing available series and parallel reductions, in order to reduce the network to a single edge. Bein *et al.* [41] have given an $O(n^3)$ algorithm for determining the *minimum* number of node reductions which must be performed in order to reduce a graph in the above manner. This number, therefore, in some sense also represents the "complexity" of a network with respect to path and flow problems.

There have also been some papers which use path- and cut-enumeration-based techniques to solve multistate flow problems. Evans [135] uses a lattice of cutsets to compute maximum flow, Mirchandani

[290] extends the disjoint products procedure of section 3.4, and Hagstrom [185] extends the inclusion-exclusion results of [360] to stochastic path problems.

The enumeration techniques given above are of little use in solving problems involving infinite or continuous edge random variables. When the edge random variables have *exponential* distributions Kulkarni *et al.* [248, 252, 251] give an interesting procedures for solving shortest path problems, maximum flow problems on planar networks, and PERT problems. We illustrate for the shortest path problem, the flow and PERT problems having similar, though more involved, solution techniques. We have a set of "runners", who begin at the source vertex s and each proceeds to traverse one of the edges going out of s. After an interval of time with known (gamma) distribution, one of the runners reaches the end of his edge. At this point the runner who has finished his edge stops, and simultaneously runners begin running along edges pointing away from the newly-reached vertex. (Runners are not sent along edges which go to previously visited vertices.) Due to the "memoryless" exponential distribution of running times, it follows that at the point at which the runners begin running from the newly-reached vertex one can assume that *all* of the runners have just started along their edge. In short, the process of these runners traversing the network is a *continuous time Markov chain.* A state of this Markov chain corresponds to a possible set of vertices which the runners have visited together with a corresponding set of edges on which the runners are currently running, and the absorbing states are those in which the sink vertex t has been labeled. The average time to absorption for this Markov chain is now precisely the expected length of a shortest path. The absorbing state probability, moreover, can be calculated easily by the appropriate ordering of the states of the Markov chain. Although the computation time is linear in the number of states of the Markov chain, this number grows exponentially in the size of the network. For networks of the order of 15–20 arcs, however, this method is fairly effective in finding the appropriate solution values. As well, it has been applied to other stochastic settings such as reliability [24], minimum spanning trees [249], and min cost flow [105] (for a unified framework, see [23]).

6.6 Bounding Techniques

Due to the particularly intractable nature of multistate network reliability problems, the dominant focus of research in this area is in developing techniques for bounding the various system measures of interest. These techniques often differ substantially from those used for binary-state problems.

The historically first technique used for approximating reliability in stochastic networks — and the one which has enjoyed the most attention — arises from the intuitively appealing notion that the expected value $E[\Phi]$ of the shortest path length/max flow/project completion time should be able to be obtained by replacing each *random* edge length/capacity/completion time by a *deterministic* parameter whose value is the *expectation* of this value, and then solving the *deterministic* version of the problem. This was in fact the solution technique proposed in the original treatment of PERT in Malcolm *et al.* [283]. It seems to be part of the folklore that the value obtained by this technique is an *upper bound* to the true value of $E\Phi$ in the PERT problem and a *lower bound* in the shortest path and maximum flow problems. (A unified account of this can be found in [420]).

Most of the succeeding research concentrated on approximating the cdf value $F(\alpha) = \Pr[\Phi \leq \alpha]$ for Φ. A early technique along the sames lines as Malcolm *et al.* was suggested by Charnes *et al.* [83]. It applied originally to the PERT problem with continuous edge parameter values, but can be modified to apply to shortest path/maximum flow problems and discretely distributed edge random variables as well. Specifically, suppose it is desired to compute the number α for which $F_{PERT}(\alpha) = \beta$ for some specified probability β. Replace each edge random variable T_e by the value t_e for which $\Pr[T_e \leq t_e] = \beta$, and solve for the deterministic shortest completion time. The resulting value is again an upper bound on the actual project completion time having the given cdf value β.

Improvements in above schemes for the PERT problem compute — for each vertex v in the graph — distributions for the intermediate random variable

$$\Phi_v = \text{ the longest path to vertex } v.$$

In all cases the computations are done in ***topological order*** $v_1 = s, v_2, \ldots, v_n = t$, that is, all edges pointing into v_i come from vertices v_j with $j < i$. The first of these schemes was proposed by Fulkerson [167]. It computes a lower bound E_i^L on $E[\Phi_{v_i}]$ using the fairly straightforward recursive formula

$$\begin{aligned} E_1^L &= 0 \\ E_i^L &= \sum_{\beta} p(\beta) \min_{j<i}(E_j^L + \beta_{v_j,v_i}) \qquad j = 2, \ldots, n \end{aligned}$$

the sum being taken over all vectors $\beta = (\beta_{ji} : j < i)$ of parameter values which can be taken by the edges pointing into vertex v_i (terms where (v_j, v_i) does not exist are ignored). Improvements on this technique primarily involved estimating the cdf values $F_i(\alpha)$ for Φ_{v_i}, in particular, computing ***upper*** and ***lower*** bounds $F_i^U(\alpha)$ and $F_i^L(\alpha)$, respectively. Note that these values can be used in turn to compute ***lower*** and ***upper*** bounds, respectively for the values $E(\Phi_{v_i})$, using the elementary formula

$$E[\Phi] = \sum_{\alpha}(1 - F(\alpha)).$$

All of them use the same type of recursive formula, with $F_1^U(\alpha) = F_1^L(\alpha) = 1$ for all nonnegative α (and zero otherwise). The first of these was given by Kleindorfer [241], namely,

$$F_i^U(\alpha) = \min_{j<i} \sum_{x_{ij}} \Pr[T_{v_jv_i} = x_{ij}] F_j^U(\alpha - x_{ij}) \qquad i = 2, \ldots, n$$

and

$$F_i^L(\alpha) = \prod_{j<i} \sum_{x_{ij}} \Pr[T_{v_jv_i} = x_{ij}] F_j^L(\alpha - x_{ij}) \qquad i = 2, \ldots, n$$

Shogan [373] gives bounds of the form

$$F_i^U(\alpha) = \sum_{\beta} p(\beta) \min_{j<i} F_j^U(\alpha - \beta_{ji}) \qquad i = 2, \ldots, n$$

and

$$F_i^L(\alpha) = \sum_{\beta} p(\beta) \prod_{j<i} F_j^L(\alpha - \beta_{ji}) \qquad i = 2, \ldots, n.$$

again with the sum taken over all vectors β of parameter values of edges pointing into v_i. He shows that the bounds apply under distributions where the indexing edges in the summand have a certain degree of dependence (i.e. are "associated"), and that the bounds are strictly better that those of Kleindorfer (the lower bounds being equal under complete independence of edge parameters). The corresponding bounds on $E[\Phi_{v_t}]$ are therefore likewise ordered, and his lower bound is strictly better than that of Fulkerson, with Kleindorfer's and Fulkerson's lower bounds not uniformly comparable. The bounds of Fulkerson, Kleindorfer, and Shogan can also be modified to apply to the shortest path problem — as long as the underlying graph is acyclic — and to the case of continuous edge parameters. Kleindorfer's bound is *a priori* polynomial-time computable, and Clingen [87] gives a polynomial-time computation of Fulkerson's bound, which can modified to compute Shogan's bound as well. Related research along these lines is found in [87], [345, 115, 113], and the work of Agnew as reported in [127]. The node reduction technique given in parts 6.2 and 6.3 have been used effectively in very similar bounding schemes for PERT problems [128], and Dodin [114] uses essentially the *inverse* of the node reduction technique to obtain yet another similar bounding technique.

Bounds for the flow problem require a different approach. A lower bound on $E[\Phi_{FLOW}]$ was first given in [19], with further refinements given in [79] and precise conditions for tightness of the bound given in [300]. It uses the chain formulation of flow (see [154], p.8). Here we assume that the random capacity C_e is binary, with the "operating" state representing normal capacity c_e with probability p_e, and the failed state representing capacity zero with probability $1 - p_e$. Let $\Gamma_1, \ldots, \Gamma_r$ be the set of all (s,t)-chains in G, and let $h_1, \ldots, h_r$ be an assignment of flow for each of the r (s,t)-chains. This flow is valid if for each edge e in G, the sum of the chain flows on chains passing through e is less than or equal to the capacity of e, and the value of the flow is $\sum_{k=1}^{r} h_k$. Consider any *fixed* chain flow $h_1, \ldots, h_r$ which is valid for the normal set of capacities (c_e: $e \in E$). In the random model, a particular chain Γ_k can therefore provide the requested portion h_k of flow if and only if all of its edges are operating, and provides zero flow otherwise. The (marginal) probability of this occurring is therefore $\gamma_k = \prod_{e \in \Gamma_k} p_e$. First, note that the value of $E[\Phi_{FLOW}]$ is clearly at least as great as the expected value of the random flow obtained by allowing the flow along each chain Γ_k to be at most h_k regardless of the operating condition of the other chains sharing edges with Γ_k. (as phrased in [79] "in the absence of rerouting"). This expectation in turn equals the sum of the expected flow values on each of the chains $\Gamma_1, \ldots, \Gamma_r$ taken as *independent* random variables. Summarizing,

$$E[\Phi_{FLOW}] \geq \sum_{k=1}^{r} h_k \gamma_k .$$

The papers [19, 79] give heuristics for the problem of finding a chain flow $h_1, \ldots, h_r$ which maximizes the right-hand side of this expression — in order to provide the best lower bound of this type — and [300] give conditions under which this bound is tight. The latter paper also gives a polynomial algorithm for finding the maximizing chain flow under these conditions, and thus provides the efficient special case algorithm referred to in part 6.2.

The method of edge-packing and noncrossing cuts discussed in parts 4.2.1 and 4.2.2 provides an effective method of bounding in the multistate network problems as well [385]. Specifically, let $P_1, \ldots, P_q$ and $\Gamma_1, \ldots, \Gamma_r$ be a collection of disjoint (s,t)-paths and (s,t)-cuts, respectively. These two collections provide natural *upper* and *lower bounding* functions Φ^L and Φ^U for the actual function Φ depending on the problem:

Shortest path:

$$\begin{aligned}
\Phi^U_{PATH} &= \text{the length of the shortest of the paths } P_1, \ldots, P_r \\
&= \min_{i=1,\ldots,r} \sum_{e \in P_i} L_e \\
\Phi^L_{PATH} &= \text{the sum of the lengths of the shortest edges from each of the cuts } \Gamma_1, \ldots, \Gamma_r \\
&= \sum_{i=1}^{r} \min_{e \in \Gamma_i} L_e
\end{aligned}$$

Maximum flow:

$$\begin{aligned}
\Phi^U_{FLOW} &= \text{the minimum capacity of the cuts } \Gamma_1, \ldots, \Gamma_r \\
&= \min_{i=1,\ldots,r} \sum_{e \in \Gamma_i} C_e \\
\Phi^L_{FLOW} &= \text{the maximum flow through the set of paths } P_1, \ldots, P_r \\
&= \sum_{i=1}^{r} \min_{e \in P_i} C_e
\end{aligned}$$

PERT network performance:

$$\begin{aligned}
\Phi^U_{PERT} &= \text{the sum of the completion times of longest edgesfrom each of the cuts } \Gamma_1, \ldots, \Gamma_r \\
&= \sum_{i=1}^{r} \max_{e \in \Gamma_i} T_e \\
\Phi^L_{PERT} &= \text{the completion time of the longest of the paths } P_1, \ldots, P_r \\
&= \max_{i=1,\ldots,r} \sum_{e \in P_i} T_e
\end{aligned}$$

These functional bounds in turn provide natural bounds for both the cdf and the expectation of the actual function Φ, in particular,

$$\Pr[\Phi^U \leq \alpha] \;\leq\; \Pr[\Phi \leq \alpha] \;\leq\; \Pr[\Phi^L \leq \alpha]$$

and

$$E[\Phi^L] \le E[\Phi] \le E[\Phi^U].$$

Further, each of the functions described above consists of one min/max and one convolution, and so the cdf's and expectations for all of these functions can be computed in polynomial time.

Two of the miscellaneous bounding techniques given in part 4.2.4 have been studied in the context of multistate problems as well. Prékopa and Boros [323] have extended the Bonferroni bounds to the maximum flow problem, and Nádas [299] has applied the Hailperin technique to PERT problems. We do not go into the details in this paper.

As a final note in this part, we discuss the extension of the techniques of parts 4.1.5 and 4.1.6. Bounding techniques using shellability and polyhedral combinatorics have been difficult to extend to multistate problems, due to their strongly combinatorial nature. One limited avenue of extension was investigated by Provan in [326]. In that paper bounds were found for reliability in the context where the components are represented by variables $y_1, \ldots, y_n$ constrained by the system

$$\begin{aligned} Ay &= b \\ y &\ge 0 \end{aligned} \tag{3}$$

where A is and $m \times n$ matrix and b an m-vector. The "failure" of component i corresponds to the variable y_i being removed from the system (or equivalently, set to zero) and the system operates when the remaining variables are sufficient to satisfy the system (3). The system value functions Φ_{PATH}, Φ_{FLOW}, and Φ_{PERT} — when in addition edge parameters have only two states — can all be represented in terms of a linear system in the form (3). Unfortunately, there is an additional restriction that the system (3) be ***nondegenerate***, that is, that all solutions to 3 have at least m nonzero components. The three problems given here do not have this property. Many variations of these problems do have representations corresponding to a nondegenerate linear system, such as requiring specified path shortest path lengths or flow values from a source to ***every*** point, or more generally having flow satisfy a set of "nondegenerate" supplies or demands at each node of the network. Furthermore by ***perturbing*** the linear system representing a degenerate problem, one arrives at a nondegenerate problem which provides a lower bound on the actual reliability, and hence the lower bound techniques of part 4.1 apply.

6.7 Monte Carlo Methods

Multistate problems — and particularly PERT problems — have always been prime candidates for Monte Carlo methods. As with the deterministic schemes, many of the Monte Carlo schemes given in part 5 can be extended to multistate problems. In the interest of brevity, we will only touch upon the major multistate Monte Carlo sampling schemes. For a general account of Monte Carlo schemes the reader is again referred to [194], and for applications to the PERT problem the accounts in [125] and the two surveys [7, 127] for details and additional information.

Monte Carlo schemes for multistate problems deal almost exclusively with the PERT problem. The earliest Monte Carlo treatment of PERT problems seems to be the paper by Van Slyke [411].

He uses the naive sampling technique to that given in part 6.1, extended to multistate problems. Specifically, suppose that an estimate for, say, the sample mean $E[\Phi]$ is desired. For each edge e let $F_e(\alpha) = \Pr[X_e \leq \alpha]$ be the cdf for the random variable X_e. The sample value $\hat{x}_e$ for this random variable is chosen by drawing a sample $\hat{U}_e$ from a uniform random number generator, and setting $\hat{x}_e = F_e^{-1}(\hat{U}_e)$ (or $x_e = \min\{x|\ F_e(x) \geq U_e\}$ for discrete distributions). After the entire sample state vector $\hat{x}$ is generated $\Phi(\hat{x})$ is computed using the appropriate deterministic algorithm. The average over all sample system values is then an unbiased estimator for $E[\Phi]$.

Early work on variance reduction for the naive Monte Carlo method concentrated on applying classical variance-reduction techniques. Burt, Gaver and Perlas [70], and later Burt and Garman [72] investigate the improvement to the PERT problem gained by using classical techniques such as antithetic variates, control variates, stratified sampling, regression, and conditional sampling. Again, as these are primarily probabilistic rather than network methods we refer the reader to a text like [194] for details.

One of the most frequently used network techniques for solving multistate problems is based on the conditional sampling method. The idea is to determine a "small" set of arcs such that if the edge parameters on this set of arcs is fixed, then the conditional system probability or expectation can be determined analytically. One then samples only from this small set, and the resulting conditional probabilities or expectations are averaged to produce the overall sample system measure. Burt and Garman [71] suggest conditioning on the *common edges* of the network, that is, edges which share two or more (s,t)-paths. In acyclic networks these can be found efficiently, and after fixing these lengths, the remaining network can be analyzed as if it were a collection of disjoint paths (see section 5).

Unfortunately, in a reasonably complex graph all, or nearly all, of the arcs in the graph will be common, and so this will not lead to a significant improvement in the sampling. Sigal, Pritsker, and Solberg [379, 380, 381] suggest the use of *uniformly directed* (s,t)*-cuts* in a conditional sampling scheme. Uniformly directed (s,t)-cuts are edge-sets having the property that each (s,t)-path in G intersects C in *exactly* one edge. (As a technical point, these should be called *exact* (s,t)-cuts, although in PERT networks the definition given by Sigal *et al.* is equivalent to that given here. See [333] for the precise distinction between these two terms.) The importance of conditioning on the edges of a uniformly directed (s,t)-cut is that this allows the activity of the (s,t)-paths to be analyzed independently on each side of the cut. As well, every graph always contains at least one uniformly directed (s,t)-cut. Kulkarni and Provan [253, 333] show how the conditional system measure can be found efficiently after the non-cut edges have been sampled, and also how the maximum cardinality uniformly directed (s,t)-cut can be found to use in such a sampling scheme. Fishman [139] has combined the use of uniformly directed (s,t)-cuts and *quasirandom sampling* to improve the method of Sigal *et al.* still further. Additional work along this line is found in [5, 6].

Most of the Monte Carlo techniques given above also apply to the shortest path problem as well, and generally do not require that the underlying graph be acyclic as do many of the bounding techniques for the PERT problem. Fishman *et al.* [17, 144], [146, 150] seem to be the only group to explicitly address the maximum flow problem using Monte Carlo methods. The approach uses the multistate extension of the bounds-based sampling method presented in part 5.4 to compute a cdf value $\Pr[\Phi \leq \alpha]$ for threshold value α. We give it in its general form, for the technique applies just as easily to both the PERT and shortest path problems. As above, for $e \in E$ let the edge parameter X_e have cdf $F_e(x)$. Suppose that you have functional bounds Φ^L and Φ^U for the multivariate measure Φ, satisfying

- $\Phi^L(x) \leq \Phi(x) \leq \Phi^U(x)$ for every state vector x
- For $k = 0, \ldots, m$, any assignment $\hat{x}^{(k)} = (\hat{x}_1, \ldots, \hat{x}_k)$ of values for the first k components of x, the conditional cdf values

$$R_k^L(x^{(k)}) \equiv \Pr[\Phi^L \leq \alpha | \, x_1 = \hat{x}_1, \ldots, x_k = \hat{x}_k]$$

  and

$$R_u^U(x^{(k)}) \equiv \Pr[\Phi^U \leq \alpha | \, x_1 = \hat{x}_1, \ldots, x_k = \hat{x}_k]$$

  can be computed in polynomial time.

The space X of importance in the multivariate version of the problem is now

$$\mathbf{X} = \{x \in \{0,1\}^E : \Phi^L(x) \leq \alpha, \ \Phi^U(x) > \alpha\}$$

and the modification to the Bounds-based Sampling Method given in part 5.4 becomes

1. Take samples $\hat{x} = (\hat{x}_1, \ldots, \hat{x}_m)$ from the space $\mathbf{X}$ by successively drawing, for $k = 1, \ldots, m$, the component state $\hat{x}_k$ from the cdf

$$\begin{aligned} &\Pr[x_k \leq \beta | \, x_1 = \hat{x}_1, \ldots, x_{k-1} = \hat{x}_{k-1} \textit{ and } \Phi^U(x) \leq \alpha, \ \Phi^L(x) > \alpha] \\ &= \left[\frac{R_k^U(x^{(k-1)}, \alpha) - R_k^L(x^{(k-1)}, \alpha)}{R_{k-1}^U(x^{(k-1)}) - R_{k-1}^L(x^{(k-1)})} \right] F_{e_k}(\alpha) \end{aligned}$$

2. Compute the proportion $\hat{R}$ of those samples for which $\Phi(x) \leq \alpha$. The number $R_0^L + \hat{R}(R_0^U - R_0^L)$ is now an unbiased estimator of R

The edge-packing/noncrossing-cuts-based functional bounds given in part 6.4 also provide excellent bounding functions for this method, since the conditional cdf's for these functions can be computed easily as well. Although in the papers by Fishman *et al.* these bounds are used only for the maximum flow problems, as given in part 6.4 they apply as well to PERT and shortest path problems, essentially as stated here.

7 Using Computational Techniques in Practice

After reading the previous six sections it would be understandable if one were confused in deciding how to apply the myriad of reliability measures, algorithms, bounds, and so on to the solution of real problems. In this final section we provide some guidance on this issue.

7.1 Where Does Reliability Fit In

Although this entire chapter has been devoted to reliability it would be inaccurate to give the impression that reliability was the only criteria of interest, or even the most important criteria in the design of most networks. In fact, there are typically several competing interests, including cost, overall system capacity or throughput and various performance criteria [159]. The most typical scenario encountered during a network design session is:

- Minimize cost subject to:
  - throughput constraint(s)
  - performance constraint(s)
  - reliability constraint(s)

The throughput constraints typically state that the network must have capacity sufficient to support traffic requirements specified for a set of origin/destination pairs. The predominant performance measure for packet switched networks is some measure of message or packet delay [175] and the predominant performance measure for circuit switched networks is some measure of lost or blocked calls [354]. Ideally the performance and reliability constraints would restrict an accurate expression for performance and reliability, respectively, to be within certain limits. However, as was stated in the introduction, since computing the values of performance and reliability measures are typically very difficult, surrogates are usually employed. A typical surrogate for delay is a path length restriction and a typical surrogate for reliability is a connectivity restriction. It should also be noted that even if exact measures of performance and reliability were included as constraints the model given above would contain approximations since throughput, performance and reliability were treated as separate constraints. Ideally, one would like a design that satisfied certain throughput and performance criteria even in the presence of failures. All of these shortcomings lead to the use of detailed performance and reliability analysis algorithms. That is, once an initial design is obtained it is typically refined, either manually or automatically, based on the results of performance and reliability analysis.

The issue of whether specific performance or reliability constraints are required and of whether performance and reliability analysis algorithms are required depends on the specifics of the problem setting. For example, if networks designed based on only cost and performance criteria naturally were very reliable then reliability analysis and reliability constraints might be unnecessary. This scenario might occur under any of the following circumstances:

1. the network components were themselves highly reliable;
2. occasional network failures were not particularly disruptive,
3. networks designed based on other (non-reliability) criteria tended to be very dense and consequently had high reliability (this might occur if the link capacities were small relative to the overall throughput required).

An analogous set of statements could be made relative to performance and throughput criteria. For example, the models described in Chapter 15 employ reliability constraints but not delay or throughput constraints because the fiber optic links used have very high capacity (and speed) so that capacity and delay criteria are met without the need of specific model consideration. On the other hand, since the link capacities are so high network designs naturally tend to be very sparse so that explicit reliability constraints are necessary.

7.2 Choosing a Measure

We have given detailed treatment to several different reliability measures and, in fact, there are still other measures which we have not mentioned and have given only cursory treatment to. One might be left with the dilemma of which measure to choose. The following considerations are fundamental to this decision:

1. significance and nature of performance criteria,

2. community of users served,

3. philosophy of service: good on the average or good at the extremes.

A fundamental decision is whether to use a connectivity measure or a performability measure. Connectivity measures include all versions of the k-terminal measure as well as certain related measures such as network resilience, the expected fraction of node pairs communicating [26, 89]. Performability measures explicitly model variations in performance caused by network failures. A connectivity measure would be useful whenever the network performance was considered to be satisfactory as long as the network was connected. This is considered to be the case in many fiber optic networks. Connectivity measures are also useful in measuring the probability that the network is able to provide some minimum level of service, i.e. the probability that the network can handle essential or emergency calls. The study of performability measures was motivated by network settings where it is possible that component failures could degrade system performance to unsatisfactory levels while, nonetheless, the network remained connected. In such cases the use of performability measures is necessary.

The choice among the two-terminal, k-terminal and all-terminal depends on the community of users of interest. The two-terminal measure measures the ability of the network to satisfy the communications needs of a specific pair of user terminals. Thus, the measure can be viewed as a user specific measure. On the other hand the all-terminal measure takes a system provider perspective. It measures the performance of the system relative to its ability to provide service all possible terminal pairs. It should be noted that in a certain sense the all-terminal measure is extremely "conservative". Specifically, the all-terminal measures is smaller than the smallest two-terminal value and, generally, could be much less.

Another system-wide reliability measure is the minimum over all two-terminal values. This can be interpreted as the reliability level guaranteed by the network to all users. Another related value is the average over all two-terminal values, which equals the resilience. The choice between the minimum and average relates to the overall philosophy of the service provider — specifically, is the objective to provide satisfactory service "on the average" or to guarantee a certain service level?

The general k-terminal measure addresses the interests of a community of users located at a subset of network nodes. Depending on the node subset chosen it could provide system wide information or user specific information.

The most appropriate performability measures is usually dictated by the performance measure of interest in the application being addressed. In the case of performability measures the choice exists between measuring the average performance level and measuring some extreme value, e.g. 95th percentile. Again, this relates to the service philosophy mentioned above.

7.3 Choosing the Right Algorithm

In addition to presenting the reader with a variety of reliability measures, this chapter has also provided numerous choices for computing each of the measures mentioned, including exact algorithms, analytic bounds and Monte Carlo simulation. Thus, one might be left with a sense of confusion in this area as well. The choice among algorithms depends largely on the size and structure of the networks involved. Ideally one would like to employ an algorithm that gives the exact reliability value. However, efficient (polynomially bounded) algorithms are only available for certain structured classes of graphs as are described in Section 3. For general graphs enumerative algorithms can only solve problems of limited size. For other situations approximate algorithms must be employed. The choice between analytic bounds and Monte Carlo is a more subtle one. Analytic bounds depend on specific problem results. Consequently, as described in Section 4, such bounds only exist for certain problem classes. Furthermore, their quality and the associated algorithm running time may differ depending on the specific problem class in question. In certain cases, the bounds available are very good and they can be computed very quickly. However, in other cases this is not true. The advantages of Monte Carlo are that some Monte Carlo approaches can be constructed for virtually all measures of interest and a Monte Carlo algorithm can be run for a long or short period of time with a commensurate increase or decrease in the level of accuracy. We should note that some of the more advanced Monte Carlo methods make use of problem structure and, consequently, do not have the first advantage.

One important issue to point out relative to analytic bounds is whether unequal failure probabilities are allowed. Some bounding methods bound the reliability polynomial and as a result only give information for the case where all failure probabilities are equal. On the other hand, it should be pointed out that the reliability polynomial provides information over the entire range of values for that one failure probability.

A related issue that should be considered is whether the directed or undirected graph model is used. As was pointed out in Section 2, the directed graph model is more general in that for a wide class of measures it allows for the easy modeling of both directed and undirected problems and of problems with and without node failures.

7.4 Design Criteria

The design criterion most commonly used as a surrogate for a constraint on all-terminal reliability is a connectivity constraint, that is, a constraint that the network connectivity must be at least c. The generalization of this criterion to SCBSs (and to other classes of systems, including to k-terminal problems and certain performability measures) is a constraint that the SCBS contain no cutset of size $c-1$ or smaller. Probably the most common scenario is the case where $c = 2$. With $c = 2$, the design criterion states that the system should contain no single point of failure.

Let us examine the reliability level guaranteed by this design criterion. If there are no cutsets of size $c-1$ or smaller then a lower bound on the system reliability would be obtained by assuming that every set of elements of size c was a cutset. For an m-element SCBS the reliability of this system would be equal to the reliability of a K-out-of-N system with K=$m-c$ and N=m. Assuming equal failure probability the system reliability would be:

$$\sum_{i=m-c}^{m} \binom{n}{i} p^i (1-p)^{m-i}$$

This value is the best lower bound possible on the system reliability level guaranteed by a c-connectivity constraint, assuming no additional system structure is known. We note for $c = 2$ that this bound is achieved for the case of all terminal reliability by a simple cycle.

It is usually very worthwhile to compute this bound and to take it into account during network design. It is often the case that the value of the bound is lower than expected. If this is the case that either the design criteria (the value of c) must be increased or a more detailed reliability analysis must be carried out together with possible design modifications. Once more complex measures of reliability are warranted in network design, the task of the network designer must not be simply to obtain a numerical measure of reliability, but rather to gain insight into the impact of the underlying network topology on the network's ability to perform its desired functions [91, 101]. Ultimately, the large number of techniques developed here for producing numerical measures of reliability contribute primarily not by giving algorithms to produce numbers, but rather by providing tools for capturing in part how network structure affects network performance.

Acknowledgements

The research of the first author was supported by National Science Foundation grant number NSFD CDR-8803012. The research of the second author was supported by NSERC Canada under grant A0579.

References

[1] H.M.F. AboElFotoh and C.J. Colbourn, *Series-parallel bounds for the two-terminal reliability problem*, ORSA J. Computing 1 (1989), 205–222.

[2] H.M.F. AboElFotoh and C.J. Colbourn, *Computing the two-terminal reliability for radio broadcast networks*, IEEE Trans. Reliability R-38 (1989), 538–555.

[3] H.M.F. AboElFotoh and C.J. Colbourn, *Efficient algorithms for computing the reliability of permutation and interval graphs*, Networks 20 (1990), 883–898.

[4] J.A. Abraham, *An improved algorithm for network reliability*, IEEE Trans. Reliability R-28 (1979), 58–61.

[5] V.G. Adlakha, *An improved conditional Monte Carlo technique for the stochastic shortest path problem* Management Sci. 32 (1986), 1360–1367.

[6] V.G. Adlakha, *A Monte Carlo technique with quasirandom points for the stochastic shortest path problem*, Amer. J. Math. Management Sci. 7 (1987), 325 358.

[7] V.G. Adlakha and V.G. Kulkarni, *A classified bibliography of research on PERT networks*, INFOR 27 (1989), 272-296.

[8] K.K. Aggarwal, K.B. Misra, and J.S. Gupta, *A fast algorithm for reliability evaluation*, IEEE Trans. Reliability R-24 (1975), 83–85.

[9] K.K. Aggarwal and S. Rai, *Reliability evaluation in computer-communication networks*, IEEE Trans. Reliability R-30 (1981), 32–35.

[10] A. Agrawal and R.E. Barlow, *A survey of network reliability and domination theory*, Oper. Res. 32 (1984), 478–492.

[11] A. Agrawal and A. Satyanarayana, *An $O(|E|)$, time algorithm for computing the reliability of a class of directed networks*, Oper. Res. 32 (1984), 493–515.

[12] A. Agrawal and A. Satyanarayana, *Network reliability analysis using 2-connected digraph reductions*, Networks 15 (1985), 239–256.

[13] D.P. Agrawal, *Graph theoretical analysis and design of multistage interconnection networks*, IEEE Trans. Computers C-32 (1983), 637-648.

[14] R. Ahlswede and D.E. Daykin, *An inequality for the weights of two families of sets, their unions, and their intersections*, Z. Wahrscheinl. Geb. 43 (1978), 183–185.

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

[16] D.J. Aldous, *The random walk construction of uniform spanning trees and uniform labeled trees*, SIAM J. Discrete Math. 3 (1990), 450–465.

[17] C. Alexopoulos and G.S. Fishman, *Characterizing stochastic flow networks using the Monte Carlo method*, Networks 21 (1991), 775–798.

[18] G. Andreatta, *Shortest path models in stochastic networks*, Stochastics in combinatorial optimization (1986), 178–186.

[19] Y.P. Aneja and K.P.K. Nair, *Maximum expected flow in a network subject to arc failures*, Networks 10 (1980), 45–57.

[20] S. Arnborg and A. Proskurowski, *Linear time algorithms for NP-hard problems restricted to k-trees*, Discrete Appl. Math. 23 (1989), 11–24.

[21] J.Y. Assous, *First and second order bounds for terminal reliability*, Networks 16 (1986), 319–329.

[22] E. Ayanoglu and C.-L. I, *A method of computing the coefficients of the network reliability polynomial*, Proc. Globecom89 (1989), 331–337.

[23] M.P. Bailey, *Constant access systems: a general framework for greedy optimization on stochastic networks*, Technical Report, Department of Operations Research, U.S. Naval Postgraduate School, Monerey, CA (1991).

[24] M.P. Bailey and V.G. Kulkarni, *A recursive algorithm for computing exact reliability measures*, IEEE Trans. Reliability R-35 (1986), 36–40.

[25] M.O. Ball, *Computing network reliability*, Oper. Res. 27 (1979), 823–836.

[26] M.O. Ball, *Complexity of network reliability computations*, Networks 10 (1980), 153–165.

[27] M.O. Ball and G.L. Nemhauser, *Matroids and a reliability analysis problem*, Math. Oper. Res. 4 (1979), 132–143.

[28] M.O. Ball and J.S. Provan, *Bounds on the reliability polynomial for shellable independence systems*, SIAM J. Algebraic and Discrete Methods 3 (1982), 166–181.

[29] M.O. Ball and J.S. Provan, *Calculating bounds on reachability and connectedness in stochastic networks*, Networks 13 (1983), 253–278.

[30] M.O. Ball and J.S. Provan, *Properties of systems which lead to efficient computation of reliability*, Proc. IEEE Global Telecommunications Conference, 1985, pp. 866–870.

[31] M.O. Ball and J.S. Provan, *Computing k-terminal reliability in time polynomial in the number of (s,K)-quasicuts*, Proc. Fourth Army Conf. Applied Math. and Computing, 1987, 901-907.

[32] M.O. Ball and J.S. Provan, *Disjoint products and efficient computation of reliability*, Oper. Res. 36 (1988), 703–716.

[33] M.O. Ball, J.S. Provan and D.R. Shier, *Reliability covering problems*, Networks 21 (1991), 345–358.

[34] M.O. Ball and R.M. Van Slyke, *Backtracking algorithms for network reliability analysis*, Annals of Discrete Math. 1 (1977), 49–64

[35] R.E. Barlow, *Set theoretic signed domination for coherent structures*, Report ORC 82–1, Univ. California Berkeley, 1982.

[36] R.E. Barlow and S. Iyer, *Computational complexity of coherent systems and the reliability polynomial*, Prob. Eng. Inf. Sci. 2 (1988), 461–469.

[37] R.E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*, To Begin With, Silver Spring, MD, 1981.

[38] D. Bauer, F. Boesch, C. Suffel, and R. Tindell, *Combinatorial optimization problems in the analysis and design of probabilistic networks*, Stevens Research Report in Mathematics 8202, Stevens Institute of Technology, Hoboken NJ, 1982.

[39] F. Beichelt and L. Spross, *Bounds on the reliability of binary coherent systems*, IEEE Trans. Reliability R–38 (1989) 425–427.

[40] F. Beichelt and P. Tittmann, *A generalized reduction method for the connectedness probability of stochastic networks*, IEEE Trans. Reliability R–40 (1991), 198–204.

[41] W.W. Bein, J. Kamburowski and M.F.M. Stallmann, *Optimal reduction of directed acyclic graphs*, OR Report 251, Operations Research Series, North Carolina State University, Raleigh, NC, 1990. To appear in *SIAM J. Computing.*

[42] S.G. Belovich and V.K. Konangi, *A linear-time approximation method for computing the reliability of a network*, Computer Networks and ISDN Systems 21 (1991), 121–127.

[43] D. Bienstock, *An algorithm for reliability analysis of planar graphs*, Networks 16 (1986), 411-422.

[44] D. Bienstock, *Some lattice-theoretic tools for network reliability analysis*, Math. Oper. Res. 13 (1988), 467–478.

[45] D. Bienstock, *Asymptotic analysis of some network reliability models*, SIAM J. Discrete Math. 1 (1988), 14–21.

[46] N.L. Biggs, Algebraic Graph Theory, Cambridge University Press, 1974.

[47] L.J. Billera, *Polyhedral theory and commutative algebra*, in: Mathematical Programming: The State Of The Art (A. Bachem, M. Grötschel, B. Korte, editors), Springer–Verlag, 1977, pp. 57–77.

[48] L.J. Billera and C.W. Lee, *The number of faces of polytope pairs and unbounded polyhedra*, European J. Combinatorics 2 (1981), 307–332.

[49] Z.W. Birnbaum and J.D. Esary, *Modules of coherent binary systems*, J. SIAM 13 (1965) 444-462.

[50] Z.W. Birnbaum, J.D. Esary, and S.C. Saunders, *Multi-component systems and structures and their reliability*, Technometrics 3 (1961), 55–77.

[51] R. Bixby, *The minimum number of edges and vertices in a graph with edge-connectivity N and M N-bonds*, Networks 5 (1975), 259–298.

[52] J.T. Blake and K.S. Trivedi, *Multistage interconnection network reliability*, IEEE Trans. Computers C–38 (1989), 1600–1603.

[53] J.T. Blake and K.S. Trivedi, *Reliability analysis of interconnection networks using hierarchical composition*, IEEE Trans. Reliability R–38 (1989), 111–119.

[54] L. Bodin, *Approximation to system reliability using a modular decomposition*, Technometrics 12 (1970), 335.

[55] F.T. Boesch, *Synthesis of reliable networks – a survey*, IEEE Trans. Reliability R–35 (1986), 240–246.

[56] F.T. Boesch, A. Satyanarayana, and C.L. Suffel, *Some alternate characterizations of reliability domination*, Prob. Eng. Inf. Sci. 4 (1990), 257–276.

[57] F.T. Boesch, A. Satyanarayana and C.L. Suffel, *Least reliable networks and the reliability domination*, IEEE Trans. Communications 38 (1990), 2004–2009.

[58] C.Botting, S.Rai, and D.P. Agrawal, *Reliability computation of multistage interconnection networks*, IEEE Trans. Reliability R–38 (1989), 138–145.

[59] T.B. Brecht and C.J. Colbourn, *Lower bounds for two-terminal network reliability*, Discrete Applied Math. 21 (1988), 185-198.

[60] T.B. Brecht and C.J. Colbourn, *Improving reliability bounds in computer networks*, Networks 16 (1986), 369–380.

[61] T.B. Brecht and C.J. Colbourn, *Multiplicative improvements in reliability bounds*, Networks 19 (1989), 521–530.

[62] A. Broder, *Generating random spanning trees*, in: Proc. Symp. Foundations of Computer Science, IEEE, 1989, pp. 442–447.

[63] R.L. Brooks, C.A.B. Smith, A.H. Stone, and W.T. Tutte, *The dissection of rectangles into squares*, Duke Math. J. 7 (1940), 312–340.

[64] J.I. Brown and C.J. Colbourn, *A set system polynomial with reliability and coloring applications*, SIAM J. Discrete Math. 1 (1988), 151–157.

[65] J.I. Brown and C.J. Colbourn, *Log concavity and the reliability polynomial*, Adv. Appl. Math., to appear.

[66] J.I. Brown and C.J. Colbourn, *Roots of the reliability polynomial*, SIAM J. Disc. Math., to appear.

[67] J.I. Brown, C.J. Colbourn and J.S. Devitt, *Network transformations and bounding network reliability*, Networks, to appear.

[68] D.B. Brown, *A computerized algorithm for determining the reliability of redundant configurations*, IEEE Trans. Reliability R-20 (1971), 121–124.

[69] J.V. Bukowski, *On the determination of large scale system reliability*, IEEE Trans. Systems, Man and Cybernetics, SMC-12 (1982), 538–548.

[70] J.M. Burt, Jr., D.P. Gaver, and M. Perlas, *Simple stochastic networks: some problems and procedures*, Naval Research Logistics Quarterly 17 (1970), 439–460.

[71] J.M. Burt, Jr. and M.B. Garman, *Monte Carlo techniques for stochastic PERT network analysis*, INFOR 9 (1971), 248–262.

[72] J.M. Burt, Jr. and M.B. Garman, *Conditional Monte Carlo: A simulation technique for stochastic network analysis*, Management Science 18 (1971), 207–217.

[73] R.G. Busacker and P.J. Gowen, *A procedure for determining a family of minimal cost flow patterns*, ORO Research Report 15, Johns Hopkins University, 1961.

[74] J.A. Buzacott, *A recursive algorithm for finding reliability measures related to the connection of nodes in a graph*, Networks 10 (1980), 311–327.

[75] J.A. Buzacott, *The ordering of terms in cut-based recursive disjoint products*, IEEE Trans. Reliability R-32 (1983), 472–474.

[76] J.A. Buzacott, *Node partition formula for directed graph reliability*, Networks 17 (1987), 227–240.

[77] J.A. Buzacott and J.S.K. Chang, *Cut-set intersections and node partitions*, IEEE Trans. Reliability R-33 (1984), 385–389.

[78] L. Caccetta and M. Kraetzl, *Blocking probabilities of certain classes of channel graphs*, preprint, Curtin University of Technology, 1990.

[79] M. Carey and C. Hendrickson, *Bounds on expected performance with links subject to failure*, Networks 14 (1984), 439–456.

[80] E.H. Carrasco and C.J. Colbourn, *Reliability bounds for networks with statistical dependence*, Proceedings of INFOCOM86, Miami, 1986, pp. 290–292.

[81] M.K. Chari, *Steiner complexes, matroid ports, and K–connectedness reliability*, preprint, University of North Carolina, 1990.

[82] M.K. Chari, T.A. Feo and J.S. Provan, *A computational study of wye-delta bounds for (s,t)-connectedness reliability*, preprint, University of North Carolina, 1990.

[83] A. Charnes, W.W. Cooper, and G.L. Thompson, *Critical path analysis via chance constrained and stochastic programming*, Oper. Res. 12 (1964), 460–470.

[84] J. Clausen and L.A. Hansen, *Finding k edge–disjoint spanning trees of minimum total weight in a network: an application of matroid theory*, Math. Programming Study 13 (1980), 88–101.

[85] S.-N. Chiou and V.O.K. Li, *Reliability analysis of a communication network with multimode components*, IEEE Trans. Sel. Areas Commun. SAC–4 (1986) 1156–1161.

[86] G.F. Clements and B. Lindström, *A generalization of a combinatorial theorem of Macaulay*, J. Combinatorial Theory 7 (1968), 230–238.

[87] C.T. Clingen, *A modification of Fulkerson's PERT algorithm*, Oper. Res. 12 (1964), 629–632.

[88] C.J. Colbourn, *The Combinatorics of Network Reliability*, Oxford University Press, Oxford and New York, 1987.

[89] C.J. Colbourn, *Network resilience*, SIAM J. Algebraic Discrete Methods 8 (1987), 404–409.

[90] C.J. Colbourn, *Edge–packings of graphs and network reliability*, Discrete Math. 72 (1988), 49-61.

[91] C.J. Colbourn, *Network reliability: numbers or insight?*, Annals of Oper. Res. 33 (1991), 87–93.

[92] C.J. Colbourn, *A note on bounding k-terminal reliability*, Algorithmica 7 (1992), 303–307.

[93] C.J. Colbourn, R.P.J. Day and L.D. Nel, *Unranking and ranking spanning trees of a graph*, J. Algorithms 10 (1989), 271–286.

[94] C.J. Colbourn, J.S. Devitt, D.D. Harms and M. Kraetzl, *Renormalization for channel graphs*, Actas XI Congreso de Metodologias en Ingenieria de Sistemas, Santiago, Chile, 1991, pp. 171–174.

[95] C.J. Colbourn, J.S. Devitt, D.D. Harms and M. Kraetzl, *Assessing reliability of multistage interconnection networks*, preprint, 1991.

[96] C.J. Colbourn, M. Elbert, E. Litvak and T. Weyant, *Performability analysis of large–scale packet–switching networks*, International Conference on Communications (SUPERCOMM/ICC) 1992, Chicago, pp. 416–419.

[97] C.J. Colbourn and E.S. Elmallah, *Reliable assignments of processors to tasks and factoring on matroids*, Discrete Math., to appear.

[98] C.J. Colbourn and D.D. Harms, *Bounds for all-terminal reliability in computer networks*, Networks 18 (1988), 1–12.

[99] C.J. Colbourn and E.I. Litvak, *Bounding network reliability by graph transformations*, in: *Reliability of Computer and Communications Networks*, AMS/ACM, 1991, pp. 91–104.

[100] C.J. Colbourn and M.V. Lomonosov, *Renewal networks: connectivity and reliability on a time interval*, Probability in the Engineering and Information Sciences 5 (1991), 361–368.

[101] C.J. Colbourn and L.D. Nel, *Using and abusing bounds for network reliability*, Proc. IEEE Telecommunications Conference (Globecom90), IEEE Press, pp. 663–667.

[102] C.J. Colbourn, L.D. Nel, T.B. Boffey and D.F. Yates, *Probabilistic estimation of damage from fire spread*, preprint, 1990.

[103] C.J. Colbourn, J.S. Provan and D.L. Vertigan, *The complexity of computing the Tutte polynomial on transversal matroids*, preprint, 1990.

[104] C.J. Colbourn and W.R. Pulleyblank, *Matroid Steiner problems, the Tutte polynomial and network reliability*, J. Combinatorial Theory B41 (1989), 20–31.

[105] G.A. Corea and V.G. Kulkarni, *Minimum cost routing on stochastic networks*, Networks 18 (1990), 527–536.

[106] D.R. Cox, *The use of complex probability in the theory of stochastic processes*, Proc. Cambridge Philosophical Soc. 51 (1955), 313-319.

[107] H.H. Crapo, *A higher invariant for matroids*, J. Comb. Theory 2 (1967), 406–417.

[108] D.E. Daykin, *A simple proof of the Kruskal-Katona theorem*, J. Combinatorial Theory A 17 (1974), 252–253.

[109] P. Dean, *A new Monte Carlo method for percolation probabilities on a lattice*, Proc. Cambridge Philosophical Soc. 59 (1963), 397–410.

[110] J. deMercado, N. Spyratos, and B.A. Bowen, *A method for calculation of network reliability*, IEEE Trans. Reliability R-25 (1976), 71–76.

[111] J.S. Devitt and C.J. Colbourn, *On implementing an environment for investigating network reliability*, in: *Computer Science and Operations Research*, Pergamon Press, 1992, to appear.

[112] G.A. Dirac, *Short proof of Menger's theorem*, Mathematika 13 (1966), 42–44.

[113] B.M.Dodin, *Approximating the distribution function in stochastic networks*, Computers and Oper. Res. 12 (1985), 207–223.

[114] B.M.Dodin, *Bounding the project completion time distribution in PERT networks*, Oper. Res. 33 (1985), 862–881.

[115] B.M. Dodin and S.E. Elmaghraby, *Approximating the criticality indices of the activities in PERT networks*, Oper. Res. 32 (1984), 493–515.

[116] W.P. Dotson and J.O. Gobien, *A new analysis technique for probabilistic graphs*, IEEE Trans. Circuits and Systems CAS-26 (1979), 855–865.

[117] P. Douilliez and E. Jamoulle, *Transportation networks with random arc capacities*, R.A.I.R.O. 3 (1972), 45–49.

[118] R.J. Duffin, *Topology of series-parallel networks*, J. Math. Anal. Appl. 10 (1965), 303–318.

[119] M.C. Easton and C.K. Wong, *Sequential destruction method for Monte Carlo evaluation of system reliability*, IEEE Trans. Reliability, R-29 (1980), 27–32.

[120] J. Edmonds, *Minimum partition of a matroid into independent subsets*, J. Res. Nat. Bur. Stand. 69B (1965), 67–72.

[121] J. Edmonds, *Optimum branchings*, J. Res. Nat. Bur. Stand. 71B (1967), 233-240.

[122] J. Edmonds, *Matroid partition*, in: Mathematics of the Decision Sciences (G.B. Dantzig and A.F. Viennott, editors), American Mathematical Society, 1968, pp. 335–345.

[123] J. Edmonds, *Edge-disjoint branchings*, in: Combinatorial Algorithms (R. Rustin, editor), Algorithmics Press, 1972, pp. 91–96.

[124] J. Edmonds and R.M. Karp, *Theoretical improvements in algorithmic efficiency for network flow problems*, J. ACM 19 (1972), 248-264.

[125] S.E. Elmaghraby, Activity Networks: Project Planning and Control by Network Models, J. Wiley & Sons (1977).

[126] S.E. Elmaghraby, *On the expected duration of PERT type networks*, Management Science 13 (1967), 299–306.

[127] S.E. Elmaghraby, *The estimation of some network parameters in the PERT model of activity networks: review and critique*, in: Advances in Project Scheduling (R. Słowiński and J.

[128] S.E. Elmaghraby, J. Kamburowski and M.F.M. Stallmann,*On the reduction of acyclic digraphs and its applications*, OR Report 233, Operations Research Series, North Carolina State University, Raleigh, NC, 1989.

[129] E.S. El Mallah and C.J. Colbourn, *Reliability of* Δ*–Y networks*, Proc. Sixteenth Southeastern Conf. Combinatorics, Graph Theory, Computing 1985, pp. 49–54.

[130] T. Elperin, I. Gertsbakh and M.V. Lomonosov, *Network reliability estimation using graph evolution models*, IEEE Trans. Reliability R–40 (1991), 572–581.

[131] G.V. Epifanov, *Reduction of a plane graph to an edge by a star–triangle transformation*, Soviet Math. Doklady 166 (1966), 13–17.

[132] P. Erdös and A. Renyi, *On random graphs I*, Publ. Math. Debrecen 6 (1959), 290–297.

[133] P. Erdös and A. Renyi, *On the evolution of random graphs*, Magyar Tud. Akad. Mat. Kutató Int. Közl. 5 (1960), 17–61.

[134] J.D. Esary and F. Proschan, *Coherent structures of non-identical components*, Technometrics 5 (1963), 191-209.

[135] J.R. Evans, *Maximum flow in probabilistic graphs – the discrete case*, Networks 6 (1976), 161–183.

[136] S. Even and R.E. Tarjan, *Network flow testing and graph connectivity*, SIAM J. Computing 4 (1975), 507–518.

[137] T.A. Feo and R. Johnson, *Partial factoring: an efficient algorithm for approximating two–terminal reliability on complete graphs*, IEEE Trans. Reliability R–39 (1990) 290–295.

[138] T.A. Feo and J.S. Provan, *Delta-wye transformations and the efficient reduction of two-terminal planar graphs*, Oper. Res., to appear.

[139] D.D. Fisher, D. Saisi and W.M. Goldstein, *Stochastic PERT networks: OP diagrams, critical paths and the project completion time*, Computers and Oper. Res. 12 (1985), 471–482.

[140] G.S. Fishman, *Estimating network characteristics in stochastic activity networks*, Management Science 31 (1985), 579–593.

[141] G.S. Fishman, *A comparison of four Monte Carlo methods for estimating the probability of (s,t)-connectedness*, IEEE Trans. Reliability R–35 (1986), 145–155.

[142] G.S. Fishman, *A Monte Carlo sampling plan for estimating network reliability*, Oper. Res. 34 (1986), 581–594.

[143] G.S. Fishman, *A Monte Carlo sampling plan for estimating reliability parameters and related functions*, Networks 17 (1987), 169–186.

[144] G.S. Fishman, *The distribution of maximum flow with applications to multistate reliability systems* Oper. Res. 35 (1987), 607–618.

[145] G.S. Fishman, *Estimating the s–t reliability function using importance and stratified sampling*, Oper. Res. 37 (1989), 462–473.

[146] G.S. Fishman, *Monte Carlo estimation of the maximum flow distribution in a network with discrete stochastic arc capacity levels*, Naval Research Logistics Quarterly 36 (1989), 829–849.

[147] G.S. Fishman, *How errors in component reliability affect system reliability*, Oper. Res. 38 (1990), 728–732.

[148] G.S. Fishman and V.G. Kulkarni, *Improving Monte Carlo efficiency by increasing variance*, Report UNC/OR/TR 90-21, University of North Carolina, 1990.

[149] G.S. Fishman and V.G. Kulkarni, *Bounds on conditional reliability*, Report UNC/OR/TR 90-22, University of North Carolina, 1991.

[150] G.S. Fishman and T.D. Shaw, *Evaluating reliability of stochastic flow networks*, Probability in the Engineering and Informational Sciences 3 (1989), 493–509.

[151] T. Flanagan, *Fiber Network Survivability*, IEEE Communications Magazine 28 (1990), 46-53.

[152] C.C. Fong and J.S. Buzacott, *An algorithm for symbolic reliability calculation with pathsets — or cutsets*, IEEE Trans. Reliability R–36 (1987), 34–37.

[153] H. Feng and S.-P. Chan, *A method of reliability evaluation for computer–communication networks*, Proc. 1990 IEEE International Symp. Circuits and Systems (1990), pp. 2682–2684.

[154] L.R. Ford and D.R. Fulkerson, Flows in Networks, Princeton University Press, 1962.

[155] H. Frank, *Shortest paths in probabilistic networks*, Oper. Res. 17 (1969), 583–599.

[156] H. Frank and W. Chou, *Network properties of the ARPA computer network*, Networks 4 (1974), 213–239.

[157] H. Frank and I.T. Frisch, *Communication, Transmission and Transportation Networks*, Addison-Wesley, 1971.

[158] H. Frank and S.L. Hakimi, *Probabilistic flow through a communications network*, IEEE Trans. Circuit Theory CT-12 (1965), 413–414.

[159] H. Frank, R.E. Kahn and L. Kleinrock, *Computer Communication Network Design-Experience with Theory and Practice*, AFIPS Conference Proceedings 40 (1972), pp. 255-270.

[160] O. Frank and W. Gaul, *On reliability in stochastic graphs*, Networks 12 (1982), 119–126.

[161] P. Frankl, *A new short proof for the Kruskal-Katona theorem*, Discrete Math. 48 (1984), 327–329.

[162] L. Fratta and U.G. Montanari, *A recursive method based on case analysis for computing network terminal reliability*, IEEE Trans. Communications COM–26 (1978), 1166–1177.

[163] L. Fratta and U.G. Montanari, *A Boolean algebra method for computing the terminal reliability in a communication network*, IEEE Trans. Circuit Theory CT–20 (1973), 203–211.

[164] H.L. Frisch, J.M. Hammersley and D.J.A.Welsh, *Monte Carlo estimates of percolation probabilities for various lattices*, Physics Review 126 (1962), 949–951.

[165] Y. Fu and S.S. Yau, *A note on the reliability of communication networks*, J. SIAM 10 (1962), 469–474.

[166] S. Fujishige, *A capacity-rounding algorithm for the minimum cost circulation problem: a dual framework of the Tardos algorithm*, Math. Programming 35 (1986), 298–308.

[167] D.R. Fulkerson, *Expected critical path lengths in PERT Networks*, Oper. Res. 10 (1962), 808–817.

[168] D.R. Fulkerson, *Networks, frames, blocking systems*, in: Mathematics of the Decision Sciences (G.B. Dantzig, A.F. Viennott, editors), American Math. Society, 1968, pp. 303–334.

[169] D.R. Fulkerson, *Blocking and anti-blocking pairs of polyhedra*, Math. Programming 1 (1971), 168–194.

[170] D.R. Fulkerson, *Packing rooted directed cuts in a weighted directed graph*, Math. Programming 6 (1974), 1–13.

[171] D.R. Fulkerson and G.H. Harding, *On edge-disjoint branchings*, Networks 6 (1976), 97–104.

[172] V. Gadasin, *Reliability of communication networks represented by semi-oriented graphs*, Engineering Cybernetics 11 (1973), 66–76.

[173] V. Gadasin, *Reliability of networks with semi-oriented structure*, Engineering Cybernetics 14 (1976), 98–104.

[174] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP–completeness, Freeman, San Francisco, 1979.

[175] B. Gavish and K. Altinkemer, *Backbone Network Design Tools with Economic Tradeoffs*, ORSA J. Computing 2 (1990), 236-252.

[176] E.N. Gilbert, *Random graphs*, Ann. Math. Stat. 30 (1959), 1141–1144.

[177] R.E. Gomory and T.C. Hu, *Multiterminal network flows*, SIAM J. Applied Math. 9 (1961), 551-570.

[178] M. Gondran and M. Minoux, Graphs and Algorithms, Wiley–Interscience, 1984.

[179] C. Greene and D.J. Kleitman, *Proof techniques in the theory of finite sets*, in: Studies in Combinatorics (G.-C. Rota, editor), MAA, 1978, pp. 22–79.

[180] G.R. Grimmett and D.J.A. Welsh, *Flow in networks with random capacities*, Stochastics 7 (1982), 205–229.

[181] M. Grötschel, L. Lovász, and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, Combinatorica 1 (1981), 169–197.

[182] J.N. Hagstrom, *Combinatorial Properties for Directed Network Reliability with a Path Length Criterion*, Technical Report 82-14, College of Business Administration, University of Illinois, Chicago (1983).

[183] J.N. Hagstrom, *Using the decomposition tree of a network in reliability computation*, IEEE Trans. Reliability R–32 (1983), 71–78.

[184] J.N. Hagstrom, *Using the decomposition tree for directed network reliability computations*, IEEE Trans. Reliability R–33 (1984), 390–395.

[185] J.N. Hagstrom, *Note on Independence of Arcs in Antiparallel for Network Flow Problems*, Networks 14 (1984), 567-570.

[186] J.N. Hagstrom, *Computational complexity of PERT Problems*, Networks 18 (1988), 139–147.

[187] J.N. Hagstrom, *Computing the probability distribution of project duration in a PERT network*, Networks 20 (1990), 231–244.

[188] J.N. Hagstrom, *Directed network reliability: domination and computing coefficients of the success–marginal expansion*, Networks 20 (1990), 65–78.

[189] J.N. Hagstrom, *Component state dependence and error in reliability computation*, preprint, 1990.

[190] J.N. Hagstrom, *Redundancy, substitutes and complements in system reliability*, preprint, University of Illinois, 1990.

[191] J.N. Hagstrom and P. Kumar, *Reliability computation on a probabilistic network with Path Length Criterion*, Technical Report, College of Business Administration, University of Illinois, Chicago (1984).

[192] T. Hailperin, *Best possible inequalities for the probability of a logical function of events*, Amer. Math. Monthly 72 (1965), 343–359.

[193] J.M. Hammersley, *A Monte Carlo solution of percolation in the cubic crystal*, Methods in Computational Physics 1 (1963), 281–298.

[194] J.M. Hammersley and D.C. Handscomb, Monte Carlo Methods, Methuen & Co. Ltd., London (1964).

[195] E. Hänsler, *A fast recursive algorithm to calculate the reliability of a communication network*, IEEE Trans. Communications COM-20 (1972), 637–640.

[196] E. Hänsler, G.K. McAuliffe, and R.S. Wilkov, *Exact calculation of computer network reliability*, Networks 4 (1974), 95–112.

[197] S. Hariri and C.S. Raghavendra, *SYREL: A symbolic reliability algorithm based on path cutset methods*, IEEE Trans. Computers C-36 (1987), 1224–1232.

[198] D.D. Harms and C.J. Colbourn, *The Leggett bounds for network reliability*, IEEE Trans. Circuits and Systems CAS-32 (1985), 609–611.

[199] D.D. Harms and C.J. Colbourn, *Renormalization of two-terminal reliability*, 1991.

[200] J.J. Harms and C.J. Colbourn, *Probabilistic single processor scheduling*, Disc. Appl. Math. 27 (1990), 101–112.

[201] H.O. Hartley and A.W. Wortham, *A statistical theory for PERT critical path analysis*, Man. Sci. 12 (1966), B469–B481.

[202] K.J. Hayhurst and D.R. Shier, *A factoring approach for the stochastic shortest path problem*, Oper. Res. Letters 10 (1991), 329–334.

[203] A.J. Hoffman and J.B. Kruskal, *Integral boundary points of convex polyhedra*, in: Linear Inequalities and Related Systems (H.W. Kuhn and A.W. Tucker, editors), Princeton University Press, 1956, pp. 233–246.

[204] E. Höpfinger and V. Steinhardt, *On the Exact Evaluation of Finite Activity Networks with Stochastic Durations of Activities*, Lecture Notes in Economics and Mathematical Systems, Vol. 117, Optimization and Operations Research, Springer-Verlag, New York.

[205] T.C. Hu, Integer Programming and Network Flows, Addison–Wesley, 1969.

[206] A.B. Huseby, *A unified theory of domination and signed domination with application to exact reliability computations*, Statistical Research Report 3, University of Oslo, 1984.

[207] A.B. Huseby, *Domination theory and the Crapo β-invariant*, Networks 19 (1989), 135–149.

[208] A.B. Huseby, *On regularity, amenability and optimal factoring strategies for reliability computations*, preprint, University of Oslo, 1990.

[209] F.K. Hwang and A.M. Odlyzko, *A probability inequality and its application to switching networks*, Bell Syst. Tech. J. 56 (1977), 821–826.

[210] K.Hwang and T.P. Chang, *Combinatorial reliability analysis of multiprocessor computers*, IEEE Trans. Reliability R–31 (1982), 469–473.

[211] I.M. Jacobs, *Connectivity in probabilistic graphs*, Technical Report 356, Electronics Research Laboratory, MIT, 1959.

[212] F. Jaeger, D. Vertigan, and D.J.A. Welsh, *On the computational complexity of the Jones and Tutte polynomials*, Math. Pr.oc. Camb. Phil. Soc. 108 (1990), 35–53.

[213] S.P. Jain and K. Gopal, *An efficient algorithm for computing global reliability of a network*, IEEE Trans. Reliability R–37 (1988) 488-492.

[214] P.A. Jensen and M. Bellmore, *An algorithm to determine the reliability of a complex system*, IEEE Trans. Reliability R–18 (1969), 169–174.

[215] M. Jerrum, *On the complexity of evaluating multivariate polynomials*, Ph.D. thesis, Department of Computer Science, University of Edinburgh, 1981; also Report CST–11–81, Computer Science, University of Edinburgh, 1981.

[216] W.S. Jewell, *Optimal flow through networks*, Interim Report 8, Massachusetts Institute of Technology, 1958.

[217] R. Johnson, *Network reliability and acyclic orientations*, Networks 14 (1984), 489–506.

[218] J. Kamburowski, *Bounds in temporal analysis of stochastic networks*, Found. Control Engrg. 10 (1985), 177–189.

[219] J.Kamburowski, *On the computational complexity of the shortest route and maximum flow problems in stochastic networks*, Found. Control Engrg. 11 (1986), 167–175.

[220] J. Kamburowski, *An overview of the computational complexity of the PERT shortest route and maximum flow problems in stochastic networks*, World Sci. Publishing (1987), 187–196.

[221] J. Kamburowski, *PERT networks under incomplete probabilistic information*, Advances in project scheduling (R. Slowinski and J. Weglarz,eds.), Elsevier Scientific Publishing Co. (1989), 433–466.

[222] N. Karmarkar, *A new polynomial time algorithm for linear programming*, Combinatorica 4 (1984), 373–396.

[223] R.M. Karp, *Reducibility among combinatorial problems*, in: Complexity of Computer Computations (R.E. Miller, J.W. Thatcher, editors), Plenum, 1972, pp. 85–103.

[224] R.M. Karp and M. Luby, *Monte Carlo algorithms for the planar multiterminal network reliability problem*, J. Complexity 1 (1985), 45–64.

[225] G. Katona, *A theorem of finite sets*, in: Theory of Graphs (P. Erdös, G. Katona, editors), Akademia Kiadó, Budapest, 1966, pp. 187–207.

[226] V.A. Kaustov, Ye.I. Litvak and I.A. Ushakov, *The computational effectiveness of reliability estimates by the method of nonedge-intersecting chains and cuts*, Soviet J. Computer Systems Sciences 24 (1986), 70-73.

[227] A.K. Kel'mans, *Some problems of network reliability analysis*, Automation and Remote Control 26 (1965), 564–573.

[228] A.K. Kel'mans, *Connectivity of probabilistic networks*, Automation and Remote Control 29 (1967), 444–460.

[229] A.K. Kel'mans, *On estimation of the probabilistic characteristics of random graphs*, Automation and Remote Control 32 (1970), 1833–1839.

[230] A.K. Kel'mans, *Asymptotic formulas for the probability of k-connectedness of random graphs*, Theory Prob. Appl. 27 (1972), 243–254.

[231] A.K. Kel'mans, *The graph with the maximum probability of remaining connected depends on the edge-removal probability*, Graph Theory Newsletter 9 (1979), 2–3.

[232] A.K. Kel'mans, *On graphs with randomly deleted edges*, Acta Math. Acad. Sci. Hung. 37 (1981), 77–88.

[233] A.K. Kel'mans, *On the analysis and synthesis of probabilistic networks*, Selected Transl. Math. Stat. Prob. (AMS) 16 (1985), 127–138.

[234] H. Kesten, *Surfaces with minimal random weights and maximal flows: a higher-dimensional version of first-passage percolation*, Illinois J. Math. 31 (1987), 99–166.

[235] L.G. Khachiyan, *A polynomial algorithm in linear programming*, Soviet Math. Doklady 20 (1979), 191–194.

[236] Y.H. Kim, K.E. Case, and P.M. Ghare, *A method for computing complex system reliability*, IEEE Trans. Reliability R–21 (1972), 215–219.

[237] N.M. Kini, A.Kumar, and D.P. Agrawal, *Quantitative reliability analysis of redundant multistage interconnection networks*, Reliability of Computer and Communications Networks, pages 153–170. AMS/ACM, 1991.

[238] G. Kirchoff, *Über die Auflösung der Gleichungen auf welche man bei der Untersuchung der Linearen Verteilung Galvanischer Ströme Geführt wird*, Poggendorg's Ann. Phys. Chem. 72 (1847), 497–508. (=*On the solution of equations obtained from the investigation of the linear distribution of galvanic currents*, IRE Trans. Circuit Theory CT–5 (1958), 4–7.),

[239] W. K. Klein-Hanevelid, *Distributions with known marginals and duality of mathematical programming with application to PERT*, Proc. seventh conference on probability theory, VNU Sci. Press (1985), 221–236.

[240] W. K. Klein-Hanevelid, *Robustness against dependence in PERT: an application of duality and distribution with known marginals*, Math. Programming Stud. (1986), 153–182.

[241] G.B Kleindorfer, *Bounding distributions for a stochastic acyclic network*, Oper. Res. 19 (1971), 1586–1601.

[242] M. Kraetzl and C.J. Colbourn, *Transformations on channel graphs*, IEEE Trans. Communications, to appear.

[243] E.V. Krishnamurthy and G. Komissar, *Computer-aided reliability analysis of complicated networks*, IEEE Trans. Reliability R-21 (1972), 86–89.

[244] J.B. Kruskal, *The number of simplices in a complex*, in: Mathematical Optimization Techniques (R. Bellman, editor), University of California Press, 1963, 251–278.

[245] P. Kubat, *Reliability Analysis for Integrated Networks with Application to Burst Switching*, IEEE Trans. Communications, COM-34 (1986), 564–568.

[246] P. Kubat, *Estimation of Reliability for Communication/Computer Networks-Simulation/Analytic Approach*, IEEE Trans. Communications, COM-37 (1989), 927–933.

[247] P. Kubat, U. Sumita and Y. Masuda, *Dynamic Performance Evaluation of Communication/Computer Systems with Highly Reliable Components*, Probability in the Engineering and Informational Sciences 2 (1988), 185–213.

[248] V.G. Kulkarni, *Shortest paths in networks with exponentially distributed arc lengths*, Networks 16 (1986), 255–274.

[249] V.G. Kulkarni, *Minimum spanning trees in undirected networks with exponentially distributed arc weights*, Networks 18 (1988), 111–124.

[250] V.G. Kulkarni, *Generating random combinatorial objects*, J. Algorithms 11 (1990), 185–207.

[251] V.G. Kulkarni and V.G. Adlakha, *Maximum flow in networks with exponentially distributed arc capacities*, Stochastic Models 1 (1985), 263–290.

[252] V.G. Kulkarni and V.G. Adlakha, *Markov and Markov-regenerative PERT networks*, Oper. Res. 34 (1987), 769–781.

[253] V.G. Kulkarni and J.S. Provan, *An improved implementation of conditional Monte Carlo estimation of path lengths in stochastic networks*, Oper. Res. 33 (1985), 1389–1393.

[254] H. Kumamoto, K. Tanaka, and K. Inoue, *Efficient evaluation of system reliability by Monte Carlo method*, IEEE Trans. Reliability R-26 (1977), 311–315.

[255] K. Kumamoto, K. Tanaka, K. Inoue and E.J. Henley, *Dagger sampling Monte-Carlo for system unavailability evaluation*, IEEE Trans. Reliability R–29 (1980), 122–125.

[256] Y.F. Lam and V.O.K. Li, *Reliability modeling and analysis of communications networks with dependent failures*, IEEE Trans. Communications COM–34 (1986), 82–84.

[257] Y.F. Lam and V.O.K. Li, *An improved algorithm for performance analysis of networks with unreliable components*, IEEE Trans. Communications COM–34 (1986), 496–497.

[258] E.L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart, and Winston, 1976.

[259] C.Y. Lee, *Analysis of switching networks*, Bell Syst. Tech. J. 34 (1955), 1287–1315.

[260] S.H. Lee, *Reliability evaluation of a flow network*, IEEE Trans. Reliability R29 (1980), 24–26.

[261] J.D. Leggett, *Synthesis of reliable networks*, Ph.D. thesis, Moore School of Engineering, University of Pennsylvania, 1968.

[262] J.D. Leggett and S.D. Bedrosian, *Synthesis of reliable networks*, IEEE Trans. Circuit Theory CT–16 (1969), 384–385.

[263] A.B. Lehman, *Wye-delta transformations in probabilistic networks*, SIAM J. 11 (1963), 773-805.

[264] L.I. Levy and A.H. Moore, *A Monte Carlo technique for obtaining system reliability confidence limits from test data*, IEEE Trans. Reliability R-16 (1967), 69–72.

[265] V.O.K. Li and J. A. Silvester, *Performance Analysis of Networks with Unreliable Components*, IEEE Trans. Communications COM-32 (1984), 1105-1110.

[266] P.M. Lin, B.J. Leon, and T.C. Huang, *A new algorithm for symbolic system reliability analysis*, IEEE Trans. Reliability R–25 (1976), 2–15.

[267] Ye.I. Litvak, *The probability of connectedness of a graph*, Engineering Cybernetics 13 (1975), 121–125.

[268] Ye.I. Litvak, *A generalized triangle-star transformation of properties of complex networks*, Engineering Cybernetics 19 (1981), 158–162.

[269] Ye.I. Litvak, *Two-sided estimates of the minimum cost of a flow in two-terminal networks*, Engineering Cybernetics 19 (1981), 132–135.

[270] E.I. Litvak, *A generalized theorem on negative cycles and estimates of the quality of flow transport in a network*, Soviet Math. Doklady 27 (1983), 369-371.

[271] Ye.I. Litvak and I.A. Ushakov, *Estimation of the parameters of structurally complex systems*, Engineering Cybernetics 22 (1984) 35–49.

[272] C.I. Liu and Y. Chow, *Enumeration of connected spanning subgraphs of a planar graph*, Acta Mathematica Hungarica 41 (1983), 27–36.

[273] M.O. Locks, *Recursive disjoint products, inclusion–exclusion, and min–cut approximations*, IEEE Trans. Reliability R–29 (1980), 368–371.

[274] M.O. Locks, *Recursive disjoint products: a review of three algorithms*, IEEE Trans. Reliability R–31 (1982), 33–35.

[275] M.V. Lomonosov, *Bernoulli scheme with closure*, Problems of Information Transmission 10 (1974), 73–81.

[276] M.V. Lomonosov, *Tender-spot of a reliable network*, preprint, Ben Gurion University of the Negev, 1989.

[277] M.V. Lomonosov and V.P. Polesskii, *An upper bound for the reliability of information networks*, Problems of Information Transmission 7 (1971), 337–339.

[278] M.V. Lomonosov and V.P. Polesskii, *Lower bound of network reliability*, Problems of Information Transmission 8 (1972), 118–123.

[279] M.V. Lomonosov and V.P. Polesskii, *Maximum of the connectivity probability*, Problems of Information Transmission 8 (1972), 324–328.

[280] R.P. Loui, *Optimal paths in graphs in graphs with stochastic or multidimensional weights* Comm. ACM 26 (1983), 670–676.

[281] L. Lovász, *On two minimax theorems in graph theory*, J. Combinatorial Theory B21 (1976), 96–103.

[282] F.S. Macaulay, *Some properties of enumeration in the theory of modular systems*, Proc. London Math. Soc. 26 (1927), 531–555.

[283] D.G. Malcolm, J.H. Roseboom, C.E. Clark, and W. Fazar, *Application of a technique for research and development program evaluation*, Oper. Res. 7 (1959), 646–669.

[284] J.J. Martin, *Distribution of the Time Through a Directed Acyclic Network*, Oper. Res. 13 (1965), 46–66.

[285] E. Mata-Montero, *Reliability of partial k-tree networks*, Ph.D. thesis, University of Oregon, 1990.

[286] K. Menger, *Zur allgemeine Kurventheorie*, Fund. Math. 10 (1927), 96–115.

[287] J.F. Meyer, *Performability: a retrospective and some pointers to the future*, Performance Evaluation 14 (1992), 139–156.

[288] M. Mihail and A.L. Buchsbaum, *Monte Carlo and Markov chain techniques for network reliability and sampling*, preprint, 1992.

[289] H. Mine, *Reliability of physical systems*, IRE Trans. Circuit Theory CT–6 (1959), 138–151.

[290] P.B. Mirchandani, *Shortest distance and reliability of probabilistic networks*, Computers and Oper. Res. 3 (1976), 347–355.

[291] P. Mirchandani and Hossein Soroush, *Stochastics in combinatorial optimization*, World Scientific Publishing Co., (1987), 128–177.

[292] K.B. Misra, *An algorithm for the reliability of redundant networks*, IEEE Trans. Reliability R–19 (1970), 146–151.

[293] K.B. Misra and T.S.M. Rao, *Reliability analysis of redundant networks using flow graphs*, IEEE Trans. Reliability R–19 (1970), 19–24.

[294] D. Monhor, *An approach to PERT: application of Dirichlet distribution*, Optimization 18 (1987), 113–118.

[295] E.F. Moore, *The Shortest Path through a Maze*, Ann. Computational Lab. Harvard Univ. 30 (1959), 285-292.

[296] E.F. Moore and C.E. Shannon, *Reliable circuits using less reliable relays*, J. Franklin Inst. 262 (1956), 191–208; 263 (1956), 281–297.

[297] F. Moskowitz, *The analysis of redundancy networks*, AIEE Trans. Commun. Electron. 39 (1958), 627–632.

[298] W.J. Myrvold, K.H. Cheung, L. Page and J. Perry, *Uniformly reliable graphs do not always exist*, Networks 21 (1991) 417–419.

[299] A. Nádas, *Probabilistic PERT*, IBM Jour. Res. and Dev. 23 (1979), 339–347.

[300] H. Nagamochi and T. Ibaraki, *Maximum flows in probabilistic networks*, Networks 21 (1991), 645-666.

[301] H. Nakazawa, *Equivalence of a nonoriented line and a pair of oriented lines in a network*, IEEE Trans. Reliability R–28 (1979), 364–367.

[302] C.St.J.A. Nash–Williams, *Edge-disjoint spanning trees of finite graphs*, J. London Math, Soc. 36 (1961), 445–450.

[303] C.St.J.A. Nash-Williams, *Decomposition of finite graphs into forests*, J. London Math. Soc. 39 (1964), 12.

[304] L.D. Nel and C.J. Colbourn, *Locating a facility in an unreliable network*, INFOR 26 (1990), 363-379.

[305] L.D. Nel and C.J. Colbourn, *Combining Monte Carlo estimates and bounds for network reliability*, Networks 20 (1990), 277–298.

[306] L.D. Nel and H.J. Strayer, *Two-terminal reliability bounds based on edge-packings by cutsets*, J. Comb. Math. Comb. Comput., to appear.

[307] A.C. Nelson, J.R. Batts, and R.L. Beadles, *A computer program for approximating system reliability*, IEEE Trans. Reliability R–19 (1970), 61–65.

[308] E.M. Neufeld and C.J. Colbourn, *The most reliable series-parallel networks*, Networks 15 (1985), 27–32.

[309] J. von Neumann, *Probabilistic logics and the synthesis of reliable organisms from unreliable components*, in: Automata Studies, 1956, pp. 43-98.

[310] V.F. Novikov, A.L. Raikin and Yu.B. Shufchuk, *The effect of reliability on communication delays in a network with maximal flows*, Automation and Remote Control 49 (1988), 1613–1619.

[311] K. Onaga, *Bounds on the average terminal capacity of probabilistic nets*, IEEE Trans. Reliability R 35 (1986), 252–259.

[312] L.B. Page and J.E. Perry, *A practical implementation of the factoring theorem for network reliability*, IEEE Trans. Reliability R–37 (1988), 259–267.

[313] L.B. Page and J.E. Perry, *Reliability of directed networks using the factoring theorem*, IEEE Trans. Reliability R–38 (1989), 556–562.

[314] K.P. Parker and E.J. McCluskey, *Probabilistic treatment of general combinational networks*, IEEE Trans. Computers C–24 (1975), 668–670.

[315] N. Pippenger, *Developments in "The synthesis of reliable organisms from unreliable components"*, Proc. AMS Symp. Pure Math. 50 (1990) 311–324.

[316] V.P. Polesskii, *A lower boundary for the reliability of information networks*, Problems of Information Transmission 7 (1971), 165–171.

[317] V.P. Polesskii, *Bounds on probability of connectedness of a random graph*, Problems of Information Transmission 26 (1990), 75–82.

[318] V.P. Polesskii, *Bounds on probability of group connectedness of a random graph*, Problems of Information Transmission 26 (1990), 161–169.

[319] T. Politof, *A characterization and efficient reliability computation of Δ-Y reducible networks*, Ph.D. thesis, University of California at Berkeley, 1983.

[320] T. Politof and A. Satyanarayana, *Efficient algorithms for reliability analysis of planar networks - a survey*, IEEE Trans. Reliability R–35 (1986), 252–259.

[321] T. Politof and A. Satyanarayana, *Network reliability and inner-four-cycle-free graphs*, Math. Oper. Res. 7 (1986), 97–111.

[322] T. Politof and A. Satyanarayana, *A linear-time algorithm to compute the reliability of planar cube-free networks*, IEEE Trans. Reliability R–39 (1990), 557–563.

[323] A. Prékopa and E. Boros, *On the existence of a feasible flow in a stochastic transportation network*, Oper. Res. 39 (1991), 119–129.

[324] A. Prékopa, E. Boros and K.-W. Lih, *The use of binomial moments for bounding network reliability*, in: *Reliability of Computer and Communications Networks*, AMS/ACM, 1991, pp. 197–212.

[325] G.M. Provan, *A logic-based analysis of Dempster-Shafer theory*, Int. J. Approx. Reason. 4 (1990), 451–495.

[326] J.S. Provan, *Polyhedral combinatorics and network reliability*, Math. Oper. Res. 11 (1986), 36–61.

[327] J.S. Provan, *Bounds on the reliability of networks*, IEEE Trans. Rel. R–35 (1986), 260–268.

[328] J.S. Provan, *The complexity of reliability computations in planar and acyclic graphs*, SIAM J. Computing 15 (1986), 694–702.

[329] J.S. Provan, *Boolean decomposition schemes and the complexity of reliability computations*, in: *Reliability of Computer and Communications Networks*, AMS/ACM, 1991, pp. 213–228.

[330] J.S. Provan and M.O. Ball, *The complexity of counting cuts and of computing the probability that a graph is connected*, SIAM J. Computing 12 (1983), 777–788.

[331] J.S. Provan and M.O. Ball, *Computing network reliability in time polynomial in the number of cuts*, Oper. Res. 32 (1984), 516–526.

[332] J.S. Provan and L.J. Billera, *Decompositions of simplicial complexes related to diameters of convex polyhedra*, Math. Oper. Res. 5 (1980), 579–594.

[333] J.S. Provan and V.G. Kulkarni, *Exact cuts in networks*, Networks 19 (1989), 281–289.

[334] S. Rai, *A cutset approach to reliability evaluation in communication networks*, IEEE Trans. Reliability R–31 (1982), 428–431.

[335] S. Rai and K.K. Aggarwal, *An efficient method for reliability evaluation of a general network*, IEEE Trans. Reliability R–27 (1978), 206–211.

[336] V. Raman, *Finding the best edge-packing for two-terminal reliability is NP-hard*, J. Comb. Math. Comb. Computing 9 (1991), 91–96.

[337] A. Ramanathan and C.J. Colbourn, *Bounds on all–terminal reliability via arc–packing*, Ars Combinatoria 23A (1987), 91–94.

[338] A. Ramanathan and C.J. Colbourn, *Counting almost minimum cutsets and reliability applications*, Math. Programming 39 (1987), 253–261.

[339] A. Ramesh, M.O. Ball, and C.J. Colbourn, *Bounding all–terminal reliability in planar networks*, Annals of Discrete Math. 33 (1987), 261–273.

[340] L.I.P. Resende, *Implementation of a factoring algorithm for reliability evaluation of undirected networks*, IEEE Trans. Reliability R–37 (1988), 462–468.

[341] M.G.C. Resende, *A program for reliability evaluation of undirected networks via polygon-to-chain reductions*, IEEE Trans. Reliability R–35 (1986), 24–29.

[342] L.J. Ringer, *Numerical operators for statistical PERT critical path analysis*, Man. Sci. 16 (1969), 136–143.

[343] L.J. Ringer, *A statistical theory for PERT in which completion times of activities are interdependent*, Man. Sci. 17 (1971), 717-723.

[344] J.T. Robacker, *Minmax theorems on shortest chains and disjoint cuts of a network*, Memo RM–1660–PR, The Rand Corporation, 1956.

[345] P. Robillard and M. Trahan, *Expected Completion Time in PERT Networks*, Oper. Res. 24 (1977), 177–182.

[346] A. Rosenthal, *A computer scientist looks at reliability computations*, in: Reliability and Fault Tree Analysis SIAM, 1975, pp. 133–152.

[347] A. Rosenthal, *Computing the reliability of complex networks*, SIAM J. Appl. Math. 32 (1977), 384–393.

[348] A. Rosenthal and D. Frisque, *Transformations for simplifying network reliability calculations*, Networks 7 (1977), 97–111.

[349] J. Roskind and R.E. Tarjan, *A note on finding minimum cost edge-disjoint spanning trees*, Math. Oper. Res. 10 (1985), 701–708.

[350] R.A. Sahner and K.S. Trivedi, *Performance and reliability analysis using directed acyclic graphs*, IEEE Trans. Software Engineering 10 (1987), 1105–1114.

[351] B. Sanso, M. Gendreau and F. Soumis, *An Algorithm for Network Dimensioning under Reliability Consideration*, Centre de Recherche sur les Transports Research Report, Universite de Montreal (1990).

[352] B. Sanso and F. Soumis, *Communication and Transportation Networks Reliability Using Routing Models*, IEEE Trans. Reliability R-40 (1991), 29–38.

[353] B. Sanso and F. Soumis, *A Centralized Optimization Routing Model and Applications in Telecommunication Networks*, Centre de Recherche sur les Transports Research Report #CRT-657, Universite de Montreal (1990).

[354] B. Sanso, F. Soumis and M. Gendreau, *On the Evaluation of Telecommunications Network Reliability Using Routing Models*, Centre de Recherche sur les Transports Research Report #CRT-623, Universite de Montreal (1990).

[355] A. Satyanarayana, *A unified formula for analysis of some network reliability problems*, IEEE Trans. Reliability R-31 (1982), 23–32.

[356] A. Satyanarayana and M.K. Chang, *Network reliability and the factoring theorem*, Networks 13 (1983), 107–120.

[357] A. Satyanarayana and J.N. Hagstrom, *A new algorithm for the reliability analysis of multi-terminal networks*, IEEE Trans. Reliability R-30 (1981), 325–334.

[358] A. Satyanarayana and J.N. Hagstrom, *Combinatorial properties of directed graphs useful in computing network reliability*, Networks 11 (1981), 357–366.

[359] A. Satyanarayana and Z. Khalil, *On an invariant of graphs and the reliability polynomial*, SIAM J. Alg. Disc. Meth. 7 (1986), 399–403.

[360] A. Satyanarayana and A. Prabhakar, *New topological formula and rapid algorithm for reliability analysis of complex networks*, IEEE Trans. Reliability R-27 (1978), 82–100.

[361] A. Satyanarayana, L. Schoppmann and C.L. Suffel, *A reliability-improving graph transformation with applications to network reliability*, Networks 22 (1992) 209–216.

[362] A. Satyanarayana and R.K. Wood, *A linear time algorithm for computing k-terminal reliability in series-parallel networks*, SIAM J. Comput. 14 (1985), 818–832.

[363] J.G. Shanthikumar, *Reliability of systems with consecutive minimal cutsets*, IEEE Trans. Reliability 36 (1987), 546–550.

[364] J.G. Shanthikumar, *Bounding network reliability using consecutive minimal cutsets*, IEEE Trans. Reliability R-37 (1988), 45–49.

[365] D.R. Shier, *Iterative algorithms for calculating network reliability*, in: Graph Theory with Applications to Algorithms and Computer Science, Wiley, 1985, pp. 741–752.

[366] D.R. Shier, *A new algorithm for performance analysis of communication systems*, IEEE Trans. Commun. COM-36 (1988), 516–519.

[367] D.R. Shier, *Network Reliability and Algebraic Structures*, Oxford University Press, Oxford and New York, 1991.

[368] D.R. Shier, *Algebraic methods for bounding network reliability*, in: *Reliability of Computer and Communications Networks*, AMS/ACM, 1991, pp. 245–259.

[369] D.R. Shier, E. Bibelnieks, J.P. Jarvis and R.J. Lakin, *Algorithms for approximating the performance of multimode systems*, Proc. IEEE Infocom 90 (1990), pp. 741–748.

[370] D.R. Shier and J.D. Spragins, *Exact and approximate dependent failure models for telecommunications networks*, Proceedings of INFOCOM85, 1985, pp. 200–205.

[371] D.R. Shier and D.E. Whited, *Algebraic methods applied to network reliability problems*, SIAM J. Algebraic Discrete Methods 8 (1987), 251–262.

[372] A.W. Shogan, *Sequential bounding of the reliability of a stochastic network*, Oper. Res. 24 (1976), 1027–1044.

[373] A.W. Shogan, *Bounding distributions for a Stochastic PERT network*, Networks 7 (1977), 359–381.

[374] A.W. Shogan, *A recursive algorithm for bounding network reliability*, IEEE Trans. Reliability R-26 (1977), 322–327.

[375] A.W. Shogan, *A decomposition algorithm for network reliability analysis*, Networks 8 (1978), 231–251.

[376] A.W. Shogan, *Modular decomposition and reliability computation in stochastic transportation networks having cutnodes*, Networks 12 (1982), 255–275.

[377] A.M. Shooman and A. Kershenbaum, *Methods for communication-network reliability analysis: probabilistic graph reduction*, Proc. 1992 Reliability and Maintainability Symposium, pp. 441–448.

[378] M.L. Shooman, Probabilistic Reliability: An Engineering Approach, McGraw–Hill, New York, 1968.

[379] C.E. Sigal, The stochastic shortest route problem, Ph.D. Dissertation, Purdue University.

[380] C.E. Sigal, A.A.B. Pritsker, and J.J. Solberg, *The stochastic shortest route problem*, Oper. Res. 28 (1980), 1122–1129.

[381] C.E. Sigal, A.A.B. Pritsker, and J.J. Solberg, *The use of cutsets in Monte Carlo analysis of stochastic networks*, Math. Computer Simulation 21 (1980), 376–384.

[382] S. Soh and S. Rai, *CAREL: Computer aided reliability evaluator for distributed computing networks*, IEEE Trans. Parallel Distributed Systems 2 (1991) 199–213.

[383] I.M. Soi and K.K. Aggarwal, *Reliability indices for topological design of computer communication networks*, IEEE Trans. Reliability R–30 (1981), 438–443.

[384] J.E. Somers, *Maximum flow in a network with a small number of random arc capacities*, Networks 12 (1982), 242-253.

[385] H.G. Spelde, *Bounds for the distribution function of network variables*, First Symposium of Operations Research (University of Heidelberg), 3 (1977), 113–123.

[386] E. Sperner, *Ein Satz über Untermengen einer endlichen Menge*, Mathematische Zeitschrift 27 (1928), 544–548.

[387] E. Sperner, *Über einen kombinatorischen Satz von Macaulay und seine Anwendung auf die Theorie der Polynomideale*, Abh. Math. Sem. Univ. Hamburg 7 (1930), 149–163.

[388] J.D. Spragins, *Dependent failures in data communication systems*, IEEE Trans. Communications COM–25 (1977), 1494–1499.

[389] J.D. Spragins, J.C. Sinclair, Y.J. Kang and H. Jafari, *Current telecommunication network reliability models: a critical assessment*, IEEE Trans Selected Areas Communications SAC–4 (1986), 1168–1173.

[390] R.P. Stanley, *The Upper Bound Conjecture and Cohen–Macaulay rings*, Studies in Applied Math. 14 (1975), 135–142.

[391] R.P. Stanley, *Cohen–Macaulay complexes*, in: Higher Combinatorics (M. Aigner, editor), Reidel, 1977, pp. 51–64.

[392] R.P. Stanley, *Hilbert functions of graded algebras*, Advances in Math. 28 (1978), 57–83.

[393] R.P. Stanley, Combinatorics and Commutative Algebra, Birkhäuser Verlag, 1983.

[394] R.P. Stanley, *An introduction to combinatorial commutative algebra*, in: Enumeration and Design (D.M. Jackson, S.A. Vanstone, editors), Academic Press, 1984, pp. 3–18.

[395] V.E. Stepanov, *Combinatorial algebra and random graphs*, Theory of Probability and Its Applications 14 (1969), 373–399.

[396] E. Tardós, *A strongly polynomial minimum cost circulation algorithm*, Combinatorica 5 (1985), 247–256.

[397] R.E. Tarjan, *A good algorithm for edge-disjoint branching*, Information Processing Letters 3 (1974), 51–53.

[398] O.R. Theologou and J.G. Cartier, *Factoring and reductions for networks with imperfect vertices*, IEEE Trans. Reliability R–40 (1991), 210–217.

[399] R.K. Tiwari and M. Verma, *An algebraic technique for reliability evaluation*, IEEE Trans. Reliability R–29 (1980), 311–313.

[400] L. Traldi, *On the star–delta transformation in network reliability*, preprint, Lafayette College.

[401] L. Traldi, *Generalized activities and K-terminal reliability*, preprint, Lafayette College.

[402] K. Truemper, *On the delta-wye reduction of planar graphs*, J. Graph Theory 13 (1989), 141-148.

[403] T. Tsuboi and K. Aruba, *A new approach to computing terminal reliability in large complex networks*, Electr. Commun. Japan 58A (1975), 52–60.

[404] S. Tsukiyama, I. Shirakawa, H. Ozaki, and H. Ariyoshi, *An algorithm to enumerate all cutsets of a graph in linear time per cutset*, J. ACM 27 (1980), 619–632.

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

[406] I. Ushakov and Ye.I. Litvak, *An upper and lower estimate of the parameters of two–terminal networks*, Engineering Cybernetics 15 (1977).

[407] L.G. Valiant, *The complexity of computing the permanent*, Theoretical Computer Science 8 (1979), 189-201.

[408] L.G. Valiant, *The complexity of enumeration and reliability problems*, SIAM J. Computing 8 (1979), 410–421.

[409] E.J. Valvo, D.R. Shier and R.E. Jamison, *Generating the most probable states of a communication system*, Proc. INFOCOM87 (1987), 1128–1136.

[410] A.Varma and C.S. Raghavendra, *Reliability analysis of redundant-path interconnection networks*, IEEE Trans. Reliability R–38 (1989), 130–137.

[411] R.M. Van Slyke, *Monte Carlo methods and the PERT problem*, Oper. Res. 11 (1963), 839–860.

[412] R.M. Van Slyke and H. Frank, *Network reliability analysis: part I*, Networks 1 (1972), 279–290.

[413] M. Veeraraghavan and K.S. Trivedi, *An improved algorithm for symbolic reliability analysis*, IEEE Trans. Reliability R–40 (1991) 347–358.

[414] D. Vertigan, *The computational complexity of Tutte invariants for planar graphs*, preprint, Mathematical Institute, Oxford University.

[415] D. Vertigan, *Bicycle dimension and special points of the Tutte polynomial*, preprint, Mathematical Institute, Oxford University.

[416] B.Ya. Vtorova-Karevskaya and Ye.I. Litvak, *The reliability of systems admitting two kinds of failures*, Engineering Cybernetics 17 (1979).

[417] D.K. Wagner, *Disjoint (s,t)-cuts in a network*, Networks 20 (1990), 361–372.

[418] J.A. Wald and C.J. Colbourn, *Steiner trees, partial 2-trees, and minimum IFI networks*, Networks 13 (1983), 159–167.

[419] J.A. Wald and C.J. Colbourn, *Steiner trees in probabilistic networks*, Microelectronics and Reliability 23 (1983), 837–840.

[420] G. Weiss, *Stochastic bounds on distributions of optimal value functions with applications to PERT, network flows and reliability*, Oper. Res. 34 (1986), 595–605.

[421] D.J.A. Welsh, Matroid Theory, Academic Press, 1976.

[422] F. Whipple, *On a theorem due to F.S. Macaulay*, J. London Math. Soc. 8 (1928), 431–437.

[423] D.E. Whited, D.R. Shier and J.P. Jarvis, *Reliability computations for planar networks*, ORSA J. Computing 2 (1990), 46–60.

[424] H. Whitney, *A logical expansion in mathematics*, Bull. Amer. Math. Soc. 38 (1932), 572–579.

[425] R.R. Willie, *A theorem concerning cyclic directed graphs with applications to network reliability*, Networks 10 (1980), 71–78.

[426] O. Wing and P. Demetriou, *Analysis of probabilistic networks*, IEEE Trans. Commun. Tech. COM–12 (1964), 38–40.

[427] R.T. Wong, *A dual ascent approach for Steiner tree problems on a directed graph*, Math. Programming 28 (1984), 271–287.

[428] R.K. Wood, *Polygon-to-chain reductions and extensions for reliability evaluation on undirected networks*, Ph.D. thesis, University of California at Berkeley, 1982.

[429] R.K. Wood, *A factoring algorithm using polygon-to-chain reductions for computing k-terminal network reliability*, Networks 15 (1985), 173–190.

[430] R.K. Wood, *Triconnected decomposition for computing K-terminal network reliability*, Networks 19 (1989), 203–220.

[431] C.-L. Yang and P. Kubat, *Efficient computation of most probable states for communication networks with multimode components*, IEEE Trans. Communications, COM-37 (1989), 535–538

[432] C.-L. Yang and P. Kubat, *An algorithm for network reliability bounds*, ORSA J. Computing 2 (1990), 336–345.

[433] R. Yarlagadda and J. Hershey, *Fast algorithm for computing the reliability of a communication network*, Int. J. Electron. 70 (1991) 549–564.

[434] Y.B. Yoo and N. Deo, *A comparison of algorithms for terminal-pair reliability*, IEEE Trans. Reliability R-37 (1988), 210–215.

[435] E. Zemel, *Polynomial algorithms for estimating network reliability*, Networks 12 (1982), 439–452.