AN INVESTIGATION INTO BOUNDS ON NETWORK RELIABILITY

Daryl D. Harms

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

Research Report CS-84-55
December 1984

An Investigation Into Bounds on Network Reliability

A Thesis
Submitted to the Faculty of Graduate Studies and Research
In Partial Fulfillment of the Requirements
For the Degree of

Master of Science

in the

Department of Computational Science
University of Saskatchewan
Saskatoon, Saskatchewan

by

Daryl Dean Harms

December 1983

ABSTRACT

Substantial research effort on computer networks has been invested in network reliability. One measure of reliability is the probability that a network is operating, in that each of its nodes is able to communicate with every other node. In the graph theoretic network model which we adopt, nodes do not fail and edges have statistically independent but equivalent probability of failure. The probability that the graph remains connected in this environment is a standard measure of reliability, often termed probabilistic connectedness. The exact calculation of probabilistic connectedness is #P-complete in general. Therefore, much work has been done on upper and lower bounds for probabilistic connectedness. These bounds are of varying degrees of sophistication, but can all be evaluated in polynomial time. The primary difficulty in their implementation is the computation of the graph parameters which they use.

Little previous comparative work has been done on these bounds to determine their relative merits and applicability. We first show that one of the apparently more promising sets of these bounds (Leggett's bounds) is seriously in error, and therefore cannot be employed. Most of the remaining bounds have been devised using the same general method of bounding certain subgraph counts (states in which the graph is connected). The bounds based on subgraph counts form a strict hierarchy with respect to accuracy; from least to most accurate, they are the Jacobs bounds, the Bauer-Boesch-Suffel-Tindell bounds, the Kruskal-Katona bounds, and the Ball-Provan bounds. The final set of bounds (Lomonosov-Polesskii bounds) is based on an entirely different approach. Although seemingly less sophisticated than the best subgraph bounds, a surprising fact is that there are a significant number of cases where the Lomonosov-Polesskii bounds actually improve on the Ball-Provan bounds.

The main conclusion of the thesis is that the best set of bounds currently available is therefore a complementary combination of the Lomonosov-Polesskii and Ball-Provan bounds.

Acknowledgements

The preparation and writing of a thesis is never a completely solitary undertaking. As a result of the short time frame employed, I relied more heavily than usual on the assistance and patience of others. I would like to express my appreciation to some of these people.

If present, a congenial working environment may often be taken for granted. Its absence, on the other hand, would be sorely felt. The support of my fellow graduate students has created a pleasant atmosphere to work in. Specifically, I must thank Eric Neufeld, not only for the time and work he put in and moral support he gave me during the final preparation of my thesis, but also for his friendship and the help he has given me ever since I joined the department. I would like to thank Daniel Zlatin for making his keys and system expertise available at all hours seven days a week. The graduate students in this department owe him a large debt. I would also like to thank my family for their continued support and understanding, with special thanks to my sister Janelle for her help in those "last few days".

I would like to thank the members of my committee Marlene Colbourn, Rick Bunt and my external Roy Billinton for their efforts and advice. I especially thank them for their patience with the abbreviated time span they were given for reading the thesis, at a time which was already very busy for each of them.

Finally, I would like to thank my supervisor Charlie Colbourn for his support and direction in all facets of this project from start to finish. This includes everything from him being always available for discussions on the general and specific ideas of the thesis to his contacts with others in the field and help in obtaining the difficult references; as well the many hours he spent reading through the early drafts, and subsequently the numerous suggestions and corrections he made on the format and style of the thesis This thesis would not have been completed without him.

Table of Contents

Chapter 1

Introduction and Terminology

1.1. Introduction

The last decade has seen the rapid growth of communication networks. Interconnected mainframe computers or microprocessor based centers communicate through high speed data transmission links. These range from local networks providing communication services within specific offices to continent-wide networks, such as Arpanet and Telenet. This appears to be just the beginning of a tremendous proliferation of communication networks.

As communications networks become more prevalent, more and more services and systems depend on them. This dependence brings an ever increasing need for *reliability*. Users, and society at large, will require more reliable service. The telephone and the electrical power industries are examples of systems on which modern society has become increasingly dependent, and in which service failures or interruptions can have serious economic and social consequences.

There is no debate on whether the qualitative concept of developing reliable networks is of prime importance. A problem comes, however, in defining and measuring what is meant by "reliability", and more specifically the "reliability of a network". We want more than just the subjective connotations of the adjective "reliable". It is therefore desirable to define the reliability of a network quantitatively. Such a measure would allow the comparison of various networks, which would be very useful in the design of networks. One method of quantifying reliability is to equate it with a statistical *probability*. A common definition of reliability is "the probability that a system or device operates for a given period of time when used under the stated conditions." Difficulty arises in determining what situations to consider as "operating". A network is designed at many different levels; failures may occur in each. We will be looking only at the topology. Even at this level, there are many notions of system operation. Here we consider a network is operating when each of its communication centers is able to communicate with every other center.

The networks are modeled using a graph theoretic approach. This allows the use of the extensive amount of graph-theoretic research. Employing a graph model still leaves us the serious problem of computing the reliability. Unfortunately, obtaining an exact solution is not computationally feasible in general. However, it turns out that it is possible to efficiently obtain upper and lower bounds on the exact solution. If these are tight enough to show relative differences between the various options a designer may be considering, they can be of practical use. A number of different sets of bounds of various degrees of sophistication have been developed. The main purpose of the thesis is to investigate these bounds. A more formal discussion necessitates the introduction of certain graph-theoretic tools.

1.2. Graph Theoretic Definitions

We use graphs to model networks. Unfortunately, graph-theoretic structures and definitions are not standardized, and this necessitates our defining terms. We only define terms used in this thesis, and refer the reader to [H2,T3] for other definitions.

We model a network as a graph $G = (V,E)$. V is a set of *vertices* or *nodes* and E is the set of *edges* of the graph represented as (un)ordered pairs of vertices. Unordered pairs represent undirected edges and indicate two way connections. Undirected edges are represented (v_i, v_j); v_i and v_j are the two nodes upon which the edge is incident. A graph of undirected edges is generally referred to as a non-oriented or *undirected graph*. The ordered pair $<v_i, v_j>$ represents a directed edge indicating a one way connection between node v_i and v_j. Graphs of directed edges are termed directed graphs or *digraphs*. We use n to denote $|V|$, the number of nodes in the graph and b to denote $|E|$, the number of edges in the graph. Multiple or *parallel edges* are edges between the same two nodes. A *loop* or self loop is an edge originating and terminating at the same node. It is always assumed there are no loops; unless otherwise mentioned, we consider undirected graphs with no parallel edges. Figure 1 shows a small graph for which b=7 and n=5.

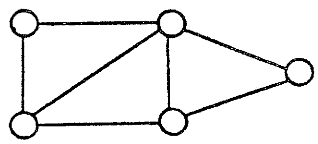

Figure 1

Two nodes are *adjacent* if there exists a common edge incident upon both of them. For an undirected graph, the *degree* of node v_i, denoted by d_i, is the number of edges incident upon v_i. If d_i is equal for all i, G is called a *regular* graph. A sequence of adjacent nodes $v_i, v_j, \ldots, v_n$ with no nodes repeated forms a *path* between v_i and v_n. Two paths between nodes v_i and v_n are *edge-disjoint* if they contain no common edges. A *circuit* is a path with the same first and last node. A graph G is *connected* if there exists at least one path between every pair of nodes v_i and v_n where v_i, $v_j \in V$. A *subgraph* G_1 of G is a graph whose nodes and edges are contained in G. That is, $G_1 = (V_1, E_1)$ where $V_1 \subseteq V$ and $E_1 \subseteq E$. If $V_1 = V$, the subgraph G_1 is a *spanning subgraph* of G. In a graph, a connected spanning subgraph consisting of n–1 edges is a *spanning tree* . The number of spanning trees in a graph is denoted as t.

A set of edges of a graph whose removal disconnects the graph is termed an *edge cutset*. The minimum number of edges in any edge cutset of the graph is called the *edge connectivity* or *cohesion* of the graph and is denoted as c. The cohesion of the graph in Figure 1 is equal to two. A cutset is an *i-j cut* if the removal of its edges breaks all paths between nodes v_i and v_j. The minimum number of edges in any i–j cut is the edge connectivity between nodes v_i and v_j.

As mentioned, we primarily consider undirected graphs with no multiple edges. However, some of the algorithms examined do manipulate digraphs and utilize multiple edges. Therefore, the following specific concepts are also required.

For a digraph, the *indegree* of a node is the number of edges directed into it, and its *outdegree* is the number of edges originating from it. An *acyclic* graph is a graph in which there are no circuits. For an undirected acyclic graph, the *successors* of a node v_i are all nodes v_j in G for which there is a path from v_i to v_j. A *strongly connected* component is a maximal set of nodes S in which for every pair of nodes $v_i, v_j \in S$ there exists paths from v_i to v_j and v_j to v_i which only contain nodes in S. The operation of *deleting* a node v_i from $G = (V,E)$ is the process of removing v_i from V and all arcs containing v_i from E. The operation of *collapsing* a set of nodes S consists of removing these nodes from V, removing all edges between two of these nodes from E, adding a new node v_s to V and replacing all remaining edges into or out of nodes in S with corresponding edges into or out of v_s. This process may create parallel edges.

1.3. Overview of the Thesis

The thesis is, on the whole, concerned with analyzing bounds for the reliability of communications networks. To achieve this, we must first develop a formal model for network reliability. Therefore, in the previous section a number of graph theoretic definitions were given; in this study, networks are modelled using probabilistic graphs. In chapter 2, we develop the parameters of this model and discuss the effects of, and rationale behind, the assumptions that are embedded within it. The determination of the measure chosen involves evaluating the probability that each network node can communicate with each other node. We survey the research that has been done on this measure, and observe that it is not feasible to obtain the exact value for this measure in general. This has motivated the development of a number of methods of obtaining sets of bounds. In chapter 2, we also introduce the available sets of bounds: Jacobs, BBST, Kruskal-Katona, Leggett, Ball-Provan, and Lomonosov-Polesskii.

In chapter 3, we show that one of the seemingly more promising of these sets of bounds (Leggett's bounds) is actually incorrect. Computational evidence of errors in these bounds is introduced. Then Leggett's fundamental theorem is shown to be in error; this renders his bounds useless. A number of minor errors in Leggett's analysis are also described.

Chapter 4 looks at the implementation of the remaining sets of bounds. Implementation is necessary to assess the relative merits of each bound, and to determine the difficulty of implementing and evaluating each bound. It is shown that each set of bounds, and all the values that they use, can be obtained in polynomial time; this requires the discussion of efficient algorithms for computing the number of spanning trees, the edge connectivity, the number of minimal edge cutsets, and a minimal cut basis. The relative difficulty of implementing the various bounds, as well as the specifics of the actual implementations, is discussed.

In chapter 5 the bounds are tested on a number of graphs. We develop a family of test cases for the bounds; these test cases consist primarily of graphs for which the actual reliability can be efficiently computed. Comparisons are made between the sets of bounds to determine their relative merits and the absolute performance of the bounds. Finally, the effect of various graph operations on the accuracy of the bounds is studied.

In chapter 6, a more detailed summary of the thesis is given, highlighting both the conclusions reached, and the original contributions of the thesis. Finally, numerous suggestions are made regarding the areas where it appears that future research would be fruitful.

Chapter 2

Problem Description and Historical Background

2.1. Problem Definition

The different approaches to defining and measuring the reliability of communications networks have been surveyed by Frank and Frisch [F1] and by Wilkov [W3]. The methods can be divided into two fairly distinct areas: deterministic and probabilistic. Deterministic methods [B7,E3,T1] define the reliability of a network by discrete measures, such as the number of edges and/or nodes that must be removed to disconnect the graph. An application of these methods is in a military environment where an intelligent enemy is attacking the network. On the other hand, probabilistic methods generally assume that the failure of links and/or nodes is due to random causes. They measure the *probability* that the network remains operative; for example, one probabilistic measure gives the probability that a certain pair of nodes can communicate, given predetermined probabilities that the edges and/or nodes are operative. In this thesis, we focus on the probabilistic approach.

Most of the work in the probabilistic area has been done on the two-terminal problem [H4]. This is the problem of computing the probability that there will be at least one path between a given pair of nodes. The main application for this measure has *not* been for communication networks. The systems modeled often consist of such things as groups of machines, devices or jobs which are represented by the graph edges. These are interconnected so that success of the system occurs whenever there is at least one path between the first given node (often referred to as the source) and the second given node (often referred to as the sink). For most of these systems, only the links correspond to any physical entities that may have a probability of failure. Different links may represent completely different types of devices with, very different probabilities of failure. Finally, the links must often be represented as directional, to accurately model actual systems. Applying this measure to communications networks, one obtains the probability that a certain pair of nodes can communicate. The implication here is that the connection between these two nodes is the only communication of importance.

However, it may often be the case that communication is essentially among peers. This implies that nodes are to be treated as equals. Two measures that are applicable in this situation are 1) the probability that every node is able to communicate with every other node, and 2) the expected number of node pairs that are able to communicate. Van Slyke and Frank [V3] note that, for networks where the probability that some node or group of nodes is disconnected is quite high (i.e., measure 1 is low), measure 2 might be more useful, as it measures the degree of "partial usefulness" of the network. In this study we look primarily at measure 1, assuming that networks are in acceptable states only when total communication is possible.

2.2. Model Definition

The model used is a *probabilistic graph* consisting of b undirected edges representing the duplex links between the communication centres and n nodes representing the communication centres. Nodes are assumed to be perfectly reliable (i.e. have no probability of failure). The edges all have an equal but statistically independent probability p of being available. Conversely, we define $q=1-p$ as the statistically independent probability of an edge being in a failed state. The network is operative when it is connected. Hence, the measure for reliability is the probability that all the nodes are able to communicate with each other. For our undirected graph representation, this corresponds to the probability that the available edges of the network contain at least a spanning tree, or conversely that the failed edges do not contain a network cut. This measure is often referred to as *probabilistic connectedness.* We represent this probability for a graph as R.

This measure does not take into account such factors as link capacities, delays, or other service parameters of the network. It just assumes a node "can communicate" with another as long as there is a possible physical route between them (a path consisting of available edges).

It is also important to clarify what is meant by the "operating probability" of an edge. This might be the probability that a link of the network will operate for a given period of time. As well, it might be the steady state probability of finding a link in an operating state in a network where links fail and are subsequently repaired. In this second case, this probability is often referred to as the *availability* of the link. Our model handles either case. If p is the probability of a link operating for a two year period, R is taken as the probability of the network operating for a two year period. If p is the availability of a link, R can be referred to as the reliability of the maintained system.

Two major assumptions are made about the link operation probabilities. One is that they are equal for all the links. The assumption here is that the network is constructed of similar links; consequently they have equivalent failure probabilities. Ideally, these values are obtained using statistical analysis of past failures of similar links under similar circumstances. Failure probabilities are generally quite low which means that a large number of links over a large period of time must be used to obtain a statistically significant sample of failures. Important factors are the actual type of link and the environment in which it is placed. It is not generally feasible to obtain separate reliability values for each individual link in a network. If all the links are of the same type and their environments are similar, it can be assumed fairly accurately that the reliability (probability of being in the operating state) is the same for all the links.

The next assumption is that of statistical independence; the probability of a link existing in a state is independent of the states of the other links in the network. The assumption is that the links fail due to random factors that affect them individually. For a real communication network, this assumption is not always valid: there may be cases where topologically separate links share a common duct for a distance, or where separate links in the same general area fail due to common events such as major natural occurrences (such as storms or earthquakes). Without this assumption, the calculation of network reliability is greatly complicated as all the pertinent conditional probabilities need to be considered and known (see for example [Z1]). Finally, it is worthwhile to note that statistical independence has often been assumed to model systems where it is well known that the failures are not independent. An example of this is the classic treatment of telephone crossbar switching networks by Lee [L1].

At first glance, the assumption that the nodes are perfectly reliable may not seem at all valid for communication networks, as the nodes in a communication network do consist of actual devices that can be assumed to exhibit failure rates comparable to those of the links of the network. This point is discussed by Van Slyke and Frank [V3]. They suggest that the problem depends on how we define our measure of the probability of being connected. They give two options.

First, we can consider a network to be connected when all its operational nodes can communicate. This method causes a number of problems. The failure probability of the nodes will be different from that of the links. Thus, we no longer have the simple case where all the components have equal availabilities. More seriously, we no longer have a problem that is coherent. A reliability problem is *coherent* if when a network is in a failed state, no set of further component failures will return it to an operating state. Take the graph G with the two subgraphs G_a and G_b in Figure 2.

If the network is in the state where the nodes and links as shown in subgraph G_a are the ones that are available, the network is considered to be disconnected. But if nodes 4, 5, 6, and 7 all fail, the network moves into the state indicated by subgraph G_b. The network would now be considered to be connected. It has moved back into an operating state.

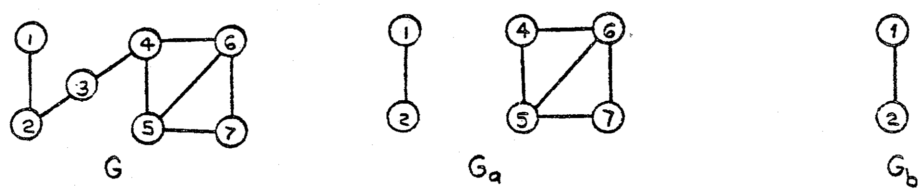

Figure 2

However, intuitively, state G_a seems to be more desirable than state G_b. In fact, when the operational probability of the nodes approaches zero, the reliability approaches one [V3]. The counterintuitive nature of this measure leads us to adopt Frank and Van Slyke's second proposed measure.

In this second approach, we consider that a network is connected when all its nodes can communicate with each other. Using this definition the network is disconnected whenever any node or set of nodes fails. As stated in [V3] the reliability of the network for this definition then becomes:

$$Reliability = (NA)(R) \tag{1}$$

where NA= prob(all nodes are up), and R= the reliability when nodes are perfectly reliable.

For a designer synthesizing a network, given a set of nodes to connect, NA is a fixed constant. Therefore, the measure of interest is R. One can always trivially determine NA as it is $\prod_{i=1}^{n} \rho_i$ where ρ_i is the operational probability of node v_i. In the rest of this paper, we only consider R as the measure of the reliability of the network. However, it might be more accurate to call it the link reliability of the network.

2.3. Probabilistic Connectedness

The basic pioneering work in probabilistic connectedness is by Moore and Shannon [M3]. They investigate the problem of constructing arbitrarily reliable networks from arbitrarily poor components. They actually consider a somewhat different type of model than the one just developed, but their results can be modified to apply to it. They model circuits of electromechanical relays using graphs consisting of undirected edges with independent and equal operating probabilities representing the relay contacts, and perfectly reliable nodes corresponding to the interconnections between the relays. They assume that the relays are in their useful life period and failures occur because of random events such as dust getting between the contacts, and are therefore constant with respect to time. Since the failures are random, they are statistically independent. The nodes do not fail since they are only looking at the failures of the contacts after the circuit has been correctly wired. From this, they develop the following important polynomial:

$$h(p) = \sum_{n=0}^{m} A_n p^n q^{m-n} \tag{2}$$

where $h(p)$= probability of the network being closed, p= the probability of a contact being closed, q=1−p= the probability of a contact being open, and A_n = the number of ways one can select a subset of n of the m contacts in the network such that if these n contacts are closed, and the remaining contacts open, then the network will be closed. This can similarly be written as:

$$h(p) = 1 - \sum_{n=0}^{m} B_n q^n p^{m-n} \tag{3}$$

where B_n= the number of subsets of n contacts such that, if all contacts in a subset are open and the other contacts closed, the network is open.

Kel'mans [K4] investigates the probabilistic connectedness of communications networks. He modified these polynomials to apply to this problem, to obtain the *reliability polynomial*:

$$R = \sum_{i=0}^{b} N_i p^i q^{b-i} \tag{4}$$

or

$$R = 1 - \sum_{i=0}^{b} C_i q^i p^{b-i} \tag{5}$$

N_i is the number of selection of exactly i of the b edges that connect the graph (the number of spanning subgraphs containing i edges). C_i is the number of selections of exactly i of the b edges which form an edge cutset or network cut. The relationship between these two equations is illustrated by the fact that:

$$N_i + C_{b-i} = \binom{b}{i}$$

Therefore, from any N_i value we can easily determine the corresponding C_{b-i} value and vice versa. Another important observation is:

$$0 \leq N_i \leq \binom{b}{i} \quad \text{and} \quad 0 \leq C_i \leq \binom{b}{i}$$

Although the reliability polynomial forms a clean and concise representation, the actual calculation of R is quite difficult. The ideal situation would be to easily calculate the exact reliability of any given network. A problem arises in calculating the reliability of general graphs. The brute force method of performing this calculation is to enumerate all the possible subgraphs of the given graph and sum up the probabilities of those which are connected (that is, to use the reliability polynomial). However, since there are b edges, each either working or failed, there are 2^b possible states that the network can be in; hence this method is of exponential complexity.

A lot of work has been put into improving on this. Important work appears in Mine [M2] and Moskowitz [M4]. They exploit what is usually referred to as the "factoring theorem":

$$R(G) = q[R(G-b_{ij})] + p[R(G \bullet b_{ij})] \tag{6}$$

$R(G)=R$ for the original graph G. $R(G-b_{ij})=R$ for the graph in which the edge b_{ij} has been removed from graph G. This corresponds to the case where edge b_{ij} is failed. $R(G \bullet b_{ij})=R$ for the graph in which the nodes v_i and v_j of graph G have been collapsed. This corresponds to the case where the edge b_{ij} is operating.

This formula factors out an edge of the network, and subsequently examines the reliability of the resulting smaller subnetworks. Although this yields an improvement over the case enumeration method, the use of this theorem still requires an exponential enumeration for general graphs. A number of algorithms with further refinements exist for computing the reliability of general graphs [B1,B8,H1,R2,S1,S2]. The orders of computation time required by them are significantly lower than the complete case enumeration method. However, even the fastest [B8] is still $O(3^{n-1})$. In order to obtain this, the algorithm has storage requirements of $O(2^{n-1})$. In fact, it has been shown that the exact calculation for general graphs is almost certainly intractable.

Valiant [V1] proved that computing the probability that a specific pair of nodes can communicate is #P-complete. Ball [B2] showed that finding either a rational value or a generating function for probabilistic connectedness is NP-hard. Finally, Provan and Ball [P4] and Jerrum [J2] showed that probabilistic connectedness is #P-complete. Therefore, it appears that computing R exactly is not practical for general graphs.

There have been formulae or algorithms developed for calculating the exact reliability of certain special types of graphs in polynomial time. Gilbert [G1] develops the following recurrence for calculating the reliability of a complete graph:

$$R(n) = 1 - \sum_{i=1}^{n-1} \binom{n-1}{i-1} R(i) q^{i(n-i)} \tag{7}$$

$R(i)=R$ for a complete graph on i nodes. $R(1)$ with no edges will, of course, equal one. Wald and Colbourn [W1,W2] develop algorithms which can be used to calculate R for 2-trees or any graph that is a subgraph of a 2-tree. As well, Neufeld and Colbourn [N2,N3] gives recurrences for the most reliable 2-trees and closely related graphs.

Unfortunately, most actual networks do not fall into these categories. It is typically not possible to obtain the exact solution of R during the topological design of medium and large-scale networks. Therefore, some method of determining approximations for R is desirable. There are two routes that can be taken here. The first option is to use simulation. A number of simulation methods are discussed in [F1,V2,V3]. The desired result of a simulation is an approximate value and a confidence interval. Nevertheless, there is no absolute guarantee that the actual reliability will fall into this interval. The second option is to use analytic methods to approximate R, or more ideally to obtain a pair of bounds (upper and lower) on R. A set of analytic bounds guarantees (given the validity of the assumptions) that the actual reliability will fall between them. In order to be of practical use, these bounds should be obtainable in polynomial time. It is desirable as well that they be sensitive enough to discriminate between the reliabilities of relatively similar alternatives that a designer may be considering.

Assumptions about the value of p can be an important parameter of these approximations or bounds. Their importance is illustrated by Kel'mans' [K4] result that it is possible to find two graphs, one of which has a higher R than the other for certain values of p but lower R for other values of p.

2.4. Analytic Approximation

The essential problem is to develop bounds on the reliability polynomial:

$$R = \sum_{i=0}^{b} N_i p^i q^{b-i} \tag{8}$$

or

$$R = 1 - \sum_{i=0}^{b} C_i q^i p^{b-i} \tag{9}$$

For any network of n vertices, at least n–1 edges are needed to connect them. Therefore, for $i<n-1$, $N_i=0$ and $C_{b-i}=\binom{b}{i}$. If c is the minimum number of edges that must be removed to disconnect a network (edge connectivity or cohesion) then for $i>b-c$, $N_i=\binom{b}{i}$ and $C_{b-i}=0$. The reliability polynomial becomes:

$$R = \sum_{i=n-1}^{b-c} N_i p^i q^{b-i} + \sum_{i=b-c+1}^{b} \binom{b}{i} p^i q^{b-i} \tag{10}$$

or

$$R = 1 - \sum_{i=c}^{b-n+1} C_i q^i p^{b-i} - \sum_{i=b-n+2}^{b} \binom{b}{i} q^i p^{b-i} \tag{11}$$

The relative importance of particular N_i (or C_i) terms depends on the value of p. For p=0.5 all the N_i (or C_i) are of equal importance and the formulae become:

$$R = (0.5)^b \sum_{i=n-1}^{b} N_i \tag{12}$$

or

$$R = 1 - (0.5)^b \sum_{i=c}^{b} C_i \tag{13}$$

For values of $p>0.5$ the N_i (or C_{b-i}) for large values of i are the more dominant in the polynomial with N_{b-c} (or C_c) being the most important. Kel'mans [K4] used this fact and observed that for p close to one this term dominates the polynomial, and can therefore be used to approximate it:

$$R \simeq 1 - C_c p^{b-c} q^c \tag{14}$$

Similarly, for values of $p<0.5$, the N_i (or C_{b-i}) for small values of i are the more dominant with N_{n-1} (or C_{b-n+1}) being the most important. Kel'mans observed that for p close to zero the polynomial can be approximated by:

$$R \simeq N_{n-1} p^{n-1} q^{b-n+1} \tag{15}$$

In order to calculate these approximations, the values for N_{n-1}, c, and C_c must be obtained. N_{n-1} is equal to t the number of spanning trees of the given graph. c is the previously defined edge-connectivity or cohesion of the graph. C_c is the number of minimum cardinality cuts (number of edge cutsets containing exactly c edges) of the graph. For example for the graph in Figure 3 where $c=2$, as shown with the dotted lines there are exactly three combinations of two edges which if removed will disconnect the graph. Thus $C_c=3$.

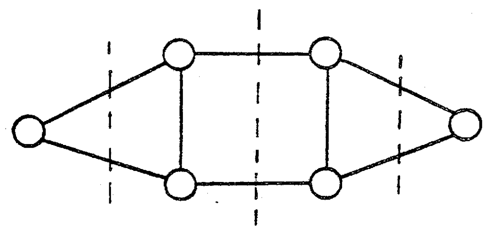

Figure 3

To be of practical use these graph theoretic values (c, t, and C_c) must of course be obtainable in polynomial time. In Chapter 4 we show that all three of these values can in fact be calculated in polynomial time.

2.5. A Survey of the Available Bounds

Kel'mans' approximations (equations 14 and 15) are only close for the extreme values of p. The approximations are only using the first terms in the respective summations; the next two terms in the summations are only one and two degrees higher. Therefore, the approximations are only fairly accurate for values of p quite close to zero and one respectively. Actually (14) gives an overapproximation or "upper bound" on R, while (15) gives an underapproximation or "lower bound" on R and therefore they can be combined to form a set of bounds on R:

$$N_{n-1}p^{n-1}q^{b-n+1} \geq R \geq 1 - C_c p^{b-c} q^c \tag{16}$$

This set of bounds is, of course, very weak as the lower and upper bounds are each only close for p values at opposite extremes. For any other p values neither of them is close. Obtaining improved bounds on R is quite desirable as they guarantee that the actual value of R falls in the range between them. Knowing the lower bound on R is particularly valuable, when designing systems where certain minimum objectives have to be met. A tight set of bounds would show a designer the relative effects of different options.

A considerable amount of effort is needed to obtain the values for c and N_{n-1} and especially C_c. Therefore, it might be worthwhile to invest a little more work to obtain a tighter set of bounds. A number of sets of bounds on R have been developed, and one purpose of this thesis is to investigate them.

From equation 10, it can be seen that once the values of N_{n-1} and N_{b-c} have been determined, what is left unknown are the N_i values for $(n-1)>i>(b-c)$. Similarly, looking at equation 11, calculating C_{b-n+1} and C_c leaves the C_i values for $(b-n+1)>i>(c)$. Unfortunately, it does not appear to be feasible to determine any of these values directly in polynomial time for general graphs. The basic idea behind almost all of the sets of bounds in the literature is to approximate these unknown values. If a set of overapproximations can be determined for N_i in this range (or underapproximations for C_i), summing these delivers an overapproximation or upper bound for R. Similarly, if a set of underapproximations can be determined for N_i in this range (or overapproximations for C_i), summing these delivers an underapproximation or lower bound for R.

2.5.1. Jacobs Bounds

The first set of bounds for R was developed by Jacobs[J1,V2] and is referred to here as the *Jacobs bounds.* In his original bounds, Jacobs only used the values for $b,n,$ and c:

$$R \leq 1 - \sum_{i=b-n+2}^{b} \binom{b}{i} p^{b-i} q^i \tag{17a}$$

and

$$R \geq 1 - \sum_{i=c}^{b} \binom{b}{i} p^{b-i} q^i \tag{17b}$$

The unknown values of C_i are overestimated as their maximum value $\binom{b}{i}$ to obtain an underestimate or lower bound on R. Similarly, they are underestimated as their minimum value of 0 to obtain an overestimate or upper bound on R.

Frank and Van Slyke [V2] modified these bounds for the case where N_{n-1} and C_c are known. These are referred to here as the *Jacobs-II bounds*:

$$R \leq 1 - \sum_{i=b-n+2}^{b} \binom{b}{i} p^{b-i} q^i - C_{b-n+1} p^{n-1} q^{b-n+1} \tag{18a}$$

and

$$R \geq 1 - \sum_{i=c+1}^{b} \binom{b}{i} p^{b-i} q^i - C_c p^{b-c} q^c \tag{18b}$$

Although these bounds improve on (16), they still suffer from the problem that the lower bound is best for p close to 0 and the upper bound is best when p is close to 1. The estimates on the unknown C_i values are as weak as possible.

2.5.2. BBST Bounds

Recently, Bauer et al. [B5] proposed a set of bounds on R. They proved that for $c \leq i \leq b-1$:

$$\frac{C_i}{\binom{b}{i}} \leq \frac{C_{i+1}}{\binom{b}{i+1}}$$

In words, this says that the percentage of a graph's subgraphs containing i edges which are edge cutsets cannot decrease as i increases. This applies to any coherent system. Using this fact the following set of bounds, which we refer to as the *BBST bounds*, can be stated:

$$R \leq 1 - \frac{C_c}{\binom{b}{c}} \left(\sum_{i=c}^{d} \binom{b}{i} p^{b-i} q^i \right) - \sum_{i=d+1}^{b} \binom{b}{i} p^{b-i} q^i \tag{19a}$$

and

$$R \geq 1 - C_c p^{b-c} q^c - \frac{C_d}{\binom{b}{d}} \left[\sum_{i=c+1}^{d} \binom{b}{i} p^{b-i} q^i \right] - \sum_{i=d+1}^{b} \binom{b}{i} p^{b-i} q^i \tag{19b}$$

where $d=b-n+1$.

Bauer et al. represent these using the binomial distribution $B_r = \sum_{i=0}^{r} \binom{b}{i} p^{b-i} q^i$. B_r can be obtained from standard tables and the bounds can be represented as:

$$R \leq 1 - \frac{C_c}{\binom{b}{c}}(B_d - B_{c-1}) - (1-B_d) \tag{20a}$$

and

$$R \geq 1 - C_c p^{b-c} q^c - \frac{C_d}{\binom{b}{d}}(B_d - B_c) - (1-B_d) \tag{20b}$$

2.5.3. Kruskal-Katona Bounds

Frank and Van Slyke [V2] showed that it is possible to apply a theorem developed by Kruskal [K8] and Katona [K2] that applies to any coherent system to obtain a set of bounds on the network reliability polynomial. In order to describe this theorem and subsequent bounds derived from it, we develop a number of combinatorial concepts. For the most part we use the notation and definitions developed in [B3]. For any non-negative integer m, the *k-canonical representation* of m is given by:

$$m = \binom{m_k}{k} + \binom{m_{k-1}}{k-1} + \dots + \binom{m_l}{l}$$

where $m_k > m_{k-1} > \dots > m_l \geq l \geq 1$. The m_i are determined successively as:

$$m_i = \max\left\{ x: \binom{x}{i} \leq m - \sum_{j=i+1}^{k} \binom{m_j}{j} \right\}$$

This k-canonical representation is unique and always exists. For $k \geq l \geq 1$ and any $i \geq k$ the *(i,k)th lower pseudopower* of $(m_k, \dots, m_l)$ is:

$$(m_k, \dots, m_l)^{i/k} = \binom{m_k}{i} + \binom{m_{k-1}}{i-1} + \dots + \binom{m_l}{i-k+l}$$

We represent $(m_k, \dots, m_l)^{i/k}$ as $m^{i/k}$. Kruskal and Katona use $F_i = \binom{b}{i} - C_i = N_{b-i}$. F_i is the number of sets of i edges that do not contain a network cut.

The theorem developed by Kruskal and Katona states that:

$$a) \quad F_k^{i/k} \geq F_i \quad ,\textit{when } i \geq k$$

$$b) \quad F_k^{i/k} \le F_i \quad ,when \ i \le k$$

Thus, knowing F_c, and using a), overapproximations can be determined for the F_i values where $i > c$. Similarly if F_{b-n+1} is known, b) can be used to find underapproximations on the F_i values where $i < b-n+1$. The following bounds on R result:

$$R \le \sum_{i=0}^{c-1} \binom{b}{i} p^{b-i} q^i + F_c p^{b-c} q^c + \sum_{i=c+1}^{d-1} f_c^{i/c} p^{b-i} q^i + F_d p^{b-d} q^d \tag{21a}$$

and

$$R \ge \sum_{i=0}^{c-1} \binom{b}{i} p^{b-i} q^i + F_c p^{b-c} q^c + \sum_{i=c+1}^{d} f_d^{i/d} p^{b-i} q^i \tag{21b}$$

where $d = b-n+1$. We refer to these as the *Kruskal-Katona* bounds.

2.5.4. Ball and Provan Bounds

Ball and Provan [B3,B4] develop a set of bounds that improve on the Kruskal-Katona bounds. They use results due to Stanley [S3] which apply to systems that are *shellable*. The set of spanning subgraphs of a graph form a construct known as a *matroid*. This is referred to as the bond-matroid or co-graphic matroid. It has been shown [P6] that all matroids are shellable. Therefore Stanley's results apply to them.

These bounds are formed in a manner which is analogous to the Kruskal-Katona bounds. A number of modified constructs must be defined in order to describe them. A different form of the reliability polynomial is used. The reliability polynomial of equation (5) can be converted to the form:

$$R = p^{n-1} \sum_{i=0}^{b} H_i q^i$$

This conversion is performed by factoring out p^{n-1} and then recombining the product of the remaining terms. The H_i terms can also be obtained directly from the F_i terms as:

$$H_i = \sum_{k=0}^{i} (-1)^{i-k} \binom{b-n+1-k}{i-k} F_k$$

for $i = 0, 1, \ldots, b-n+1$.

As it takes at least c edges to disconnect the network,

$$H_i = \binom{b+i-2}{i} \quad \text{for } i < c.$$

As it takes at least $n-1$ edges to connect the network,

$$H_i = 0 \quad \text{for } i > d \text{ , where } d = b-n+1.$$

Also, when t and C_c are known we can use:

$$H_c = \binom{b+c-2}{c} - C_c \quad \text{for } i = c$$

and

$$\sum_{i=0}^{d} h_i = t$$

The *(i,k)th upper pseudopower* of $(m_k,...,m_l)$ is:

$$(m_k,...,m_l)^{<i/k>} = \binom{m_k-k+i}{i} + \binom{m_{k-1}-k+i}{i-1} + \dots + \binom{m_l-k+i}{i-k+l}$$

Again, we represent $(m_k,...,m_l)^{<i/k>}$ as $m^{<i/k>}$. For any non-negative integers m, d, and k the *(k,d)-factor of* m is the number x for which:

$$x - x^{<k/d>} = m$$

The following bounds are derived, using the notation,

$$a = t - \sum_{i=0}^{c} h_i$$

$$r_0 = \max\left\{ r: \sum_{i=c+1}^{r} h_c^{<i/c>} \leq a \right\}$$

$$m = (a_d-1,...,a_l-1)$$

where $(a_d,...,a_l)$ is the d-canonical vector for the (k,d)-factor of a. Then:

$$R \leq p^{n-1}\left(\sum_{i=0}^{c} h_i q^i + \sum_{i=c+1}^{r_0} h_c^{<i/c>} q^i + \left[a - \sum_{i=c+1}^{r_0} h_c^{<i/c>} \right] q^{r_0+1} \right) \tag{22a}$$

and

$$R \geq p^{n-1}\left(\sum_{i=0}^{c} h_i q^i + \sum_{i=c+1}^{d} m^{<i/d>} q^i \right) \tag{22b}$$

We refer to these as the *Ball-Provan bounds*. Ball and Provan [B3] also show that by substituting $(c-1)$ for c in equations (22a) and (22b), a weaker bound can be obtained for the case where the value of C_c is not known. They note that this substitution can also be applied to the Kruskal-Katona bounds (equations 21a and 21b). The possible usefulness of these weaker bounds is evident when the difficulty involved in the calculation of C_c is considered.

Provan [P6] developed another set of bounds on R. He used the fact that the spanning subgraphs of a graph also belong to another more general class called *polyhedral systems*. Although tighter for directed graphs, these bounds are equivalent to the Ball-Provan bounds for undirected graphs [P7].

2.5.5. Leggett's Bounds

Leggett [L2,L3] also devised a set of bounds on R. Let $S_i = \binom{b}{i} - N_i = C_{b-i}$. S_i is the number of combinations of i of the b edges which do not connect the graph (the number of disconnected subgraphs containing i edges). Using S_i, the reliability polynomial (equation 10) can be rewritten as:

$$R = \sum_{i=n-1}^{b-c} \left(\binom{b}{i} - S_i \right) p^i q^{b-i} + \sum_{i=b-c+1}^{b} \binom{b}{i} p^i q^{b-i} \tag{23}$$

His general approach was to form bounds on the S_i values using graph theoretic structure.

The Kruskal-Katona bounds and the Ball-Provan bounds are obtained by applying results known for more general systems (coherent systems and shellable systems) of which these sets of spanning subgraphs or network cuts are subsets. Leggett developed the following theorem which forms the basis of his bounds:

$$S_{r+1} \le f_{n-1}\left(\frac{b-c-r}{r+1}\right) S_r \qquad (24)$$

where

$$f_{n-1} = \frac{\left(1+\frac{1}{n-1}\right)^{n-1}}{\left(1+\frac{1}{2}\right)^{2}}$$

His actual bounds are developed from this theorem. A lower bound can be obtained from the following recurrence in which f_{n-1} of the above equation is replaced by X_r:

$$S_{r+1} \le X_r\left(\frac{b-c-r}{r+1}\right) S_r \qquad (25a)$$

where:

$$X_r = \frac{\left(1+\frac{1}{r}\right)^{r}}{\left(1+\frac{1}{r-n+3}\right)^{r-n+3}}$$

Starting with the known value $S_{b-n+1} = \binom{b}{b-n+1} - t$, equation (25*a*) can be successively applied to obtain overestimates for the values of S_i in the range of $b-n+1 \geq i \geq c$. Placing these into equation (23) provides an underestimate or lower bound for R.

By substituting $r-1$ for r in equation (25*a*) and rearranging, Leggett obtained:

$$S_{r-1} \le X_{r-1}\left(\frac{r}{b-c-r+1}\right) S_r \qquad (25b)$$

If the value of S_c is known, equation (25*b*) can be successively applied to obtain overestimates for the values of S_i in the range of $c \leq i \leq b-n+1$. Placing these into equation 23 provides an overestimate or upper bound for R. We refer to these as *Leggett's bounds.*

2.5.6. Lomonosov & Polesskii Bounds

There has been work reported in the Russian literature [K5,L3,L4,P3] on the development of bounds on R. Although these bounds do not appear to be very sophisticated, they are of interest as the general approach taken is completely different from that taken for the bounds already developed. A lower bound on R can be obtained by exploiting the fact that the probability of failure of a graph can be no larger than the product of the probabilities that each member of any set of edge disjoint connected spanning subgraphs contained in it fail. This fact follows directly from the assumption of statistical independence of edge failures. In [P2], Pollesskii proves that a graph must contain at least $\left\lfloor \frac{c}{2} \right\rfloor$ edge disjoint spanning trees. This fact can also be easily shown from a stronger result independently proven by Tutte [T5] and Nash-Williams [N1]. Pollesskii [P3] uses it to develop the following simple lower bound on R:

$$R \geq 1-\left(1-p^{n-1}\right)^{\left\lfloor \frac{c}{2} \right\rfloor} \tag{26}$$

Each of the edge disjoint spanning trees must fail before the graph fails. Lomonosov and Polesskii [L4] develop this further to obtain the following improved lower bound:

$$R \geq n(1-q^{\frac{c}{2}})^{n-1}-(n-1)(1-q^{\frac{c}{2}})^{n} \tag{27a}$$

These two lower bounds only use a graph's values for n and c.

In [L4], Lomonosov and Polesskii also develop an upper bound on R. To develop this bound, we define a *cut basis* of a graph G as a set of $n-1$ maximal edge cutsets $L=\{L_1,L_2,\cdots,L_{n-1}\}$ such that every two nodes of G are disconnected by some L_i where $1\leq i\leq n-1$. Each of these L_i cutsets must form a minimal j-k cut for some pair of nodes $v_j,v_k \in G$. It follows that for the graph G to be connected *at least one* of the edges in *every* cut L_i must be available. Therefore:

$$R \leq \prod_{i=1}^{n-1}\left(1-q^{|L_i|}\right) \tag{27b}$$

We refer to equations (27a) and (27b) as the *Lomonosov-Polesskii bounds.*

2.6. Summary

We have developed the use of undirected probabilistic graphs with perfectly reliable nodes and with edges which fail with equivalent but statistically independent probabilities as a network model. Defining the reliability of the network as the probability that every node is able to communicate with every other node corresponds to the probability that the underlying graph remains connected. This measure is often referred to as probabilistic connectedness. The infeasibility of the exact calculation for probabilistic connectedness motivates the development of bounds on the reliability which are computable in polynomial time. We have introduced sets of bounds, of varying degrees of sophistication. These are the Jacobs-II, BBST, Kruskal-Katona, Ball-Provan, Leggett and Lomonosov-Polesskii bounds.

It should be noted that other bounds have been developed with different assumptions about the network model. For example, Zemel [Z1] investigates the case when statistical independence of the link failures is not assumed and Kel'mans [K5] looks at the problem when the links are allowed to have different failure probabilities.

Chapter 3

Leggett's Bounds are Incorrect

3.1. Introduction

In this chapter we take a closer look at the bounds developed by Leggett [L2,L3] and show that they are not generally correct. Leggett's general approach of using graph theoretic structure is very promising. The Kruskal-Katona bounds [K2,K7,V2] and BBST bounds [B5] require that the system be "coherent". The Ball and Provan bounds [B4,P5] are applicable to arbitrary "shellable independence systems". These approaches do not exploit any information about graph-theoretic structure other than the number of spanning trees, and the number of minimal cutsets. Improvements on the Ball-Provan bounds, if any, will likely come by cleverly exploiting some graph structure which does not hold for arbitrary shellable independence systems.

In fact, there are cases where Leggett's bounds are tighter than any of these other bounds. This is true for the ten node complete graph (K_{10}) for which Leggett published the values which his bounds deliver. Unfortunately, this "improvement" is the result of a number of fatal errors in Leggett's thesis [L2]. In this chapter, we develop Leggett's approach, and devise families of counterexamples to his bounds. The possibility of difficulties with Leggett's bounds was pointed out over ten years ago by Van Slyke and Frank [V2], who noted that Leggett states bounds which are apparently stronger than the Kruskal-Katona bounds, however "his proof seems inadequate".

3.2. Bounds on S_i

The bounds developed by Leggett were introduced in chapter 2. We take a closer look at them here. The general scheme employed in Leggett's work [L2,L3] is to devise an upper bound on S_{i+1} in terms of S_i. S_{n-1} is easily computed from the number of spanning trees; this can be used to obtain bounds on S_n, S_{n+1}, and so on. This in turn provides a lower bound for the reliability. The upper bound on S_{i+1} in terms of S_i gives a lower bound on S_i in terms of S_{i+1}. Together with S_{b-c}, the number of minimal cutsets, this provides an upper bound on the reliability polynomial.

The key to success in Leggett's approach is to obtain a very close bound on S_{i+1} in terms of S_i. Leggett developed the following theorem [L2,Thm. III.2] which forms the basis of his bounds:

$$S_{r+1} \leq f_{n-1}\left(\frac{b-c-r}{r+1}\right) S_r \tag{24}$$

where:

$$f_{n-1} = \frac{\left(1+\dfrac{1}{n-1}\right)^{n-1}}{\left(1+\dfrac{1}{2}\right)^{2}}$$

A lower bound can be obtained from the following recurrence in which f_{n-1} of the above equation is replaced by X_r:

$$S_{r+1} \leq X_r \left(\frac{b-c-r}{r+1} \right) S_r \tag{25a}$$

where:

$$X_r = \frac{\left(1+\dfrac{1}{r}\right)^r}{\left(1+\dfrac{1}{r-n+3}\right)^{r-n+3}}$$

Starting with the known value $S_{n-1} = \binom{b}{n-1} - t$, equation (25*a*) can be successively applied to obtain overestimates for the values of S_i in the range of $n-1 \leq i \leq b-c$. Placing these into equation (23) provides an underestimate or lower bound for R.

By substituting $r-1$ for r in equation (25*a*) and rearranging, Leggett obtains:

$$S_{r-1} \leq X_{r-1} \left(\frac{r}{b-c-r+1} \right) S_r \tag{25b}$$

If the value of S_c is known, equation (25*b*) can be successively applied to obtain overestimates for the values of S_i in the range of $c \leq i \leq n-1$. Placing these into equation (23) will provide an overestimate or upper bound for R.

Leggett obtains improved approximations on some of these S_i values with the claim [L2, Thm. III.3] that for graphs which are not regular

$$S_r = \sum_{i=1}^{n-1} \sigma_i \binom{b-i}{b-r-i}, \qquad \text{for } b-c \geq r \geq b-2c+1 \tag{26}$$

where σ_i is the number of nodes of degree i.

Leggett also developed "slightly weakened" but easily computed closed form expressions for his bounds. Denoting the cumulative binomial distribution from j to k as

$$B(j;k,p) = \sum_{i=j}^{k} \binom{k}{i} p^i q^{k-i}$$

The reliability of the network may be written as

$$R = B(n-1;b,p) - \sum_{r=n-1}^{b} S_r p^{b-r} \tag{27}$$

Using the relationship between the S_r values developed in (25*a*) along with (27) it can be determined that

$$R \geq B(n-1;b,p) - \left(\frac{S_{n-1} q^c}{\binom{b-c}{n-1}} \right) \sum_{r=n-1}^{b-c} W_r \binom{b}{r} p^r q^{b-r} \tag{28}$$

where $W_r = \prod_{i=n-1}^{r} X_i$.

Leggett claims [L2, App. A.2] that the W_r with the maximum value is

$$W_{b-c} \leq \left\{ \frac{n(b-c-n+1)}{2(b-c)} \right\}^{\frac{1}{2}} \tag{29}$$

Using (28) and (29) the following closed form lower bound can be determined for R

$$R \geq B(n-1;b,p) - \frac{S_{n-1}q^c}{\binom{b-c}{n-1}} \left\{ \frac{n(b-c-n+1)}{2(b-c)} \right\}^{\frac{1}{2}} B(n-1;b-c,p) \tag{30}$$

The following closed form upper bound can be similarly derived:

$$R \leq B(n-1;b,p) - S_{b-c}q^c \left\{ \frac{2(b-c)}{n(b-c-n+1)} \right\}^{\frac{1}{2}} B(n-1;b-c,p) \tag{31}$$

Leggett developed a number of further weaker but simpler sets of bounds from equations (30) and (31) [L2,L3]. These become applicable for large and dense networks.

3.3. Problems with the Bounds

In this section we first give computational results that indicate that there are problems with the correctness of the bounds. We then show analytically that there are errors in the theory behind the bounds.

3.3.1. Computational Results

We implemented programs to calculate the lower bound on R using equation (25a), to calculate the upper bound on R using equation (25b) alone, and to calculate the upper bound on R using equations (25b) and (26) together.

We first verified our implementations by comparing with values reported by Leggett. In [L2, pp.46] and [L3, pp.385] a table is given of the values the bounds determine for a complete graph on ten nodes (K_{10}). As well in [L2, pp. 83-88] lists are given for the N_r values calculated for both the upper and lower bounds for K_i for i=6 to 10 and three other regular graphs. These examples show that Leggett used equations (25b) and (26) together for his upper bounds. An interesting observation is that for K_{10} these bounds are tighter than any of the other bounds already mentioned.

We next tested the performance of the bounds, to determine whether they are generally correct. A problem arises here: the problem of calculating the exact reliability of a graph is #P-complete, and therefore it is not generally feasible to obtain the exact reliability of medium and large graphs to compare with the bounds. However, we can obtain the actual values for complete graphs [G1] and for 2-tree and subgraphs of 2-trees [W1,W2] and hence we can compare the bounds to the true values here. Even when the exact reliability is not known, if we can find any types of graphs where the value delivered by the upper bound is less than the value delivered by the lower bound we know that the bounds must be wrong for these cases.

We found cases of graphs where Leggett's upper bound is less than his lower bound. We look at two of these here; fortunately, they are also 2-trees, or subgraphs of 2-trees, and it is therefore possible to obtain actual reliability values. First we look at the complete bipartite graph $K_{2,n-2}$. Figure 4 shows $K_{2,n-2}$ for n=6:

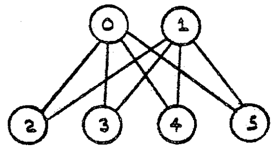

Figure 4

This family of graphs consists of subgraphs of 2-trees and therefore it is possible to determine their actual reliability values efficiently. Table 1 shows the values delivered by Leggett's bounds for a number of graphs in this family using various values of p. Both the value of the upper bound as calculated only using equation (25b) and the upper bound as calculated using equation (25b) and (26) together are included.

As can be seen from Table 1, the bounds are consistent when n=8 ($K_{2,6}$). However, for $n \geq 9$ the upper bound is less than the lower bound. The lower bound is the first to become incorrect. However, as n increases the upper bounds also become incorrect; the tighter bound using equations (25b) and (26) together falls below the actual reliability first.

For the p and n values given in Table 1 for $K_{2,n-2}$, whenever any of the bounds are incorrect, the upper bound is less than the lower bound. In this case, we know the bounds are wrong even when we do not know what the actual reliability is. This is not generally the case. If we construct a family of graphs by starting with two nodes with a single edge incident on them and add an arbitrary number of new nodes each with exactly two edges incident on them (one between the new node and each of the two original nodes) we have a family of 2-trees which we will refer to as *diamonds*. Figure 5 shows a diamond with 6 nodes.

Table 1: Bounds on $K_{2,n-2}$ Bipartite Graphs

n	p	actual	lower(25a)	upper(25b&26)	upper(25b)
8	0.3	0.012107	0.012103	0.012195	0.012430
	0.5	0.162354	0.162017	0.163712	0.167803
	0.7	0.562380	0.559967	0.564978	0.574825
	0.9	0.941446	0.939315	0.941686	0.943526
9	0.3	0.006669	0.006812*	0.006674†	0.006899
	0.5	0.125671	0.129041*	0.125975†	0.130999
	0.7	0.514456	0.523070*	0.515745†	0.527976
	0.9	0.932059	0.933182*	0.932269†	0.934346
10	0.3	0.003609	0.003844*	0.003382*†	0.003586*†
	0.5	0.096207	0.104083*	0.091105*†	0.096832†
	0.7	0.469284	0.493957*	0.462367*†	0.476208†
	0.9	0.922744	0.928660*	0.922749†	0.924958†
11	0.3	0.001927	0.002192*	0.001404*†	0.001583*†
	0.5	0.073132	0.085307*	0.057632*†	0.063913*†
	0.7	0.427523	0.471064*	0.404841*†	0.419803*†
	0.9	0.913517	0.925257*	0.913223†	0.915518†

* (lower bound > actual value) or (upper bound < actual value)
† (upper bound < lower bound)

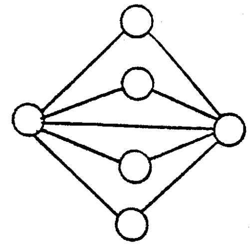

Figure 5

There is a close relationship between these diamonds and $K_{2,n-2}$. Adding an edge between nodes 0 and 1 in Figure 4 yields a graph equivalent to the one in Figure 5. Table 2 shows the values delivered by Leggett's bounds for various diamonds.

This table shows that when n=7 the lower bounds are incorrect. However, for this case the upper bound *is not* less than the lower bound and therefore if the actual reliability value was not known we would not know that the bounds are wrong.

The closed form bounds (equations (30) and (31)) were also implemented. Experimental results show that graphs can be found where these bounds also deliver erroneous values.

Table 2: Bounds on Diamond Graphs

n	p	actual	lower(25a)	upper(25b&26)	upper(25b)
6	0.1	0.000358	0.000358	0.000358	0.000360
	0.3	0.045870	0.045642	0.045870	0.046290
	0.5	0.285156	0.282961	0.285156	0.288437
	0.7	0.676415	0.671279	0.676415	0.681750
	0.9	0.960491	0.958322	0.960491	0.961384
7	0.1	0.000078	0.000078*	0.000078	0.000078
	0.3	0.025354	0.025448*	0.025483	0.025958
	0.5	0.221680	0.222397*	0.222876	0.227907
	0.7	0.620111	0.620151*	0.621746	0.630559
	0.9	0.950971	0.949942*	0.951088	0.952416
8	0.1	0.000016	0.000017*	0.000016*†	0.000017†
	0.3	0.013754	0.014158*	0.013656*†	0.014109†
	0.5	0.170166	0.176110*	0.169286*†	0.175455†
	0.7	0.566223	0.578090*	0.565749*†	0.576867†
	0.9	0.941477	0.943374*	0.941545†	0.943123†
9	0.1	0.000003	0.000004*	0.000003*†	0.000003*†
	0.3	0.007360	0.007940*	0.006756*†	0.007162*†
	0.5	0.129578	0.141593*	0.121184*†	0.128173*†
	0.7	0.516069	0.544702*	0.507249*†	0.519939†
	0.9	0.932065	0.938305*	0.931969†	0.933716†
10	0.1	0.000001	0.000001*	0.000001*†	0.000001*†
	0.3	0.003899	0.004515*	0.002688*†	0.003039*†
	0.5	0.098160	0.115808*	0.07624*†	0.083860*†
	0.7	0.469962	0.518130*	0.446010*†	0.459804*†
	0.9	0.922745	0.934330*	0.922409*†	0.924283†

* (lower bound > actual value) or (upper bound < actual value)
† (upper bound < lower bound)

They were derived from equations (25*a*) and (25*b*) by Leggett using equation (29). They should therefore be looser than the bounds determined from equations (25*a*) and (25*b*). It is true that they only first become wrong for the $K_{2,n-2}$ and diamond graphs at higher values of n. However, they also give incorrect values for some types of graphs for which the original bounds give correct values. One such graph is the complete graph on five nodes (K_5). For this graph the upper closed form bound (equation (31)) is less than the lower closed form bound (equation (30)) for all values of p.

3.3.2. Theoretical Analysis

There definitely appear to be some real problems with Leggett's bounds. We document a number of the minor errors first and then show there are problems with Leggett's basic theorems.

The closed form bounds are the easiest to compute. They are derived from the original bounds using equation (29). Leggett claims to prove the correctness of equation (29) [L2, App. A.2]. However in his last step where he takes the antilogarithms of both sides of the equation, he appears to be in error. Looking at the K_5 graph, $K_{b-c}=K_6=X_4X_5X_6=1.17649$ while $\left(\frac{n(b-c-n+1)}{2(b-c)}\right)^{\frac{1}{2}}=(\frac{10}{12})^{\frac{1}{2}}$ which is *not* less than or equal to 1.17649. Thus the closed form bounds are not derived correctly from the original bounds.

Many of the most serious problems are with the original bounds. A number of basic graph-theoretical results are incorrect. Leggett assumes that $S_{b-c}=\sigma_c$ where σ_c is the number of nodes in the graph of degree c. This is true for some graphs, such as the complete graph where $S_{b-c}=\sigma_c=n$. It is also true for the two types of graphs ($K_{2,n-2}$ and diamonds) we chose to look at in the previous section. However, it is not generally true. S_{b-c} may be much larger than σ_c. As an example in a cycle $\sigma_c=n$ while $S_{b-c}=\binom{n}{2}=\frac{n(n-1)}{2}$. It is even possible for σ_c to be zero, as in the following simple graph:

$c=1$

$\sigma_c=0$

$S_{b-c}=1$

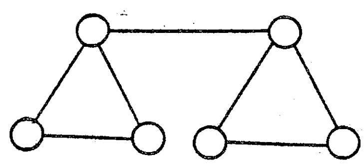

Figure 6

σ_c is actually the number of S_{b-c} cuts containing c edges which disconnect exactly one node from the rest of the network. It only counts a subset of the total number of minimum cardinality cuts. So $\sigma_c \leq S_{b-c}$. In the previous section we chose to look at two types of graphs ($K_{2,n-2}$ and diamonds) for which it is true that $\sigma_c=S_{b-c}$. For graphs where σ_c is quite a bit less than S_{b-c} the effect of this assumption is to deliver an upper bound that is looser than it would be if the actual value of S_{b-c} were used. So in some cases this assumption saves the upper bound from being wrong. It can, however, cause the situation in which, although the lower bound is larger than the actual value, it is still less than the upper. In this situation, it is not obvious that the bounds are not working when the actual value is unknown.

Tables 1 and 2 illustrate that there is something wrong with the bounds themselves. The upper bound obtained using both equations (25*b*) and (26) becomes incorrect sooner than the one using only equation (25*b*). This leads us to suspect that equation (26) is not correct. In fact, this can be shown using some of the actual examples and values that Leggett reported. Leggett states that equation (26) is to be applied for non regular graphs. This appears to be a mistake as he applied it to the regular graphs for which he gives the subtree counts [L2, pp. 83-88]. Even for complete graphs equation (26) is not true for the entire range of $b-c \geq r \geq b-2c+1$. This is illustrated by Table 3 which is taken from a table in [L2].

Table 3 shows that, for complete graphs, it appears that equation (26) is only true in the range $b-c \geq r \geq b-2c+3$. For the values of r where equation (26) is not true it gives underestimates for S_r. Therefore, overestimates for these N_r values are obtained. This example might lead us to suspect that equation (26) might be transformed into the following inequality which could still be used to determine an upper bound:

$$S_r \leq \sum_{i=1}^{n-1} \sigma_i \binom{b-i}{b-r-i}, \qquad \text{for } b-c \geq r \geq b-2c+1 \tag{26'}$$

However, even inequality (26)' is not true. Leggett gave the subgraph counts calculated by his bounds on three regular graphs [L2, pp.86-88]. Not only does equation (26) not give the correct subgraph counts for any r in the range $b-c \geq r \geq b-2c+1$ but for each of these graphs the value it delivers for $r=b-c-1$ is *lower* than the actual value.

Tables 1 and 2 show that there are also cases where the upper bound using equation (25*b*) alone is incorrect. The lower bound (which uses equation (25*a*)) is wrong even more often in these tables. Equations (25*a*) and (25*b*) are both developed from the basic theorem [L2,Thm. III.2] given as equation 3.

$$S_{r+1} \leq f_{n-1}\left(\frac{b-c-r}{r+1}\right) S_r \tag{24}$$

where:

Table 3: Connected Subgraph Counts from [L2 pp. 83]				
	K_6 Graph		K_7 Graph	
r	N(r) real	N(r) upper bound	N(r) real	N(r) upper bound
5	1296	1296		
6	3660	3745*†	16807	16807
7	5700	5715*†	68295	77818
8	6165	6165*	156555	162170
9	4945	4945*	258125	260115
10	2997	2997	331506	331695*†
11	1365	1365	343130	343161*†
12	455	455	290745	290745*
13	105	105	202755	202755*
14	15	15	116175	116175*
15	1	1	54257	54257
16			20349	20349
17			5985	5985
18			1330	1330
19			210	210
20			21	21
21			1	1

* (terms calculated using equation (26))

† (terms calculated using equation (26) but not equal to the actual value)

$$f_{n-1} = \frac{\left(1+\frac{1}{n-1}\right)^{n-1}}{\left(1+\frac{1}{2}\right)^{2}}$$

We show here that there are errors in this basic theorem. In assessing equation (24), it is useful to note that $(1+\frac{1}{n})^n$ is asymptotic to e; hence f_{n-1} (as well as X_r for all r) are bounded above by the absolute constant 1.21.

In its published form, a simple n-vertex cycle forms a counterexample to equation (24), since $S_{n-2} = \binom{n}{2}$, $S_{n-3} = \binom{n}{3}$, $b=n$, and $c=2$. Hence equation (24) asserts that $3 \leq f(n)$, which is impossible. Of course, Leggett's claim is not required for $i<n$ and $i>b-c$ in order to obtain the bounds; hence the cycle counterexample is not completely satisfactory. We develop here families of graphs which provide counterexamples to equation (24), even within the range required, that is $n-1 \leq i \leq b-c$.

Theorem 3.1: There is a graph G for which S_{b-c} and S_{b-c-1} do not satisfy equation (24).
Proof:

Construct a graph from a complete t-vertex graph and a cycle of length s by identifying a single vertex on the cycle with one of the vertices of the complete graph. The resulting graph has cohesion 2, and has $b = \binom{t}{2}+s$. Observe that any edge cutset with fewer than $t-1$ edges must induce an edge cutset on the cycle, and cannot cut the complete graph. Hence, for this graph, when $j<t-3$,

$$S_{b-2-j}=\sum_{i=0}^{j}\binom{s}{i+2}\binom{\binom{t}{2}}{j-i}$$

In particular, then, $S_{b-2}=\binom{s}{2}$ and $S_{b-3}=\binom{s}{3}+\binom{s}{2}\binom{t}{2}$. Equation (24) asserts that $S_{b-2} \leq \frac{1.21}{s+\binom{t}{2}-2} S_{b-3}$. One can easily verify that whenever $s \geq \binom{t}{2}$, this claim is false. •

Theorem 3.1 produces an infinite family of counterexamples to Leggett's major result (equation (24)), hence invalidating his proposed bounds. It is worth noting that the construction employed in theorem 3.1 can also be used to show that other coefficients do not satisfy equation (24); that is, the counterexamples do not arise *simply* because we examine the coefficient corresponding to minimal cutsets.

Nevertheless, equation (24) could still be useful if we could establish that certain of the coefficients do satisfy this bound. In particular, one might ask (in contrast to theorem 3.1) whether the claim holds when examining S_{n-1}; the motivation is that, although counterexamples exist for the coefficients with almost all the edges, we may be able to salvage the claim when we consider coefficients corresponding to few edges. Unfortunately, equation (24) is also false here:

Theorem 3.2: There is a graph G for which S_n and S_{n-1} do not satisfy equation (24).
Proof:

We construct an infinite class of graphs for which S_n and S_{n-1} do not satisfy equation (24). Let $G_{h,k}$ be the graph obtained by connecting two extreme points by h vertex-disjoint paths with k-1 intermediate vertices. The result has $n=hk-h+2$ vertices and hk edges. For such a graph, the edge connectivity $c=2$. The complete bipartite $K_{2,n-2}$ graphs looked at in a previous section were examples of $G_{h,k}$ graphs where $h=n-2$ and $k=1$. Figure 7 shows a small $G_{h,k}$ graph where $h=4$ and $k=3$.

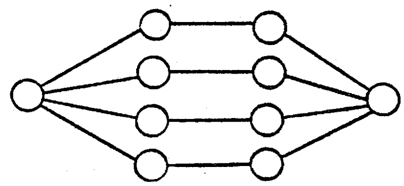

Figure 7

Whenever $5 \leq h \leq 12$ and k is sufficiently large, $G_{h,k}$ forms one of the desired counterexamples. This can be seen as follows. The number of spanning trees of $G_{h,k}$ is hk^{h-1}, and hence S_{n-1} is $\binom{kh}{kh-h+1} - hk^{h-1}$. Similarly, S_n is $\binom{kh}{kh-h+2} - \binom{h}{2} k^{h-2}$. Equation (24) asserts that $S_n \leq 1.21 \frac{b-c-n+1}{n} S_{n-1}$. Rewriting, this is equivalent to $\frac{S_n hk-h+2}{S_{n-1} h-3} \leq 1.21$.

As $k \rightarrow \infty$ and h is held fixed, the numerator above is dominated by the coefficient of k^{h-1}, which is $\frac{h^{h-1}}{(h-2)!} - h\binom{h}{2}$. Similarly, the denominator is dominated by the coefficient of k^{h-1}, which is $\frac{(h-3)h^{h-1}}{(h-1)!} - h(h-3)$. When $5 \leq h \leq 12$, this ratio *exceeds* 1.21; in fact, for $h=5$, the limit as $k \rightarrow \infty$ is approximately 1.278. This is in contradiction to equation (24); hence, for sufficiently large k, $G_{h,k}$ forms a counterexample of the form desired, when $5 \leq h \leq 12$. •

3.4. Conclusion

Theorems 3.1 and 3.2, along with the numerous more minor errors documented earlier, suggest that there is no reasonable expectation of applying Leggett's bounds to general networks. Nevertheless, the basic idea of exploiting graph-theoretical structure to obtain bounds is sound; in fact, it appears to be the most promising method of improving on the current best, the Ball-Provan bounds.

We have shown that despite the soundness of Leggett's underlying approach, the bounds developed in [L2,L3] are seriously in error. The problems extend back to the basic theorem upon which they are developed. In view of the fatal difficulties with equation 24, and the other severe problems noted, Leggett's bounds must be abandoned. We do not consider them in the discussions and testing in the rest of this document.

Chapter 4

Implementation Considerations

In this chapter, we look at algorithms to calculate the graph theoretic values used by the bounds introduced in chapter 2, as well as the calculation of the actual bounds themselves. We implemented each bound, to obtain reliability values for the purposes of comparison. To be of practical use, any graph theoretic values used in the bounds must be obtainable in polynomial time. In the sections which follow, we therefore describe efficient algorithms to evaluate the bounds.

4.1. Graph Theoretic Values

The most complicated part of the implementation of these bounds is the implementation of the algorithms for determining the required graph theoretic values. The Jacobs-II, BBST, Kruskal-Katona, and Ball-Provan bounds all use the values for the graph's number of spanning trees (t), its edge connectivity (c) and the number of minimum cardinality network cuts (C_c). The Lomonosov-Polesskii bounds use the value for the graph's edge connectivity, and the cardinality of the cuts in a cut basis of the graph.

4.1.1. Calculation of Number of Spanning Trees in a Graph

N_{n-1} is equal to t, the number of spanning trees for the graph. The original theory behind counting spanning trees in a graph was developed by Kirchoff [K6]. Later, Tutte [T4] put it into a form which is more appropriate for computation. His work applies to the more general case of directed graphs. It also allows for multiple edges. For undirected graphs with no multiple edges, the number of spanning trees can be determined by:

1. First forming the degree matrix D for the graph.
   The *degree matrix* is the matrix defined by:
   $D(i,j) = d_i$ if $i = j$
   $D(i,j) = -1$ if $i \neq j$ and *there is an edge* (i,j)
   $D(i,j) = 0$ if $i \neq j$ and *there is no edge* (i,j)
2. The determinant of D minus its 'n'th row and column is equal to the number of spanning trees in the graph.

The degree matrix can be created in $O(n^2)$ time and there are well-known algorithms for calculating the determinant of a matrix in $O(n^3)$ time (see for example, [A1]). It should be noted that algorithms have been developed which improve on this running time (for example, Strassen's algorithm [S4]). For our purposes, however, the $O(n^3)$ classical method suffices.

4.1.2. Calculation of Edge Connectivity

The calculation of the edge connectivity of a graph is based on the classical theorem by Menger [M1], which states that the edge connectivity between any two nodes is equal to the number of edge disjoint paths between the two nodes. The number of edge disjoint paths between any two nodes can be determined by the use of Ford and Fulkerson's well-known Maximum Flow, Minimum Cut theorem [F3]. Their labeling algorithm determines the maximum flow between any two nodes in a directed graph where each edge has a predetermined capacity. An undirected graph must first be converted to a directed graph by replacing each of its undirected edges (i,j) with the two directed edges $\langle i,j\rangle$ and $\langle j,i\rangle$. The number of edge disjoint paths between any two nodes can be determined by applying the labeling algorithm with each edge given a capacity of one. This is often referred to as a *0/1-flow problem*. The overall edge connectivity c can be determined [E2] by choosing some arbitrary node v_s and successively solving the 0/1-flow problem between s and every other node $v_i \in V$. The minimum of the values obtained in these $n-1$ iterations is equal to c. A single 0/1-flow problem can be performed in $O(bn)$ time [E2]. Therefore, the edge connectivity of a graph can be determined in $O(bn^2)$ time.

4.1.3. Calculation of the Number of Minimum Cardinality Cuts

C_c is equal to the number of minimum cardinality cuts (number of edge cutsets containing exactly c edges) in the graph. Its calculation is more complicated than that of the edge connectivity or that of the number of spanning trees in a graph. It has been shown that its calculation is NP-hard for directed graphs [P3]. However, Ball and Provan [B3], using a result due to Bixby [B6], have recently developed an algorithm for calculating it in polynomial time for undirected graphs. The algorithm takes as input the set of nodes and edges of the graph (V and E), the edge connectivity of the graph (c), and again some arbitrarily chosen original node (v_s) and returns the number of minimum cardinality cuts in the graph (C_c).

NUMCUTS (V,E,c,v_s)
1) mincuts= 0
2) for each $v_i \in V-v_s$:
    a) find k, the 0/1-flow between v_s and v_i
    b) if $k = c$ then:
        i) find minstcuts, the number of minimum cardinality cuts between v_s and v_i.
        ii) mincuts= mincuts + minstcuts
    c) collapse v_s and v_i into single node v_s
3) return(mincuts)

Step 2a is the $O(bn)$ 0/1-flow problem discussed in the previous section. That leaves 2b(i) as the only other substantial step. Ball and Provan show that minstcuts can be calculated using the following algorithm:

STCUT (V,E)
1) find and delete from the graph G= (V,E) all nodes not on a path from v_s to v_t.
2) find a maximum 0/1-flow F_{ij}
3) generate the flow induced network $G_1=(V,E_1)$ with E_1 defined as follows: for each $\langle i,j\rangle \in E$
    if $F_{ij}=0$ then
        add $\langle j,i\rangle$ to E_1
    if $F_{ij}=1$ then
        add $\langle i,j\rangle$ to E_1
4) collapse all strongly connected components in G_1 into single nodes to obtain the reduced graph $G_2=(V_2,E_2)$
5) find m, the number of antichains in G_2-v_t
6) return (m)

Step 1 employs the well-known algorithm for finding biconnected components in a graph [T2]. If this algorithm is started at node v_s, all nodes on a path from v_s to v_t must be in the same biconnected component as v_s and v_t. This step can be executed in $O(b)$ time. Step 2 is again the 0/1-flow problem described in the previous section. It leaves the edges of the graph with a maximum flow indicated on its edges. That is, $F_{ij} = 1$ if there is flow on the edge $\langle i,j\rangle$ in this maximum flow, and $F_{ij} = 0$ if there is no flow on the edge $\langle i,j\rangle$. It is of $O(bn)$. In step 3, the flow indicators left by step 2 are used to create the graph G_1. This is of

$O(b)$. Tarjan [T2] developed an $O(b)$ algorithm that can be used to find the strongly connected components of G_1 for step 4 where these are collapsed to leave the reduced graph G_2. Step 4 will always leave G_2 as an acyclic graph which will be a compact representation of all the minimum cardinality cuts between v_s and v_t. An *antichain* in an acyclic network is any subset of nodes M such that for all pairs of nodes i and j in M, i is not a predecessor or a successor of j. Ball and Provan [B3] show that there is a one-to-one correspondence between the minimum cardinality cuts between v_s and v_t in G and the antichains in G_2–v_t. Therefore, counting antichains in G_2–v_t gives the desired result. The following recursive algorithm returns the number of antichains in a given graph:

```
ANTICHAIN (V,E)
    1) choose v_i ∈ V of in-degree 0
    2) set num=0
    3) if V–v_i is an empty set then
            return(num)
    4) num= num + ANTICHAIN (V–v_i,E)
    5) find S, the set of successors of v_i
    6) if V–S–v_i is an empty set then
          return(num)
    7) num= num + ANTICHAIN (V–S–v_i,E)
    8) return(num)
```

Each call to ANTICHAIN counts at least one antichain and the steps in it can be performed in $O(b)$ time. So, step 5 of STCUTS can be executed in $O(bm)$ time where m is the number of minimum cardinality cuts between v_s and v_t. Therefore STCUTS can be executed in $O(max[bn,bm])$. This means that the overall calculation of the number of minimum cardinality cuts in the graph as calculated by NUMCUTS is of $O(max[bn^2,bC_c])$. In an important result, Bixby [B6] determined that for undirected graphs $C_c \leq \binom{n}{2}$. Therefore for undirected graphs the number of minimum cardinality network cuts can be calculated in $O(bn^2)$ time.

An important fact to note is that no polynomial-time-bounded algorithms have been developed to calculate any of the spanning subgraph counts N_i for $n-1 < i < b-c$ (or edge cutset counts C_i for $c < i < b-n+1$) for general graphs.

4.1.4. Calculation of Cut Basis

As introduced in chapter 2, Lomonosov and Polesskii's upper bound (equation 25b) is determined using a cut basis $L=\{L_1,L_2, \cdots ,L_{n-1}\}$ of the graph. Actually, to calculate the bound, we only need to know the cardinality of (number of edges in) each of these n–1 cuts. A set of values corresponding to the cardinality of the cuts in a cut basis can be calculated using the following method.

1. Solve the 0/1-flow problem between every pair of nodes $v_j,v_k \in V$ and order these $\binom{n}{2}$ values in decreasing order. This creates a complete graph on the nodes in V with a weight on each edge corresponding to the cardinality of the minimum cut (i.e. the 0/1-flow) between the two nodes upon which the edge is incident.

2 Create a maximum weight spanning tree for the nodes in V with a weight on each edge corresponding to the cardinality of the minimum cut (i.e. the 0/1-flow determined in step 1) between the two nodes the edge is incident upon. This is done by:
   a. starting the tree with the edge corresponding to the largest value as determined in step 1.
   b. adding the edge corresponding to the next value from this list *if* it does not form a

circuit.

c. successively considering edges corresponding to the values from the list adding them to the tree if they do not form a circuit until $n-1$ edges have been placed in the tree.

The $n-1$ weights on the edges selected for the tree created in step 2 correspond to the cardinality of the $n-1$ cuts in a cut basis for the original graph. The circuit detection in step 2 can be done efficiently using depth first search. In step 1 the $O(bn)$ 0/1-flow problem is solved $\binom{n}{2}$ times and therefore the cardinality of the cuts in a cut basis of a graph can be determined in $O(bn^3)$ time. It should be noted that step 2 is essentially Kruskal's spanning tree algorithm [K7].

4.2. Implementation of the Bounds

In this section we look at the calculation of the actual bounds themselves These can all be calculated in linear time, given the graph theoretic values computed earlier.

4.2.1. Jacobs and BBST Bounds

After the values for c, t, and C_c have been determined the calculation of the Jacobs-II or BBST bounds is almost trivial. They both consist of the summation of the $e_i=\binom{b}{i}p^{b-i}q^i$ values for certain ranges of i. These summations can be done quite efficiently using the fact that $e_{i+1} = e_i\frac{(b-i+1)}{i}(\frac{q}{p})$. After the first e_i value in a summation has been calculated this relationship can be used to successively determine the rest. As well, as mentioned by Bauer et al. [B5], these summations correspond to the binomial distribution and can also be found using table look ups.

In the Jacobs-II bounds the actual value of C_{b-n+1} is only used in the upper bound and is overapproximated as $\binom{b}{n-1}$ in the lower bound. Similarly, the actual value of C_c is only used in the lower bound and is underapproximated as 0 in the upper bound. In the BBST bounds the actual values of both C_{b-n+1} and C_c are used in the lower bound but only the actual value of C_c is used in the upper bound where the value of C_{b-n+1} is overapproximated. In cases where both the upper and lower bounds are going to be used both C_c and C_{b-n+1} have to be calculated anyway and therefore they might as well be used in both the upper and lower bounds to make them as tight as possible. When this is done for the Jacobs bounds we obtain:

$$R \leq 1 - C_c p^{b-c}q^c - C_d p^{b-d}q^d - \sum_{i=d+1}^{b} \binom{b}{i} p^{b-i}q^i \tag{32a}$$

and

$$R \geq 1 - C_c p^{b-c}q^c - \sum_{i=c+1}^{d-1} \binom{b}{i} p^{b-i}q^i - C_d p^{b-d}q^d - \sum_{i=d+1}^{b} \binom{b}{i} p^{b-i}q^i \tag{32b}$$

where $d=b-n+1$

Similarly the BBST bounds become:

$$R \leq 1 - \frac{C_c}{\binom{b}{c}}\left[\sum_{i=c}^{d-1}\binom{b}{i} p^{b-i}q^i\right] - C_d p^{b-d}q^d - \sum_{i=d+1}^{b} \binom{b}{i} p^{b-i}q^i \tag{33a}$$

and

$$R \geq 1 - C_c p^{b-c} q^c - \frac{C_d}{\binom{b}{d}} \left[\sum_{i=c+1}^{d} \binom{b}{i} p^{b-i} q^i \right] - \sum_{i=d+1}^{b} \binom{b}{i} p^{b-i} q^i \tag{33b}$$

where again $d = b-n+1$

For the representation using the binomial distributions B_r, they become:

$$R \leq 1 - \frac{C_c}{\binom{b}{c}}(B_{d-1} - B_{c-1}) - C_d p^{b-d} q^d - (1-B_d) \tag{34a'}$$

and

$$R \geq 1 - C_c p^{b-c} q^c - \frac{C_d}{\binom{b}{d}}(B_d - B_c) - (1-B_d) \tag{34b}$$

To avoid confusion we refer to the bounds obtained from equations (32a) and (32b) as the *Jacobs-III bounds* and the bounds obtained from equations (33a) and (33b) (or equations (34a) and (34b)) as the *BBST-II bounds*. These are the bounds for which we report results in the next chapter.

4.2.2. Kruskal-Katona and Ball-Provan Bounds

These are more sophisticated sets of bounds and consequently their implementation is somewhat more complicated. In [B3], Ball and Provan give a thorough discussion of implementation considerations that applies to both the Kruskal-Katona and the Ball-Provan bounds. The major new feature in their calculation is the need to determine k-canonical representations and calculate pseudopowers.

Once the k-canonical representations have been determined, the calculation of the pseudopowers consists of the simple summation of the appropriate $\binom{j}{i}$ values. The only complication here is that for the lower pseudopower (Kruskal-Katona bounds) we are to take $\binom{j}{i} = 0$ whenever $i \leq 0$ and $j < i$. For the upper pseudopower (Ball-Provan bounds), we take the cases $\binom{j}{0} = 1$ for $j \geq -1$.

In the Kruskal-Katona bounds the c-canonical representation of F_c and the d-canonical representation of F_d are needed, while the c-canonical representation of H_c is needed for the Ball-Provan bounds. For the Ball-Provan bounds the (k,d)-factor of $a = t - \sum_{i=0}^{c} h_i$ must also be calculated. In [B3], Ball and Provan give the following algorithm for this (k,d)-factor calculation:

KDCALC (k,d,x,num,M)
1) $i = d$
2) while $num > 0$
 a) if $\binom{x}{i} - \binom{x-d+k}{i-d+k} > num$
  $x = x-1$

b) else

i) M[i] $= x$

ii) $num = num - (\binom{x}{i} - \binom{x-d+k}{i-d+k})$

iii) $x = x-1$

iv) $i = i-1$

If the value a is placed in num and x is initialized to $b+1$ the algorithm places the d-canonical representation of the (k,d)-factor of a in the vector M. This algorithm can also be used to determine the r-canonical representations needed for both the Kruskal-Katona and Ball-Provan bounds since the r-canonical representation of a number z is the $(0,r)$-factor of z. The c-canonical representation of either H_c or F_c can be found using KDCALC with k set to 0, d set to c and x initialized to $b+c-2$. We had a small problem here: the first time through we have $\binom{x}{c} - \binom{x}{0}$ and if we take $\binom{x}{0}$ as equal to 1 we end up with the c-canonical representation of $(num+1)$. For example, in [B4,pp. 172] the 2-canonical form of 24 is given as $\binom{7}{2}+\binom{4}{1}$ which is actually the 2-canonical form of 25. One way to fix this is to initialize num to one less than the value desired. For example to obtain the c-canonical representation of H_c, set $num=(H_c-1)$. Communication with Provan verifies his awareness of this problem in the published version of the paper.

In [B3], Ball and Provan show that the calculation of these two sets of bounds can be performed in linear time. In KDCALC the successive recalculation of $\binom{x}{i} - \binom{x-d+k}{i-d+k}$ can be performed using only four arithmetic steps by the use of the following two identities.

$$\binom{j-1}{i} = \frac{j-m}{j}\binom{j}{i} \qquad \text{for } j>m\geq 0$$

and

$$\binom{j-1}{i-1} = \frac{m}{j}\binom{j}{i} \qquad \text{for } j\geq m>0$$

These two identities can also be used to determine the successive lower or upper pseudopowers (i.e. $z^{(i+1)/k}$ from $z^{i/k}$ or $z^{<i+1/k>}$ from $z^{<i/k>}$) by keeping track of the previous value for each of the $\binom{m_i}{i}$ calculations.

The number of trees in a graph can be as large as n^{n-2} (for complete graphs) [B3]. In both the Kruskal-Katona and the Ball-Provan bounds, k-canonical representations are determined and pseudopowers are calculated for values that are of the same order of magnitude as t (F_d and a). Ball and Provan [B3] state that the limiting factor in their implementation is the size of the numbers that had to be manipulated.

4.2.3. Lomonosov-Polesskii Bounds

Once the cardinality of the cuts in a cut basis and the edge connectivity c have been calculated, computing Lomonosov and Polesskii's bounds is trivial. Both the calculation of c and the calculation of the cut cardinalities are based on the same 0/1-flow problem. These bounds do not need the values for C_c or t.

4.3. Verification of Implementations

It is (of course) very important to verify the correctness of any implementation, especially if it is to be used for testing purpose. Some simple checks on the correctness of the values obtained from the algorithms that determine the values of t, c, and C_c can be carried out on families of graphs where these values are known. For example for a complete graph $c = n-1$, $C_c = n$ and, as already mentioned, $t = n^{n-2}$. In all cases, our bounds delivered the correct values. The correctness of the implementation of the algorithm to calculate the cardinality of the cuts in a cut basis was verified on a number of small examples.

Checking the correctness of the values obtained for the actual bounds is more difficult. The major problem here is that the original developers have not always reported examples. Ball-Provan [B3] develop a section describing the tests they performed using their bounds as well as the Kruskal-Katona bounds, and even include a table of their results. Unfortunately, they do not report any of the actual values delivered by the bounds and only report the value they describe as the l-norm of the difference between the upper and lower bounds (this l-norm is described in the next chapter). However, in [B4], Ball and Provan did report the actual values delivered by their bounds as well as by the Kruskal-Katona bounds on a small graph. Our implementation delivered these values for both sets of bounds. In [B5], Bauer et al. reported the values obtained by their bounds as well as the bounds delivered by the Kruskal-Katona bounds for a single six node graph. Our implementation of the BBST bounds delivered the same values that they reported. However, our implementation does not deliver the exact values they reported for the upper bound of the Kruskal-Katona bounds. Working this example through by hand, we obtain the same values as delivered by our program and therefore feel our implementation is correct for this example.

We found no other examples in the literature reporting actual values delivered by these or any of the other bounds we are considering. We worked through the calculation of each of the bounds for a number of small example graphs by hand and found no inconsistencies. However, it would still be desirable to have more examples on which to test the results, especially for the more sophisticated Kruskal-Katona and Ball-Provan bounds.

As already mentioned, some of the numbers involved in the calculations can be very large. At the same time, for dense graphs and values of p close to one, the difference between the values delivered by the bounds may be very small. For large graphs a large number of mathematical manipulations may be performed, especially by the more sophisticated bounds. Therefore, consideration must be given to the precision of the number representations.

Our implementations are on a VAX 11/750 running UNIX 4.1BSD. The main implementation is in the C language. This includes algorithms for calculating t, c, C_c, the cardinality of the cutsets in a cut basis and programs to calculate each of the sets of bounds that have been discussed. As well, algorithms for generating and calculating the exact reliability of special subclasses of graphs used for test purposes and described in chapter 5 are also included.

Like Ball and Provan, we use double precision-floating point numbers (64 bits) for our calculations. With this implementation, we are able to calculate the Jacobs-III and BBST-II bounds for networks with up to 10^{20} trees. Calculation of the more sophisticated Kruskal-Katona and Ball-Provan bounds is restricted to networks with less than 10^{13} trees. The Lomonosov-Polesskii bounds do not use the value for t or any values of this magnitude, and thus their calculation does not suffer from these limitations.

In order to test larger graphs as well as to be sure of the accuracy of the precision of the values reported, we also implemented a number of the actual bounds in BC. BC is a C-like language and compiler in which arbitrary precision computations can be performed. The BC language is unfortunately less flexible than C, and thus harder to work with. Using our BC implementation we were able to verify the precision of our C implementation results as being correct, at least up to the eighth decimal place. Unless otherwise stated the results we report are from the C implementation rounded to the first six digits to the right of the decimal. Most of the time this is adequate for our purposes. However, if what we are trying to show necessitates more precision we use the results from our BC implementation.

Chapter 5

Testing the Bounds

A number of different methods for obtaining bounds on the reliability polynomial have been described in the preceding chapters. However, little work has been done in testing them to determine their actual performance. Even the original developers of the various bounds have not typically given many results showing the performance of their bounds. In fact, as mentioned in Chapter 4, we had trouble finding enough examples in the literature to verify our implementations.

Ball and Provan are a notable exception to this, however. In [B3], they compare their bounds with the Kruskal-Katona bounds. They tested both sets of bounds on a number of different types of graphs. These were: the Arpanet topology for 1979, a number of "complete", "street" and "ladder" networks, as well as a number of undirected and directed graphs which were randomly generated (see [B3] for a descriptions of these graphs). They measured what they termed the l-norm of the difference between the upper and lower bounds. Let $r(p)$ be this range between the values delivered by the upper and lower bound for a given value of p. Then the *l-norm* is $\int_{p=0}^{1} r(p)dp$. They use the ratio of the respective l-norms as the means of showing the improvement of their bounds (the Ball-Provan bounds) over the Kruskal-Katona bounds.

5.1. Testing Philosophy

In testing the performance of bounds, the first problem is to determine exactly what to test. The fact that very little work has been done with regards to testing means that accepted standards or methods are lacking. Actually, the nature of the problem makes it very hard to set any standards for testing a set of bounds.

Difficulty arises due to the fact that there an infinite variety of graphs and it is therefore feasible to test only a small portion of them. The approach Ball and Provan chose was to randomly generate graphs to be tested. The idea is to obtain a representative sample of general graphs, but this is very difficult. Moreover, it does not allow us to assess the worst case performance of the bounds. Another approach is to test the bounds on graphs which constitute a representative sample for graphs to which the bounds will actually be applied. This necessitates making some strong assumptions about the type of graphs involved, and therefore restricts the applicability of any results. Hence, one must avoid being too constrictive with these assumptions.

One intended application for this reliability measure is computer communication networks. Typical computer communications networks have very sparse topologies; for example, local area networks are often set up in simple configurations such as trees or cycles. Even a large network like the Arpanet only has a cohesion of two (see Figure 8). Sparse configurations have often been dictated by factors such as the complexity of routing and flow control that arises as networks become denser. Therefore, the performance delivered by the different sets of bounds for sparse networks is of the most interest. Nonetheless, we cannot restrict our attention to just these sparse topologies. Very sparse configurations are known to be unreliable. This fact is the motivation for a good deal of research on determining more reliable topologies. Therefore denser configurations should also be examined, and obtaining a quantitative measure of their reliability is of interest. Consequently, determining the performance delivered by the bounds on denser graphs is also worthwhile.

Once it has been determined which graphs are to be tested, one must determine the possible values for the edge reliability, p. Ball and Provan's approach [B3] to this is to use the l-norm, and thereby avoid reporting results for any specific edge availability. The l-norm gives an indication of the overall performance of the bounds for all values of p. Unfortunately, for any single graph, this range between the upper and lower bound $r(p)$ may vary quite dramatically for different values of p.

Arpanet Configuration for September 1979 (from [B3])

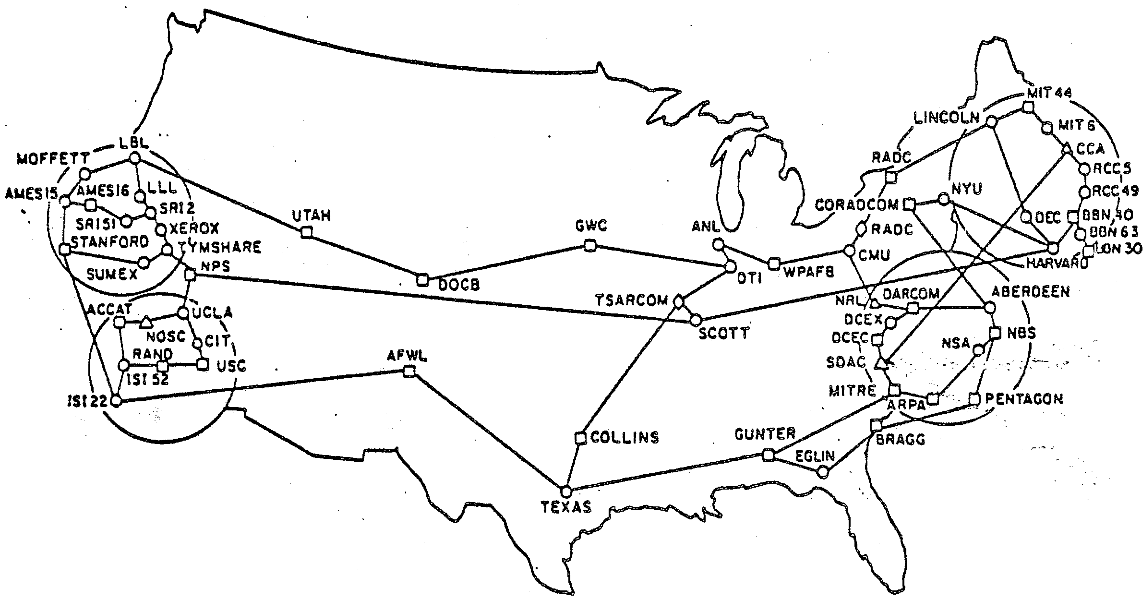

Figure 8

The large variation exhibited by $r(p)$ for $0 \leq p \leq 1$ limits the usefulness of any single measure that attempts to give an indication of the overall performance of the bounds, on even a single network. Therefore it may be preferable to consider specific values of p. In fact, a useful feature of these bounds is the ability to use them to perform sensitivity analysis on the effects of varying the value of p. An example where this might be applied is for determining the relative benefits obtained by the construction of a given network using one of a number of different types of links with correspondingly different values of p.

If we consider the possible applications of the bounds, we can narrow down the range of values of p to emphasize in our testing. Again, we should be careful not to be overly constrictive and hence severely limit the generality of any result we might obtain. This is particularly important when one recalls that the model from which the bounds have been developed allows much latitude in what p represents. If p represents the availability of a link, it may be quite high. In military applications, link availabilities are often required to be very close to 1. In [F2], Frank and Chou study the Arpanet using the same assumptions of equivalent and statistically independent link failures and give a value of 0.98 for p. On the other hand, if p represents the probability that a link fails in a given period of time, the value for p might be quite a bit lower, particularly if the "given period of time" is quite large. It should be remembered, however, that probabilistic connectedness is most applicable for high values of R. If the network is only fully connected for a small fraction of the time (say, $R<0.5$) it would typically be more useful to use a measure such as the expected number of node pairs which are able to communicate. This would give information concerning the partial usefulness of the network for the time when it is not connected. We therefore restrict our emphasis to values of p large enough so that R is reasonably high.

The size of the range $r(p)$ between an upper and lower bound gives one good indication of the performance of the bounds. However, it is more desirable to compare the values delivered by the bounds to the actual reliability of the graphs on which we are testing them. Where this actual value falls between the upper and lower bound shows their relative performance. In particular, it would be very valuable to ascertain whether or not one or both of the bounds is tight. A bound is *tight* if we can find a graph whose actual reliability is equal to the bound. Therefore, a bound is tight if it delivers the closest possible bound on the actual reliability given the information which it uses. Van Slyke and Frank [V2] have shown that while the Kruskal-Katona bounds are tight for coherent systems they are not tight for networks. Ball and Provan [B4] show that their bounds are tight for "shellable independence systems" but it is still open as to whether they are tight for networks. If one can establish that a bound is tight, it proves that one cannot improve the values delivered by the bound without employing new graph theoretic information.

When one cannot establish that a particular bound is tight, it is still desirable to determine which bound gives the best performance. It is essential also to, at least qualitatively, determine when the performance of a bound is relatively poor.

In the remaining sections, we first identify suitable test cases, then determine the relative merits of each bound, and finally study the effects of various graph transformations.

5.2. Basic Test Results

We must now determine which graphs to test, and what values of p on which to test them. We chose to restrict our tests mainly to graphs for which we can obtain the exact value of R. This provides the necessary basis for comparison.

Unfortunately, it is not generally feasible to obtain the actual reliability for medium and large scale networks. If we could, there would be no need for bounds in the first place. However, there are some classes of graphs for which the actual reliability can be obtained in polynomial time. As previously mentioned, we can obtain the exact reliability for complete graphs (K_n) [G1] as well as for 2-trees and subgraphs of 2-trees (partial 2-trees) [W1,W2]. The exact reliability of the family of graphs $G_{h,k}$ introduced in Chapter 3 (section 3.3, Theorem 3.2) can also be determined exactly. For this family of graphs,

$$N_i = \binom{h}{i-n+2} k^{h-(i-n+2)} \qquad \text{for } n-1 \leq i \leq b-c \tag{35}$$

With equation (35), the N_i values needed by equation (11) can all be calculated for this family of graphs. An infinite number of other graphs for which it is also possible to determine the exact reliability can be created by properly "combining" two graphs of the above types as edge disjoint subgraphs of a larger graph. This process of combining the subgraphs simply consists of identifying a single node in the first graph with a single node in the second graph. The reliability of the new graph is equal to the product of the reliabilities of the two edge disjoint subgraphs. This procedure can be repeated to combine any number of graphs. As well as being one of the above types of graphs, the subgraphs may also be any trivial graph. We define a *trivial* graph to be any graph for which $b-c \leq n-1$. For a trivial graph, once C_c and t are known there are no unknown N_i values, and therefore any of the bounds (other than the Lomonosov-Polesskii bounds) delivers an exact value for their reliability. An example of a trivial graph is the complete graph on four nodes (K_4). Any cycle is a trivial graph. Therefore, the other family of graphs introduced in Chapter 3 (section 3.3., Thm 3.1), consisting of a complete graph combined with a cycle, are examples of combination graphs for which reliability can be easily computed.

We focus on two types of 2-trees: *ladders* (see Figure 9) and the diamonds defined in Chapter 3. Ladders have only two nodes of degree two and have been shown to be instances of the most reliable series-parallel networks [N2,N3]. Diamonds, on the other hand, have the maximum number of nodes of degree two; they are the least reliable 2-trees that we can find. Of course, partial 2-trees are less reliable.

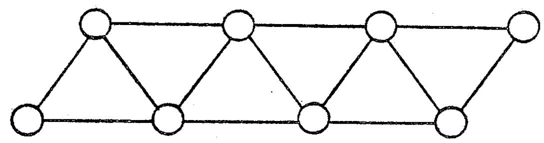

Figure 9

2-trees and $G_{k,t}$ graphs, like actual computer networks, are sparse. Complete graphs, on the other hand, are the densest family of graphs. Combination graphs can be combinations of very reliable and very unreliable graphs. For example, the family of graphs constructed by connecting a complete subgraph and a cycle contains a combination of the most reliable type of graph with a very unreliable graph. We also include one other dense graph (which is 6-cohesive) and the Arpanet configuration for 1979 (figure 8) for which the exact reliability is not known. In total, this forms a large collection of sample graphs on which to test these bounds. To our knowledge, this is the first substantial set of data tested which gives the actual values delivered by these bounds.

We must also determine values for p to be employed in the analysis. The values of p used vary as to the type of graphs tested. For the sparse graphs, we show representative samples of p below 0.9, and then concentrate on the region between 0.9 and 0.99. For the very dense complete graphs we use values below 0.9.

Our main interest is to perform a comparative test on these bounds to determine when each of them might be appropriate. In the appendix, we provide tables which report a large number of computational results. Small portions of these tables are excerpted in the analysis of the bounds, which follows. A "*" beside the value of a bound indicates that bound improves on the corresponding Ball-Provan bound. In all other cases the Ball-Provan bounds deliver the best values. All values are rounded off to six digits to the right of the decimal place. In section 5.2.1 we compare the Jacobs-III, BBST-II, Kruskal-Katona and Ball-Provan bounds and in section 5.2.2 we investigate the Lomonosov-Polesskii bounds.

5.2.1. Subgraph Bounds

The Jacobs-III, BBST-II, Kruskal-Katona and Ball-Provan bounds are all based on the approach of bounding R by finding both a set of overapproximations and a set of underapproximations on the subgraph counts of the reliability polynomial. They therefore exhibit similar general behaviour. For values of p near 0 and near 1 the value of $r(p)$ is quite small. However, for values of p near .5, the value of $r(p)$ for each of these bounds can become as large as 0.8 [B3]. Intuitively, this trend makes sense. We noted earlier that varying the value of p effectively varies the "weight" or the contribution which each of the subgraph counts N_i makes towards the total value of the reliability. Recall from Chapter 2 that for values of p near zero, the number of trees (N_{n-1}) becomes dominant. Since the bounds employ the value of t, it is to be expected that $r(p)$ will therefore be small for p close to 0. Similarly, since the term using C_c is dominant for p near 1, it is to be expected that $r(p)$ will also be small for p near 1. If p=0.5, all the N_i terms are of equal importance in R and therefore $r(p)$ is quite large as all these N_i values (for n-1$\leq i \leq b$-c) are only approximated by the bounds. In view of the intended applications of the bounds, we are most interested in values of p near 1.

We found that we can rank these bounds into a definite hierarchy. For the graphs tested, the value obtained from the Jacobs-III lower bound is never greater than the value obtained from the BBST-II lower bounds. At the same time the value obtained for the Jacobs-III upper bound is never less than the value obtained for the BBST-II upper bound. The improvement delivered by the BBST-II bounds is not always very large. This is illustrated in Table 4. However, we found cases in which the ratio between the $r(p)$ of the Jacobs-III bounds and the $r(p)$ of the BBST-II bounds is about 1.7, as shown in Table 5. It comes as no surprise that the BBST-II bounds improve on the Jacobs-III bounds; recall that the Jacobs-III bounds make the loosest possible assumptions about the subgraph counts (0 or $\binom{b}{i}$). In Chapter 4, we saw that the calculation of the BBST-II bounds is essentially no more complicated than that of the Jacobs-III bounds. Since the BBST-II bounds always provide an improvement, the BBST-II bounds should be used in preference to the Jacobs-III bounds.

Table 4 15-Node Diamond

n=15 b=27 c=2 C_c=13 t=61440

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
Jcb-III	0.9	0.475249	0.402272	0.990667	0.113146	0.515418	0.877521
BBST-II		0.476827	0.400694	0.971577	0.094056	0.494750	
Jcb-III	0.92	0.621424	0.298497	0.989653	0.069732	0.368229	0.919921
BBST-II		0.622552	0.297369	0.976015	0.056094	0.353463	
Jcb-III	0.94	0.771416	0.182782	0.990036	0.035838	0.218620	0.954198
BBST-II		0.772086	0.182112	0.981939	0.027741	0.209853	
Jcb-III	0.96	0.900704	0.078695	0.992504	0.013105	0.091800	0.979399
BBST-II		0.900985	0.078414	0.989104	0.009705	0.088119	
Jcb-III	0.98	0.980510	0.014302	0.996862	0.002050	0.016352	0.994812
BBST-II		0.980560	0.014252	0.996256	0.001444	0.015697	
Jcb-III	0.99	0.996545	0.002156	0.998989	0.000288	0.002444	0.998701
BBST-II		0.996552	0.002149	0.998898	0.000197	0.002346	

Table 5 (5,2) $G_{h,k}$ Graph

n=7 b=10 c=2 C_c=5 t=80 h=5 k=2

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
Jcb-III	0.9	0.912537	0.038264	0.969933	0.019132	0.057396	0.950801
BBST-II		0.934402	0.016399	0.963556	0.012755	0.029153	
Jcb-III	0.92	0.945489	0.022849	0.979763	0.011425	0.034274	0.968338
BBST-II		0.958545	0.009793	0.975954	0.007616	0.017409	
Jcb-III	0.94	0.970905	0.011206	0.987714	0.005603	0.016809	0.982111
BBST-II		0.977308	0.004803	0.985846	0.003735	0.008538	
Jcb-III	0.96	0.988175	0.003848	0.993947	0.001924	0.005771	0.992023
BBST-II		0.990374	0.001649	0.993305	0.001282	0.002931	
Jcb-III	0.98	0.997446	0.000556	0.998279	0.000277	0.000833	0.998002
BBST-II		0.997763	0.000239	0.998187	0.000185	0.000423	
Jcb-III	0.99	0.999426	0.000074	0.999537	0.000037	0.000112	0.999500
BBST-II		0.999468	0.000032	0.999525	0.000025	0.000057	

Considering the two more sophisticated bounds (Kruskal-Katona and Ball-Provan), we found in all cases that we tested, that the values delivered by both the Ball-Provan upper and lower bounds were never worse than the values delivered by the Kruskal-Katona upper and lower bounds. Again, the improvement can be relatively small (see, for example the upper bounds in Table 6). On the other hand, this improvement is sometimes quite large, as the ratio of their respective $r(p)$ values can be quite high (as in Table 7). The fact that the Ball-Provan bounds improve on the Kruskal-Katona bounds is also not surprising; in [B3], Ball and Provan perform an extensive comparison to show that their bounds consistently improve on the Kruskal-Katona bounds.

The implementations described in Chapter 4 suggest that the calculations of the Kruskal-Katona and Ball-Provan bounds are about equally complicated. Thus, in any situation where the Kruskal-Katona bounds are being considered, the Ball-Provan bounds would be preferred.

Thus we can, for most purposes, narrow the choice down to the BBST-II bounds or the Ball-Provan bounds. We found that both the Kruskal-Katona and Ball-Provan bounds

Table 6 10-node Ladder Graph

n=10 b=17 c=2 C_c=2 t=2584

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P K-K	0.9	0.930207 0.872668	0.040811 0.098350	0.980324 0.980940	0.009306 0.009922	0.050117 0.108272	0.971018
B-P K-K	0.92	0.958374 0.920429	0.024197 0.062142	0.987511 0.987700	0.004940 0.005129	0.029138 0.067272	0.982571
B-P K-K	0.94	0.979065 0.958531	0.011775 0.032309	0.992976 0.993015	0.002136 0.002175	0.013911 0.034483	0.990840
B-P K-K	0.96	0.992209 0.984436	0.004010 0.011783	0.996860 0.996864	0.000641 0.000645	0.004651 0.012428	0.996219
B-P K-K	0.98	0.998553 0.997317	0.000575 0.001811	0.999208 0.999208	0.000080 0.000080	0.000655 0.001891	0.999128
B-P K-K	0.99	0.999714 0.999540	0.000077 0.000251	0.999801 0.999801	0.000010 0.000010	0.000087 0.000261	0.999791

Table 7 8-node Complete Graph

n=8 b=28 c=7 C_c=8 t=262144

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P K-K	0.5	0.746441 0.498721	0.190651 0.438371	0.968901 0.980187	0.031809 0.043095	0.222461 0.481467	0.937092
B-P K-K	0.6	0.907526 0.616341	0.079243 0.370428	0.994386 0.995851	0.007617 0.009082	0.086859 0.379510	0.986769
B-P K-K	0.7	0.981079 0.776842	0.017165 0.221402	0.999167 0.999312	0.000923 0.001068	0.018088 0.222470	0.998244
B-P K-K	0.8	0.998575 0.942675	0.001323 0.057223	0.999939 0.999947	0.000041 0.000049	0.001364 0.057272	0.999898
B-P K-K	0.9	0.999990 0.998533	0.000009 0.001466	0.999999 0.999999	0.000000 0.000000	0.000010 0.001466	0.999999

always delivered better upper and lower bounds than the upper and lower BBST-II bounds. In [B5], Bauer et al. indicate that their bounds are weaker than the Kruskal-Katona bounds; in fact, Table 8 illustrates that they may be substantially worse.

Nonetheless, the calculation of the BBST-II bounds is definitely simpler than the calculation of the Ball-Provan bounds. Once the values for t, c, and C_c have been determined, the BBST-II bounds can be determined using only some simple summations, or by just a few table lookups. They also do not manipulate numbers as large as the Ball-Provan bounds do. However, it must be remembered that the implementation of algorithms for calculating t, c, and C_c required a substantial effort. If this amount of effort is to be expended, it appears worthwhile to implement the Ball-Provan bounds.

5.2.2. Lomonosov-Polesskii Bounds

The Lomonosov-Polesskii bounds do not form bounds on the subgraph counts of the reliability polynomial. Hence, they exhibit quite different behaviour than the other bounds. The subgraph bounds generally apply the same principle in obtaining both their upper and lower bound. We have found therefore that when comparing any two sets of these subgraph bounds both the lower and upper bound of one of them is better in all cases. They can therefore be ordered in a definite hierarchy. For the Lomonosov-Polesskii bounds, this is not the case. They are not developed using the approach of bounding the subgraph counts, and hence they cannot be placed in this hierarchy. It is possible for them to be better than one of the subgraph bounds in one case, and worse in another. Also, since the lower and upper bounds are not developed using the same principle, these two bounds may exhibit quite different behaviour. We therefore treat the upper and lower bounds separately.

The Lomonosov-Polesskii bounds do not appear to be very sophisticated. The lower bound uses only the value of n and c along with q in its calculation. It is very surprising therefore that there are cases where this lower bound actually improves on the Ball-Provan lower bound. In most of the cases tested this bound is worse than *all* of the subgraph lower bounds (as one might expect). However, for some values of p for the complete graphs (K_n) with $n>6$ this bound delivers a higher value and therefore is better than any of the other sets of bounds. Table 9 shows that for K_8, one obtains a better bound than the Ball-Provan lower bound for $0.3 \leq p \leq 0.9$. In testing higher values of p we encounter precision problems; therefore for Table 10 we use our BC implementation.

Table 8 10-node Complete Graph

$n=10$ $b=45$ $c=9$ $C_c=10$ $t=1e+08$

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.5	0.727790	0.252659	0.994716	0.014267	0.266927	0.980449
BBST-II		0.112874	0.867575	0.999970	0.019521	0.887096	
B-P	0.6	0.913862	0.083512	0.999191	0.001817	0.085329	0.997374
BBST-II		0.116076	0.881298	1.000000	0.002626	0.883924	
B-P	0.7	0.986971	0.012832	0.999923	0.000120	0.012952	0.999803
BBST-II		0.195679	0.804124	1.000000	0.000197	0.804321	
B-P	0.8	0.999445	0.000550	0.999997	0.000002	0.000553	0.999995
BBST-II		0.634453	0.365542	1.000000	0.000005	0.365547	
B-P	0.9	0.999999	0.000001	1.000000	0.000000	0.000001	1.000000
BBST-II		0.989327	0.010673	1.000000	0.000000	0.010673	

Table 9 8-node Complete Graph							
n=8 b=28 c=7 C_c=8 t=262144							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.290388	0.189481	0.547838	0.067969	0.257450	0.479869
L-P		0.281924	0.197945	0.547930	0.068061	0.266006	
B-P	0.4	0.522905	0.263366	0.855168	0.068897	0.332262	0.786271
L-P		0.602658*	0.183613	0.819755*	0.033484	0.217097	
B-P	0.5	0.746441	0.190651	0.968901	0.031809	0.222461	0.937092
L-P		0.846916*	0.090176	0.946578*	0.009486	0.099662	
B-P	0.7	0.981079	0.017165	0.999167	0.000923	0.018088	0.998244
L-P		0.994229*	0.004015	0.998470*	0.000226	0.004241	
B-P	0.9	0.999990	0.000009	0.999999	0.000000	0.000010	0.999999
L-P		0.999997*	0.000002	0.999999	0.000000	0.000002	

Table 10 8-node Complete Graph (BC Implementation)				
n=8 b=28 c=7 C_c=8 t=262144				
Bounds	p	lower	upper	$r(p)$
B-P	0.9	.999989678198	.999999388297	.000009710099
L-P		.999997203540*	.999999300000*	.000002096461
B-P	0.91	.999995274024	.999999699662	.000004425638
L-P		.999998661940*	.999999665192*	.000001003252
B-P	0.92	.999998039416	.999999864798	.000001825382
L-P		.999999413137*	.999999853199*	.000000440062
B-P	0.93	.999999282036	.999999945482	.000000663446
L-P		.999999769492*	.999999942352*	.000000172860
B-P	0.94	.999999776601	.999999980968	.000000204367
L-P		.999999921634*	.999999980404*	.000000058770
B-P	0.95	.999999944318	.999999994544	.000000050226
L-P		.999999978127*	.999999994531*	.000000016404
B-P	0.96	.999999989924	.999999998825	.000000008901
L-P		.999999995413*	.999999998853	.000000003440
B-P	0.97	.999999998898	.999999999839	.000000000941
L-P		.999999999388*	.999999999847	.000000000459

Most of the graphs tested have cohesions of at most two. The reason for this emphasis is that actual networks are generally quite sparse. Other than the complete graphs, the only other graph employed for testing with a cohesion greater than 2 is a 6-cohesive graph (Appendix Table A6.2). For this graph, the Lomonosov-Polesskii lower bound is the best for values of $p \geq 0.38$. This is (of course) only a very small set of tests. Nevertheless, it appears that this bound is very useful for graphs with high cohesion.

The fact that their are cases where the Lomonosov-Polesskii lower bound improves on the Ball-Provan lower bound while using less graph theoretic information shows that the Ball-Provan upper bound is not tight for networks in general.

The Lomonosov-Polesskii upper bound appears to deliver even better results. It uses the values of the cardinalities of the cuts in a cut basis; this information is not used by the subgraph bounds. Its calculation is very easy once these cardinalities are determined; also, it does not use the values for t or C_c. Typically, the upper bound is tightest whenever the Lomonosov-Polesskii lower bound is also tightest. However, Table 10 shows that this need not be the case. But this upper bound does delivers an improvement in many more cases than the lower bound. It is the tightest upper bound for a wide range of p values in K_6. Of even more interest is the fact that it improves on the Ball-Provan bounds in a number of the sparse networks tested. It gives the tightest results (for some ranges of p) for the $G_{h,k}$ graphs in which the number of paths h is large compared to the number of edges on a path k. It also delivers the lowest value for the 1979 version of the Arpanet (figure 8) which we tested for $0.7 \leq p \leq 0.98$ (Table 11). Recall that Frank and Chou [F2] give 0.98 as the value of p for the Arpanet, and hence this bound has practical import. But it is its performance on the 2-trees tested that is most impressive. It delivers the most accurate bound for some values of p in each of the 2-trees on which we tested it. It gives its best performance for the 2-tree diamond class of networks where it delivers a very large improvement over the Ball-Provan bounds (see Table 12).

There are, however, many cases in which the Lomonosov-Polesskii upper bound delivers much worse values than the Ball-Provan upper bound. One case is the networks that are a combination of a cycle and a complete graph (see Table 13). Even in the cases where they are better this is generally only for a subrange of values of p. It usually delivers better values than the Ball-Provan upper bound for a middle range of p (see Tables 9,10,11 and 12). This is where the subgraph bounds are at their worst.

5.3. Behaviour of Bounds

The performance of each bound is highly dependent on the graph parameters employed. In order to examine the behaviour of the bounds in more detail, in this section we study the effects of various graph operations on the accuracy of the bounds obtained.

Table 11 Arpanet (figure 8)

n=59 b=71 c=2 C_c=57 t=2.72817e+11

Bounds	p	lower	upper	$r(p)$
B-P	0.7	0.000148	0.002483	0.002335
L-P		0.000000	0.017175	0.017175
B-P	0.9	0.129562	0.839673	0.710111
L-P		0.015086	0.675115*	0.660029
B-P	0.92	0.229425	0.905370	0.675945
L-P		0.044769	0.780114*	0.735345
B-P	0.94	0.397043	0.939865	0.542822
L-P		0.123792	0.871119*	0.747327
B-P	0.96	0.642423	0.963840	0.321417
L-P		0.311079	0.941208*	0.630129
B-P	0.97	0.779467	0.975281	0.195815
L-P		0.468286	0.966686*	0.498399
B-P	0.98	0.901379	0.986323	0.084944
L-P		0.669216	0.985141*	0.315924
B-P	0.99	0.979896	0.995639	0.015743
L-P		0.882061	0.996286	0.114225

Table 12 15-node Diamond							
n=15 b=27 c=2 C_c=13 t=61440							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000074	0.000075	0.000320	0.000171	0.000246	0.000149
L-P		0.000001	0.000149	0.000157*	0.000008	0.000156	
B-P	0.5	0.008009	0.015688	0.058184	0.034488	0.050175	0.023696
L-P		0.000488	0.023208	0.023756*	0.000060	0.023268	
B-P	0.7	0.134717	0.158732	0.519635	0.226186	0.384918	0.293449
L-P		0.035268	0.258181	0.293453*	0.000004	0.258185	
B-P	0.9	0.784790	0.092731	0.924677	0.047156	0.139887	0.877521
L-P		0.549043	0.328478	0.877521*	0.000000	0.328478	
B-P	0.92	0.859553	0.060368	0.946875	0.026954	0.087322	0.919921
L-P		0.659729	0.260192	0.919921*	0.000000	0.260192	
B-P	0.94	0.921977	0.032221	0.966813	0.012615	0.044836	0.954198
L-P		0.773763	0.180435	0.954198*	0.000000	0.180435	
B-P	0.96	0.967377	0.012022	0.983526	0.004127	0.016150	0.979399
L-P		0.880890	0.098509	0.979399*	0.000000	0.098508	
B-P	0.98	0.992929	0.001883	0.995380	0.000568	0.002452	0.994812
L-P		0.964662	0.030150	0.994812*	0.000000	0.030151	
B-P	0.99	0.998438	0.000263	0.998775	0.000074	0.000338	0.998701
L-P		0.990370	0.008331	0.998701*	0.000000	0.008331	

Table 13 Cycle(20)-Complete(5) Graph							
n=24 b=30 c=2 C_c=190 t=2500							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.9	0.377378	0.014170	0.417759	0.026211	0.040380	0.391548
BBST-II		0.312628	0.078920	0.628929	0.237381	0.316301	
L-P		0.292477	0.099071	0.825838	0.434290	0.533361	
B-P	0.92	0.505140	0.011609	0.537435	0.020686	0.032295	0.516749
BBST-II		0.448169	0.068580	0.687777	0.171028	0.239609	
L-P		0.417290	0.099459	0.885011	0.368262	0.467721	
B-P	0.94	0.652660	0.007751	0.673630	0.013219	0.020970	0.660411
BBST-II		0.611769	0.048642	0.761219	0.100808	0.149450	
L-P		0.573479	0.086932	0.933723	0.273312	0.360244	
B-P	0.96	0.806729	0.003598	0.816153	0.005826	0.009424	0.810327
BBST-II		0.786315	0.024012	0.851926	0.041599	0.065612	
L-P		0.750826	0.059501	0.970024	0.159697	0.219198	
B-P	0.98	0.939403	0.000697	0.941163	0.001063	0.001761	0.940100
BBST-II		0.935142	0.004958	0.947347	0.007247	0.012204	
L-P		0.917387	0.022713	0.992427	0.052327	0.075040	
B-P	0.99	0.983032	0.000109	0.983300	0.000159	0.000268	0.983141
BBST-II		0.982347	0.000794	0.984211	0.001070	0.001865	
L-P		0.976145	0.006996	0.998102	0.014961	0.021956	

5.3.1. Doubling all Edges in a Cycle

Ball and Provan [B3] suggest that multiple (or parallel) edges can be utilized to allow the representation of networks containing links with different probabilities of failure. This is of interest, as it relaxes one of the major constraints of the model; all of the bounds, however, apply to the more general case of graphs with multiple edges.

The probability $P_{i,j}$ that at least one of a set of m multiple edges between nodes n_i and n_j is available is $1-q^m$. The use of multiple edges increases the range between $n-1$ and $b-c$, and thus increases the number of N_i values that the subgraph bounds must approximate. The effect of increasing this range is of interest. Using multiple edges, we can perform a test on this effect. If every edge of a graph is replaced by two edges with properly adjusted values of p, a second functionally equivalent graph can be created. For this new graph, however, the size of the range $n-1$ to $b-c$ is twice as large as in the original graph.

Table 14 shows results obtained from the bounds for the graphs obtained by starting with a simple 10 node cycle with one edge between the adjacent nodes and successively doubling the edges 3 times to end up with an equivalent 10 node cycle with eight edges between the adjacent nodes. The original cycle is a trivial graph and therefore all the bounds other than the Lomonosov-Polesskii bounds give the exact reliability value. The Lomonosov-Polesskii lower bound also gives the exact value. A number of very interesting observations can be made from this table. Comparing the performance of the BBST-II and Ball-Provan bounds, for both sets of bounds the range $r(p)$, which starts at 0 for the original cycle, increases with each successive doubling of the edges of the graph. However, this increase proceeds at a much slower rate for the Ball-Provan bounds than it does for the BBST-II bounds. By the third edge doubling (80 edge graph) the BBST-II bounds have been driven essentially completely apart. The Ball-Provan bounds are still delivering an $r(p)$ value that is less than 0.05.

Another very interesting observation comes from studying the behaviour of the Lomonosov-Polesskii bounds. They continue to deliver the same upper and lower bounds regardless of the number of times the edges are doubled. Doubling the edges doubles the

Table 14 Doubling Edges of a Cycle

Bounds	p	lower	Δlower	upper	Δupper	range	actual
$n=10$ $b=10$ $c=2$ $C_c=45$ $t=10$							
B&P	0.996094	0.999328	0.000000	0.999328	0.000000	0.000000	0.999328
BBSTII		0.999328	0.000000	0.999328	0.000000	0.000000	
L&P		0.999328	0.000000	0.999863	0.000535	0.000535	
$n=10$ $b=20$ $c=4$ $C_c=45$ $t=5120$							
B&P	0.9375	0.998898	0.000430	0.999498	0.000170	0.000600	0.999328
BBSTII		0.993268	0.006060	0.999693	0.000365	0.006425	
L&P		0.999328	0.000000	0.999863	0.000535	0.000535	
$n=10$ $b=40$ $c=8$ $C_c=45$ $t=2.62144e+06$							
B&P	0.75	0.988692	0.010636	0.999812	0.000484	0.011119	0.999328
BBSTII		0.306545	0.692783	1.000000	0.000672	0.693455	
L&P		0.999328	0.000000	0.999863	0.000535	0.000535	
$n=10$ $b=80$ $c=16$ $C_c=45$ $t=1.34218e+09$							
B&P	0.5	0.950115	0.049213	0.999937	0.000609	0.049822	0.999328
BBSTII		0.005788	0.993540	1.000000	0.000672	0.994212	
L&P		0.999328	0.000000	0.999863	0.000535	0.000535	

value of c and the cardinality of each of the cutsets in the cut basis. The Lomonosov-Polesskii lower bound delivers the exact reliability for all these graphs. This in fact is the type of graphs used by Lomonosov and Polesskii in [L5] to show that this bound is tight. A cycle on n nodes with m edges between the adjacent nodes is $2m$-cohesive and has the lowest value of R of any graph of n nodes and $c=2m$. Therefore no bound using *only* n and c can deliver a bound any higher than the Lomonosov-Polesskii lower bound.

5.3.2. Cycle to Ladder Transformation

In this section, we again start with a 10 node cycle. This time we add one edge at a time to produce a 10 node ladder. This is illustrated in Figure 10.

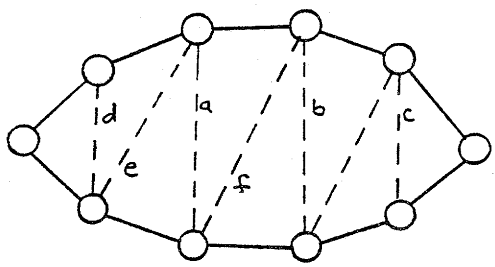

Figure 10

In this procedure, we transform the cycle which has the maximum number of minimum cardinality cuts ($C_c=\binom{n}{2}$) into a graph that only contains 2 minimum cardinality cuts; the values of c and n remain constant.

Let G_a denote the graph formed by the addition of edge "a" to the cycle. The next graph formed by the addition of edge "b" to G_a is G_b. All these intermediate graphs are subgraphs of the 2-tree ladder and therefore we can obtain exact values for their reliabilities. The values delivered by the bounds for $p=0.98$ are displayed in Table 15. The graphs G_a and G_b are still trivial graphs; therefore, both the BBST-II and Ball-Provan bounds deliver their exact reliability. For G_c, these two subgraph bounds have a single N_i value to approximate. This number increases by one for each graph until the final ladder graph where these bounds are approximating 5 of the N_i values.

Looking at the behaviour of the lower bound, the Ball-Provan lower bound increases as the actual probability increases with each successive edge addition. However, as shown by the fact that Δlower increases, it does not keep pace with the actual increases in the value of R. The BBST-II lower bound begins to increase with the increasing actual reliability for G_c and G_d; surprisingly, after this point this lower bound actually decreases as more edges are added, despite the fact that the actual R value continues to increase.

At a p value as high as 0.98 the most important of the approximated subgraph counts is N_{b-c-1}. These lower bounds use the known value for t (N_{n-1}) to obtain an underapproximation for this value. As the distance between N_{n-1} and N_{b-c-1} increases, this approximation becomes worse.

This procedure also shows a problem with the Lomonosov-Polesskii lower bound, since it only uses the values n and c. Since these two values remain constant for these graphs, it continues to deliver the same bound for all of them despite the increasing value of R.

Switching attention to the performance of the BBST and Ball-Provan upper bounds, we see that after G_c (where they first must approximate an N_i value) Δupper actually decreases even though the number of N_i values that need to be approximated increases. Again the

Table 15 Cycle to Ladder Transformation

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
n=10 b=10 c=2 C_c=45 t=10 Cycle							
B-P	0.98	0.983822	0.000000	0.983822	0.000000	0.000000	0.983822
BBST-II		0.983822	0.000000	0.983822	0.000000	0.000000	
L-P		0.983822	0.000000	0.996406	0.012584	0.012583	
n=10 b=11 c=2 C_c=21 t=34 G_a							
B-P	0.98	0.991826	0.000000	0.991826	0.000000	0.000000	0.991826
BBST-II		0.991826	0.000000	0.991826	0.000000	0.000000	
L-P		0.983822	0.008004	0.996797	0.004971	0.012974	
n=10 b=12 c=2 C_c=13 t=90 G_b							
B-P	0.98	0.994815	0.000000	0.994815	0.000000	0.000000	0.994815
BBST-II		0.994815	0.000000	0.994815	0.000000	0.000000	
L-P		0.983822	0.010993	0.997187	0.002372	0.013365	
n=10 b=13 c=2 C_c=9 t=194 G_c							
B-P	0.98	0.996168	0.000177	0.996632	0.000287	0.000464	0.996345
BBST-II		0.995682	0.000663	0.996829	0.000484	0.001147	
L-P		0.983822	0.012523	0.997578	0.001233	0.013756	
n=10 b=14 c=2 C_c=5 t=418 G_d							
B-P	0.98	0.997531	0.000346	0.998061	0.000184	0.000530	0.997877
BBST-II		0.996478	0.001399	0.998291	0.000414	0.001813	
L-P		0.983822	0.014055	0.997970*	0.000093	0.014147	
n=10 b=15 c=2 C_c=4 t=773 G_e							
B-P	0.98	0.997871	0.000428	0.998446	0.000147	0.000574	0.998299
BBST-II		0.996200	0.002099	0.998654	0.000355	0.002454	
L-P		0.983822	0.014477	0.998361*	0.000062	0.014539	
n=10 b=16 c=2 C_c=3 t=1419 G_f							
B-P	0.98	0.998212	0.000501	0.998824	0.000111	0.000611	0.998713
BBST-II		0.995867	0.002846	0.999003	0.000290	0.003136	
L-P		0.983822	0.014891	0.998753*	0.000040	0.014930	
n=10 b=17 c=2 C_c=2 t=2584 Ladder							
B-P	0.98	0.998553	0.000575	0.999208	0.000080	0.000655	0.999128
BBST-II		0.995469	0.003659	0.999344	0.000216	0.003875	
L-P		0.983822	0.015306	0.999144*	0.000016	0.015322	

most important subgraph count is N_{b-c-1}. These upper bounds use the known value for N_{b-c} to obtain overapproximations on this value. Increasing the number of N_i values approximated does not increase the distance between these two values; therefore, the accuracy of the approximation depends primarily on the actual relation between N_{b-c} and N_{b-c-1} (or actually C_c and C_{c+1}). For graphs near the cycle (which has the maximum C_c value for a given n), the bounds must give a larger overapproximation than they can give for the smaller C_c values for graphs nearer the ladder.

In this example, the Lomonosov-Polesskii upper bound again shows very good performance. It does not give the exact value for the cycle, G_a, or G_b as do the subgraph bounds. However, by the time G_d is reached, its Δupper value is already half that of the Ball-Provan upper bound. For a cycle, almost every edge must be in two cutsets of the cut basis and therefore a fair amount of overcounting takes place, increasing the resulting value obtained for the bound. As the successive graphs move closer to the ladder the cutsets become more disjoint and thus the bound improves.

5.3.3. Subgraph Counts

As has been seen, the performance of the subgraph bounds is dependent on the accuracy of its approximations of the different N_i values. Therefore, it is of interest to examine the actual values these bounds obtain for these approximations. In this section we examine the subgraph count approximations calculated by the best of the subgraph bounds (the Ball-Provan bounds), for a series of $G_{h,k}$ graphs. The final value delivered by any of the subgraph bounds consists of these N_i values "weighted" by the appropriate factor determined by p and i. Hence, for the high value of p (0.98) used in the previous section the most important of these are the approximations for the N_{b-c-1} values.

For $G_{h,k}$ graphs we can easily obtain the actual subgraph counts from equation (35) to compare against the overapproximations of the Ball-Provan upper and the underapproximations of the Ball-Provan lower bounds. A $G_{h,k}$ graph is obtained by connecting two extreme points by h vertex-disjoint paths with k-1 intermediate vertices. The resulting graph has $n=hk-h+2$, b$=hk$, and $c=2$. An interesting property of such a graph is the number of unknown N_i values $(b-c)$-$(n-1)$ is equal to h-4 and is therefore independent of the value of k. We use this fact and test a set of graphs with fixed k=3. We begin with the trivial graph where h=4 (any graph with $h\leq 4$ is trivial for all values of k) and consider the graphs obtained by successively incrementing h (i.e. adding another path, see Figure 11).

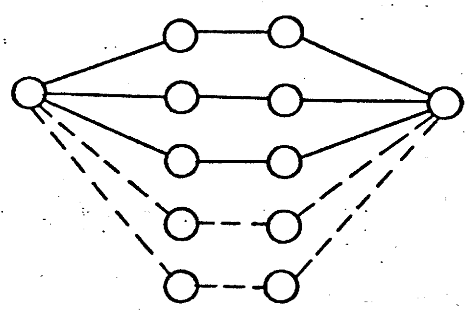

Figure 11

The number of terms which the Ball-Provan bounds approximate successively increases by one for this series of graphs. The actual subgraph counts and the approximations calculated by the Ball-Provan bounds for these graphs are displayed in Table 16.

As illustrated in the previous section, for high values of p, the most important of these approximations is the one for N_{b-c-1}. Therefore, the Δlower and Δupper values for this term are of interest. For the graph with h=5 the N_{b-c-1} value is the only one approximated by the bounds. Here Δupper is significantly less than Δlower for N_{b-c-1}. However, by the next graph (with h=6) where two N_i values are approximated, Δlower is already larger than Δupper. As the distance between N_{n-1} and N_{b-c-1} increases this difference becomes ever larger. So, for high values of p we expect the error in the lower bound to increase as h and the number of approximated N_i values increases (see Appendix A4).

We observe that the Δlower values increase for each successive graph. The ratio between C_c and n is increasing and therefore we have the same situation we would have had in the previous section had we moved from the ladder to a cycle.

Considering the actual subgraph count values we see that they are relatively small near N_{b-c} and quite large near N_{n-1}. This means that the assumption that N_{b-c-1} is the dominant unknown term is only applicable when p is quite near one.

Table 16 Subgraph Counts						
$n=10$ $b=12$ $c=2$ $C_c=12$ $t=108$ $h=4$ $k=3$						
i		lower	Δlower	upper	Δupper	actual
10	$b-c$	54	0	54	0	54
9	$n-1$	108	0	108	0	108
$n=12$ $b=15$ $c=2$ $C_c=15$ $t=405$ $h=5$ $k=3$						
13	$b-c$	90	0	90	0	90
12		238	32	325	55	270
11	$n-1$	405	0	405	0	405
$n=14$ $b=18$ $c=2$ $C_c=18$ $t=1458$ $h=6$ $k=3$						
16	$b-c$	135	0	135	0	135
15		444	96	621	81	540
14		918	297	1961	746	1215
13	$n-1$	1458	0	1458	0	1458
$n=16$ $b=21$ $c=2$ $C_c=21$ $t=5103$ $h=7$ $k=3$						
19	$b-c$	189	0	189	0	189
18		737	208	1057	112	945
17		1776	1059	4172	1337	2835
16		3295	1808	8387	3284	5103
15	$n-1$	5103	0	5103	0	5103
$n=18$ $b=24$ $c=2$ $C_c=24$ $t=17496$ $h=8$ $k=3$						
22	$b-c$	252	0	252	0	252
21		1153	359	1660	148	1512
20		3231	2439	7790	2120	5670
19		6842	6766	27714	14106	13608
18		12030	8382	38850	18438	20412
17	$n-1$	17496	0	17496	0	17496
$n=20$ $b=27$ $c=2$ $C_c=27$ $t=59049$ $h=9$ $k=3$						
25	$b-c$	324	0	324	0	324
24		1707	561	2457	189	2268
23		5473	4733	13365	3159	10206
22		13058	17560	55545	24927	30618
21		25790	35446	138407	77171	61236
20		42962	35770	153117	74385	78732
19	$n-1$	59049	0	59049	0	59049
$n=22$ $b=30$ $c=2$ $C_c=30$ $t=196830$ $h=10$ $k=3$						
28	$b-c$	405	0	405	0	405
27		2418	822	3475	235	3240
26		8758	8252	21500	4490	17010
25		23328	37908	102226	40990	61236
24		51082	102008	388424	235334	153090
23		95341	167099	737852	475412	262440
22		150250	144995	630082	334837	295245
21	$n-1$	196830	0	196830	0	196830

5.3.4. Combination Graphs

In this section, we look briefly at the effect that combining different types of subgraphs has on the bounds. A very interesting member of this group is the combination of a cycle and a complete graph, since the complete graph is the densest and hence the most reliable graph while the cycle is one of the least reliable graphs. We call this type of graph a $G_{k,c}$ graph. It has already been shown that a $G_{k,c}$ graph in which the number of nodes in the complete subgraph is small while the number of nodes in the cycle is comparatively large is an example where the value delivered by the Lomonosov-Polesskii upper bound is quite poor (see Table 13). Due to the large cycle subgraph there will be a lot of overlap of the edges among the cutsets which loosens the obtained bound. In Table 17 we look at a different $G_{k,c}$ graph; here the complete subgraph has more nodes than the cycle. The Lomonosov-Polesskii upper bound actually delivers the best value for a range of p values. As the complete graph is relatively large here, there will be less overlap of edges among cutsets and therefore a better bound is obtained.

For both of these graphs the BBST-II bounds deliver significantly worse values than the Ball-Provan bounds. The BBST-II bounds do not appear to handle the case in which one section of a network is very dense and another section is quite sparse very well. In Table 18 we show the results of a test of an extreme case of this type of situation; a 6 node complete graph combined with a single edge. This graph actually has the same number of trees as the original 6 node complete graph but the c and C_c values have been decreased down to 1. For this case, the BBST-II bounds are worse than the Ball-Provan bounds by orders of magnitude.

In Table 19 we show the results of a test on a 6 node complete graph with a single new node added to it; this time the node is connected to some arbitrary node of the complete subgraph with 15 multiple edges. The second half of the table shows the performance of the bounds on the 6 node complete graph itself. This graph has the same c and C_c values and very close to the same reliability as the original 6 node graph. However it has fifteen times as many trees. Both the BBST-II and Ball-Provan upper bounds are somewhat weaker than for the original K_6. Since the number of trees has been drastically altered, both their lower bounds have been weakened to a more significant extent. Again, it is the BBST-II bound which suffers the worst effects especially for the lower values of p. It should be noted, however, that the effect this operation had on the relative performance between the BBST-II and Ball-Provan bounds was not as great as the previous operation (Table 18).

Table 17 Cycle(5)-Complete(8) Graph

n=12 b=33 c=2 C_c=10 t=1.31072e+06

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.5	0.065860	0.109845	0.404236	0.228531	0.338377	0.175705
BBST-II		0.006654	0.169051	0.941900	0.766195	0.935247	
L-P		0.003174	0.172531	0.299503*	0.123798	0.296329	
B-P	0.6	0.151125	0.181377	0.583088	0.250586	0.431963	0.332502
BBST-II		0.006781	0.325721	0.979331	0.646829	0.972551	
L-P		0.019591	0.312911	0.492189*	0.159687	0.472598	
B-P	0.7	0.313481	0.213811	0.707171	0.179879	0.393689	0.527292
BBST-II		0.007619	0.519673	0.981048	0.453756	0.973429	
L-P		0.085025	0.442267	0.684700*	0.157408	0.599675	
B-P	0.8	0.574501	0.162703	0.821460	0.084256	0.246960	0.737204
BBST-II		0.032974	0.704230	0.981172	0.243968	0.948198	
L-P		0.274878	0.462326	0.849270	0.112066	0.574393	
B-P	0.9	0.869491	0.049048	0.934868	0.016329	0.065377	0.918539
BBST-II		0.346274	0.572265	0.983792	0.065253	0.637518	
L-P		0.659002	0.259537	0.960595	0.042056	0.301593	

Table 18 K_n + One Node Connected By Single Edge							
n=7 b=16 c=1 C_c=1 t=1296							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.9	0.890117	0.009829	0.900000	0.000054	0.009882	0.899946
BBST-II		0.572674	0.327272	0.949081	0.049135	0.376407	
L-P		0.296123	0.603823	0.899955*	0.000009	0.603832	
B-P	0.92	0.913647	0.006335	0.920000	0.000018	0.006353	0.919982
BBST-II		0.666854	0.253128	0.953962	0.033980	0.287108	
L-P		0.366924	0.553058	0.919985*	0.000003	0.553061	
B-P	0.94	0.936414	0.003582	0.940000	0.000004	0.003586	0.939996
BBST-II		0.767626	0.172370	0.960723	0.020727	0.193097	
L-P		0.457616	0.482380	0.939996*	0.000000	0.482381	
B-P	0.96	0.958403	0.001596	0.960000	0.000001	0.001597	0.959999
BBST-II		0.867125	0.092874	0.970025	0.010026	0.102901	
L-P		0.576717	0.383282	0.960000	0.000001	0.383283	
B-P	0.98	0.979600	0.000400	0.980000	0.000000	0.000400	0.980000
BBST-II		0.951819	0.028181	0.982737	0.002737	0.030918	
L-P		0.740469	0.239531	0.980000	0.000000	0.239531	
B-P	0.99	0.989900	0.000100	0.990000	0.000000	0.000100	0.990000
BBST-II		0.982236	0.007764	0.990716	0.000716	0.008480	
L-P		0.850306	0.139694	0.990000	0.000000	0.139694	

Finally, in Table 20 we look at results for an 8 node graph constructed from a basic K_8 with one new node connected to it by a single edge and a second new node connected with fifteen multiple edges. Here we have multiplied the number of trees from the original K_8 by fifteen as well as altering C_c and c to one. Here, even for high values of p there is a very significant difference between the values delivered by the BBST-II and Ball-Provan bounds.

5.3.5. Summary

The last four sections have illustrated some very interesting observations concerning the behaviour of the bounds. Section 5.3.1 indicated that for the subgraph bounds increasing the number of N_i values that the bounds are approximating (increasing the range from n–1 to b–c) has a dramatic negative effect on accuracy. In fact, this has a greater relative effect on the BBST-II bounds than on the Ball-Provan bounds.

The transformation from a cycle to a ladder graph developed in section 5.3.2 showed that increasing this range (n–1 to b–c) does not necessarily decrease the performance of both the upper bounds. In this situation, the performance of the upper bounds actually improves as this range is successively incremented. At this high value of p (0.98), the N_{b-c-1} term becomes dominant. Since the distance between it and the N_{b-c} term, which the subgraph upper bounds use to approximate it, is not increasing, the most important factor becomes how close an approximation is derived for it by the specific N_{b-c} of the particular graph. The C_c values of the graphs near the cycle, which are large with respect to n, yield worse approximations for the respective C_{c+1} values of the graphs than the lower C_c values of the graphs nearer the ladder.

The effect of increasing the range from n–1 to b–c is further examined in section 5.3.3 where a series of $G_{h,k}$ graphs with increasing values of h are tested. In this section the actual approximations delivered by the best of the subgraph bounds (the Ball-Provan bounds) for the individual N_i are determined. Table 16 shows the approximation used by the lower bound for N_{b-c-1} becoming much worse than the approximation used by the upper bound as the range of approximated N_i values increases.

In section 5.3.4 it is seen that taking a complete graph and altering the C_c and c values by adding a single node and edge (Table 18) has a more drastic effect on the performance of the bounds than does significantly altering the number of trees of the graph by connecting to the new node with a relatively large number of multiple edges (Table 19). This again shows the greater importance of the relationship between the C_c and C_{c+1} than other factors such as the number of approximated N_i values and the t value at high values of p. As well, the performance of the BBST-II bounds again degraded to a much greater extent than did the Ball-Provan bounds.

Table 19a K_6 + One Node Connected By 15 Multiple Edges

n=7 b=30 c=5 C_c=6 t=19440

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.5	0.560513	0.254404	0.910995	0.096078	0.350482	0.814917
BBST-II		0.032892	0.782026	0.999261	0.184344	0.966369	
L-P		0.641375*	0.173542	0.853189*	0.038272	0.211814	
B-P	0.7	0.932454	0.052513	0.992724	0.007757	0.060270	0.984967
BBST-II		0.106825	0.878142	0.999959	0.014992	0.893134	
L-P		0.956759*	0.028208	0.987909*	0.002942	0.031150	
B-P	0.9	0.999724	0.000216	0.999953	0.000013	0.000229	0.999940
BBST-II		0.929202	0.070738	0.999993	0.000053	0.070791	
L-P		0.999792*	0.000148	0.999950*	0.000010	0.000158	
B-P	0.92	0.999918	0.000062	0.999984	0.000004	0.000066	0.999980
BBST-II		0.971669	0.028311	0.999996	0.000016	0.028328	
L-P		0.999932*	0.000048	0.999984	0.000004	0.000052	
B-P	0.94	0.999983	0.000012	0.999996	0.000001	0.000013	0.999995
BBST-II		0.992314	0.007681	0.999999	0.000004	0.007685	
L-P		0.999984*	0.000011	0.999996	0.000001	0.000012	

Table 19b K_6

n=6 b=15 c=5 C_c=6 t=1296

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.5	0.755219	0.059722	0.839783	0.024842	0.084564	0.814941
BBST-II		0.491586	0.323355	0.887094	0.072153	0.395508	
L-P		0.712266	0.102675	0.853215	0.038274	0.140949	
B-P	0.7	0.971910	0.013057	0.989103	0.004136	0.017193	0.984967
BBST-II		0.841059	0.143908	0.996673	0.011706	0.155614	
L-P		0.968083	0.016884	0.987909*	0.002942	0.019826	
B-P	0.9	0.999890	0.000050	0.999950	0.000010	0.000060	0.999940
BBST-II		0.998700	0.001240	0.999974	0.000034	0.001274	
L-P		0.999851	0.000089	0.999950	0.000010	0.000099	
B-P	0.92	0.999966	0.000014	0.999983	0.000003	0.000017	0.999980
BBST-II		0.999596	0.000384	0.999990	0.000010	0.000394	
L-P		0.999951	0.000029	0.999984	0.000004	0.000033	
B-P	0.94	0.999993	0.000002	0.999996	0.000001	0.000003	0.999995
BBST-II		0.999915	0.000080	0.999997	0.000002	0.000082	
L-P		0.999988	0.000007	0.999996	0.000001	0.000008	

Table 20 K_6 + One Node By Single Edge + One Node By 15 Multiple Edges							
n=8 b=31 c=1 C_c=1 t=19440							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.7	0.392413	0.297064	0.699823	0.010346	0.307411	0.689477
BBST-II		0.007610	0.681867	0.967742	0.278265	0.960133	
L-P		0.018713	0.670764	0.691536*	0.002059	0.672823	
B-P	0.9	0.824659	0.075287	0.900000	0.000054	0.075341	0.899946
BBST-II		0.171465	0.728481	0.968973	0.069027	0.797508	
L-P		0.224580	0.675366	0.899055*	0.000009	0.675375	
B-P	0.92	0.867944	0.052038	0.920000	0.000018	0.052056	0.919982
BBST-II		0.277461	0.642521	0.970174	0.050192	0.692714	
L-P		0.290739	0.629243	0.919985*	0.000003	0.629246	
B-P	0.94	0.908401	0.031595	0.940000	0.000004	0.031599	0.939996
BBST-II		0.432299	0.507697	0.972480	0.032484	0.540181	
L-P		0.379793	0.560203	0.939996*	0.000000	0.560203	
B-P	0.96	0.944852	0.015147	0.960000	0.000001	0.015148	0.959999
BBST-II		0.637346	0.322653	0.976842	0.016843	0.339496	
L-P		0.503317	0.456682	0.960000	0.000001	0.456683	
B-P	0.98	0.975917	0.004083	0.980000	0.000000	0.004083	0.980000
BBST-II		0.862806	0.117194	0.984986	0.004986	0.122180	
L-P		0.684389	0.295611	0.980000	0.000000	0.295611	
B-P	0.99	0.988940	0.001060	0.990000	0.000000	0.001060	0.990000
BBST-II		0.954497	0.035503	0.991365	0.001365	0.036867	
L-P		0.813105	0.176895	0.990000	0.000000	0.176895	

Because of their different nature the Lomonosov-Polesskii bounds are not affected at all when the edges of a graph are doubled. Therefore there comes a point as the edges of a graph are successively doubled when the Lomonosov-Polesskii bounds both deliver better bounds than do any of the subgraph bounds. Section 5.3.2 shows a major problem due to the unsophisticated nature of the Lomonosov-Polesskii lower bound. Even though R actually increases if n is held constant and a number of edges are succesively added to a graph, this bound continues to deliver the same value as it only uses these two values (c and n). The Lomonosov-Polesskii upper bound performs much better in this situation. Its perfomance as shown by all of these operations is mainly dependent on the overlap in cutsets of the minimal cut basis. If there is a good deal of overlap between the edges in these cutsets such as is the case for graphs that contain large sets of nodes in series (i.e. cycles or $G_{h,k}$ graphs with high values of k) these edges are counted more than once for this upper bound and thus its accuracy is degraded. If there are few cases where an edge is a member of more than one of the cutsets in the cutbasis such as in the case for graphs that contain a large number of parallel branchings of nodes (i.e. $K_{2,n-2}$ graphs , diamonds, and $G_{h,k}$ graphs with large values of h but small values of k) this upper bound delivers very accurate values.

Chapter 6

Conclusions and Future Research

6.1. Summary and Conclusions

In this thesis, we have investigated bounds on network reliability. We summarize the thesis here, to reiterate conclusions drawn and to point out the original contributions. Reliability is defined as the probability that every node in the network is able to communicate with every other node. We have modeled networks using probabilistic graphs consisting of perfectly reliable nodes, and undirected edges with equivalent but statistically independent failure probabilities. In this graph theoretic model, reliability corresponds to the probability that the probabilistic graph representing the network is connected. This measure is often termed probabilistic connectedness and we denote it as R.

In Chapter 2, we developed the need for measures of reliability, and established the wide applicability of the chosen model; we then reviewed previous research on the determination of probabilistic connectedness. R can be represented conveniently by the reliability polynomial as a summation of the counts of the number of spanning subgraphs containing equivalent numbers of edges. However, computing the exact value of R is not feasible. This motivated the development of a number of methods of obtaining lower and upper bounds on the value of R.

Most of these bounds have been developed using the general principle of obtaining an upper bound on R by using facts known about the individual subgraph counts of the reliability polynomial to find a set of overapproximations of them. Similarly, a lower bound is obtained by finding a set of underapproximations for these individual subgraph counts. These bounds all use the value for c, the edge connectivity of the graph, as well as the actual values of two of the subgraph counts which can be obtained directly from the values for t, the number of spanning trees in the graph and C_c, the number of minimum cardinality cuts in the graph.

The simplest of these sets of bounds are the Jacobs bounds [J1], which simply make the weakest possible assumptions about the unknown subgraph counts. In his original bounds, Jacobs does not use the values for t and C_c. Frank and Van Slyke [V2] modify the Jacobs bounds to use these values to obtain what we refer to as the Jacobs-II bounds. The BBST [B5] and Kruskal-Katona [V2] bounds utilize theorems that are valid for any systems that are coherent, to obtain bounds on the subgraph counts. These theorems are applicable since the set of subgraph counts is coherent. The Kruskal-Katona bounds use a stronger theorem, and are more sophisticated than the BBST bounds. The Ball-Provan bounds [B3] utilize a theorem that applies to "shellable independence systems" to form bounds on the counts for a slightly different form of the reliability polynomial. The set of counts for this polynomial form a shellable independence system and therefore this theorem is also applicable. The final bounds on subgraph counts are due to Leggett [L2], who purports to obtain bounds using graph theoretic structure.

Lomonosov and Polesskii develop a set of bounds using a completely different approach. They develop a lower bound using the fact that each of a set of edge disjoint spanning subgraphs (in this case, spanning trees) of a graph must fail before the graph can fail. They also develop an upper bound, using the fact that at least one of the edges in every cutset of a cut basis of a graph must be available for the graph to be connected.

In chapter 3, we took a closer look at Leggett's bounds and proved that they are incorrect, and that the errors are fundamental. This is one of the main contributions of the thesis and appears also in [H3]; for many years, the correctness of Leggett's bounds has been in doubt. Owing to the number of errors, these "bounds" were totally eliminated from consideration in the subsequent analysis.

In chapter 4, we investigated the implementations of the rest of the sets of bounds. We showed that the bounds and all the values used by them can be calculated in polynomial time. In the process, we refined two of the bounds, the Jacobs and the BBST bounds, to employ all of the information calculated. The modified bounds use both t and C_c, in both their upper and lower bounds. We named these versions the Jacobs-III and BBST-II bounds.

The Jacobs-III bounds and the BBST-II bounds can both be easily implemented and calculated once the graph theoretic values of c, t, and C_c have been determined. The implementation of either the Kruskal-Katona or the Ball-Provan bounds is definitely more complicated than that of the other two bounds. However, the overriding difficulty appears to remain the calculation of the graph theoretic values c, t, and C_c. The calculation of the Lomonosov-Polesskii bounds is trivial once the values for c and the cardinalities of the cuts in a cut basis have been determined. Another contribution of the thesis is the implementations themselves. Although the bounds employing subgraph counts have been implemented before, this appears to be the first implementation of the Lomonosov-Polesskii bounds.

In chapter 5, we discussed the results obtained when we tested the bounds on a number of graphs for which we were able to obtain the exact reliability. Our tests indicate that the four sets of bounds which employ subgraph counts form a definite hierarchy with respect to accuracy. This ranking from best to worst is:

1. Ball-Provan bounds
2. Kruskal-Katona bounds
3. BBST-II bounds
4. Jacobs-III bounds

The BBST-II bounds are as easy to implement as the Jacobs-III bounds; furthermore, the difficulty of implementation for the Kruskal-Katona and for the Ball-Provan bounds was of roughly the same magnitude. Hence the choice among these bounds can be effectively narrowed down to either the BBST-II or the Ball-Provan bounds. Since the major difficulty of obtaining the bounds is actually the calculation of the graph theoretic values (c, t, and C_c), and since there is a marked improvement in performance delivered by the Ball-Provan bounds over the BBST-II bounds, our contention is that the Ball-Provan bounds are generally the best choice among bounds employing subgraph counts.

The Lomonosov-Polesskii bounds do not appear to be very sophisticated. The performance they deliver is therefore quite amazing. The lower bound uses only c and n in its calculations, and in most cases delivered poor results. Nevertheless, for the few graphs we tested which had $c>5$, it delivered a better bound than even the Ball-Provan bounds for certain ranges of p.

The upper bound uses some information that the four subgraph bounds did not use (cardinality of the cutsets in a cut basis of the graph). It still does not appear to be very sophisticated as it does not use the values for t and C_c. However, for a significant number of graphs, and for certain ranges of p, it delivers a lower value than the Ball-Provan upper bound. For cases like the 2-tree diamonds, it improves remarkably on the Ball-Provan bound.

The Lomonosov-Polesskii bounds do not fit into the hierarchy developed for the other bounds; at times they are an improvement, but at other times they are worse than any of these other bounds. However, they can be used in conjunction with any of these other bounds in a complementary fashion. Therefore, the tightest bounds currently available would be a combination of the Lomonosov-Polesskii and Ball-Provan bounds, choosing the lowest of the two upper bounds and the highest of the two lower bounds. This can deliver quite large improvement over taking either one separately.

Thus, another significant contribution of this thesis has been to establish that the best available bound does not always arise via the standard technique of bounding subgraph counts. Hence, although the Ball-Provan bounds are very sophisticated combinatorial analyses of subgraph counts, they do always outperform the substantially different Lomonosov-Polesskii bounds.

6.2. Future Research

The results obtained in the thesis suggest some potentially fruitful avenues for further research. First we consider the bounds which use subgraph counts. One way of improving the existing bounds would be to use some graph theoretic property that does not apply for general coherent, shellable, or polyhedral systems; we already have bounds that are tight for these systems. However, the possible set of spanning subgraph counts in the reliability polynomial only forms a subset of any of these types of systems. Therefore, it might still be possible to find a tighter bound that does not apply generally for any of these systems, yet still applies for the set of spanning subgraph counts of a graph.

Another significant improvement would be to obtain any of the intermediate subgraph counts exactly. However, it is an open problem whether any of the N_i values for $n-1<i<b-c$ can be obtained in polynomial time. Currently, there are no known polynomial time algorithms for calculating any of these values. In the last section of chapter 5, it was illustrated that for values of p that are reasonable for computer networks (say, $p=0.98$) the dominant factor in the performance of both the upper and lower bounds of any spanning subgraph bound is the accuracy of their approximation of the N_{b-c-1} value. Clearly, if the N_{b-c-1} value could be determined exactly a marked improvement could be expected in these bounds for p in this range.

The remarkable performance of the relatively unsophisticated Lomonosov-Polesskii bounds indicates strong promise in pursuing their approach to obtaining bounds on R. Their lower bound makes use of the fact that for the network to fail every member of any set of edge disjoint spanning trees must fail. A tree is very unreliable since the failure of any one of its edges causes it to fail. It should be possible to obtain an improved lower bound if we partition a graph into edge disjoint spanning subgraphs which are more reliable than trees. We would then obtain a higher value when the product of the reliability of these subgraphs is taken. The exact reliability of these subgraphs would have to be known, and so candidates for these subgraphs are the partial 2-trees. The major effort in this approach would be the algorithm for suitably partitioning the original graph into the edge disjoint subgraphs. We are currently pursuing this line of research.

The Lomonosov-Polesskii upper bound is based on the fact that at least one edge in every cutset of a cut basis must be available for a graph to be connected. The cutsets of a cut basis are not generally edge disjoint. Therefore, the product of the probabilities that at least one edge in every cutset is available results in overcounting for the cases where an edge is in two cutsets; thus a high value is obtained. An area which merits serious future research is to exploit information about the intersections of the edge cutsets, to reduce the current overcounting and therefore obtain a more accurate bound.

Breakthroughs in any of the suggested areas of research could have a dramatic effect on the accuracy of bounds for R. Moreover, each would extend the research undertaken in this thesis in an interesting way.

References

[A1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley, 1974.

[B1] M.O. Ball, "Computing network reliability", *Operations Research*, Vol 27, 1979, pp. 823-838.

[B2] M.O. Ball, "Complexity of network reliability computations", *Networks*, Vol. 10, 1980, pp. 153-165.

[B3] M.O. Ball and J.S. Provan, "Calculating bounds on reachability and connectedness in stochastic networks",*Networks*, Vol. 13, 1983, pp. 253-278.

[B4] M.O. Ball and J.S. Provan, "Bounds on the reliability polynomial for shellable independance systems", *SIAM J. Alg. Disc. Meth.*, Vol. 3, 1982, pp. 166-181.

[B5] D. Bauer, F. Boesch, C. Suffel and R. Tindell, "Combinatorial optimization problems in the analysis and design of probabilistic networks", *Networks*, to appear.

[B6] R.E. Bixby, "The minimum number of edges and vertices in a graph with edge connectivity n and m n-bonds", *Networks*, Vol. 5, 1975, pp. 253-298.

[B7] F.T. Boesch, F. Harary and J.A. Kabell, "Graphs as models of communication network vulnerability: connectivity and persistence", *Networks*, Vol. 11, 1981, pp. 57-63.

[B8] J.A. Buzacott, "A recursive algorithm for finding reliability measures related to the connection of nodes in a graph", *Networks*, Vol. 10, 1980, pp. 311-327.

[E1] S. Even, *Graph Algorithms*, Computer Science Press, Potomac, 1979.

[E2] S. Even and R.E. Tarjan, "Network flow and testing graph connectivity," *SIAM J. Computing*, Vol. 4, 1975, pp. 507-508.

[E3] Exoo, G., "On a measure of communication network vulnerability", *Networks*, Vol. 12, 1982, pp. 405-409.

[F1] H. Frank and I.T. Frisch, "Analysis and design of survivable networks", *IEEE Trans.Comm.*, Vol. COM-18, 1970, pp. 501-519.

[F2] H. Frank and W. Chou, "Network properties of the ARPA computer network", *Networks*, Vol. 4, 1974, pp. 213-239.

[F3] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, Princeton, N.J., 1962.

[F4] Y. Fu, "Application of topological methods to probabilistic communication networks", *IEEE Trans.Comm.*, Vol. COM-13, 1965, pp. 301-307.

[G1] E.N. Gilbert, "Random graphs", *Ann. Math. Statist.*, Vol. 30, 1959, pp. 1141-1144.

[G2] R.E. Gomory and T.C. Hu, "Multiterminal network flows", *SIAM J. Applied Math*, Vol. 9, 1961, pp. 551-570.

[H1] E. Hansler, G.K. McAuliffe, R.S. Wilkov, "Exact calculation of computer network reliability", *Networks*, Vol. 4, 1974, pp. 95-112.

[H2] F. Harary, *Graph Theory*, Addison-Wesley, Reading MA., 1969.

[H3] D.D. Harms and C.J. Colbourn, "On bounds for the reliability of a network", Tech. Report 83-13, University of Saskatchewan, Saskatoon, 1983.

[H4] C.L. Hwang, F.A. Tillman and M.H. Lee, "System-reliability evaluation techniques for complex/large systems", *IEEE Trans. Reliability*, Vol. R-30, 1981, pp. 416-422.

[J1] I.M. Jacobs, "Connectivity in probabilistic graphs", Tech. Report #356, MIT, Electronics Research Laboratory, Cambridge, Mass., 1959.

[J2] M. Jerrum, On the Complexity of Evaluating Multivariate Polynomials, Ph.D. thesis, University of Edinburgh, 1981.

[K1] M. Karonski, "A view of random graphs", *Journal of Graph Theory*, Vol. 6, 1982, pp. 349-389.

[K2] G. Katona, "A theorem of finite sets", Theory of Graphs (Proceedings of Tihany Collequium, Sept. 1966),eds. P.Erdos and G.Katona, 1966, pp. 209-214.

[K3] A.K. Kel'mans, "Some problems of network reliability analysis", *Automation and Remote Control*, Vol. 26, 1965, pp. 564-573.

[K4] A.K. Kel'mans, "Connectivity of probabilistic networks", *Automation and Remote Control*, No. 3, 1967, pp. 98-116.

[K5] A.K. Kel'mans, "Bounds on the probability characteristics of random graphs", *Automation and Remote Control*, Vol. 32, 1965, pp. 1833-1839.

[K6] G. Kirchoff, "Uber die auflosung der gleichungen auf welche man sei der untersuchung der linearen verteilung galvanischer strome gefuhrt wind" *Poggendorg's Ann.Phys.Chem.*, Vol. 72, 1847, pp. 497-508 [English translation, *IRE Trans. Circuit Theory*, Vol. CT-5, 1958, pp. 4-8].

[K7] J.B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem", *Proceedings of the American Math. Soc.*, Vol. 7, 1956, pp. 48-50.

[K8] J.B. Kruskal, "The number of simplices in a complex", *Mathematical Optimization Techniques*, ed. R.Bellman, U. of California Press, Berkeley, 1963, pp. 251-278.

[L1] C.Y.Lee, "Analysis of switching networks," *Bell Syst.Tech.J.*, Vol. 34, 1955, pp. 1287-1315.

[L2] J.D. Leggett, Synthesis of Reliable Networks, Ph.D. thesis, Univ. Pennsylvania, Philadelphia, 1968.

[L3] J.D. Leggett and S.D. Bedrosian, "Synthesis of reliable networks", *IEEE Trans. Circuit Theory*, Vol. CT-16, 1969, pp. 384-385.

[L4] M.V. Lomonosov and V.P. Polesskii, "An upper bound for the reliability of information networks", *Problems of Information Transmission*, Vol. 7, 1971, pp. 337-339.

[L5] M.V. Lomonosov and V.P. Polesskii, "Lower bound of network reliability", *Problems of Information Transmission*, Vol. 8, 1972, pp. 118-123.

[M1] K. Menger, "Zur allgemeinen kurventheorie", *Fund. Math.*, Vol. 10, 1927, pp. 96-115.

[M2] H. Mine, "Reliability of physical systems", *IRE Trans. Circuit Theory*, Vol. CT-6, 1959, pp. 138-151.

[M3] E.F. Moore and C.E. Shannon, "Reliable circuits using less reliable relays", *J. Frankl. Inst.*, Vol. 262, 1956, pp 191-208, 281-297.

[M4] F. Moskowitz, "The analysis of redundancy networks", *AIEE Trans. (Commun. Electron.)*, Vol. 39, 1968, pp. 627-632.

[N1] C.St.J.A. Nash-Williams, "Edge-disjoint spanning trees of finite graphs", *J. London Math. Soc.*, Vol. 36, 1961, pp. 445-450.

[N2] E. Neufeld, Construction of Reliable Series-Parallel Networks: A Combinatorial Approach, M.Sc. thesis, Dept. of Computational Science, Univ. of Saskatchewan, 1983.

[N3] E.M. Neufeld and C.J. Colbourn, "The most reliable series-parallel networks", *Networks*, to appear Approach, M.Sc. thesis, Dept. of Computational Science, Univ. of Saskatchewan, 1983.

[P1] V.P. Polesskii, "A method for the construction of a structurally reliable communication network", *Discrete Automata and Communication Networks* [in Russian], Nauka, Moscow, 1970, pp. 13-19.

[P2] V.P. Polesskii, "A lower boundary for the reliability of information networks", *Problems of Information Transmission*, Vol. 7, 1971, pp. 165-171.

[P3] 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*, to appear.

[P4] J.S. Provan and M.O. Ball, "Computing network reliability in time polynomial in the number of cuts", Working Paper MS/S 81-002, College of Business and Management, University of Maryland, College Park, 1981.

[P5] J.S. Provan and L.J. Billera, "Decompositions of simplicial complexes related to diameters of convex polyhedra", *Math. Oper. Res.*, Vol. 5, 1980, pp. 579-594.

[P6] J.S. Provan, "Polyhedral combinatorics and network reliability", Tech. Report No. UNC/ORSA/TR-82/9, University of North Carolina, Chapel Hill, 1983.

[P7] J.S. Provan, private communication, 1983.

[R1] A. Rosenthal, "A computer scientist looks at reliability computations", in *Reliability and Fault Tree Analysis*, R.E. Barlow, J. Fussel and N. Singpurwalla, Eds., Philadelphia, 1975, pp. 133-152.

[R2] A. Rosenthal, "Computing the reliability of complex networks", *SIAM J. Appl. Math.*, Vol. 32, 1977, pp. 384-393.

[R3] A. Rosenthal and D. Frisque, "Transformations for simplifying network reliability calculations", *Networks*, Vol. 7, 1977, pp. 97-111.

[S1] A. Satyanarayana and A. Prabhakar, "New topological formula and rapid algorithm for reliability analysis of complex networks." *IEEE Trans. Reliability*, Vol. R-27, 1978, pp. 82-100.

[S2] A. Satyanarayana and M.K. Chang, "Network reliability and the factoring theorem", *Networks*, Vol. 13,1983, pp. 107-120.

[S3] R.P. Stanley, "Cohen-Macauley complexes", in *Higher Combinatorics*, M. Aigner, Ed. Reidel, Dordrecht, 1977, pp. 51-62.

[S4] V.Strassen, "Gaussian elimination is not optimal", *Numerische Mathematik*, Vol. 13,1969, pp. 354-356.

[T1] M. Tainiter, "A new deterministic network reliability measure", *Networks*, Vol. 6, 1976, pp. 191-204.

[T2] R.E. Tarjan, "Depth-first search and linear graph algorithms", *SIAM J. Computing*, Vol. 1, 1972, pp. 146-160.

[T3] W.T. Tutte, *Connectivity in Graphs*, London: Oxford Univ. Press, 1966.

[T4] W.T. Tutte, "The dissection of equilateral triangles into equilateral triangles", *Proc. Cambridge Phil. Soc.*, Vol. 44,1948, pp. 203-217.

[T5] W.T. Tutte, "On the problem of decomposing a graph into n connected factors", *J. London Math. Soc.*, Vol. 36, 1961, pp. 221-230.

[V1] L.G. Valiant, "The complexity of enumeration and reliability problems", *SIAM J. Computing*, Vol. 8, 1979, pp. 410-421.

[V2] R. Van Slyke and H. Frank, "Network reliability analysis: part I", *Networks*, Vol. 1, 1972, 279-290.

[V3] R. Van Slyke and H. Frank, "Network reliability analysis: part II", in *Reliability and Fault Tree Analysis*, R.E. Barlow, J. Fussel and N. Singpurwalla, Eds., Philadelphia, 1975, pp. 619-650.

[W1] J.A. Wald and C.J. Colbourn, "Steiner trees in probabilistic networks", *Microelectronics and Reliability*, to appear.

[W2] J.A. Wald and C.J. Colbourn, "Steiner trees, partial 2-trees, and minimum IFI networks", *Networks*, Vol. 13, 1983, pp. 159-167.

[W3] R.S. Wilkov, "Analysis and design of reliable computer networks", *IEEE Trans. Comm.*, Vol. COM-20, 1972, 660-678.

[Z1] E. Zemel, "Polynomial algorithms for estimating network reliability", *Networks*, Vol. 12, 1982, 439-452.

Appendices

Table A1.1 7-node Ladder							
n=7 b=11 c=2 C_c=2 t=144							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.029911	0.002490	0.034607	0.002205	0.004696	0.032402
K-K		0.028426	0.003976	0.035500	0.003098	0.007074	
Jcb-III		0.018202	0.014200	0.039243	0.006841	0.021041	
BBST-II		0.024760	0.007642	0.038478	0.006076	0.013718	
L-P		0.003791	0.028611	0.048462	0.016060	0.044671	
B-P	0.5	0.246094	0.027344	0.295898	0.022460	0.049805	0.273438
K-K		0.225586	0.047852	0.304199	0.030761	0.078613	
Jcb-III		0.102051	0.171387	0.343750	0.070312	0.241699	
BBST-II		0.177386	0.096052	0.334961	0.061523	0.157575	
L-P		0.062500	0.210938	0.329727	0.056289	0.267227	
B-P	0.7	0.655498	0.050697	0.743106	0.036911	0.087609	0.706195
K-K		0.606135	0.100060	0.754447	0.048252	0.148312	
Jcb-III		0.346645	0.359550	0.823600	0.117405	0.476955	
BBST-II		0.495306	0.210889	0.806256	0.100061	0.310950	
L-P		0.329417	0.376778	0.742223*	0.036028	0.412805	
B-P	0.9	0.965777	0.008418	0.979074	0.004879	0.013297	0.974195
K-K		0.954585	0.019610	0.979887	0.005692	0.025302	
Jcb-III		0.903455	0.070740	0.990266	0.016071	0.086811	
BBST-II		0.930513	0.043682	0.987109	0.012914	0.056596	
L-P		0.850306	0.123889	0.976185*	0.001990	0.125880	
B-P	0.92	0.979313	0.004878	0.986922	0.002731	0.007609	0.984191
K-K		0.972595	0.011596	0.987310	0.003119	0.014715	
Jcb-III		0.942343	0.041848	0.993240	0.009049	0.050897	
BBST-II		0.958207	0.025984	0.991389	0.007198	0.033182	
L-P		0.897405	0.086786	0.985221*	0.001030	0.087815	
B-P	0.94	0.989199	0.002320	0.992768	0.001249	0.003569	0.991519
K-K		0.985888	0.005631	0.992911	0.001392	0.007023	
Jcb-III		0.971189	0.020330	0.995687	0.004168	0.024498	
BBST-II		0.978825	0.012694	0.994796	0.003277	0.015972	
L-P		0.938223	0.053296	0.991955*	0.000436	0.053732	
B-P	0.96	0.995647	0.000771	0.996816	0.000398	0.001170	0.996418
K-K		0.994504	0.001914	0.996849	0.000431	0.002345	
Jcb-III		0.989504	0.006914	0.997757	0.001339	0.008253	
BBST-II		0.992076	0.004342	0.997457	0.001039	0.005380	
L-P		0.970620	0.025798	0.996547*	0.000129	0.025928	
B-P	0.98	0.999044	0.000108	0.999205	0.000053	0.000161	0.999152
K-K		0.998878	0.000274	0.999207	0.000055	0.000329	
Jcb-III		0.998163	0.000989	0.999332	0.000180	0.001169	
BBST-II		0.998528	0.000624	0.999290	0.000138	0.000762	
L-P		0.992143	0.007009	0.999168*	0.000016	0.007025	
B-P	0.99	0.999780	0.000014	0.999801	0.000007	0.000021	0.999794
K-K		0.999757	0.000037	0.999801	0.000007	0.000044	
Jcb-III		0.999662	0.000132	0.999817	0.000023	0.000155	
BBST-II		0.999710	0.000084	0.999812	0.000018	0.000101	
L-P		0.997969	0.001825	0.999796*	0.000002	0.001827	

Table A1.2 10-node Ladder							
n=10 b=17 c=2 C_c=2 t=2584							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.005502	0.002149	0.009068	0.001418	0.003567	0.007651
K-K		0.005296	0.002355	0.012617	0.004966	0.007321	
Jcb-III		0.002933	0.004718	0.015625	0.007974	0.012692	
BBST-II		0.004282	0.003368	0.015438	0.007788	0.011156	
L-P		0.000144	0.007507	0.013744	0.006093	0.013600	
B-P	0.5	0.105248	0.064796	0.209076	0.039032	0.103828	0.170044
K-K		0.092529	0.077515	0.270027	0.099983	0.177498	
Jcb-III		0.020874	0.149170	0.334229	0.164185	0.313355	
BBST-II		0.054182	0.115862	0.329620	0.159576	0.275439	
L-P		0.010742	0.159302	0.220891	0.050847	0.210149	
B-P	0.7	0.450040	0.190847	0.728754	0.087867	0.278714	0.640887
K-K		0.355730	0.285157	0.788083	0.147196	0.432353	
Jcb-III		0.083372	0.557515	0.901347	0.260460	0.817975	
BBST-II		0.170318	0.470569	0.889318	0.248431	0.719000	
L-P		0.149308	0.491579	0.683711*	0.042824	0.534403	
B-P	0.9	0.930207	0.040811	0.980324	0.009306	0.050117	0.971018
K-K		0.872668	0.098350	0.980940	0.009922	0.108272	
Jcb-III		0.757689	0.213329	0.995787	0.024769	0.238097	
BBST-II		0.782998	0.188020	0.992285	0.021267	0.209287	
L-P		0.736099	0.234919	0.973260*	0.002242	0.237161	
B-P	0.92	0.958374	0.024197	0.987511	0.004940	0.029138	0.982571
K-K		0.920429	0.062142	0.987700	0.005129	0.067272	
Jcb-III		0.846068	0.136503	0.996316	0.013745	0.150248	
BBST-II		0.862039	0.120532	0.994107	0.011536	0.132068	
L-P		0.812118	0.170453	0.983708*	0.001137	0.171590	
B-P	0.94	0.979065	0.011775	0.992976	0.002136	0.013911	0.990840
K-K		0.958531	0.032309	0.993015	0.002175	0.034483	
Jcb-III		0.918975	0.071865	0.997152	0.006312	0.078177	
BBST-II		0.927284	0.063556	0.996002	0.005162	0.068718	
L-P		0.882412	0.108428	0.991313*	0.000473	0.108901	
B-P	0.96	0.992209	0.004010	0.996860	0.000641	0.004651	0.996219
K-K		0.984436	0.011783	0.996864	0.000645	0.012428	
Jcb-III		0.969684	0.026535	0.998265	0.002046	0.028581	
BBST-II		0.972722	0.023497	0.997845	0.001626	0.025123	
L-P		0.941846	0.054373	0.996356*	0.000137	0.054510	
B-P	0.98	0.998553	0.000575	0.999208	0.000080	0.000655	0.999128
K-K		0.997317	0.001811	0.999208	0.000080	0.001891	
Jcb-III		0.995000	0.004128	0.999409	0.000281	0.004409	
BBST-II		0.995469	0.003659	0.999344	0.000216	0.003875	
L-P		0.983822	0.015306	0.999144*	0.000016	0.015322	
B-P	0.99	0.999714	0.000077	0.999801	0.000010	0.000087	0.999791
K-K		0.999540	0.000251	0.999801	0.000010	0.000261	
Jcb-III		0.999216	0.000575	0.999828	0.000037	0.000612	
BBST-II		0.999281	0.000510	0.999819	0.000028	0.000538	
L-P		0.995734	0.004057	0.999793*	0.000002	0.004059	

Table A1.3 15-node Ladder							
n=15 b=27 c=2 C_c=2 t=317811							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000312	0.000378	0.001032	0.000342	0.000721	0.000690
K-K		0.000310	0.000380	0.003703	0.003013	0.003393	
Jcb-III		0.000147	0.000543	0.005108	0.004418	0.004961	
BBST-II		0.000226	0.000464	0.005080	0.004390	0.004854	
L-P		0.000001	0.000690	0.001682	0.000992	0.001682	
B-P	0.5	0.021743	0.055297	0.125904	0.048864	0.104162	0.077040
K-K		0.020692	0.056348	0.266049	0.189009	0.245357	
Jcb-III		0.002371	0.074669	0.352922	0.275882	0.350551	
BBST-II		0.007925	0.069115	0.350924	0.273884	0.343000	
L-P		0.000488	0.076551	0.113297*	0.036257	0.112809	
B-P	0.7	0.198837	0.346333	0.725430	0.180260	0.526593	0.545170
K-K		0.154219	0.390951	0.824877	0.279707	0.670658	
Jcb-III		0.005382	0.539788	0.964374	0.419204	0.958992	
BBST-II		0.020577	0.524593	0.958909	0.413739	0.938333	
L-P		0.035268	0.509902	0.596262*	0.051092	0.560994	
B-P	0.9	0.833941	0.131805	0.980907	0.015161	0.146966	0.965746
K-K		0.702229	0.263517	0.981000	0.015254	0.278770	
Jcb-III		0.483145	0.482601	0.998564	0.032818	0.515418	
BBST-II		0.491312	0.474434	0.995627	0.029881	0.504315	
L-P		0.549043	0.416703	0.968403*	0.002657	0.419360	
B-P	0.92	0.896847	0.083028	0.987696	0.007821	0.090849	0.979875
K-K		0.795269	0.184606	0.987712	0.007837	0.192443	
Jcb-III		0.630179	0.349696	0.998408	0.018533	0.368229	
BBST-II		0.636013	0.343862	0.996310	0.016435	0.360297	
L-P		0.659729	0.320146	0.981192*	0.001317	0.321463	
B-P	0.94	0.946738	0.042973	0.993014	0.003303	0.046276	0.989711
K-K		0.882461	0.107250	0.993016	0.003305	0.110555	
Jcb-III		0.779847	0.209864	0.998467	0.008756	0.218620	
BBST-II		0.783311	0.206400	0.997221	0.007510	0.213910	
L-P		0.773763	0.215948	0.990243*	0.000532	0.216480	
B-P	0.96	0.980308	0.015580	0.996864	0.000976	0.016556	0.995888
K-K		0.951859	0.044029	0.996864	0.000976	0.045005	
Jcb-III		0.907047	0.088841	0.998847	0.002959	0.091800	
BBST-II		0.908501	0.087387	0.998324	0.002436	0.089823	
L-P		0.880890	0.114998	0.996037*	0.000149	0.115147	
B-P	0.98	0.996709	0.002378	0.999208	0.000121	0.002499	0.999087
K-K		0.991419	0.007668	0.999208	0.000121	0.007789	
Jcb-III		0.983165	0.015922	0.999517	0.000430	0.016352	
BBST-II		0.983424	0.015663	0.999424	0.000337	0.016000	
L-P		0.964662	0.034425	0.999104*	0.000017	0.034442	
B-P	0.99	0.999458	0.000328	0.999801	0.000015	0.000343	0.999786
K-K		0.998652	0.001134	0.999801	0.000015	0.001149	
Jcb-III		0.997400	0.002386	0.999844	0.000058	0.002444	
BBST-II		0.997439	0.002347	0.999831	0.000045	0.002391	
L-P		0.990370	0.009416	0.999788*	0.000002	0.009418	

Table A1.4 20-node Ladder							
n=20 b=37 c=2 C_c=2 t=3.90882e+07							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000017	0.000045	0.000120	0.000058	0.000103	0.000062
K-K		0.000017	0.000045	0.001346	0.001284	0.001329	
Jcb-III		0.000007	0.000055	0.001936	0.001874	0.001929	
BBST-II		0.000012	0.000051	0.001930	0.001868	0.001919	
L-P		0.000000	0.000062	0.000206	0.000144	0.000206	
B-P	0.5	0.004365	0.030538	0.078045	0.043142	0.073680	0.034903
K-K		0.004305	0.030598	0.271280	0.236377	0.266975	
Jcb-III		0.000284	0.034619	0.371699	0.336796	0.371415	
BBST-II		0.001106	0.033797	0.370584	0.335681	0.369478	
L-P		0.000020	0.034883	0.058111*	0.023208	0.058091	
B-P	0.7	0.082450	0.381299	0.725452	0.261703	0.643002	0.463749
K-K		0.068650	0.395099	0.838856	0.375107	0.770205	
Jcb-III		0.000275	0.463474	0.986939	0.523190	0.986664	
BBST-II		0.002457	0.461292	0.983976	0.520227	0.981518	
L-P		0.007637	0.456112	0.519997*	0.056248	0.512360	
B-P	0.9	0.714673	0.245830	0.980983	0.020480	0.266310	0.960503
K-K		0.542379	0.418124	0.981000	0.020497	0.438621	
Jcb-III		0.269840	0.690663	0.999499	0.038996	0.729659	
BBST-II		0.271454	0.689049	0.997308	0.036805	0.725854	
L-P		0.391747	0.568756	0.963571*	0.003068	0.571824	
B-P	0.92	0.812622	0.164566	0.987710	0.010522	0.175088	0.977188
K-K		0.658415	0.318773	0.987712	0.010524	0.329297	
Jcb-III		0.422412	0.554776	0.999309	0.022121	0.576896	
BBST-II		0.423688	0.553500	0.997576	0.020388	0.573888	
L-P		0.516856	0.460332	0.978683*	0.001495	0.461827	
B-P	0.94	0.897883	0.090700	0.993016	0.004433	0.095132	0.988583
K-K		0.784436	0.204147	0.993016	0.004433	0.208580	
Jcb-III		0.614759	0.373824	0.999174	0.010591	0.384416	
BBST-II		0.615609	0.372974	0.998020	0.009437	0.382411	
L-P		0.660455	0.328128	0.989174*	0.000591	0.328719	
B-P	0.96	0.960471	0.035087	0.996864	0.001306	0.036393	0.995558
K-K		0.901993	0.093565	0.996864	0.001306	0.094871	
Jcb-III		0.815802	0.179756	0.999233	0.003675	0.183431	
BBST-II		0.816208	0.179350	0.998682	0.003124	0.182474	
L-P		0.810338	0.185220	0.995719*	0.000161	0.185381	
B-P	0.98	0.993323	0.005723	0.999208	0.000162	0.005885	0.999046
K-K		0.980637	0.018409	0.999208	0.000162	0.018571	
Jcb-III		0.962087	0.036959	0.999606	0.000560	0.037518	
BBST-II		0.962170	0.036876	0.999493	0.000447	0.037323	
L-P		0.940101	0.058945	0.999064*	0.000018	0.058963	
B-P	0.99	0.998964	0.000817	0.999801	0.000020	0.000837	0.999781
K-K		0.996877	0.002904	0.999801	0.000020	0.002924	
Jcb-III		0.993831	0.005950	0.999859	0.000078	0.006028	
BBST-II		0.993844	0.005937	0.999841	0.000060	0.005997	
L-P		0.983141	0.016640	0.999783*	0.000002	0.016642	

Table A1.5 25-node Ladder							
n=25 b=47 c=2 C_c=2 t=4.80753e+09							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000001	0.000005	0.000020	0.000014	0.000019	0.000006
K-K		0.000001	0.000005	0.000514	0.000508	0.000513	
Jcb-III		0.000000	0.000005	0.000754	0.000749	0.000754	
BBST-II		0.000001	0.000005	0.000753	0.000747	0.000752	
L-P		0.000000	0.000006	0.000025	0.000020	0.000025	
B-P	0.5	0.000840	0.014973	0.082391	0.066578	0.081551	0.015813
K-K		0.000837	0.014976	0.275808	0.259995	0.274971	
Jcb-III		0.000034	0.015779	0.385468	0.369655	0.385434	
BBST-II		0.000149	0.015664	0.384755	0.368942	0.384606	
L-P		0.000001	0.015813	0.029806*	0.013992	0.029805	
B-P	0.7	0.031695	0.362793	1.033670	0.639182	1.001970	0.394488
K-K		0.027796	0.366692	0.843939	0.449451	0.816143	
Jcb-III		0.000012	0.394476	0.995092	0.600604	0.995080	
BBST-II		0.000309	0.394179	0.993251	0.598763	0.992942	
L-P		0.001571	0.392917	0.453488*	0.059000	0.451917	
B-P	0.9	0.582322	0.372966	0.981282	0.025994	0.398960	0.955288
K-K		0.392497	0.562791	0.981000	0.025712	0.588503	
Jcb-III		0.138164	0.817124	0.999825	0.044537	0.861662	
BBST-II		0.138421	0.816867	0.998231	0.042943	0.859811	
L-P		0.271206	0.684082	0.958763*	0.003475	0.687557	
B-P	0.92	0.709467	0.265040	0.987745	0.013238	0.278278	0.974507
K-K		0.513636	0.460871	0.987712	0.013205	0.474076	
Jcb-III		0.263090	0.711417	0.999700	0.025193	0.736609	
BBST-II		0.263310	0.711197	0.998337	0.023830	0.735027	
L-P		0.394722	0.579785	0.976180*	0.001673	0.581459	
B-P	0.94	0.831873	0.155583	0.993018	0.005562	0.161144	0.987456
K-K		0.665257	0.322199	0.993016	0.005560	0.327759	
Jcb-III		0.458234	0.529222	0.999555	0.012099	0.541321	
BBST-II		0.458395	0.529061	0.998554	0.011098	0.540158	
L-P		0.552661	0.434795	0.988106*	0.000650	0.435445	
B-P	0.96	0.930922	0.064305	0.996864	0.001637	0.065942	0.995227
K-K		0.831251	0.163976	0.996864	0.001637	0.165613	
Jcb-III		0.709315	0.285912	0.999490	0.004263	0.290176	
BBST-II		0.709401	0.285826	0.998953	0.003726	0.289552	
L-P		0.735810	0.259417	0.995400*	0.000173	0.259590	
B-P	0.98	0.987767	0.011238	0.999208	0.000203	0.011441	0.999005
K-K		0.962589	0.036416	0.999208	0.000203	0.036619	
Jcb-III		0.931937	0.067068	0.999678	0.000673	0.067740	
BBST-II		0.931957	0.067048	0.999552	0.000547	0.067595	
L-P		0.911355	0.087650	0.999024*	0.000019	0.087669	
B-P	0.99	0.998113	0.001663	0.999801	0.000025	0.001688	0.999776
K-K		0.993638	0.006138	0.999801	0.000025	0.006163	
Jcb-III		0.988187	0.011589	0.999873	0.000097	0.011686	
BBST-II		0.988190	0.011586	0.999851	0.000075	0.011661	
L-P		0.974241	0.025535	0.999778*	0.000002	0.025537	

Table A2.1 7-node Diamond							
n=7 b=11 c=2 C_c=5 t=112							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.024356	0.000998	0.027785	0.002431	0.003428	0.025354
K-K		0.022526	0.002828	0.028730	0.003376	0.006204	
Jcb-III		0.014252	0.011102	0.035294	0.009939	0.021041	
BBST-II		0.019353	0.006001	0.033381	0.008027	0.014028	
L-P		0.003791	0.021563	0.030443	0.005089	0.026653	
B-P	0.5	0.210449	0.011231	0.247070	0.025390	0.036621	0.221680
K-K		0.185059	0.036621	0.255859	0.034179	0.070801	
Jcb-III		0.084961	0.136719	0.326660	0.104980	0.241699	
BBST-II		0.143555	0.078125	0.304688	0.083008	0.161133	
L-P		0.062500	0.159180	0.233597*	0.011917	0.171097	
B-P	0.7	0.598543	0.021568	0.663693	0.043582	0.065150	0.620111
K-K		0.537172	0.082939	0.675701	0.055590	0.138528	
Jcb-III		0.326601	0.293510	0.803556	0.183445	0.476955	
BBST-II		0.442226	0.177885	0.760196	0.140085	0.317970	
L-P		0.329417	0.290694	0.623577*	0.003466	0.294160	
B-P	0.9	0.947193	0.003778	0.957285	0.006314	0.010092	0.950971
K-K		0.933226	0.017745	0.958146	0.007175	0.024919	
Jcb-III		0.891662	0.059309	0.978473	0.027502	0.086811	
BBST-II		0.912707	0.038264	0.970581	0.019610	0.057874	
L-P		0.850306	0.100665	0.950989*	0.000018	0.100683	
B-P	0.92	0.966197	0.002204	0.971989	0.003588	0.005792	0.968401
K-K		0.957811	0.010590	0.972400	0.003999	0.014589	
Jcb-III		0.933213	0.035188	0.984110	0.015709	0.050897	
BBST-II		0.945552	0.022849	0.979483	0.011082	0.033931	
L-P		0.897405	0.070996	0.968407*	0.000006	0.071001	
B-P	0.94	0.981072	0.001056	0.983798	0.001670	0.002726	0.982128
K-K		0.976937	0.005191	0.983949	0.001821	0.007012	
Jcb-III		0.964983	0.017145	0.989482	0.007354	0.024498	
BBST-II		0.970922	0.011206	0.987255	0.005127	0.016332	
L-P		0.938223	0.043905	0.982129*	0.000001	0.043906	
B-P	0.96	0.991672	0.000353	0.992568	0.000543	0.000896	0.992025
K-K		0.990244	0.001781	0.992603	0.000578	0.002358	
Jcb-III		0.986177	0.005848	0.994430	0.002405	0.008253	
BBST-II		0.988178	0.003847	0.993680	0.001655	0.005502	
L-P		0.970620	0.021405	0.992026*	0.000001	0.021406	
B-P	0.98	0.997952	0.000050	0.998075	0.000073	0.000124	0.998002
K-K		0.997744	0.000258	0.998078	0.000076	0.000333	
Jcb-III		0.997163	0.000839	0.998331	0.000329	0.001169	
BBST-II		0.997446	0.000556	0.998225	0.000223	0.000779	
L-P		0.992143	0.005859	0.998002*	0.000000	0.005858	
B-P	0.99	0.999493	0.000007	0.999510	0.000010	0.000016	0.999500
K-K		0.999466	0.000034	0.999510	0.000010	0.000044	
Jcb-III		0.999388	0.000112	0.999543	0.000043	0.000155	
BBST-II		0.999426	0.000074	0.999529	0.000029	0.000104	
L-P		0.997969	0.001531	0.999500*	0.000000	0.001531	

Table A2.2 10-node Diamond							
n=10 b=17 c=2 C_c=8 t=1280							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.003025	0.000874	0.005465	0.001566	0.002441	0.003899
K-K		0.002791	0.001108	0.009189	0.005290	0.006398	
Jcb-III		0.001453	0.002446	0.014145	0.010246	0.012692	
BBST-II		0.002122	0.001777	0.013399	0.009499	0.011277	
L-P		0.000144	0.003755	0.004392*	0.000493	0.004248	
B-P	0.5	0.068474	0.029686	0.146423	0.048263	0.077950	0.098160
K-K		0.054474	0.043686	0.212555	0.114395	0.158081	
Jcb-III		0.010879	0.087280	0.324234	0.226074	0.313355	
BBST-II		0.027379	0.070781	0.305801	0.207641	0.278423	
L-P		0.010742	0.087418	0.099917*	0.001758	0.089175	
B-P	0.7	0.367293	0.102669	0.600297	0.130335	0.233004	0.469962
K-K		0.264144	0.205818	0.668615	0.198653	0.404471	
Jcb-III		0.077356	0.392606	0.895331	0.425369	0.817975	
BBST-II		0.120425	0.349537	0.847214	0.377252	0.726789	
L-P		0.149308	0.320654	0.470243*	0.000281	0.320935	
B-P	0.9	0.896300	0.026445	0.942172	0.019427	0.045872	0.922745
K-K		0.832909	0.089836	0.942996	0.020251	0.110088	
Jcb-III		0.745331	0.177414	0.983428	0.060683	0.238097	
BBST-II		0.757868	0.164877	0.969422	0.046677	0.211555	
L-P		0.736099	0.186646	0.922745*	0.000000	0.186646	
B-P	0.92	0.933965	0.015967	0.960787	0.010855	0.026822	0.949932
K-K		0.892120	0.057812	0.961048	0.011116	0.068928	
Jcb-III		0.835073	0.114859	0.985322	0.035390	0.150248	
BBST-II		0.842984	0.106948	0.976483	0.026551	0.133499	
L-P		0.812118	0.137814	0.949932*	0.000000	0.137815	
B-P	0.94	0.963649	0.007911	0.976517	0.004957	0.012868	0.971560
K-K		0.940982	0.030578	0.976573	0.005013	0.035591	
Jcb-III		0.910436	0.061124	0.988613	0.017053	0.078177	
BBST-II		0.914553	0.057007	0.984015	0.012455	0.069462	
L-P		0.882412	0.089148	0.971560*	0.000000	0.089148	
B-P	0.96	0.984529	0.002742	0.988849	0.001578	0.004320	0.987271
K-K		0.975940	0.011331	0.988856	0.001585	0.012915	
Jcb-III		0.964480	0.022791	0.993061	0.005790	0.028581	
BBST-II		0.965985	0.021286	0.991380	0.004109	0.025395	
L-P		0.941846	0.045425	0.987271*	0.000000	0.045425	
B-P	0.98	0.996405	0.000399	0.997015	0.000211	0.000610	0.996804
K-K		0.995037	0.001767	0.997015	0.000211	0.001978	
Jcb-III		0.993228	0.003576	0.997637	0.000833	0.004409	
BBST-II		0.993460	0.003344	0.997377	0.000573	0.003917	
L-P		0.983822	0.012982	0.996804*	0.000000	0.012982	
B-P	0.99	0.999146	0.000054	0.999227	0.000027	0.000081	0.999200
K-K		0.998954	0.000246	0.999227	0.000027	0.000274	
Jcb-III		0.998700	0.000500	0.999312	0.000112	0.000612	
BBST-II		0.998732	0.000468	0.999276	0.000076	0.000544	
L-P		0.995734	0.003466	0.999200*	0.000000	0.003466	

Table A2.3 15-node Diamond							
n=15 b=27 c=2 C_c=13 t=61440							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000074	0.000075	0.000320	0.000171	0.000246	0.000149
K-K		0.000073	0.000076	0.002860	0.002711	0.002787	
Jcb-III		0.000028	0.000121	0.004990	0.004841	0.004961	
BBST-II		0.000044	0.000105	0.004806	0.004657	0.004762	
L-P		0.000001	0.000149	0.000157*	0.000008	0.000156	
B-P	0.5	0.008009	0.015688	0.058184	0.034488	0.050175	0.023696
K-K		0.006823	0.016873	0.211773	0.188077	0.204951	
Jcb-III		0.000460	0.023236	0.351012	0.327316	0.350551	
BBST-II		0.001534	0.022162	0.338028	0.314332	0.336494	
L-P		0.000488	0.023208	0.023756*	0.000060	0.023268	
B-P	0.7	0.134717	0.158732	0.519635	0.226186	0.384918	0.293449
K-K		0.083388	0.210061	0.685121	0.391672	0.601733	
Jcb-III		0.004972	0.288477	0.963964	0.670515	0.958992	
BBST-II		0.007910	0.285539	0.928446	0.634997	0.920536	
L-P		0.035268	0.258181	0.293453*	0.000004	0.258185	
B-P	0.9	0.784790	0.092731	0.924677	0.047156	0.139887	0.877521
K-K		0.629627	0.247894	0.925418	0.047897	0.295791	
Jcb-III		0.475249	0.402272	0.990667	0.113146	0.515418	
BBST-II		0.476827	0.400694	0.971577	0.094056	0.494750	
L-P		0.549043	0.328478	0.877521*	0.000000	0.328478	
B-P	0.92	0.859553	0.060368	0.946875	0.026954	0.087322	0.919921
K-K		0.739615	0.180306	0.947060	0.027139	0.207445	
Jcb-III		0.621424	0.298497	0.989653	0.069732	0.368229	
BBST-II		0.622552	0.297369	0.976015	0.056094	0.353463	
L-P		0.659729	0.260192	0.919921*	0.000000	0.260192	
B-P	0.94	0.921977	0.032221	0.966813	0.012615	0.044836	0.954198
K-K		0.845912	0.108286	0.966842	0.012644	0.120930	
Jcb-III		0.771416	0.182782	0.990036	0.035838	0.218620	
BBST-II		0.772086	0.182112	0.981939	0.027741	0.209853	
L-P		0.773763	0.180435	0.954198*	0.000000	0.180435	
B-P	0.96	0.967377	0.012022	0.983526	0.004127	0.016150	0.979399
K-K		0.933637	0.045762	0.983528	0.004129	0.049891	
Jcb-III		0.900704	0.078695	0.992504	0.013105	0.091800	
BBST-II		0.900985	0.078414	0.989104	0.009705	0.088119	
L-P		0.880890	0.098509	0.979399*	0.000000	0.098508	
B-P	0.98	0.992929	0.001883	0.995380	0.000568	0.002452	0.994812
K-K		0.986641	0.008171	0.995380	0.000568	0.008740	
Jcb-III		0.980510	0.014302	0.996862	0.002050	0.016352	
BBST-II		0.980560	0.014252	0.996256	0.001444	0.015697	
L-P		0.964662	0.030150	0.994812*	0.000000	0.030151	
B-P	0.99	0.998438	0.000263	0.998775	0.000074	0.000338	0.998701
K-K		0.997479	0.001222	0.998775	0.000074	0.001296	
Jcb-III		0.996545	0.002156	0.998989	0.000288	0.002444	
BBST-II		0.996552	0.002149	0.998898	0.000197	0.002346	
L-P		0.990370	0.008331	0.998701*	0.000000	0.008331	

Table A2.4 20-node Diamond							
n=20 b=37 c=2 C_c=18 t=2.62144e+06							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000002	0.000004	0.000018	0.000012	0.000016	0.000005
K-K		0.000002	0.000004	0.001067	0.001062	0.001066	
Jcb-III		0.000000	0.000005	0.001929	0.001924	0.001929	
BBST-II		0.000001	0.000005	0.001877	0.001872	0.001876	
L-P		0.000000	0.000005	0.000005*	0.000000	0.000005	
B-P	0.5	0.000790	0.004846	0.022827	0.017191	0.022037	0.005636
K-K		0.000720	0.004916	0.217006	0.211370	0.216286	
Jcb-III		0.000019	0.005617	0.371434	0.365798	0.371415	
BBST-II		0.000074	0.005562	0.361396	0.355760	0.361321	
L-P		0.000020	0.005616	0.005638*	0.000002	0.005618	
B-P	0.7	0.042256	0.140868	0.463435	0.280311	0.421179	0.183124
K-K		0.025814	0.157310	0.693993	0.510869	0.668179	
Jcb-III		0.000253	0.182871	0.986917	0.803793	0.986664	
BBST-II		0.000400	0.182724	0.960250	0.777126	0.959851	
L-P		0.007637	0.175487	0.183124*	0.000000	0.175487	
B-P	0.9	0.653186	0.181328	0.914578	0.080064	0.261392	0.834514
K-K		0.442283	0.392231	0.915009	0.080495	0.472727	
Jcb-III		0.265835	0.568679	0.995494	0.160980	0.729659	
BBST-II		0.265943	0.568571	0.975774	0.141260	0.709830	
L-P		0.391747	0.442767	0.834514*	0.000000	0.442767	
B-P	0.92	0.763709	0.127149	0.937755	0.046897	0.174046	0.890858
K-K		0.574368	0.316490	0.937835	0.046977	0.363467	
Jcb-III		0.416880	0.473978	0.993777	0.102919	0.576896	
BBST-II		0.416966	0.473892	0.978185	0.087327	0.561219	
L-P		0.516856	0.374002	0.890858*	0.000000	0.374002	
B-P	0.94	0.864064	0.073081	0.959691	0.022546	0.095627	0.937145
K-K		0.724351	0.212794	0.959699	0.022554	0.235349	
Jcb-III		0.608153	0.328992	0.992569	0.055424	0.384416	
BBST-II		0.608210	0.328935	0.982179	0.045034	0.373969	
L-P		0.660455	0.276690	0.937145*	0.000000	0.276691	
B-P	0.96	0.942231	0.029357	0.979184	0.007596	0.036953	0.971588
K-K		0.870003	0.101585	0.979184	0.007596	0.109181	
Jcb-III		0.809668	0.161920	0.993099	0.021511	0.183431	
BBST-II		0.809696	0.161892	0.988142	0.016554	0.178446	
L-P		0.810338	0.161250	0.971588*	0.000000	0.161250	
B-P	0.98	0.987872	0.004952	0.993903	0.001079	0.006031	0.992824
K-K		0.972159	0.020665	0.993903	0.001079	0.021744	
Jcb-III		0.958932	0.033892	0.996450	0.003626	0.037518	
BBST-II		0.958937	0.033887	0.995436	0.002612	0.036499	
L-P		0.940101	0.052723	0.992824*	0.000000	0.052723	
B-P	0.99	0.997483	0.000719	0.998345	0.000143	0.000862	0.998202
K-K		0.994895	0.003307	0.998345	0.000143	0.003450	
Jcb-III		0.992705	0.005497	0.998734	0.000532	0.006028	
BBST-II		0.992706	0.005496	0.998571	0.000369	0.005865	
L-P		0.983141	0.015061	0.998202*	0.000000	0.015061	

Table A2.5 25-node Diamond							
n=25 b=47 c=2 C_c=23 t=1.04858e+08							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000000	0.000000	0.000001	0.000001	0.000001	0.000000
K-K		0.000000	0.000000	0.000409	0.000409	0.000409	
Jcb-III		0.000000	0.000000	0.000754	0.000754	0.000754	
BBST-II		0.000000	0.000000	0.000738	0.000738	0.000738	
L-P		0.000000	0.000000	0.000000*	0.000000	0.000000	
B-P	0.5	0.000077	0.001260	0.009128	0.007791	0.009051	0.001338
K-K		0.000074	0.001264	0.220750	0.219412	0.220676	
Jcb-III		0.000001	0.001337	0.385434	0.384096	0.385434	
BBST-II		0.000003	0.001335	0.377234	0.375896	0.377230	
L-P		0.000001	0.001337	0.001338*	0.000000	0.001337	
B-P	0.7	0.012587	0.101688	0.423790	0.309515	0.411204	0.114275
K-K		0.008172	0.106103	0.697642	0.583367	0.689470	
Jcb-III		0.000011	0.114264	0.995091	0.880816	0.995080	
BBST-II		0.000018	0.114257	0.973919	0.859644	0.973902	
L-P		0.001571	0.112704	0.114275*	0.000000	0.112704	
B-P	0.9	0.522935	0.270679	0.908659	0.115045	0.385723	0.793614
K-K		0.303090	0.490524	0.908863	0.115249	0.605773	
Jcb-III		0.136331	0.657283	0.997993	0.204379	0.861662	
BBST-II		0.136336	0.657278	0.979659	0.186045	0.843323	
L-P		0.271206	0.522408	0.793614*	0.000000	0.522408	
B-P	0.92	0.658946	0.203767	0.931727	0.069014	0.272781	0.862713
K-K		0.431467	0.431246	0.931755	0.069042	0.500288	
Jcb-III		0.259936	0.602777	0.996546	0.133833	0.736609	
BBST-II		0.259941	0.602772	0.980873	0.118160	0.720932	
L-P		0.394722	0.467991	0.862713*	0.000000	0.467992	
B-P	0.94	0.794586	0.125812	0.954456	0.034058	0.159870	0.920398
K-K		0.600465	0.319933	0.954457	0.034059	0.353992	
Jcb-III		0.453564	0.466834	0.994886	0.074488	0.541321	
BBST-II		0.453568	0.466830	0.983368	0.062970	0.529800	
L-P		0.552661	0.367737	0.920398*	0.000000	0.367737	
B-P	0.96	0.909511	0.054330	0.975642	0.011801	0.066131	0.963841
K-K		0.793047	0.170794	0.975642	0.011801	0.182595	
Jcb-III		0.703962	0.259879	0.994138	0.030297	0.290176	
BBST-II		0.703964	0.259877	0.987964	0.024123	0.284000	
L-P		0.735810	0.228031	0.963841*	0.000000	0.228030	
B-P	0.98	0.980981	0.009859	0.992567	0.001727	0.011586	0.990840
K-K		0.951478	0.039362	0.992567	0.001727	0.041089	
Jcb-III		0.928553	0.062287	0.996294	0.005454	0.067740	
BBST-II		0.928554	0.062286	0.994852	0.004012	0.066299	
L-P		0.911355	0.079485	0.990840*	0.000000	0.079485	
B-P	0.99	0.996220	0.001483	0.997936	0.000233	0.001717	0.997703
K-K		0.990969	0.006734	0.997936	0.000233	0.006968	
Jcb-III		0.986851	0.010852	0.998537	0.000834	0.011686	
BBST-II		0.986851	0.010852	0.998288	0.000585	0.011437	
L-P		0.974241	0.023462	0.997703*	0.000000	0.023462	

Table A3.1 5-node Complete Graph							
n=5 b=10 c=4 C_c=5 t=125							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.1	0.008038	0.000059	0.008139	0.000041	0.000100	0.008098
K-K		0.007903	0.000195	0.008157	0.000059	0.000254	
Jcb-III		0.006787	0.001311	0.008275	0.000177	0.001488	
BBST-II		0.007672	0.000425	0.008239	0.000142	0.000567	
L-P		0.005526	0.002572	0.013987	0.005890	0.008462	
B-P	0.2	0.080897	0.001049	0.082680	0.000734	0.001783	0.081946
K-K		0.078485	0.003460	0.082994	0.001049	0.004509	
Jcb-III		0.058667	0.023278	0.085091	0.003146	0.026424	
BBST-II		0.074396	0.007550	0.084462	0.002517	0.010066	
L-P		0.059794	0.022151	0.121503	0.039558	0.061708	
B-P	0.3	0.252176	0.004084	0.259119	0.002859	0.006943	0.256260
K-K		0.242783	0.013477	0.260345	0.004085	0.017562	
Jcb-III		0.165593	0.090667	0.268513	0.012253	0.102919	
BBST-II		0.226855	0.029405	0.266062	0.009802	0.039207	
L-P		0.200250	0.056010	0.333446	0.077186	0.133196	
B-P	0.4	0.481691	0.007963	0.495228	0.005574	0.013536	0.489654
K-K		0.463377	0.026277	0.497617	0.007963	0.034239	
Jcb-III		0.312884	0.176770	0.513542	0.023888	0.200658	
BBST-II		0.432323	0.057331	0.508764	0.019110	0.076441	
L-P		0.409364	0.080290	0.573952	0.084298	0.164588	
B-P	0.5	0.701172	0.009766	0.717773	0.006835	0.016602	0.710938
K-K		0.678711	0.032227	0.720703	0.009765	0.041992	
Jcb-III		0.494141	0.216797	0.740234	0.029296	0.246094	
BBST-II		0.640625	0.070313	0.734375	0.023437	0.093750	
L-P		0.632813	0.078125	0.772476	0.061538	0.139664	
B-P	0.6	0.862294	0.007963	0.875831	0.005574	0.013536	0.870257
K-K		0.843980	0.026277	0.878219	0.007962	0.034239	
Jcb-III		0.693487	0.176770	0.894145	0.023888	0.200658	
BBST-II		0.812926	0.057331	0.889367	0.019110	0.076441	
L-P		0.816509	0.053748	0.901465	0.031208	0.084956	
B-P	0.7	0.953429	0.004084	0.960372	0.002859	0.006943	0.957513
K-K		0.944036	0.013477	0.961597	0.004084	0.017562	
Jcb-III		0.866846	0.090667	0.969765	0.012252	0.102919	
BBST-II		0.928108	0.029405	0.967315	0.009802	0.039207	
L-P		0.932619	0.024894	0.967992	0.010479	0.035372	
B-P	0.8	0.990616	0.001049	0.992399	0.000734	0.001783	0.991665
K-K		0.988204	0.003461	0.992713	0.001048	0.004509	
Jcb-III		0.968386	0.023279	0.994810	0.003145	0.026424	
BBST-II		0.984115	0.007550	0.994181	0.002516	0.010066	
L-P		0.985242	0.006423	0.993615	0.001950	0.008373	
B-P	0.9	0.999433	0.000059	0.999534	0.000042	0.000100	0.999492
K-K		0.999297	0.000195	0.999551	0.000059	0.000254	
Jcb-III		0.998181	0.001311	0.999669	0.000177	0.001488	
BBST-II		0.999067	0.000425	0.999634	0.000142	0.000567	
L-P		0.999020	0.000472	0.999600	0.000108	0.000580	

Table A3.2 6-node Complete Graph							
n=6 b=15 c=5 C_c=6 t=1296							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.1	0.005909	0.000305	0.006376	0.000161	0.000466	0.006214
K-K		0.005767	0.000448	0.006575	0.000361	0.000809	
Jcb-III		0.004519	0.001695	0.006769	0.000554	0.002249	
BBST-II		0.005490	0.000725	0.006764	0.000550	0.001274	
L-P		0.003224	0.002990	0.011517	0.005302	0.008292	
B-P	0.2	0.084405	0.007891	0.096314	0.004018	0.011909	0.092297
K-K		0.079769	0.012528	0.100834	0.008537	0.021065	
Jcb-III		0.044643	0.047653	0.105581	0.013284	0.060938	
BBST-II		0.070942	0.021354	0.105460	0.013163	0.034517	
L-P		0.055189	0.037108	0.137366	0.045069	0.082178	
B-P	0.3	0.284157	0.032744	0.332763	0.015862	0.048606	0.316901
K-K		0.259055	0.057846	0.348731	0.031830	0.089676	
Jcb-III		0.092606	0.224295	0.367332	0.050431	0.274726	
BBST-II		0.211169	0.105732	0.366783	0.049882	0.155614	
L-P		0.218106	0.098795	0.398505	0.081604	0.180399	
B-P	0.4	0.536887	0.058661	0.622164	0.026616	0.085278	0.595548
K-K		0.475839	0.119709	0.645614	0.050066	0.169774	
Jcb-III		0.114029	0.481519	0.676980	0.081432	0.562951	
BBST-II		0.356981	0.238567	0.675856	0.080308	0.318874	
L-P		0.466968	0.128580	0.667144	0.071596	0.200176	
B-P	0.5	0.755219	0.059722	0.839783	0.024842	0.084564	0.814941
K-K		0.667786	0.147155	0.858398	0.043457	0.190613	
Jcb-III		0.190247	0.624694	0.888489	0.073548	0.698242	
BBST-II		0.491586	0.323355	0.887094	0.072153	0.395508	
L-P		0.712266	0.102675	0.853215	0.038274	0.140949	
B-P	0.6	0.899547	0.036974	0.950212	0.013691	0.050665	0.936521
K-K		0.821158	0.115363	0.958639	0.022118	0.137481	
Jcb-III		0.413411	0.523110	0.976362	0.039841	0.562951	
BBST-II		0.656363	0.280158	0.975238	0.038717	0.318874	
L-P		0.883379	0.053142	0.949838*	0.013317	0.066459	
B-P	0.7	0.971910	0.013057	0.989103	0.004136	0.017193	0.984967
K-K		0.930988	0.053979	0.991120	0.006153	0.060132	
Jcb-III		0.722496	0.262471	0.997222	0.012255	0.274726	
BBST-II		0.841059	0.143908	0.996673	0.011706	0.155614	
L-P		0.968083	0.016884	0.987909*	0.002942	0.019826	
B-P	0.8	0.996047	0.002007	0.998571	0.000517	0.002523	0.998054
K-K		0.986645	0.011409	0.998768	0.000714	0.012123	
Jcb-III		0.938786	0.059268	0.999724	0.001670	0.060938	
BBST-II		0.965085	0.032969	0.999602	0.001548	0.034517	
L-P		0.995424	0.002630	0.998401*	0.000347	0.002977	
B-P	0.9	0.999890	0.000050	0.999950	0.000010	0.000060	0.999940
K-K		0.999538	0.000402	0.999953	0.000013	0.000415	
Jcb-III		0.997729	0.002211	0.999979	0.000039	0.002249	
BBST-II		0.998700	0.001240	0.999974	0.000034	0.001274	
L-P		0.999851	0.000089	0.999950	0.000010	0.000099	

Table A3.3 8-node Complete Graph							
n=8 b=28 c=7 C_c=8 t=262144							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.1	0.004306	0.001105	0.005940	0.000529	0.001634	0.005411
K-K		0.004243	0.001168	0.007352	0.001941	0.003110	
Jcb-III		0.002868	0.002543	0.007819	0.002408	0.004951	
BBST-II		0.003965	0.001447	0.007819	0.002408	0.003855	
L-P		0.001550	0.003861	0.010519	0.005108	0.008969	
B-P	0.2	0.090529	0.047982	0.159346	0.020835	0.068817	0.138511
K-K		0.085401	0.053110	0.198001	0.059490	0.112600	
Jcb-III		0.030948	0.107563	0.212718	0.074207	0.181770	
BBST-II		0.071192	0.067319	0.212717	0.074206	0.141525	
L-P		0.057826	0.080685	0.192524	0.054013	0.134698	
B-P	0.3	0.290388	0.189481	0.547838	0.067969	0.257450	0.479869
K-K		0.247798	0.232071	0.628548	0.148679	0.380749	
Jcb-III		0.032023	0.447846	0.667217	0.187348	0.635194	
BBST-II		0.172654	0.307215	0.667213	0.187344	0.494559	
L-P		0.281924	0.197945	0.547930	0.068061	0.266006	
B-P	0.4	0.522905	0.263366	0.855168	0.068897	0.332262	0.786271
K-K		0.386744	0.399527	0.902309	0.116038	0.515564	
Jcb-III		0.009605	0.776666	0.935412	0.149141	0.925807	
BBST-II		0.214577	0.571694	0.935406	0.149135	0.720829	
L-P		0.602658*	0.183613	0.819755*	0.033484	0.217097	
B-P	0.5	0.746441	0.190651	0.968901	0.031809	0.222461	0.937092
K-K		0.498721	0.438371	0.980187	0.043095	0.481467	
Jcb-III		0.007247	0.929845	0.994706	0.057614	0.987459	
BBST-II		0.225868	0.711224	0.994699	0.057607	0.768831	
L-P		0.846916*	0.090176	0.946578*	0.009486	0.099662	
B-P	0.6	0.907526	0.079243	0.994386	0.007617	0.086859	0.986769
K-K		0.616341	0.370428	0.995851	0.009082	0.379510	
Jcb-III		0.074042	0.912727	0.999849	0.013080	0.925807	
BBST-II		0.279014	0.707755	0.999843	0.013074	0.720829	
L-P		0.961012*	0.025757	0.988587*	0.001818	0.027575	
B-P	0.7	0.981079	0.017165	0.999167	0.000923	0.018088	0.998244
K-K		0.776842	0.221402	0.999312	0.001068	0.222470	
Jcb-III		0.364804	0.633440	0.999998	0.001754	0.635194	
BBST-II		0.505435	0.492809	0.999994	0.001750	0.494559	
L-P		0.994229*	0.004015	0.998470*	0.000226	0.004241	
B-P	0.8	0.998575	0.001323	0.999939	0.000041	0.001364	0.999898
K-K		0.942675	0.057223	0.999947	0.000049	0.057272	
Jcb-III		0.818229	0.181669	0.999999	0.000101	0.181770	
BBST-II		0.858473	0.141425	0.999998	0.000100	0.141525	
L-P		0.999647*	0.000251	0.999910*	0.000012	0.000264	
B-P	0.9	0.999990	0.000009	0.999999	0.000000	0.000010	0.999999
K-K		0.998533	0.001466	0.999999	0.000000	0.001466	
Jcb-III		0.995049	0.004950	1.000000	0.000001	0.004951	
BBST-II		0.996145	0.003854	1.000000	0.000001	0.003855	
L-P		0.999997*	0.000002	0.999999	0.000000	0.000002	

Table A3.4 10-node Complete Graph							
n=10 b=45 c=9 C_c=10 t=1e+08							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.2	0.091362	0.125867	0.279682	0.062453	0.188320	0.217229
K-K		0.089205	0.128024	0.396634	0.179405	0.307429	
Jcb-III		0.016615	0.200614	0.428660	0.211431	0.412045	
BBST-II		0.063113	0.154116	0.428660	0.211431	0.365547	
L-P		0.070760	0.146469	0.273324*	0.056095	0.202564	
B-P	0.3	0.253727	0.395239	0.775526	0.126560	0.521798	0.648966
K-K		0.224236	0.424730	0.870858	0.221892	0.646622	
Jcb-III		0.005219	0.643747	0.911850	0.262884	0.906631	
BBST-II		0.107529	0.541437	0.911850	0.262884	0.804321	
L-P		0.373152*	0.275814	0.690242*	0.041276	0.317090	
B-P	0.4	0.472446	0.428830	0.964829	0.063553	0.492383	0.901276
K-K		0.348049	0.553227	0.980419	0.079143	0.632371	
Jcb-III		0.000270	0.901006	0.996629	0.095353	0.996359	
BBST-II		0.112706	0.788570	0.996629	0.095353	0.883924	
L-P		0.734596*	0.166680	0.912872*	0.011596	0.178276	
B-P	0.5	0.727790	0.252659	0.994716	0.014267	0.266927	0.980449
K-K		0.468755	0.511694	0.996073	0.015624	0.527317	
Jcb-III		0.000036	0.980413	0.999970	0.019521	0.999934	
BBST-II		0.112874	0.867575	0.999970	0.019521	0.887096	
L-P		0.930584*	0.049865	0.982559*	0.002110	0.051974	
B-P	0.6	0.913862	0.083512	0.999191	0.001817	0.085329	0.997374
K-K		0.585332	0.412042	0.999349	0.001975	0.414017	
Jcb-III		0.003641	0.993733	1.000000	0.002626	0.996359	
BBST-II		0.116076	0.881298	1.000000	0.002626	0.883924	
L-P		0.989180*	0.008194	0.997643*	0.000269	0.008463	
B-P	0.7	0.986971	0.012832	0.999923	0.000120	0.012952	0.999803
K-K		0.711289	0.288514	0.999936	0.000133	0.288647	
Jcb-III		0.093369	0.906434	1.000000	0.000197	0.906631	
BBST-II		0.195679	0.804124	1.000000	0.000197	0.804321	
L-P		0.999135*	0.000668	0.999823*	0.000020	0.000688	
B-P	0.8	0.999445	0.000550	0.999997	0.000002	0.000553	0.999995
K-K		0.892959	0.107036	0.999998	0.000003	0.107038	
Jcb-III		0.587956	0.412039	1.000000	0.000005	0.412044	
BBST-II		0.634453	0.365542	1.000000	0.000005	0.365547	
L-P		0.999977*	0.000018	0.999995*	0.000000	0.000018	
B-P	0.9	0.999999	0.000001	1.000000	0.000000	0.000001	1.000000
K-K		0.997181	0.002819	1.000000	0.000000	0.002819	
Jcb-III		0.987970	0.012030	1.000000	0.000000	0.012030	
BBST-II		0.989327	0.010673	1.000000	0.000000	0.010673	
L-P		1.000000*	0.000000	1.000000	0.000000	0.000000	

Table A4.1 (5,3)-$G_{h,k}$ Graph							
n=12 b=15 c=2 C_c=15 t=405 h=5 k=3							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000223	0.000006	0.000239	0.000010	0.000016	0.000229
K-K		0.000216	0.000013	0.000244	0.000015	0.000028	
Jcb-III		0.000180	0.000049	0.000263	0.000034	0.000083	
BBST-II		0.000204	0.000025	0.000251	0.000022	0.000046	
L-P		0.000015	0.000214	0.000990	0.000761	0.000975	
B-P	0.5	0.022858	0.000977	0.025513	0.001679	0.002655	0.023834
K-K		0.021667	0.002167	0.026337	0.002502	0.004669	
Jcb-III		0.015595	0.008240	0.029480	0.005646	0.013885	
BBST-II		0.019714	0.004120	0.027496	0.003662	0.007782	
L-P		0.003174	0.020660	0.054554	0.030719	0.051380	
B-P	0.7	0.267558	0.011959	0.300071	0.020554	0.032513	0.279517
K-K		0.252983	0.026534	0.310162	0.030645	0.057178	
Jcb-III		0.178614	0.100903	0.348654	0.069137	0.170040	
BBST-II		0.229065	0.050452	0.324363	0.044846	0.095297	
L-P		0.085025	0.194492	0.388470	0.108953	0.303445	
B-P	0.9	0.857739	0.009037	0.882310	0.015534	0.024571	0.866776
K-K		0.846724	0.020052	0.889936	0.023160	0.043212	
Jcb-III		0.790520	0.076256	0.919026	0.052250	0.128505	
BBST-II		0.828648	0.038128	0.900668	0.033892	0.072020	
L-P		0.659002	0.207774	0.904373	0.037597	0.245371	
B-P	0.92	0.905994	0.006024	0.922372	0.010354	0.016377	0.912018
K-K		0.898653	0.013365	0.927454	0.015436	0.028802	
Jcb-III		0.861192	0.050826	0.946844	0.034826	0.085652	
BBST-II		0.886605	0.025413	0.934608	0.022590	0.048003	
L-P		0.751318	0.160700	0.937809	0.025791	0.186491	
B-P	0.94	0.945833	0.003289	0.954776	0.005654	0.008943	0.949122
K-K		0.941823	0.007299	0.957552	0.008430	0.015728	
Jcb-III		0.921366	0.027756	0.968140	0.019018	0.046773	
BBST-II		0.935244	0.013878	0.961458	0.012336	0.026214	
L-P		0.840455	0.108667	0.964577	0.015455	0.124122	
B-P	0.96	0.975586	0.001255	0.978997	0.002156	0.003412	0.976841
K-K		0.974057	0.002784	0.980056	0.003215	0.006000	
Jcb-III		0.966253	0.010588	0.984095	0.007254	0.017842	
BBST-II		0.971547	0.005294	0.981546	0.004705	0.009999	
L-P		0.919065	0.057776	0.984115	0.007274	0.065050	
B-P	0.98	0.993892	0.000201	0.994439	0.000346	0.000546	0.994093
K-K		0.993648	0.000445	0.994608	0.000515	0.000960	
Jcb-III		0.992398	0.001695	0.995255	0.001162	0.002856	
BBST-II		0.993246	0.000847	0.994847	0.000754	0.001601	
L-P		0.976892	0.017201	0.996007	0.001914	0.019115	
B-P	0.99	0.998483	0.000028	0.998560	0.000049	0.000077	0.998511
K-K		0.998448	0.000063	0.998584	0.000073	0.000136	
Jcb-III		0.998272	0.000239	0.998675	0.000164	0.000403	
BBST-II		0.998391	0.000120	0.998617	0.000106	0.000226	
L-P		0.993825	0.004686	0.999000	0.000489	0.005175	

Table A4.2 (7,3)-$G_{h,k}$ Graph							
n=16 b=21 c=2 C_c=21 t=5103 h=7 k=3							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000012	0.000002	0.000016	0.000003	0.000004	0.000013
K-K		0.000011	0.000002	0.000021	0.000008	0.000009	
Jcb-III		0.000009	0.000005	0.000025	0.000012	0.000017	
BBST-II		0.000010	0.000003	0.000024	0.000010	0.000014	
L-P		0.000000	0.000013	0.000074	0.000061	0.000074	
B-P	0.5	0.005303	0.001467	0.009027	0.002257	0.003724	0.006770
K-K		0.004943	0.001826	0.012310	0.005540	0.007367	
Jcb-III		0.002534	0.004236	0.015725	0.008955	0.013191	
BBST-II		0.003774	0.002995	0.014406	0.007636	0.010632	
L-P		0.000259	0.006510	0.017679	0.010909	0.017419	
B-P	0.7	0.135111	0.043704	0.235453	0.056638	0.100342	0.178815
K-K		0.116599	0.062216	0.298994	0.120179	0.182395	
Jcb-III		0.042636	0.136179	0.378219	0.199404	0.335583	
BBST-II		0.074195	0.104620	0.344661	0.165846	0.270466	
L-P		0.026112	0.152703	0.266984	0.088169	0.240872	
B-P	0.9	0.767436	0.052231	0.864860	0.045193	0.097425	0.819667
K-K		0.724529	0.095138	0.894964	0.075297	0.170435	
Jcb-III		0.621092	0.198575	0.958238	0.138571	0.337146	
BBST-II		0.652797	0.166870	0.924523	0.104856	0.271726	
L-P		0.514728	0.304939	0.868746	0.049079	0.354018	
B-P	0.92	0.843672	0.035813	0.908374	0.028889	0.064702	0.879485
K-K		0.812049	0.067436	0.926393	0.046908	0.114344	
Jcb-III		0.739076	0.140409	0.967854	0.088369	0.228778	
BBST-II		0.760591	0.118894	0.944977	0.065492	0.184386	
L-P		0.629854	0.249631	0.914034	0.034549	0.284179	
B-P	0.94	0.909573	0.020067	0.944584	0.014944	0.035011	0.929640
K-K		0.890511	0.039129	0.953368	0.023728	0.062857	
Jcb-III		0.848316	0.081324	0.975608	0.045968	0.127292	
BBST-II		0.860286	0.069354	0.962878	0.033238	0.102592	
L-P		0.751055	0.178585	0.950763	0.021123	0.199708	
B-P	0.96	0.959915	0.007836	0.973073	0.005322	0.013158	0.967751
K-K		0.951902	0.015849	0.976069	0.008318	0.024167	
Jcb-III		0.934847	0.032904	0.984409	0.016658	0.049563	
BBST-II		0.939507	0.028244	0.979453	0.011702	0.039946	
L-P		0.867338	0.100413	0.977832	0.010081	0.110493	
B-P	0.98	0.990460	0.001281	0.992524	0.000783	0.002063	0.991741
K-K		0.989049	0.002692	0.992959	0.001218	0.003910	
Jcb-III		0.986153	0.005588	0.994275	0.002534	0.008123	
BBST-II		0.986916	0.004825	0.993463	0.001722	0.006546	
L-P		0.960140	0.031601	0.994415	0.002674	0.034275	
B-P	0.99	0.997733	0.000183	0.998021	0.000105	0.000288	0.997916
K-K		0.997524	0.000392	0.998080	0.000164	0.000556	
Jcb-III		0.997103	0.000813	0.998265	0.000349	0.001162	
BBST-II		0.997212	0.000704	0.998149	0.000233	0.000937	
L-P		0.989067	0.008849	0.998601	0.000685	0.009534	

Table A4.3 (15,3)-$G_{h,k}$ Graph							
n=32 b=45 c=2 C_c=45 t=7.17445e+07 h=15 k=3							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000000	0.000000	0.000000	
B-P	0.5	0.000007	0.000023	0.000125	0.000095	0.000117	0.000030
K-K		0.000007	0.000023	0.002212	0.002181	0.002204	
Jcb-III		0.000002	0.000028	0.003306	0.003276	0.003304	
BBST-II		0.000003	0.000027	0.003156	0.003126	0.003153	
L-P		0.000000	0.000030	0.000179	0.000148	0.000179	
B-P	0.7	0.003785	0.022197	0.097477	0.071495	0.093692	0.025982
K-K		0.003250	0.022731	0.360724	0.334743	0.357474	
Jcb-III		0.000075	0.025907	0.508806	0.482825	0.508732	
BBST-II		0.000294	0.025688	0.485682	0.459701	0.485389	
L-P		0.000163	0.025819	0.059053*	0.033071	0.058890	
B-P	0.9	0.372592	0.280530	0.828136	0.175014	0.455544	0.653122
K-K		0.258847	0.394275	0.891919	0.238797	0.633072	
Jcb-III		0.154194	0.498928	0.995069	0.341947	0.840875	
BBST-II		0.154556	0.498566	0.956848	0.303726	0.802292	
L-P		0.156423	0.496699	0.739700*	0.086578	0.583277	
B-P	0.92	0.519642	0.239818	0.873874	0.114414	0.354232	0.759460
K-K		0.391646	0.367814	0.916148	0.156688	0.524502	
Jcb-III		0.282976	0.476484	0.992008	0.232548	0.709032	
BBST-II		0.283281	0.476179	0.959780	0.200320	0.676499	
L-P		0.262423	0.497037	0.824798*	0.065338	0.562375	
B-P	0.94	0.687592	0.167683	0.915941	0.060666	0.228348	0.855275
K-K		0.570269	0.285006	0.940558	0.085283	0.370290	
Jcb-III		0.476992	0.378283	0.988675	0.133400	0.511683	
BBST-II		0.477212	0.378063	0.965417	0.110142	0.488205	
L-P		0.420078	0.435197	0.897453*	0.042178	0.477375	
B-P	0.96	0.850349	0.081817	0.954559	0.022393	0.104211	0.932166
K-K		0.775650	0.156516	0.965101	0.032935	0.189452	
Jcb-III		0.719322	0.212844	0.987555	0.055389	0.268233	
BBST-II		0.719438	0.212728	0.975363	0.043197	0.255925	
L-P		0.631912	0.300254	0.953097*	0.020931	0.321185	
B-P	0.98	0.965639	0.016747	0.985863	0.003477	0.020224	0.982386
K-K		0.945788	0.036598	0.987830	0.005444	0.042042	
Jcb-III		0.931435	0.050951	0.992449	0.010063	0.061014	
BBST-II		0.931461	0.050925	0.989676	0.007290	0.058215	
L-P		0.866011	0.116375	0.988069	0.005683	0.122058	
B-P	0.99	0.992866	0.002673	0.996024	0.000485	0.003158	0.995539
K-K		0.989262	0.006277	0.996327	0.000788	0.007065	
Jcb-III		0.986701	0.008838	0.997079	0.001540	0.010378	
BBST-II		0.986705	0.008834	0.996607	0.001068	0.009902	
L-P		0.959318	0.036221	0.997004	0.001465	0.037687	

Table A4.4 (5.2)-$G_{h,k}$ Graph							
n=7 b=10 c=2 C_c=5 t=80 h=5 k=2							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.021058	0.000375	0.022184	0.000750	0.001125	0.021433
K-K		0.020383	0.001050	0.022184	0.000750	0.001800	
Jcb-III		0.015432	0.006001	0.024434	0.003001	0.009002	
BBST-II		0.018862	0.002572	0.023434	0.002000	0.004572	
L-P		0.003791	0.017643	0.028704	0.007270	0.024913	
B-P	0.5	0.201172	0.004883	0.215820	0.009765	0.014648	0.206055
K-K		0.192383	0.013672	0.215820	0.009765	0.023438	
Jcb-III		0.127930	0.078125	0.245117	0.039062	0.117188	
BBST-II		0.172573	0.033482	0.232096	0.026041	0.059524	
L-P		0.062500	0.143555	0.229889	0.023834	0.167389	
B-P	0.7	0.599845	0.011118	0.633199	0.022236	0.033354	0.610963
K-K		0.579833	0.031130	0.633199	0.022236	0.053366	
Jcb-III		0.433078	0.177885	0.699906	0.088943	0.266828	
BBST-II		0.534727	0.076236	0.670258	0.059295	0.135532	
L-P		0.329417	0.281546	0.622516*	0.011553	0.293099	
B-P	0.9	0.948410	0.002391	0.955584	0.004783	0.007174	0.950801
K-K		0.944105	0.006696	0.955584	0.004783	0.011479	
Jcb-III		0.912537	0.038264	0.969933	0.019132	0.057396	
BBST-II		0.934402	0.016399	0.963556	0.012755	0.029153	
L-P		0.850306	0.100495	0.950981*	0.000180	0.100675	
B-P	0.92	0.966910	0.001428	0.971194	0.002856	0.004284	0.968338
K-K		0.964339	0.003999	0.971194	0.002856	0.006855	
Jcb-III		0.945489	0.022849	0.979763	0.011425	0.034274	
BBST-II		0.958545	0.009793	0.975954	0.007616	0.017409	
L-P		0.897405	0.070933	0.968404*	0.000066	0.070998	
B-P	0.94	0.981411	0.000700	0.983512	0.001401	0.002101	0.982111
K-K		0.980150	0.001961	0.983512	0.001401	0.003362	
Jcb-III		0.970905	0.011206	0.987714	0.005603	0.016809	
BBST-II		0.977308	0.004803	0.985846	0.003735	0.008538	
L-P		0.938223	0.043888	0.982128*	0.000017	0.043905	
B-P	0.96	0.991782	0.000241	0.992504	0.000481	0.000721	0.992023
K-K		0.991350	0.000673	0.992504	0.000481	0.001154	
Jcb-III		0.988175	0.003848	0.993947	0.001924	0.005771	
BBST-II		0.990374	0.001649	0.993305	0.001282	0.002931	
L-P		0.970620	0.021403	0.992025*	0.000002	0.021406	
B-P	0.98	0.997967	0.000035	0.998071	0.000069	0.000104	0.998002
K-K		0.997904	0.000098	0.998071	0.000069	0.000167	
Jcb-III		0.997446	0.000556	0.998279	0.000277	0.000833	
BBST-II		0.997763	0.000239	0.998187	0.000185	0.000423	
L-P		0.992143	0.005859	0.998002*	0.000000	0.005858	
B-P	0.99	0.999495	0.000005	0.999509	0.000009	0.000014	0.999500
K-K		0.999487	0.000013	0.999509	0.000009	0.000022	
Jcb-III		0.999426	0.000074	0.999537	0.000037	0.000112	
BBST-II		0.999468	0.000032	0.999525	0.000025	0.000057	
L-P		0.997969	0.001531	0.999500*	0.000000	0.001531	

Table A4.5 (5,10)-$G_{h,k}$ Graph

n=47 b=50 c=2 C_c=225 t=50000 h=5 k=10

Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000000	0.000000	0.000000	
B-P	0.5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000002	0.000002	0.000002	
B-P	0.7	0.000044	0.000004	0.000054	0.000006	0.000010	0.000048
K-K		0.000043	0.000005	0.000054	0.000006	0.000011	
Jcb-III		0.000034	0.000014	0.000062	0.000014	0.000028	
BBST-II		0.000040	0.000008	0.000057	0.000008	0.000017	
L-P		0.000001	0.000047	0.014315	0.014267	0.014314	
B-P	0.9	0.187152	0.020233	0.235664	0.028279	0.048512	0.207385
K-K		0.181850	0.025535	0.237325	0.029940	0.055476	
Jcb-III		0.136689	0.070696	0.275254	0.067869	0.138565	
BBST-II		0.166772	0.040613	0.249803	0.042418	0.083031	
L-P		0.043989	0.163396	0.636179	0.428794	0.592190	
B-P	0.92	0.316466	0.029105	0.386248	0.040677	0.069783	0.345571
K-K		0.308839	0.036732	0.388638	0.043067	0.079800	
Jcb-III		0.243876	0.101695	0.443197	0.097626	0.199321	
BBST-II		0.287150	0.058421	0.406587	0.061016	0.119437	
L-P		0.101038	0.244533	0.749066	0.403495	0.648027	
B-P	0.94	0.496462	0.033739	0.577356	0.047155	0.080894	0.530201
K-K		0.487621	0.042580	0.580126	0.049925	0.092505	
Jcb-III		0.412315	0.117886	0.643372	0.113171	0.231057	
BBST-II		0.462479	0.067722	0.600933	0.070732	0.138454	
L-P		0.218308	0.311893	0.850192	0.319991	0.631884	
B-P	0.96	0.712618	0.026891	0.777091	0.037582	0.064473	0.739509
K-K		0.705572	0.033937	0.779299	0.039790	0.073728	
Jcb-III		0.645552	0.093957	0.829707	0.090198	0.184155	
BBST-II		0.685534	0.053975	0.795883	0.056374	0.110349	
L-P		0.434306	0.305203	0.930477	0.190968	0.496172	
B-P	0.98	0.912699	0.008859	0.933940	0.012382	0.021241	0.921558
K-K		0.910378	0.011180	0.934667	0.013109	0.024289	
Jcb-III		0.890604	0.030954	0.951274	0.029716	0.060670	
BBST-II		0.903776	0.017782	0.940131	0.018573	0.036354	
L-P		0.758055	0.163503	0.982158	0.060600	0.224102	
B-P	0.99	0.977059	0.001785	0.981338	0.002494	0.004279	0.978844
K-K		0.976592	0.002252	0.981485	0.002641	0.004893	
Jcb-III		0.972609	0.006235	0.984830	0.005986	0.012221	
BBST-II		0.975262	0.003582	0.982585	0.003741	0.007323	
L-P		0.919543	0.059301	0.995510	0.016666	0.075967	

Table A4.6 (7,6)-$G_{h,k}$ Graph							
n=37 b=42 c=2 C_c=105 t=326592 h=7 k=6							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000000	0.000000	0.000000	
B-P	0.5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000042	0.000042	0.000042	
B-P	0.7	0.001424	0.000656	0.003578	0.001498	0.002154	0.002080
K-K		0.001363	0.000717	0.004381	0.002301	0.003018	
Jcb-III		0.000681	0.001400	0.005978	0.003897	0.005297	
BBST-II		0.001010	0.001070	0.005332	0.003251	0.004321	
L-P		0.000031	0.002049	0.036843	0.034763	0.036812	
B-P	0.9	0.319055	0.107932	0.568863	0.141876	0.249808	0.426987
K-K		0.292329	0.134658	0.606698	0.179711	0.314369	
Jcb-III		0.186945	0.240042	0.752211	0.325224	0.565266	
BBST-II		0.222138	0.204849	0.683277	0.256290	0.461139	
L-P		0.103631	0.323356	0.703448	0.276461	0.599817	
B-P	0.92	0.457369	0.111783	0.700179	0.131027	0.242811	0.569152
K-K		0.426202	0.142950	0.728873	0.159721	0.302671	
Jcb-III		0.316726	0.252426	0.864520	0.295368	0.547793	
BBST-II		0.350831	0.218321	0.797716	0.228564	0.446885	
L-P		0.192839	0.376313	0.798740	0.229588	0.605901	
B-P	0.94	0.626078	0.093411	0.814866	0.095377	0.188787	0.719489
K-K		0.596641	0.122848	0.830764	0.111275	0.234122	
Jcb-III		0.504424	0.215065	0.931634	0.212145	0.427210	
BBST-II		0.531022	0.188467	0.879536	0.160047	0.348514	
L-P		0.340636	0.378853	0.881415	0.161926	0.540779	
B-P	0.96	0.804794	0.053738	0.904886	0.046354	0.100092	0.858532
K-K		0.785586	0.072946	0.910089	0.051557	0.124503	
Jcb-III		0.731759	0.126773	0.961257	0.102725	0.229498	
BBST-II		0.746047	0.112485	0.933269	0.074737	0.187222	
L-P		0.561248	0.297284	0.945497	0.086965	0.384249	
B-P	0.98	0.948064	0.012799	0.969818	0.008955	0.021754	0.960863
K-K		0.942859	0.018004	0.970390	0.009527	0.027531	
Jcb-III		0.929741	0.031122	0.981110	0.020247	0.051369	
BBST-II		0.932939	0.027924	0.974845	0.013982	0.041906	
L-P		0.831127	0.129736	0.986095	0.025232	0.154968	
B-P	0.99	0.987628	0.002194	0.991177	0.001355	0.003549	0.989822
K-K		0.986675	0.003147	0.991243	0.001421	0.004568	
Jcb-III		0.984392	0.005430	0.992972	0.003150	0.008580	
BBST-II		0.984926	0.004896	0.991926	0.002104	0.006999	
L-P		0.947122	0.042700	0.996506	0.006684	0.049384	

Table A4.7 (10,6)-$G_{h,k}$ Graph							
n=52 b=60 c=2 C_c=150 t=1.00777e+08 h=10 k=6							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.3	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000000	0.000000	0.000000	
B-P	0.5	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000	0.000000	0.000000	
Jcb-III		0.000000	0.000000	0.000000	0.000000	0.000000	
BBST-II		0.000000	0.000000	0.000000	0.000000	0.000000	
L-P		0.000000	0.000000	0.000001	0.000001	0.000001	
B-P	0.7	0.000068	0.000097	0.000530	0.000365	0.000462	0.000165
K-K		0.000067	0.000098	0.001491	0.001326	0.001424	
Jcb-III		0.000025	0.000140	0.002233	0.002068	0.002208	
BBST-II		0.000040	0.000125	0.002046	0.001881	0.002006	
L-P		0.000000	0.000165	0.008955	0.008790	0.008955	
B-P	0.9	0.131280	0.165882	0.551469	0.254307	0.420189	0.297162
K-K		0.110530	0.186632	0.650711	0.353549	0.540182	
Jcb-III		0.050185	0.246977	0.855504	0.558342	0.805319	
BBST-II		0.055675	0.241487	0.787257	0.490095	0.731582	
L-P		0.028294	0.268868	0.605006	0.307844	0.576712	
B-P	0.92	0.239485	0.207880	0.696998	0.249633	0.457513	0.447365
K-K		0.204877	0.242488	0.756100	0.308735	0.551223	
Jcb-III		0.124262	0.323103	0.944326	0.496961	0.820064	
BBST-II		0.129852	0.317513	0.874829	0.427464	0.744977	
L-P		0.072284	0.375081	0.725403	0.278038	0.653119	
B-P	0.94	0.414678	0.210243	0.812495	0.187574	0.397817	0.624921
K-K		0.368226	0.256695	0.835154	0.210233	0.466929	
Jcb-III		0.279118	0.345803	0.975905	0.350984	0.696787	
BBST-II		0.283868	0.341053	0.916855	0.291934	0.632987	
L-P		0.173000	0.451921	0.834999	0.210078	0.661999	
B-P	0.96	0.657438	0.146781	0.896298	0.092079	0.238860	0.804219
K-K		0.614626	0.189593	0.901154	0.096935	0.286529	
Jcb-III		0.545103	0.259116	0.976906	0.172687	0.431804	
BBST-II		0.548046	0.256173	0.940313	0.136094	0.392267	
L-P		0.379059	0.425160	0.923057	0.118838	0.543998	
B-P	0.98	0.901929	0.042633	0.962546	0.017984	0.060617	0.944562
K-K		0.885640	0.058922	0.963197	0.018635	0.077557	
Jcb-III		0.862669	0.081893	0.981408	0.036846	0.118739	
BBST-II		0.863478	0.081084	0.971345	0.026783	0.107867	
L-P		0.720911	0.223651	0.980195	0.035633	0.259284	
B-P	0.99	0.977402	0.008089	0.988267	0.002776	0.010865	0.985491
K-K		0.973878	0.011613	0.988387	0.002896	0.014510	
Jcb-III		0.969206	0.016285	0.991626	0.006135	0.022420	
BBST-II		0.969359	0.016132	0.989726	0.004235	0.020367	
L-P		0.904424	0.081067	0.995012	0.009521	0.090588	

Table A5.1 Cycle(7)-Complete(5) Graph							
n=11 b=17 c=2 C_c=21 t=875							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.5	0.034279	0.010155	0.057220	0.012787	0.022942	0.044434
K-K		0.029045	0.015389	0.087456	0.043022	0.058411	
Jcb-III		0.007690	0.036743	0.172668	0.128234	0.164978	
BBST-II		0.015113	0.029321	0.147194	0.102760	0.132081	
L-P		0.005859	0.038574	0.137484	0.093050	0.131625	
B-P	0.6	0.108777	0.029272	0.173355	0.035306	0.064579	0.138049
K-K		0.087551	0.050498	0.243874	0.105825	0.156323	
Jcb-III		0.019407	0.118642	0.454929	0.316880	0.435522	
BBST-II		0.039002	0.099047	0.387680	0.249631	0.348677	
L-P		0.030233	0.107816	0.316683	0.178634	0.286450	
B-P	0.7	0.265023	0.050398	0.371878	0.056457	0.106855	0.315421
K-K		0.211839	0.103582	0.465766	0.150345	0.253927	
Jcb-III		0.073818	0.241603	0.771648	0.456227	0.697830	
BBST-II		0.105214	0.210207	0.663895	0.348474	0.558680	
L-P		0.112990	0.202431	0.549693	0.234272	0.436703	
B-P	0.8	0.523707	0.048203	0.619815	0.047905	0.096108	0.571910
K-K		0.447764	0.124146	0.687369	0.115459	0.239606	
Jcb-III		0.281270	0.290640	0.933984	0.362074	0.652714	
BBST-II		0.310637	0.261273	0.833198	0.261288	0.522561	
L-P		0.322123	0.249787	0.777760	0.205850	0.455638	
B-P	0.92	0.887172	0.010048	0.905161	0.007941	0.017989	0.897220
K-K		0.861112	0.036108	0.915922	0.018702	0.054809	
Jcb-III		0.811260	0.085960	0.961331	0.064111	0.150071	
BBST-II		0.818012	0.079208	0.938159	0.040939	0.120146	
L-P		0.781899	0.115321	0.962052	0.064832	0.180152	
B-P	0.94	0.933096	0.005066	0.941976	0.003814	0.008879	0.938162
K-K		0.918816	0.019346	0.947277	0.009115	0.028462	
Jcb-III		0.891938	0.046224	0.970085	0.031923	0.078148	
BBST-II		0.895454	0.042708	0.958019	0.019857	0.062565	
L-P		0.861784	0.076378	0.978543	0.040381	0.116758	
B-P	0.96	0.968822	0.001785	0.971884	0.001277	0.003063	0.970607
K-K		0.963351	0.007256	0.973726	0.003119	0.010375	
Jcb-III		0.953205	0.017402	0.981784	0.011177	0.028579	
BBST-II		0.954491	0.016116	0.977371	0.006764	0.022880	
L-P		0.930766	0.039841	0.990428	0.019821	0.059662	
B-P	0.98	0.991878	0.000265	0.992322	0.000179	0.000444	0.992143
K-K		0.990998	0.001145	0.992593	0.000450	0.001595	
Jcb-III		0.989387	0.002756	0.993796	0.001653	0.004409	
BBST-II		0.989586	0.002557	0.993115	0.000972	0.003530	
L-P		0.980487	0.011656	0.997602	0.005459	0.017114	
B-P	0.99	0.997933	0.000036	0.997993	0.000024	0.000060	0.997969
K-K		0.997808	0.000161	0.998029	0.000060	0.000221	
Jcb-III		0.997582	0.000387	0.998194	0.000225	0.000612	
BBST-II		0.997609	0.000360	0.998099	0.000130	0.000490	
L-P		0.994820	0.003149	0.999400	0.001431	0.004580	

Table A5.2 Cycle(20)-Complete(5) Graph							
n=24 b=30 c=2 C_c=190 t=2500							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.9	0.377378	0.014170	0.417759	0.026211	0.040380	0.391548
K-K		0.339838	0.051710	0.482129	0.090581	0.142291	
Jcb-III		0.311937	0.079611	0.874759	0.483211	0.562822	
BBST-II		0.312628	0.078920	0.628929	0.237381	0.316301	
L-P		0.292477	0.099071	0.825838	0.434290	0.533361	
B-P	0.91	0.438367	0.013085	0.475226	0.023774	0.036860	0.451452
K-K		0.402241	0.049211	0.531866	0.080414	0.129625	
Jcb-III		0.375796	0.075656	0.875017	0.423565	0.499221	
BBST-II		0.376409	0.075043	0.656966	0.205514	0.280557	
L-P		0.350825	0.100627	0.856596	0.405144	0.505771	
B-P	0.92	0.505140	0.011609	0.537435	0.020686	0.032295	0.516749
K-K		0.471740	0.045009	0.585388	0.068639	0.113648	
Jcb-III		0.447645	0.069104	0.874002	0.357253	0.426357	
BBST-II		0.448169	0.068580	0.687777	0.171028	0.239609	
L-P		0.417290	0.099459	0.885011	0.368262	0.467721	
B-P	0.94	0.652660	0.007751	0.673630	0.013219	0.020970	0.660411
K-K		0.628444	0.031967	0.702972	0.042561	0.074528	
Jcb-III		0.611443	0.048968	0.877373	0.216962	0.265930	
BBST-II		0.611769	0.048642	0.761219	0.100808	0.149450	
L-P		0.573479	0.086932	0.933723	0.273312	0.360244	
B-P	0.95	0.730195	0.005621	0.745168	0.009352	0.014973	0.735816
K-K		0.711896	0.023920	0.765617	0.029801	0.053721	
Jcb-III		0.699211	0.036605	0.886459	0.150643	0.187248	
BBST-II		0.699441	0.036375	0.804673	0.068857	0.105232	
L-P		0.660817	0.074999	0.953530	0.217714	0.292713	
B-P	0.96	0.806729	0.003598	0.816153	0.005826	0.009424	0.810327
K-K		0.794528	0.015799	0.828769	0.018442	0.034242	
Jcb-III		0.786171	0.024156	0.902920	0.092593	0.116749	
BBST-II		0.786315	0.024012	0.851926	0.041599	0.065612	
L-P		0.750826	0.059501	0.970024	0.159697	0.219198	
B-P	0.97	0.878266	0.001892	0.883135	0.002977	0.004869	0.880158
K-K		0.871579	0.008579	0.889555	0.009397	0.017976	
Jcb-III		0.867051	0.013107	0.927096	0.046938	0.060045	
BBST-II		0.867125	0.013033	0.900870	0.020712	0.033745	
L-P		0.838758	0.041400	0.983035	0.102877	0.144277	
B-P	0.98	0.939403	0.000697	0.941163	0.001063	0.001761	0.940100
K-K		0.936835	0.003265	0.943460	0.003360	0.006625	
Jcb-III		0.935116	0.004984	0.956832	0.016732	0.021716	
BBST-II		0.935142	0.004958	0.947347	0.007247	0.012204	
L-P		0.917387	0.022713	0.992427	0.052327	0.075040	
B-P	0.99	0.983032	0.000109	0.983300	0.000159	0.000268	0.983141
K-K		0.982617	0.000524	0.983647	0.000506	0.001030	
Jcb-III		0.982343	0.000798	0.985660	0.002519	0.003318	
BBST-II		0.982347	0.000794	0.984211	0.001070	0.001865	
L-P		0.976145	0.006996	0.998102	0.014961	0.021956	

Table A5.3 Cycle(7)-Complete(8) Graph							
n=14 b=35 c=2 C_c=21 t=1.83501e+06							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.5	0.020906	0.037662	0.173491	0.114923	0.152586	0.058568
K-K		0.019502	0.039066	0.469691	0.411123	0.450189	
Jcb-III		0.000053	0.058515	0.912321	0.853753	0.912267	
BBST-II		0.001187	0.057381	0.880123	0.821555	0.878936	
L-P		0.000916	0.057653	0.168470*	0.109902	0.167555	
B-P	0.6	0.067984	0.088548	0.339482	0.182950	0.271498	0.156532
K-K		0.057850	0.098682	0.598135	0.441603	0.540285	
Jcb-III		0.000009	0.156523	0.994742	0.838210	0.994733	
BBST-II		0.001245	0.155287	0.959634	0.803102	0.958388	
L-P		0.008098	0.148434	0.347289	0.190757	0.339191	
B-P	0.7	0.188280	0.140559	0.502208	0.173369	0.313928	0.328839
K-K		0.142460	0.186379	0.700146	0.371307	0.557686	
Jcb-III		0.000460	0.328379	0.999927	0.671088	0.999467	
BBST-II		0.001702	0.327137	0.964652	0.635813	0.962949	
L-P		0.047476	0.281363	0.567000	0.238161	0.519525	
B-P	0.8	0.438227	0.138431	0.677782	0.101124	0.239555	0.576658
K-K		0.308160	0.268498	0.801845	0.225187	0.493684	
Jcb-III		0.018508	0.558150	0.999468	0.422810	0.980960	
BBST-II		0.019727	0.556931	0.964845	0.388187	0.945118	
L-P		0.197912	0.378746	0.782688	0.206030	0.584776	
B-P	0.92	0.864218	0.033187	0.911233	0.013828	0.047016	0.897405
K-K		0.732029	0.165376	0.933888	0.036483	0.201859	
Jcb-III		0.452910	0.444495	0.991422	0.094017	0.538512	
BBST-II		0.453580	0.443825	0.972415	0.075010	0.518836	
L-P		0.690036	0.207369	0.962209	0.064804	0.272173	
B-P	0.94	0.920967	0.017256	0.944822	0.006599	0.023855	0.938223
K-K		0.831224	0.106999	0.956362	0.018139	0.125138	
Jcb-III		0.639054	0.299169	0.990188	0.051965	0.351135	
BBST-II		0.639490	0.298733	0.977795	0.039572	0.338305	
L-P		0.796310	0.141913	0.978593	0.040370	0.182283	
B-P	0.96	0.964336	0.006284	0.972829	0.002209	0.008493	0.970620
K-K		0.921368	0.049252	0.976973	0.006353	0.055605	
Jcb-III		0.827797	0.142823	0.991264	0.020644	0.163467	
BBST-II		0.828000	0.142620	0.985495	0.014875	0.157495	
L-P		0.894066	0.076554	0.990438	0.019818	0.096372	
B-P	0.98	0.991181	0.000962	0.992455	0.000312	0.001274	0.992143
K-K		0.982470	0.009673	0.993085	0.000942	0.010615	
Jcb-III		0.963149	0.028994	0.995687	0.003544	0.032539	
BBST-II		0.963189	0.028954	0.994539	0.002396	0.031350	
L-P		0.968968	0.023175	0.997602	0.005459	0.028634	
B-P	0.99	0.997836	0.000133	0.998010	0.000041	0.000174	0.997969
K-K		0.996448	0.001521	0.998097	0.000128	0.001650	
Jcb-III		0.993339	0.004630	0.998493	0.000524	0.005154	
BBST-II		0.993346	0.004623	0.998311	0.000342	0.004965	
L-P		0.991599	0.006370	0.999400	0.001431	0.007801	

Table A5.4 Cycle(20)-Complete(8) Graph							
n=27 b=48 c=2 C_c=190 t=5.24288e+06							
Bounds	p	lower	Δlower	upper	Δupper	$r(p)$	actual
B-P	0.91	0.409820	0.041782	0.506399	0.054797	0.096579	0.451602
K-K		0.255569	0.196033	0.657333	0.205731	0.401764	
Jcb-III		0.161369	0.290233	0.979902	0.528300	0.818533	
BBST-II		0.161369	0.290233	0.842029	0.390427	0.680659	
L-P		0.287623	0.163979	0.856820	0.405218	0.569198	
B-P	0.92	0.479226	0.037630	0.563217	0.046361	0.083990	0.516856
K-K		0.321025	0.195831	0.694851	0.177995	0.373825	
Jcb-III		0.224150	0.292706	0.973747	0.456891	0.749598	
BBST-II		0.224150	0.292706	0.847485	0.330629	0.623335	
L-P		0.352399	0.164457	0.885156	0.368300	0.532758	
B-P	0.93	0.554606	0.032251	0.624235	0.037378	0.069630	0.586857
K-K		0.399906	0.186951	0.734234	0.147377	0.334328	
Jcb-III		0.304791	0.282066	0.966950	0.380093	0.662160	
BBST-II		0.304791	0.282066	0.855416	0.268559	0.550626	
L-P		0.427372	0.159485	0.910894	0.324037	0.483522	
B-P	0.94	0.634531	0.025924	0.688759	0.028304	0.054228	0.660455
K-K		0.492388	0.168067	0.775558	0.115103	0.283170	
Jcb-III		0.404543	0.255912	0.960287	0.299832	0.555743	
BBST-II		0.404543	0.255912	0.866677	0.206222	0.462134	
L-P		0.512347	0.148108	0.933772	0.273317	0.421424	
B-P	0.95	0.716733	0.019107	0.755518	0.019678	0.038786	0.735840
K-K		0.596766	0.139074	0.818748	0.082908	0.221982	
Jcb-III		0.522174	0.213666	0.955128	0.219288	0.432954	
BBST-II		0.522174	0.213666	0.882201	0.146361	0.360027	
L-P		0.606096	0.129744	0.953554	0.217714	0.347457	
B-P	0.96	0.797906	0.012432	0.822432	0.012094	0.024526	0.810338
K-K		0.708368	0.101970	0.863370	0.053032	0.155003	
Jcb-III		0.652315	0.158023	0.953511	0.143173	0.301196	
BBST-II		0.652315	0.158023	0.902778	0.092440	0.250463	
L-P		0.705801	0.104537	0.970034	0.159696	0.264233	
B-P	0.97	0.873511	0.006651	0.886281	0.006119	0.012770	0.880162
K-K		0.818476	0.061686	0.908221	0.028059	0.089745	
Jcb-III		0.783766	0.096396	0.957879	0.077717	0.174113	
BBST-II		0.783766	0.096396	0.928552	0.048390	0.144786	
L-P		0.806279	0.073883	0.983038	0.102876	0.176759	
B-P	0.98	0.937608	0.002493	0.942274	0.002173	0.004666	0.940101
K-K		0.913863	0.026238	0.950569	0.010468	0.036706	
Jcb-III		0.898769	0.041332	0.969994	0.029893	0.071225	
BBST-II		0.898769	0.041332	0.957997	0.017896	0.059228	
L-P		0.898921	0.041180	0.992427	0.052326	0.093506	
B-P	0.99	0.982747	0.000394	0.983466	0.000325	0.000719	0.983141
K-K		0.978428	0.004713	0.984795	0.001654	0.006367	
Jcb-III		0.975660	0.007481	0.988033	0.004892	0.012373	
BBST-II		0.975660	0.007481	0.985949	0.002808	0.010289	
L-P		0.970254	0.012887	0.998102	0.014961	0.027847	

Table A6.1 Arpanet (1979 version, Figure 8)				
n=59 b=71 c=2 C_c=57 t=2.72817e+11				
Bounds	p	lower	upper	$r(p)$
B-P	0.5	0.000000	0.000000	0.000000
K-K		0.000000	0.000000	0.000000
Jcb-III		0.000000	0.000000	0.000000
BBST-II		0.000000	0.000000	0.000000
L-P		0.000000	0.000001	0.000001
B-P	0.7	0.000148	0.002483	0.002335
K-K		0.000148	0.007398	0.007250
Jcb-III		0.000045	0.008790	0.008745
BBST-II		0.000086	0.008590	0.008503
L-P		0.000000	0.017175	0.017175
B-P	0.9	0.129562	0.839673	0.710111
K-K		0.113389	0.881892	0.768503
Jcb-III		0.021977	0.976842	0.954865
BBST-II		0.026460	0.954940	0.928480
L-P		0.015086	0.675115*	0.660029
B-P	0.92	0.229425	0.905370	0.675945
K-K		0.196889	0.917562	0.720673
Jcb-III		0.068567	0.994977	0.926410
BBST-II		0.072917	0.973728	0.900811
L-P		0.044769	0.780114*	0.735345
B-P	0.94	0.397043	0.939865	0.542822
K-K		0.344436	0.941535	0.597100
Jcb-III		0.190670	0.996848	0.806178
BBST-II		0.194455	0.978356	0.783901
L-P		0.123792	0.871119*	0.747327
B-P	0.96	0.642423	0.963840	0.321417
K-K		0.584093	0.963901	0.379808
Jcb-III		0.450476	0.994542	0.544065
BBST-II		0.453031	0.982062	0.529031
L-P		0.311079	0.941208*	0.630129
B-P	0.97	0.779467	0.975281	0.195815
K-K		0.732273	0.975286	0.243013
Jcb-III		0.634748	0.993728	0.358980
BBST-II		0.636434	0.985494	0.349061
L-P		0.468286	0.966686*	0.498399
B-P	0.98	0.901379	0.986323	0.084944
K-K		0.874712	0.986323	0.111611
Jcb-III		0.824434	0.994344	0.169910
BBST-II		0.825232	0.990446	0.165215
L-P		0.669216	0.985141*	0.315924
B-P	0.99	0.979896	0.995639	0.015743
K-K		0.973574	0.995639	0.022065
Jcb-III		0.962586	0.997151	0.034565
BBST-II		0.962749	0.996358	0.033609
L-P		0.882061	0.996286	0.114225

Table A6.2 A 6-cohesive Graph				
n=12 b=36 c=6 C_c=12 t=6.04662e+07				
Bounds	p	lower	upper	$r(p)$
B-P	0.3	0.080240	0.265126	0.184887
K-K		0.078728	0.372950	0.294222
Jcb-III		0.014365	0.405289	0.390924
BBST-II		0.053708	0.405286	0.351578
L-P		0.046991	0.252380*	0.205390
B-P	0.37	0.163247	0.565863	0.402617
K-K		0.154568	0.693551	0.538983
Jcb-III		0.010357	0.742084	0.731727
BBST-II		0.084000	0.742080	0.658080
L-P		0.158294	0.491546*	0.333252
B-P	0.38	0.176684	0.607615	0.430931
K-K		0.165988	0.731090	0.565102
Jcb-III		0.009309	0.780124	0.770814
BBST-II		0.086886	0.780119	0.693233
L-P		0.181280*	0.525582*	0.344302
B-P	0.5	0.382003	0.925124	0.543120
K-K		0.310007	0.957567	0.647560
Jcb-III		0.000915	0.986472	0.985557
BBST-II		0.100103	0.986465	0.886362
L-P		0.546704*	0.840943*	0.294239
B-P	0.7	0.856081	0.997084	0.141003
K-K		0.602483	0.997603	0.395120
Jcb-III		0.053569	0.999999	0.946430
BBST-II		0.148819	0.999993	0.851174
L-P		0.959801*	0.992010*	0.032209
B-P	0.9	0.999531	0.999992	0.000461
K-K		0.983049	0.999993	0.016944
Jcb-III		0.937171	0.999999	0.062828
BBST-II		0.943495	0.999999	0.056505
L-P		0.999934*	0.999989*	0.000055
B-P	0.91	0.999754	0.999996	0.000242
K-K		0.989590	0.999996	0.010406
Jcb-III		0.960956	1.000000	0.039043
BBST-II		0.964886	0.999999	0.035114
L-P		0.999965*	0.999994*	0.000029
B-P	0.92	0.999882	0.999998	0.000116
K-K		0.994137	0.999998	0.005861
Jcb-III		0.977766	1.000000	0.022234
BBST-II		0.980004	1.000000	0.019996
L-P		0.999983*	0.999997*	0.000014
B-P	0.93	0.999949	0.999999	0.000050
K-K		0.997042	0.999999	0.002957
Jcb-III		0.988666	1.000000	0.011334
BBST-II		0.989807	1.000000	0.010193
L-P		0.999992*	0.999999	0.000006