1

AFIT/GOR/ENS/88D-01

RESPONSE SURFACE ANALYSIS OF STOCHASTIC NETWORK PERFORMANCE

THESIS

Thomas G. Bailey
Captain, USAF

AFIT/GOR/ENS/88D-1

Approved for public release; distribution unlimited

AFIT/GOR/ENS/88D-01

RESPONSE SURFACE ANALYSIS OF STOCHASTIC NETWORK PERFORMANCE

THESIS

Presented to the School of Engineering
of the Air Force Institute of Technology
Air University
In Partial Fulfillment of the
Requirements for the Degree of
Master of Science in Operations Research

Thomas G. Bailey, B.S., M.A.
Captain, USAF

December 1988

Approved for public release; distribution unlimited

Preface

The following document reflects the time, patience, and assistance of many people. My personal thanks in this small way for their contributions is not enough.

First and foremost, I extend my deepest gratitude to my adviser, Major Ken Bauer, for the interest, insight, and perspective he gave to me and this project. I also thank Dr. Yupo Chan for his contributions to my thesis, and his efforts in securing the support and funding for this research. Finally, I offer my appreciation to the thesis sponsors, Dr. Albert Marsh and Captain Dave Knue, for their support.

Finally, I thank my lovely wife Beverly for all her patience, love, and support during the past 20 months of graduate school.

Glenn Bailey

Table of Contents

List of Figures

List of Tables

Table Page

AFIT/GOR/ENS/88D-1

Abstract

The objective of this thesis was to analyze stochastic binary networks for the purpose of improving their performance as measured by expected maximum flow and source-to-sink reliability. The capacity and survivability of the networks' nodes and arcs formed the parameters of interest in the experimental design used to develop a response surface model. Estimates of network performance was provided by Monte Carlo simulation using a FORTRAN based program designed for this study called MAXFLO. MAXFLO implemented an original form of maximum flow calculation using minimal cuts instead of paths to improve the simulation's speed. MAXFLO was also compiled and run on a VAX 8650, VAX 11/785, and SUN-3 workstation under UNIX and VMS systems to insure portability and simulation performance. (Keywords; Monte Carlo method, network flows,

Additional research investigated the use of a scalar internal control variate to reduce the variance of the maximum flow estimates. Specifically, the effect of the number of failed nodes of a selected control subset was regressed out of the simulation response to reduce the variance as much as 24%. This feature was incorporated into MAXFLO as a user option for any network. The results indicated further variance reduction may be realized by

statistical analysis, experimental data. (KR)

expanding to a multivariate set of controls that includes both nodes and arcs.

Finally, response surface methodology was implemented to provide an efficient analysis of stochastic network performance. Nineteen parameters of particular interest in a specific network were screened using a Plackett-Burman design, resulting in five parameters of significant influence. A full 2^5 factorial orthogonal design was developed, with two first-order polynomials approximating the response surfaces of expected maximum flow and network reliability regressed from the experimental results. In addition to the descriptive insight provided by the response surfaces, a prescriptive example of an optimized network improvement strategy was developed by incorporating the response surface equations in a linear programming formulation. Additionally, a correlation of response surface coefficients and control variate effectiveness was empirically shown, suggesting promising future research in this area.

RESPONSE SURFACE ANALYSIS OF STOCHASTIC NETWORK PERFORMANCE

I. Introduction

This thesis' objective is to improve the analysis of stochastic networks. It accomplishes this task by developing an efficient Monte Carlo simulation program, using the variance and bias reduction techniques of control variates and antithetic random numbers, and introducing the technique of response surface methodology (RSM) to the simulation output analysis. The immediate application of this study is limited to analyzing Department of Defense (DOD) communication networks. However, the theoretic aspects of the research can be expanded to networks and simulation in general (Bauer, 1988a).

Background (Marsh and Knue, 1988)

One current area of stochastic network research is reliability and performance improvement. The stochastic nature of the problem makes finding a solution difficult because for each network there exists, as a function of its individual components' probability of survival, an exponentially large number of possible network configurations or subsets, with each subset having a different flow pattern. Further compounding the problem is the lack of independence of survival probabilities of certain components. Consequently, a complete enumeration and subsequent maximum flow calculation of all possible network configurations is impractical.

Therefore, Monte Carlo simulation is a popular method of analyzing network effectiveness, with the estimated expected maximum flow as the measure of performance. Specifically, current techniques embed a maximum flow algorithm in a Monte Carlo routine for a defined number of replications, or *sample size*. In each sample, all stochastic components are individually evaluated according to their probability of survival (P_s) and a separate, independent random number draw from a uniform distribution U(0,1). If the random draw is higher than P_s, then that component is "destroyed" in the current sample's maximum flow evaluation. In the course of the simulation, those subsets of the original network most likely to survive are evaluated, thus producing an unbiased estimate of the expected maximum flow and variance.

The purpose of the simulation analysis is to find those components whose increase in capacity or survivability will most improve the network's performance as expressed in terms of expected maximum flow. One obvious procedure to accomplish this task is to run several simulations of a particular network, each time changing a parameter selected by the analyst either for its potential in improving network performance, or simply because it can realistically be improved. Unfortunately, the large number of components in most networks make this approach a time-intensive procedure. They not only increase the number of factors available for analysis, but increase the computational burden as well. Furthermore, any re-e' aluation of a network due to changes in the survival rates of any node or arc further adds to the workload.

Organization of Research

The objective of this thesis is to improve the present analysis of stochastic networks by introducing a more efficient Monte Carlo simulation procedure, variance and bias reduction techniques, and response surface methodology (RSM). Accomplishing this task requires more than an efficient rewrite of current simulation programs, however. It also requires original research into *stochastic network performance as it relates to variance and bias reduction, and RSM.* This research is conducted in two major areas.

Simulation. The current simulation technique is to embed a standard path-augmenting maximal flow algorithm in a simple Monte Carlo simulation with sample sizes up to 100,000. (Marsh, 1988). The thesis offers a new simulation code using the same Monte Carlo technique but with two specific improvements. First, a unique maximal flow algorithm using minimal cuts instead of augmented paths attempts to reduce the time of calculation for each sample. Second, variance reduction using internal control variates and antithetic random numbers tries to reduce the number of replications required for a given confidence interval as well as lessen its bias. As the next chapter points out, there are no published experimental results of either technique. Yet, if either technique is successful, an improvement in simulation efficiency will be realized. The final code incorporates the successful techniques.

Additionally, the code is designed for portability. In other words, it should run on any computer with a ANSI-standard FORTRAN 77 compiler to the extent that such portability can actually be achieved. This is particularly important due to the number of different machines DOD has to run the program. Because writing the required interface with another program that determines the network component survivability is beyond the scope of this thesis, the code for this study will only provide a rudimentary interface for manual parameter input and simulation output. FORTRAN 77 is selected as the programming language because of its widespread availability (Marsh; 1988b).

Response Surface Methodology. This second aspect of the thesis is perhaps its most promising feature. Following the techniques of RSM as described by Box and Draper (1987), a well designed experiment of simulation output and the resulting response surface equation offers the following advantages:

1. The functional relationship of the network components to the maximum flow is described in a

first- or second-order polynomial equation. Furthermore, the metamodel's coefficients are a direct measure of the expected maximum flow's sensitivity to changes in network parameters.

2. A well screened model finds all significant relationships between network components and the expected maximum flow, including any interactions.

3. Once the response surface model is found, it is no longer necessary to run the simulation model. This is an important feature if the original network model is large or if repeated analysis of the network is expected.

4. The response surface model not only supports network optimization, but provides a clear algebraic or graphic description of the network's flow and how individual components contribute to its performance.

5. The resulting polynomial equation is easily incorporated in other models for further analysis or optimization (e.g. cost minimization, spreadsheet).

Thesis Objectives

In order to provide a more efficient method of analyzing the improvement of stochastic network performance, this thesis considers the following topics:

1. Why a minimal cut-set based algorithm would be more efficient than the standard path-augmenting algorithm in finding maximal flow and network reliability within the context of Monte Carlo simulation.

2. Does a simple function of the arcs or nodes exist such that it could be used as an internal control variate?

3. What insight does RSM offer for stochastic network performance and sensitivity?

Chapter II formally defines network and simulation terminology and concepts, and reviews current research in these areas. Chapter III describes in greater detail the new simulation program, and how experimental design and RSM will be implemented. Chapter IV gives the results of the research questions listed above. Chapter V summarizes the thesis and offers suggestions for further research.

II. Literature Review

The literature review covers the two principal areas of research interest - network reliability and maximal flow, and the simulation topics of experimental design and variance reduction.

Networks

Definition. Network modelling is a subset of a field of study referred to in the literature as *graph theory*, of which several disciplines, including operations research, applied and theoretical mathematics, and electrical engineering, all share an active interest and conduct ongoing research. There is an overwhelming choice of references in the literature to use for definitions, but most of those in the field of operations research refer to a seminal work by Ford and Fulkerson called Flows In Networks (1962). This thesis also uses their work as its principal reference, with additional descriptions and definitions provided by Chachra and others (1979).

One additional point about these definitions needs to be made. The literature is occasionally inconsistent in distinguishing the terminology used for networks and graphs. Chachra's introduction and summary of definitions is excellent in this regard; thus, its choice as a reference. However, some of his terms used by this thesis are, by strict definition, for graphs. But because the distinguishing feature of networks, directed edges, doesn't change the essential concept of the following definitions, this study adapts his terms for use in the network context.

A graph $G = (V, E)$ is defined as the set of points V, or vertices, connected by the set of vertex pairs E, or edges (Chachra and others, 1979:40). This definition is refined by Ford and Fulkerson to describe the condition where the edges E acquire a specific orientation or direction. In that case G becomes (N, A), a directed linear graph or *network*, where the set N, or *nodes*, are connected by directed edges A, or *arcs*. Furthermore, each node N_i and arc A_i in G can have a non-negative, real number associated with it representing maximum steady-state flow capacity per unit time (1962:2-4).

Returning to Chachra, *adjacent arcs* are two arcs with one node in common, while *adjacent nodes* are two nodes connected by one arc. The number of adjoining arcs of a node is the *degree* of that node. If an arc is incident with only one node (i.e. it starts and ends at the same node), it is called a *loop*. If two arcs share the same nodes at both endpoints and have the same direction, they are *strictly parallel arcs*.

In any network, an *arc sequence* is a bounded series of adjacent arcs from node N_O to node N_N, in the direction of N_O to N_N, which can contain a non-distinct subset of nodes N_s.

If all arcs in an arc sequence are distinct or unique, then that sequence is called a *path*, and if all nodes *N* in the path are distinct the path is called a *simple path*. By contrast, if nodes N_0 and N_N of a path are equal, then the path is referred to as a *closed path* or *cycle*.

If a network contains no cycles, it is called an *acyclic network*. A network that contains neither loops or parallel arcs is a *simple, directed network*. A *planar* network is one that can be set in a 2 dimensional plane such that all arcs cross only at the nodes. A connected graph, or network, exists if every pair of nodes is connected by a simple path. A subgraph, or *subnetwork* G_1, is a graph or network completely contained in *G* (1979:Ch 1).

Finally, Ford and Fulkerson cover the concepts of source and sink nodes. For any two distinct nodes S and T, if the static flow from S equals the flow into T, and for all intermediate nodes the static flow in equals the static flow out, then S is referred to as the *source node* and T the *sink node* (1962:4). For this thesis, however, a more narrow definition is used. In all networks, the source node S is defined as a node whose adjacent arcs are oriented such that all flow moves away from it, and a sink node T where its adjacent arcs direct all flow into it.

Additionally, a network may also contain multiple source nodes *S* or sink nodes *T*, or both (Ford and Fulkerson, 1962:1-

5). Accordingly, this study will allow for any combination of single or multiple source and sink nodes in networks with single commodity flow. Furthermore, the networks in this thesis are restricted to simple, acyclic, directed networks, can contain capacitated nodes, and can be non-planar. (The above definitions cover the major concepts of network theory necessary to understanding this thesis' efforts. However, they constitute just a few of the terms used in the field. For further detailed explanations, the reader is directed to Jensen and Barnes (1980), and Harary (1972) in addition to the references cited above.)

Maximal Flow. Given the above definitions, the current measure of network performance is the maximal flow from *S* to *T* in the network *G*. The current method used in the Monte Carlo simulation (Marsh, 1988), the *labeling algorithm*, is the same one suggested by Ford and Fulkerson. A widely implemented routine, it is considered more efficient than an equivalent linear programming formulation (Hillier and Lieberman, 1986:305). From Ford and Fulkerson, the algorithm works as follows.

Two routines are used; the first one, Routine A, incorporates a labeling process while the second procedure, Routine B, handles the change in flow. Routine A essentially searches for a flow augmenting path from *S* to *T*, carrying enough information with it through its labeling process that if it finds an unlabeled path from *S* to *T*, Routine B increments the flow amount accordingly, then modifies the labels before returning to Routine A for another path search. If Routine A fails to find another *S*-*T* path, then the current flow from Routine B is the maximal flow (1962:17-22). Because of its popularity, there are many additional descriptions and computer implementations of the labeling algorithm. Further information and refinements are offered by Chachra and others, (1979:122-129), Jensen and Barnes (1980:154-164), Nijenhuis and Wilf (1975:148-151), and Hillier and Lieberman (1986:304-310).

There is an alternative method of finding the maximal *S*-*T* flow of *G*, Lowever, that uses *cuts* instead of paths. Ford and Fulkerson define any collection of arcs in (*N*,*A*) that separates *S* from *T* as a *disconnecting set D*. *D* is a *proper* disconnecting set if none of its proper subsets are themselves disconnecting sets. If this is the case, *D* is also a *cut*, and the capacity of cut *D* is the summation of the flow capacities of its component arcs. Then, using the concept of disconnecting sets, the max-flow min-cut theorem states: "For any network the maximal flow value from s to t is equal to the minimal cut capacity of all cuts separating s and t." (1962:10-15).

Before pursuing max-flow min-cut theorem any further, the concept and terminology of cuts needs to be clarified.

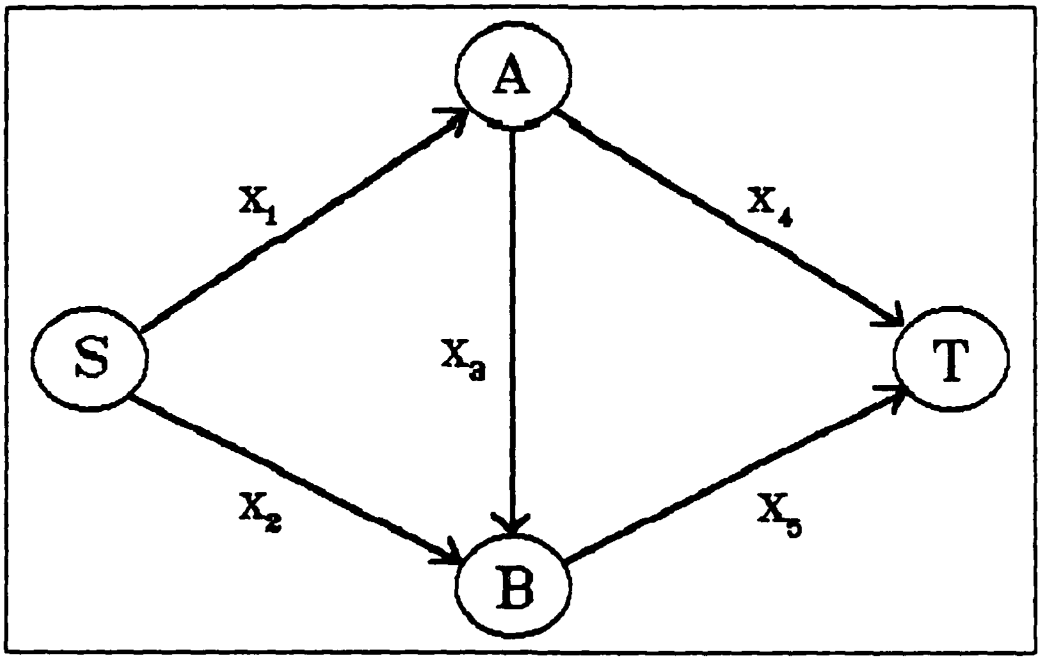

Figure 2-1. Example Lexicographical Network

The literature is somewhat inconsistent, which leads to misunderstanding cuts and the application of the max-flow min-cut theorem. Therefore, the following explanation, with reference to Figure 2-1, attempts to clarify this issue.

From the network shown above, the set of arcs *A* is (X_1,X_2,X_3,X_4,X_5). Using Ford and Fulkerson's terminology, out of 31 $(2^5 - 1)$ possible arc combinations, 14 form disconnecting sets. However, the total number of proper disconnecting sets is only 4; (X_1,X_2), (X_1,X_5), (X_4,X_5), and (X_2,X_3,X_4). In other words, each of these four proper disconnecting sets are contained within at least one of the

14 disconnecting sets, and no subset of any one of the four exists that still disconnects S and T. For instance, if any arc from the proper disconnecting set (X_2,X_3,X_4) is removed, a path between S and T becomes feasible. Also note that the proper disconnecting set (X_2,X_3,X_4) is a subset of disconnecting sets (X_1,X_2,X_3,X_4), (X_2,X_3,X_4,X_5), and (X_1,X_2,X_3,X_4,X_5). Finally, the cut capacity of (X_2,X_3,X_4), where C_N is the capacity of arc X_N, is $C_2 + C_3 + C_4$.

When referring to cuts, this thesis will use the terms given by Bellmore and Jensen. They refer to any disconnecting set as a cut, and any proper disconnecting set as a *proper cut*. For the interested reader, they also provide a slightly different explanation of proper cuts from the standpoint of graph subsets (1970:777) that is equivalent to Ford and Fulkerson's definition.

Continuing with maximum flow calculations, Ford and Fulkerson state that either cuts (disconnecting sets) or proper cuts (proper disconnecting sets) can be used as a "cut" in the max-flow min-cut algorithm (1962:15). Since the number of proper cuts can never be greater than cuts (Bellmore and Jensen, 1970:777), a better tactic is to use proper cuts. Therefore, instead of using the labeling algorithm or a linear programming formulation, this thesis will investigate the idea of generating all proper cuts, or the *proper* *cutset*, to calculate maximal flow using Ford and Fulkerson's max-flow min-cut theorem.

Pathsets vs. Cutsets. Using the proper cutset in the context of communication network reliability is discussed in detail by Bellmore and Jensen; in particular, one point especially pertinent to this study is covered. Specifically, it is how can one determine from a graph or network *G* which approach is more efficient - enumerating all simple paths necessary to run the labeling algorithm, or finding all proper cuts in order to use the max-flow min-cut algorithm. One answer is that, for networks having a single source and a single sink, where *N* represents the number of nodes and *M* is the number of arcs in *G*, the number of simple paths is bounded by 2^{M-N+1} and the number of proper cuts bounded by 2^{N-2}. Therefore, enumerating the proper cutsets can a better approach if $2^{N-2} \leq 2^{M-N+1}$ (1970:778).

Based on this formula, it would initially appear that for sparse networks, pathset enumeration is a more practical solution. For example, a network containing 20 nodes and 30 arcs would have an upper bound on the number of simple paths of $2^{30-20+1}$, or 2048, whereas the upper bound on the number of proper cuts is 2^{30-2}, or 268,435,456. However, there are two points that argue in favor of the proper cutset approach.

First, these are *upper* bounds on the number of simple paths or proper cuts; in theory the proper cutset could be closer in number to the simple paths. Second, and most importantly, the relative merits of paths versus cuts discussed in the literature is usually in the context of an analytic methodology. Instead, the author contends that a proper cutset, even one significantly larger than its simple path counterpart, can be implemented more efficiently in a Monte Carlo simulation than the Ford-Fulkerson labeling algorithm. This idea, and its implementation, will be more fully explained in Chapter III.

Another reason to consider cutsets is when a multiple source or multiple sink node network is modeled. As Figure 2-2 below shows, the degree of the network no longer predicts the number of cuts with respect to the paths. For instance,

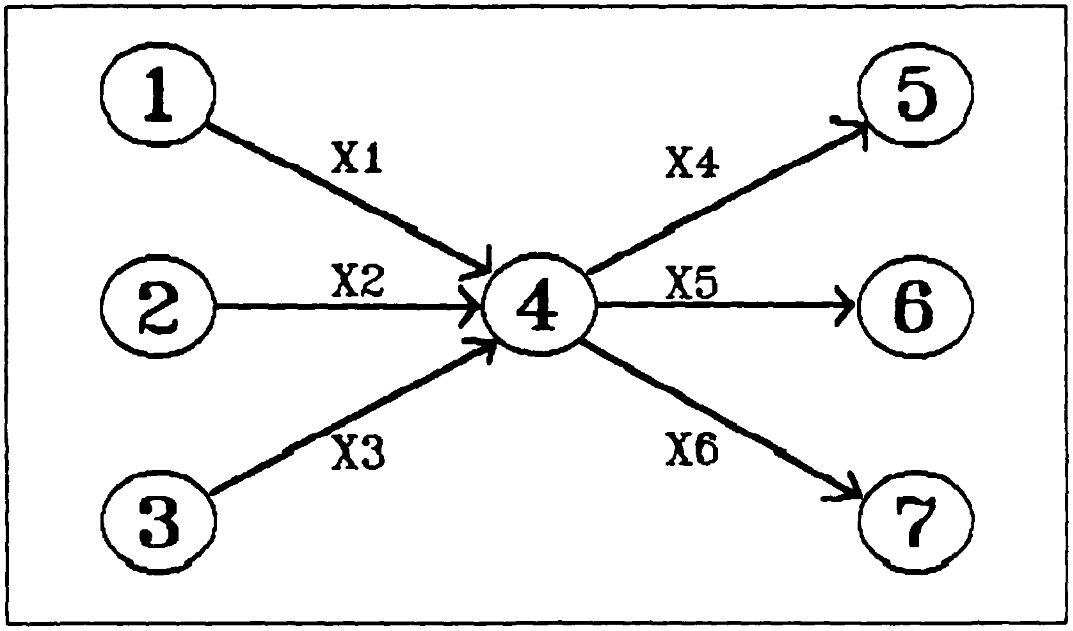

Fig. 2-2. Multi-Terminal Network

in this example there are 9 possible paths: (X_1, X_4), (X_1, X_5), (X_1, X_6), (X_2, X_4), (X_2, X_5), (X_2, X_6), (X_3, X_4), (X_3, X_5), and (X_3, X_6). Yet, there are only 2 cuts: (X_1, X_2, X_3) and (X_4, X_5, X_6). Clearly, in this situation, using the cutset for maximum flow calculations is easier.

Furthermore, adding additional source or sink nodes will increase the number of paths multiple times, but not add to the cutset. For example, adding a fourth source node with an arc (X_7) straight from it to the intermediate node 4 will not increase the number of cuts; X_7 is merely added to the first cut. As for paths, however, adding X_7 increases the number of paths to 12 (4 x 3).

Conversely, if the eighth node is added as an intermediary node, then the number of cuts increases while the number of paths remain the same. For instance, assume a node No. 8 were inserted between nodes 1 and 4, with arc X_1 connecting nodes 1 and 8, and a new arc, X_7 connecting nodes 8 and 4. The additional arc X_7 would become part of the existing path from node 1 to node 4, but the number of cuts would increase to 3: (X_1, X_2, X_3), (X_7, X_2, X_3), and (X_4, X_5, X_6). Further additions of intermediary nodes in this fashion would then increase the number of nodes *exponentially*.

The placement of the nodes is important because the networks modeled in this thesis resemble the basic topology in Figure 2-2. Generally, there are fewer intermediary nodes of the type just discussed, which further encourages the idea of using proper cuts instead of paths in the min-cut max-flow algorithm. However, the algorithm this study uses is sensitive to the number of intermediary nodes; thus, in certain situations the labeling algorithm may be a better choice. A procedure for making this choice is a good topic for further research.

Stochastic Networks. As in the case of network definitions, the terminology in the literature concerning stochastic networks is inconsistent. Therefore a brief summary of modeling efforts and definitions of stochastic network terms is offered.

A *stochastic activity network* represents networks used for management scheduling of large projects where the completion times are stochastic (Bauer and others, 1988b:1). Research in this area on estimating path and arc probabilities has been done by Fishman (1985), and on variance reduction by Fishman (1983b), Bauer and others (1988b), and Burt and Garman (1971). However, the current effort concerns maximal flow and network reliability, rendering this area of network research less relevant.

The probabilistic network more applicable to this thesis is a *stochastic binary network* (SBS). Ball defines an SBS to be "a system that fails randomly as a function of the random failure of its components...(where) each component may take on either of two states: operative or failed and that the states of any two components are independent" (1980:154). Furthermore, he defines a *stochastic coherent binary network* (SCBS) as one where the pathset defines the minimal subset required for system operation and the cutset defines the minimal subset required for failure (1980:154). The networks this thesis addresses fit Ball's definition with one exception: component failure is *not* necessarily independent. However, failure dependencies among network components is easily implemented in a Monte Carlo simulation, so this caveat is trivial.

Before proceeding, two points need to be made. First, a closely related aspect to SCBS networks is the concept of *reliability*. Terms used to describe this concept varies in the literature, with *terminal reliability* and *S-T connectedness* among the more popular versions. Although they essentially mean the same thing, this thesis will use the term reliability to mean the probability of at least one path connecting *S* and *T* in a SCBS, or alternatively as the probability of all components of the cutset failing (Bellmore and Jensen, 1970:778).

The second point is that, given the above definition of reliability, the SCBS formulation provides a splendid complement to the max-flow min-cut theorem. The key feature is that the network cutset can be used to estimate both network reliability and maximal flow in the same simulation. Furthermore, as Chapter III shows, the computational cost of adding a reliability estimator to the max flow estimator is almost trivial.

At this point the distinction between SBS and SCBS formulations and another class of stochastic networks best described by the author as *randomly-capacitated networks* (RCN) needs to be made. In an RCN, arc *capacity* varies over a range of values as a continuous function of a probability distribution. Arc capacity in a SBS/SCBS network, by contrast, is based solely on the binary (operative-failed) status of the arc; if the arc is operative, there is only one arc capacity. The networks investigated by this study are not part of the RCN category of stochastic networks - they belong in the SCBS class. RCN systems are mentioned because much research has been devoted to them, and it's important to understand the difference between the two models. For further explanation or research results in this class of networks, see Fishman (1987a), Somers (1982), and Evans (1976).

In either RCN or SCBS structures, the difficulty of assessing network reliability in an analytical form is well known. Ball summarizes the computational difficulty of such calculations, proving that most network reliability issues fall in the class of NP-hard combinatorial problems; i.e., no polynomial bounded algorithm exists (1980). Progress in the area of network reliability has focused on special stochastic network structures research by Shamir (1979), Rosenthal (1977), Agrawal and Satyanairayana (1984), and Agrawal and Barlow (1984); approximating techniques by Wallace (1987) and Ball (1978), and Monte Carlo simulations by Fishman (1987b). This last area is most relevant to the thesis and deserves further explanation.

Fishman (1986) gives an excellent overview of Monte Carlo methods in estimating network reliability. His article explains four ways to calculate network reliability for an undirected graph version of a SCBS; dagger sampling by Kumato and others (1980), sequential destruction by Easton and others (1980), bounds estimation by Fishman (unpublished), and estimation based on failure sets by Karp and Luby (1983). The last technique and source, as Fishman explains it, uses failure sets, or equivalently cutsets, to estimate the graph's reliability, and is most closely related to this study's methodology. However, instead of sampling the entire cutset as this thesis proposes, Karp and Luby's Monte Carlo simulation procedure repeatedly samples single, randomly selected cuts *K* times to determine network reliability. (Fishman, 1986). This is an interesting approach, but because the max-flow min-cut algorithm requires evaluation of the *entire* cutset, Karp and Luby's sampling technique is not applicable.

The literature search found only two articles, both by Fishman, that deal with Monte Carlo estimation of maximal flow on a network. The first paper develops an algorithm that offers both computational efficiency and reduced variance of an unbiased estimator of maximal flow. However, he models only randomly decreasing arc capacities instead of nodes, using a cumulative process that describes the arc deterioration as normally distributed (Fishman, 1987a). In short, his algorithm applies to RCN formulations instead of a SCBS structure.

The second Fishman paper is more closely related to this study's efforts. It combines two methods of importance sampling (see Simulation Topics below) in a Monte Carlo simulation to reduce the variance of the reliability estimators of communication networks typically described by an SCBS (1987b). However, this thesis is investigating the effect of control variates, not importance sampling, in variance reduction. Nonetheless, Fishman provides a proven approach to reducing the variance of the estimator; a comparison of the two variance reduction techniques would be a very interesting continuation of this research.

Simulation Topics

Definitions. Law and Kelton define Monte Carlo simulation as "... a scheme employing random numbers, that is U(0,1) random numbers [uniform distribution from 0 to 1], which is used for solving certain stochastic or deterministic problems where the passage of time plays no substantive role" (1982:49). Monte Carlo simulation is widely used to solve analytically intractable problems or as an approximating method for NP-hard problems. Hence, its appeal for estimating stochastic network performance.

An important feature of Monte Carlo simulation is how to improve the statistical output of the simulation beyond what's available from simple or *crude* sampling. Kleijnen describes six techniques available for variance reduction in Monte Carlo simulations:

1. *Stratified Sampling*, where the simulation response is weighted based on which *strata* the random numbers belong to.

2. *Importance Sampling* uses distortion of original input variable distribution, where the response later adjusts for the bias (As mentioned earlier, Fishman shows this technique can be used in calculating SCBS networks).

3. *Selective Sampling* is where input variables are sampled according to their expected frequency of occurrence.

4. *Common Random Numbers* use the same stream of pseudorandom numbers to analyze two or more systems or system variable.

5. *Antithetic Variates* use negative correlation induced by two runs, one using *R* random numbers, the other 1-*R* random numbers.

6. *Control Variates* regress out effects of a variable having both a known expectation and correlation with the response.

(1974:Ch III). Additional explanations of Monte Carlo simulation and variance reduction techniques are also offered by Hammersley and Handscomb (1964), and Law and Kelton (1982:Ch 11).

This thesis will investigate the last two techniques - antithetic variates and control variates. The antithetic technique, known as the *assignment rule* or *correlation induction strategy*, is implemented through assigning a common random number stream or its antithetic counterpart at the experimental design points (Schruben and Margolin, 1978). The theory behind it, and the combined effect of it and control variates on variance reduction of the simulation response, is covered shortly. The principal focus of this research is on the effectiveness of the second variance reduction technique, control variates. As a further refinement only internal control variates will be used (Law and Kelton, 1982:359).

Control Variates. One of the better explanations of using internal control variates in Monte Carlo simulations is given by Lavenberg and Welch. The following is a summary of their presentation.

Let μ be an unknown quantity whose unbiased estimator Y is the result of a single Monte Carlo simulation ($\mu = E[Y]$).

If the expectation μ_o of a random variable C is both known and correlated with Y, then C is a control variable that can help calculate an unbiased estimator of μ whose variance is smaller than Y. Therefore, for any constant α

$$Y(\alpha) = Y - \alpha(C - \mu_o) \qquad (2.1)$$

is an unbiased estimator of μ. Furthermore,

$$\mathrm{Var}[Y(\alpha)] = \mathrm{Var}(Y) + \alpha^2\mathrm{Var}(C) - 2\alpha\mathrm{Cov}(Y,C) \qquad (2.2)$$

From Eq (2.2), it can be shown Y(α) has a smaller variance than Y if

$$2\alpha\mathrm{Cov}(Y,C) > \alpha^2\mathrm{Cov}(Y,C) \qquad (2.3)$$

Continuing, the value of A which minimizes the variance of the estimator Y(α) is

$$A = \mathrm{Cov}(Y,C) / \mathrm{Var}(C) \qquad (2.4)$$

for α = A we find

$$\mathrm{Var}[Y(A)] = \mathrm{Var}(Y) - \frac{[\mathrm{Cov}(Y,C)]^2}{\mathrm{Var}(C)} = (1 - p^2YC)\mathrm{Var}(Y) \qquad (2.5)$$

where pYC is the correlation between Y and C. Therefore, if Y and C are correlated at all, there will be some reduction of variance over the old estimator Y (1981). Similar explanations can be found in Lavenberg and others (1982), Law and Kelton (1982:360), and Bauer (1987:Ch 2).

The control variate technique applies in the network simulation as follows. It stands to reason that certain functions of surviving nodes would be correlated to the resultant maximal flow (Bauer, 1988a). If that is the case, and given that we already know the probability of node survival, then it stands to reason that some function of the nodes meets the definition of control variables and theoretically can be used in maximizing the variance reduction.

The multiple control version of Eq (2.1) is also available, but for a couple of reasons this thesis will only explore scalar control variates. First, a search of the literature reveals that *no simulation experiments* have been conducted to explore the concept of variance reduction in stochastic networks as applied to expected maximum flow. Therefore, it is reasonable to first start with a scalar control variate. Second, multiple controls generally reduce the efficiency of the variance reduction due to the necessity of estimating the vector version of α (Bauer, 1987:14; Lavenberg and others, 1982:184). Consequently, a scalar control should show a significant variance reduction before moving to the multiple control stage. Further explanations of multiple control variates are given by Lavenberg and Welch (1981), Lavenberg and others (1982), Bauer (1987), and Rubinstein and Marcus (1985).

RSM and Experimental Design. The objective of experimental design and RSM is to express the simulation response "...as a function (a first or second degree polynomial) of the independent variables" (Kleijnen, 1974:79). Hence, this thesis will use experimental design and linear regression on one network of particular interest in order to gain more insight into network performance and establish the methodology for future analysis. Since only two network parameters can be improved - component capacity and probability - these two provide the only *types* of independent variables in the experimental design. Unfortunately, almost every component of a network contains both, leaving the number of potential factors for the experimental design in the hundreds. Therefore, a combination of user knowledge of the network, and group and factor screening designs will be necessary to reduce the full factorial design to a manageable size.

A search of the published literature failed to find any research of response surfaces and stochastic networks to base this study on or compare results with. The following sources provided the guidance for conducting the experimental design and response surface analysis: Kleijnen (1974), Box and Draper (1987), and Montgomery (1984).

Antithetic Variates. The technique of antithetic variate reduction investigated by this thesis, the Schruben-Margolin correlation induction strategy or assignment rule, is straight-forward. Rather than running either independent random number streams or the same common random number stream R at all design points (where R is the set of random numbers $r_1,r_2,...,r_i$), they assign R's antithetic counterpart A (where A is the set of random numbers $1-r_1,1-r_2,...,1-r_i$) to half of the experimental design points based on an orthogonal blocking strategy. In other words, the design matrix is divided into two orthogonal blocks (i.e. a one-half fractional design blocked on a higher order interaction). From there, all design points in one block are assigned the random number stream R, and the design points in the other block are assigned the antithetic number stream A (1978:507-514).

Incorporating this method in the experimental design should reduce the variance of the response surface, thus giving a more accurate estimate of the response. For example, if the random number stream R turns out an artificially high or low estimate of the maximum flow or reliability, all observations of the experimental region will be biased high or low. By using antithetic streams at blocked design points, that bias should be countered in the opposite direction. However, Schruben and Margolin's strategy requires the following assumptions for variance reductio to be valid:

1. Zero correlation exists between two observations using d:fferent independent random number streams.

2. A positive correlation exits between the responses of any two distinct design points with the same random number stream R or A.

3. A negative correlation exits between the responses of any two distinct points with one point using R and the other using A random number streams.

4. The simulation has equal variance across the region of interest.

(1978:508). Unfortunately, the size of the experimental design precludes this analysis from offering a complete proof of Assumptions (2) and (3); instead, only empirical evidence is offered.

One final question posed by using Schruben and Margolin's assignment rule is this: Does combining their correlation induction strategy with control variates offer a better estimator of network performance? This *combined strategy* is the subject of a recent paper by Tew and Wilson (1987), where they compare the combined strategy to independent random number streams, common random numbers, the assignment rule, and control variates, and develop a methodology for determining the superiority of the combined method (1987:415). This thesis' objective of investigating SCBS scalar control variates and response surfaces precludes a thorough investigation of this area. However, the methodology of this study conducts a comparison of common random numbers, independent random numbers, and the assignment rule in an example experimental design to empirically determine the best approach for the larger networks.

III. METHODOLOGY

This chapter covers two important areas of the thesis' methodology - Simulation Code and Experimental Design. In the Simulation Code section, the logic and algorithms used to implement the Monte Carlo simulation in the FORTRAN code called MAXFLO are explained in detail. Also, the procedures and tests used to verify MAXFLO are also described. The Experimental Design section covers the selection of control variates and the experimental design used for developing the response surface equations. Finally, in the Example Problem section, a simple network problem is offered as an illustrative example of the methodology.

Simulation Code

The purpose of this section is not to describe, line-by-line, every function and nuance of MAXFLO. For that, the reader is referred to the source code and in-line comments in Appendix E. Rather, it is to cover the theoretical principle of proper cutsets and proper cutset generation, and their advantages in SCBS simulation; random number generation; and MAXFLO verification.

Cutsets and Simulation. Referring to Figure 3-1 on the following page, the proper pathset and cutset can be represented in computer memory in two-dimensional arrays, or lexicographically (in matrix form) as shown in Tables 3-1 and 3-2.

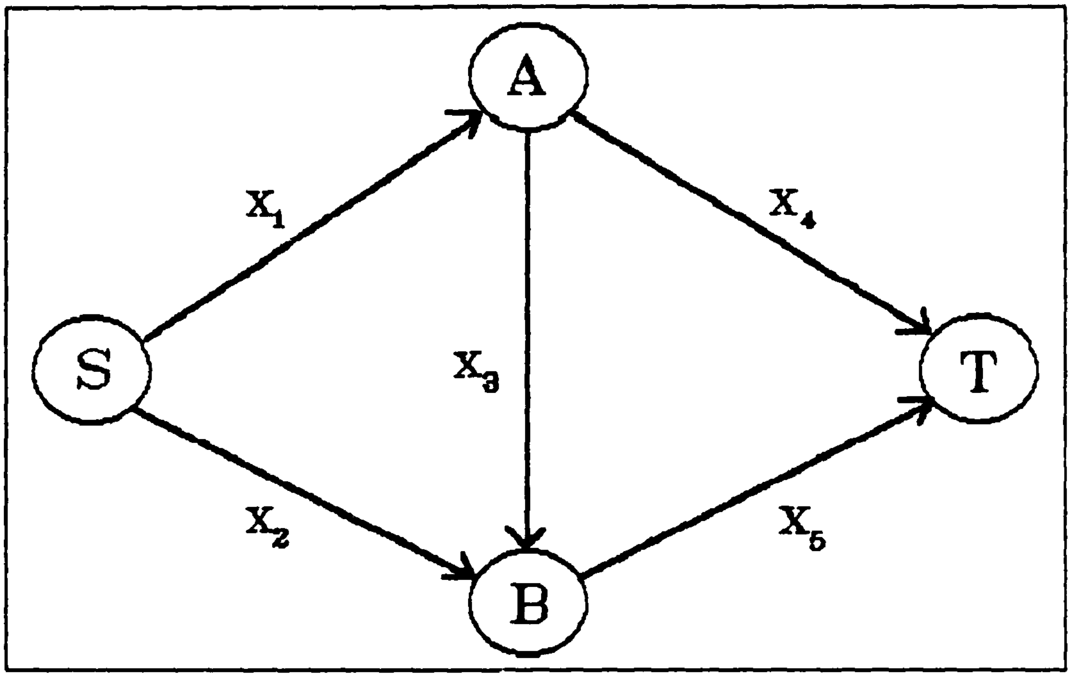

Figure 3-1. Example Lexicographical Network

In Tables 3-1 and 3-2 X_N represents the capacity of that arc as shown in Figure 3-1, the columns are individual arcs whose elements are arc capacity, and rows are individual paths or proper cuts. (Note that there exists for each column an unique X_N.) Once the cutset matrix is determined, the maximum flow of this network can be found by taking the minimum of the summation of each row's elements as postulated by the max-flow min-cut theorem of Ford and Fulkerson (1962:11).

This lexicographical representation highlights one key insight of the proper cutset matrix: *Changes in individual*

TABLE 3-1 Pathsets of Figure 3-1 Network

	ARCS				
Path #	S to A	S to B	A to B	A to T	B to T
1	X1		X3		X5
2	X1			X4	
3		X2			X5

TABLE 3-2 Proper Cutsets of Figure 3-1 Network

	ARCS				
Cut #	S to A	S to B	A to B	A to T	B to T
1	X1	X2			
2	X1				X5
3		X2	X3	X4	
4				X4	X5

arc capacities affect only the elements of the appropriate column; thus the composition of the proper cutset (i.e. the placement of elements in the matrix) is not a function of network parameters, but of network topology. In other words, once the proper cutset matrix is found, variations in maximum flow due to changes in X_N can be calculated using the same matrix.

This insight can be taken one step further when considering the lower bound of X_N. According to Ford and Fulkerson, where X_N is any real number greater than or *equal to* 0, the max-flow min-cut theorem still applies (1962:22). Therefore, where variations in X_N include 0, the proper cutset matrix remains valid for maximum flow calculations. This leads to a second key insight: *Loss of an arc (or node) in a SCBS is equivalent to setting the respective arc (and incident arcs) capacity X_N in the proper cutset matrix to 0.* Therefore, a Monte Carlo simulation using proper cutsets can, for each replication, simply substitute 0 for those X_N whose respective arcs are simulated to have been lost.

There are two additional simplifications related to the characteristics of SCBS networks to take advantage of in the simulation algorithm. First, because arc capacity can only be either X_N or 0, an equivalent procedure to replacing X_N with 0 is to simply ignore the column representing the failed arc in the current replication's maximum flow calculation. (Remember that each column N represents arc N with capacity X_N unique to that column.) This is accomplished by using a one dimensional array representing the status of arcs based on the current replication's comparison of random number draws and the individual arcs' probability of survival. This *state vector* is used by the maximum flow calculation routine in deciding which columns in the cutset matrix to ignore in the current replication.

The second simplification is more accurately described as taking advantage of a characteristic of the max-flow min-cut algorithm that especially manifests itself in a SCBS network. Simply stated, once the value of *any* proper cut in the matrix is found to be zero, there is no point in calculating the remaining cuts' values. This is because the max-flow min-cut algorithm search is for the minimal value of all proper cuts, which can be no lower than zero. A SCBS amplifies this effect since, again, it's arcs' capacities can only be X_N or 0, thus increasing the number of proper cuts whose values will be zero in any given replication.

Implementing this second advantage is easy. Each replication finds the maximum flow by going through the cutset matrix and calculating every proper cut's value. During this procedure, the current cut value is compared to the minimal value found from the preceding cuts calculated thus far. If the comparison shows the current cut's value lower than the current minimal value, it replaces the minimal value used for subsequent comparisons. Then, after the last proper cut value is compared, the final minimal value will be the maximum flow for that replication.

Now, if the comparison routine described above also checked for and found the current proper cut's value to be zero, the replication could be terminated at that point. As pointed out, this additional check increases the efficiency of the simulation by avoiding the need to go through the entire cutset. A further refinement would be to sort the cutset matrix by row according to probability of failure in order to minimize the average number of proper cuts the replication goes through before finding a summation value of zero. However, this additional feature is not implemented in MAXFLO.

At this point, the representation of *capacitated* nodes in proper cutsets should be covered. Fortunately, the networks analyzed in this study do not contain capacitated nodes; though, since the possibility exists, MAXFLO can model them in the manner about to be described. But because capacitated nodes can adversely affect the number of proper cutsets, the reader should be aware of this facet of network theory. Therefore, the situations where, and the degree to which, the number of proper cuts differs from the number of simple paths needs further explanation.

Again, Ford and Fulkerson provide a solution by simply treating the capacity of the node as another arc (1962:24). For instance, if in Figure 3-1 node A contained an internal capacity X_A, then the resulting pathset and proper cutset would include an additional 'arc' A to A' as shown in Tables 3-3 and 3-4, respectively. Notice that the number of paths in Table 3-3 didn't change from Table 3-1. There are still three paths, with Paths 1 and 2 picking up the extra 'arc' A to A' that represents node A's internal capacity. This makes intuitive sense because a node 'arc' only adds a capacity constraint to an existing path. It cannot provide any additional choice in direction or branching since it does not

TABLE 3-3 Pathsets of Figure 3-1 Network
With Node A Capacity

Path #	ARCS					
	S to A	S to B	A to A'	A' to B	A' to T	B to T
1	X1		XA		X4	
2	X1		XA	X3		X5
3		X2				X5

TABLE 3-4 Cutsets of Figure 3-1 Network
With Node A Capacity

Cut #	ARCS					
	S to A	S to B	A to A'	A' to B	A' to T	B to T
1	X1	X2				
2		X2	XA			
3	X1					X5
4			XA			X5
5		X2		X3	X4	
6					X4	X5

connect to a distinct second node as arcs are normally defined to do.

However, the proper cutset matrix in Table 3-4 is different from the one in Table 3-2. Not only is an additional column (A to A') added, but two additional proper cuts are created as well. Again, this makes intuitive sense since proper cutsets are partially a function of the number of arcs available to form them, including the internal 'arcs' of node capacities. Put another way, the additional capacity constraint of a node has to be accounted for in the proper cutset since it can theoretically be the limiting factor in the network's maximum flow. Ford and Fulkerson show, however, that the max-flow min-cut theorem still applies to networks with capacitated nodes (1962:25).

A major question arising from expanded proper cutsets due to capacitated nodes is how detrimental this characteristic is to the efficiency of this study's simulation methodology. In a *simulation* context there should be considerable computational advantages of matrix row addition and the zero lower bound limit versus the labeling/pathset algorithm; yet, such efficiencies could be offset by a substantially larger proper cutset over the simple pathset.

Recall from Chapter II that for single terminal networks, where N is the number of nodes and M is the number of arcs in an uncapacitated network G, the number of simple paths is bounded by 2^{M-N+1} and the number of proper cuts by 2^{N-2} (Jensen and Bellmore, 1970:778). If it is assumed that all nodes in the network are capacitated, then there are $2N$ nodes to consider, giving a theoretical bound of 2^{2N-2}. Yet, since the additional 'arcs' create no new paths the upper bound remains 2^{M-N+1}. Therefore, the potential number of proper cuts versus simple paths is much higher in a capacitated network. Nonetheless, it stands to reason that the new upper bound of 2^{2N-2} for the cutsets is seldom realized for the same reason that no new paths are created. Simply stated, the combinatoric possibilities are somewhat limited for the new 'arcs' since they do not provide the additional paths needed to derive proper cutsets or to approach the theoretical upper bound.

The situation changes somewhat in the case of multiple terminal networks. In these situations, capacitated nodes can drastically increase the number of cutsets. The network in Figure 2-2, repeated here in Figure 3-3, illustrates this point.

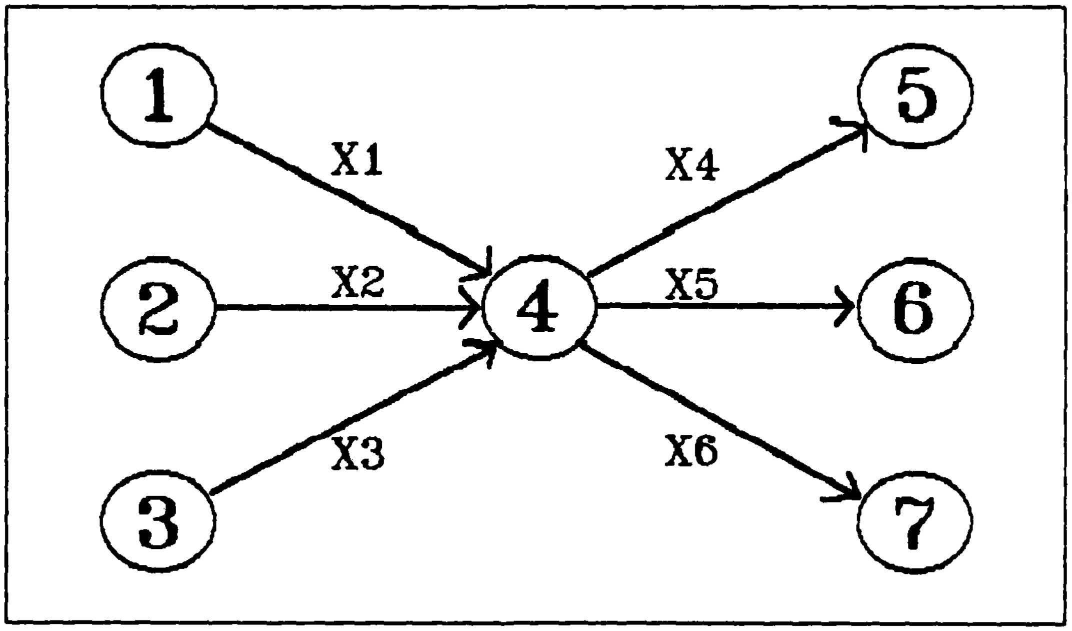

Figure 3-2. Multi-Terminal Network

Currently, there exists only 2 proper cuts in the non-capacitated node version of Figure 3-2 - (X_1,X_2,X_3) and (X_4,X_5,X_6); and 9 paths - (X_1,X_4), (X_1,X_5), (X_1,X_6), (X_2,X_4), (X_2,X_5), (X_2,X_6), (X_3,X_4), (X_3,X_5), (X_3,X_6). Now, assume that nodes 1 through 4 take on capacities that are modeled as internal 'arcs', and referred to as 'arcs' X_{11}, X_{21}, X_{31}, and X_{41}, respectively. In this situation, *10* proper cuts now exist - (X_1,X_2,X_3), (X_{11},X_2,X_3), (X_1,X_{21},X_3), (X_{11},X_{21},X_3), (X_1,X_2,X_{31}), (X_{11},X_2,X_{31}), (X_1,X_{21},X_{31}), (X_{11},X_{21},X_{31}), (X_4,X_5,X_6), and (X_{41}); but, the number of paths remains the same. The impact is clear: Capacitated terminal nodes *exponentially* increase the number of arcs. The effect is similar to the one resulting from adding intermediary nodes to the arcs as described in Chapter II.

Again, the networks in this study contain non-capacitated nodes with topologies resembling Figure 3-2 more than Figure 3-1; hence, the concept of using proper cuts in calculating maximum flow. The extent to which the proper cutset differs in size from the pathset cannot always be manually determined, nor can the effect this difference has on simulation efficiency be predicted. This study will provide some answers, but the definitive answer is beyond the scope of this text and best left to future research.

Finally, a few comments about the cutset algorithm and its implementation in MAXFLO. All nodes are represented as numeric integers, with the source node S starting at 1 and the sink node T equal to the total number of nodes for single terminal networks. (Hence, no integer between 1 and the total number of nodes may be skipped.) In the case of multiple source nodes, node number 1 is reserved as a *dummy* single source node, and the node numbers immediately following 1 are reserved for the actual source nodes. In a similar manner for multiple sink nodes, the last numbers are reserved for the sink nodes, and an additional number is created for a dummy single sink node. Furthermore, *dummy arcs* from the dummy single source node to the actual multiple source nodes, and from the multiple sink nodes to the dummy sink node, are required. This type of input is awkward, but allows for a faster generation of all simple paths.

Arcs are referred to by the source node integer called the *Tail* and the destination node integer called the *Head*. All node and arc capacities are represented as integer values, while their probabilities of survival are stored as positive real numbers between 0 and 1. This procedure is used because of its ease in programming the matrix representation of the pathset and proper cutset, and the state vector in the simulation. Furthermore, such integer depictions of the network do not require a sophisticated user interface.

A subroutine is also available to change the network parameters without having to re-enter the entire network. However, the nature of proper cutsets limits what kind of changes can be made before the network has to be re-entered and the proper cutset recalculated. In general, the rule is this: *No additional nodes or arcs can be added - only existing ones can be modified or taken away.* A few examples illustrate this rule.

For a non-capacitated node, entering 0 will retain the node for simulation purposes, but no internal 'arc' will be generated. However, that node may not take on future capacity - to do so will require the network to be completely re-entered and a new proper cutset calculated. On the other hand, if that node is initially entered with a capacity, that node must always retain some integer capacity. Capacitated nodes cannot be entered with zero capacity because of the way MAXFLO retains the cutset arrays. Instead, either an artificially low capacity must be entered or the probability of survival set to .00, to emulate a node with potential capacity. There are no restrictions on changing node probabilities of survival.

Arcs are somewhat more flexible. Again, no arc can be added to the network without recalculating the proper cutset. However, capacity can be changed, including the ability to reduce it to zero. Like nodes, there are no restrictions on changing arc probabilities of survival.

The inability to add nodes and arcs arises from the fact that adding components alters the network topology, thus requiring a recalculation of the proper cutset. This doesn't mean that a new network has to be entered every time a new arc or node addition is modeled, however. *The trick is to include all possible future nodes and arcs in the current network with their survival probabilities set to zero.* This way the cutset accounts for their potential presence, but the simulation will ignore their effect. Then, to 'add' one of the new components to the network, that component's survival probability is simply set to a value above zero.

Proper Cutset Generation. Using the cutset approach requires generating the proper cutset from the pathset. Once it is generated, an option is provided to save the cutset and its parameters, thus eliminating the need to regenerate the network cutset for future use. But it must be produced the first time to be used at all - and it turns out to be the most difficult subroutine in MAXFLO.

The difficulty lies in separating proper cuts from the larger class of cuts in a reasonable amount of time. At first glance, this appears to be a non-polynomial (NP) problem since the upper bound of proper cuts is known to be 2^{2N-2}. Fortunately, an algorithm by Shier and Whited (1985) provides a faster way of calculating proper cuts from the pathset.

The algorithm can best be described by an example network problem given in their article that is similar to Figure 3-1. From that network, the *path polynomial* is written as

$$X_1X_3X_5 + X_1X_4 + X_2X_5 \qquad (3.1)$$

where all arithmetic operators are Boolean. The *inverse polynomial* of Eq (3.1) is found by complementation to give

$$(X_1 + X_3 + X_5)(X_1 + X_4)(X_2 + X_5). \qquad (3.2)$$

Expanding Eq (3.2) out and deleting non-miminal elements will then give the *proper cutset polynomial*. For this example, expanding the first two terms in Eq (3.2) gives

$$(X_1 + X_1X_3 + X_1X_5 + X_1X_4 + X_3X_4 + X_4X_5)(X_2 + X_5), \qquad (3.3)$$

and expanding the remaining two terms of Eq (3.3) gives

$$\begin{aligned} &X_1X_2 + X_1X_2X_3 + X_1X_2X_5 + X_1X_2X_4 + X_2X_3X_4 + X_2X_4X_5 \\ &+ X_1X_5 + X_1X_3X_5 + X_1X_5 \quad + X_1X_4X_5 + X_3X_4X_5 + X_4X_5. \qquad (3.4) \end{aligned}$$

Since the first term X_1X_2 is contained in terms 2,3,4; the seventh term X_1X_5 in terms 8,9,10; and the twelfth term X_4X_5 in terms 6,11, Eq (3.4) is reduced to

$$X_1X_2 + X_2X_3X_4 + X_1X_5 + X_4X_5 \qquad (3.5)$$

which is the proper cutset polynomial (1985:315). Note that Eq (3.5) gives the same answer found in Table 3-2.

Shier and Whited also offer several modifications to the above algorithm that considerably improve its efficiency. These algorithms are incorporated in the MAXFLO cutset subroutine, but will not be explained in this chapter. (The reader is referred to their article for a detailed explanation.) They also report excellent computational results on networks approximately half the size of this study's networks; enough so to indicate that this algorithm is quite sufficient for identifying the proper cutset (1985:315-317). Additional references for cutset generation and network reliability are found in Provan and Ball (1984) and Bellmore and Jensen (1970).

Random Number Generator. A critical feature of any simulation is the correct generation of pseudo-random numbers. A detailed account of pseudo-random number generation is beyond the scope of this chapter; instead, the reader is referred to an excellent and detailed explanation of this topic by Law and Kelton (1982:Cha 3). What is pertinent is the author's implementation of a pseudo-random number generating function provided by Schrage (1979) as recommended by Law and Kelton (1982:227). This function requires a computer with a 32-bit word or larger and the NOOVERFLOW option activated on VMS FORTRAN compilers.

MAXFLO Verification. The term *verification* describes the procedure for determining whether a computer program correctly simulates the model, whereas in *validation* the objective is to ascertain if the model itself correctly reflects the actual system (Law and Kelton, 1982:337-338). This study assumes that systems exist which can actually be modelled this way; thus, validation will not be accomplished. This leaves the verification process, which is conducted as described below.

There are two essential features of MAXFLO to verify to insure the output results are correct - proper cutset generation and Monte Carlo simulation. The pathset and proper cutset algorithms were checked using several small networks ($N \leq 6$, $M \leq 15$) whose paths and cutsets were both exhaustively enumerated and graphically deduced. Additionally, a deterministic, 10-node 21-arc network from Jensen and Barnes (1980:148) was tested for pathset and cutset generation, and for maximal flow by setting all probabilities to 1.0. In all cases, pathset and cutset generation works correctly, as well as finding the same maximal flow of Jensen and Barnes' network.

This leads to the second verification task of confirming the Monte Carlo simulation output of stochastic networks. This is a more difficult because calculating the expected value of a SCBS in order to compare it to the simulation response's confidence interval is quite complex when $N \geq 6$. On the small test networks, the confidence intervals did contain the expected values, but as additional verification the following technique was employed.

A sample network of 6 capacitated nodes, 7 arcs, and 3 paths was developed by the author. This network's expected maximum flow was modeled on a spreadsheet to accommodate changes in 3 selected network parameters. Eight (2^3) runs were made, comparing the simulation response's confidence intervals to the spreadsheet calculations. The results are that all but one confidence interval contains the expected maximal flow. Since the confidence intervals' α was .05, this test gives no reason to doubt the simulation's accuracy. (A complete presentation of this project follows shortly.)

Experimental Design.

The purpose of this section is to describe the experimental design and procedures for finding the response surface equations for maximal flow and network reliability. Additionally, the procedure for selecting control variates is also discussed.

Screening Designs. The initial problem is finding those factors of the SCBS who have the greatest affect on network flow and reliability. As the Example Problem section shows, this isn't as easy or obvious as it first appears. For example, both component survival probabilities and capacities influence the expected maximum flow, providing N + 2M possible factors and requiring 2^{N+2M} experimental design points for a complete, 2-level factorial design. In the case of reliability, only the component parameter of survival probability affects network reliability, thus requiring $2^{(N+M)}$ design points. Obviously, in either case a reduction in the number of factors is necessary.

Part of the answer lies in reducing the number of network components under consideration for improvement. One way to do this is to consider only those arcs or nodes that can be realistically improved in survivability or capacity. It hardly makes sense to include in an experimental design parameters whose components cannot change.

Another way to reduce the number of parameters is to conduct a preliminary factor screening experiment based on the Plackett-Burman designs (1946:323). The principal reason for employing their designs are their small size and ability to detect mutually unaliased main effects (Box and Draper, 1987:162,506). From this initial screening, a reduced number of factors showing significant main effects will be used to form the full first-order fractional design.

The possibility of a second-order response model cannot be ruled out; hence, the methodology must also include procedures for determining the existence of second-order effects, and conducting a second-order experiment if necessary. Guidance for checking second-order effects comes from Montgomery (1984:449-450), and for conducting second-order designs from Montgomery (1984:462-470) and Box and Draper (1987:Ch 7).

Control Variate Selection. Chapter II describes the mathematics for scalar control variates used in MAXFLO; the current question is which scalar controls to investigate. Since this is a new area of research, Bauer (1988) offers as a general class of controls the *total number* of nodes that are up (or down) in a given subset. This control variate is an aggregate scalar measure of how many nodes in the subset are operative, and not a multivariate measure of the individual performance of multiple nodes. For purposes of clarity, this class of control variates is referred to as *survival variables*.

Recall that a high correlation of a control variate with the response results in a large variance reduction. Therefore, the objective is to begin with a class of controls that has a known expectation, and whose existence and effectiveness has the greatest effect on network performance. In the case of SCBS networks, survival variables best meet these requirements.

As a group, nodes generally have a greater influence on the network than arcs. This is because the loss of a node affects all arcs incident to it, and by extension any and all paths associated with those arcs. By contrast, the loss of an arc only affects those paths containing that arc. There are exceptions to this observation, the most obvious one being the case where all paths go through a single arc. But this exception rarely exists in the networks analyzed in this study, thus making survival variables a logical control to research.

This thesis restricts its research to the total number of operative nodes in a given subset as the scalar control variate. More specifically, this means that certain survival variables believed to be highly correlated to network maximum flow and reliability are identified by the analyst to MAXFLO as forming the control subset of interest. Mathematically, this idea is expressed as follows.

Because of the stochastic binary nature of the network, the random variable Y_I is defined as

$$Y_I = \begin{cases} 0 \text{ with probability of } P_I \\ 1 \text{ with probability of } 1 - P_I \end{cases} \qquad (3.6)$$

where P_I is the probability of survival (P_s) of component i. The control variate is defined as

$$SV = \sum_{1}^{N} Y_I \qquad (3.7)$$

with expectation

$$E(\sum_{1}^{N} Y_I) = \sum_{1}^{N} P_I = \mu_{sv} \qquad (3.8)$$

where N is the number of components in the subset. Therefore, the controlled estimate of the response Y is given by

$$Y(\bar{\beta}) = \bar{Y} - \bar{\beta}(SV - \mu_{sv}) \qquad (3.9)$$

where

$$SV = \sum_{1}^{M} SV_J. \qquad (3.10)$$

and M is the sample size.

MAXFLO automatically calculates the expected number of operative nodes of the control subset by simply adding their individual probabilities of survival. After each simulation, MAXFLO regresses out of the maximum flow the influence of the control subset, and gives the resulting mean, standard deviation, and 95% confidence interval for the uncontrolled and controlled maximum flow estimate. The control subset can include any combination of nodes from none to the entire network.

As for node selection, one obvious method for choosing which ones to put in the control subset is intuition based on network topology. However, a more precise procedure offered by the author is to use coefficients from the response surface polynomials as a guide for node selection. Since the coefficients are a measure of response sensitivity to network parameters, one can also say that they measure, *relative to each other*, the degree of correlation to maximum flow. Therefore, this study will use the RSM polynomial equations to help select the control subset, and compare the results to intuitive selections.

Example Problem.

The preceding topics lay the foundation for the experiments run in Chapter IV. The following example problem uses a procedure and format similar to the one used in Chapter IV's experiments.

Example Network. The example problem is based on the network in Figure 3-3 on the following page. The capacity of each component is represented as a integer value while the P_s

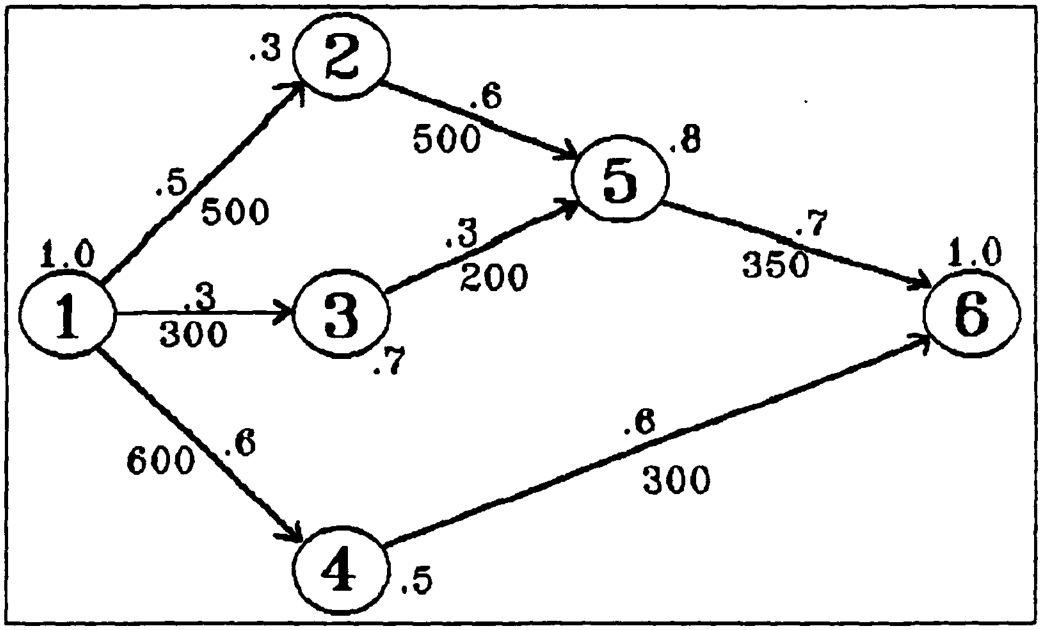

Figure 3-3. Example Problem Network

for that component is shown at two-digit significance (of course no greater than 1.0). For this sample model, all the P_s are independent.

Experiment Objectives. The objectives of this experiment are:

1. Verify MAXFLO Monte Carlo simulation routine by comparing simulation results to expected values calculated by spreadsheet.

2. Investigate effects of internal nodes on network performance as expressed in terms of maximum flow. This includes testing for quadratic effects and, if necessary, expanding the first-order design to determine second-order coefficients.

3. Use the results of Item (2) to select a subset of the internal nodes to use as control variates, and

investigate their effect on reducing the response variance.

4. Demonstrate methodology used on larger networks in Chapter IV.

Experimental Design. The experimental design selected for this project is a 2^3 full factorial orthogonal design shown in Table 3-5 on the following page. The three factors were selected based on the desire to analyze the effects of improving internal processing nodes. Since there are four candidates (nodes 2, 3, 4, and 5) and only three factors in the design, node 5 was dropped due to having the highest existing P_s. Nodes 2, 3, and 4 were entered in MAXFLO and are referred to as N2, N3, and N4.

The uncoded range of improvement for all three nodes is .2, based on a general assumption that hardening the network components is a difficult, marginal task. Transforming the uncoded values of the survival probabilities P_s into the coded values used in the experimental design follows Box and Draper's definition,

$$X_i = \frac{\delta_i - \delta_{io}}{S_i} \qquad (3.11)$$

where X_i is the coded value from -1 to 1, δ_{io} is the centerpoint of the range of interest, S_i is one-half the range of interest, and δ_i is the current point of interest (1987:20-21). For example, in the case of N2's P_s of .3, the range of improvement is .2, S_i is .1, and the centerpoint δ_{io} is .4.

TABLE 3-5 Experimental Design for Example Network In Figure 3-3

	Survival Probabilities							
	Uncoded			Coded			Antith.	Response -
Run #	N2	N3	N4	N2	N3	N4	Random Number	Maximum Flow
1	.3	.7	.5	-1	-1	-1	R	79.915
2	.3	.7	.7	-1	-1	1	A	99.660
3	.3	.9	.5	-1	1	-1	A	80.065
4	.3	.9	.7	-1	1	1	R	102.240
5	.5	.7	.5	1	-1	-1	A	90.190
6	.5	.7	.7	1	-1	1	R	113.195
7	.5	.9	.5	1	1	-1	R	90.355
8	.5	.9	.7	1	1	1	A	111.375
9	.4	.8	.6	0	0	0	4310089	97.080
10	.4	.8	.6	0	0	0	29153819	95.650
11	.4	.8	.6	0	0	0	513446243	96.900
12	.4	.8	.6	0	0	0	85491536	96.585
13	.4	.8	.6	0	0	0	3191455	97.120
14	.4	.8	.6	0	0	0	1801087584	95.265

If the design point calls for a coded value of 1, then the uncoded setting for the simulation is given by

$$1 = \frac{\delta_i - .4}{.1} \qquad (3.12)$$

or $\delta_i = .5$. Finally, the intent of this example in investigating the impact of internal nodes on maximum flow reduces the potential number of factors enough to preclude the use of screening designs.

Each design point represents one simulation of the sample network of Figure 3-3 with the appropriate parameters set according to Table 3-5. Each simulation ran 10,000 separate network samples to calculate the response and deviation. Additionally, six center-point runs were made to test for quadratic effects. This test will be covered shortly in the Experimental Results section.

Sampling Procedure. Another feature incorporated in this example design is the Schruben-Margolin assignment rule. This rule proposes that instead of using the same random number stream or independent random streams at all design points, use a common random number stream on one-half of a design that is blocked on a high-order interaction, and employ its antithetic random number stream on the other half. In order for this technique to produce a variance reduction, there must exist a negative correlation between the response of a common random number experiment and its antithetic counterpart (1978:504-520). Additional assumptions of this technique are covered in Chapter II.

Aside from the formal requirements, Schruben and Margolin's assignment rule also holds an intuitive appeal based on its antithetic approach. For example, if the common random number stream selected for the experiment turns out an artificially high or low estimate of the response, all design points in the experiment will be biased high or low. By using antithetic streams at blocked design points, that bias should be countered in the opposite direction.

This intuitive appeal is no substitute for meeting the assumptions stated in the literature review, however.

Furthermore, it is not clear such a sampling approach is better than common or independent random numbers. A conclusive proof of the assignment rule's effectiveness would require, among other things, evidence that a negative correlation exits between a given random number stream and its antithetic counterpart at all design points; clearly an exhaustive task for larger designs.

Instead, three empirical tests are made to assess the significance of antithetic sampling in SCBS networks. First, an evaluation of the simulation's sensitivity to antithetic random number streams at the first design point is made. Second, the existence of a negative correlation between a random number stream and its antithetic counterpart at the same design point is tested. Finally, the first eight design points in Table 3-5 are rerun using common and independent random numbers. The standard errors from the resulting response surfaces are compared to see if any approach is significantly better (or worse).

The first test offers empirical evidence of the relationship between bias and antithetic random numbers. Figure 3-4 on the following page shows the plot of MAXFLO's estimates of the first design point of Table 3-5 for various sample sizes against the actual calculated expected flow of 78.061. The regular random number stream for seed 33425688, $(r_1, r_2, \ldots, r_N)$, consistently underestimates the actual flow for sample sizes above 2000, whereas its antithetic

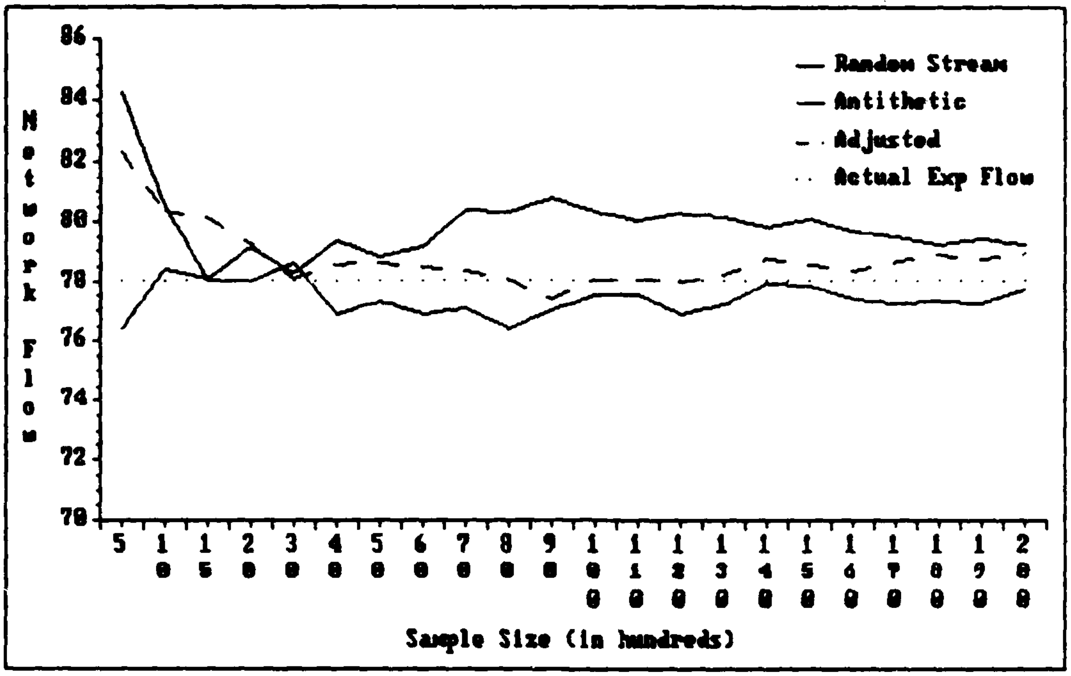

Figure 3-4. Plot of Maximum Flow Estimates

counterpart $(1-r_1, 1-r_2, \ldots\ ., 1-r_N)$ overestimates for sample sizes larger than 1000. The adjusted estimate, using non-synchronized *antithetic pairs* $(r_1, 1-r_1, r_2, 1-r_2, \ldots, r_{N/2}, 1-r_{N/2})$ appears to correct this bias. In other words, where the regular random number stream fails to "visit" the higher flow network configurations often enough, its antithetic stream will counter by sampling them too often. Thus, it appears that antithetic techniques can correct the bias of small sample sizes. However, research using *synchronized* antithetic pairs is recommended before drawing any conclusions.

The next test looks at the requirement of negative correlation of regular and antithetic random number streams for the Schruben-Margolin assignment rule. Table 3-6 shows the results of 24 simulations (10,000 samples each) using 12 independent random number streams and their antithetic counterparts at the first design point in Table 3-5. SAS PROC CORR ran this data to determine the amount and direction of correlation.

The result of .00793 correlation indicates the antithetic responses are nearly independent of the regular random number stream. Unfortunately, this does not support the assumption of negative correlation between the two random number streams. Therefore, the assumption of negative correlation at all design points may not hold. While this is not conclusive evidence for all design points, it does cast doubt on the advantages of using the Scruben-Margolin assignment rule.

TABLE 3-6 Comparison of Simulation Output at Design Point 1 (Table 3-5) of Regular and Antithetic Number Streams

Random Number Seed	Regular Stream	Antith. Stream	Actual Exp Flow
4310089	79.915	77.940	78.061
29153819	78.560	79.27	"
513446243	77.860	75.515	"
85491536	77.630	78.010	"
3191455	76.885	78.025	"
1801087	77.185	77.455	"
30131595	79.495	78.235	"
6718321	77.830	80.925	"
968328	77.445	79.415	"
74599049	78.110	78.300	"
51427813	78.115	77.235	"
108979503	78.110	75.700	"

The third test compares the standard errors of the response surfaces derived from the first eight design points in Table 3-5 using different sampling techniques. (A detailed explanation of how these parameters are estimated follows shortly in the Experimental Results section.) Comparing the results of assignment rule, common random number, and independent random number sampling techniques in Table 3-7 indicates that the Schruben-Margolin procedure has the lowest error of .373. Common random number sampling is a close second; however, independent random number sampling is clearly at a disadvantage with a standard error rate three times that of the assignment rule. (Also note the similarities of parameter estimates between the three techniques.) This simple test, while not conclusive, offers strong

TABLE 3-7 Parameter Estimates and Standard Errors for First-Order Response Surface Model (Coded Variables)

Sampling Technique	Variable	Parameter Estimate	Stnd. Error
Schruben-Margolin	Intercept	95.874	.373
	N2p	5.404	.373
	N4p	10.743	.373
Common Random Number	Intercept	94.370	.485
	N2p	5.380	.485
	N4p	11.500	.485
Independent Random Number	Intercept	94.860	1.155
	N2p	6.480	1.155
	N4p	11.440	1.155

evidence that either the Schruben-Margolin assignment rule or common random numbers is the best sampling strategy.

The previous tests show that simulation of SCBS networks is subject to a small bias, yet sensitive enough to the antithetic aspects of random number generation to allow for correction. Furthermore, the antithetic requirements of the assignment rule may not exist, nor does that sampling technique offer any major advantage over common random numbers. Because of these doubts, and the simplicity of common random number sampling, Chapter IV's experiments use the latter technique. For current research efforts in this regard, see Wilson and Tew (1987), and Nozari and others (1987).

While the following chapter uses common random number sampling, this example uses the assignment rule for the purpose of demonstrating the technique. For this example, the random number stream assignments are shown in the ANTITHETIC VAR column in Table 3-5, where R is the normal random number stream whose seed is 4310089, and A is its antithetic version. The random number stream assignments are based on a three-way interaction blocked design. Centerpoint simulations use regular, independent random number streams based on the seeds shown in the ANTITHETIC VAR column.

Experimental Results. The first objective is the verification of the simulation routine. Table 3-8 shows the simulations' estimated responses to the expected values.

Since all 14 simulations' confidence intervals (α = .05) contain the actual expected maximum flow, chances are MAXFLO's Monte Carlo routine performs properly. (Specifically, this test *fails to disprove* the hypothesis of MAXFLO correctly performing the Monte Carlo simulation.) Additionally, the diagnostics routine of MAXFLO shows 3 paths and 10 proper cuts formed by this network; this data too is confirmed by manual inspection of the example network to be correct.

A caveat about the confidence intervals should be mentioned. MAXFLO calculates small-sample confidence intervals using the t-distribution statistic. Technically, calculating this confidence interval assumes random sampling

TABLE 3-8 Comparison of Simulation Estimate of Calculated Expected Maximum Flow

	Uncoded			Simulation Estimate	Actual
Run #	N2	N3	N4	95% Conf. Interval	Exp. Flow
1	.3	.7	.5	79.915 ± 2.797	78.061
2	.3	.7	.7	99.660 ± 3.041	99.661
3	.3	.9	.5	80.065 ± 2.827	79.896
4	.3	.9	.7	102.240 ± 3.059	101.496
5	.5	.7	.5	90.190 ± 3.013	89.398
6	.5	.7	.7	113.195 ± 3.231	110.998
7	.5	.9	.5	90.355 ± 3.032	91.111
8	.5	.9	.7	111.375 ± 3.217	112.711
9	.4	.8	.6	97.080 ± 3.049	95.416
10	.4	.8	.6	95.650 ± 3.036	95.416
11	.4	.8	.6	96.900 ± 3.016	95.416
12	.4	.8	.6	96.585 ± 3.056	95.416
13	.4	.8	.6	97.120 ± 3.034	95.416
14	.4	.8	.6	95.605 ± 3.061	95.416

from a continous, normal distribution; though, it is also appropriate for populations with moderate deviations from normality, and in certain cases where there is a normal approximation to a binomial distribution (Mendenhall and others, 1986:287-288;330-331). Calculating the confidence intervals of these simulation estimates requires these assumptions because of the high frequency (.75 to .85) of zero flow, and the discrete nature of SCBS networks. Apparently, the t-statistic is robust enough to use on the example network distribution, and there is no reason to suspect it to be less so on the larger networks. But the assumptions and limitations of using it for these networks should be kept in mind.

The second objective is investigating the affect of the internal nodes on network performance. This is accomplished by linear regression, using the SAS procedure PROC GLM to calculate the sums of squares and coefficients. The results, shown in Table 3-9, indicate that the survival probability of node 3 (N3p) has virtually no effect on the expected flow. However, N2p accounts for 20% of the total sums of squares, and N4p an overwhelming 79%. These results do not reflect the main effects the author expected, though. A closer examination of the model shows why N4p dominates the ANOVA table.

The other two main effects are part of the top two paths in the network, where both contain more components than the

TABLE 3-9 Analysis of Variance Table for First-Order Response Surface Model (2^3)

Source	D.F.	Sums of Squares	Mean Square	F Stat.
Model:	3	1157.12	385.71	284.72
N2p	1	233.66		172.48
N3p	1	.14		.11
N4p	1	923.32		681.58
Error	4	5.42	1.36	
Total	7	1162.54		
R Square	.995			

bottom path that includes N4p. Therefore, the effect of increasing N2p or N3p is mitigated by the very high probability of path failure due to at least one of the other components failing. By contrast the bottom path contains only three stochastic components; thus, any improvement in one of its component's survivability will affect that path's reliability to a greater degree than the top two paths.

To calculate the parameter estimates, N3p's sum of squares is moved into the error sums of squares, giving the results shown in Table 3-10. (This is the same procedure used to calculate the results of Table 3-7.) Remembering that these estimates are for *coded* variables, the following polynomial equation describes the response for the range of variables described in the experiment by Table 3-5:

$$Y = 95.874 + 5.404(N2p) + 10.743(N4p) \qquad (3.13)$$

TABLE 3-10 Parameter Estimates for First-Order Response Surface Model (Coded Variables)

Variable	D.F.	Parameter Estimate	Stnd. Error	T Stat.
Intercept	1	95.874	.373	257.08
N2p	1	5.404	.373	14.49
N4p	1	10.743	.373	28.81

Tables 3-7 and 3-10, and Eq (3.13) not only predict the simulation output, but the parameter estimates measure the sensitivity of the estimated maximum flow to their respective components as well. Also, as covered shortly, the coefficients provide a guide for selecting nodes for the control variate subset.

Before proceeding to that aspect of the simulation, a test for the presence of second-order effects in the network should be conducted. Following Montgomery (1984:449-450), 6 runs were made at the design center using 6 independent common random number streams (Runs 9-14 in Table 3-5) to calculate a pure estimate of error σ^2. For this example, σ^2 is found by dividing Eq (3.14)

$$(97.08)^2 + (95.65)^2 + (96.9)^2 + (96.585)^2 + (97.12)^2 + (95.605)^2 - (578.94)^2/6 \qquad (3.14)$$

by 5, giving an estimate of error of 2.3292.

Next, the sums of squares for pure quadratic, SS_{QUAD}, is found by

$$\frac{N_1 N_2 (Y_1 - Y_2)^2}{N_1 + N_2} \qquad (3.15)$$

where N_1 is the number of experimental design points, N_2 is the number of centerpoints, Y_1 is the average response of the experimental design, and Y_2 is the average response at the centerpoint. For this example, SS_{QUAD} is 1.301.

Finally, an F-statistic for quadratic effects is given by Eq (3.16)

$$F = \frac{SS_{QUAD}}{\sigma^2} \qquad (3.16)$$

which in this example is .5586. This is considerably lower than 6.61 for the test statistic $F_{.05,1,8}$, thus failing to disprove the hypothesis of no quadratic effects. Therefore, there is no reason to develop a second-order model.

The third objective is to select nodes for the control variate subset to reduce the variance of the simulation output. Since the response surface in Eq (3.13) indicates N2 and N4 exert the greatest influence on expected maximum flow, they are the most likely candidates for consideration. Since this example network is small enough to investigate all four intermediate nodes, various combinations of nodes were tried to test the validity of using response surface coefficients. Table 3-11 on the following page summarizes the results of different control variate subsets on the design centerpoint,

TABLE 3-11 Variance Reduction Based on Survival of Nodes in Control Subset at Centerpoint of Design

Nodes in Control Subset	Estimated Max Flow	Std. Dev.	95% Conf Interval
0	97.080	155.58	± 3.049
4	96.489	146.39	± 2.869
2,4	96.356	144.78	± 2.838
2,3,4,5	95.791	147.03	± 2.880
3	97.074	155.58	± 3.050
3,4	96.314	150.166	± 2.944
Actual Exp Flow	95.416		

where the random number seed is 4310089 and the sample size is 10,000.

The control subset with the greatest variance reduction is {2,4}, giving a 7% reduction in variance and the confidence interval over the uncontrolled {0} response. By contrast, control subset {3} very slightly *increases* the variance, subset {3,4} barely shows a variance reduction, while subset {2,3,4,5} comes in third best after {4} and {2,4}. Clearly, including Node 3 in the control subset adds nothing but statistical noise to the regression, while Nodes 2 and 4 contribute substantially to the variance reduction. This behavior is predicted by the response surface of Eq (3.9), suggesting the methodology for selecting nodes for the control subset is sound.

The 7% reductions in variance is somewhat less than expected. The low reduction may be partially due to the fact that no single node is a 'choke point'; i.e., the paths aren't dependent enough on a node for it to exert a greater influence on the expected maximum flow. Furthermore, as pointed out earlier, a large number of stochastic components with low survival rates tends to diminish the effects of hardening a given node. It therefore makes sense that this feature would also diminish the correlation that node has with the overall flow in the network, thereby mitigating its effectiveness as a control variate. Finally, a second class of controls representing arc survivability has not been considered; yet, it could significantly contribute to variance reduction.

These two factors - topology and reliability - will vary among networks such that predicting the effectiveness of control variates is difficult. Keeping this in mind, and using the response surface equation as a selection guide, the control subset of nodes should provide a simple but useful technique for variance reduction. Thus, following the methodology just presented, the next chapter presents the results of the larger networks.

IV. Experimental Results

This chapter presents the experimental results of Networks A, B, and C. Specifically, a response surface analysis of Network C is conducted, followed by variance reduction investigations on all three networks.

Response Surface Analysis

Example Network. A response surface analysis was done on Network C, whose topology is given in Figures 4-1 and 4-2 on the following pages. (The network's link list, the document specifying all network component capacities and survival probabilities, is found in Appendix C.) Figure 4-2 differs from the original network configuration in the numbering of the nodes (due to dummy single source and sink nodes), and the use of arc equivalents to replace nodes. The use of arc equivalents is a powerful technique for reducing the number of cuts in a network, thus increasing simulation efficiency. Therefore, a short explanation of its implementation is in order.

The original configuration of Network C in Figure 4-1 contains 39 nodes, 53 arcs, 198 paths, and 1,037 cuts. The vast majority of the cuts (specifically 2^{10}, or 1024) are due to the configuration of Nodes 13 through 22. Based on the discussion of cutsets in Chapter II, it follows that any reduction in the number of nodes in this group will *exponen-*

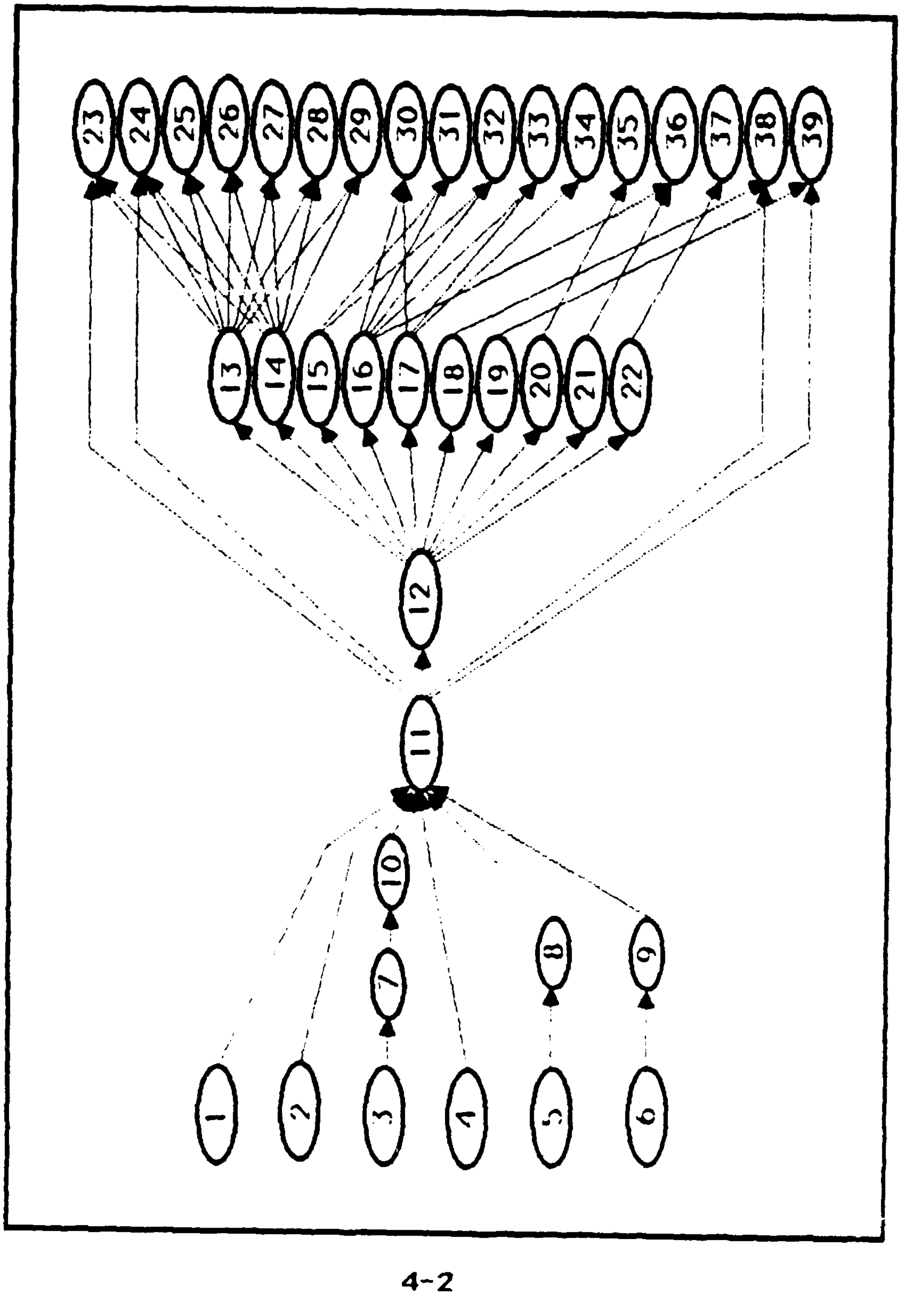

Figure 4.1. Network C Topology

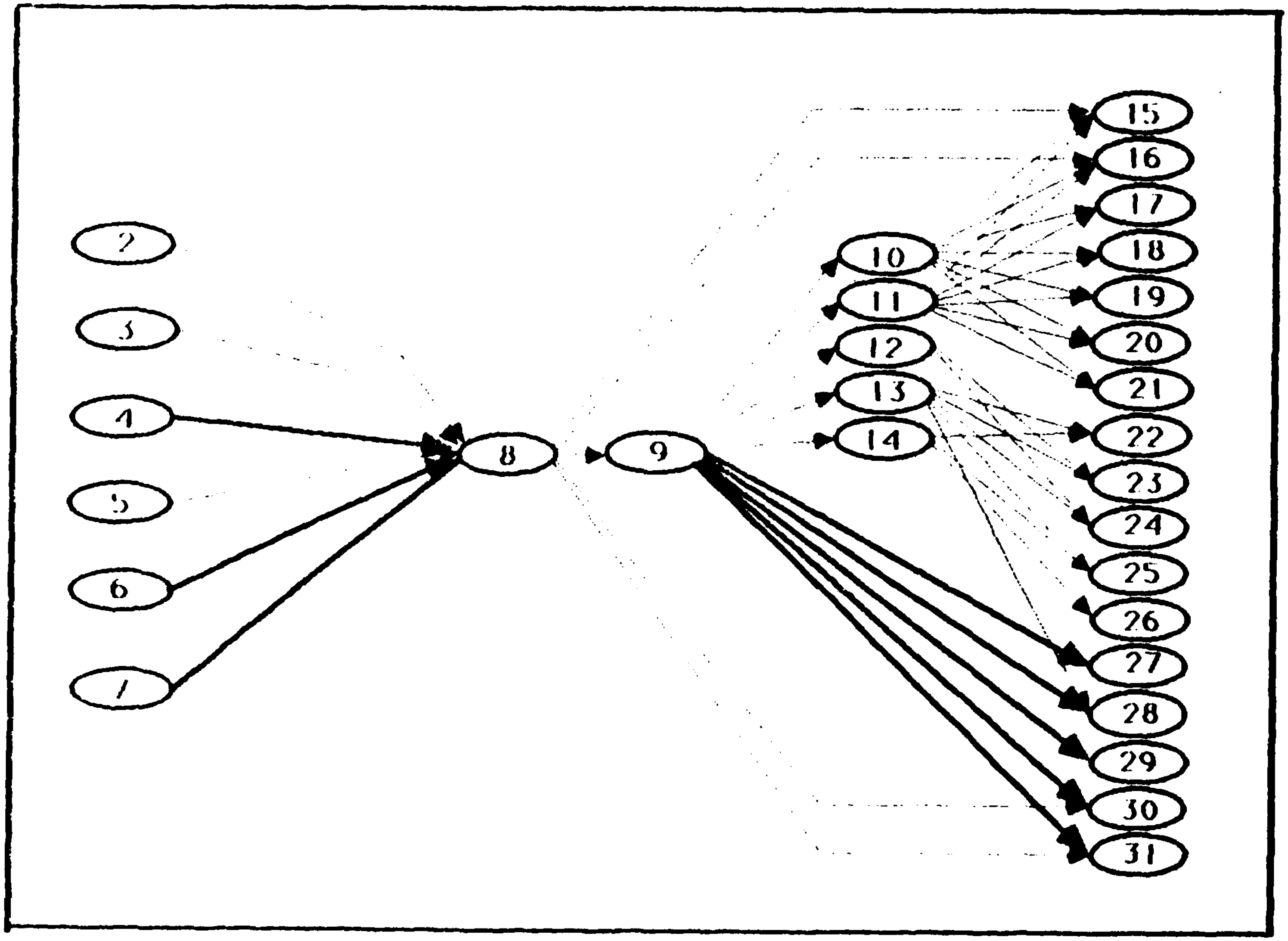

Figure 4 2. Network C Topology With Equivalent Arcs *tially* reduce the number of cuts. Arc equivalents is one such method of eliminating nodes that meet certain conditions.

Specifically, if a sequence of nodes exists such that each node has only one incoming and one outgoing arc, those nodes and incident arcs can be replaced with an equivalent arc. Furthermore, if all components' P_s in the sequence are independent, then the equivalent arc's P_s is the product of the replaced components' P_s, and its capacity is the minimum value of the replaced components' capacities. For instance, in Figure 4-1 the segment from Node 12 to Node 38 contains three independent components: Arc 12-18 (P_s = .7, Capacity = 4800), Node 18 (P_s = .5), and Arc 18-38 (P_s = .7, Capacity = 1200). This segment is replaced in Figure 4-2 by Arc 9-30 whose P_s is .245 and capacity is 1200.

Additionally, because the arc equivalent is an equal structure, it *does not introduce bias in the Monte Carlo simulation*. Therefore, not only is the simulation more efficient, but its estimate of maximum flow is equally valid. (If one or more components in the sequence is *dependent*, an arc equivalent is still possible; however, calculating the arc equivalent's components is more difficult. Since Network C does not contain dependent components, an example is not offered.)

Using arc equivalents, Nodes 7, 8, 9, 10, 18, 19, 20, 21, and 22 in Figure 4-1 are absent in Figure 4-2, resulting in an *equivalent network* with only 30 nodes, 44 arcs, 198 paths, and *34* cuts - a considerable reduction of the size of the cutset. Another interesting observation is that the number of paths in Figures 4-1 and 4-2 is the same. Indeed, this observation is an example of a unique characteristic of arc equivalents. Specifically, *arc equivalents only reduce the number of proper cuts, while the number of paths remains unchanged*; although, both pathsets and cutsets benefit by the reduced number of network components. Thus, Network C, as shown in its equivalent form in Figure 4-2, is used by MAXFLO for the experimental design.

Experimental Design and Results. The design objective is to find those components whose improvements will best increase network performance as measured by estimated maximum flow and network reliability. Because Network C contains so many possible factors (118 to be exact), a combination of intuition and Plackett-Burman screening designs is used. (It is also possible to screen *all* components by using a combined group and factor screening design. Although this technique is beyond the scope of this text, it is a good topic for future study.) For this network, the following 19 factors from Figure 4-2 were selected.

The first two candidates are obvious due to their position - the P_s for Nodes 8 and 9 (or N8p and N9p). The survival rates for Nodes 10, 11, 13, and 14 (N10p, N11p, N13p, and N14p) are also good selections since between the four of them they affect 19 paths. The four arcs that go directly from Node 8 to sink Nodes 15, 16, 30, and 31 are good choices since they collectively represent the shortest and most reliable paths in the network. Since both their survival rates (A8-15p, A8-16p, A8-30p, and A8-31p) and capacities (A8-15c, A8-16c, A8-30c, and A8-31c) are relatively low, all eight are included in the screening design.

On the "supply side" of the network, the capacities of Arcs 2-8, 3-8, 5-8, 7-8, and 8-9 (A2-8c, A3-8c, A5-8c, A7-8c, and A8-9c) should be included since network capacity beyond Node 9 exceeds the capacity of the source nodes and their incident arcs. Because the survival rates of these arcs are somewhat higher, those parameters are not examined. One point to emphasize is that larger designs are available to accommodate more factors; indeed, Plackett and Burman offer two-level screening designs for up to 99 factors (1946:324).

The resulting 19 factor screening is shown in Table 4-1 on the following pages. The low (-) values are those that currently exist, while the high (+) values represent the potential improved capacity or P_s. Capacity improvements are based on standard increments of 300, 1200, 2400, 9600, and 19200, while P_s improvements are a uniform increase of .2. The design was run on a VAX 8650 under VMS 4.6 using a sample size of 10000 and regular random number stream with 3036869 as the seed. The output results from MAXFLO are shown in Table 4-2.

Table 4-1. Screening Design for Network C

Run	N8p	N9p	N10p	N11p	N13p	N14p
1	+	+	-	-	+	+
2	-	+	+	-	-	+
3	+	-	+	+	-	-
4	+	+	-	+	+	-
5	-	+	+	-	+	+
6	-	-	+	+	-	+
7	-	-	-	+	+	-
8	-	-	-	-	+	+
9	+	-	-	-	-	+
10	-	+	-	-	-	-
11	+	-	+	-	-	-
12	-	+	-	+	-	-
13	+	-	+	-	+	-
14	+	+	-	+	-	+
15	+	+	+	-	+	-
16	+	+	+	+	-	+
17	-	+	+	+	+	-
18	-	-	+	+	+	+
19	-	-	-	+	+	+
20	-	-	-	-	-	-

Run	A8-15c	A8-15p	A8-16c	A8-16p	A8-30c	A8-30p
1	+	+	-	+	-	+
2	+	+	+	-	+	-
3	+	+	+	+	-	+
4	-	+	+	+	+	-
5	-	-	+	+	+	+
6	+	-	-	+	+	+
7	+	+	-	-	+	+
8	-	+	+	-	-	+
9	+	-	+	+	-	-
10	+	+	-	+	+	-
11	-	+	+	-	+	+
12	-	-	+	+	-	+
13	-	-	-	+	+	-
14	-	-	-	-	+	+
15	+	-	-	-	-	+
16	-	+	-	-	-	-
17	+	-	+	-	-	-
18	-	+	-	+	-	-
19	+	-	+	-	+	-
20	-	-	-	-	-	-

Table 4-1. Screening Design for Network C (Cont.)

Run	A8-31c	A8-31p	A2-8c	A3-8c	A5-8c	A7-8c	A8-9c
1	-	-	-	-	+	+	-
2	+	-	-	-	-	+	+
3	-	+	-	-	-	-	+
4	+	-	+	-	-	-	-
5	-	+	-	+	-	-	-
6	+	-	+	-	+	-	-
7	+	+	-	+	-	+	-
8	+	+	+	-	+	-	+
9	+	+	+	+	-	+	-
10	-	+	+	+	+	-	+
11	-	-	+	+	+	+	-
12	+	-	-	+	+	+	+
13	+	+	-	-	+	+	+
14	-	+	+	-	-	+	+
15	+	-	+	+	-	-	+
16	+	+	-	+	+	-	-
17	-	+	+	-	+	+	-
18	-	-	+	+	-	+	+
19	-	-	-	+	+	-	+
20	-	-	-	-	-	-	-

Coded Value	Uncoded Values						
	N8p	N9p	N10p	N11p	N13p	N14p	
-	.7	.7	.5	.8	.3	.7	
+	.9	.9	.7	1.0	.5	.9	
	A8-15c	A8-15p	A8-16c	A8-16p	A8-30c	A8-30p	
-	75	.6	75	.3	1200	.6	
+	300	.8	300	.5	2400	.8	
	A8-31c	A8-31p	A2-8c	A3-8c	A5-8c	A7-8c	A8-9c
-	1200	.7	1200	1200	300	300	9600
+	2400	.9	2400	2400	1200	1200	19200

SAS PROC GLM was used to calculate the regression results, which appear in Table 4-3. The results show that out of the original 19 factors, only five account for a

Table 4-2. MAXFLO Estimates of Table 4-1 Design Points

Run	Estimated Maximum Flow	Network Reliability
1	2136.702	84.40
2	1471.080	65.57
3	1626.870	84.32
4	2044.843	84.52
5	1900.489	66.73
6	1702.523	64.89
7	1859.627	64.52
8	1729.518	64.95
9	2555.430	83.80
10	2245.587	65.67
11	2646.071	82.40
12	2214.773	65.91
13	2084.413	83.00
14	2123.717	84.62
15	2621.388	83.47
16	2774.981	84.13
17	1883.010	66.43
18	1749.056	64.03
19	2310.270	79.78
20	1169.152	62.78

significant portion of the sums of squares for expected maximum flow: N8p, N9p, A2-8c, A3-8c, and A5-8c. Together, these five factors explain 95% of the variation of expected maximum flow. Because this is a screening design, only the main effects are measured (Plackett and Burman, 1946:323); however, the number of factors is reduced enough to allow for a full factorial design.

Interestingly, only *one* factor out of the 19 screened accounts for a significant amount of the sums of squares for reliability: N8p. As Table 4-3 shows, 98% of the sums of squares is explained by the variation of N8p. Since N8p

Table 4-3. Sums of Squares for Table 4-1 Design

Source	Sums of Squares	
	Maximum Flow	Reliability
MODEL	3299594.387	1705.785
N8p	1249935.001	1673.718
N9p	196741.382	14.416
N10p	246.837	0.808
N11p	3649.321	0.007
N13p	2223.266	0.255
N14p	168.386	0.001
A8-15c	30.076	0.002
A8-15p	3943.274	0.481
A8-16c	359.484	0.421
A8-16p	5383.399	3.715
A8-30c	261.075	0.318
A8-30p	3749.855	2.113
A8-31c	80347.080	0.648
A8-31p	25760.694	5.429
A2-8c	153612.938	0.662
A3-8c	1203365.268	1.270
A5-8c	339612.880	0.392
A7-8c	17895.632	0.592
A8-9c	12308.539	0.538

appears to be a significant factor in maximum flow as well, it is an obvious choice for survival rate improvement. But before such conclusions can be drawn, several additional procedures need to be accomplished. These include a factorial design that tests for possible interactions, a check for second order effects (with a possible follow-up second order experimental design), and regression diagnostics.

Since there are five remaining factors, a full factorial design requires a only 32 design points (2^5). Additional design centerpoints are also required to test for second order effects, and to form the basis of a second order design. Since 10 centerpoints are required if we expand to a 2^5 central composite, uniform precision design (Montgomery, 1984:463), all 10 are simulated in addition to the required 32 design points. These centerpoints also provide a good statistical sampling for second order effects. Table 4-4 shows the experimental results. There were 10000 samples taken at each design point with a regular random number stream seed of 3036869, while the centerpoints used independent random number streams. (Note that the range of P_s for N8p and N9p is reduced by .04. This allows for a uniform precision second-order design if necessary.)

Following Table 4-4, Table 4-5 shows the results of the SAS PROC GLM regression of the data in Table 4-4 (without the centerpoints). The first-order model has an R-Square value is .988, indicating a high degree of fit of this model to the data. (Small, but statistically significant, two-way interactions are also present; however, they are ignored because of their *practical* insignificance). Furthermore, an additional check for second order effects is calculated as described in Chapter 3 using the centerpoint data from runs 33 through 42 in Table 4-4. The resulting F-statistic is 1.1017, considerably lower than the $F_{.05,1,9}$ value of 5.12.

Thus, it appears that the response of maximum expected flow for the coded variables is described by the first-order polynomial in Eq (4.1) on the following page. A more useful

Table 4-4. 2^5 Experimental Design for Network C

Run	N8p	N9p	A2-8c	A3-8c	A5-8c	Est. Max Flow	Est. Rel.
1	-	-	-	-	-	1169.152	62.78
2	-	-	-	-	+	1376.310	"
3	-	-	-	+	-	1548.608	"
4	-	-	-	+	+	1750.167	"
5	-	-	+	-	-	1310.505	"
6	-	-	+	-	+	1516.162	"
7	-	-	+	+	-	1687.208	"
8	-	-	+	+	+	1886.432	"
9	-	+	-	-	-	1288.522	64.21
10	-	+	-	-	+	1527.113	"
11	-	+	-	+	-	1743.144	"
12	-	+	-	+	+	1974.992	"
13	-	+	+	-	-	1464.682	"
14	-	+	+	-	+	1700.581	"
15	-	+	+	+	-	1915.268	"
16	-	+	+	+	+	2144.053	"
17	+	-	-	-	-	1434.863	77.33
18	+	-	-	-	+	1679.648	"
19	+	-	-	+	-	1889.871	"
20	+	-	-	+	+	2127.018	"
21	+	-	+	-	-	1614.300	"
22	+	-	+	-	+	1857.175	"
23	+	-	+	+	-	2065.141	"
24	+	-	+	+	+	2298.823	"
25	+	+	-	-	-	1573.297	79.52
26	+	+	-	-	+	1865.037	"
27	+	+	-	+	-	2129.271	"
28	+	+	-	+	+	2414.532	"
29	+	+	+	-	-	1781.073	"
30	+	+	+	-	+	2070.851	"
31	+	+	+	+	-	2332.181	"
32	+	+	+	+	+	2614.178	"
33	0	0	0	0	0	1801.424	71.37
34	0	0	0	0	0	1833.608	71.02
35	0	0	0	0	0	1820.961	71.29
36	0	0	0	0	0	1820.931	71.43
37	0	0	0	0	0	1816.268	71.20
38	0	0	0	0	0	1815.133	70.86
39	0	0	0	0	0	1803.838	70.43
40	0	0	0	0	0	1797.171	70.91
41	0	0	0	0	0	1779.531	70.25
42	0	0	0	0	0	1816.184	70.92

Table 4-4. 2^5 Experimental Design for Network C (Cont.)

Coded Value	Uncoded Values				
	N8p	N9p	A2-8c	A3-8c	A5-8c
-	.70	.70	1200	1200	300
+	.86	.86	2400	2400	1200
0	.78	.78	1800	1800	750

Table 4-5. ANOVA and Parameter Estimates of 2^5 Experimental Design

Source	DF	Sum of Squares	F-Value
Model	5	3742567.705	428.800
N8p	1	1031177.244	590.73
N9p	1	345985.548	198.21
A2-8c	1	239270.791	137.07
A3-8c	1	1661489.497	951.82
A5-8c	1	464644.626	266.18
Error	26	45385.368	
Total	31	3787953.074	

Parameter	Estimate	Param. = 0 T Value	Std. Error of Parameter
Intercept	1804.692	244.35	7.386
N8p	179.511	24.30	7.386
N9p	103.981	14.08	7.386
A2-8c	86.471	11.71	7.386
A3-8c	227.863	30.85	7.386
A5-8c	120.500	16.32	7.386

$$Y = 1804.692 + 179.511(N8p) + 103.981(N9p) + 86.471(A2\text{-}8c) + 227.863(A3\text{-}8c) + 120.5(A5\text{-}8c) \qquad (4.1)$$

version of Eq (4.1) using the uncoded values is found by converting the coefficients. For this example, the uncoded version is

$$Y = -2{,}103.19 + 2243.389(N8p) + 1299.763(N9p) + .144(A2-8c) + .380(A3-8c) + .268(A5-8c) \qquad (4.2)$$

Both equations are good only for the region of the response surface defined by the input domain of Table 4-4.

Before continuing with an analysis of this section's results, two tests were conducted to confirm the statistical assumptions of linear regression. Specifically, a plot of residuals versus predicted values is shown in Figure 4-3 to substantiate the presence of constant variance, and the plot of residuals in Figure 4-4 is presented to confirm a normal distribution (Box and Draper, 1987:119-123,128-131).

The plot in Figure 4-3 has a slight pattern but, noting the disparate scale of the two axes, the residuals do not appear to be practically significant. Figure 4-4 varies slightly from a normal plot to one indicating a heavy-tailed distribution. Given the frequency that zero flow occurs in this network, a slightly skewed distribution is not surprising. Furthermore, the small sample size may also contribute to the slight departure from normality. Finally, as a matter of curiosity, all 10000 sample results from the first run in Table 4-4 were collected to plot the histogram of the maximum flow distribution shown in Figure 4-5 on the following page.

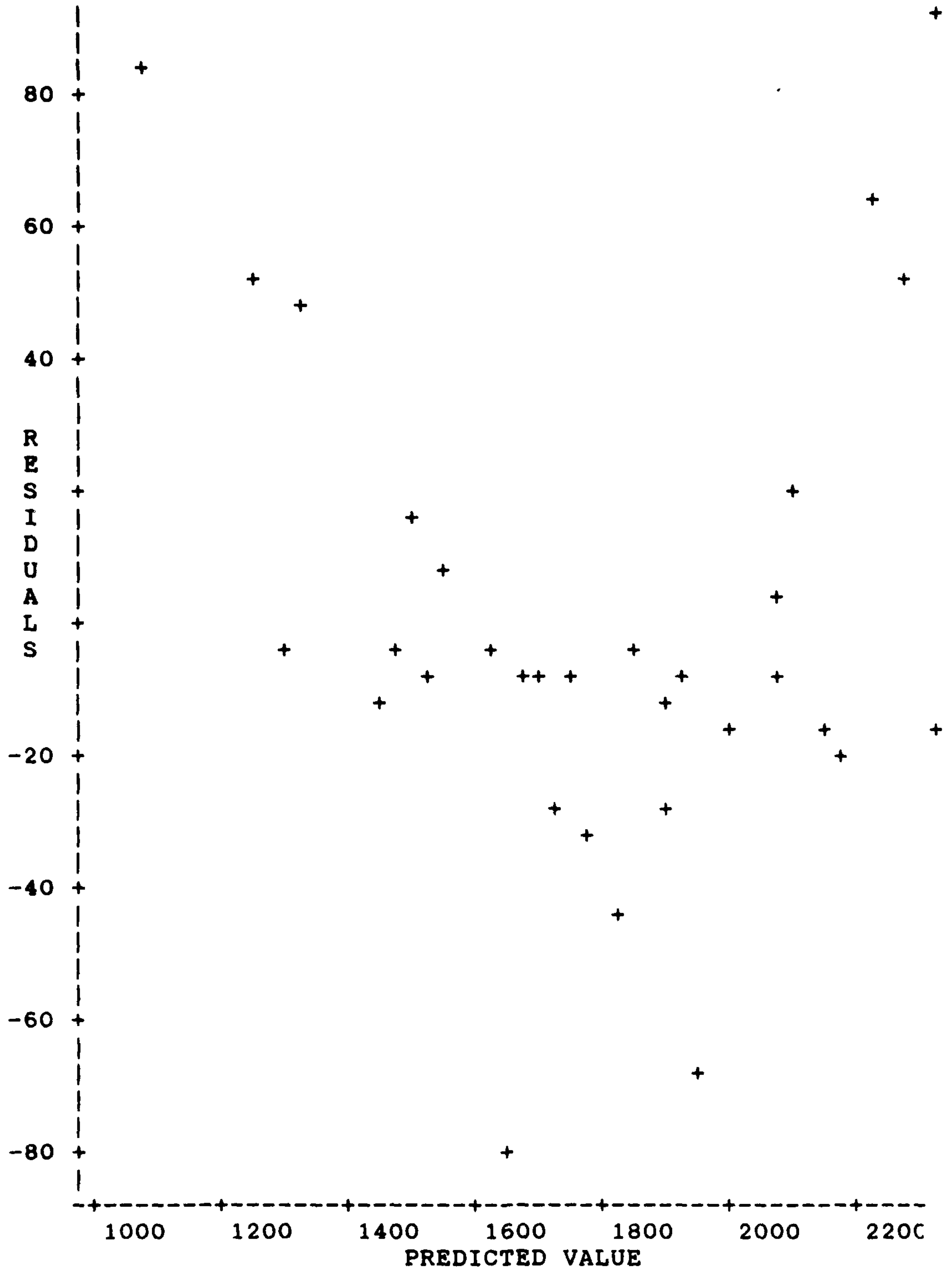

Figure 4-3. Plot of Residuals Versus Predicted Values

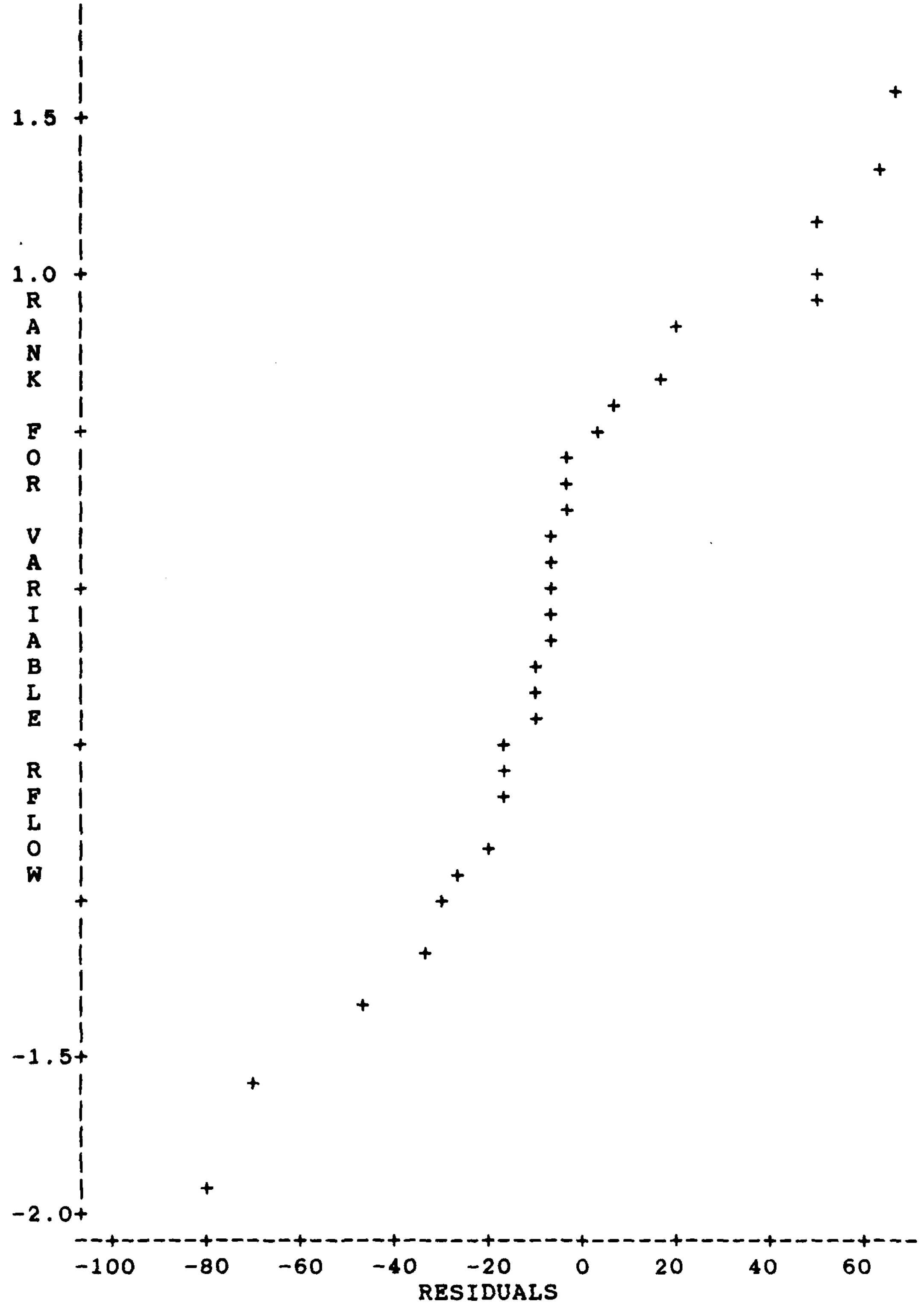

Figure 4-4. Normal Probability Plot

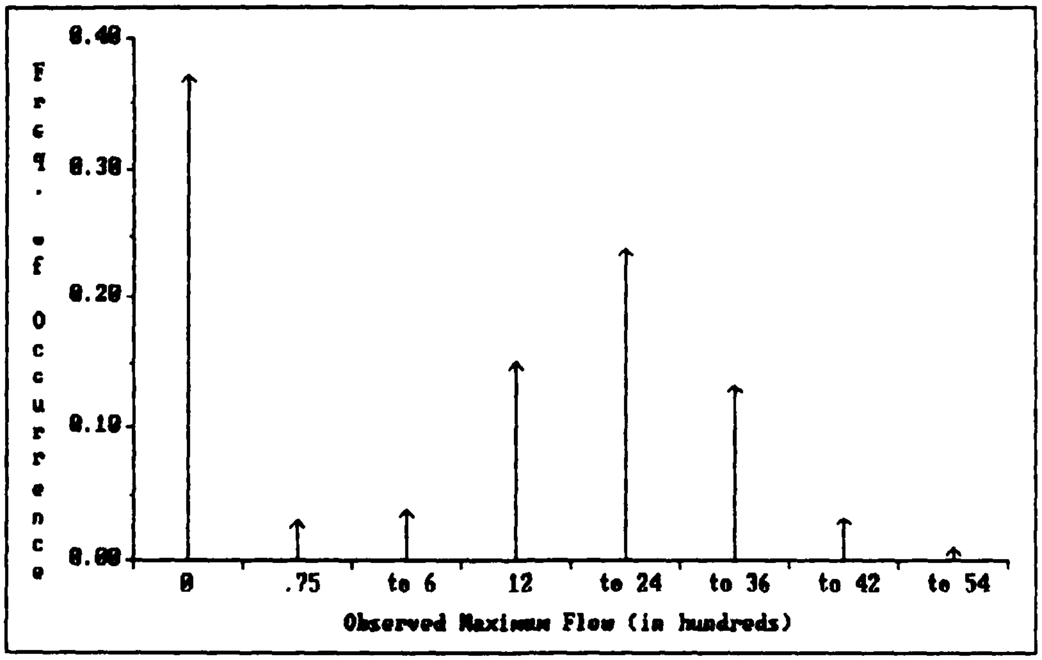

Figure 4-5. Histogram of Sample Maximum Flow at Design Point 1 of Table 4-4

Analysis of Response Surface. Given that Eqs (4.1) and (4.2) accurately describe the response surface of Network C maximum flow, several insights into this network's performance are available.

First, any improvement in network maximum flow should focus on getting more information from the source nodes to Node 8. This is clearly demonstrated by the fact that *three* of the five significant parameters are the capacities of arcs incident to the source nodes. This occurs in spite of the fact that *ten* arcs from Nodes 8 and 9 to the sink nodes were screened *for both capacity and survival rate.* Apparently, the network flow is diverse enough after Nodes 8 and 9 to insure that some flow will get through.

A second useful observation is obtained by comparing the response surface of expected maximum flow to that of network reliability. Following the same procedure used for finding Eqs (4.1) and (4.2), the uncoded version of the network reliability response surface (in percentages) is given by the first-order polynomial

$$Y = 62.84 + 9.4(N8p) + .94(N9p) + .71(A8-31p) \qquad (4.3)$$

The insight provided by this response surface is the overwhelming influence of Node 8 on network reliability (which is probably due to the node's position in the network). Apparently, flow from the source nodes arrives often enough (though in not enough quantity) that if Node 8 survives, then at least one of the sink nodes will receive flow as well. Since Node 8 is also the second most influential component in the maximum flow response surface, any improvement in it will produce increased network performance in both areas.

Comparing the two response surfaces leads to one final observation of the relationship between maximum flow and network reliability. Simply stated, the two responses estimate two very different types of network performance - one the average *quantity* of flow, and the other *how often any amount of flow* gets through. Therefore, the choice of response variable should reflect the type of network improvement being sought; i.e., the measure of network performance should also be the measure of network effectiveness. (Since the two estimates compliment each other, and because MAXFLO routinely calculates both of them, the author recommends considering both measures.)

The previous observations are examples of one type of analysis provided by RSM called *descriptive analysis*, where the polynomial approximation of the response surface is studied within the context of gaining insight to the network's performance and component interaction. Another type of analysis is *prescriptive analysis*, where the response surface polynomial is used to prescribe or recommend a course of action. A typical application of the second type of analysis would involve the response surface equation as the objective function in an optimization model. The following example illustrates this important feature.

Assume we wish to maximize the expected maximum flow of Network C as described by Eq (4.2), subject to the following constraints:

1. The cost of hardening nodes 8 and 9 is $10k per .1 unit of P_s. The total cost of hardening cannot exceed $15k.

2. The cost of increasing arc capacity for A2-8c, A3-8c, and A5-8c is $5k per 100 units. The total cost of increased capacity cannot exceed $150k.

3. The total cost of improvement cannot exceed $160k.

4. Eq (4.2) is valid only for the region of space defined by the experimental design. Therefore, the five components' values are implicitly bound by the uncoded values given in Table 4-4.

Let the *improvement variables* H_8 and H_9 represent the amount of hardening for nodes 8 and 9; and, C_{2-8}, C_{3-8}, and C_{5-8} the increase of capacity for arcs A2-8c, A3-8c, and A5-8c, respectively. Since the coefficients of Eq (4.2) are applicable to both the original, uncoded variables *and* the improvement variables, the objective function can be re-written for just improvement variables (minus the intercept term). Thus, a linear programming formulation that maximizes network maximum flow subject to the listed constraints is

$$\begin{aligned}\text{Maximize } Z = 2234.389(H_8) + 1299.763(H_9) + .144(C_{2-8})\\ + .380(C_{3-8}) + .268(C_{5-8})\end{aligned} \tag{4.4}$$

subject to

$$\begin{aligned} H_8 + H_9 &\le .15 \\ C_{2-8} + C_{3-8} + C_{5-8} &\le 3000 \\ 100(H_8) + 100(H_9) + .05(C_{2-8}) + .05(C_{3-8}) + .05(C_{5-8}) &\le 160 \end{aligned} \tag{4.5}$$

and

$$\begin{gathered} 0 \le H_8 \le .16 \qquad 0 \le H_9 \le .16 \\ 0 \le C_{2-8} \le 1200 \qquad 0 \le C_{3-8} \le 1200 \qquad 0 \le C_{5-8} \le 900. \end{gathered} \tag{4.6}$$

The three inequalities in (4.5) formulate the cost restrictions of Items (1), (2), and (3) respectively, while the constraints in (4.6) reflect the implicit bounds of the design space mentioned in Item (4).

Using standard linear programming techniques, the optimal solution for this sample problem is 1147.558, where H_6 = .15, H_9 = 0.0, C_{2-6} = 800, C_{3-6} = 1200, and C_{5-6} = 900. Adding the intercept to the optimal flow improvement gives an estimated maximum flow of the improved network of 2227.475. As a further enhancement, multiple optimization is possible by using Eqs (4.2) and (4.3) as the goals in a goal programming formulation. (For additional explanations of linear programming and multiple optimization techniques, see Hillier and Lieberman (1986:Ch 3) or Chvatal (1980).)

The previous experimental designs and response surface equations should also provide excellent guidance for selecting nodes for the control subset. This concept, as well as variance reduction tests on the other two networks, are covered in the following section.

Control Variate Results

The selection of control variates and subsequent tests for variance reduction is simple and straight-forward. Those nodes whose positions in the network indicate that a large correlation between survival rate and network performance may exist are chosen for the control subset. Since MAXFLO automatically calculates variance and confidence intervals for both normal and control variate estimates of maximum flow, testing is simply a matter of running the simulation. The one restriction is that MAXFLO only accepts independent nodes for the control subset.

Since a response surface analysis of Network C was just presented, this section vill begin with the control variate experiments of that network. Of particular interest is a comparison of the influential and insignificant nodes in the response surface to the variance reduction they offer when part of the control subset. Subsequent sections report the control variate experiments on Networks A and B.

Network C. The original screening design in Table 4-1 looks at six nodes (N8, N9, N10, N11, N13, N14), whose position in Figure 4-2 indicates a possibly strong influence on expected maximum flow. Subsequent research found that only two, N8 and N9, are significant. Therefore, it stands to reason that these same two nodes will provide the largest variance reduction in the estimated maximum flow. Table 4-6 on the following page shows the results of various nodes in the control subset for a 10000 sample size simulation at design point 1 of Table 4-4.

As expected, Nodes 8 and 9 significantly reduce the variance of the estimated maximum flow and slightly adjust the point estimate of maximum flow downward. Clearly, Node 8 offers the best single-node control set reduction of 22% from the uncontrolled estimate, while Node 9 is a distant second with a variance reduction of 4%. Combined, Nodes 8

Table 4-6. Variance Reduction of Estimated Maximum Flow for Network C

Nodes in Control Subset	Estimated Maximum Flow	Variance	95% Conf. Interval
0	1169.152	1214.264	23.800
8	1168.604	943.087	18.483
9	1164.596	1162.458	22.786
8,9	1161.518	925.328	18.137
10	1169.202	1214.289	23.801
11	1168.862	1213.998	23.795
13	1169.171	1214.289	23.804
14	1169.170	1214.320	23.800

and 9 reduce the variance from the uncontrolled estimate by slightly under 24%. By contrast, Nodes 10, 11, 13, and 14 have no discernable affect on variance when included in the control subset.

The relative value of the nodes in variance reduction appears to parallel their influence in the experimental designs of the preceding section. Therefore, it seems that response surface techniques can be applied in selecting nodes for the control subset. Furthermore, as a topic for further research, the reverse procedure may also hold true; that is, *nodes producing significant variance reduction will also influence the estimated maximum flow response surface.* (This assumes uniformity of effect across *all* design points.) If so, testing nodes (and arcs) for variance reduction may be a more efficient way to screen factors than Plackett-Burman designs. (Additional selection procedures are also available without using RSM; specifically stepwise and all regression. For further information, see Bauer (1987) or Draper and Smith (1981:Ch 6).)

Network A. Figure 4-6 on the following page shows the topology of Network A, while the link list is given in Appendix A. This network differs from Network C in that several nodes and arcs are dependent on the survival rates of other nodes and arcs. Table 4-7 shows the variance reduction for several independent nodes selected for their position in the network. (Note that Figure 4-6 and Table 4-7 use original node numbers; i.e., prior to using a dummy source node and arc equivalents. Also, this network has 63 paths and 64 proper cuts.)

The results are not as impressive as those for Network C. Nodes 14 and 15 provide the best variance reduction with 10% off the uncontrolled results, while Node 14 is a close

Table 4-7. Variance Reduction of Estimated Maximum Flow for Network A

Nodes in Control Subset	Estimated Maximum Flow	Variance	95% Conf. Interval
0	618.960	1526.244	29.914
14	606.112	1400.871	27.463
15	617.143	1495.116	29.307
14,15	605.314	1370.794	26.873
2,3,4	618.125	1522.272	29.836

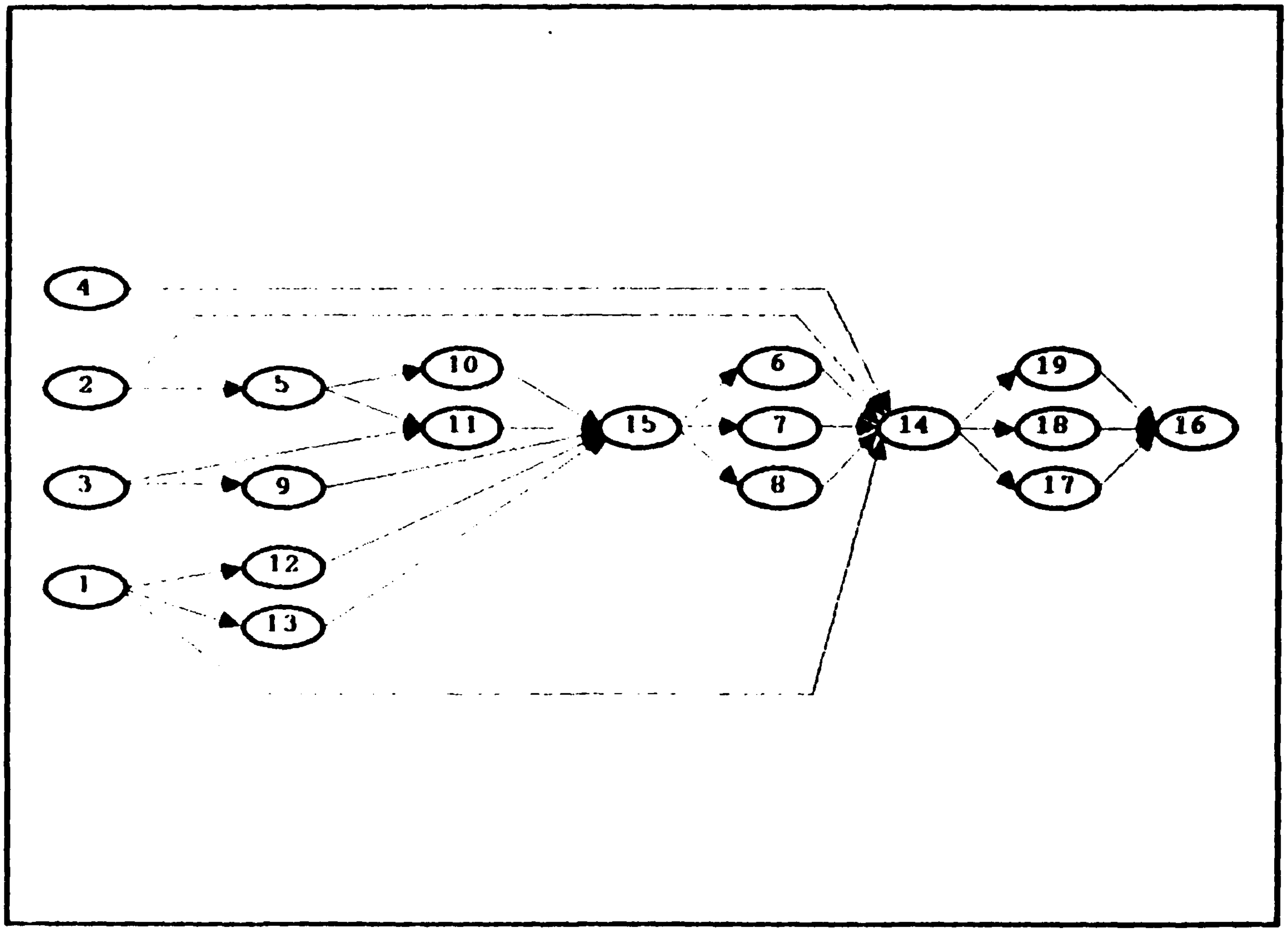

Figure 4 6. Network A Topology

close second with 8%. This is partially due to the fact that the survival rate of Arcs A1-14 and A4-14 are 1.0, thus diminishing the correlation of survival rate and maximum flow for those nodes positioned prior to Node 14.

Network B. Figure 4-7 on the following page shows the topology for Network B, while the link list is given in Appendix B. As in Network C, all components in Network B are independent, and like Network A, the node numbers in Figure 4-7 do not reflect changes due to arc equivalents or dummy source and sink nodes. This network has 177 paths and 60 proper cuts.

The results of the variance reduction tests are shown below in Table 4-8. The best reduction is 1% by Node 16. No other node or set of nodes reduces the variance by any measurable amount. This lack of significant reduction is largely due to the presence of three arcs that directly connect source nodes to sink nodes - A4-24, A6-25, and A6-26.

Table 4-8. Variance Reduction of Estimated Maximum Flow for Network B

Nodes in Control Subset	Estimated Maximum Flow	Variance	95% Conf. Interval
0	353.010	328.045	6.430
16	353.679	325.121	6.373
6	353.010	327.846	6.426
10	352.948	327.231	6.414
4,6	353.071	327.847	6.426
6,10	353.802	327.001	6.410

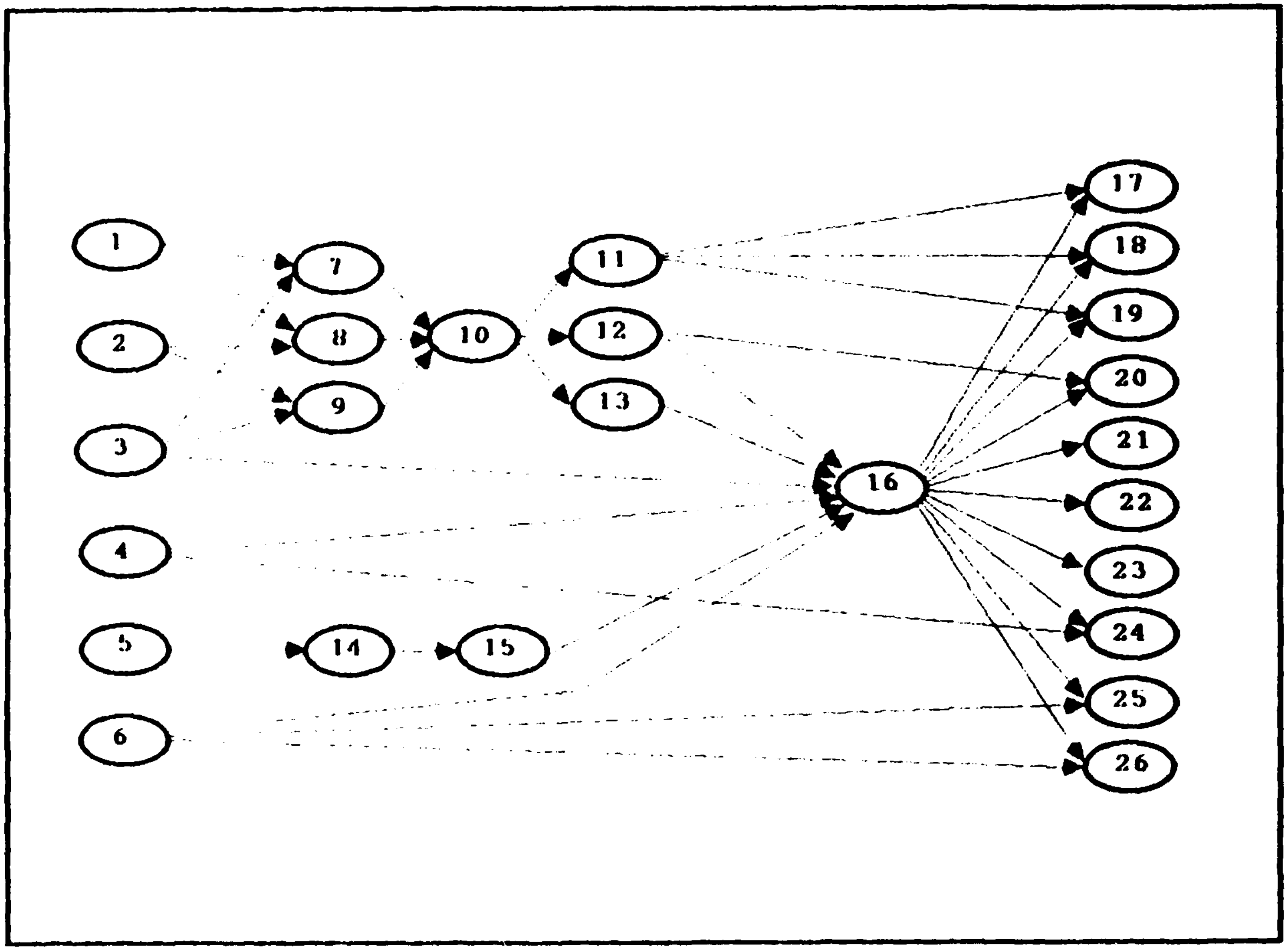

Figure 4-7. Network B Topology

Arc 4-24 alone has a capacity of 600 that is operative, on average, 50% of the time, thus contributing an expected flowof 300. This accounts for almost 85% of the overall maximum flow estimated by MAXFLO, thereby negating the influence of node survival on network performance. Thus, a strong case exists for including arc controls in future studies.

Control Variate Analysis. The previous results show a wide range of effectiveness in reducing the variance of the estimated maximum flow. Apparently, the ability of a control subset to reduce the variance is a function of network topology, and the network components' survival rates and capacities. This observation, of course, implies that a simple, significant, workable function of nodes does exist for use as an internal control variate.

The variance reduction shown in this paper is significant considering that only nodal scalar variables were investigated. Based on these results, further research into using both nodal and arc based controls should provide even larger reductions. All three networks indicate that arcs, to a various extent, influence network performance. Therefore, these components should be included in any further research of variance reduction of estimated flow of stochastic binary networks.

Finally, not only is RSM a powerful and insightful tool for stochastic network analysis, but strong empirical evidence is made for a unique relationship between the maximum flow response surface and control variate performance. Indeed, this correlation may even be more pronounced if true multiple control variates are compared to the response surface equations. Further research in this area seems promising, particularly in the area of improving the efficiency of experimental design screening.

V. Conclusions and Recommendations

This chapter presents a brief summary of this study's results and makes recommendations for future research.

Conclusions

This research offers several substantial improvements in analyzing stochastic binary network performance and improvement strategies. First, a simulation algorithm using proper cutsets for estimating maximum flow and reliability was developed to take advantage of networks that have fewer proper cuts than simple paths. The FORTRAN based program, called MAXFLO, also incorporates optional antithetic random number streams; variable simulation sample sizes; user-defined control variates; user-defined network component dependencies; automatic mean, variance, and confidence interval calculations for the estimated expected maximum flow; a point estimator for network reliability; options for saving and loading network cutsets; and options for examining and changing network parameters. The code was compiled and run on VAX mainframes and SUN workstations under VMS, UNIX, and SunOS operating systems. Thus, a high degree of portability was achieved among ANSI FORTRAN 77 environments.

Second, a simple function of nodes for use as internal control variates is shown to be a new, workable class of controls. The control variate experiments on three networks demonstrate that variance reduction as high as 24% is possible with a simple scalar measure of the expected number of surviving nodes in a selected group. Furthermore, closer analysis of the networks indicates that the use of arc variables as multiple controls appears very promising and warrants further research. Finally, empirical evidence suggests a node's utility as a control variate can be incorporated as an additional screening tool for experimental designs estimating the maximum flow response surface.

Finally, RSM was shown to be a powerful technique for analyzing and improving stochastic binary network performance. RSM provides a clear algebraic description of network flow and reliability, and how individual components influence its performance (as demonstrated by two example networks analyzed in this study). Moreover, the resulting polynomial equations provide a solid basis for use in optimization models. For example, the response surface equation can be used as an objective function in a linear programming model, subject to various constraints such as cost, survivability limitations, and network capacity restrictions. In short, RSM provides a method for rationally modelling the unpredictable and complicated behavior of a stochastic binary network.

Recommendations

In the process of meeting the objectives and answering the questions posed by this thesis, new questions were raised in regard to stochastic binary networks, and the areas of simulation, control variates, and RSM. The following list of future research recommendations summarizes the issues raised in this study.

First, the following items in simulation need further investigation:

1. Conduct a comparison of the pathset/labeling and the cutset algorithms in terms of simulation speed and efficiency. One valuable outcome of this research would be a heuristic guide for when to use one method over the other.

2. Research the effect of antithetic random number streams and antithetic pairs on maximum flow and reliability estimators; and, most importantly, on bias reduction.

3. Examine the use of stratified sampling as a method of reducing inherent bias due to unlikely network topologies.

4. Develop faster, more efficient algorithms for Monte Carlo simulation, and port the program to a microcomputer environment. Specific areas of improvement include sorting the cutset by likelihood of failure and more efficient storage of pathset and cutset files.

5. Expand the model to include multiple arcs between nodes. A further enhancement would be to expand simulation capability to include randomly capacitated components.

6. Explore the relationship of maximum flow and reliability across different network topologies. Further define their complementary nature, and perhaps extend this research to other measures of network performance.

Second, the following areas of control variates and variance reduction need further research:

1. Discover if additional variance reduction occurs if arc, as well as node, survival rates are placed in the control subset; and if so, how much.

2. Expand Item (1) from a scalar control to multiple controls, and compare results for any additional variance reduction.

3. Compare variance reduction results to current efforts in this area; specifically, those by Fishman (1983).

Finally, the following items of RSM warrant further research:

1. Investigate the advantages or disadvantages of using the Schruben-Margolin assignment rule for experimental designs of stochastic networks.

2. Expand RSM to other measures of network performance, i.e. shortest path.

3. Examine the feasibility of using control variates as a screening methodology in conjunction with, or as a substitute for, traditional screening designs.

4. Develop a "hybrid" screening design using both factor and group screening techniques as a method for examining all possible network variables.

5. Conduct further experiments on a variety of stochastic networks. Particularly, employ the response surface polynomials in other optimization models for specific problems.

Appendix A: Network A Link List

Component	Probability of Survival	Capacity
N1	1.0	-
N2	.3	-
N3	.7	-
N4	.5	-
N5	.8	-
N6	1.0	-
N7	.3	-
N8	.7	-
N9	.5	-
N10	.8	-
N11	1.0	-
N12	.3	-
N13	.7	-
N14	.5	-
N15	.8	-
N16	1.0	-
N17	.3	-
N18	.7	-
N19	.5	-
A1-12	1.0	1200
A1-13	1.0	1200
A1-14	1.0	1200
A2-14	.6	1200
A2-5	.3	1200
A5-10	.6	1200
A5-11	.7	1200
A3-11	1.0	1200
A3-9	1.0	1200
A4-14	1.0	1200
A7-14	.6	4800
A8-14	.3	4800
A6-14	.6	4800
A12-15	.7	4800
A13-15	1.0	4800
A9-15	1.0	4800
A11-15	1.0	4800
A10-15	.6	4800
A15-6	.3	4800
A15-7	.6	4800
A15-8	.7	4800
A14-19	1.0	4800
A14-18	1.0	4800

Component	Probability of Survival	Capacity
A14-17	1.0	4800
A17-16	.6	4800
A18-16	.3	4800
A19-16	.6	4800

Dependent Nodes and Arcs

Independent Component	Dependent Components
N8	N17
N15	N16
N7	N18
N6	N19
A15-6	A19-16
A15-7	A18-16
A15-8	A17-16
A6-14	A14-19
A7-14	A14-18
A8-14	A14-17

Appendix B: Network B Link List

Component	Probability of Survival	Capacity
N1	.70	-
N2	.15	-
N3	.03	-
N4	1.00	-
N5	1.00	-
N6	.04	-
N7	.40	-
N8	1.00	-
N9	.01	-
N10	.70	-
N11	.11	-
N12	1.00	-
N13	.06	-
N14	.09	-
N15	.18	-
N16	.07	-
N17	1.00	-
N18	1.00	-
N19	1.00	-
N20	1.00	-
N21	1.00	-
N22	1.00	-
N23	1.00	-
N24	1.00	-
N25	1.00	-
N26	1.00	-
A1-7	.80	150
A1-8	.80	200
A2-8	.50	750
A2-9	.50	750
A3-7	.80	200
A3-9	.50	750
A3-16	.60	150
A4-16	.80	200
A5-14	.80	1200
A6-16	.50	1200
A6-25	.60	75
A6-26	.60	75
A7-10	.50	1200
A8-10	.70	1200
A9-10	.50	2400
A10-11	.50	1200

Component	Probability of Survival	Capacity
A10-12	.70	1200
A10-13	.70	1200
A11-17	.50	1200
A11-18	.50	75
A11-19	.50	1200
A12-20	.50	1200
A12-16	.70	1200
A13-16	.70	600
A14-15	.80	1200
A15-16	.80	1200
A16-17	.60	75
A16-18	.60	75
A16-19	.60	75
A16-20	.60	75
A16-21	.60	75
A16-22	.60	75
A16-23	.60	75
A16-24	.60	75
A16-25	.60	75
A16-26	.60	75
Dependent Nodes and Arcs		
ALL NODES AND ARCS ARE INDEPENDENT		

Appendix C: Network C Link List

Component	Probability of Survival	Capacity
N1	.3	-
N2	.7	-
N3	.5	-
N4	.8	-
N5	1.0	-
N6	.3	-
N7	.7	-
N8	.5	-
N9	.8	-
N10	1.0	-
N11	.7	-
N12	.7	-
N13	.5	-
N14	.8	-
N15	1.0	-
N16	.3	-
N17	.7	-
N18	.5	-
N19	.8	-
N20	1.0	-
N21	.3	-
N22	.7	-
N23	.5	-
N24	.8	-
N25	1.0	-
N26	.3	-
N27	.7	-
N28	.5	-
N29	.8	-
N30	1.0	-
N31	.3	-
N32	.7	-
N33	.5	-
N34	.8	-
N35	1.0	-
N36	.3	-
N37	.7	-
N38	.5	-
N39	.8	-
A1-11	.9	1200
A2-11	1.0	2400
A3-7	1.0	1200

Component	Probability of Survival	Capacity
A4-11	.6	300
A5-8	.3	1200
A6-9	.6	1200
A7-10	.7	300
A8-11	.9	1200
A9-11	1.0	300
A10-11	1.0	300
A11-12	.9	9600
A11-23	.6	75
A11-24	.3	75
A11-38	.6	1200
A11-39	.7	1200
A12-13	1.0	4800
A12-14	1.0	4800
A12-15	.6	4800
A12-16	.3	4800
A12-17	.6	4800
A12-18	.7	2400
A12-19	.9	4800
A12-20	1.0	4800
A12-21	1.0	4800
A12-22	.6	2400
A13-23	.3	4800
A13-24	.6	4800
A13-25	.7	2400
A13-26	.9	1200
A13-27	1.0	1200
A13-28	1.0	1200
A13-29	.3	1200
A14-23	.6	4800
A14-24	.3	4800
A14-25	.6	2400
A14-26	.7	1200
A14-27	.9	1200
A14-28	1.0	1200
A14-29	1.0	1200
A15-31	.6	2400
A15-32	.3	1200
A16-30	.6	300
A16-31	.7	2400
A16-32	.9	1200
A16-33	1.0	300
A16-36	1.0	2400
A17-30	.6	1200
A17-33	.3	300
A17-34	.6	300

Component	Probability of Survival	Capacity
A18-38	.7	1200
A19-39	.9	1200
A20-35	1.0	2400
A21-36	1.0	2400
A22-37	.6	600

Dependent Nodes and Arcs
ALL NODES AND ARCS ARE INDEPENDENT

Appendix D: MAXFLO FORTRAN Source Code

```
      PROGRAM MAXFLO
*
*
*     VERSION 1.0
*
*     MAXIMUM FLOW AND NETWORK RELIABILITY
*     MONTE CARLO SIMULATION CODE
*
*     DESIGNED AND WRITTEN BY CAPTAIN THOMAS GLENN BAILEY
*     AS PART OF THESIS AFIT/GOR/88D-01  :  NOVEMBER 1988
*
*

*********************************************************
*
*     VARIABLE DEFINITIONS
*
*********************************************************

*
*                 GLOBAL
*

*     AC              -       MAX NUMBER OF ARCS
*     ADEPEND(AC)     -       ARC DEPENDENCIES ARRAY
*     APROB(AC)       -       ARC SURVIVAL PROB. ARRAY
*     CONTROL(ND)     -       CONTROL VARIATE ARRAY
*     CUT(RW,AC)      -       PROPER CUTSET ARRAY
*     DEPEND(ND)      -       NODE DEPENDENCIES ARRAY
*     HEAD(AC)        -       DESTINATION NODE FOR
*                             CUT(-,AC)
*     MUC             -       EXPECTED VALUE OF CONTROL
*     N               -       CURRENT NUMBER OF ARCS
*     ND              -       MAX NUMBER OF NODES
*     NDTOT           -       CURRENT NUMBER OF NODES
*     NPROB(ND)       -       NODE SURVIVAL PROB. ARRAY
*     NTAPE           -       DISK OR TAPE I/O #
*     NPRINT          -       LINE PRINTER #
*     P               -       CURRENT NUMBER OF PATHS
*     PATH(RW,AC)     -       PATHSET ARRAY
*     RW              -       MAX NUMBER OF CUTS OR PATHS
*     T               -       CURRENT NUMBER OF CUTS
*     TAIL(AC)        -       ORIG. NODE FOR CUT(-,AC)
*     TCNTL           -       NUMBER OF NODES IN CONTROL
*                             SUBSET
```

```
*
*                 MAIN PROGRAM
*

*     SHOW                -        MENU SELECTION VARIABLE

*
*                 CUTSET SUBROUTINE
*

*     ARCCAP(ND,ND)       -        NODE/INCIDENCE ARRAY WITH
*                                  ARC CAPACITIES
*     C                   -        PREVIOUS COLUMN
*     COL(AC)             -        COLUMN STATUS ARRAY
*     CYC                 -        PATH CYCLING FLAG
*     FLAG                -        GENERAL PURPOSE FLAG
*     I                   -        ROW POSITION
*     INN                 -        CONTAINMENT CHECK
*     J                   -        COLUMN POSITION
*     K                   -        GENERAL PURPOSE VARIABLE
*     L                   -        GENERAL PURPOSE VARIABLE
*     M                   -        GENERAL PURPOSE VARIABLE
*     MARK(ND,ND)         -        'SHADOW' OF ARCCAP(ND,ND)
*                                  RECORDS PREVIOUS PASSAGES
*                                  IN PATH SEARCH ALGORITHM
*     MATCOL(ND,ND)       -        'SHADOW' OF ARCCAP(ND,ND)
*                                  RECORDS PROPER ORDER OF ARCS
*                                  FOR APROB(AC),HEAD(AC), AND
*                                  TAIL(AC) ARRAYS
*     OUTT                -        CONTAINMENT CHECK
*     PP                  -        TEMPORARY NUMBER OF PATHS
*     R                   -        PREVIOUS ROW
*     ROW(RW)             -        ROW STATUS ARRAY
*     SK                  -        DUMMY SINK NODE STATUS
*     SR                  -        DUMMY SOURCE NODE STATUS
*     XB(RW)              -        PATHSET INVERSION STATUS
*                                  ARRAY
*     XB1(RW)             -                 "
*     XR(RW)              -                 "
*     XR1(RW)             -                 "

*
*                 DIAG SUBROUTINE
*

*     HD, I, J            -        GENERAL PURPOSE VARIABLES
*     K, TL               -                 "
```

```
*
*                 SIMULATE SUBROUTINE
*

*     I,J,H,K,L       -       GENERAL PURPOSE VARIABLES
*     ANTI            -       ANTITHETIC RANDOM NUMBER
*                             STREAM FLAG
*     BHAT            -       ESTIMATED COVARIANCE/CONTROL
*                             VARIANCE RATIO
*     C11             -       UNCONTROLLED CONF. INTERVAL
*     C12             -       CONTROLLED CONF. INTERVAL
*     CMEAN           -       MEAN OF NODES IN CONTROL
*                             SUBSET THAT SURVIVE
*     CNTL            -       NUMBER OF NODES ASSIGNED
*                             TO CONTROL SUBSET
*     COL(AC)         -       ARC STATUS ARRAY DURING
*                             SIMULATION
*     COVAR           -       COVARIANCE OF RESPONSE
*                             TO CONTROL
*     CTOT            -       SUMMATION OF ALL SAMPLES'
*                             NUMBER OF NODES SURVIVING
*     CVAR            -       CONTROL VARIANCE
*     DNODE(ND)       -       STATUS OF DEPENDENT NODES
*                             IN CURRENT SIMULATION
*     FLAG            -       RESERVED
*     MEAN            -       RESERVED
*     MINCUT          -       VALUE OF CURRENT CUT IN
*                             CUTSET OF THE CURRENT
*                             SIMULATION SAMPLE
*     MUY             -       CONTROLLED POINTE ESTIMATE OF
*                             RESPONSE
*     RDM             -       CURRENT RANDOM NUMBER DRAW
*     RLBL            -       UNCONTROLLED POINT ESTIMATE
*                             OF NETWORK RELIABILITY
*     S11             -       CONTROLLED ESTIMATE OF
*                             RESPONSE STANDARD DEVIATION
*     SEED            -       RANDOM NUMBER SEED
*     SIM             -       USER-DEFINED SIMULATION
*                             SAMPLE SIZE (100,000 MAX)
*     STAT(SM,2)      -       SIMULATION'S SAMPLE RESULTS
*     VAR             -       VARIANCE OF UNCONTROLLED
*                             RESPONSE
*     YMEAN           -       MEAN RESPONSE
*     YVAR            -       VARIANCE OF CONTROLLED
*                             RESPONSE
*     YTOT            -       SUMMATION OF ALL SAMPLES'
*                             RESPONSES
```

```
*
*                 SAVECUT SUBROUTINE
*

*     FNAME              -       CUTSET FILENAME
*     I, j=J             -       GENERAL PURPOSE VARIABLES

*
*                 CHGCUT SUBROUTINE
*

*     ARC                -       ARC NUMBER
*     CAP                -       NEW CAPACITY
*     FLAG               -       NODE CHANGES
*     HD                 -       ARC DESTINATION
*     I,J,K              -       GENERAL PURPOSE VARIABLES
*     PROB               -       NEW PROBABILITY OF SURVIVAL
*     TL                 -       ARC ORIGIN

*
*                 DPND SUBROUTINE
*

*     HD                 -       ARC DESTINATION
*     I,J,K              -       GENERAL PURPOSE VARIABLE
*     TL                 -       ARC ORIGIN

**********************************************************
*
*                 MAIN PROGRAM
*
**********************************************************

*
*
*     DEFINITION AND DECLARATION OF VARIABLES
*
*

*
*     PARAMETERS
*

      INTEGER AC,RW,ND,NTAPE,NPRINT
      PARAMETER (RW=1000,AC=80,ND=50,NTAPE=7,NPRINT=8)
```

```
*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T,P,TCNTL
      INTEGER CUT(RW,AC), PATH(RW,AC), CONTROL(ND)
      INTEGER HEAD(AC), TAIL(AC), DEPEND(ND), ADEPEND(AC)
      REAL    NPROB(ND), APROB(AC)
      REAL    MUC
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/PATH2/P,PATH
      COMMON/CUT2/MUC,TCNTL,CONTROL
      COMMON/CUT3/DEPEND,ADEPEND

*
*     LOCAL VARIABLES
*
      INTEGER SHOW

*
*     MAIN CONTROL MENU
*

   50 PRINT*,'1. Enter Network.'
      PRINT*,'2. Save/Retrieve Network.'
      PRINT*,'3. Simulate.'
      PRINT*,'4. Diagnostics.'
      PRINT*,'5. Change Network Parameters.'
      PRINT*,'6. Enter Node Dependencies.'
      PRINT*,'7. Exit.'
      READ*,SHOW
      IF (SHOW.EQ.1) THEN
              CALL CUTSET
          ELSE IF (SHOW.EQ.2) THEN
              CALL SAVECUT
          ELSE IF (SHOW.EQ.3) THEN
              CALL SIMULATE
          ELSE IF (SHOW.EQ.4) THEN
              CALL DIAG
          ELSE IF (SHOW.EQ.5) THEN
              CALL CHGCUT
          ELSE IF (SHOW.EQ.6) THEN
              CALL DPND
          ELSE IF (SHOW.EQ.7) THEN
              GO TO 100
          ELSE
              GO TO 50
      END IF
      GO TO 50
  100 END
```

```
*
*
*     RANDOM NUMBER GENERATOR
*
*

      FUNCTION RANDOM(IX)
      INTEGER A,P,IX,B15,B16,XHI,XALO,LEFTLO,FHI,K
      DATA A/16807/,B15/32768/,B16/65536/,P/2147483647/
      XHI=IX/B16
      XALO=(IX-XHI*B16)*A
      LEFTLO=XALO/B16
      FHI=XHI*A+LEFTLO
      K=FHI/B15
      IX=(((XALO-LEFTLO*B16)-P)+(FHI-K*B15)*B16)+K
      IF(IX.LT.0)IX=IX+P
      RANDOM=FLOAT(IX)*4.656612875E-10
      RETURN
      END

*
*
*     NETWORK ENTRY and
*     PATHSET AND CUTSET GENERATION SUBROUTINE
*
*

      SUBROUTINE CUTSET

*
*     PARAMETERS
*

      INTEGER AC,RW,ND
      PARAMETER (RW=1000,AC=80,ND=50)

*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T,P
      INTEGER CUT(RW,AC), PATH(RW,AC)
      INTEGER HEAD(AC), TAIL(AC)
      REAL NPROB(ND), APROB(AC)
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/PATH2/P,PATH
```

```
*
*     LOCAL VARIABLES
*

      INTEGER I,J,R,C,FLAG,CYC
      INTEGER ROW(RW), COL(AC)
      INTEGER ARCCAP(ND,ND), MARK(ND,ND), MATCOL(ND,ND)
      INTEGER INN,OUTT,XR(RW),XR1(RW),XB(RW),XB1(RW)
      INTEGER K,L,M,PP,SR,SK

*
*     INPUT NETWORK TOPOLOGY
*

*
*     NODES - IDENTIFY, CAPACITY, SURVIVAL PROBABILITY
*

      PRINT*, 'Please enter the total number of nodes: '
      READ*, NDTOT
      PRINT*, 'Source node ID number must be 1.'
      PRINT*, ' '
      PRINT*, 'Terminal node ID number must equal total
     +number of nodes.'
      DO 400 I = 1,NDTOT
          PRINT*, 'Node ID number: ', I
          PRINT*, 'Enter node ',I,' capacity: '
          READ*, ARCCAP(I,I)
          PRINT*, 'Enter node ',I,' probability of
     +    survival: '
          READ*,NPROB(I)
          IF (NPROB(I).GT.1.) NPROB(I) = 1.0
          IF (NPROB(I).LT.0.) NPROB(I) = 0.0
  400 CONTINUE

*
*     IDENTIFY 'DUMMY' SOURCE AND SINK NODES
*

      SR = 0
      SK = 0
      PRINT*,'Enter 1 if source node 1 is a DUMMY node:'
      READ*,SR
      PRINT*,'Enter 1 if sink node is a DUMMY node:'
      READ*,SK
```

```
*
*     ARCS - IDENTIFY, CAPACITY, SURVIVAL PROBABILITY
*

      FLAG = 0
  420 IF (FLAG.NE.1) THEN
          PRINT*, 'Enter 0 to enter arc.'
          PRINT*, 'Enter 1 to exit arc entry.'
          READ*, FLAG
          IF (FLAG.NE.1) THEN
            PRINT*, 'Enter arc head:'
            READ*, J
            PRINT*, 'Enter arc tail: '
            READ*,I
            IF (I.GT.NDTOT .OR. J.GT.NDTOT) THEN
              PRINT*,'NODE OUT OF RANGE!'
              GO TO 420
            END IF
            PRINT*, 'Enter arc capacity: '
            READ*, ARCCAP(I,J)
            GO TO 420
          END IF
      END IF

*
*     CALCULATE PATHSET/CUTSET MATRICES COLUMNS
*

      N = 1
      DO 428 I = 1,NDTOT
          DO 425 J = 1,NDTOT
              IF (ARCCAP(I,J).GT.0) THEN
                MATCOL(I,J) = N
                HEAD(N) = J
                TAIL(N) = I
                N = N + 1
              END IF
  425     CONTINUE
  428 CONTINUE
      N = N - 1
```

```
*
*     CALCULATE ALL ACYCLIC PATHS
*     FROM NODE 1 (SOURCE) TO NODE NDTOT (SINK)
*     (REF: SHIER AND WHITED: IEEE TRANS. ON
*           RELIABILITY, OCTOBER 1985)
*

      P = 1
      I = 1
      J = 1
      FLAG = 0
      CYC = 0
  430 IF (FLAG.EQ.0) THEN
  435     IF (J.LT.NDTOT) THEN

*             FIND NEXT ARC SEGMENT

              R = I
              C = J
              I = J
              J = 1
  440         IF (ARCCAP(I,J).GT.0.AND.MARK(I,J).EQ.0
     +             .AND.I.NE.J.OR.J.GT.NDTOT) THEN
                      GO TO 445
                  ELSE
                      J = J + 1
                      GO TO 440
              END IF

*             RECORD ARC SEGMENT

  445         IF (MATCOL(I,J).GT.0)
     +            PATH(P,MATCOL(I,J)) = ARCCAP(I,J)
              IF (ARCCAP(I,I).GT.0)
     +            PATH(P,MATCOL(I,I)) = ARCCAP(I,I)

*             MARK ROW(I) TO GUARD AGAINST CYCLING

              IF (ROW(I).EQ.1) THEN
                  J = NDTOT + 1
                  CYC = 1
              END IF
              ROW(I) = 1
          GO TO 435
          END IF

*         MARK SEGMENT

          MARK(R,C) = 1
```

```
*         CLEAR MARK

          IF (I.GT.1.AND.CYC.EQ.0) THEN
              DO 460 L = 1,NDTOT
                  MARK(I,L) = 0
  460         CONTINUE
          END IF

*         DETERMINE IF PATH EXISTS

          IF (J.EQ.NDTOT) THEN
                  P = P + 1
                  CALL ENDSNODE(I,J,NDTOT,FLAG,CYC,ROW)
              ELSE
                  CALL ENDSNODE(I,J,NDTOT,FLAG,CYC,ROW)
                  DO 465 M = 1,AC
                      PATH(P,M) = 0
  465             CONTINUE
          END IF
          I = 1
          J = 1
      GO TO 430
      END IF
      P = P - 1

*
*     ELIMINATE DUMMY SOURCE AND DUMMY SINK NODES
*

      IF (SR.EQ.1) THEN
        DO 475 J = 1,N
          IF (TAIL(J).EQ.1) THEN
            DO 470 I = 1,P
              PATH(I,J) = 0
  470       CONTINUE
          END IF
  475   CONTINUE
      END IF

      IF (SK.EQ.1) THEN
        DO 485 J = 1,N
          IF (HEAD(J).EQ.NDTOT) THEN
            DO 480 I = 1,P
              PATH(I,J) = 0
  480       CONTINUE
          END IF
  485   CONTINUE
      END IF
```

```
*
*     REDUCE PATH MATRIX
*

*     ZERO OUT ROW(x) AND COL(x) ARRAYS

      DO 500 I = 1,RW
          ROW(I) = 0
  500 CONTINUE

      DO 510 J = 1,AC
          COL(J) = 0
  510 CONTINUE

*     IDENTIFY ZERO COLUMNS IN PATH MATRIX

      DO 550 J = 1,N
          I = 1
  535     IF (PATH(I,J).GT.0.OR.I.GT.P) THEN
                  GO TO 540
              ELSE
                  I = I + 1
                  GO TO 535
          END IF
  540     IF (I.GT.P) COL(J) = 1
  550 CONTINUE

*     PACK PATH MATRIX

      J = 1
  555 IF (J.LE.N) THEN
          IF (COL(J).EQ.1) THEN
                  DO 570 K = J,N
                      DO 560 I = 1,P
                          PATH(I,K) = PATH(I,K+1)
  560                 CONTINUE
                      COL(K) = COL(K+1)
                      HEAD(K) = HEAD(K+1)
                      TAIL(K) = TAIL(K+1)
                      APROB(K) = APROB(K+1)
  570             CONTINUE
                  N = N - 1
                  GO TO 555
              ELSE
                  J = J + 1
                  GO TO 555
          END IF
      END IF
```

```
*
*     ENTER ARC PROBABILITIES
*

      DO 650 I = 1,N
          PRINT*,'Enter arc probability of survival for
     +    arc',I
          PRINT*,'HEAD:',HEAD(I)
          PRINT*,'TAIL:',TAIL(I)
          READ*,APROB(I)
          IF (APROB(I).GT.1.) APROB(I) = 1.0
          IF (APROB(I).LT.0.) APROB(I) = 0.0
  650 CONTINUE

*
*
*     CALCULATE MIN-CUTS
*
*


*
*     INITIALIZE MATRIX
*

      T = 0
      DO 710 J = 1, N
          IF (PATH(1,J).GT.0) THEN
              T = T + 1
              CUT(T,J) = PATH(1,J)
          END IF
  710 CONTINUE

*
*     LOOP CONTROL FOR MULTIPLYING PATH POLYNOMIAL
*     CLEAR SW3 DATA ARRAYS
*

      PRINT*,'THERE ARE THIS MANY PATHS:',P

      DO 890 PP = 2, P

      DO 720 I = 1,RW
        ROW(I) = 0
        XB(I)  = 0
        XB1(I) = 0
        XR(I)  = 0
        XR1(I) = 0
  720 CONTINUE

      DO 730 J = 1,N
        COL(J)  = 0
  730 CONTINUE
```

```
*
*     DETERMINE XR AND XR1 MEMBERS OF A
*
      DO 750 J = 1,N
        IF (PATH(PP,J).GT.0) THEN
          DO 740 I = 1,T
            IF (PATH(PP,J).EQ.CUT(I,J)) THEN
              XR(I) = XR(I) + 1
              XR1(I) = J
            END IF
  740     CONTINUE
        END IF
  750 CONTINUE

*
*     MARK COLUMN OF PATH TO INDICATE XR1
*

      DO 760 I = 1,T
        IF (XR(I).EQ.1) THEN
          COL(XR1(I)) = 3
        END IF
  760 CONTINUE

*
*     DETERMINE XB
*

      DO 770 I = 1,T
        DO 765 J = 1,N
          IF (CUT(I,J).GT.0) THEN
            XB(I) = XB(I) + 1
            XB1(I) = J
          END IF
  765   CONTINUE
  770 CONTINUE

      DO 775 I = 1,T
        IF ( (XB(I).EQ.1).AND.(PATH(PP,XB1(I)).GT.0) ) THEN
            COL(XB1(I)) = 1
          ELSE
            XB(I) = 0
        END IF
  775 CONTINUE

*
*     MULTIPLY PATH TO CUTSET
*

      TT = T
      DO 795 J = 1,N
```

```
*     AVOID XB IN a

        IF ( (PATH(PP,J).GT.0).AND.(COL(J).NE.1) ) THEN
          DO 790 I = 1,TT

*           AVOID XB,XR,XR1 IN A

            IF ( (XR(I).EQ.0).AND.(XB(I).EQ.0) ) THEN
              T = T + 1
              DO 780 L = 1,N
                CUT(T,L) = CUT(I,L)
  780         CONTINUE
              CUT(T,J) = PATH(PP,J)

*             IDENTIFY RESIDUAL XR1 IN A

              IF (COL(J).EQ.3) THEN
                XR1(T) = J
              END IF
            END IF
  790     CONTINUE
        END IF
  795 CONTINUE

*
*     CHECK FOR XR1 CONTAINMENT
*

      DO 815 J = 1,N
        DO 810 I = 1,TT
          IF (XR1(I).EQ.J) THEN
            DO 805 K = TT + 1,T
              IF (XR1(K).EQ.J) THEN
                OUTT = 0
                INN  = 0
                L = 1
  800           IF (L.LE.N) THEN
                  IF (CUT(I,L).GT.CUT(K,L)) THEN
                      OUTT = OUTT + 1
                    ELSEIF (CUT(K,L).GT.CUT(I,L)) THEN
                      INN = INN + 1
                  END IF
                  IF ( (OUTT.GT.0).AND.(INN.GT.0) ) L = N
                  L = L + 1
                  GO TO 800
                END IF
                IF ((INN.GT.0).AND.(OUTT.EQ.0)) ROW(K) = 1
              END IF
  805       CONTINUE
          END IF
  810   CONTINUE
  815 CONTINUE
```

```
*
*     MOVE XB,XR,XR1 TERMS INTO A
*

      DO 825 I = 1,TT
        IF ( (XR(I).GT.0).OR.(XB(I).GT.0) )THEN
          T = T + 1
          DO 820 J = 1,N
            CUT(T,J) = CUT(I,J)
  820     CONTINUE
        END IF
  825 CONTINUE

*
*     MOVE A TO TOP OF CUT MATRIX
*

      K = 1
      DO 835 I = TT + 1,T
        DO 830 J = 1,N
          CUT(K,J) = CUT(I,J)
  830   CONTINUE
        ROW(K) = ROW(I)
        K = K + 1
  835 CONTINUE
      T = K - 1

*
*     PACK CUT
*

      I = 1
  840 IF (I.LE.T) THEN
        IF (ROW(I).GT.0) THEN
          DO 850 K = I+1,T
            DO 845 J = 1,N
              CUT(K-1,J) = CUT(K,J)
  845       CONTINUE
          ROW(K-1) = ROW(K)
  850     CONTINUE
          I = I - 1
          T = T - 1
        END IF
        I = I + 1
        GO TO 840
      END IF

  890 CONTINUE
```

```
*     REMINDER OF NODE AND ARC DEPENDENCIES

      PRINT*,'REMINDER: IF NODE OR ARC DEPENDENCIES EXIST,
     +ENTER THEM'
      PRINT*,'SEPARATELY USING ITEM (6) IN MAIN MENU.'

      END

*
*
*     SUBROUTINE ENDSNODE
*
*

      SUBROUTINE ENDSNODE(I,J,NDTOT,FLAG,CYC,ROW)

*
*     PARAMETER
*

      INTEGER RW
      PARAMETER (RW=1000)

*
*     ARRAY DECLARATION
*

      INTEGER M
      INTEGER ROW(RW)

      IF (I.EQ.1.AND.CYC.EQ.1.AND.J.EQ.NDTOT) THEN
          FLAG = 1
      END IF
      IF (I.EQ.1.AND.CYC.EQ.0.AND.J.GE.NDTOT) THEN
          FLAG = 1
      END IF
      DO 590 M = 1,NDTOT
          ROW(M) = 0
  590 CONTINUE
      CYC = 0
      END
```

```
*
*
*     NETWORK PARAMETERS DIAGNOSTIC SUBROUTINE
*
*

      SUBROUTINE DIAG

*
*     PARAMETERS
*

      INTEGER AC,RW,ND
      PARAMETER (RW=1000,AC=80,ND=50)

*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T,P,TCNTL
      INTEGER CUT(RW,AC), PATH(RW,AC), CONTROL(ND)
      INTEGER HEAD(AC), TAIL(AC), DEPEND(ND), ADEPEND(AC)
      REAL NPROB(ND), APROB(AC), MUC
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/PATH2/P,PATH
      COMMON/CUT2/MUC,TCNTL,CONTROL
      COMMON/CUT3/DEPEND,ADEPEND

*
*     LOCAL VARIABLE
*

      INTEGER SHOW,I,J,K,HD,TL

  600 PRINT*,'1. Show pathset (Not available if cutset
     +retrieved).'
      PRINT*,'2. Show cutset.'
      PRINT*,'3. Show individual network components.'
      PRINT*,'4. Show control variate subset.'
      PRINT*,'5. Show node and arc dependencies.'
      PRINT*,'6. Exit diagnostics.'
      READ*,SHOW
      IF (SHOW.EQ.1) THEN
              GO TO 910
          ELSEIF (SHOW.EQ.2) THEN
              GO TO 920
          ELSEIF (SHOW.EQ.3) THEN
              GO TO 610
          ELSEIF (SHOW.EQ.4) THEN
              GO TO 700
          ELSEIF (SHOW.EQ.5) THEN
              GO TO 800
          ELSEIF (SHOW.EQ.6) THEN
              GO TO 990
          ELSE
              GO TO 600
      END IF

*     SHOW INDIVIDUAL NETWORK PARAMETERS

  610 PRINT*,'Enter 0 for node.'
      PRINT*,'Enter 1 for arc.'
      PRINT*,'Enter 2 to return to menu.'
      READ*,SHOW

      IF (SHOW.EQ.0) GO TO 630
      IF (SHOW.EQ.1) GO TO 670
      IF (SHOW.EQ.2) GO TO 600
      GO TO 600

*     SHOW NODE PARAMETERS

  630 PRINT*,'Enter node.'
      READ*,K
      IF (K.GT.NDTOT .OR. K.LE.0) THEN
          PRINT*,'NODE DOES NOT EXIST'
          GO TO 610
      END IF

*     DETERMINE IF NODE IS CAPACITATED

      J = 0
      I = 1
  635 IF (I.LE.N) THEN
          IF (HEAD(I).EQ.K .AND. TAIL(I).EQ.K) THEN
              J = I
              I = N
          END IF
          I = I + 1
          GO TO 635
      END IF
      SHOW = 0
      IF (J.GT.0) THEN
            I = 1
  640       IF (I.LE.T) THEN
              IF (CUT(I,J).GT.0) THEN
                  SHOW = CUT(I,J)
                  I = T
              END IF
              I = I + 1
              GO TO 640
            END IF
          ELSE
            SHOW = 0
      END IF

      PRINT*,'NODE:',K
      PRINT*,'CAPACITY:',SHOW
      PRINT*,'PROB. OF SURVIVAL:',NPROB(K)
      GO TO 610

*     SHOW ARC PARAMETERS

  670 PRINT*,'Enter arc head.'
      READ*,HD
      PRINT*,'Enter arc tail.'
      READ*,TL
      J = 0
      I = 1
  675 IF (I.LE.N) THEN
          IF (HEAD(I).EQ.HD .AND. TAIL(I).EQ.TL) THEN
              J = I
              I = N
          END IF
          I = I + 1
          GO TO 675
      END IF

      IF (J.EQ.0) THEN
          PRINT*,'ARC DOES NOT EXIST!'
          GO TO 610
      END IF

      SHOW = 0
      I = 1
  680 IF (I.LE.T) THEN
          IF (CUT(I,J).GT.0) THEN
            SHOW = CUT(I,J)
            I = T
          END IF
          I = I + 1
          GO TO 680
      END IF
      PRINT*,'Arc head:',HD
      PRINT*,'Arc tail:',TL
      PRINT*,'Capacity:',SHOW
      PRINT*,'Probability of survival:',APROB(J)
      GO TO 610

*     SHOW CONTROL VARIATE SUBSET

  700 PRINT*,'Expected number of nodes in subset to
     +survive:',MUC
      PRINT*,'Total number of nodes in control
     +subset:',TCNTL

      DO 710 I = 1,NDTOT
          IF (CONTROL(I).EQ.1) PRINT*,I
  710 CONTINUE

      GO TO 600

*     NODE/ARC DEPENDENCY MENU

  800 PRINT*,'Enter 0 for node dependencies.'
      PRINT*,'Enter 1 for arc dependencies.'
      PRINT*,'Enter 2 to return to menu.'
      READ*,I

      IF (I.EQ.0) GO TO 805
      IF (I.EQ.1) GO TO 840
      IF (I.EQ.2) GO TO 600
      GO TO 800

*     SHOW NODE DEPENDENCIES

  805 DO 830 I = 1,NDTOT

          K = 0
          J = 1
  810     IF (J.LE.NDTOT) THEN
              IF (DEPEND(J).EQ.I) THEN
                K = I
                J = NDTOT
              END IF
              J = J + 1
              GO TO 810
          END IF
          IF (K.GT.0) THEN
              PRINT*,'The following nodes are dependent.'
              DO 820 J = NDTOT,1,-1
                IF (DEPEND(J).EQ.K) PRINT*,J
  820         CONTINUE
              PRINT*,K
              PRINT*,' '
              PRINT*,'Enter any number to continue:'
              READ*,SHOW
          END IF

  830 CONTINUE
      GO TO 800

*     SHOW ARC DEPENDENCIES

  840 DO 890 I = 1,N

          K = 0
          J = 1
  850     IF (J.LE.N) THEN
              IF (ADEPEND(J).EQ.I) THEN
                K = I
                J = N
              END IF
              J = J + 1
              GO TO 850
          END IF

          IF (K.GT.0) THEN
              PRINT*,'The following arcs are dependent.'
              DO 860 J = N,1,-1
                IF (ADEPEND(J).EQ.K) THEN
                  PRINT*,'HEAD:',HEAD(J)
                  PRINT*,'TAIL:',TAIL(J)
                  PRINT*,' '
                END IF
  860         CONTINUE
              PRINT*,'HEAD:',HEAD(K)
              PRINT*,'TAIL:',TAIL(K)
              PRINT*,' '
              PRINT*,'Enter any number to continue:'
              READ*,SHOW
          END IF

  890 CONTINUE
      GO TO 800
```

```
*     SHOWPATH

 910 PRINT*,'Enter path number'
     READ*,I
     PRINT*,'There are ',P,' paths.'
       PRINT*,'Path No.',I
     DO 915 J = 1,N
         IF (PATH(I,J).GT.0) PRINT*,TAIL(J),'  ',HEAD(J)
 915 CONTINUE
     GO TO 600

*     SHOWCUT

 920 PRINT*,'Enter cut number'
     READ*,I
     PRINT*,'There are ',T,' cuts.'
     PRINT*,'TAIL',' ','HEAD',' ','CAPACITY',' ','PROB'
       PRINT*,'Cut No.',I
     DO 925 J = 1,N
       IF (CUT(I,J).GT.0) THEN
       PRINT*,TAIL(J),' ',HEAD(J),' ',CUT(I,J),'
    +  ',APROB(J)
     END IF
 925 CONTINUE
     GO TO 600

 990 END
```

```
*
*
*     CUTSET FILE SAVE AND RETRIEVAL SUBROUTINE
*
*

      SUBROUTINE SAVECUT

*
*     PARAMETERS
*

      INTEGER AC,RW,ND,NTAPE,NPRINT
      PARAMETER (RW=1000,AC=80,ND=50,NTAPE=7,NPRINT=8)

*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T,TCNTL
      INTEGER CUT(RW,AC), CONTROL(ND), ADEPEND(AC)
      INTEGER HEAD(AC), TAIL(AC), DEPEND(ND)
      REAL NPROB(ND), APROB(AC), MUC
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/CUT2/MUC,TCNTL,CONTROL
      COMMON/CUT3/DEPEND,ADEPEND

*
*     LOCAL VARIABLES
*

      CHARACTER*8 FNAME
      INTEGER I,J

*
*     MENU
*

   10 PRINT*,'Enter 0 to Save : Enter 1 to Retrieve : Enter
     +2 to Exit'
      READ*,I

      IF (I.EQ.0) THEN
              GO TO 30
          ELSEIF (I.EQ.1) THEN
              GO TO 100
          ELSEIF (I.EQ.2) THEN
              GO TO 220
          ELSE
              GO TO 10
      END IF
```

```
*
*     SAVE CUTSET
*

   30 PRINT*,'Enter Cutset Filename to Save:'
      READ*,FNAME

      OPEN (UNIT=NTAPE,FILE=FNAME,STATUS='NEW',ERR=199)

      WRITE (NTAPE,185,ERR=199) NDTOT,N,T

      DO 50 I = 1,NDTOT
          WRITE (NTAPE,188,ERR=199) NPROB(I), DEPEND(I)
   50 CONTINUE

      DO 55 J = 1,N
          WRITE (NTAPE,187,ERR=199)
     +           APROB(J),TAIL(J),HEAD(J),ADEPEND(J)
   55 CONTINUE

      DO 65 I = 1,T
        DO 60 J = 1,N
          WRITE (NTAPE,184,ERR=199) CUT(I,J)
   60   CONTINUE
   65 CONTINUE

      GO TO 180

*
*     RETRIEVE CUTSET
*

  100 PRINT*,'Enter Cutset Filename to Retrieve:'
      READ*,FNAME

      OPEN (UNIT=NTAPE,FILE=FNAME,STATUS='OLD',ERR=199)

      READ (NTAPE,185,ERR=199) NDTOT,N,T

      DO 150 I = 1,NDTOT
          READ (NTAPE,188,ERR=199) NPROB(I), DEPEND(I)
  150 CONTINUE

      DO 155 J = 1,N
          READ (NTAPE,187,ERR=199)
     +          APROB(J),TAIL(J),HEAD(J),ADEPEND(J)
  155 CONTINUE
```

```
      DO 165 I = 1,T
        DO 160 J = 1,N
          READ (NTAPE,184,ERR=199) CUT(I,J)
  160   CONTINUE
  165 CCNTINUE

  180 CLOSE (UNIT=NTAPE,ERR=199)

*     CLEAR CONTROL ARRAY

      MUC = 0.
      TCNTL = 0
      DO 200 I = 1,ND
          CONTROL(I) = 0
  200 CONTINUE
      PRINT*,'CONTROL VARIATES ELIMINATED.'

      GO TO 210

*
*     I/O FORMAT STATEMENTS
*

  184 FORMAT (I6)
  185 FORMAT (3I6)
  186 FORMAT (F8.6)
  187 FORMAT (F8.6,I6,I6,I6)
  188 FORMAT (F8.6,I4)

*
*     TERMINATION/ERROR CHECK ROUTINES
*

  199 PRINT*,'Error occurred in file transfer.'
      GO TO 220

  210 PRINT*,'File transfered.'

  220 END
```

```
*
*
*     MONTE CARLO SIMULATION SUBROUTINE
*
*

      SUBROUTINE SIMULATE

*
*     PARAMETERS
*

      INTEGER AC,RW,ND,SM,UPBOUND
      PARAMETER (RW=1000,AC=80,ND=50,
     +           SM=100000,UPBOUND=100000)

*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T,TCNTL
      INTEGER CUT(RW,AC), DEPEND(ND), ADEPEND(AC)
      INTEGER HEAD(AC), TAIL(AC), CONTROL(ND)
      REAL    NPROB(ND), APROB(AC), MUC
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/CUT2/MUC,TCNTL,CONTROL
      COMMON/CUT3/DEPEND,ADEPEND

*
*     LOCAL VARIABLES
*

      INTEGER I,J,K,L,H,CNTL,SIM,ANTI,MINCUT,FLAG
      INTEGER COL(AC), STAT(SM,2), DNODE(ND)
      REAL SEED,RDM,VAR,MEAN,RLBL,CMEAN,
      REAL YMEAN,YVAR,CVAR,COVAR
      REAL BHAT,MUY,S11,CI1,CI2,YTOT,CTOT

*
*     MENU
*

   50 PRINT*,'Enter 0 to continue simulation:'
      PRINT*,'Enter 1 to quit:'
      READ*, I
      IF (I.EQ.1) GO TO 500

      PRINT*,'Enter simulation sample size (100k maximum
     +allowed).'
```

```
      READ*,SIM
      IF (SIM.GT.100000) SIM = 100000
      IF (SIM.LE.1)      SIM = 200

      PRINT*,'Enter random seed.'
      READ*,SEED
      IF (SEED.LE.0.) SEED = 44645361.

      PRINT*,'Enter 0 for REGULAR random number stream:'
      PRINT*,'Enter 1 for ANTITHETIC random number stream:'
      READ*,ANTI

*
*     CONTROL VARIATE MENU
*

   55 PRINT*,'Enter 0 for no change in control variate:'
      PRINT*,'Enter 1 for no (0) control variates:'
      PRINT*,'Enter 2 to enter new control variates:'
      PRINT*,'NOTE: First simulation run is defaulted to 0
     +C.V.'
      READ*,I
      IF (I.EQ.0) THEN
              GO TO 75
          ELSEIF (I.EQ.1) THEN
              GO TO 70
          ELSEIF (I.EQ.2) THEN
              GO TO 60
          ELSE
              GO TO 55
      END IF

*
*     NEW CONTROL VARIATES
*

   60 MUC   = 0.
      TCNTL = 0
      DO 62 J = 1,NDTOT
          CONTROL(J) = 0
   62 CONTINUE

   63 PRINT*,'Enter 0 for another control variate:'
      PRINT*,'Enter 1 to exit:'
      READ*,I
      IF (I.EQ.0) THEN
              GO TO 65
          ELSEIF (I.EQ.1) THEN
              GO TO 75
          ELSE
              GO TO 63
      END IF
   65 PRINT*,'Enter node number:'
      READ*,J

      IF (J.GT.NDTOT) THEN
          PRINT*,'NODE DOES NOT EXIST!'
          GO TO 63
      END IF

      IF (DEPEND(J).GT.0) THEN
          PRINT*,'NODE IS DEPENDENT ON ANOTHER NODE.'
          PRINT*,'INELIGIBLE FOR CONTROL SUBSET.'
          GO TO 63
      END IF

      IF (CONTROL(J).EQ.0) THEN
          TCNTL = TCNTL + 1
          MUC = MUC + NPROB(J)
      END IF
      CONTROL(J) = 1
      GO TO 63

*
*     CLEAR CONTROL VARIATES
*

   70 MUC   = 0.
      TCNTL = 0
      DO 72 J = 1,NDTOT
          CONTROL(J) = 0
   72 CONTINUE

*
*
*     SIMULATION ITERATION
*
*

   75 DO 200 H = 1,SIM
        CNTL = TCNTL

*
*       CLEAR COL(x) AND DNODE(x) ARRAYS
*

        DO 80 J = 1,N
          COL(J) = 0
   80   CONTINUE

        DO 85 J = 1,NDTOT
          DNODE(J) = 0
   85   CONTINUE
```

```
*
*     DETERMINE LOSS OF NODES
*

        DO 100 K = 1,NDTOT

*         DETERMINE IF NODE IS DEPENDENT

          IF (DEPEND(K).GT.0) THEN
              IF (DNODE(DEPEND(K)).EQ.1) THEN
                  RDM = 1.1
                ELSE
                  RDM = -.1
              END IF
                  ELSE
              RDM = RANDOM(SEED)
              IF (ANTI.EQ.1) RDM = 1. - RDM
          END IF

*         IDENTIFY LOSS OF NODE FOR OTHER DEPENDENT NODES
*         MARK OFF ARCS LOST DUE TO NODE LOSS

          IF (RDM.GT.NPROB(K)) THEN
            IF (DEPEND(K).EQ.0) DNODE(K) = 1
            IF (CONTROL(K).EQ.1) CNTL = CNTL - 1
            DO 90 J = 1,N
              IF ( (HEAD(J).EQ.K).OR.(TAIL(J).EQ.K) ) THEN
                COL(J) = 1
              END IF
   90       CONTINUE
          END IF

  100   CONTINUE
        IF (CNTL.LT.0) CNTL = 0

*
*     DETERMINE REMAINING ARCS STATUS
*

        DO 130 J = 1,N
          IF ((HEAD(J).NE.TAIL(J)).AND.(COL(J).EQ.0)) THEN
            IF (ADEPEND(J).GT.0) THEN
                COL(J) = COL(ADEPEND(J))
              ELSE
                RDM = RANDOM(SEED)
                IF (ANTI.EQ.1) RDM = 1. - RDM
                IF (RDM.GT.APROB(J)) COL(J) = 1
            END IF
          END IF
  130   CONTINUE
```

```
*
*     CALCULATE MAX-FLOW BY FINDING MIN(MIN-CUT)
*     AND STORE IN STAT ARRAY
*

        L = UPBOUND
        K = 1
  135   IF (K.LE.T) THEN
          MINCUT = 0
          DO 140 J = 1,N
            IF (COL(J).EQ.0) THEN
              MINCUT = MINCUT + CUT(K,J)
            END IF
  140     CONTINUE
          IF (MINCUT.LT.L) L = MINCUT
          IF (L.EQ.0) K = T
          K = K + 1
          GO TO 135
        END IF

        STAT(H,1) = L
        STAT(H,2) = CNTL

  200 CONTINUE

*
*     CALCULATE MEAN, STANDARD DEVIATION, AND 95%
*     CONFIDENCE INTERVAL FOR NORMAL RESPONSE AND
*     CONTROLLED VARIATION RESPONSE
*

*     CLEAR VARIABLES

      MEAN  = 0.
      VAR   = 0.
      RLBL  = 0.
      YTOT  = 0.
      CTOT  = 0.
      YMEAN = 0.
      CMEAN = 0.
      YVAR  = 0.
      CVAR  = 0.
      COVAR = 0.
      MUY   = 0.
      BHAT  = 0.
      S11   = 0.
      CI1   = 0.
      CI2   = 0.
```

```
*     CALCULATE RESPONSE TOTAL (YTOT) AND MEAN (YMEAN)
*               CONTROL  TOTAL (CTOT) AND MEAN (CMEAN)
*               PERCENTAGE OF RUNS S-T CONNECTED (RLBL)
*     (REF: BAUER, PHD DISSERTATION, PURDUE UNIV., 1987)

      DO 210 I = 1,SIM
          YTOT = YTOT + STAT(I,1)
          CTOT = CTOT + STAT(I,2)
          IF (STAT(I,1).GT.0) RLBL = RLBL + 1.
  210 CONTINUE

      YMEAN = YTOT/SIM
      CMEAN = CTOT/SIM
      RLBL = RLBL/SIM

*     CALCULATE VARIANCE OF RESPONSE (YVAR) AND CONTROL
*     (CVAR) COVARIANCE OF RESPONSE AND CONTROL (COVAR)

      DO 220 I = 1,SIM
        YVAR = YVAR + (  (STAT(I,1) - YMEAN)**2 )
        CVAR = CVAR + (  (STAT(I,2) - CMEAN)**2 )
        COVAR = COVAR+((STAT(I,1)-YMEAN)*(STAT(I,2)-CMEAN))
  220 CONTINUE

*     CALCULATE ESTIMATOR OF BETA (BHAT) - EQ 2.1.9

      IF ( (TCNTL.GT.0).AND.(CVAR.GT.0.) ) BHAT=COVAR/CVAR

*     CALCULATE POINT ESTIMATOR OF MUy (MUY) IN EQ 2.1.10
*               USING EQ 2.1.7

      MUY = 0.
      DO 230 I = 1,SIM
          MUY = MUY + STAT(I,1) - ( BHAT*(STAT(I,2)-MUC) )
  230 CONTINUE

      MUY = MUY/SIM

*     CALCULATE VAR OF CONTROL EST. (VAR) - EQ 2.1.18
*     USING EST. OF CONTROL RESP. (YHAT) - EQ 2.1.19
*     CALCULATE S11 - EQ 2.1.21

      S11 = 0.
      VAR = 0.

      DO 240 I = 1,SIM
          YHAT = MUY + BHAT*( STAT(I,2) - MUC )
          VAR  = VAR + (  (STAT(I,1) - YHAT)**2 )
          S11  = S11 + (  (STAT(I,2) -   MUC)**2 )
  240 CONTINUE
```

```
      VAR = SQRT( VAR/(SIM-2) )
      IF (CVAR.GT.0.) S11 = SQRT( S11/(SIM*CVAR) )

*     CALCULATE 95% CONFIDENCE INTERVALS
*               NO CONTROL VARIATE - (CI1)
*               CONTROL VARIATE    - (CI2)

      YVAR = YVAR/(SIM-1)
      CI1  = 1.96*( SQRT(YVAR/SIM) )
      CI2  = 1.96*VAR*S11

*
*     RESULTS
*

      PRINT*,'NORMAL STATISTICS'
      PRINT*,'Mean:',YMEAN
      PRINT*,'Std. Dev.:',SQRT(YVAR)
      PRINT*,'Confidence intvl. (+-):',CI1
      PRINT*,' '

      IF (TCNTL.GT.0) THEN
          PRINT*,'CONTROL VARIATE STATISTICS'
          PRINT*,'Mean:',MUY
          PRINT*,'Std. Dev.:',VAR
          PRINT*,'Confidence intvl. (+-):',CI2
          PRINT*,' '
      END IF

      PRINT*,'Reliability:'
      PRINT*,RLBL
      PRINT*,' '


      PRINT*,'Enter any number to return to menu:'
      READ*,I
      GO TO 50

*
*     END SUBROUTINE
*

  500 END
```

```
*
*
*     CUTSET MODIFICATION SUBROUTINE
*
*

      SUBROUTINE CHGCUT

*
*     PARAMETERS
*

      INTEGER AC,RW,ND
      PARAMETER (RW=1000,AC=80,ND=50)

*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T,MUC,TCNTL
      INTEGER CUT(RW,AC)
      INTEGER HEAD(AC), TAIL(AC), CONTROL(ND)
      REAL    NPROB(ND), APROB(AC)
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/CUT2/MUC,TCNTL,CONTROL
*
*     LOCAL VARIABLES
*

      INTEGER I,J,K,CAP,HD,TL,ARC,FLAG
      REAL PROB
      FLAG = 0

*
*     MENU
*

   50 PRINT*,'Enter 0 to modify cutset:'
      PRINT*,'Enter 1 to quit:'
      READ*, I
      IF (I.EQ.1) GO TO 500

   55 PRINT*,'Enter 0 to modify node:'
      PRINT*,'Enter 1 to modify arc:'
      READ*,K
      IF ( (K.LT.0) .OR. (K.GT.1) ) GO TO 55
      IF (K.EQ.0) GO TO 60
```

```
*
*     ENTER ARC INFORMATION
*

      PRINT*,'Enter arc head:'
      READ*,HD
      PRINT*,'Enter arc tail:'
      READ*,TL
      GO TO 80

*
*     ENTER NODE INFORMATION
*

   60 PRINT*,'Enter node:'
      READ*,HD
      IF (HD.GT.NDTOT .OR. HD.LT.1) THEN
          PRINT*,'ERROR: Node does not exist!'
          GO TO 50
      END IF
      TL = HD

*
*     SEARCH FOR NODE OR ARC
*

   80 ARC = 0
      J = 1
   85 IF (J.LE.N) THEN
        IF ( (HEAD(J).EQ.HD) .AND. (TAIL(J).EQ.TL) ) THEN
          ARC = J
          J = N
        END IF
        J = J + 1
        GO TO 85
      END IF

      IF ( (K.EQ.1) .AND. (ARC.EQ.0) ) THEN
          PRINT*,'ERROR: ARC DOES NOT EXIST!'
          GO TO 50
      END IF
      IF ( (K.EQ.0) .AND. (ARC.EQ.0) ) THEN
        PRINT*,'WARNING: This is a non-capacitated node..'
        PRINT*,'Algorithm does not allow this node to
     +  change capacity.'
        PRINT*,'Only survival probability parameter may be
     +           changed.'
        CAP = 1
        GO TO 90
      END IF
```

```
*
*     ENTER CAPACITY AND PROBABILITY CHANGE
*

      PRINT*,'Enter new capacity:'
      READ*,CAP
   90 PRINT*,'Enter new probability:'
      READ*,PROB

      IF (PROB.LT.0.) PROB = 0.
      IF (PROB.GT.1.) PROB = 1.
      IF (CAP.LT.1)   CAP  = 1

      IF ( (K.EQ.1) .AND. (CAP.LE.0) ) THEN
          CAP = 1
          PROB = 0.
      END IF

*
*     CHANGE APPROPRIATE COLUMN IN CUTSET MATRIX
*

*     ARCS AND NODES

      IF (ARC.GT.0) THEN
        DO 100 I = 1,T
          IF (CUT(I,ARC).GT.0) CUT(I,ARC) = CAP
  100   CONTINUE
        APROB(ARC) = PROB
      END IF

*     NODE ONLY

      IF (K.EQ.0) THEN
          NPROB(HD) = PROB
          FLAG = 1
      END IF
      GO TO 50

*
*     TERMINATE ROUTINE
*

  500 IF (FLAG.EQ.1) THEN
          MUC = 0.
          TCNTL = 0
          DO 510 I = 1,NDTOT
              CONTROL(I) = 0
  510     CONTINUE
          PRINT*,'WARNING: CONTROL VARIATES ELIMINATED'
      END IF
      END
```

```
*
*
*     DEPENDENT ARC AND NODE SUBROUTINE
*
*

      SUBROUTINE DPND

*
*     PARAMETERS
*

      INTEGER RW,ND,AC
      PARAMETER (RW=1000,ND=50,AC=80)

*
*     GLOBAL VARIABLES
*

      INTEGER NDTOT,N,T
      INTEGER CUT(RW,AC)
      INTEGER HEAD(AC), TAIL(AC)
      REAL    NPROB(ND), APROB(AC)
      INTEGER DEPEND(ND), ADEPEND(AC)
      COMMON/CUT1/NDTOT,N,T,CUT,NPROB,APROB,HEAD,TAIL
      COMMON/CUT3/DEPEND,ADEPEND

*
*     LOCAL VARIABLES
*

      INTEGER I,J,K,HD,TL

*
*     MENU
*

    5 PRINT*,'0. Enter dependent nodes.'
      PRINT*,'1. Enter dependent arcs.'
      PRINT*,'2. Return to main menu.'
      PRINT*,'8. Clear dependent nodes (ALL nodes
     +independent).'
      PRINT*,'9. Clear dependent arcs (ALL arcs
     +independent).'
      READ*,I
      IF (I.EQ.0) THEN
          GO TO 10
        ELSE IF (I.EQ.8) THEN
          GO TO 80
        ELSE IF (I.EQ.1) THEN
          GO TO 200
        ELSE IF (I.EQ.9) THEN
          GO TO 300
        ELSE IF (I.EQ.2) THEN
          GO TO 500
        ELSE
          GO TO 5
      END IF

*
*     NODE DEPENDENCY ENTRY ROUTINE
*

   10 K = 0
   15 PRINT*,'Enter dependent node number or 0 to quit.'
      READ*,I

      IF (I.GT.NDTOT .OR. I.LT.0) THEN
          PRINT*,'NODE DOES NOT EXIST.'
          GO TO 15
      END IF
      IF (I.EQ.0) GO TO 30

      DEPEND(I) = -1
      K = K + 1
      GO TO 15

   30 IF (K.LE.1) THEN
          PRINT*,'MUST ENTER A MINIMUM OF TWO NODES.'
          GO TO 5
      END IF

*
*     IDENTIFY LOWEST NODE IN DEPENDENCY SET
*     AND SET DEPENDENT NODES TO THAT NODE NUMBER
*

      J = 0
      I = 1
   45 IF (I.LE.NDTOT) THEN
          IF (DEPEND(I).EQ.-1) THEN
              J = I
              I = NDTOT
          END IF
          I = I + 1
          GO TO 45
      END IF
      DEPEND(J) = 0
      DO 60 I = 1,NDTOT
          IF (DEPEND(I).EQ.-1) DEPEND(I) = J
   60 CONTINUE

      PRINT*,'IMPORTANT: The following node is the KEY
     +node.',J
      PRINT*,'The parameters for this node apply to the
     +dependency set.'
      GO TO 5

*
*     SET ALL NODES INDEPENDENT
*

   80 DO 85 I = 1,ND
          DEPEND(I) = 0
   85 CONTINUE

      PRINT*,'NOTICE: All nodes are now INDEPENDENT.'
      GO TO 5

*
*     ARC DEPENDENCY ENTRY ROUTINE
*

  200 K = 0
  215 PRINT*,'Enter 0 to enter dependent arc:'
      PRINT*,'Enter 1 when finished:'
      READ*,I

      IF (I.EQ.0) GO TO 220
      IF (I.EQ.1) GO TO 240
      GO TO 215

  220 PRINT*,'Enter arc head:'
      READ*,HD
      PRINT*,'Enter arc tail:'
      READ*,TL

      I = 1
      J = 0
  230 IF (I.LE.N) THEN
        IF (HEAD(I).EQ.HD .AND. TAIL(I).EQ.TL) THEN
            J = I
            I = N
        END IF
        I = I + 1
        GO TO 230
      END IF
      IF (J.EQ.0) THEN
        PRINT*,'ARC DOES NOT EXIST.'
        GO TO 215
      END IF

      ADEPEND(J) = -1
      K = K + 1
      GO TO 215

  240 IF (K.LE.1) THEN
          PRINT*,'MUST ENTER A MINIMUM OF TWO ARCS.'
          GO TO 5
      END IF
*
*     IDENTIFY LOWEST ARC IN DEPENDENCY SET
*     AND SET DEPENDENT ARCS TO THAT NODE NUMBER
*

      J = 0
      I = 1
  245 IF (I.LE.N) THEN
          IF (ADEPEND(I).EQ.-1) THEN
              J = I
              I = N
          END IF
          I = I + 1
          GO TO 245
      END IF

      ADEPEND(J) = 0
      DO 260 I = 1,N
          IF (ADEPEND(I).EQ.-1) ADEPEND(I) = J
  260 CONTINUE

      PRINT*,'IMPORTANT: The following arc is the KEY arc.'
      PRINT*,'HEAD:',HEAD(J)
      PRINT*,'TAIL:',TAIL(J)
      PRINT*,'The parameters for this arc apply to the
     +dependency set.'
      GO TO 5

*
*     SET ALL ARCS INDEPENDENT
*

  300 DO 385 I = 1,AC
          ADEPEND(I) = 0
  385 CONTINUE

      PRINT*,'NOTICE: All arcs are now INDEPENDENT.'
      GO TO 5
```

```
*
*     TERMINATE ROUTINE
*
  500 END
```

Bibliography

Agrawal, Avinash and Richard Barlow. "A Survey of Network Reliability and Domination Theory," Operations Research, 32: (May-June 1984).

----- and A. Satyanairayana. "An O(|E|) Time Algorithm for Computing the Reliability of a Class of Directed Networks," Operations Research, 32: 493-515 (May-June 1984).

Ball, Michael O. "Complexity of Network Reliability Computations," Networks, 10: 153-165 (Summer 1980).

-----. "Computing Network Reliability," Operations Research, 27: 823-838 (July-August 1978).

Bauer, Kenneth W., Assistant Professor, Department of Operational Sciences, Air Force Institute of Technology. Personal interview, Wright-Patterson AFB OH, 31 March 1988.

----- et al. "Using Path Control Variates in Activity Network Simulation." Unpublished working paper No. 88-01. Department of Operational Sciences, Air Force Institute of Technology, Wright Patterson AFB, OH, 1988.

-----. Control Variate Selection for Multiresponse Simulation. PhD Dissertation. School of Industrial Engineering, Purdue University, West Lafayette, IN, April 1987.

Bellmore, M. and P. A. Jensen. "An Implicit Enumeration Scheme for Proper Cut Generation," Technometrics, 12: 775-788 (November 1970).

Box, George E. P. and Norman Draper. Response Model-Building and Response Surfaces. New York: John Wiley, 1987.

Burt, John M. and Mark G. Garman. "Conditional Monte Carlo: A Simulation Technique for Stochastic Network Analysis," Management Science, 18: 207-217 (November 1971).

Chachra, Vinod and others. Applications of Graph Theory Algorithms. New York: North Holland, 1979.

Chvatal, Vasek. Linear Programming. New York: W. H. Freeman, 1980.

Draper, N. R. and H. Smith. Applied Regression Analysis (Second Addition). New York: John Wiley, 1981.

Easton, M. C. and C. K. Wong. "Sequential Destruction Method for Monte Carlo Evaluation of System Reliability," IEEE Transactions on Reliability, 29: 27-32 (April 1980).

Evans, J. R. "Maximum Flow in Probabilistic Graphs - The Discrete Case," Networks, 6: 161-183 (1976).

Fishman, George S. An Alternative to the Monte Carlo Estimation of Network Reliability, August 1983. Contract N00014-76-C-0302. North Carolina University at Chapel Hill: Curriculum in Operations Research and Systems Analysis (AD-A131849).

-----. "A Comparison of Four Monte Carlo Methods for Estimating the Probability of S-T Connectedness," IEEE Transactions on Reliability, 35: (June 1986).

-----. "A Monte Carlo Plan for Estimating Reliability Parameters and Related Functions," Networks, 17: 169-186 (Summer 1987).

-----. "Estimating Critical Path and Arc Probabilities in Stochastic Activity Networks," Naval Reserach Logistics Quarterly, 32: 249-261 (May 1985).

-----. "The Distribution of Maximum Flow With Applications to Multistate Reliability," Operations Research, 35: 607-618 (July-August 1987).

-----. "The Monte Carlo Estimation of Function Variation," Proceedings of the 1987 Winter Simulation Conference. 347-350. New York: Assoc. for Computing Machinery, 1987.

-----. Variance Reduction in the Simulation of Stochastic Activity Networks, January 1983. Contract N00014-76-C-0302. North Carolina University at Chapel Hill: Curriculum in Operations Research and Systems Analysis (AD-A124251).

Ford, L. R. and D. R. Fulkerson. Flows In Networks. Princeton: University Press, 1962.

Hammersley, J. M. and D. C. Handscomb. Monte Carlo Methods. London: Methuen, 1964.

Harary, F. Graph Theory. Reading, MA: Addison-Wesley, 1972.

Hillier, Frederick S. and Gerald Lieberman. Introduction to Operations Research (Fourth Edition). Oakland: Holden-Day, 1986.

Jensen, Paul A. and J. Wesley Barnes. Network Flow Programming. New York: John Wiley, 1980.

Karp, P. and M. G. Luby. "A New Monte Carlo Method for Estimating the Failure Probability of an N-Component System," Unpublished paper. Computer Sciences Division, University of California, Berkeley (1983).

Kleijnen, Jack P. C. Statistical Techniques in Simulation, Part I. New York: Marcel Dekker, 1974.

Kumato et al. "Dagger-Sampling Monte Carlo for System Unavailability Evaluation," IEEE Transactions on Reliability, 29: 122-125 (1980).

Lavenberg, S. S. and P. D. Welch. "A Perspective on the Use of Control Variables to Increase the Efficiency of Monte Carlo Simulations," Management Science, 27: 322-335 (March 1981).

----- et al. "Statistical Results on Control Variables with Application to Queueing Network Simulation," Operations Research, 30: 182-202 (January-February 1982).

Law, Averill M. and W. D. Kelton. Simulation Modeling and Analysis. New York: McGraw-Hill Book Company, 1982.

Marsh, Albert B. Department of Defense, 1988.

----- and David Knue, Department of Defense. Personal interview. Air Force Institute of Technology, Wright-Patterson AFB OH, 7 April 1988.

----- and David Knue, Department of Defense. Briefing and personal interview, 25 August 1988.

Mendenhall, William and others. Mathematical Statistics with Applications (Third Edition). Boston: Duxbury, 1986.

Montgomery, Douglas C. Design and Analysis of Experiments (Second Edition). New York: John Wiley, 1984.

Nijenhaus, Albert and Herbert Wilf. Combinatorial Algorithms. New York: Academic Press, 1975.

Plackett, R. L. and J. P. Burman. "The Design of Optimum Multifactorial Experiments," Biometrika, 33: 305-325,328-332 (1946).

Provan, J. Scott and Michael O. Ball. "Computing Network Reliability in Time Polynomial in the Number of Cuts," Operations Research, 32: 516-526 (May-June 1984).

Rosenthal, Arnie. "Computing the Reliability of Complex Networks," SIAM Journal of Applied Mathematics, 32: 384-394 (March 1977).

Rubinstein, Reuven Y. and Ruth Marcus. "Efficiency of Multivariate Control Variates in Monte Carlo Simulation," Operations Research, 33: 661-677 (May-June 1985).

Schrage, L. "A More Portable Fortran Random Number Generator," ACM Transactions on Mathematical Software, 5: 132-138 (1979).

Schruben, Lee W. and Barry H. Margolin. "Pseudorandom Number Assignment in Statistically Designed Simulation and Distribution Sampling Experiments," Journal of the American Statistical Association, 73: 504-520 (September, 1978).

Shamir, A. "A Linear Time Algorithm for Finding Minimum Cutsets in Reducible Graphs," SIAM Journal of Computing, 8: 645-655 (1979).

Shier, R. and E. Whited. "Algorithms for Generating Minimal Cutsets by Inversion," IEEE Transactions on Reliability, 34: 314-318 (October 1985).

Somers, J. E. "Maximum Flow in Networks with a Small Number of Random Arc Capacities," Networks, 12: 241-253 (1982).

Tew, J. D. and J. R. Wilson. "Metamodel Estimation Using Integrated Correlation Methods," Proceedings of the 1987 Winter Simulation Conference, 409-418. New York: Association for Computing Machinery, 1987.

Wallace, Stein A. "Investing in Arcs in a Network to Maximize Expected Flow," Networks, 17: 87-103 (Spring 1987).

VITA

Captain Thomas G. Bailey [redacted] from the United States Air Force Academy in 1978. After completing pilot training at Vance AFB, Oklahoma in 1979, he remained as an instructor and maintenance check pilot for the 71st Flying Training Wing. From there, he served as a C-5 Aircraft Commander and Chief, Operations System Management Division, 60th Military Airlift Wing, Travis AFB, California, until entering the School of Engineering, Air Force Institute of Technology, in June 1987. [redacted]

[redacted]

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION
Unclassified

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE

3. DISTRIBUTION / AVAILABILITY OF REPORT
Approved for public release;
distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
AFIT/GOR/ENS/88D-1

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
School of Engineering

6b. OFFICE SYMBOL (*If applicable*)
AFIT/ENS

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (*City, State, and ZIP Code*)
Air Force Institute of Technology (AU
Wright-Patterson AFB, Ohio 45433-6583

7b. ADDRESS (*City, State, and ZIP Code*)

8a. NAME OF FUNDING / SPONSORING ORGANIZATION
Department of Defense

8b. OFFICE SYMBOL (*If applicable*)

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (*City, State, and ZIP Code*)

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO.

11. TITLE (*Include Security Classification*)
RESPONSE SURFACE ANALYSIS OF STOCHASTIC NETWORK PERFORMANCE (unclassified)

12. PERSONAL AUTHOR(S)
Thomas G. Bailey, Captain, USAF

13a. TYPE OF REPORT
MS Thesis

13b. TIME COVERED
FROM ________ TO ______

14. DATE OF REPORT (*Year, Month, Day*)
1988 December

15. PAGE COUNT
161

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	03	
12	04	

18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
Monte Carlo Method, Network Flows, Response, Statistical Analysis

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

Title: See Box 11

Thesis Chairman: Kenneth W. Bauer, PhD, Asst. Professor
Major, USAF

10 Jan 89

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED □ SAME AS RPT. □ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
Unclassified

22a. NAME OF RESPONSIBLE INDIVIDUAL
Kenneth W. Bauer, Major, USAF

22b. TELEPHONE (*Include Area Code*)
513-255-3362

22c. OFFICE SYMBOL
AFIT/ENS

Previous editions are obsolete.

This thesis analyzed stochastic binary networks for the purpose of improving their performance as measured by expected maximum flow and source-to-sink reliability. The capacity and survivability of the networks' nodes and arcs formed the parameters of interest in the experimental design used to develop a response surface model. Nineteen parameters of particular interest in a specific network were screened using a Plackett-Burman design, resulting in five parameters of significant influence. A full 2^5 factorial orthogonal design was developed, with two first-order polynomials approximating the response surfaces of expected maximum flow and network reliability regressed from the experimental results. In addition to the descriptive insight provided by the response surfaces, a prescriptive example of an optimized network improvement strategy was developed by incorporating the response surface equations in a linear programming formulation.

Additional research investigated the use of a scalar internal control variate to reduce the variance of the maximum flow estimates. Specifically, the effect of the number of failed nodes of a selected control subset was regressed out of the simulation output to reduce the variance as much as 24%. The results indicated further variance reduction may be realized by expanding to a multiple set of controls that includes both arcs and nodes. Additionally, a correlation of response surface coefficients and control variate effectiveness was empirically shown, suggesting promising future research in this area.