T-3916

SIMULTANEOUS CAPITAL BUDGETING AND CONTINGENCY RESERVE ALLOCATION

by

Joseph A. Waldron

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mathematics).

Golden, Colorado

Date 16 April 1990

Signed: Joseph A. Waldron

Joseph A. Waldron

Approved: Robert E. D. Woolsey

Robert E. D. Woolsey
Thesis Advisor

Golden, Colorado
Date 16 April 1990

Ardel Boes

Ardel J. Boes
Professor and Head,
Department of Mathematics

ABSTRACT

This paper presents a method that jointly solves capital allocation and contingency reserve problems. The method uses chance constraints to capture the uncertainties inherent in estimates of budget requirements. The method also gives a decision maker control over contingency levels by setting a "confidence" (probability) that his total budget will not be exceeded in project execution.

A mathematical explanation of the method is presented followed by a explanation of the process through the use of an example. Also included are the results of the sample problem and suggestions for further research.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to thank the U. S. Army for giving me the opportunity to pursue a graduate degree, and I would like to thank Dr. Mike Anderson of the TRADOC Analysis Command at Fort Leavenworth, Kansas for his assistance with the sample problem I used in this paper.

I am also grateful to the faculty of the Colorado School of Mines and especially to the members of my committee, Dr. R. E. D. Woolsey, Dr. Frank E. Grange, and Prof. William R. Astle. The personal assistance provided by Dr. Grange along with access to facilities at Martin Marietta was extremely beneficial and educational.

Finally, I am greatly indebted to my wife and family for their support and encouragement. Their support was essential to provide me with the time needed for my academic studies and the countless hours and miles I spent training and preparing for the Leadville Trail 100 mile run.

Chapter 1
INTRODUCTION

Capital budgeting is a technique for solving resource constrained financial decision problems. Given a list of projects being considered for implementation with insufficient resources available, a choice must be made to determine which projects to select and which to reject. Each individual project is assigned a payoff value which will benefit the organization, and each project requires some amount of resources.

The project value may be expressed in terms of net present value (NPV) or may even be a dimensionless "utility index," such as in the Analytical Hierarchy Process proposed by Saaty (1980). The resources required for each project may include manpower, money, time, or raw materials. The overall value to the organization is the sum of the payoffs of the projects selected. Therefore, the "best" solution is the one that gives the greatest total value without exceeding the resource constraints.

Many methods have been proposed and used over the years to solve these problems. The numerous methods currently in the literature assume resources available and project requirements are deterministic; however, in reality there may be a high degree of uncertainty.

The data used in solving most capital budgeting problems are estimates of the actual costs and benefits. There will naturally be some variability in the certainty of completing the projects selected with the resources available. All good planners set aside some funds for emergencies or unplanned costs so they will have some degree of confidence they will not run out of money. If the stochastic nature of the problem is recognized from the beginning, the problem can be solved to determine how much contingency funding is necessary. This can give the decision maker a statistical assurance of achieving the highest payoff while remaining within the expected budget.

A typical capital budgeting problem can be formulated as follows. Determine x_j ($j = 1,\ldots,n$) to

$$\text{maximize} \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \le b_i, \qquad i = 1,\ldots,m$$

$$x_j = 0 \text{ or } 1, \qquad j = 1,\ldots,n$$

where x_j represents the projects being considered, c_j represents the benefit or value of project x_j, a_{ij} ($i = 1,\ldots,m$ and $j = 1,\ldots,n$) represents the cost of project j in year i , and b_i represents budget resources available for year i.

An example of such a problem is provided by the U. S. Army (1989). Presented with an extensive list of modernization actions, and inadequate resources to accomplish them all, a decision must be made to determine the best actions to select. Through the Analytical Hierarchy Process, the list is developed into packages which will either be selected for funding or rejected. Through the analytical hierarchy approach (Saaty 1980) each package is evaluated and determined to have some payoff towards meeting the organization's requirements. The annual cost of each package is determined from equipment cost estimates, and research and development costs over the next several years. The resources available are limited by the organization's budget.

Such basic data can be very difficult to estimate accurately. If this problem is solved as a 0-1 integer program, variations in estimated data may cause optimal or near optimal solutions to be infeasible. The decision maker must determine how much of each type of resource to set aside for contingencies to cover the inherent uncertainties in the estimates.

Common ways of determining the amount of resources needed to be set aside are by experience or by a flat rate rule-of-thumb, like setting aside 20% of the total budget. Any budget constraints which are slack at optimality create a natural contingency. This total surplus and the flat contingency may far exceed what is required for that period. Some of these reserves may be used more effectively elsewhere.

It is important to evaluate the entire problem by simultaneously selecting projects and determining the proper amount of contingency resource. The following is a list of criteria and motivations for examining attributes of effective capital budgeting methods.

(a) Managers cannot afford to overrun their budget. The method should give a level of assurance that the actual costs of the projects selected will not exceed the budget.

(b) Unused portions of budgets, while normally used as "year-end funds" are usually the least effectively spent funds, and have an additional administrative cost to be allocated before they are lost. Excessive contingency reserves may also eliminate certain solutions from consideration. Very worthwhile projects may not be considered if too much of the budget is set aside.

(c) Risk levels may vary considerably from one project to the next and from one year to the next. Simple flat contingency rules treat all projects as having the same risk level. For example, a project requiring new technology with a payoff years away does not have the same uncertainty as a project using current technology and an immediate payoff.

(d) The method must be understandable and understandably computable or it will not be used.

(e) The data must be obtainable, or the method cannot be used.

(f) The model must fit real world data. For example, the model should allow for dependencies between the costs of different projects, as well as allowing for different independent parameters.

Chapter 2
LITERATURE REVIEW

Capital Budgeting Methods

Capital budgeting problems have been solved by both heuristic and optimum mathematical programming methods. An optimum code for solving linear programs with binary (0-1) variables was proposed by Balas (1965). An optimal solution is often prohibitive because of the iterations which must be considered for the combinations of projects. When the data used are not deterministic, the value of an optimal solution is even more questionable. Another optimum method which may be used for such problems was proposed by Lawler and Bell (1966). Heuristic techniques which are known for their simplicity and robustness have been proposed by Senju and Toyoda (1968) and others.

In a recent study, Khan (1987) found that more than 91% of cities surveyed used some informal and or formal capital budgeting techniques. Although cost-benefit analysis was found to be used more frequently than any other technique, the study also found that a number of governments are beginning to take interest in new techniques.

Stochastic Programming

Charnes and Cooper (1959) initially examined chance-constrained programming. Later they showed that by using different kinds of decision rules and optimizing objectives, random elements can be eliminated and a programming problem can be formulated that is deterministic (Charnes and Cooper 1963). While this approach meets most of the criteria for solving capital budgeting problems under uncertainty, one problem is that the values in the constraint matrix had to be held constant. This assumes that each project has the same risk level and does not allow for real world differences in the distributions of the costs.

Bracken and McCormick (1968) explored the formulation and solution of a deterministic nonlinear programming problem that is equivalent to a stochastic linear programming problem of the chance-constrained type. Their work is restricted to problems with normally distributed random variables. Again this is not necessarily a good assumption, and the data required for a normal distribution may be very difficult to obtain. While an estimated mean may not be difficult to estimate, very few managers could give the variance needed to describe a normal distribution.

Smith (1973) compared deterministic equivalents derived for zero-order, linear, and two-piece rules as they applied to irrigation systems. His decision variables were stochastic functions and the problems resulted in non-linear objective functions and non-linear constraints. This is much more difficult to solve and the results are difficult to understand and interpret because of the stochastic decision variables.

Allen, Braswell, and Rao (1974) developed methods for approximating a chance-constraint set when information concerning the random variables is derived from actual samples. The methods they propose use a distribution-free tolerance region to construct various sets whose elements have the common property of satisfying the chance constraint with a preassigned level of confidence. Their simplest and most conservative method derives linear constraint sets from a distribution-free region at a given confidence level. The method is very understandable and makes it easy to compute a solution with current 0-1 integer programs such as Senju and Toyoda (1968).

The following figure graphically shows a subjective comparison of each of the methods being considered.

Legend for figure:

○ Does not meet criteria satisfactorily

◐ Meets criteria to a degree

● Fully meets criteria

Criteria	Standard	Charnes & Cooper	Bracken & McCormick	Smith	Allen et al
Meet budget	○	◐	●	●	●
Reserve Size	○	○	○	●	●
Flex. Risk Levels	○	○	◐	●	●
Understandable	●	●	●	○	●
Computable	●	◐	●	◐	●
Data Obtainable	●	●	◐	●	●
Fits Real World	○	○	○	●	●

Figure 2.1

Chapter 3
CHANCE-CONSTRAINED PROGRAMMING

Chance-constrained programming admits random data variations and permits constraint violations up to specified limits. The method discussed here is a general method of solving chance-constrained integer programming problems that can be used regardless of the underlying distributions or any of their parameters.

The method derives deterministic linear constraints which can be used to obtain a solution. These constraints are determined by sampling a distribution-free chance constraint set from within a distribution-free tolerance region. The limit to which constraints may be violated is designated as α, while β is designated as the confidence level that a constraint will hold with a probability α.

Problem Formulation

Some optimal models may require stochastic decision variables instead of considering operating decisions to be deterministic (Smith 1973). These solutions give a set of policies for operating decisions that depend on past realizations of stochastic variables and updated forecasts of those variables yet to be realized. Since for any given period the stochastic decision variable becomes a deterministic value, the capital budgeting problem can be formulated without stochastic decision variables.

The simplest application of chance-constrained programming uses zero-order decision rules. When operating decisions are considered to be invariant over time, they are deterministic or zero-order. These rules assume that the model solution produces a set of decisions to optimize the expected value of the objective function.

The chance constrained formulation below has the vector α of probability measures that specify the probability that each constraint will not be violated. Determine x_j ($j = 1,...,n$) to

$$Maximize \sum_{j=1}^{n} c_j x_j$$

subject to

$$P\left[\sum_{j=1}^{n} a_{ij} x_j \le b_i\right] \ge \alpha_i, \qquad i = 1,\ldots,m$$

$$x_j = 0 \text{ or } 1, \qquad j = 1,\ldots,n$$

Here some or all of the coefficients a_{ij}, b_i, c_j ($i = 1,...,m$ and $j = 1,...,n$) are not necessarily constant, but may have some or all of their elements as random variables. The α_i's are prescribed probabilities with which the constraints must be satisfied.

Distribution-Free Tolerance Regions

(Allen, Braswell, and Rao, 1974)

Let Y be an n - dimensional random variable with a cumulative distribution (cdf) H_Y. Let T be a random region in the sample space of Y, and assume that the exact shape and size of T depends on the observed values of a sample drawn from a population with cdf of H_Y. Define the coverage U of the region T as the probability measure of T. Since T is random U is also random. If the cdf of U is independent of H_Y, and if

$$P[U \geq \alpha] = \beta \quad \text{for } 0 \leq \alpha \leq 1,\ 0 \leq \beta \leq 1,$$

then T is called a 100α percent distribution-free tolerance region at a probability level β.

To find the sample size N to draw from the population, use a sequence of cutting functions of the form $\phi_j(\mathbf{A}) = a_j$, $j = 1, \ldots, n$, to construct the tolerance region. Allen shows that U_{N-n+1} is a Beta random variable with parameters $N - n + 1$ and n.

Therefore since $P[U_{N-n+1} \geq \alpha] = \beta$:

$$I_{1-\alpha}(n,\ N - n + 1) = \beta \tag{3.1}$$

where $I_x(p, q)$ is an incomplete beta function as defined in Appendix A.

The size of the sample space required to be drawn from H_Y can be determined by solving for N. For fixed values of α, β and n, there may be no sample size N for which (3.1) holds exactly. Therefore, we want the smallest integer N for which

$$I_{1-\alpha}(n,\ N - n + 1) \geq \beta. \tag{3.2}$$

Using a trial and error method to solve for the smallest integer value of N can be quite arduous. This can be accomplished more conveniently from specially prepared graphs or tables. Appendix B has a sample graph prepared using various values of α and β with n =8. Using a graph such as this, N can easily be determined given values for α and β.

Distribution-Free Chance Constraints

The concept of a distribution-free chance constraint set is defined as, "A distribution-free chance constraint set $S(\alpha, \beta)$ for the constraint $\boldsymbol{Ax} - b \leq 0$ has the property that, for any $\boldsymbol{x} \in S(\alpha, \beta)$, it can be asserted with a confidence of at least β that $P[\boldsymbol{Ax} - b \leq 0] \geq \alpha$."

The procedure for constructing this set consist of two steps:

Step 1. Construct a 100α percent distribution-free region with confidence β from a sample of $\boldsymbol{A}$, designated $\boldsymbol{A}_k$, for $k = 1, \ldots, N$. This region is specified by $T(\alpha, \beta)$.

Step 2. Determine the set $S(\alpha, \beta)$ such that, for any $\boldsymbol{x} \in S(\alpha, \beta)$, $\boldsymbol{Ax} \leq b$ for all $\boldsymbol{A} \in T(\alpha, \beta)$.

Linear Constraint Set

A distribution-free set can be represented as a linear constraint. First construct a distribution free tolerance region $T_L(\alpha, \beta)$ using a sequence of cutting functions. The required sample size N is the smallest integer N which satisfies (3.2). The tolerance region is given by

$$T_L(\alpha,\beta) = \{\boldsymbol{A} \mid \boldsymbol{A} \le \boldsymbol{W}\},$$

where $\boldsymbol{W} = (W_1, \ldots, W_n)$ are determined by

$$W_1 = a_{1(1)} = \max_k a_{1k},$$

$$W_2 = a_{2(2)} = \max_{k \neq (1)} a_{2k}$$

$$\vdots$$

$$W_n = a_{n(n)} = \max_{k \neq (1), \ldots, (n-1)} a_{nk}.$$

where a_{nk} is the k^{th} sample of the n^{th} element of the vector $\boldsymbol{A}$. Then the set S_L is given by

$$S_L(\alpha,\beta) = \{x \mid \boldsymbol{W}\boldsymbol{x} \le 1, x = 0,1\}.$$

Once the linear constraint set has been formulated, the problem can be solved. Using any of the 0-1 integer programming methods currently in use, solving the resulting deterministic problem is relatively easy.

Chapter 4
SAMPLE PROBLEM

To demonstrate the procedure for solving a capital budgeting problem using distribution-free chance constraints, a sample problem is presented. The problem contains unclassified data from the U. S. Army Modernization Memorandum and Field Long Range Research, Development, and Acquisition Plan. The objective of the memorandum is to determine the best modernization actions to fund.

The proposed actions are organized into projects. Annual costs of each action are determined from estimates of the project sub-component costs, and each is assigned a benefit. For example, one project is labeled A01 with a benefit of 4.3 units at a cost of $252 for the first year, $262 for the second year, etc. The budget allocation for the first year is $1600.

Because the entire problem is larger than necessary to suitably demonstrate the procedure, Table 4.1 contains abbreviated data from this problem. Appendix C lists the detailed data provided for the entire problem. Also, the values selected for probability of constraints holding (α) and confidence level (β) are arbitrarily selected to be α =.80 and β =.90 for the sample problem. The problem formulated with the linear constraint set will be solved using the method of Senju and Toyoda, since this is the method currently used by the Army to solve the problem.

Table 4.1

Sample Problem Data

Project	Benefit	Yr 1	Yr 2	Yr 3	Yr 4	Yr 5
A01	4.3	252	262	233	195	160
B01	12.5	625	650	725	790	685
C01	9.0	345	800	850	850	400
D01	2.0	89	92	95	89	92
E01	15.5	550	570	1445	1275	1250
F01	6.3	80	120	160	120	80
G01	13.2	705	845	985	0	0
H01	9.2	178	186	266	201	227
Budget Available		1600	1800	2000	1700	1650

Formulating this data into a 0-1 integer program yields:

Maximize:

$$Z = 4.3x_1 + 12.5x_2 + 9.0x_3 + 2.0x_4 + 15.5x_5 + 6.3x_6 + 13.2x_7 + 9.2x_8$$

Subject To:

$$252x_1 + 625x_2 + 345x_3 + 89x_4 + 550x_5 + 80x_6 + 705x_7 + 178x_8 \le 1600$$
$$262x_1 + 650x_2 + 800x_3 + 92x_4 + 570x_5 + 120x_6 + 845x_7 + 186x_8 \le 1800$$
$$233x_1 + 725x_2 + 850x_3 + 95x_4 + 1445x_5 + 160x_6 + 985x_7 + 266x_8 \le 2000$$
$$195x_1 + 790x_2 + 850x_3 + 89x_4 + 1275x_5 + 120x_6 + 0x_7 + 201x_8 \le 1700$$
$$160x_1 + 685x_2 + 400x_3 + 92x_4 + 1250x_5 + 80x_6 + 0x_7 + 227x_8 \le 1650$$
$$x_j = 0,1 \quad \text{for} \quad j = 1 \text{ to } 8$$

The problem can easily be solved as it is formulated, but this does not account for any uncertainty in the cost estimates. The decision maker does not have any information about the size of the contingency fund required.

This is not an adequate formulation to make prudent decisions about the best set of projects to fund.

The coefficients (a_{ij} 's) of the constraint matrix do not describe the inherent uncertainty in the cost estimates. Any of the coefficients may have the same type of distribution, or each may have a completely different distribution. For example, one coefficient with a high degree of uncertainty may be best described by a normal distribution with a large variance, while another has a uniform distribution over a fairly small range of values.

It is also possible that some coefficients have both dependent and independent components. For example, the costs of several similar items being considered from one contractor may be considered to have the same variances, but different means.

Often, the best data obtainable are the the highest , lowest, and most likely estimates for the cost of a project. These give a triangular distribution for the costs of each package. Therefore, the rest of this example will assume that each cost coefficient has a triangular distribution. Appendix D is an example of how samples are derived for a triangular distribution. The values describing each distribution were arbitrarily selected for each cost coefficient. Table 4.2 shows these additional data for each cost coefficient.

Table 4.2

Triangular Distributions for Sample Problem

Project	Benefit		Yr 1	Yr 2	Yr 3	Yr 4	Yr 5
A01	4.3	L	245	255	220	186	152
		M	252	262	233	195	160
		H	282	280	256	212	183
B01	12.5	L	617	642	714	773	679
		M	625	650	725	790	685
		H	645	695	765	829	705
C01	9.0	L	333	791	841	842	386
		M	345	800	850	850	400
		H	368	826	878	875	427
D01	2.0	L	82	85	88	79	81
		M	89	92	95	89	92
		H	105	116	119	108	109
E01	15.5	L	541	562	1410	1250	1190
		M	550	570	1445	1275	1250
		H	575	589	1480	1312	1283
F01	6.3	L	70	105	138	114	68
		M	80	120	160	120	80
		H	101	141	189	150	90
G01	13.2	L	675	800	965	0	0
		M	705	845	985	0	0
		H	775	924	1050	0	0
H01	9.2	L	155	160	250	190	220
		M	178	186	266	201	227
		H	198	205	285	225	240

In the table, the L is the lowest value estimate, the M is the most likely, and the H is the highest estimated cost. The resources to be allocated per year are assumed to be known for this example, but could be stochastic by changing the constraints from $\boldsymbol{Ax} \leq \mathrm{b}$ to $\boldsymbol{Ax} / \mathrm{b} \leq 1$.

The α's and β's are then selected for each constraint on the basis of the confidence level required. The values picked will dictate the number of samples needed to derive the deterministic constraints; these sample sizes can be derived from tables of the incomplete beta function.

As a comparison of how large the sample size (N) will be for various α's and β's, Table 4.3 shows the required number of samples for this problem with $n = 8$.

Table 4.3

Sample Size Requirements

$\alpha \backslash \beta$	.80	.90	.95
.80	50	57	63
.90	101	116	129
.95	204	234	260

From the table, if $\alpha = .80$ and $\beta = .90$, a problem with 8 variables will require 57 simulated samples. To demonstrate the procedure, Table 4.4 contains a sample of 20 points for each $a_{i,j}$. ($a_{1,1,k}$, . . . ,$a_{1,8,k}$ where $k = 1$ to 20).

Table 4.4

Simulated Samples Generated for $a_{1,1,k}$, . . . ,$a_{1,8,k}$

k	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$	$a_{1,8}$
1	252.7	638.8	350.7	91.7	556.6	73.9	736.1	184.9
2	266.1	632.8	346.2	86.8	551.5	81.9	699.8	170.8
3	265.9	623.8	340.3	85.8	559.0	79.5	712.8	180.2
4	267.1	621.0	339.8	98.2	557.2	77.4	704.3	165.6
5	269.5	637.0	342.4	90.8	560.3	88.3	703.1	178.8
6	267.6	628.7	344.0	95.3	562.4	78.8	698.6	164.3
7	273.1	631.9	353.0	86.2	549.3	96.7	723.7	175.9
8	255.4	627.2	359.1	98.2	554.9	89.9	711.3	179.8
9	262.4	630.1	353.4	96.6	565.0	87.8	697.9	172.2
10	253.0	626.7	341.0	93.6	557.0	79.3	734.1	174.6
11	275.0	635.6	349.8	92.3	559.0	74.6	738.6	176.5
12	258.7	624.9	348.0	94.3	549.1	79.7	701.1	173.5
13	246.9	636.7	357.4	93.1	566.5	79.0	727.3	177.5
14	257.6	622.8	351.3	98.4	558.4	89.7	691.3	181.4
15	253.2	627.2	345.8	91.2	543.6	91.3	697.1	170.7
16	262.5	626.4	349.4	89.6	556.9	94.5	744.2	186.0
17	255.1	627.9	343.1	99.1	560.6	85.5	751.3	183.1
18	270.7	629.2	347.0	91.2	565.3	75.8	706.2	177.2
19	252.7	624.4	354.3	98.8	546.6	85.3	741.4	172.6
20	252.3	628.7	354.1	98.9	557.0	98.2	715.1	179.7

The next step in the algorithm is to pick the largest value in the first column ($a_{1,1,k}$ = 275.0), record that value and cross off the rest of the row

(row k=11). Next pick the largest value of $a_{1,2,k}$ (from the remaining rows, record that value and cross off the rest of the row. Continue for all $a_{1j,k}$ for j = 1 to 8. Now the first constraint can easily be written using the values recorded for each a_{ij}.

$$275.0x_1 + 638.8x_2 + 359.1x_3 + 99.1x_4 + 566.5x_5 + 98.2x_6 + 744.2x_7 + 181.4x_8 \leq 1600$$

This process is then repeated for each of the remaining constraints for years 2 through 5.

The deterministic linear program below was obtained by using α =.80 and β=.90 for each constraint and generating all 57 samples required. Appendix E presents a BASIC program which generates these data.

Maximize:

$$Z = 4.3x_1 + 12.5x_2 + 9.0x_3 + 2.0x_4 + 15.5x_5 + 6.3x_6 + 13.2x_7 + 9.2x_8$$

Subject To:

$$278.0x_1 + 643.1x_2 + 366.0x_3 + 103.0x_4 + 572.6x_5 + 97.4x_6 + 768.2x_7 + 195.7x_8 \leq 1600$$

$$278.0x_1 + 689.4x_2 + 823.5x_3 + 113.3x_4 + 584.0x_5 + 137.8x_6 + 913.7x_7 + 201.5x_8 \leq 1800$$

$$252.7x_1 + 757.6x_2 + 876.6x_3 + 111.9x_4 + 1467.3x_5 + 188.0x_6 + 1046.4x_7 + 278.4x_8 \leq 2000$$

$$209.5x_1 + 825.4x_2 + 871.0x_3 + 106.5x_4 + 1305.2x_5 + 147.4x_6 + 0x_7 + 220.8x_8 \leq 1700$$

$$181.9x_1 + 704.5x_2 + 426.0x_3 + 105.9x_4 + 1272.8x_5 + 87.6x_6 + 0x_7 + 237.9x_8 \leq 1650$$

$$x_j = 0,1 \quad \text{for} \quad j = 1 \text{ to } 8$$

Solving this problem with a modified Senju and Toyoda heuristic as the Army used gives $Z^* = 31.0$ with $x_5, x_6, x_8 = 1$, and $x_1, x_2, x_3, x_4, x_7 = 0$. Appendix F contains the program used to obtain this result (Grange 1973 and Woolsey 1975). This solution gives a 90% confidence level that each constraint will hold with a probability of 80%. By comparison, the optimal solution using Haverly Systems Linear Program is $Z^* = 35.0$ with $x_1, x_4, x_7, x_8 = 1$, and $x_2, x_3, x_5, x_6 = 0$.

Contingency Funds

To compute the contingency funds required, compare the slack in each chance constraint at optimality to the slack in the most likely constraints. Using year 1 for example,

Slack in the chance constraint (CC) :

$$= 1600 - 572.6x_5 - 97.4x_6 - 195.7x_8$$

$$= 1600 - 865.7$$

$$= 734.3$$

Slack in the most likely constraint (MLC):

$$= 1600 - 550.0x_5 - 80.0x_6 - 178.0x_8$$

$$= 1600 - 808$$

$$= 792.0$$

Contingency reserve required:

$$792.0 - 734.3 = 57.7$$

Therefore, the amount of contingency that should be set aside for year 1 is \$ 57.7, which is 6.7% of the funds needed to accomplish the projects selected for that year (\$865.7). Table 4.5 shows the reserves required for the entire project.

Table 4.5

Annual Contingency Funds

Year	MLC Slack	CC Slack	100α Reserve	% Reserve
1	792.0	734.3	57.7	6.7
2	924.0	876.7	47.3	5.1
3	129.0	66.3	62.7	3.2
4	104.0	26.2	77.8	4.6
5	93.0	51.7	41.3	2.6

The chance constraint slack column can be interpreted as the amount of the budget that could be reallocated, or new projects which require funds during these periods could be added to the evaluation.

Verification of Solution

The following test is provided to verify that the solution above gives a probability level of .80 (α) that the constraint holds. With the right hand sides set at the levels needed to meet the deterministic solution, we will use the solution with $x_5, x_6, x_8 = 1$, $x_1, x_2, x_3, x_4, x_7 = 0$. The distribution in each of these constraints is sampled and the left hand side terms are summed and compared to the right hand side. After repeated sampling, the constraint should hold with a probability of at least .80 (or at least 80% of the samples do not violate the constraint).

To test that the confidence level of .90 (β) is met, repeat the test above and observe how often the observed value of α is above .80.

Appendix G contains a program which conducts both of these tests. The results listed show that both parameters α and β are easily met for the Senju and Toyoda solution set.

Chapter 5
CONCLUSION

The solution obtained for the sample was shown to hold for the parameters specified. As Allen, Braswell, and Rao showed, the result for a linear constraint set are conservative. Perhaps the biggest advantage of this method is that the entire problem is evaluated simultaneously. The size of resource reserve is determined by the projects selected and the projects selected are determined by the resources available. Therefore, the entire set of projects has a high level of assurance of being completed within budget and without excessive "year-end funds." Traditional methods which use a flat contingency rate may cause too much reserve to be allocated one year, and not enough allocated the next year to cover the variability.

The effectiveness of this technique for solving capital budgeting problems can be evaluated using the criteria outlined in the introduction.

(a) The method gives a decision maker a level of confidence that the projects selected will not exceed the resources available, by allowing the manager to specify a statistical level of assurance that the actual costs of the projects selected will not exceed the budget.

(b) Resources are allocated at the level necessary to meet expected variations, therefore excess year end funds are kept to a minimum.

(c) Each project is evaluated at its own risk level.

(d) The method is understandable and easily computable.

(e) The method is flexible enough to use the data that is most readily available.

(f) The procedure for solving the problem remains the same, even when real world dependencies are known, or specific distributions are appropriate to certain types of resource requirements. Better data available will give a better solution.

Possible topics for further research may include examining alternative search methods for obtaining the deterministic coefficients, or evaluating nonlinear solutions through the use of convex constraint sets.

REFERENCES CITED

Allen, F. M., R. N. Braswell, and P. V. Rao. 1974. "Distribution-Free Approximations for Chance Constraints." Operations Research 22 (3): 610-621.

Balas E. 1965. "An Additive Algorithm for Solving Linear Programs with 0-1 Variables," Operations Research 13: 517-549.

Bracken, J. and G. P. McCormick. 1968. Selected Applications of Nonlinear Programming. New York: John Wiley and Sons.

Charnes, A. and W. W. Cooper. 1959. "Chance-Constrained Programming." Management Science 6 (1).

Charnes, A. and W. W. Cooper. 1963. "Deterministic Equivalents for Optimizing and Satisficing Under Chance Constraints." Operations Research 11 (1): 18-39.

Grange, F. E. II. 1973. "Solution of a Resource Allocation Problem with a Modification of the Method of Senju and Toyoda." Master's Thesis, Colorado School of Mines, Golden, Colorado.

Khan, A. 1987. "Capital Budgeting Practices in Large U.S. Cities." The Engineering Economist. 33 (1): 1-12.

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

Lawler, E. L. and M. D. Bell. 1966. "A Method for Solving Discrete Optimization Problems." Operations Research 14 (6): 1098-1112.

Pearson, K. 1934. Tables of the Incomplete Beta Function. Cambridge, England: Cambridge University Press.

Saaty, T. L. 1980. The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation. University of Pittsburg Press.

Senju, S. and Y. Toyoda. 1968. "An Approach to Linear Programming with 0-1 Variables." Management Science 15 (4): B-196-207.

Smith, D. V. "Decision Rules in Chance-Constrained Programming: Some Experimental Comparisons." Management Science 19 (6): 688-702.

U. S. Army. 1989. Army Modernization Memorandum and Field Long Range Research, Development, and Acquisition Plan Analytical Procedures. Fort Leavenworth, Kansas: Training and Doctrine Analysis Command.

Wilks, S. S. 1963. Mathematical Statistics. New York: Wiley.

Woolsey, R. E. D. and H. S. Swanson. 1975. Operations Research for Immediate Application: A Quick and Dirty Manual. New York: Harper and Row.

SELECTED BIBLIOGRAPHY

Charnes, A. and M. J. L. Kirby. 1967. "Some Special P-Models in Chance-Constrained Programming." Management Science 14 (3): 183-195.

Choi, T.S. and Levary, R. R. 1989. "Multinational Capital Budgeting Using Chance-constrained Goal Programming." International Journal of Systems Sciences 20 (3): 395-414.

Lee, C. J. 1988. "Capital Budgeting Under Uncertainty: The Issue of Optimal Timing." Journal of Business Finance and Accounting 15 (2): 155-169.

Reutlinger, S. 1970. Techniques for Project Appraisal Under Uncertainty. Johns Hopkins Press.

Taha, H. A. 1975. Integer Programming Theory, Applications, and Computations. University of Arkansas Press.

Weingartner, H. M. 1963. Mathematical Programming and the Analysis of Capital Budgeting Problems. Englewood Cliffs, N. J.: Prentice-Hall.

APPENDIX A

Incomplete Beta Function

(Pearson 1934)

The beta function is defined as

$$B(p,q) = \int_0^1 x^{p-1}(1-x)^{q-1}\,dx ,$$

and

$$B_x(p,q) = \int_0^x x^{p-1}(1-x)^{q-1}\,dx .$$

Then the incomplete beta function $I_x(p,q)$ is defined as

$$I_x(p,q) = \frac{B_x(p,q)}{B(p,q)} ,$$

which can be written as

$$I_x(p,q) = \frac{\Gamma(p+q)}{\Gamma(p)\Gamma(q)}\int_0^x x^{p-1}(1-x)^{q-1}\,dx .$$

Substituting for the parameters $x = 1\text{-}\alpha$, $p = n$ and $q = N - n + 1$ gives

$$I_{1-\alpha}(n, N-n+1) = \frac{B_{1-\alpha}(n, N-n+1)}{B(n, N-n+1)} ,$$

It is now advantageous to take advantage of the relationship

$$I_x(p,q) = 1 - I_{1-x}(q,p),$$

so

$$I_{1-\alpha}(n, N-n+1) = 1 - I_{\alpha}(n, N-n+1),$$

which gives

$$I_{1-\alpha}(n, N-n+1) = 1 - \frac{\Gamma(N+1)}{\Gamma(n, N-n+1)\,\Gamma(n)} \int_0^{\alpha} \alpha^{N-n}(1-\alpha)^{n-1} d\alpha .$$

APPENDIX B

Sample-Size Graph

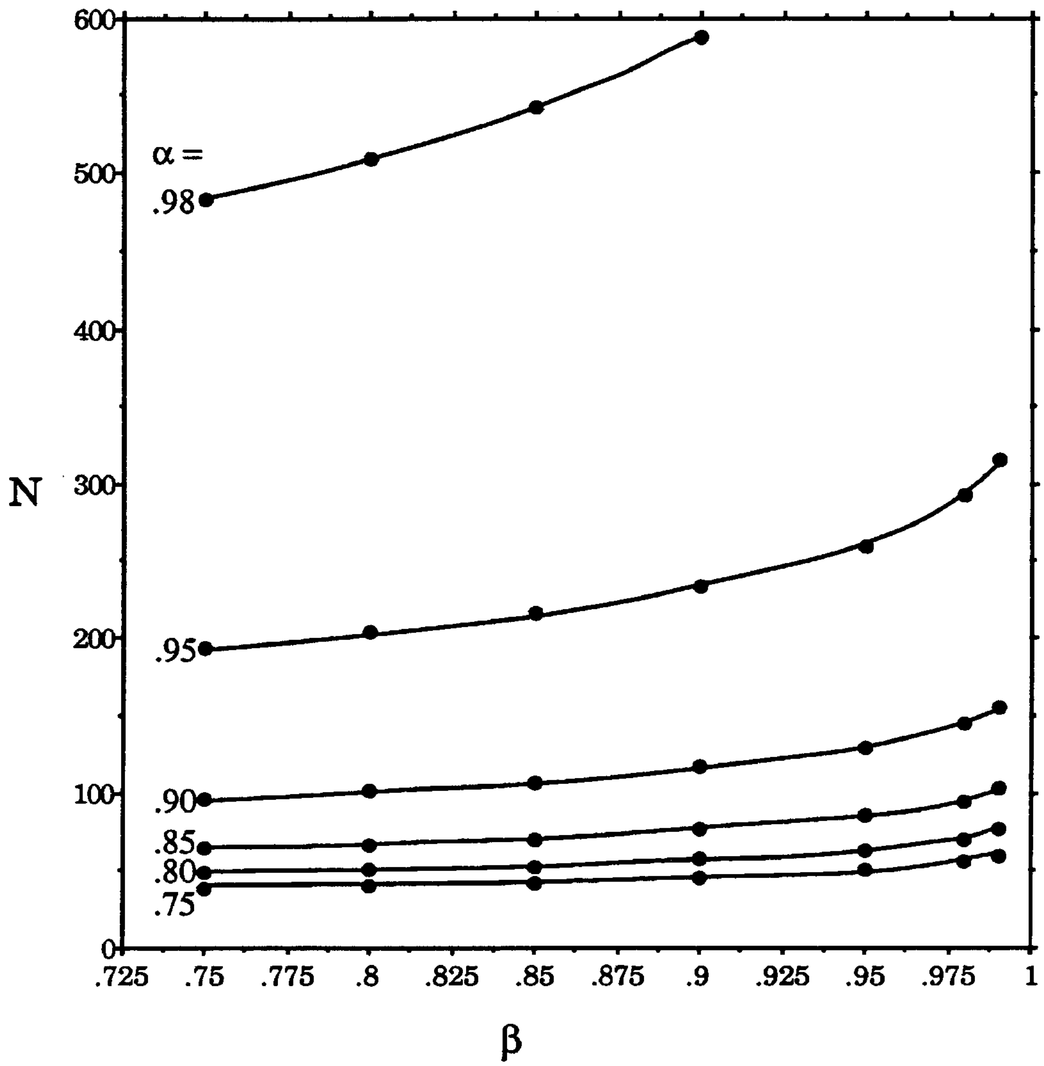

Figure B.1

Sample Size Graph (n = 8)

APPENDIX C

Sample Problem Data

This is a sample of the data used by the U. S. Army for their modernization problem. The Name column lists project labels, the Benefit column lists the value of each project in dimensionless units, and the remaining columns are the annual costs of each project in dollars.

Table C.1
Modernization Problem

Name	Benefit	FY91	FY92	FY93	FY94	FY95
MDAA01	4.3	252	262	233	195	160
MDAA02	1.8	50	210	212	215	218
MDAA03	1.2	0	10	40	40	95
MDBB01	12.5	625	650	725	790	685
MDCC01	9.0	345	800	850	850	400
MDCC02	3.5	0	80	95	100	95
MDDD01	2.0	89	92	95	89	92
MDDD02	0.8	20	25	30	0	0
MDDD03	0.4	15	15	0	0	50
MDEE01	15.5	550	570	1445	1275	1250
MDEE02	7.5	155	185	185	300	300
MDFF01	6.3	80	120	160	120	80
MDGG01	13.2	705	845	985	0	0
MDGG02	4.7	569	544	524	45	0
MDGG03	4.3	97	203	310	450	480
MDHH01	9.2	178	186	266	201	227
MDHH02	3.8	430	330	160	285	290

APPENDIX D
Triangular Distribution

The triangular distributions in the sample problems are only one of many possible distributions which may be used. The following is an example of how the distribution is derived and used.

Derivation

A triangular density function is shown in figure D.1 below.

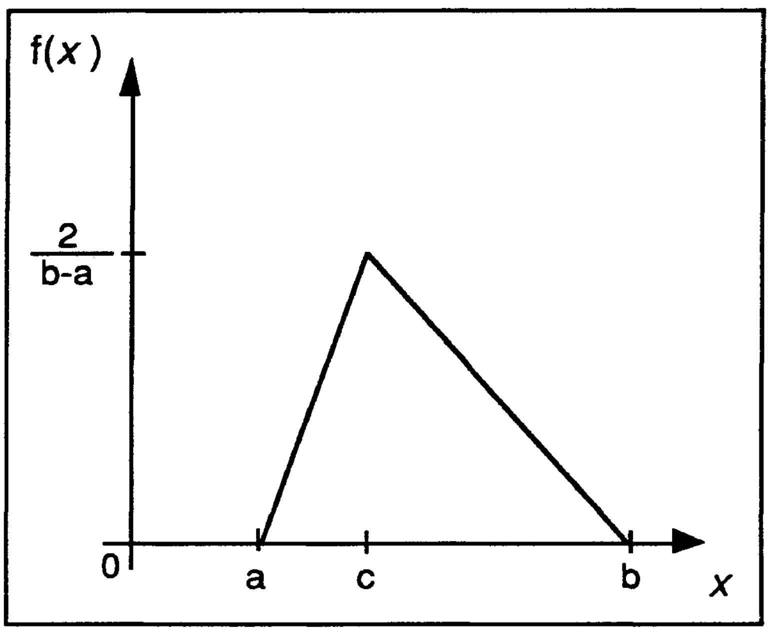

Figure D.1

The triangular probability density function is

$$f(x) = \begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)} & a \le x \le c \\ \dfrac{2(b-x)}{(b-a)(b-c)} & c < x \le b \\ 0 & \text{otherwise} \end{cases}$$

and the cumulative distribution is

$$F(x) = \begin{cases} 0 & x < a \\ \dfrac{(x-a)^2}{(b-a)(c-a)} & a \le x \le c \\ 1 - \dfrac{(b-x)^2}{(b-a)(b-c)} & c < x \le b \\ 1 & b < x \end{cases}$$

which is shown as:

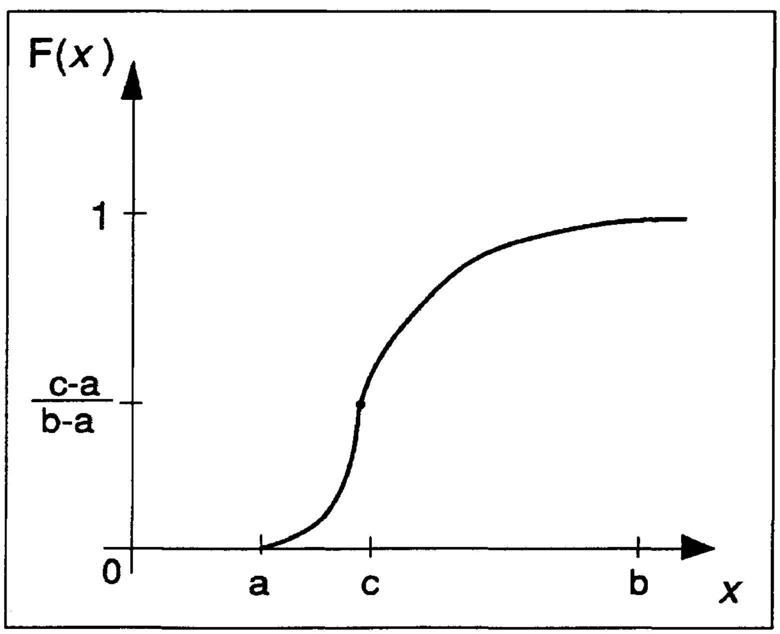

Figure D.2

Using the inverse transform method (Law 1982), Generate u~U[0,1], where U[0,1] is a uniform distribution from 0 to 1. Then,

Let u = F(x).

then , for $0 \le u < \frac{c-a}{b-a}$,

$$u = \frac{(x-a)^2}{(b-a)(c-a)},$$

solving for x gives

$$x = a + \sqrt{u(b-a)(c-a)}. \qquad \text{(D.1)}$$

for $\frac{c-a}{b-a} \le u < 1$,

$$u = 1 - \frac{(b-x)^2}{(b-a)(b-c)},$$

solving for x gives

$$x = b - \sqrt{(1-u)(b-a)(b-c)}. \qquad \text{(D.2)}$$

Therefore, for any u which is less than (c - a) / (b - a), equation (D.1) is used to generate values for the triangular distribution. For any u generated which is greater than (c - a) / (b - a), equation (D.2) is used, therefore,

$$F^{-1}(u) = \begin{cases} 0 & u < 0 \\ a + \sqrt{u(b-a)(c-a)} & 0 < u \le \frac{c-a}{b-a} \\ b - \sqrt{(1-u)(b-a)(b-c)} & \frac{c-a}{b-a} < u \le 1 \\ 1 & 1 < u \end{cases}$$

Program

The following program verifies the generation of pseudo-random numbers for a triangular distribution.

```
'Program to check the generation of random numbers
' from a triangular distribution
'
  DIM x(10000)
     OPEN "Random #'s" for OUTPUT AS #1
   RANDOMIZE TIMER                         'Accept a random # as the seed
   INPUT "# of random numbers";n&          'Allow up to 10,000 numbers
   INPUT "a, b, c";a,b,c                   'a = minimum, b = maximum
                                           'c = mode
Calculations:
  FOR i = 1 TO n&
          u = RND
          test = (c-a)/(b-a)
             GOSUB Triangular
               Print #1, x(i)              'Send random #'s to an output file
     sum = sum + x(i)
  NEXT i
     mean = sum /n&                        'Calculate Sample Mean
    FOR i = 1 TO n&
       Square = (x(i) - mean)^2
       SumSquare = SumSquare + Square
    NEXT i
    variance = SumSquare / (n& - 1)        'Calculate Sample Variance
  GOSUB Results
END
```

```
Triangular:                              'Sample a value from a
                                         'triangular distribution based on
                                         'a random number (u).
      IF u > test THEN
        x(i) = b - SQR ((1-u) * (b-a) * (b-c))
      ELSE
        x(i) = a + SQR( u * (b-a) * (c-a))
      END IF
    RETURN
Results:
      PRINT:   PRINT "Hypothetical mean    Hypothetical Variance"
      PRINT (a+b+c)/3,(a^2+b^2+c^2-a*b-a*c-b*c)/18
      PRINT "Sample mean          Sample Variance"
       PRINT USING " ####.#####        ";mean,variance:PRINT
       PRINT USING "####
  RETURN
```

Summary

The following figure graphically portrays a summary of the program results from generating 10,000 random numbers on a Macintosh IIcx using Microsoft QuickBasic. Random numbers generated within QuickBasic are are distributed uniformly from 0 to 1. The parameters used were a=0, b=1, c=.5.

Results

****

Hypothetical Mean	Hypothetical Variance
0.50000	0.04166
Sample Mean	**Sample Variance**
0.50305	0.04188

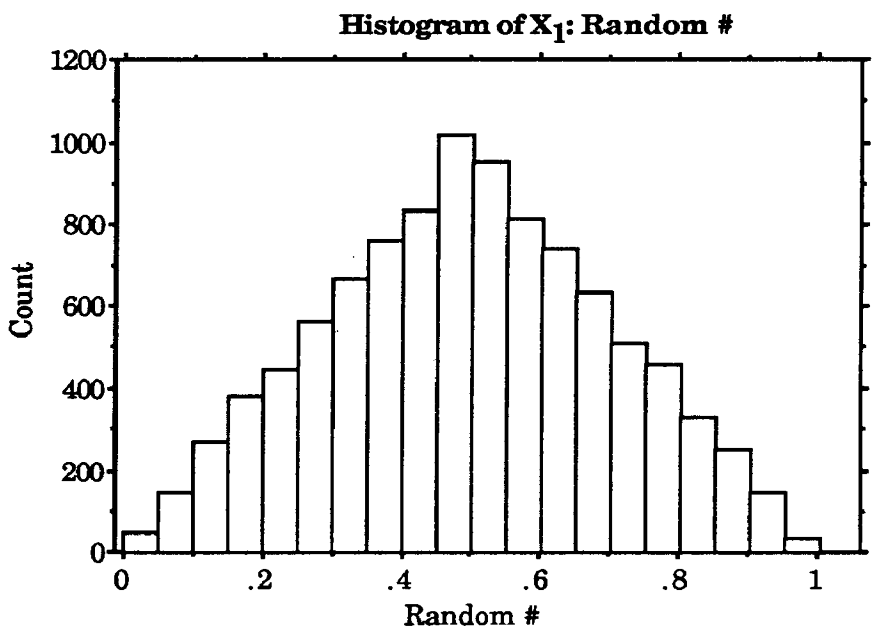

Figure D.3

APPENDIX E

Triangular Distribution Programs

```
'  Program which gives the deterministic constraint coefficients
'   of triangular distribution
'
   DIM STATIC x,a,b,c
   GOSUB DataInput
   RANDOMIZE 10                        'Uses the same random # seed for
each run
Calculations:
   PRINT"Starting calculations"        'just to let you know
  FOR i = 1 TO 5
   xmax(i,j) = 0
     FOR j = 1 TO 8
        FOR k = 1 TO N&
          IF k <> kdel(j) THEN         'skips calculating data for a
                                       'value of k already used.
            u = RND
                                       'Calculate the triangular
                                       'distribution value based on the
                                       'random # u.
            GOSUB Sample
               IF x(i,j) > xmax(i,j) THEN
               xmax(i,j) = x(i,j)
              t = j
              kdel(t) = k
             ELSE
             END IF
          ELSE
          END IF
```

```
      NEXT k
    NEXT j
  NEXT i
    GOSUB Results
END
DataInput:
   INPUT "Name of output";Prob$
  OPEN Prob$ FOR OUTPUT AS #1
   OPEN "I", #2,"SampleData"
   INPUT "Number of samples needed";N&
                                        'Input  Lowest, Most likely, and
                                        'Highest Expected values from a
                                        'file called SampleData
  FOR j = 1 TO 8
      INPUT  #2,  a(1,j),a(2,j),a(3,j),a(4,j),a(5,j)
      INPUT  #2,  c(1,j),c(2,j),c(3,j),c(4,j),c(5,j)
       INPUT #2, b(1,j),b(2,j),b(3,j),b(4,j),b(5,j)
  NEXT j
                                        'Calculate the break point
                                        'between the right and left part of
                                        'the distribution-called test(i,j)
  FOR i = 1 TO 5
      FOR j = 1 TO 8
            IF b(i,j)-a(i,j) = 0  THEN
                  test(i,j) = 0
            ELSE test(i,j) = (c(i,j)-a(i,j))/(b(i,j)-a(i,j))
            END IF
      NEXT j
  NEXT i
CLOSE #2
RETURN
Sample:
```

```
  IF u < test(i,j) THEN
   x(i,j) = a(i,j) + SQR( u * (b(i,j)-a(i,j)) * (c(i,j)-a(i,j)))
  ELSE
   x(i,j) = b(i,j) - SQR ((1-u) * (b(i,j)-a(i,j)) * (b(i,j)-c(i,j)))
  END IF
  RETURN
Results:
  FOR i = 1 TO 5
   PRINT "Constraint";i
    PRINT USING "####.##   "; xmax(i,1);xmax(i,2);xmax(i,3);xmax(i,4);
             xmax(i,5);xmax(i,6);xmax(i,7);xmax(i,8)
    PRINT #1, USING " ####.##   "; xmax(i,1);xmax(i,2);xmax(i,3);
             xmax(i,4);xmax(i,5);xmax(i,6);xmax(i,7);xmax(i,8)
  NEXT i
  CLOSE #1
RETURN
```

```
' Program which gives the values for each sample of the coefficients in the
' constraints with triangular distributions
'
  DIM STATIC x,a,b,c
  RANDOMIZE 10
  GOSUB DataInput
Calculations:
  PRINT"start calculations"
  FOR i = 1 TO 5
    FOR k = 1 TO N&
      FOR j = 1 TO 8
        u = RND
                                          'pick the upper or lower tail of the
                                          'distribution based on the value of
                                          'the random variable u.
        IF u < test(i,j) THEN
            x(i,j) = a(i,j) + SQR( u * (b(i,j)-a(i,j)) * (c(i,j)-a(i,j)))
        ELSE
            x(i,j) = b(i,j) - SQR ((1-u) * (b(i,j)-a(i,j)) * (b(i,j)-c(i,j)))
        END IF
      NEXT j
        GOSUB Results
    NEXT k
  NEXT i
    CLOSE #1
END

DataInput:
      INPUT "Name of output";Prob$
      OPEN Prob$ FOR OUTPUT AS #1
```

```
        OPEN "I", #2,"SampleData"
        INPUT "number of samples needed",N&
                                            'Input  values of Lowest, Most
                                            'from a file called SampleData
     FOR j = 1 TO 8
          INPUT #2, a(1,j),a(2,j),a(3,j),a(4,j),a(5,j)
          INPUT #2, c(1,j),c(2,j),c(3,j),c(4,j),c(5,j)
          INPUT #2, b(1,j),b(2,j),b(3,j),b(4,j),b(5,j)
     NEXT j
       FOR i = 1 TO 5
         FOR j = 1 TO 8
            IF b(i,j)-a(i,j) = 0  THEN test(i,j) = 0 ELSE test(i,j) = (c(i,j)-
a(i,j))/(b(i,j)-a(i,j))
         NEXT j
       NEXT i
       CLOSE #2
  RETURN
Results:
   PRINT "x"i; "k"k
   PRINT USING "####.##   ";
x(i,1);x(i,2);x(i,3);x(i,4);x(i,5);x(i,6);x(i,7);x(i,8)
   PRINT #1, USING "####.##   ";
x(i,1);x(i,2);x(i,3);x(i,4);x(i,5);x(i,6);x(i,7);x(i,8)
  RETURN
```

APPENDIX F

0-1 IP Program

```
'       A translation of Senju and Toyoda (1968) to solve 0-1 Integer
'       Programming Problems based on the FORTRAN program
'        by Grange (1973) and Woolsey (1975).
'
DIM STATIC A(120,250),B(120),C(250),X(250),S(120),G(250),Y(250)
Start:
  CLS
   INPUT "Want to run a problem already saved";YN$
   INPUT "Problem Name";Prob$
  IF YN$ = "n" THEN OPEN Prob$ FOR OUTPUT AS #1: GOTO Enter
  IF YN$ = "y" THEN GOTO Check
   PRINT "Type just 'y' or 'n'
  GOTO Start
Enter:
   INPUT "Max or Min";Maxmin$
   INPUT "# of Variables";n
   INPUT "# of Constraints";m
   WRITE #1,Maxmin$,n,m
  FOR j=1 TO n
   PRINT "Coefficient of X"j"in Objective Function":INPUT C(j)
  WRITE #1,C(j)
  NEXT j
  FOR i = 1 TO m
  FOR j = 1 TO n
   PRINT "Coefficient of X"j "in Constraint"i:INPUT A(i,j)
   WRITE #1,A(i,j)
  NEXT j
  NEXT i
```

```
  FOR i = 1 TO m
'
'                                    Input right hand sides
'
   PRINT "Right hand side of constraint"i:INPUT B(i)
  WRITE #1,B(i)
  NEXT i
  CLOSE #1
  GOTO Check
Solve:
CLS
'
'                                    Start timing
'
   ToolBox "I"
  TrapNo% = &HA975
  Ticks& = 0&
   ToolBox "L", TrapNo%, Ticks&
  Begin&=Ticks&
'
'                                    Check to see if problem is maximization
'
   IF Maxmin$ ="max" THEN 7
'
'                                    Problem is minimization so solve
'                                     1 's complement
'
  FOR i = 1 TO m
  B(i) = -B(i)
  FOR j = 1 TO n
  B(i) = B(i) + A(i,j)
  NEXT j
```

```
  NEXT i
'
'                                   Initialize X(j)=1
'
7 :
  FOR j = 1 TO n
  X(j) = 1
  NEXT j
'
'                                   Compute LHS surplus
'
  FOR i = 1 TO m
  S(i)=-1
  FOR  j= 1 TO n
  S(i) =S(i) + A(i,j)/B(i)
  IF S(i) < 0 THEN S(i) = 0
  NEXT j
  NEXT i
'
'                                   If any surpluses are non zero, another
'                                   variable must leave basis
'
201 :
  FOR i = 1 TO m
  IF S(i) > 0 THEN 13
  NEXT i
'
'                                   Try to add back non-basic variables
'                                     without becoming infeasible
'
  FOR i = 1 TO m
  S(i) = B(i)
```

```
    FOR j = 1 TO n
    S(i) = S(i) - A(i,j)*X(j)
    NEXT j
    NEXT i
    FOR j = 1 TO n
    IF X(j) = 0 THEN Y(j)=1
    NEXT j
23 :
    XMAX = 0
    FOR j = 1 TO n
     IF XMAX < C(j)*Y(j) THEN XMAX = C(j)*Y(j)
     IF XMAX = C(j)*Y(j) AND XMAX <> 0 THEN k = j
    NEXT j
    IF XMAX =0 THEN 25
    FOR i = 1 TO m
    IF S(i) < A(i,k) THEN 28
    NEXT i
    FOR i = 1 TO m
    S(i) = S(i) - A(i,k)
    NEXT i
    X(k) = 1
28 :
    Y(k) = 0
    GOTO 23
25 :
    IF Maxmin$ = "max" THEN 30
'
'                                  Change basis to its complement for
'                                    minimization problem
'
    FOR j = 1 TO n
    IF X(j) < 0 THEN 26
```

```
  IF X(j) = 0 THEN 261 ELSE 262
261 :
  X(j) = 1
  GOTO 26
262 :
  X(j) = 0
26 :
  NEXT j
'
'                                 Compute value of objective function
'

30 :
  ToolBox "L", TrapNo%, Ticks&
  Objfun = 0
  PRINT:PRINT "Variable" "   Cost" "      In/Out" "      Value"
  FOR j = 1 TO n
  PRINT j "      "C(j) "      "X(j) "      "C(j)*X(j)
  Objfun = Objfun +C(j)*X(j)
  NEXT j
  PRINT " "
  PRINT "********************"
  PRINT "Objective Function";Objfun
  PRINT:    PRINT "I spent";(Ticks& - Begin&)/60;"seconds on this task."
  CLOSE #1
  GOTO Finish
13 :
'
'                                 Compute gradients
'

  FOR j=1 TO n
  G(j) = 0
  FOR i = 1 TO m
```

```
 G(j) = A(i,j)/B(i)*X(j)*S(i)+G(j)
 NEXT i
 IF G(j) <> 0 THEN G(j) = C(j)/G(j)
 IF XLeg < G(j) THEN XLeg = G(j)
 NEXT j
'
'                              Find variable with the least effective
'                                 gradient and drop it from the basis
'
 FOR j = 1 TO n
 IF G(j) <> 0 AND XLeg > G(j) THEN XLeg = G(j)
 NEXT j
 FOR j = 1 TO n
 IF X(j) = 0 OR G(j) > XLeg THEN NEXT j
 X(j) = 0
'
'                              Subtract A(i,j) from the surplus
'                               in each constraint
'
 FOR i = 1 TO m
 S(i) = S(i) - A(i,j)/B(i)
 IF S(i) < 0 THEN S(i) = 0
 NEXT i
 GOTO 201

Check:
 OPEN "I", #1, Prob$
 PRINT"Is this the correct data?"
 WHILE NOT EOF(1)
   INPUT #1, Maxmin$,n,m
   PRINT  Maxmin$: PRINT n "Variables"  m "Constraints"
   PRINT "Coefficients of Objective Function"
```

```
      FOR j=1 TO n
       INPUT #1,C(j):PRINT C(j);
      NEXT j
      FOR i = 1 TO m
       PRINT:PRINT "Coefficients of Constraint"i
      FOR j = 1 TO n
       INPUT #1,A(i,j):PRINT A(i,j);
      NEXT j
      NEXT i
       PRINT:PRINT "Right Hand Sides"
      FOR i = 1 TO m
       INPUT #1, B(i):PRINT B(i);
      NEXT i
   WEND
       PRINT:INPUT " Type y OR n";YN$
      IF YN$ = "y" THEN Solve
      IF YN$ <> "n" THEN CLOSE #1: GOTO Check
      CLOSE #1
      OPEN Prob$ FOR OUTPUT AS #1
      GOTO Enter
Finish:
   INPUT "Want to try another ?",YN$
   IF YN$ = "y" THEN Start
   IF YN$ <> "n" THEN Finish
END
```

APPENDIX G

Verification Program and Results

```
'  Program to test the sample problem results
'
DIM Alpha(1200)
DIM STATIC x, a, b, c, test, Count, Alpha
RANDOMIZE 10                    'Uses the same random # seed for each run
  GOSUB DataInput
Calculations:
   PRINT"Starting calculations"
      FOR i = 1 TO 5
            AlphaSum = 0
            z = 0
            Sum = 0
        FOR Repeat = 1 TO M&   '100 iterations to get enough for alpha-bar
            Count = 0           'Count is # of times a constraint is violated
         FOR k = 1 TO N&
            GOSUB Sample
            LHS = x(i,1) + x(i,2) + x(i,3)
            IF LHS > RHS(i) THEN Count = Count + 1
         NEXT k
            Hold = N& - Count
            Alpha(Repeat) = Hold / N&
             AlphaSum = AlphaSum + Alpha(Repeat)
        NEXT Repeat
           AlphaBar = AlphaSum / M&
           GOSUB AlphaVar
           GOSUB Results
      NEXT i
CLOSE #1
```

```
END

DataInput:
   INPUT "Name of output file";Prob$
   OPEN Prob$ FOR OUTPUT AS #1
   OPEN "I", #2,"SolutionData"
   INPUT "Number of samples needed (57)";N&
   INPUT "Number of iterations needed (100)";M&
                         '  Input values of Lowest (a), Most likely (c),
                         '  and Highest (b) Expected values from a file
                         '  called SolutionData
  FOR j = 1 TO 3
     INPUT #2, a(1,j),a(2,j),a(3,j),a(4,j),a(5,j)
     INPUT #2, c(1,j),c(2,j),c(3,j),c(4,j),c(5,j)
     INPUT #2, b(1,j),b(2,j),b(3,j),b(4,j),b(5,j)
  NEXT j
    INPUT #2, RHS(1),RHS(2),RHS(3),RHS(4),RHS(5)
                         'Calculate the break point between the right
                         'and left part of the distribution-called test(i,j)
  FOR i = 1 TO 5
    FOR j = 1 TO 3
      IF b(i,j)-a(i,j) = 0  THEN
            test(i,j) = 0
      ELSE
            test(i,j) = (c(i,j)-a(i,j))/(b(i,j)-a(i,j))
      END IF
    NEXT j
  NEXT i
  CLOSE #2
    RETURN
Sample:                        'Sample a value from a triangular distribution
                               'based on a random number (u).
```

```
  FOR j = 1 TO 3
       u = RND
       IF u > test(i,j) THEN
        x(i,j) = b(i,j) - SQR ((1-u) * (b(i,j)-a(i,j)) * (b(i,j)-c(i,j)))
       ELSE
        x(i,j) = a(i,j) + SQR( u * (b(i,j)-a(i,j)) * (c(i,j)-a(i,j)))
       END IF
  NEXT j
    RETURN
AlphaVar:
       FOR Repeat = 1 TO M&
            z = (Alpha(Repeat) - AlphaBar) ^2
            Sum = z + Sum
       NEXT Repeat
            AlphaVar = Sum / (M& -1)
    RETURN
Results:
   PRINT "Results of Constraint"i:PRINT
   PRINT "Alpha Bar = "AlphaBar
    PRINT "Alpha Variance ="AlphaVar
   PRINT"**********************":PRINT
    RETURN
```

Results of sampling each constraint 57 times to obtain observed α's and repeating this for 100 iterations to obtain the mean and variance (Alpha Bar and Alpha Variance) for each constraint. Notice that each observed α is greater than the α selected (.80), β =1 for each constraint. The values set for the parameters α and β would hold if this test were repeated for each feasible solution.

Results of Constraint 1
Alpha Bar = 1
Alpha Variance = 0
Beta = 1
****
Results of Constraint 2
Alpha Bar = .9757888
Alpha Variance = 4.214505E-04
Beta = 1
****
Results of Constraint 3
Alpha Bar = .9840347
Alpha Variance = 3.425759E-04
Beta = 1
****
Results of Constraint 4
Alpha Bar = .9942103
Alpha Variance = 9.361061E-05
Beta = 1
****
Results of Constraint 5
Alpha Bar = .9722802
Alpha Variance = 4.239374E-04
Beta = 1
****

The observed values of α and β appear to be greater than may be expected. The reason for this is due to the phenomenon of duality gaps in integer programming, where constraints can be binding but not tight. This can be shown graphically for three variables in Figure G.1. The corner points of the cube represent the feasible solutions and the curved surface represents the chance-constrained set (S). The linear constraint set (S_L) may not contain all of the feasible solutions in S. Also, the optimal solution may not be on the boundary of S_L as in linear programming.

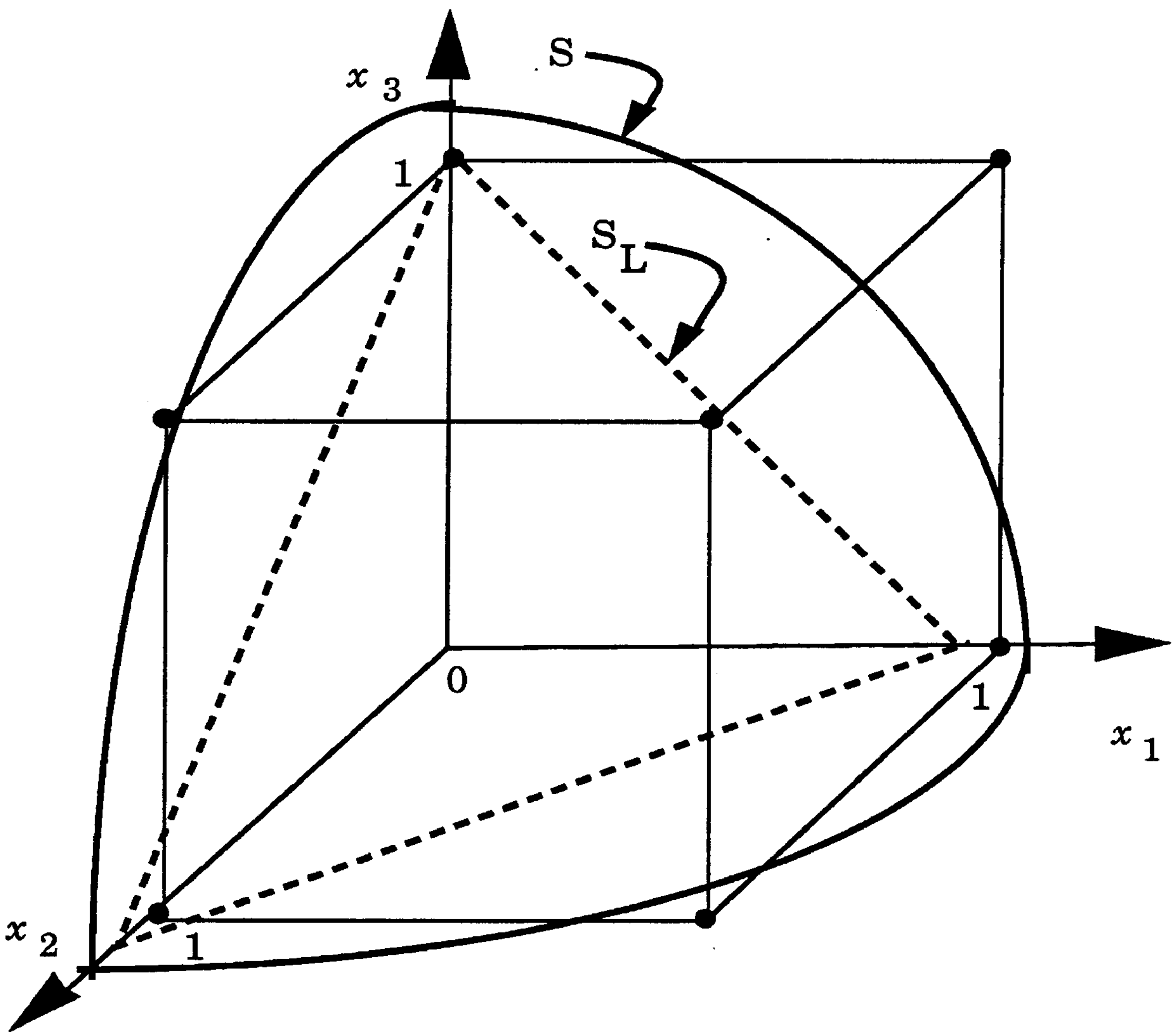

Figure G.1