EVALUATION OF SOLUTION METHODS FOR DETERMINISTIC EQUIVALENTS OF CHANCE-CONSTRAINED MODELS

HÜSEYIN SARPER*

SUMMARY: Once chance-constrained programs (CCP) are converted into deterministic equivalents, nonlinear terms result. This note compares three approximation methods and reviews other solution methods to solve the resulting nonlinear programs. Naslund's approximation is recommended as an easy and effective method.

Key Words: Stochastic programming, nonlinear optimization, linearization of nonlinear terms.

INTRODUCTION

Risk is a major element in decision process. CCP is one way to include risk into ordinary linear mathematical models. CCP describes constraints in the form of probability levels of attainment. Rao (7) provides a complete treatment on the origins and the theory of CCP. In CCP, coefficients have typically been assumed to follow a normal distribution with known means and variances. Consider the following model:

$$\text{Maximize} \quad F(X) = \sum_{j=1}^{n} c_j x_j \tag{1}$$

subject to

$$\Pr\left[\sum_{j=1}^{n} a_{ij} x_j \le b_i\right] \ge p_i \quad \text{for } i = 1, 2, \ldots\ldots\ldots\ldots m \tag{2}$$

$$x_j \,\varepsilon [0,1] \quad \text{for } j = 1, 2, \ldots n \tag{3}$$

Where c_j, a_{ij}, and b_i are normal random variables. p_i's are pre-specified probabilities. The above model represents the extreme case of CCP when all three coefficient types are random variables. Decision variables, X_j's, are not always binary variables, but many real-life decision processes often involve such variables. Typically, b_i's and/or a_{ij}'s are considered as random. If c_j's are also normal random variables, the objective function, F(X), will also be a normal variable. This, however, does not mean that the distribution of the optimum value itself will be a normal one too.

* From University of Southern Colorado, 2200 Bonforte Boulevard, Pueblo, CO 81001-4901, USA.

AN EXAMPLE

Consider the following product selection problem in which available resources and the required resources for each product are both normal and independent variables. The profit coefficients, however, are deterministic.

Maximize $[10X_1 + 15X_2 + 20X_3 + 14X_4]$ subject to
$(100;5)X_1+(150;6)X_2+(215;8)X_3+(85;3)X_4 \le (500;15)$
$(25;2)X_1 + (15;2)X_2 + (10;2)X_3 + (35;3)X_4 \le (74;4)$
$(40;3)X_1 + (0.5;0.1)X_2 + (20;2)X_3 + (5;1)X_4 \le (60;5)$
$X_1, X_2, X_3, X_4, \varepsilon[0,1]$

The numbers in each parenthesis indicate the mean and the standard deviation of the respective coefficient. If management requires that each constraint should have at least 99% probability of not being violated, the constraints take the following deterministic equivalent, but nonlinear form.

$100X_1+150X_2 + 215X_3 + 85X_4 + 2.33(25X_1^2 + 36X_2^2 + 64X_3^2 + 9X_4^2 + 225)^{1/2} \leq 500$

$25X_1 + 15X_2 + 10X_3 + 34X_4 + 2.33(4X_1^2 + 4X_2^2 + 4X_3^2 + 9X_4^2 + 16)^{1/2} \leq 74$

$40X_1 + 0.5X_2 + 20X_3 + 5X_4 + 2.33(9X_1^2 + 0.01X_2^2 + 4X_3^2 + X_4^2 + 25)^{1/2} \leq 60$

Complete enumeration shows that $X_1 = 0$ and $X_2 = X_3 = X_4 = 1$ is the optimal answer. 2.33 is the standard normal variate for 0.99 probability.

The difficulty of CCP solution starts at this point. Except only when b_i's are random, each constraint and/or the objective function consist of two segments: linear and nonlinear terms. The purpose of this note is to compare three methods that approximate the nonlinear segment into a linear one. The nonlinear segment has the form of

$$\left(\sum_{m=1}^{M} V_m X_m^2\right)^{1/2}$$

where V_m is the variance, if any, of the b_i term. The problem has m-1 decision variables with variances V_1 through V_{m-1}. Once approximated, variable coefficients of the linearized segment are added to the linear segment coefficients to end up with a completely linear model. Wasil, *et al.* (12) review six major nonlinear optimization software for PC's, but the authors show that such codes do not always perform well. That is not to say that the nonlinear programs encountered in CCP cannot be optimally solved using nonlinear programming methods. Even with new powerful packages, typical solution may require that the user make initial guesses and accept solutions optimal to some tolerances. If the non-linearity can be eliminated, it is clear that much larger programs can be easily solved with efficient LP packages. To solve CCP's, Rakes, *et al.* (6) and Lee and Olson (2) have used separable programming after significant amount of pre-processing even for the small problems they present. Tabucanan, *et al.* (10) have developed an elaborate sectioning search method for another small problem. These procedures are too time-consuming and difficult to repeat in solving problems with large number of variables and constraints. Another method based on cutting plan algorithm has been recently suggested in (11), but this method is also hard to implement and suitable only in continuous decision variables case.

APPROXIMATION METHODS

Byrne, *et al.* (1) approximates the non-linearity in objective function by converting each normal variable into a 3-point discrete distribution. This method is too cumbersome and not considered further. Three other methods are discussed.

Naslund's approximation (3)

The nonlinear square root portion of each constraint of the deterministic equivalent is converted into an approximate linear form and then added to the rest of the constraint. If the objective function has any nonlinear (square root) terms due to stochastic c_j coefficients, this approximation, shown below, can also be used to linearize the objective function.

$$\left\{\sum_{m=1}^{M} V_m X_m^2\right\}^{1/2} \leq \left\{\sum_{m=1}^{M} V_m\right\}^{1/2} - \sum_{n=1}^{M}\left\{(1-x_n)\left[\left(\sum_{m=1}^{M} V_m\right)^{1/2} - \left(\sum_{m=1}^{M} V_m - V_n\right)^{1/2}\right]\right\} \quad (4)$$

At the end of the approximation process, a constant is obtained and it is carried over to the right hand side after changing its sign. This method requires no additional variables or constraints. The right hand side of (4) can be easily programmed as a subroutine of any MPS formatted input generator code. Then, any general purpose mathematical optimizer (LINDO) can be used for solution.

Chance-constrained programming system (CHAPS) [8, 9]

This method is based on the separation, linearization and iterative adjustment of the nonlinear constraints of deterministic equivalents. Reference (8) states that this method is suitable when the number of variables is between two and two hundred. Unlike Naslund's method, CHAPS cannot be used if c_j's are also random variables. Although not clearly stated, this method is applicable with continuous, not 0-1, type linear programs. References (8,

9) contain numerous examples, but none of them is a 0-1 type problem. To briefly illustrate how CHAPS works, consider a nonlinear constraint for any of the original constraints of the example above. Each can be expressed as follows:

$$\sum_{j\varepsilon J} \bar{a}_{ij} + e_i \left[\mathrm{Var}\left(b_i\right) + \sum_{j\varepsilon J} \mathrm{Var}\left(a_{ij}\right) X_j^2 \right]^{1/2} \leq \bar{b}_i$$

for all $i\varepsilon I$, (5)

Indices I and J represent the constraint and the variable sets. Term e_i is the normal variate. The set of feasible solutions is enlarged by introducing a new slack variable for each constraint. The separated form of (5) above is written as:

$$\sum_{j\varepsilon J} \bar{a}_{ij} X_j + e_i y_{in} \leq \bar{b}_i \qquad i\varepsilon I, \tag{6}$$

$y_{ij} \geq (y^2_{i\,j-1} + \mathrm{Var}(a_{ij})X^2_j)^{1/2}$ $i\,\varepsilon I, j\varepsilon J$, (7)

$y_{io} = [\mathrm{Var}(b_i)]^{1/2}$ and $X_j \geq 0$ $i\,\varepsilon I, j\varepsilon J$, (8)

where n is the largest index number in set J and y_{ij}'s are new additional non-negative decision variables. All constraints of type (7) above are replaced by linear approximate constraints of:

$-y_{ij} + r_{ijk} y_{i,j-1} + s_{ijk} X_j \geq 0$ $i\varepsilon I, j\varepsilon J, k=1...p$, (9)

The index p is the degree of linearization or fineness and r_{ijk} and s_{ijk} are constants whose formulas are given in references (8) and (9). The authors state that p=6 to 8 is sufficient to reach optimality in most problems. The approximate size of the resulting linear programming problem is r + n variables and 6r + m constraints. r, n, and m are the number of random variables, number of decision variables, and the number of rows of the original CCP.

Olson and Swenseth's approximation (4)

Olson and Swenseth (4) present a method which places a bound on the chance constraint at least as tight as the nonlinear form, thus overachieving the chance constraint at the expense of the other constraints or the objective function. This method is based on convexity of the general variance-covariance matrix and is applicable only for continuous CCP models. Since 0-1 type problems are very important in managerial decision-making, this approximation method is not always applicable. The authors show that,

$$\sum_{m=1}^{M} \left(V_m X_m^2\right)^{1/2} \leq \left(V_1^{1/2} X_1 + \ldots\ldots\ldots + V_m^{1/2} X_m\right) \tag{10}$$

Simply, this method suggests the replacement of the nonlinear term by linear right hand side shown in [10].

COMPARISON OF THE THREE METHODS

The deterministic equivalent (nonlinear) of the example problem was solved using the first two methods. The CHAPS required 18 variables, 54 constraints and found an integer optimum of $35 (0-1-1-0) after 53 LINDO iterations. The continuous optimum value was $48.45. Naslund's method of approximating the problem into a linear 0-1 program required only 4 variables and 3 constraints as in the original problem. After 6 LINDO iterations, integer optimum of $49 (0-1-1-1 or the correct answer) and continuous optimum of $49.01 were found. The linearized constraints using Naslund's method are shown below.

$101.56X_1+152.27X_2 + 219.13X_3 + 85.56X_4 \leq 464.37$
$25.79X_1 + 15.79X_2 + 10.79X_3 + 36.84X_4 \leq 64.03$
$41.79X_1 + 0.50X_2 + 20.77X_3 + 5.19X_4 \leq 48.19$

In this example, Naslund's approximation (IP which results after using this approximation and its solution) appears better than CHAPS with respect to computation time, problem size and the optimum value found. The lower objective function value (inferior) of CHAPS may be explained with use of a linearization factor of only four in the example and the fact that this is a 0-1 type problem. Solution using the last method is not attempted because reference [4] only considers the case when a_{ij}'s are random variables.

Cattle feed problem with continuous variables

Minimum cost cattle feed problem under a probabilistic protein constraint (other constraints are deterministic) was originally formulated in [5]. It was desired that this constraint hold with at least 95% probability. To test the performance of CHAPS, Seppälä [9] compared CHAPS solution against the feasible direction algorithm of Zoutendijk's used in [5]. To test effectiveness of their approximation, Olson and Swenseth [4] also solved the same problem, but required that each $X_j \geq 0.01$ rather than 0 as in Seppälä's comparison. Here, the problem is re-solved using $X_j \geq 0$. This problem is also solved using Naslund's method of linearizing the single chance-constraint involved. Table 1 compares all four methods.

Table 1: Comparison of five solution methods for the cattle feed problem.

Method	Decision Variables				Objective
used	X_1	X_2	X_3	X_4	Z
Zoutendijk's	0.635900	0	0.312700	0.051500	29.8924
CHAPS	0.635875	0	0.312660	0.051459	29.8888
Olson's	0.621274	0	0.314560	0.064570	30.1195
Naslund's	0.609260	0	0.315380	0.075360	30.3093
GRGM	0.635876	0	0.312666	0.051458	29.8888

Although Naslund's approximation yields the highest minimum cost, it must be remembered that this approximation does not require any additional variables or constraints as is the case with CHAPS. For this classic problem, the third method gives a minimum cost which is 0.63% less than that of Naslund's. Finally, the problem was solved using generalized reduced gradient method (GRGM). Table 1 shows that nonlinear programming solution using GRGM finds the minimum found by CHAPS.

Linearization of square root of sum of 0/1 variables

The purpose of this example is to demonstrate the accuracy of Naslund's approximation in linearizing a typical nonlinear expression often encountered in 0/1 CCP problems. Consider the expression,

$$(103.7X^2_1 + 112.5X^2_2 + 68.5X^2_3 + 76X^2_4 + 40X^2_5 + 102X^2_6 + 61X^2_7 + 75X^2_8 + 14X^2_9 + 36X^2_{10})^{1/2} \qquad (11)$$

where X1...X10 ε[0.1]. Application of Naslund's approximation to the above nonlinear expression results in the following linear form:

$$2.05X_1 + 2.24X_2 + 1.34X_3 + 1.49X_4 + 0.77X_5 + 2.02X_6 + 1.19X_7 + 1.47X_8 + 0.26X_9 + 0.69X_{10} + 12.72 \qquad (12)$$

The constant, 12.72, has to be ignored if [11] is a part of an objective function and carried over to the right hand side if [11] is part of a constraint. If all X's are 0, then [12] equals to 12.72. This represents the largest error value between the original expression [11] and its approximation [12]. If all X's are 1's, then [11] equals to 26.2431 while [12] results in 26.2400 with error rate of 0.012%. In project selection type decisions, the number of variables are often large and the approximation performs better as the number of variables increases. A code is used in comparing the actual values of [11] with that of [12] for all 1.024 combinations of the ten 0-1 variables: The average amount of error is 8.9% over all combinations, but the error rate falls rapidly as the number of 1's is a given combination is increased. For example:

- The average error is 7.93% if there are at least two 1's.

- The average error is 3.30% if there are at least five 1's.

- The average error is 2.00% if there are at least seven 1's.

If all X's are 0, Olson and Swenseth's approximation gives the correct value of 0, but results in a value of 80.333 if all X's are 1's. Clearly, this method does not apply in 0-1 case. As stated previously, CHAPS method is also not applicable in 0-1 case.

CONCLUSION

This note has shown that the concept of CCP is solvable. The main thrust has been to emphasize that Naslund's approximation method is the easiest way to linearize the non-linear terms often encountered in solving CCP's. Other solution methods either require special skills that some practicing analysts may be lacking (direct nonlinear solution or separable programming) or not applicable for all problem types. At least, Naslund's method would be an excellent way to get a quick initial solution for a large CCP problem. This solution can then be used as a bound in solving the problem as a nonlinear program.

A sophisticated analyst may approach the problem using sequential quadratic programming method (SQP) or even the sequential linear programming algorithm. Both of these methods automatically linearize the problem by computing (usually by finite differences) the gradients of the constraints. Starting approximations would be required. In addition, if integrality constraints are required, branch and bound extension could be added to the SQP procedure. Again, this approach would be difficult for most practicing analysts.

REFERENCES

1. Byrne R, et al : A Chance-Constrained Approach to Capital Budgeting. Journal of Financial and Quantitative Analysis, 2:339-364, 1967.

2. Lee SM, Olson DL : A Gradient Algorithm for Chance Constrained Nonlinear Goal Programming. European Journal of Operations Research, 22:359-369, 1985.

3. Naslund B : A Model of Capital Budgeting Under Risk. Journal of Business, 39:257-271, 1966.

4. Olson DL, Swenseth SR : A Linear Approximation for Chance-Constrained Programming. Journal of Operational Research Society, 38:261-267, 1987.

5. Panne VD, Popp V : Minimum-Cost Cattle Feed Under Probabilistic Protein Constraints. Management Science, 9:405-430, 1962.

6. Rakes TR, et al : Aggregate Production Planning Using Chance-Constrained Goal Programming. International Journal of Production Research, 22:673-684, 1984.

7. Rao S : Optimization Theory and Applications, second edition, J Wiley and Sons NY, 1984.

8. Seppälä Y, Orpana T : Experimental Study on the Efficiency and Accuracy of a Chance-Constrained Programming Algorithm. European Journal of Operations Research, 16:345-357, 1984.

9. Seppälä Y : A Chance-Constrained Programming Algorithm. BIT, 12:376-399, 1972.

10. Tabucanan MT, et al : A Probabilistic Programming Model for Blending Aggregates. Applied Mathematical Modeling, 4:257-260, 1980.

11. Weintraub A, Vera J : A Cutting Plane Approach for Chance-Constrained Linear Programs. Operations Research, 39:776-785, 1991.

12. Wasil E, et al : State of the Art in Nonlinear Optimization Software for the Microcomputer. Computers and Operations Research, 16:497-512, 1989.

Correspondence:
Hüseyin Sarper
University of Southern Colorado,
2200 Bonforte Boulevard,
Pueblo, CO 81001-4901, USA.