

Please cite this article as:

Andrzej Grzybowski, Monte Carlo optimization procedure for chance constrained programming - simulation study results, Scientific Research of the Institute of Mathematics and Computer Science, 2009, Volume 8, Issue 1, pages 39-46.

The website: <http://www.amcm.pcz.pl/>

Scientific Research of the Institute of Mathematics and Computer Science

MONTE CARLO OPTIMIZATION PROCEDURE FOR CHANCE CONSTRAINED PROGRAMMING - SIMULATION STUDY RESULTS

Andrzej Grzybowski

*Institute of Mathematics, Czestochowa University of Technology, Poland
azgrzybowski@gmail.com*

Abstract. In the paper a chance constrained linear programming problem is considered in the case of joint chance constraints with random right hand sides. It is assumed that due to its complex stochastic nature the problem cannot be reduced to any deterministic equivalent problem. In such a case a Monte Carlo method involving evolutionary algorithms with soft selection are proposed to solve the problem. The simulation results are presented and discussed.

Introduction

Chance Constrained Programming (CCP) or, more general, stochastic programming deals with a class of optimization models and algorithms in which some of the data may be subject to significant uncertainty. Such models are appropriate when data cannot be observed without error or evolve over time and decisions have to be made prior to observing the entire data stream. The concept of CCP was introduced in the classical work of Charnes and Cooper [1]. Now CCP belongs to the major approaches for dealing with random parameters in optimization problems. Typical areas of application are engineering design applications, see [2], finance (e.g.[3]), budgeting [4, 5] or portfolio analysis [6]. In models built for such real-world problems uncertainties like product demand, cost of supply, price of a final product, demographic conditions, currency exchange rates, rates of return etc. enter the inequalities describing the natural constraints that should be satisfied for proper working of a system under consideration.

Stochastic optimization problems belong to the most difficult problems of mathematical programming. Most of the existing computational methods are applicable only to convex problems. There are, however, many important applied optimization problems which are, at the same time, stochastic and non-convex. Many of them are also multi-extremal problems. Discussion of various computational aspect of CCP problems can be found in [6-8] or [9]. In our paper the method of evolutionary search with soft selection is proposed in order to find a *satisfactory* solution. The statistical performance of such a solution is studied via computer simulations.

1. Chance Constrained Linear Programming Problem

Let us consider a classical (deterministic) linear programming problem:

$$\text{maximize } f(x_1, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

Subject to the following constraints (s.t.):

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n &\leq b_i \quad i = 1, \dots, m \\ x_1 \geq 0, \dots, x_n &\geq 0 \end{aligned}$$

where f is the objective function, $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$ is the decision variable vector, $\mathbf{A} = [a_{ij}]_{m \times n}$ is the matrix of (technical) coefficients of the system of linear inequalities, a coefficient vector $\mathbf{b} = [b_1, b_2, \dots, b_m]^T$ will be addressed as a right hand side of the constraints system, $\mathbf{c} = [c_1, c_2, \dots, c_n]^T$ is a vector of the objective function coefficients.

As we have mentioned before, in many applications the elements of the tuple $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ cannot be considered as known constants. All or part of them are uncertain. Thus it is difficult or even impossible to know which solution will appear to be feasible. In such circumstances, one would rather insist on decisions guaranteeing feasibility 'as much as possible'. This loose term refers to the fact that constraint violation can almost never be avoided because of unexpected (or simply random) events. On the other hand, after proper estimating the distribution of the random parameters, it makes sense to call decisions feasible (in a stochastic meaning) whenever they are feasible with high probability, i.e., only a low percentage of realizations of the random parameters leads to constraint violation under this given decision. It leads to CCP formulation of the problem, where the deterministic *constraint* are replaced with a *probabilistic* or *chance ones* in one of the two following ways:

1.1. Individual chance constraints

$$\begin{aligned} \text{maximize } E f(x_1, \dots, x_n) &= E(c_1)x_1 + E(c_2)x_2 + \dots + E(c_n)x_n \\ \text{s.t.} & \\ \Pr(a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n &\leq b_i) \geq q_i \quad i = 1, \dots, m \\ x_1 \geq 0, \dots, x_n &\geq 0 \end{aligned}$$

where $\mathbf{q} = [q_1, q_2, \dots, q_m]$, $q_i \in [0, 1]$, $i = 1, \dots, m$, is the vector of prescribed values of so called probability levels given for each constraint separately.

1.2. Joint chance constraints

$$\begin{aligned} \text{maximize } E f(x_1, \dots, x_n) &= E(c_1)x_1 + E(c_2)x_2 + \dots + E(c_n)x_n \\ \text{s.t.} & \end{aligned}$$

$$\Pr(a_{i1}x_1+a_{i2}x_2+\dots+a_{in}x_n \leq b_i, \quad i = 1, \dots, m) \geq q$$

$$x_1 \geq 0, \dots, x_n \geq 0$$

where $q \in [0,1]$ is the probability level.

The value of probability level(s) is chosen by the decision maker in order to model the safety requirements. Sometimes, the probability level is strictly fixed from the very beginning (e.g., $q = 0.95, 0.99$ etc.). In other situations, the decision maker may only have a vague idea of a properly chosen level. It is obvious that higher values of q lead to fewer feasible decisions \mathbf{x} , and hence to smaller optimal values of expected gain. In some simple cases (especially in case of individual chance constraints) the problem can be replaced with its deterministic equivalent, see e.g. [4, 9].

The main challenge in designing algorithms for general stochastic programming problems arises from the need to calculate conditional expectation and/or probability associated with multi-dimensional random variables, see [9, 8, 11]. This makes the CCP problems most difficult problems of mathematical programming. The computational challenges and methods in the field of optimization under uncertainty are addressed e.g. in [8] and [9]. A brief survey on some of the most relevant developments can be found in [7].

In our paper we consider the situation where, due to assumed complex stochastic nature of the problem no deterministic equivalent is available. In order to find satisfactory stochastically feasible solution we propose a criterion based on expected utility of a given solution and adopt an algorithm of evolutionary search with soft selection.

2. Problem formulation

In our studies we examine the linear programming models in the case where all problem describing parameters i.e. the matrix \mathbf{A} and vectors \mathbf{b} , \mathbf{c} , are random with the expectations equal $E(\mathbf{A}) = \mathbf{\Lambda}$, $E(\mathbf{b}) = \mathbf{\beta}$, $E(\mathbf{c}) = \mathbf{\chi}$. In the sequel such a problem will be denoted $CCLP(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$. The performance of the solution found by the Monte Carlo method for the $CCLP(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$ is compared with the performance of the optimal solution found for the deterministic linear programming problem given by the parameters $\mathbf{\Lambda}$, $\mathbf{\beta}$, $\mathbf{\chi}$ - in the sequel the latter will be denoted by $DLP(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$.

The decision-maker dealing with the $CCLP(\mathbf{\Lambda}, \mathbf{\beta}, \mathbf{\chi})$ problem should maximize both the probability q that a given systems of constraints will be satisfied and the expected value of the objective function. However, the goals appear to be contradictory (at least to some extent) and therefore in our studies we propose to use the following index of performance of a given solution:

$$IP(\mathbf{x}) = p_{\mathbf{x}}U(E(f(\mathbf{x})))$$

where p_x is the probability that the system of constraints is satisfied when one use the solution \mathbf{x} , U is a utility function which allow to take into account the decision maker preferences connected with both goals. The index IP can be interpreted as a conditional expected utility of the expected value of the objective function, under the condition that the system of constraints is satisfied. We assume that if the condition is not fulfilled then the utility of any gain equals zero.

The Monte Carlo method - generally speaking - is a numerical method based on random sampling. It is therefore a method which allow to analyze a given decision rule in terms of its statistical performance. Thus in our studies the above index of performance takes the following statistical form:

$$SIP(\mathbf{x}) = \frac{N_s}{N_{SIP}} \cdot U\left(\frac{1}{N} \sum_{i=1}^N f_i(\mathbf{x})\right)$$

where N_{SIP} is a number of i.i.d. realizations of $CCLP(\mathbf{A}, \mathbf{\beta}, \mathbf{\chi})$, N_s is the number of successful realizations (i.e. the realizations for which the system of constraints was satisfied), $f_i(\mathbf{x})$ is the value of the objective function obtained in the i -th random realization of the problem. A single random realization of $CCLP(\mathbf{A}, \mathbf{\beta}, \mathbf{\chi})$ is the realization of a random tuple $(\mathbf{A}, \mathbf{b}, \mathbf{c})$ with the probability distribution satisfying the condition $E(\mathbf{A}) = \mathbf{A}$, $E(\mathbf{b}) = \mathbf{\beta}$, $E(\mathbf{c}) = \mathbf{\chi}$.

In our simulation study we take into account the problems with positive objective functions and we use the utility function given by:

$$U(y) = \sqrt{y} \mathbf{1}_{[0, \infty)}(y)$$

3. Algorithm of the evolutionary search with soft selection (ES-SS)

To find the solution to the $CCLP(\mathbf{A}, \mathbf{\beta}, \mathbf{\chi})$ problem we adopt the algorithm of the evolutionary search with soft selection described e.g in [10, 12]. The algorithm implemented in our simulations is as follows:

- Step 1. Set the initial population of k vectors $\mathbf{v}_i \in \mathbb{R}^n$, $i = 1, 2, \dots, k$ (it is so called initial parent population).
- Step 2. Assign to each vector \mathbf{v}_i , $i = 1, \dots, k$, its fitness i.e. the value of the criterion $SIP(\mathbf{v}_i)$.
- Step 3. Select parent \mathbf{v} by *soft selection* i.e. with probability proportional to the its *fitness*.
- Step 4. Create a descendant \mathbf{w} from the chosen parent \mathbf{v} by its random mutation: $\mathbf{w} = \mathbf{v} + \mathbf{Z}$, where \mathbf{Z} is a random n -dimensional vector with coefficients having expected value equal to zero and given standard deviation σ_z .
- Step 5. Repeat steps 3 and 4 for k times to create a new k -element generation of n -dimensional vectors (descendants).
- Step 6. Replace the parent population with the descendant population

Step 7. Repeat the second to sixth steps for N_G times, where N_G is a sufficiently large number.

Step 8. Return the last generation and the fitness of its elements.

In our simulations the initial population in Step 1 was generated as a population of mutations of the optimal solution of the $DLP(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ problem. We also record the value of the best fitness achieved during the simulations and the vector \mathbf{v} the best value was assign to. The latter is the solution of $CCLP(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ problem found by the evolutionary search - it is denoted by \mathbf{x}_{ES} . The optimal solution for the $DLP(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ problem is denoted by \mathbf{x}_D .

4. Simulation study of the solutions performance

To estimate the performance of the solution \mathbf{x}_{ES} we compare it with the performance with optimal deterministic solution \mathbf{x}_D in two frameworks: stochastic and deterministic (ideal). To do this we propose the following performance indicators:

- the statistical performance rate: $SPR = \frac{SIP(\mathbf{x}_{ES})}{SIP(\mathbf{x}_D)}$
- the rate between the $SIP(\mathbf{x}_{ES})$ and the utility of the optimal objective function value in deterministic case: $SDR = \frac{SIP(\mathbf{x}_{ES})}{U(f(\mathbf{x}_D))}$

The values of the indicators obviously depend on the problem parameters, i.e. on the tuple $(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ and the dimensions of its elements. Thus we compute the values of the indicators for various values of the parameters and compare its statistical characteristics: maximum value, minimum value, median, mean value and standard deviation. In order to obtain the statistical data we use the following simulation procedure.

Step 1. Set the parameters n, k, N_{SIP}, N_G and K .

Step 2. Randomly generate the tuple $(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$.

Step 3. Solve the $DLP(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ problem by the simplex algorithm and obtain the solution \mathbf{x}_D , the optimal value $f(\mathbf{x}_D)$ and $SIP(\mathbf{x}_D)$

Step 4. Solve $CCLP(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ problem by the ES-SS algorithm and obtain the solution \mathbf{x}_{ES} and $SIP(\mathbf{x}_{ES})$.

Step 5. Compute and record the values of SPR and SDR for this setup.

Step 7. Repeat the second to fifth steps for K times, where K is a sufficiently large number.

Step 8. Return: maximum values, minimum values, medians, mean values and standard deviations of SPR and SDR .

In our research we use the following values of the parameters: $N_{SIP} = 500$, $n = 5, 10, 15$ and $k = 10$, $K = 50$. In the ES-SS algorithm the distributions of mutations are normal with constant standard deviation equal to 0.1. The values of the parameter m are drawn from the set $\{n-2, \dots, n+5\}$. The elements of the tuple $(\mathbf{A}, \mathbf{\beta}, \boldsymbol{\chi})$ are

drawn from the interval $[-200,700]$. In the *SIP* procedure the distributions of each element of the matrix and both vectors are normal with mean equal zero and standard deviation being equal to 10% of the value of the element.

5. Results and final remarks

In the following tables we present the results of our simulations. The first one shows the results obtained for the indicator *SPR* in case of $n = 5,10,15$.

Table 1. Statistical characteristics of *SPR*, $n = 5,10,15$

<i>SPR</i> , $n = 5$				
Min	Max	Median	Mean	Stand. Dev.
3.49	16.11	6.51	7.77	3.95
<i>SPR</i> , $n = 10$				
Min	Max	Median	Mean	Stand. Dev.
3.07	216.87	19.28	42.50	53.96
<i>SPR</i> , $n = 15$				
Min	Max	Median	Mean	Stand. Dev.
6.94	98.36	66.95	55.56	36.64

We can see that the performance of the solution \mathbf{x}_{ES} in comparison with the performance of the deterministic optimal solution \mathbf{x}_D is very good. The expected utility of \mathbf{x}_{ES} is at least (see first column) three times greater than the expected utility of \mathbf{x}_D . The ratio of the expected utilities may be even greater than 200, see the second column of the Table 1. In average, the greater is the number of the decision variables n , the greater is the ratio, see columns fourth and fifth.

Another question is how big is the *expected* utility of the mean value of the objective function when we use the solution \mathbf{x}_{ES} in comparison with the utility of the optimal value of the objective function in the ideal, deterministic case. The answer is given by the values of the indicator *SDR* presented in Table 2.

Table 2. Statistical characteristics of *SDR*, $n = 5,10,15$

<i>SDR</i> , $n = 5$				
Min	Max	Median	Mean	Stand. Dev.
63%	90%	87%	85%	6.0%
Number of best generation, $n = 10$				
Min	Max	Median	Mean	Stand. Dev.
23%	85%	80%	77%	14%
Number of best generation, $n = 15$				
Min	Max	Median	Mean	Stand. Dev.
77%	86%	80%	81%	28%

We see that the average expected utility amounts to about 80% of the ideal optimal value. Because the standard deviations are rather small it indicates a very good performance of the solution x_{ES} . Obtained values of the index *SDR* show that any other algorithms cannot gain much better performance in the stochastic framework.

The last question addressed in this paper is how many generation should be created to obtain a satisfactory solution. Table 3 provides us with the statistical characteristics of the number of a generation containing best element in our simulations.

Table 3. Number of the best generation - statistical characteristics, $n = 5, 10, 15$

Number of best generation, $n = 5$				
Min	Max	Median	Mean	Stand. Dev.
2	99	23	36.85	33.69
Number of best generation, $n = 10$				
Min	Max	Median	Mean	Stand. Dev.
1	52	17	21.64	15.53
Number of best generation, $n = 15$				
Min	Max	Median	Mean	Stand. Dev.
1	67	36	36,16	20.65

The first two columns of Table 3 contain the minimal and maximal number of generations needed to obtain the element with highest value of expected utility. We see that a hundred of generation was always enough - 99 was the maximal value. The median and mean of these numbers show that - in average - forty generations is sufficient to obtain satisfactory solution for considered CCLP problems.

Final remarks

The solution for CCLP(Λ, β, χ) problem found by the evolutionary search may be called satisfactory rather than optimal. The optimality cannot be proved and we even don't believe that it is optimal. But the solution is relatively easy to find and, in considered stochastic framework, much better than the optimal solution found in deterministic case. The improvement, measured in terms of average value of statistical performance rate *SPR* amounts to about 7 (for $n = 5$) or even to about 60 (for $n = 15$). The solution may be considered as satisfactory especially because of the high values of the indicator *SDR*: its average values are about 80%. Taking into account that the standard deviations of random variables disturbing all elements of the tuple (Λ, β, χ) amounts to 10% of its original values, one should not expect much more , even when applying more sophisticated solutions.

References

- [1] Charnes A., Cooper W.W., Chance-constrained programming, *Management Sciences* 1959, 6, 73-80.
- [2] Taflanidis T.T., Beckan J.L., Efficient framework for optimal robust stochastic system design using stochastic simulation, *Comput. Methods Appl. Mech. Engrg.* 2008, 198, 88-101.
- [3] Steuer R., Na P., Multiple criteria decision making combined with finance: A categorized bibliographic study, *European Journal of Operational Research* 2003, 150, 496-515.
- [4] Dima I.C., Grzybowski A., Modele optymalizacyjne w problemach budżetowania w warunkach niepewności i/lub ryzyka, [in:] *Budżetowanie*, I.C. Dima, J. Grabara (eds.), *Prace Naukowe WZ PCz, seria monografie Nr 19*, Częstochowa 2009, 153-166.
- [5] De P.K., Acharya D., Sahu K.C., A chance-constrained goal programming model for capital budgeting, *Journal of the Operational Research Society* 1982, 33(7), 635-638.
- [6] Norikin V.I., Pug G.C., Ruszczynski A., A Branch and Bound Method for Stochastic Global Optimization, *Mathematical Programming: Series A and B* 1998, 83, 3, 425-450.
- [7] Schuëller G.I., Jensen H.A., Computational methods in optimization considering uncertainties - An overview, *Comput. Methods Appl. Mech. Engrg.* 2008, 198, 2-13.
- [8] Sen S., *Stochastic Programming: Computational Issues and Challenges*, [in:] *Encyclopedia of Operations Research and Management Science*, 2nd edition, ed. S. Gass, C. Harris, Kluwer Academic Publishers, Boston 2001, 821-827.
- [9] Ruszczynski A., Shapiro A., *Stochastic Programming*, *Handbooks in Operations Research and Management Science*, Vol. 10. Elsevier, Amsterdam 2003.
- [10] Galar R., *Soft selection in random global adaptation in R^n : A biocybernetic model of development*, Technical University Press, Wrocław 1990 (in Polish).
- [11] Sen S., Higle J.L., *An Introductory Tutorial on Stochastic Linear Programming Models*, *Interfaces* 1999, 29, 33-61.
- [12] Obuchowicz A., Korbicz J., *Global optimization via evolutionary search with soft selection*, Manuscript, available: <http://www.mat.univie.ac.at/~neum/glopt/mss/gloptpapers.html>