Single Commodity Stochastic Network Design under Probabilistic Constraints with Continuously Distributed Random Variable.

András Prékopa
Olga Myndyuk
RUTCOR, Rutgers University

RUTCOR/DIMACS/CCICADA Workshop on Stochastic Networks: Reliability, Resiliency, and Optimization
October 12 - 14, 2011

Outline

- Definition of the network
- Examples
- Feasibility condition, Gale-Hoffman inequalities
- Prékopa-Boros elimination algorithm
- Network design problem formulation
- Mathematical properties, log-concavity
- Solution algorithms
- Approximation of the probabilistic constraint
- Numerical examples

Definition of a Network

Definition 1.

A network $G=(N,y)$ is a finite collection of nodes N and a capacity function $y(i,j)$ on the set of arcs (i,j) $N\times N$ which is assumed to take nonnegative values or ∞.

Definition 2.

A flow is a real-valued function $f(i,j)$ $(i,j)\in N\times N$ which satisfies the following conditions:

$f(i,j)+f(i,j)=0$

$f(i,j)\leq y(i,j),\ (i,j)\in N\times N$

$y(i,j)\geq 0$ or $y(i,j)=+\infty$

Feasibility Conditions

Definition 3

$A\ demand\ \ function\ d(i), i \in N$ is a real-valued function on the set of nodes.

Demand is said to be *feasible* if there exists a flow satisfying
Definition 2 and the relations $f(N,i) \geq d(i), i \in N$

Theorem 1
The *demand* $d(i), i \in N$ is feasible iff the following inequalities hold:
$d(S) \leq y(\overline{S}, S), S \subset N, where\ \overline{S} = N \setminus S$ (Gale-Hoffman inequalities)

Network Optimization Problem Formulation

Let $\boldsymbol{x_i}$ represent capacity assigned to node $\boldsymbol{i}$

Assume it is diminished by random deficiency $\xi_{\boldsymbol{i}}$ (normally distributed)

We define a demand function corresponding to the network as $\boldsymbol{d_i} = \xi_{\boldsymbol{i}} - \boldsymbol{x_i}$

Then in order to satisfy the demand with the required reliability level p the following probabilistic constraints need to be satisfied:

$\boldsymbol{P}(\boldsymbol{d}(\boldsymbol{S}) \leq \boldsymbol{y}(\overline{\boldsymbol{S}}, \boldsymbol{S}), \text{ all } \boldsymbol{S} \subset \boldsymbol{N}) \geq \boldsymbol{p}$

Network Optimization Problem Formulation (continued)

The unknown decision veriables are the node capacities $x_i, i \in N$

and the arc capacities $y_{ij} (i, j) \in N \times N$.

The static problem formulation is as follows:

$$\min\{\sum_{i \in N} c_i(x_i) + \sum_{(i,j) \in N \times N} c_{ij}(y_{ij})\}$$

subject to

$$P(d(S) \le y(S, \overline{S}), S \subset N) \ge p$$

$$A_1 x + A_2 y \ge b$$

$$x, y \ge 0$$

Stochastic Programming problem Formulation

$$\min \boldsymbol{c}^{\boldsymbol{T}} \boldsymbol{x}$$

subject to

$$\boldsymbol{P}(\boldsymbol{y}(\boldsymbol{S},\overline{\boldsymbol{S}}) \geq \boldsymbol{d}(\boldsymbol{S}), \boldsymbol{S} \subset \boldsymbol{N}_1, \boldsymbol{S} \text{ non-eliminated}) \geq \boldsymbol{p}$$

where $\boldsymbol{N}_1$ is a collection of the nodes with random demand,

$$\boldsymbol{y}(\boldsymbol{S},\overline{\boldsymbol{S}}) \geq \boldsymbol{d}(\boldsymbol{S}), \boldsymbol{S} \subset \boldsymbol{N}_2, \boldsymbol{N}_2 = \boldsymbol{N} \setminus \boldsymbol{N}_1$$

$$\boldsymbol{d}(\boldsymbol{S}) = \sum_{\boldsymbol{i} \in \boldsymbol{S}} (\xi_{\boldsymbol{i}} - \boldsymbol{x}_{\boldsymbol{i}}), \boldsymbol{l}_{\boldsymbol{i}} \leq \boldsymbol{x}_{\boldsymbol{i}} \leq \boldsymbol{u}_{\boldsymbol{i}}$$

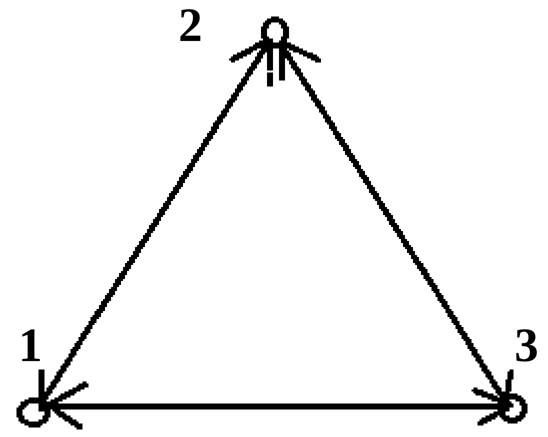
Illustrative Example

Consider the network of 3 nodes. Assume that the line capacities are symmetrical and introduce the notations:

$$y_1 = y_{12} = y_{21}, y_2 = y_{23} = y_{32}, y_3 = y_{13} = y_{31}$$

Then the system relations give the following system of inequalities:

$$\begin{aligned}
&\xi_1 - x_1 + \xi_2 - x_2 + \xi_3 - x_3 \le 0,\\
&\xi_1 - x_1 \le y_1 + y_3,\\
&\xi_2 - x_2 \le y_1 + y_2,\\
&\xi_3 - x_3 \le y_2 + y_3,\\
&\xi_1 - x_1 + \xi_2 - x_2 \le y_2 + y_3,\\
&\xi_2 - x_2 + \xi_3 - x_3 \le y_1 + y_3,\\
&\xi_1 - x_1 + \xi_3 - x_3 \le y_1 + y_2.
\end{aligned}$$



Prékopa-Boros Elimination (by topology)

Theorem 2

The inequality in the **Theorem 1** is redundant among the Gale-Hoffman inequalities iff at least one of the graphs $G(S), G(\overline{S})$ is not connected.
In that case the inequality $d(S) \leq y(\overline{S}, S)$ is the sum of other Gale-Hoffman inequalities.

Based on **Theorem 2**, we can eliminate those inequalities which are the sums of others.
We subsequently eliminate the sets $S \subset N$ according to their cardinalities, and look at sets $\overline{S}, S$ such that at least on of $G(S), G(\overline{S})$ is not connected.
Then eliminate the corresponding inequality.

Mathematical Properties of the Constraint Set. Log-concavity

The set $\{\mathbf{s} \mid F(\mathbf{s}) = p\}$ is an infinite set of pLEP of the continuously distributed random vector ξ and Tx is an element of a shaded region (see Figure 1, the case of r=2) iff $\{Tx \geq M\xi\} \geq p$

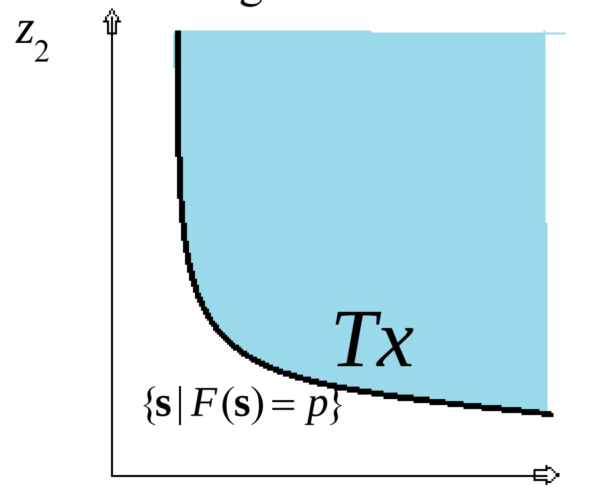


Figure 2

The convexity of the set is assured by the use of log-concavity theorem and the following theorem (Prékopa).

Theorem 3

Let $\xi \in {}^{\boldsymbol{n}}$ be a random vector that has log-concave probability distribution, i.e.

$\boldsymbol{P}(\lambda \boldsymbol{A} + (1-\lambda)\boldsymbol{B}) \geq \boldsymbol{P}^{\lambda}(\boldsymbol{A})\boldsymbol{P}^{1-\lambda}$ where $\boldsymbol{A}, \boldsymbol{B} \subset {}^{\boldsymbol{n}}$ are convex ses and $0 < \lambda < 1$.

Let $\boldsymbol{M}$ be an $\boldsymbol{r} \times \boldsymbol{n}$ matrix with real-valued entries. Then the vector $\zeta = \boldsymbol{M}\xi$ has logconcave distribution.

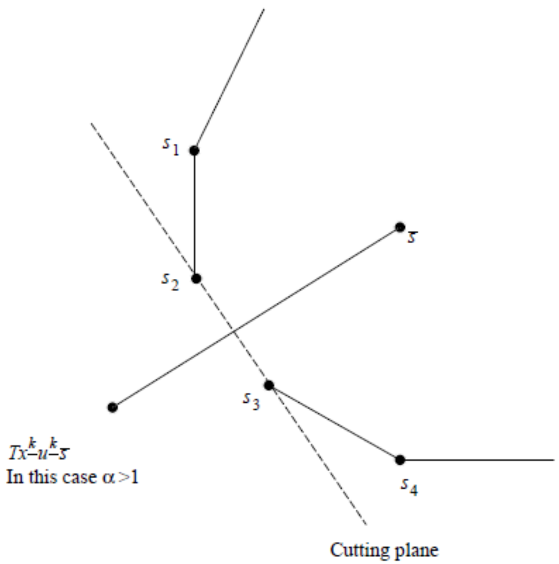
Multivariate Quantiles (multivariate Value at Risk)

Let $X=(X_1,X_2,\ldots,X_r)^T$ be a continuously distributed random variable.
It's p-quantile is defined as $F(z)=P(X\le z), z\in {}^{r}, 0\le p\le 1$
where $Q_p=\{z\,|\,F(z)=p\}$

If X has a non-degenerate normal distribution the Prékopa's log-concavity theory ensures that the set $\{z\,|\,F(z)\ge p\}$ is a convex set.

In case of degenerate normal distribution (partial sums of random variables in the constraints) convexity is assured using **Theorem 3.**

The PVB (Prékopa, Vizvári, Badics) Algorithm



PVB (Prékopa, Vizvári, Badics) algorithm

Step 0. Enumerate all p-efficient points $s_1, \ldots, s_N$. Initialize $k \leftarrow 0$ and go to Step 1.

Step 1. Determine the vectors $w_1, \ldots, w_h$.

Step 2. Solve the LP:

$$\begin{aligned}
&\min c^T x \\
&\text{subject to} \\
&\qquad Ax \geq b \\
&w_l^T(Tx - u - \overline{s}) = 0, \quad l = 1, \ldots, h \\
&v_i^T(Tx - u - \overline{s}) \geq 0, \quad i = 1, \ldots, k \\
&\qquad x \geq 0, \quad u \geq 0.
\end{aligned}$$

If $k = 0$, then ignore the constraint for $i = 1, \ldots, k$. Let (x^k, u^k) be an optimal solution. Go to Step 3.

Step 3. For $t = Tx^k - u^k - \overline{s}$ solve problem (28) and let α be the optimum value. If $\alpha \leq 1$, Stop, (x^k, u^k) is an optimal solution to problem (26). Otherwise go to Step 4.

Step 4. Create the cut

$$v_{k+1}^T(Tx - u - \overline{s}) \geq 0,$$

set $k \leftarrow k + 1$ and go to Step 2.

The Supporting Hyperplane Algorithm

The supporting hyperplane method, developed originally by Veinott (1967) was adapted by Szantai (1988) to solve probabilistic optimization problem with continuously distributed random vector with logconcave p.d.f.

Let $\boldsymbol{h}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{Tx} \geq \boldsymbol{M}\xi) - \boldsymbol{p}$. Suppose also that the convex polyhedron

$\boldsymbol{K}^0 = \{\boldsymbol{x} \mid \boldsymbol{Ax} \geq \boldsymbol{b}, \boldsymbol{x} \geq 0\}$ is bounded and that Slater's condition is satisfied:

$\exists \boldsymbol{x}^0 \in \boldsymbol{K}^0$ such that $\boldsymbol{h}(\boldsymbol{x}^0) > 0$.

The supporting hyperplane can be summarized as follows:

Step 0. Find x^0 satisfying $Ax^0 \geq b$, $x^0 \geq 0$, $h(x^0) > 0$. Go to Step 1.

Step 1. Solve the LP:

$$\begin{aligned} &\min c^T x \\ &\text{subject to} \\ &\qquad Ax \geq b \\ &\nabla h(x^i)(x - x^0) \geq 0, \quad i = 1, \ldots, k \\ &\qquad x \geq 0. \end{aligned}$$

Let x^{*k} be an optimal solution. Go to Step 2.

Step 2. Check for the sign of $h(x^{*k})$. If $h(x^{*k}) \geq 0$, Stop, optimal solution to problem has been found. Otherwise go to Step 3.

Step 3. Find λ^k such that $0 < \lambda^k < 1$ and

$$h(x^0 + \lambda^k(x^{*k} - x^0)) = 0.$$

Define $x^{k+1} = x^0 + \lambda^k(x^{*k} - x^0)$ and go to step 4.

Step 4. Introduce the cut:

$$\nabla h(x^{k+1})(x - x^0) \geq 0,$$

set $k \leftarrow k + 1$ and go to Step 1.

Combination of the PVB (Prékopa, Vizvári, Badics) and the supporting hyperplane algorithms (hybrid algorithm)

The combined application of PVB and supporting hyperplane algorithms provides solution method for problem , where r.v. has continuous distribution. At each iteration both lower and upper bounds for the optimum are available, and if they are sufficiently close we may stop.

Step 0. Find satisfying Slater's condition. Initialize $k \leftarrow 0$;

Step 1. Choose $x^i \ldots x^r$ linearly independent feasible vectors such that : $F(Tx^i) = p, i - 1, \ldots, r$

Set $k \leftarrow r$

Step 2. Solve the LP:

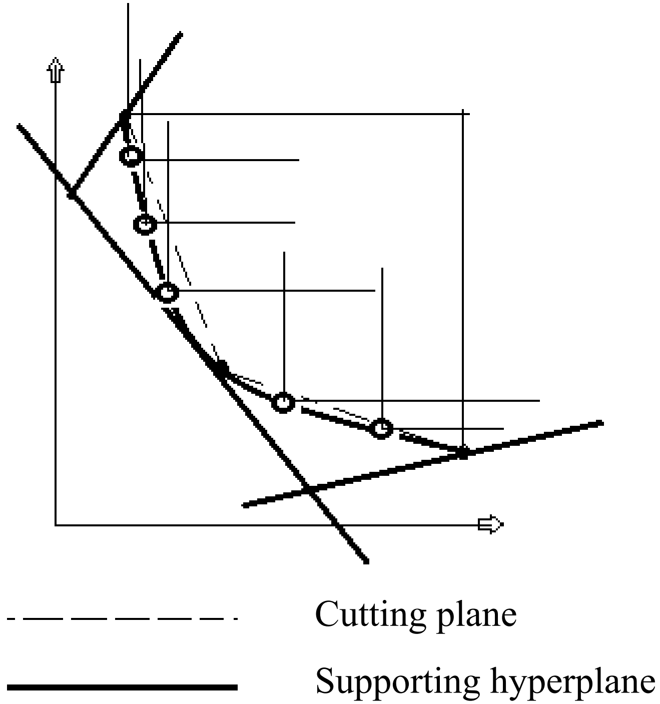
$$\min c^T x$$

subject to **(5)**

$$Ax \geq b$$

$$\nabla h(x^i)(x - x^0) \geq 0, i = 1, \ldots, k$$

$$x \geq 0$$



Combination of the PVB (Prékopa, Vizvári, Badics) and the Supporting Hyperplane Algorithms (hybrid algorithm) (continued)

Let x^* be an optimal solution. If $F(Tx^{*k}) \geq p$ then Stop, we found an optimal solution. Otherwise let $x^{k+1} = x^0 + \lambda^k (x^{*k} - x^0)$ where λ^k is such that $F(T(x^0 + \lambda^k (x^{*k} - x^0))) = p$. Set $k \leftarrow k+1$ and $U_k = c^T x^{k+1}$

Obviously, U_k is an upper bound on the optimum value

Step 3. Solve the following LP by PVB algorithm:

$$\min c^T x$$
subject to
$$Ax \geq b$$
$$Tx - u - \bar{s} = \sum_{i=1}^{k} (s_i - \bar{s}) \qquad \textbf{(5)}$$
$$\sum_{i=1}^{k} \lambda_i = 1$$
$$x \geq 0, u \geq 0, \lambda \geq 0,$$

where $s_i = Tx^i, i = 1, \ldots, k; \; s = (1/k) \sum_{j=1}^{k} s_j$

Vectors $s_1, \ldots, s_k$ span the whole space ^{r}

Let L_k be the optimum value of problem **(5)**

Then it gives us the lower bound for the optimum value of problem **(1)**

Combination of the PVB (Prékopa, Vizvári, Badics) and the Supporting Hyperplane Algorithms (hybrid algorithm) (continued)

Step 4. If $U_k - L_k \le \varepsilon$ (tolerance level) then STOP, otherwise go to **Step 5**

Step 5. Having solved problem (5)the number of the vectors appended to the set $\{s^{(1)},\dots,s^{(k)}\}$is the same as the number of cutting planes encountered, let it be *l*. They belong to the boundary of prob. constraint set. Then we identify those bases that turned optimal auxiliary LP in the course of cutting plane method , take the intersections of the normals of the corresponding cutting planes with the p=efficient surface (quantile surface). Set $k \leftarrow k+l$ and go to **Step 2.**

Approximation of the Boundary Surface of the Probabilistic Constraints with Normally Distributed Random Variable

One of the most important problems encountered during the implementation of the hybrid algorithm for the case of normally distributed ξ is approximation of the boundary of probabilistic constraint. The choice of approximating function was a hyperboloid-like surface.

Consider the function defining the prob. constraint:

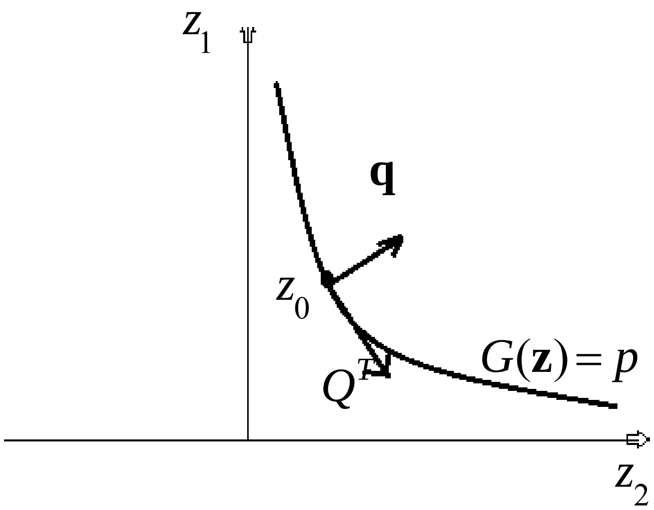
$$G(x) = P(g_1(x,\xi) \geq 0, \ldots, g_r(x,\xi) \geq 0) = P(Tx \geq M\xi)$$

We will construct the approximation of the p-efficient set of the normal distribution in the form of a set of inequalities:

$$q^T(z - z_0) \geq \sqrt{(z - z_0)^T Q^T AQ(z - z_0) + k^2} - k$$

where Q is an orthogonal matrix of change of coordinates.

Construction of the Approximating Function



$$\mathbf{z} \in R^m, \mathbf{x} \in R^{m-1}$$

$$G(\mathbf{x}) = G(Q\mathbf{z}) = \langle \mathbf{q}, \mathbf{z} \rangle$$

After finding the gradient and Hessian of the function G we can evaluate all the important coefficients in the approximation function.

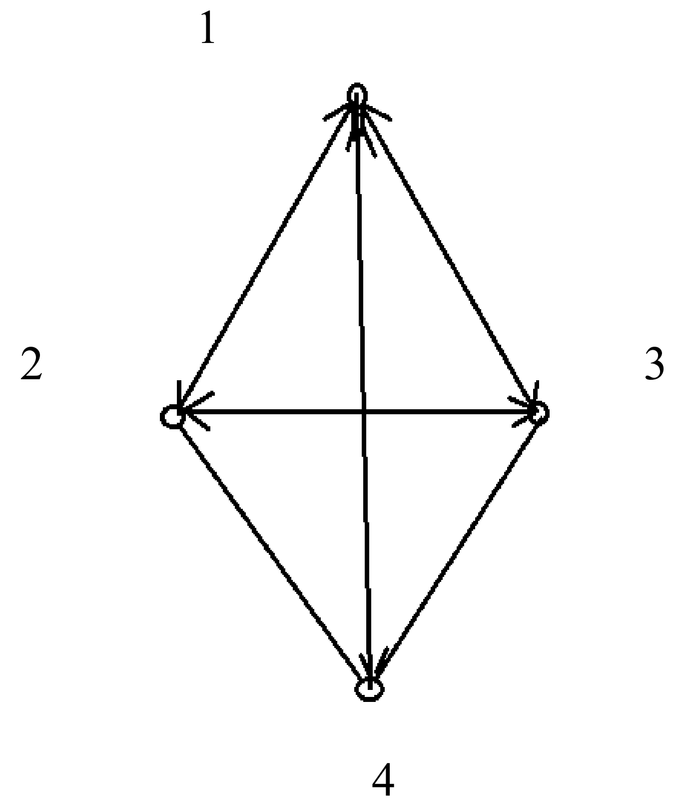
They are as follows:

$A = \frac{(\min \mu_i)^4}{4M^2}\mathbf{I}$, where $\min \mu_i$ is the smallest eigenvalue of $\nabla^2 G(0)$;

$k = \frac{5}{24}\frac{(\min \mu_i)^3}{M^2}$; M is an approximation of the derivative of the curvature of G

Network optimization problem (simple numerical example)

We tested the hybrid algorithm on various networks, here is the example of a power network between 4 cities



The input parameters are as follows:

Demand μ : [2 3 4 5]

Demand σ : [1.4142 1.7321 2 2.2361]

Cost of production: [2 2 2 2]

Cost of transmission capacities: [1 1 1 1 1 1]

p: 0.9

Optimal solution found:

Optimal production amount:

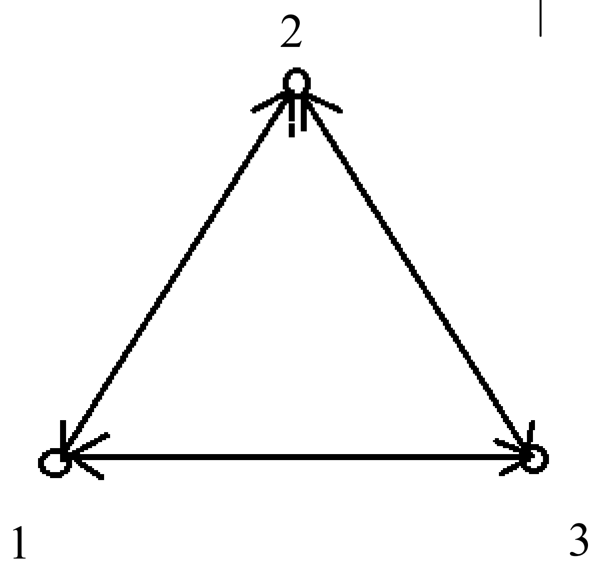
3.0958 4.1878 5.5071 6.6923

Optimal line capacities:

0.7406 0.9512 0.9003 1.0110 1.0708 1.0722

Network optimization (numerical example 2)

demand μ : [2 3 4]
demand σ : [1 1 1]
cost_production: [2 2 2]
cost_capacity: [1 1 1]
p: 0.95



Optimal solution found:

Optimal production amount:
3.0337 4.0335 5.0335

Optimal line capacities:
0.7301 0.7300 0.7300

Future work

- Consider other distributions
- Work on online algorithm
- Improve on timing of the p-quantile approximations in higher dimensions (≥ 50)

References:

- Prékopa, A. (1995). *Stochastic Programming*. Kluwer Academic Publishers, Dordrecht.
- Prékopa, A. (1970). *On Probabilistic Constrained Programming*. Proceedings of the Princeton Symposium on Math. Prog. Princeton University Press, Princeton, NJ
- Prékopa, A. (1973a). *On Logarithmic Concave Measures and Functions*. Acta Sci.Math. (Szeged) 34, 335–343
- Prékopa, A., B. Vizvári and T. Badics (1998). *Programming under Probabilistic Constraint with Discrete Random Variable*. In: New Trends in Mathematical Programming (F. Giannessi et al. eds.), Kluwer Academic Publishers, Dordrecht, 235–255
- Szántai, T. (1988). *A Computer Code for Solution of Probabilistic Constrained Stochastic Programming Problems*. In: Numerical Techniques for Stochastic Optimization (Y. Ermoliev, and R. J.-B. Wets, eds), Springer-Verlag, New York, 229–235
- Veinott, A.F. (1967). *The Supporting Hyperplane Method for Unimodal Programming*.Operations Research 15, 147–152.