Multi-Stage Stochastic Programming with CVaR: Modeling, Algorithms and Robustness

Václav Kozmík

Faculty of Mathematics and Physics
Charles University in Prague

December 11, 2014
Outline

- Introduction
  - Outline
  - Original results
- Structure of risk-averse multistage stochastic models
  - Studied less than the risk-neutral case
  - Risk measures bring multiple modeling possibilities
    - Nested models
    - Multiperiod risk measures
    - Sum of stage risks
- Stochastic Dual Dynamic Programming Algorithm
  - Originated in 1991 by Pereira & Pinto
  - Well-studied for risk-neutral case
  - Cut generation in risk-averse case is straightforward
  - Upper bound estimation is challenging in risk-averse case
Outline

- Variance reductions schemes
  - Monte Carlo methods are widely applied, but they usually bring high variance
  - Many variance reduction schemes have been proposed: importance sampling, stratified sampling, Quasi Monte Carlo, etc.
  - Application of such schemes could be hard in practice, the proposals usually focus on the method and toy examples, not on large-scale real world applications

- Contamination technique for multistage problems
  - Captures various changes in the probability distribution
  - First results applied to two-stage setups
  - In multistage case, contamination was studied and applied for smaller problems
  - Extension to large-scale setups requires advanced algorithms and techniques
Original results

- New estimator for the policy value under the nested model setup with CVaR
  - Provides better results than the state-of-the-art estimators
  - Can be used to build valid stopping rules for the SDDP algorithm
  - General importance sampling scheme for mean-CVaR objectives
    - Closed-form solution provided for normal distribution
    - Sampling algorithm presented for any general distribution to get the suitable parameter of the variance reduction scheme

- Contamination technique for large-scale multistage programs
  - We provide an extension which accounts for the fact, that we can never compute precise optimal solution
  - Based on lower bounds from cutting-plane algorithms and upper bounds from policy value estimators

- Numerical study and comparison of two multi-stage models based on CVaR
Multistage stochastic optimization

- Consider $T$ stage stochastic program:
  - Data process $\xi = (\xi_1, \xi_2, \ldots, \xi_T)$
  - Decision process $x = (x_1, \ldots, x_T)$
  - Stages should reflect the timing of decisions, steps can be unequal
  - Filtration $\mathcal{F}_t$ generated by the projection $\Pi_t \xi = \xi_{[t]} := (\xi_1, \ldots, \xi_t)$
  - Probability distribution of $\xi$: $\mathbb{P}$
  - $\mathbb{P}_t$ denotes the marginal probability distribution of $\xi_t$
  - $\mathbb{P}_t \left[ \cdot | \xi_{[t-1]} \right]$ denotes the conditional probability distribution

- The decision process is nonanticipative:
  - Decisions taken at any stage of the process do neither depend on future realizations of stochastic data nor on future decisions
  - $x_t$ is $\mathcal{F}_t$-measurable
  - The sequence of decisions and observations is:
    $$x_1, \xi_2, x_2(x_1, \xi_2), \ldots, x_T(x_{T-1}, \xi_2, \ldots, \xi_T)$$
  - its random outcome $f(x, \xi)$
Scenario tree
Multistage stochastic optimization

- Nested form of multistage stochastic linear program (MSLP):

\[
\min_{x_1 \in \mathcal{X}_1} c_1^T x_1 + \mathbb{E}_P [Q_2(x_1, \xi_2)] \quad \text{with} \quad \mathcal{X}_1 := \{x_1 | A_1 x_1 = b_1, x_1 \geq 0\}
\]

- With \( Q_t(x_{t-1}, \xi_{[t]}) \), \( t = 2, \ldots, T \), defined recursively as

\[
Q_t(x_{t-1}, \xi_{[t]}) = \min_{x_t} c_t(\xi_{[t]})^T x_t + \mathbb{E}_{P_{t+1}[\cdot | \xi_{[t]}]} \left[ Q_{t+1}(x_t, \xi_{[t+1]}) \right]
\]

- \( \mathcal{X}_t(x_{t-1}, \xi_{[t]}) \): \( A_t(\xi_{[t-1]}) x_t = b_t(\xi_{[t-1]}) - B_t(\xi_{[t-1]}) x_{t-1}, x_t \geq 0 \ a.s. \),

- In the case of stagewise independence the conditional distributions boil down to marginal distributions \( \mathbb{P}_t \) of \( \xi_t \)

- We assume:
  - Constraints involving random elements hold almost surely
  - All optimal solutions exist, which is related with the relatively complete recourse
  - All conditional expectations exist
Risk-averse multistage programs

- In the risk-neutral programs possible risks are not reflected
- Risk measure is a functional which assigns a real value to the random outcome \( f(x, \xi) \)
- Risk measures depend on decisions and probability distribution \( \mathbb{P} \).
  - Filtration \( \mathcal{F}_1 \subset \cdots \subset \mathcal{F}_t \cdots \subset \mathcal{F} \) should be taken into account
- Risk monitoring in individual stages should be incorporated
  \[
  \min_{x_1} c_1^\top x_1 + \rho_2 \left( \min_{x_2} c_2(\xi_{[1]})^\top x_2 + \cdots + \rho_T \left( \min_{x_T} c_T(\xi_{[T-1]})^\top x_T \right) \right)
  \]
  - Different risk measures \( \rho_t \) can be applied in each stage
- **Coherence** of \( \rho \) is mostly expected [Artzner et al., 2007]
Time consistency

- Many different definitions
- Need to distinguish between time consistency of the risk measure and time consistency of the model
- **TC1** [Carpentier, et al., 2012] The sequence of dynamic optimization problems is dynamically consistent if the optimal strategies obtained when solving the original problem remain optimal for all subsequent problems.
- **TC2** [Shapiro, 2009] At each state of the system, optimality of a decision policy should not involve states which cannot happen in the future.
- Risk-neutral stochastic programs are time consistent
- In general, time consistency for risk-averse stochastic programs does not hold true
Nested CVaR risk measure

- Consider sequence of random costs $\mathbf{Z} = (Z_1, \ldots, Z_T)$
- Nested CVaR risk measure is given by:

$$
\rho^n (\mathbf{Z}) = \text{CVaR}_\alpha \left[ \cdot | \mathcal{F}_1 \right] \circ \cdots \circ \text{CVaR}_\alpha \left[ \cdot | \mathcal{F}_{T-1} \right] \left( \sum_{t=1}^{T} Z_t \right)
$$

- The interpretation is not straightforward
  - can be viewed as the cost we would be willing to pay at the first stage instead of incurring the sequence of random costs $Z_1, \ldots, Z_T$
  - cf. Ruszczyński [2010]
Nested CVaR model

- Risk measures are usually combined with expectation to get efficient solutions
- Given risk coefficients $\lambda_t$ and random loss variable $Z$ we define:

$$\rho_{t,\xi_{[t-1]}} [Z] = (1 - \lambda_t) \mathbb{E} \left[ Z \mid \xi_{[t-1]} \right] + \lambda_t \text{CVaR}_{\alpha_{t}} \left[ Z \mid \xi_{[t-1]} \right]$$

- Nested model can be written:

$$\begin{align*}
\min_{A_1 x_1 = b_1, x_1 \geq 0} c_1^\top x_1 + \rho_{2,\xi_{[1]}} & \left[ \min_{B_2 x_1 + A_2 x_2 = b_2, x_2 \geq 0} c_2^\top x_2 + \cdots \right] \\
\cdots + \rho_{T,\xi_{[T-1]}} & \left[ \min_{B_T x_{T-1} + A_T x_T = b_T, x_T \geq 0} c_T^\top x_T \right]
\end{align*}$$

- Time consistent w.r.t. [TC1] and [TC2]
Nested CVaR model

- Allows to develop dynamic programming equations, using:
  \[
  \text{CVaR}_\alpha [Z] = \min_u \left[ u + \frac{1}{\alpha} \mathbb{E} [Z - u]_+ \right]
  \]

- Denote \( Q_t(x_{t-1}, \xi[t]) \), \( t = 2, \ldots, T \) as the optimal value of:
  \[
  Q_t(x_{t-1}, \xi[t]) = \min_{x_t, u_t} c_t^\top x_t + \lambda_{t+1} u_t + Q_{t+1}(x_t, u_t, \xi[t])
  \]
  \[
  \text{s.t. } A_t x_t = b_t - B_t x_{t-1}
  \]
  \[
  x_t \geq 0,
  \]

- Recourse function \( Q_{t+1}(x_t, u_t, \xi[t]) \) is given by \( (Q_{T+1}(\cdot) \equiv 0) \):
  \[
  \mathbb{E}_{\mathbb{P}_{t+1}[\cdot | \xi[t]]} \left[ (1 - \lambda_{t+1}) Q_{t+1}(x_t, \xi[t+1]) + \frac{\lambda_{t+1}}{\alpha_{t+1}} \left[ Q_{t+1}(x_t, \xi[t+1]) - u_t \right]_+ \right]
  \]
Multiperiod CVaR risk measure

- Based on the following risk measure:

\[ \rho^m(Z) = \sum_{t=2}^{T} \mu_t \mathbb{E} [\text{CVaR}_{\alpha_t} [Z_t | F_{t-1}]]. \]

- The difference between this risk measure and the nested CVaR risk measure is that here we apply expectation instead of the nesting.

- Easier interpretation
  - Averaging of the risks in future stages

- Polyhedral risk measure
  - Solution of a multi-stage stochastic linear program of a special form
  - Optimization of the original problem can be combined with the optimization problem which defines the risk measure.
Multiperiod CVaR model

- Stochastic programming model:

\[
\min_{x_1, \ldots, x_T} \mathbf{c}_1^T x_1 + \mu_2 \mathbb{E} \left[ \rho_{2,\xi[1]} \left[ \mathbf{c}_2^T x_2 \right] \right] + \cdots + \mu_T \mathbb{E} \left[ \rho_{T,\xi[T-1]} \left[ \mathbf{c}_T^T x_T \right] \right]
\]

s.t. \( A_1 x_1 = b_1 \)

\( A_2 x_2 = b_2 - B_2 x_1 \)

\[ \vdots \]

\( A_T x_T = b_T - B_T x_{T-1} \)

\( x_t \geq 0, \ x_t \in \mathcal{L}_p (\Omega, \mathcal{F}_t, \mathbb{P}) \quad t = 1, \ldots, T. \)

- Time consistent w.r.t. [TC1] and [TC2]
- Dynamic programming equations are developed, similarly to the nested case
Sum of CVaR model

- The weighted sum of CVaR model is based on the following risk measure:

\[ \rho^s(Z) = \sum_{t=2}^{T} \mu_t \text{CVaR}_{\alpha_t} [Z_t] \]

with \( \sum_{t=2}^{T} \mu_t = 1, \mu_t \geq 0 \forall t \).

- No nesting or averaging is present
- Easy interpretation - weighted sum of CVaR for all stages
- Related to multi-criteria optimization
  - “We want to hedge against risk in all stages separately”
- Dynamic programming equations show that all \( u_t \) are decided in the first stage
- Corresponding model is time consistent w.r.t. [TC1], but not w.r.t. [TC2]
Asset allocation model

- At stage \( t \) we observe the price ratio between the new price and the old price \( r_t \)
- \( x_t \) contains the optimal allocation (in USD, say)
- The total portfolio value is tracked as a multiple of the initial value
- Dynamic programming equations are very simple:

\[
\min_{x_t, u_t} - 1^T x_t + \lambda_{t+1} u_t + Q_{t+1}(x_t, u_t) \\
\text{s.t. } r_t^T x_{t-1} - 1^T x_t = 0 \\
\quad x_t \geq 0
\]

- Transaction costs of \( f_t 1^T |x_t - x_{t-1}| \) can be included
- We solve problems up to 15 stages with \( 10^{24} \) scenarios, using SDDP with importance sampling
SDDP algorithm

- Starts with SAA of the problem - scenario tree, given or sampled
- Forward iteration
  - Samples $\xi^1, \ldots, \xi^J$ sample paths
  - Policy is evaluated using all the cuts collected so far
  - Value of the policy gives the upper bound
- Backward iteration
  - Subset of the scenarios from the forward iteration is chosen
  - For every chosen node the Benders’ cut is calculated
    - Using all of its immediate descendants
  - Optimal value of the root problem gives the lower bound
- The bounds are compared and the process is repeated
- Relies on the stage-independence assumption
  - Cuts are valid for all nodes from given stage
  - Low memory requirements to store scenarios
  - Linear complexity w.r.t. number of stages
- CPLEX and COIN-OR used as solvers for the LPs
Upper bound overview

- **Risk-neutral problems**
  - The value of the current optimal policy can be estimated easily
  - Expectation at each node can be estimated by single chosen descendant

- **Risk-averse problems**
  - To estimate the CVaR value we need more descendants in practice
  - Leads to intractable estimators with exponential computational complexity, denoted by $U^e$

- **Current solution (to our knowledge)**
  - Run the risk-neutral version of the same problem and determine the number of iterations needed to stop the algorithm, then run the same number of iterations on the risk-averse problem
  - Inner approximation scheme proposed by Philpott et al. [2013]
    - Works with different policy than the outer approximation
    - Probably the best alternative so far
    - Does not scale well with the dimension of $x$
Upper bound enhancements

- State-of-the-art estimator runs with exponential complexity
- We need an estimator with linear complexity to build valid bounds
- Ideally it should be unbiased, or in practice, have small bias
- We start with the linear estimator from the risk-neutral case and include:
  - Importance sampling, with an additional assumption needed
  - Further enhancements to reduce bias and volatility

Assumption

Let \( a_t(x_{t-1}, \xi_t) \) approximate the recourse value of our decisions \( x_{t-1} \) after the random parameters \( \xi_t \) have been observed, and let \( a_t(x_{t-1}, \xi_t) \) be cheap to evaluate.

For example in our portfolio model:

\[
a_t(x_{t-1}, \xi_t) = -\xi_t^T x_{t-1} = -r_t^T x_{t-1}
\]
Importance sampling

- We start with standard pmf, all probabilities equal for $D_t$ scenarios:

$$g_t(\xi_t) = \frac{1}{D_t} \mathbb{I}\left[\xi_t \in \{\xi^1_t, \ldots, \xi^{D_t}_t\}\right]$$

- This is not required by SDDP and can be easily relaxed

- Denote $u_a = \text{VaR}_\alpha \left[a_t(\mathbf{x}_{t-1}, \xi_t)\right]$ 

- We change the measure to put more weight to the CVaR nodes:

$$h_t(\xi_t|\mathbf{x}_{t-1}) = \begin{cases} 
\frac{\beta_t}{\alpha_t} g_t \mathbb{I}\left[\xi_t \in \{\xi^1_t, \ldots, \xi^{D_t}_t\}\right], & \text{if } a_t(\mathbf{x}_{t-1}, \xi_t) \geq u_a \\
\frac{1 - \beta_t}{1 - \alpha_t} g_t \mathbb{I}\left[\xi_t \in \{\xi^1_t, \ldots, \xi^{D_t}_t\}\right], & \text{if } a_t(\mathbf{x}_{t-1}, \xi_t) < u_a,
\end{cases}$$

- We select forward nodes according to this measure

- $\mathbb{E}_{g_t} [Z] = \mathbb{E}_{h_t} \left[Z \frac{g_t}{h_t}\right]$

- We start with $\beta_t = \frac{1}{2}$ and optimize its value later
Linear estimators

- The nodes can be selected randomly from the standard i.i.d. measure or from the importance sampling measure.
- For stages $t = 2, \ldots, T$ is given by:
  \[
  \hat{v}_t(\xi_{t-1}^{j}) = (1 - \lambda_t) \left( (c_t^j)^\top x_t^j + \hat{v}_{t+1}(\xi_t^j) \right) + \\
  + \lambda_t u_{t-1}^{j-1} + \frac{\lambda_t}{\alpha_t} \left[ (c_t^j)^\top x_t^j + \hat{v}_{t+1}(\xi_t^j) - u_{t-1}^{j-1} \right] +
  \]
- $\hat{v}_{T+1}(\xi_T^j) \equiv 0$
- Along a single path for scenario $j$ the cost is estimated by:
  \[
  \hat{v}(\xi^j) = c_1^\top x_1 + \hat{v}_2
  \]
Linear estimators

- For scenarios selected via the original pmf we have the naive estimator
  
  \[ U^n = \frac{1}{M} \sum_{j=1}^{M} \hat{v}(\xi^j) \]

- With weights again defined via
  
  \[ w(\xi^j) = \prod_{t=2}^{T} \frac{g_t(\xi_t)}{h_t(\xi_t | x_{t-1})} \]

- For scenarios selected via the IS pmf we have the IS estimator
  
  \[ U^i = \frac{1}{\sum_{j=1}^{M} w(\xi^j)} \sum_{j=1}^{M} w(\xi^j) \hat{v}(\xi^j) \]
Linear estimators - validity

Function $U^i$ provides an asymptotic upper bound estimator for the SAA version of the presented optimization problem.

**Proposition**

Assume finite optimal value, relatively complete recourse and interstage independence. Let $\varphi$ denote the optimal value. Let $\xi$ denote a sample path selected under the empirical distribution, and let $\hat{v}(\xi)$ be defined for that sample path. Then $E_g[\hat{v}(\xi)] \geq \varphi$. Furthermore if $\xi^j, j = 1, \ldots, M$, are i.i.d. and generated by the IS pmfs then $U^i \rightarrow E_g[\hat{v}(\xi)]$, w.p.1, as $M \rightarrow \infty$. 
Upper bound enhancements

- Linear estimator still degrades for problems with 10 or 15 stages
- The reason for the bias of the estimator comes from poor estimates of CVaR
  - Once the cost estimate for stage $t$ exceeds $u_{t-1}$ the difference is multiplied by $\alpha_{t-1}$
  - When estimating stage $t-1$ costs in the nested model we sum stage $t-1$ costs and stage $t$ estimate which means that we usually end up with costs greater than $u_{t-2}$ so another multiplication occurs
  - This brings both bias and large variance

**Assumption**

For every stage $t = 2, \ldots, T$ and decision $x_{t-1}$ the approximation function $a_t$ satisfies:

$$Q_t \geq \text{VaR}_{\alpha_t}[Q_t] \text{ if and only if } a_t \geq \text{VaR}_{\alpha_t}[a_t].$$
Improved estimator

- Provided that the equivalence assumption holds we can reduce the bias of the estimator
  - The positive part operator in the equation is used only in the case of CVaR node

- For stages $t = 2, \ldots, T$ we have

$$
\hat{v}^a_t(\xi_{t-1}^j) = (1 - \lambda_t) \left( (c^j_t)^\top x^j_t + \hat{v}^a_{t+1}(\xi^j_t) \right) + \lambda_t u_{t-1}^j + \\
+ \mathbb{I}[a_t > \text{VaR}_{\alpha_t}[a_t]] \frac{\lambda_t}{\alpha_t} \left[ (c^j_t)^\top x^j_t + \hat{v}^a_{t+1}(\xi^j_t) - u_{t-1}^j \right] +
$$

- $\hat{v}^a_{T+1}(\xi_T^j) \equiv 0$

- $U^a = \frac{1}{\sum_{j=1}^M w(\xi^j)} \sum_{j=1}^M w(\xi^j) \hat{v}^a(\xi^j)$
Improved estimator - validity

Function $U^a$ provides an asymptotic upper bound estimator for the SAA version of the presented optimization problem.

Proposition

Assume finite optimal value, relatively complete recourse and interstage independence. Let $\varphi$ denote the optimal value Let $\xi$ denote a sample path selected under the empirical distribution and let the “perfect ordering” assumption hold. Then $E_g [\hat{v}^a(\xi)] \geq \varphi$. If $\xi^j$, $j = 1, \ldots, M$, are i.i.d. and generated by the IS pmfs then $U^a \rightarrow E_g [\hat{v}^a(\xi)]$, w.p.1, as $M \rightarrow \infty$. Furthermore, if the subproblems induce the same policy for both $\hat{v}(\xi)$ and $\hat{v}^a(\xi)$ then $E_g [\hat{v}(\xi)] \geq E_g [\hat{v}^a(\xi)]$. 
Improved estimator results

- Comparison with exponential estimator $U^e$ from the literature:

<table>
<thead>
<tr>
<th>$T$</th>
<th>$z$</th>
<th>$U^n$ (s.d.)</th>
<th>$U^i$ (s.d.)</th>
<th>$U^a$ (s.d.)</th>
<th>$U^e$ (s.d.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>-0.9518</td>
<td>-0.9515 (0.0020)</td>
<td>-0.9517 (0.0012)</td>
<td>-0.9517 (0.0011)</td>
<td>-0.9518 (0.0019)</td>
</tr>
<tr>
<td>3</td>
<td>-1.8674</td>
<td>-1.8300 (0.0145)</td>
<td>-1.8285 (0.0108)</td>
<td>-1.8656 (0.0060)</td>
<td>-1.8013 (0.0302)</td>
</tr>
<tr>
<td>4</td>
<td>-2.7811</td>
<td>-2.4041 (0.1472)</td>
<td>-2.3931 (0.1128)</td>
<td>-2.7764 (0.0126)</td>
<td>-2.6027 (0.0883)</td>
</tr>
<tr>
<td>5</td>
<td>-3.6794</td>
<td>-3.4608 (0.1031)</td>
<td>-3.4963 (0.1008)</td>
<td>-3.6731 (0.0303)</td>
<td>-2.9031 (0.5207)</td>
</tr>
<tr>
<td>10</td>
<td>-7.6394</td>
<td>$9.3 \times 10^4$ (1.4 \times 10^4)</td>
<td>$9.0 \times 10^4$ (8.7 \times 10^4)</td>
<td>-7.5465 (0.2562)</td>
<td>$1.5 \times 10^7$ (1.3 \times 10^6)</td>
</tr>
<tr>
<td>15</td>
<td>-11.5188</td>
<td>NA</td>
<td>NA</td>
<td>-11.0148 (0.6658)</td>
<td>NA</td>
</tr>
</tbody>
</table>

- For $T = 2, \ldots, 5$ variance reduction of $U^a$ relative to $U^e$: 3 to 25 to 50 to 300.
- Computation time for $U^n$ for $T = 5, 10, 15$: 8.7 sec. to 31.6 sec. to 67.4 sec.
- Computation time for $U^a$ for $T = 5, 10, 15$: 6.8 sec. to 30.0 sec. to 66.5 sec.
- Can be extended to handle more complex models, for instance asset allocation with transaction costs.
Variance reduction

- We consider a functional from our model:

\[ Q_{\alpha} [Z] = (1 - \lambda) \mathbb{E} [Z] + \lambda \text{CVaR}_{\alpha} [Z] \]

- We define:

\[
Q^{s} = (1 - \lambda) Z + \lambda \left( u_{Z} + \frac{1}{\alpha} [Z - u_{Z}]_{+} \right) \\
Q^{i} = \frac{g}{h} \left( (1 - \lambda) Z + \lambda \left( u_{Z} + \frac{1}{\alpha} [Z - u_{Z}]_{+} \right) \right)
\]

- It clearly holds \( Q = \mathbb{E}_{h} [Q^{i}] = \mathbb{E}_{g} [Q^{s}] \)

- Our aim is to find suitable parameter \( \beta \) for our importance sampling scheme, so that \( \text{var}_{h} [Q^{i}] < \text{var}_{g} [Q^{s}] \)
Example - normal distribution

![Diagram of normal distributions with different parameters and lambda values.]

- **N(0,1)**
- **N(1,1)**
- **N(1,0.5)**
- **N(-1,1)**

**Axes:**
- **X-axis:** lambda
- **Y-axis:** beta
Other distributions

- We can also estimate the suitable $\beta$ by sampling
  - We choose a mesh of possible values, e.g. $B = \{0.01, 0.02, \ldots, 0.99\}$
  - For each of them, we sample prescribed number of scenarios, $Z^j$
  - We compute the mean and variance of the values $Q^j$ given by $Z^j$
  - The lowest variance is selected as a suitable choice of $\beta$

- Even though we start with log-normal distribution, convolution and nested structure of our model brings complex transformations

- Different values of $\beta$ should be selected for every stage, as the parameters of the distributions also vary

- We have values $\beta$ estimated by the algorithm for general distributions on 100,000 scenarios

- An alternative would be $\beta = 0.3$ which provides low variance for most of our charts
Optimal beta for the asset allocation model

Asset allocation - values of beta

- Beta values over stages for 5-, 10-, and 15-stage models.
Results

- Standard Monte Carlo setup $\hat{Q}^s$ ($\beta_t = \alpha_t = 0.05$)
- Improved estimator $\hat{Q}^i$ with variable $\beta_t$ based on our analysis
- Lower bound $z$

<table>
<thead>
<tr>
<th>$T$</th>
<th>total scenarios</th>
<th>$z$</th>
<th>$\hat{Q}^s$ (s.d.)</th>
<th>$\hat{Q}^i$ (s.d.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6,250,000</td>
<td>-3.5212</td>
<td>-3.5166 (0.0168)</td>
<td>-3.5170 (0.0111)</td>
</tr>
<tr>
<td>10</td>
<td>$\approx 10^{15}$</td>
<td>-7.3885</td>
<td>-7.2833 (0.2120)</td>
<td>-7.2838 (0.0303)</td>
</tr>
<tr>
<td>15</td>
<td>$\approx 10^{24}$</td>
<td>-10.4060</td>
<td>-10.1482 (0.8184)</td>
<td>-10.1245 (0.1355)</td>
</tr>
</tbody>
</table>

- For $T = 5, 10, 15$ we achieved roughly 35%, 85% and 85% reduction of standard deviation
- Negligible effect on computation times
Contamination for multistage risk-averse problems

- Captures various changes in the probability distribution
- Assume that it’s possible to reformulate the stochastic program as:

\[
\min_{x \in X} F(x, \mathbb{P}) := \min_{x \in X} \int_{\Omega} f(x, \xi) \mathbb{P}(d\xi)
\]

- Simplest case of contamination, we obtain global bounds
- It’s possible to reformulate our CVaR model with auxiliary variable \( u \) in this way

- Let \( \mathbb{Q} \) be another fixed probability distribution and define contaminated distributions

\[
\mathbb{P}_k := (1 - k)\mathbb{P} + k\mathbb{Q}, \ k \in [0, 1]
\]

- Suppose that the stochastic program has a solution

\[
\varphi(k) := \min_{x \in X} F(x, \mathbb{P}_k)
\]

for all these distributions
Contamination for multistage risk-averse problems

- Suppose nonempty, bounded set of optimal solutions $\mathcal{X}^*(\mathbb{P})$ of the initial stochastic program
- Then the directional derivative is given by:
  \[
  \varphi'(0^+) = \min_{x \in \mathcal{X}^*(\mathbb{P})} F(x, \mathcal{Q}) - \varphi(0)
  \]
- $\varphi(k)$ concave on $[0, 1]$
- The contamination bounds follow:
  \[
  (1 - k) \varphi(0) + k \varphi(1) \leq \varphi(k) \leq \varphi(0) + k \varphi'(0^+), \ k \in [0, 1]
  \]
Contamination for multistage risk-averse problems

- For large-scale problems we cannot compute a precise solution
  - We apply SDDP to the sampled distributions $\hat{P}$ and $\hat{Q}$
  - We have deterministic lower bound $\underline{\varphi}$ for problems under $\hat{P}$ and $\hat{Q}$
  - We use our estimator to obtain upper bound $\overline{\varphi}$ under $\hat{P}$
- Let $\tilde{x}^*$ be the approximate solution of the initial problem
  - We compute upper estimate $\bar{F}(\tilde{x}^*, \hat{Q})$ of $F(\tilde{x}^*, \hat{Q})$ in the similar manner as we compute our improved upper bound
    - The solution of the initial problem is represented by the sets of cuts
    - We sample the scenarios from $\hat{Q}$
    - We calculate the solution using the cuts from the initial problem and the scenario from the contaminating problem
    - This solution is used in the upper bound calculation
- The approximate contamination bounds are given by:
  \[(1 - k)\underline{\varphi}(\hat{P}) + k\overline{\varphi}(\hat{Q}) \leq \varphi(\hat{P}_k) \leq (1 - k)\overline{\varphi}(\hat{P}) + k\bar{F}(\tilde{x}^*, \hat{Q})\]
Numerical results

- Monthly data from Prague Stock Exchange, January 2009 to February 2012
- Risk aversion coefficients set to $\lambda_t = 10\%$
- Contaminating distribution $Q$ was obtained by increasing the variance by 20%
- 3 and 5 stage problems with 1,000 descendants per node
- We have calculated the derivative values for 10 times and used their mean as well as empirical statistical upper bound
Results - 3 stages without transaction costs

Contamination bounds for parameter k

Objective function

-2.043
-2.042
-2.041
-2.040
-2.039
-2.038
-2.037
-2.036

0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
0.40
0.45
0.50
0.55
0.60
0.65
0.70
0.75
0.80
0.85
0.90
0.95
1.00

-2.036
-2.037
-2.038
-2.039
-2.040
-2.041
-2.042
-2.043
Results - 5 stages without transaction costs
Model comparison

- Monthly data from Prague Stock Exchange, January 2009 to February 2012

<table>
<thead>
<tr>
<th>asset</th>
<th>mean</th>
<th>std. deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAA</td>
<td>1.0290</td>
<td>0.1235</td>
</tr>
<tr>
<td>CETV</td>
<td>0.9984</td>
<td>0.2469</td>
</tr>
<tr>
<td>ČEZ</td>
<td>0.9990</td>
<td>0.0647</td>
</tr>
<tr>
<td>ERSTE GROUP BANK</td>
<td>1.0172</td>
<td>0.1673</td>
</tr>
<tr>
<td>KOMERČNÍ BANKA</td>
<td>1.0110</td>
<td>0.1157</td>
</tr>
<tr>
<td>ORCO</td>
<td>1.0085</td>
<td>0.2200</td>
</tr>
<tr>
<td>PEGAS NONWOVENS</td>
<td>1.0221</td>
<td>0.0863</td>
</tr>
<tr>
<td>PHILIP MORRIS ČR</td>
<td>1.0213</td>
<td>0.0719</td>
</tr>
<tr>
<td>TELEFÓNICA C.R.</td>
<td>0.9993</td>
<td>0.0595</td>
</tr>
<tr>
<td>UNIPETROL</td>
<td>1.0079</td>
<td>0.0843</td>
</tr>
<tr>
<td>VIENNA INSURANCE GROUP</td>
<td>1.0074</td>
<td>0.1100</td>
</tr>
</tbody>
</table>
Model comparison

- Two different models, nested and multi-period CVaR risk measure
- Two different distributions:
  - $P$ based on the input data
  - $Q$ constructed from $P$ by increasing the variance by 20% to test the stability
- We have repeated the sampling for 10 times
- The CVaR levels $\alpha_t$ were always set to 5%
- No transaction costs, $f_t = 0$
- Three-stage model with 1,000 descendants per node, total of 1,000,000 scenarios
- Best performing assets - AAA, PEGAS and PHILLIP MORRIS
Model comparison

- Both models relatively stable with respect to variance of the underlying distribution
- Nested model has more stable solutions and better diversification
- When increasing $\lambda_t$, solutions become more stable in both models

<table>
<thead>
<tr>
<th>$\lambda_t$</th>
<th>model</th>
<th>distr.</th>
<th>AAA</th>
<th>PEGAS</th>
<th>PHILIP M.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>nested</td>
<td>$\hat{P}$</td>
<td>0.2388 (0.1133)</td>
<td>0.3893 (0.1109)</td>
<td>0.3720 (0.1011)</td>
</tr>
<tr>
<td></td>
<td>nested</td>
<td>$\hat{Q}$</td>
<td>0.2718 (0.1600)</td>
<td>0.3582 (0.0902)</td>
<td>0.3700 (0.1565)</td>
</tr>
<tr>
<td>0.1</td>
<td>multiper.</td>
<td>$\hat{P}$</td>
<td>0.6034 (0.3681)</td>
<td>0.2262 (0.2084)</td>
<td>0.1704 (0.2000)</td>
</tr>
<tr>
<td></td>
<td>multiper.</td>
<td>$\hat{Q}$</td>
<td>0.6032 (0.3453)</td>
<td>0.1660 (0.1562)</td>
<td>0.2308 (0.2369)</td>
</tr>
<tr>
<td>0.2</td>
<td>nested</td>
<td>$\hat{P}$</td>
<td>0.1774 (0.0681)</td>
<td>0.4132 (0.0774)</td>
<td>0.4032 (0.0907)</td>
</tr>
<tr>
<td></td>
<td>nested</td>
<td>$\hat{Q}$</td>
<td>0.1730 (0.0541)</td>
<td>0.3471 (0.0566)</td>
<td>0.4545 (0.0429)</td>
</tr>
<tr>
<td>0.2</td>
<td>multiper.</td>
<td>$\hat{P}$</td>
<td>0.3081 (0.1472)</td>
<td>0.2993 (0.1757)</td>
<td>0.3926 (0.0990)</td>
</tr>
<tr>
<td></td>
<td>multiper.</td>
<td>$\hat{Q}$</td>
<td>0.3127 (0.1776)</td>
<td>0.3963 (0.0975)</td>
<td>0.2910 (0.1781)</td>
</tr>
</tbody>
</table>
Future research

- Statistical properties of the proposed upper bound estimators
- Approximation functions applicable in importance sampling schemes for various practical problems
- Application in hydroelectric scheduling under inflow uncertainty
- Develop analogous schemes and estimators for other risk measures
  - Spectral risk measures based on CVaR
- Scenario reduction techniques under stage-wise independence
- More general structures without the stage-wise independence assumption
  - Markov chains
  - Additive dependence models
- Implement parallel processing for SDDP
Conclusion

Thank you!

Václav Kozmík
vaclav@kozmik.cz
References


References


Publications

- **DUPAČOVÁ, J. and KOZMÍK, V. (2014):** *Stress Testing for Risk-Averse Stochastic Programs*
  - To appear in Acta Mathematica Universitatis Comenianae

- **KOZMÍK, V. (2014):** *On Variance Reduction of Mean-CVaR Monte Carlo Estimators*
  - Accepted in Computational Management Science, doi: 10.1007/s10287-014-0225-7

- **DUPAČOVÁ, J. and KOZMÍK, V. (2014):** *Structure of Risk-Averse Multistage Stochastic Programs*
  - Accepted in OR Spectrum, doi: 10.1007/s00291-014-0379-2

- **KOZMÍK, V. and MORTON, D. (2014):** *Evaluating policies in risk-averse multi-stage stochastic programming*
  - Accepted in Mathematical Programming Series A, doi: 10.1007/s10107-014-0787-8


Citations


