A Solver for Problems with Second-Order Stochastic Dominance Constraints

Victor Zverovich, Gautam Mitra, Csaba Fábián

AMPL Optimization OptiRisk SYSTEMS

ICSP2013, Bergamo, Italy. July 8-12, 2013

Second-Order Stochastic Dominance

Let R and R' be random variables defined on the probability space $(\Omega, \mathcal{F}, P)$.

R dominates R' with respect to SSD if and only if $\mathrm{E}[U(R)] \geq \mathrm{E}[U(R')]$ for any nondecreasing and concave utility function U.

This sets out the use of SSD relation to determine preferences of a risk-averse decision maker.

Denoted as $R \succeq_{SSD} R'$.

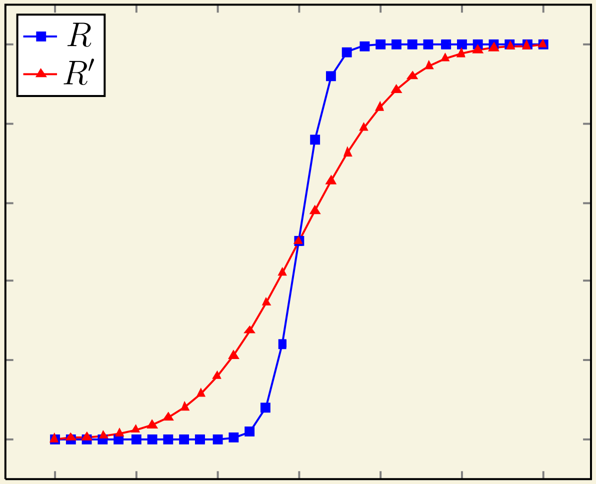
Strict relation: $R \succ_{SSD} R' \Leftrightarrow R \succeq_{SSD} R' \text{ and } R' \not\succeq_{SSD} R$.

Alternative Definitions of SSD

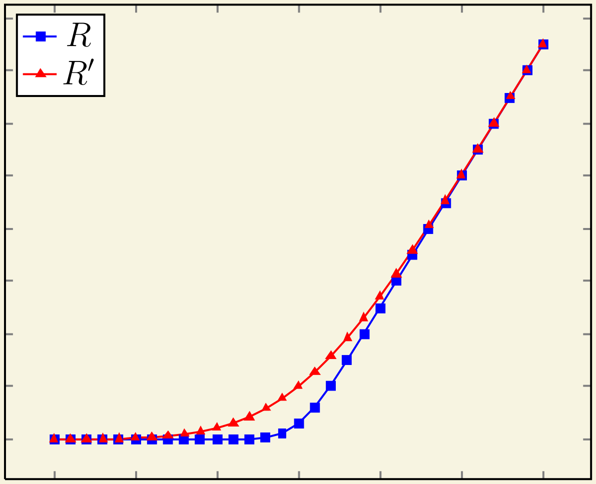
- Definition using the performance function (Fishburn and Vickson, 1978):
  $$F_R^{(2)}(t) \leq F_{R'}^{(2)}(t) \text{ for all } t \in \mathbb{R},$$
  where the performance function $F_R^{(2)}(t) = \int_{-\infty}^{t} F_R(u)\mathrm{d}u$ represents the area under the graph of the cumulative distribution function $F_R(t) = P(R \leq t)$ of a real-valued random variable R.

- Definition using the Tail function (Ogryczak and Ruszczyński, 2002):
  $$\mathrm{Tail}_\alpha(R) \geq \mathrm{Tail}_\alpha(R') \text{ for all } 0 < \alpha \leq 1,$$
  where $\mathrm{Tail}_\alpha(R)$ denotes the unconditional expectation of the smallest $\alpha \cdot 100\%$ of the outcomes of R.

Illustration of Second-Order Stochastic Dominance



CDF



Performance Functions

Portfolio Problem/Constraints

There are n assets and at the beginning of a time period an investor has to decide what proportion x_i of the initial wealth to invest in asset i. So a portfolio is represented by a vector $\mathbf{x} = (x_1, x_2, \ldots, x_n) \in X \subset \mathbb{R}^n$, where X is a bounded convex polytope representing the set of feasible portfolios; in particular it can be defined as

$$X = \{\mathbf{x} \in \mathbb{R}^n_+ : \sum_{i=1}^{n} x_i = 1\},$$

if short positions are not allowed and there are no other modelling restrictions. Let $\mathbf{R}$ denote the n-dimensional random vector of asset returns at the end of the time period. Then the real-valued random variable $R_{\mathbf{x}} = \mathbf{R}^T \mathbf{x}$ is the random return of portfolio $\mathbf{x}$.

Model of Dentcheva and Ruszczynski

Dentcheva and Ruszczyński (2006) proposed the following model with an SSD constraint:

$$
\begin{array}{ll}
\text{maximize} & f(x) \\
\text{s. t.} & x \in X, \\
 & R_{\mathbf{x}} \succeq_{SSD} \widehat{R},
\end{array}
$$

where f is a concave continuous function, $\widehat{R}$ is a reference random return such as the return of a stock market index.

Special case: $f(x) = \mathrm{E}[R_{\mathbf{x}}]$

Model of Roman, Darby-Dowman, and Mitra

Roman et al. (2006) formulated a multiobjective LP model, the Pareto efficient solutions of which are SSD efficient portfolios.

Assuming finite discrete distributions of returns with equiprobable outcomes, **Fábián** et al. (2009) converted it into a more efficient computational model with single objective and a finite system of inequalities representing an SSD constraint:

$$\begin{array}{ll} \text{maximize} & \vartheta \\ \text{s. t.} & \vartheta \in \mathbb{R}, \mathbf{x} \in X \\ & \mathrm{Tail}_{\frac{i}{S}}(R_{\mathbf{x}}) \geq \mathrm{Tail}_{\frac{i}{S}}(\widehat{R}) + \vartheta, \quad i = 1, 2, \ldots, S. \end{array}$$

Here one seeks a portfolio with a distribution which dominates the reference one or comes close to it uniformly (the smallest tail difference ϑ is maximized).

Model with SSD Constraints

Fábián et al. (2010) proposed an enhanced version of the model of Roman et al. which is expressed in the following SSD constrained form:

$$\begin{aligned} \text{maximize} \quad & \vartheta \\ \text{s.t.} \quad & \vartheta \in \mathbb{R}, \mathbf{x} \in X, \\ & R_{\mathbf{x}} \succeq_{_{SSD}} \widehat{R} + \vartheta. \end{aligned}$$

In this model one computes a portfolio that dominates a sum of the reference return and a riskless return ϑ.

Formulation Using Tails

Let S denote the number of equiprobable outcomes,

$\mathbf{r}^{(1)}, \mathbf{r}^{(2)}, \ldots, \mathbf{r}^{(S)}$ - the realisations of $\mathbf{R}$,

$\hat{r}^{(1)}, \hat{r}^{(2)}, \ldots, \hat{r}^{(S)}$ - the realisations of $\widehat{R}$.

The enhanced model can be formulated as follows:

$$\begin{array}{ll} \text{maximize} & \vartheta \\ \text{s. t.} & \vartheta \in \mathbb{R}, \mathbf{x} \in X, \\ & \operatorname{Tail}_{\frac{i}{S}}(R_{\mathbf{x}}) \geq \operatorname{Tail}_{\frac{i}{S}}(\widehat{R}) + \dfrac{i}{S}\,\vartheta, \\ & \quad i = 1, 2, \ldots, S. \end{array}$$

Cutting-Plane Formulation Using Tails

Fábián et al. (2009) obtained the cutting-plane representation of the Tail function:

$$\mathrm{Tail}_{\frac{i}{S}}(R_{\mathbf{x}}) = \quad \min \frac{1}{S} \sum_{j \in J_i} \mathbf{r}^{(j)T} \mathbf{x}$$

$$\text{such that } J_i \subset \{1, 2, \ldots, S\}, \quad |J_i| = i.$$

Cutting-plane representation of the enhanced model:

$$\begin{aligned} &\text{maximize} && \vartheta \\ &\text{s.t.} && \vartheta \in \mathbb{R}, \mathbf{x} \in X, \\ & && \frac{1}{S} \sum_{j \in J_i} \mathbf{r}^{(j)T} \mathbf{x} \geq \widehat{\tau}_i + \frac{i}{S} \vartheta, \quad \forall J_i \subset \{1, 2, \ldots, S\}, \\ & && \qquad |J_i| = i, \; i = 1, 2, \ldots, S, \end{aligned}$$

where $\widehat{\tau}_i = \mathrm{Tail}_{\frac{i}{S}}(\widehat{R})$.

Cutting-Plane Method

By changing the scope of optimisation we get a problem of minimising a piecewise-linear convex function:

$$\begin{aligned} &\text{minimize} && \varphi(\mathbf{x}) \\ &\text{s. t.} && \mathbf{x} \in X, \end{aligned}$$

where

$$\begin{aligned} \varphi(\mathbf{x}) = \quad & \max\left(-\frac{1}{i} \sum_{j \in J_i} \mathbf{r}^{(j)T} \mathbf{x} + \frac{S}{i}\, \widehat{\tau}_i \right), \\ & \text{such that } J_i \subset \{1, 2, \ldots, S\}, |J_i| = i, \\ & i = 1, 2, \ldots, S. \end{aligned}$$

It can be regularised by the level method.

Cut Generation

The cut $l(x)$ at the iteration k is constructed as follows:

Let $\mathbf{x}^* \in X$ denote the solution of the approximation function at iteration k and $\mathbf{r}^{(j_1^*)} \leq \mathbf{r}^{(j_2^*)} \leq \ldots \leq \mathbf{r}^{(j_S^*)}$ denote the ordered realisations of $R_{\mathbf{x}^*}$.

Select $i^* \in \operatorname{argmax}_{1 \leq i \leq S} \left(-\frac{1}{i} \sum_{j \in J_i^*} \mathbf{r}^{(j)T} \mathbf{x}^* + \frac{S}{i} \, \widehat{\tau}_i \right)$. Then

$$l(\mathbf{x}) = -\frac{1}{i^*} \sum_{j \in J_{i^*}^*} \mathbf{r}^{(j)T} \mathbf{x} + \frac{S}{i^*} \, \widehat{\tau}_{i^*}.$$

Sets $J_i^* = (j_1^*, \ldots, j_i^*)$ correspond to ordered realisations.

Why a New Solver?

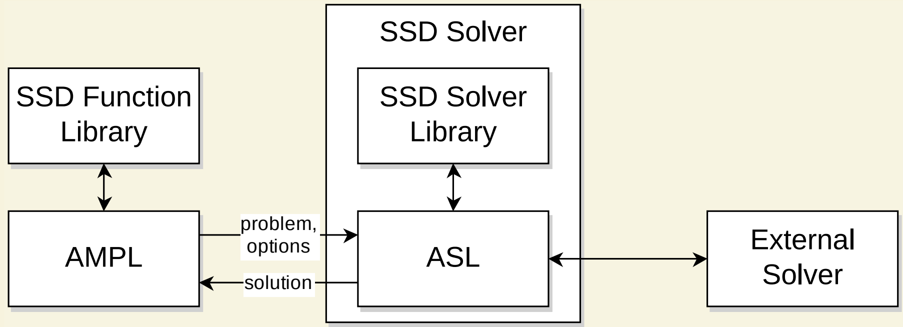
- Old implementation:
  - Cuts are a part of the model
  - Difficult to reuse
- New implementation:
  - Cuts are added automatically by the solver
  - Easy to use
  - "Clean" model
  - Faster

AMPL Solver Library

AMPL Solver Library (ASL) is an open-source library for connecting solvers to AMPL.

- C interface:
  - described in **Hooking Your Solver to AMPL**
  - used by most solvers
- **C++ interface**:
  - makes connecting new solvers super easy
  - type-safe: no casts needed when working with expression trees
  - efficient: no overhead compared to the C interface
  - used by several CP solvers and the SSD solver

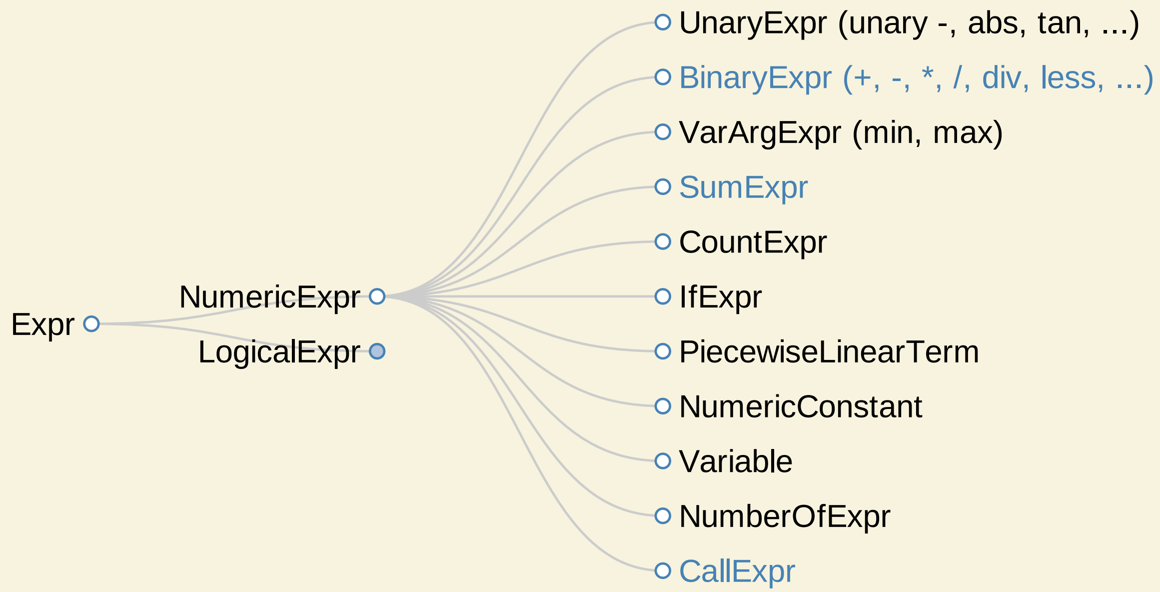
SSD Solver Architecture



- ASL does all the heavy lifting such as interaction with AMPL and an external solver which makes SSD solver implemenation very simple (~300 LOC!)
- Function library provides the `ssd_uniform` function that is translated into an SSD relation by the solver.
- External solver is used for subproblems.
- Solver library is optional but facilitates testing.

Expression Trees

The solver extracts linear expressions from the expression trees representing arguments of `ssd_uniform`.



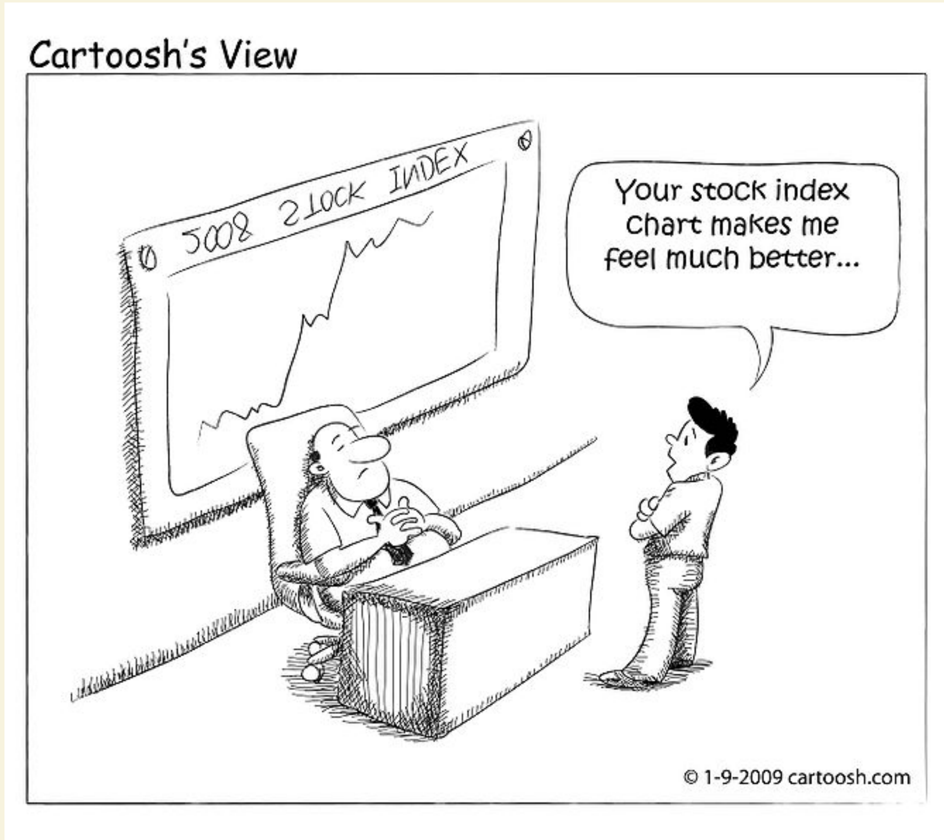
Portfolio Model in AMPL with Cuts

```
param nASSET integer >= 0;  # number of assets
set ASSETS := 1..nASSET;    # set of assets

param nSCEN > 0;
param asset_returns{1..nASSET, 1..nSCEN};
param index_returns{1..nSCEN};

param nCUT integer >= 0 default 0; # number of cuts
set CUTS := 1..nCUT;        # set of cuts

param cut_const {CUTS};     # constant in cut
param cut {CUTS,ASSETS};    # multipliers in cut
param scaling_factor {CUTS} default 1;

# portfolio: investments into different assets
var Invest {ASSETS} >= 0 default 1 / nASSET;
var Dom;                    # dominance measure

maximize Uniform_Dominance: Dom;

subject to Dom_constraint {c in CUTS}:
  scaling_factor[c] * Dom + cut_const[c]
    <= sum {a in ASSETS} cut[c,a] * Invest[a];

subject to Budget: sum {a in ASSETS} Invest[a] = 1;
```

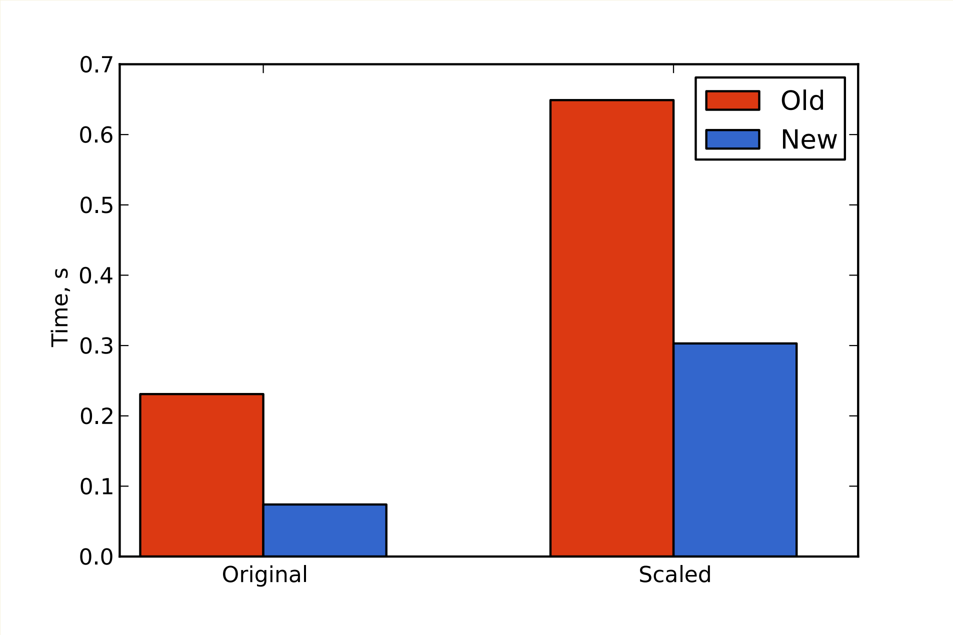
Portfolio Model in AMPL using SSD Solver

```
include ssd.ampl;

param NumScenarios;
param NumAssets;

set Scenarios = 1..NumScenarios;
set Assets = 1..NumAssets;

# Return of asset a in senario s.
param Returns{a in Assets, s in Scenarios};

# Reference return in scenario s.
param Reference{s in Scenarios};

# Fraction of the budget to invest in asset a.
var invest{a in Assets} >= 0 <= 1;

subject to ssd_constraint{s in Scenarios}:
  ssd_uniform(sum{a in Assets} Returns[a, s] * invest[a], Reference[s]);

subject to budget: sum{a in Assets} invest[a] = 1;
```

Reference Returns

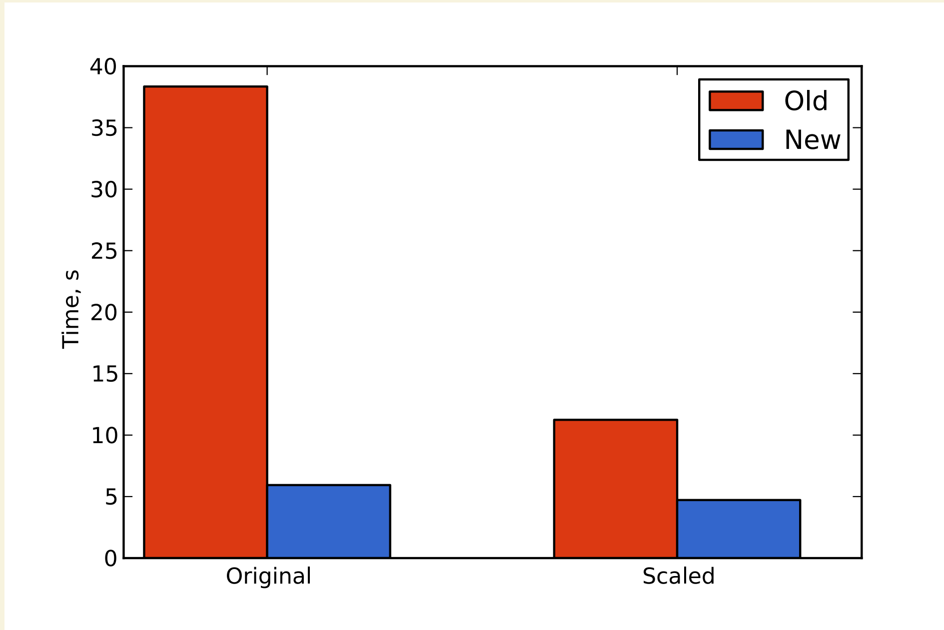


Performance



- 100 scenario problem with FTSE100 used as a reference.
- The new implementation is 2-3 times faster.

Performance



- 30000 scenario problem with FTSE100 used as a reference.
- The new implementation is 2-6.5 times faster.

Summary

- AMPL solver interface and ASL make implementation of high-level solvers/algorithms that use other solvers easy. The same technique can be applied to
  - other-cutting plane methods
  - decomposition methods, e.g. Bender's decomposition
- New solver provides an efficient implementation of a cutting-plane algorithm for solving problems with SSD constraints.
- This is in line with our approach that different types of optimisation models are matched with corresponding solvers.

References

- Dentcheva, D. and Ruszczyński, A. (2006). Portfolio optimization with stochastic dominance constraints. Journal of Banking & Finance, 30 , 433–451.
- Fábián, C. I., Mitra, G., and Roman, D. (2009). Processing second-order stochastic dominance models using cutting-plane representations. Mathematical Programming, Series A. DOI: 10.1007/s10107-009-0326-1.
- Fábián, C. I., Mitra, G., Roman, D., and Zverovich, V. (2010). An enhanced model for portfolio choice with ssd criteria: a constructive approach. Quantitative Finance. First published on: 11 May 2010.
- Fishburn, P. C. and Vickson, R. G. (1978). Theoretical foundations of stochastic dominance. In Stochastic Dominance: An Approach to Decision-Making Under Risk, (pp. 37–113). D.C. Heath and Company, Lexington, Massachusetts.
- Roman, D., Darby-Dowman, K., and Mitra, G. (2006). Portfolio construction based on stochastic dominance and target return distributions. Mathematical Programming, eries B, 108, 541–569.

Thank you!

AMPL Optimization

OptiRisk SYSTEMS