A recursive algorithm for set-valued risk measures

Zach Feinstein

ORFE,
Princeton University

Joint work with Birgit Rudloff

Leibniz Universität Hannover
November 27, 2013

Outline

1. Set-valued risk measures
   - Definition
   - Time consistency
2. Recursive algorithm
   - Bellman's Principle
3. Computation
   - Benson's Algorithm
   - Polyhedral Risk Measures
   - Conditionally Convex Risk Measures
4. Market Extension
   - Idea and Interpretation
   - Computation

1.1 Set-valued risk measures: Definition

- Probability space $(\Omega, (\mathcal{F}_t)_{t\in[0,T]}, \mathbb{P})$
- d assets (may include different currencies)
- Portfolio vectors in physical units (numéraire free), i.e. number of units in d assets
- Claim: $X \in L^p_d(\mathcal{F}_T)$ payoff (in physical units) at time T
- Convex transaction costs at time t: closed convex set $\mathbb{R}^d_+ \subseteq C_t[\omega] \subseteq \mathbb{R}^d$ (solvency region), positions transferable into non-negative portfolios
- Eligible portfolios M, linear subspace of $\mathbb{R}^d$ of portfolios that can be used to compensate risk (e.g. Dollars & Euros)

1.1 Set-valued risk measures: Definition

- $M_t := L^p_d(\mathcal{F}_t; M), \quad M_{t,+} := M_t \cap L^p_d(\mathcal{F}_t)_+$
- $\mathcal{P}(\mathcal{Z}; C) := \{A \subseteq \mathcal{Z} : A = A + C\}$
- $\mathcal{G}(\mathcal{Z}; C) := \{A \subseteq \mathcal{Z} : A = \operatorname{cl}\operatorname{co}(A + C)\}$

- $M_t := L^p_d(\mathcal{F}_t; M), \quad M_{t,+} := M_t \cap L^p_d(\mathcal{F}_t)_+$
- $\mathcal{P}(\mathcal{Z}; C) := \{A \subseteq \mathcal{Z} : A = A + C\}$
- $\mathcal{G}(\mathcal{Z}; C) := \{A \subseteq \mathcal{Z} : A = \operatorname{cl}\operatorname{co}(A + C)\}$

Conditional Set-Valued Risk Measure

A set-valued function $R_t : L^p_d(\mathcal{F}_T) \to \mathcal{P}(M_t; M_{t,+})$ is a conditional risk measure if

1.1 Set-valued risk measures: Definition

- $M_t := L_d^p(\mathcal{F}_t; M), \quad M_{t,+} := M_t \cap L_d^p(\mathcal{F}_t)_+$
- $\mathcal{P}(\mathcal{Z}; C) := \{A \subseteq \mathcal{Z} : A = A + C\}$
- $\mathcal{G}(\mathcal{Z}; C) := \{A \subseteq \mathcal{Z} : A = \operatorname{cl}\operatorname{co}(A + C)\}$

Conditional Set-Valued Risk Measure

A set-valued function $R_t : L_d^p(\mathcal{F}_T) \to \mathcal{P}(M_t; M_{t,+})$ is a conditional risk measure if

1. Finite at zero: $\emptyset \neq R_t(0) \neq M_t$;
2. M_t translative: $R_t(X + m) = R_t(X) - m$ for any $m \in M_t$;
3. $L_d^p(\mathcal{F}_T)_+$ monotone: if $X - Y \in L_d^p(\mathcal{F}_T)_+$ then $R_t(X) \supseteq R_t(Y)$.

1.2 Set-valued risk measures: Time consistency

Multi-Portfolio Time Consistency

A dynamic risk measure $(R_t)_{t=0}^{T}$ is ***multi-portfolio time consistent*** if the relation

$$R_s(X) \subseteq \bigcup_{Y \in \mathbf{Y}} R_s(Y) \Rightarrow R_t(X) \subseteq \bigcup_{Y \in \mathbf{Y}} R_t(Y)$$

for any times $t < s$, any $X \in L_d^p(\mathcal{F}_T)$ and any $\mathbf{Y} \subseteq L_d^p(\mathcal{F}_T)$

Multi-Portfolio Time Consistency

A dynamic risk measure $(R_t)_{t=0}^T$ is ***multi-portfolio time consistent*** if the relation

$$R_s(X) \subseteq \bigcup_{Y \in \mathbf{Y}} R_s(Y) \Rightarrow R_t(X) \subseteq \bigcup_{Y \in \mathbf{Y}} R_t(Y)$$

for any times $t < s$, any $X \in L_d^p(\mathcal{F}_T)$ and any $\mathbf{Y} \subseteq L_d^p(\mathcal{F}_T)$

Composition of One-Step Risk Measures

Let $(R_t)_{t=0}^T$ be a risk measure then $\left(\tilde{R}_t\right)_{t=0}^T$ is the multi-portfolio time consistent version if

$$\tilde{R}_T(X) := R_T(X); \qquad \tilde{R}_t(X) := \bigcup_{Z \in \tilde{R}_{t+1}(X)} R_t(-Z)$$

2. Recursive algorithm: Setting

- Discrete time $t \in \{0, 1, ..., T\}$
- Finite probability space $(\Omega, (\mathcal{F}_t)_{t=0}^{T}, \mathbb{P})$
- Let Ω_t be the set atoms in $\mathcal{F}_t$
- The set of successor nodes for $\omega_t \in \Omega_t$ is given by $\mathrm{succ}(\omega_t) = \{\omega_{t+1} \in \Omega_{t+1} : \omega_{t+1} \subseteq \omega_t\}$
- $R_t(X)[\omega_t] = \{u(\omega_t) : u \in R_t(X)\}$

2. Recursive algorithm: Setting

- $R_t(X)[\omega_t] = \{u(\omega_t) : u \in R_t(X)\}$
- $R_t(X)[\omega_t]$ behaves like static risk measure

2. Recursive algorithm: Setting

- $R_t(X)[\omega_t] = \{u(\omega_t) : u \in R_t(X)\}$
- $R_t(X)[\omega_t]$ behaves like static risk measure

Pointwise Representation

If R_t has closed and conditionally convex images then $u \in R_t(X)$ if and only if $u(\omega_t) \in R_t(X)[\omega_t]$ for every $\omega_t \in \Omega_t$.

- $R_t(X)[\omega_t] = \{u(\omega_t) : u \in R_t(X)\}$
- $R_t(X)[\omega_t]$ behaves like static risk measure

Pointwise Representation

If R_t has closed and conditionally convex images then $u \in R_t(X)$ if and only if $u(\omega_t) \in R_t(X)[\omega_t]$ for every $\omega_t \in \Omega_t$.

- If R_t is closed and conditionally convex, then R_t has closed and conditionally convex images.

2. Recursive algorithm: Setting

- $R_t(X)[\omega_t] = \{u(\omega_t) : u \in R_t(X)\}$
- $R_t(X)[\omega_t]$ behaves like static risk measure

Pointwise Representation

If R_t has closed and conditionally convex images then $u \in R_t(X)$ if and only if $u(\omega_t) \in R_t(X)[\omega_t]$ for every $\omega_t \in \Omega_t$.

- If R_t is closed and conditionally convex, then R_t has closed and conditionally convex images.
- R_t is ***local*** if $1_D R_t(X) = 1_D R_t(1_D X)$ for every $D \in \mathcal{F}_t$

2. Recursive algorithm: Setting

- $R_t(X)[\omega_t] = \{u(\omega_t) : u \in R_t(X)\}$
- $R_t(X)[\omega_t]$ behaves like static risk measure

Pointwise Representation

If R_t has closed and conditionally convex images then $u \in R_t(X)$ if and only if $u(\omega_t) \in R_t(X)[\omega_t]$ for every $\omega_t \in \Omega_t$.

- If R_t is closed and conditionally convex, then R_t has closed and conditionally convex images.
- R_t is ***local*** if $1_D R_t(X) = 1_D R_t(1_D X)$ for every $D \in \mathcal{F}_t$
- If R_t is local then $R_t(X)[\omega_t] = R_t(1_{\omega_t} X)[\omega_t]$

2. Recursive algorithm: Setting

- Let $m \in \text{int}(M_+)$

Approximation

R_t^δ is a δ-***approximation*** of R_t if for every $X \in L_d^p(\mathcal{F}_T)$

$$R_t^\delta(X) + \delta m \mathbf{1} \subseteq R_t(X) \subseteq R_t^\delta(X)$$

2. Recursive algorithm: Setting

- Let $m \in \text{int}(M_+)$

Approximation

R_t^δ is a δ-***approximation*** of R_t if for every $X \in L_d^p(\mathcal{F}_T)$

$$R_t^\delta(X) + \delta m \mathbf{1} \subseteq R_t(X) \subseteq R_t^\delta(X)$$

- Current vector optimization algorithms output a polyhedral approximation for the solution

2.1 Recursive algorithm: Bellman's Principle

- Want to find the multi-portfolio time consistent version of R_t

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well

2.1 Recursive algorithm: Bellman's Principle

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!
- If $(R_t)_{t=0}^T$ is polyhedral (graph R_t is a convex polyhedron for all times t)

2.1 Recursive algorithm: Bellman's Principle

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!
- If $(R_t)_{t=0}^T$ is polyhedral (graph R_t is a convex polyhedron for all times t) then $\left(\tilde{R}_t\right)_{t=0}^T$ is polyhedral

2.1 Recursive algorithm: Bellman's Principle

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!
- If $(R_t)_{t=0}^T$ is polyhedral (graph R_t is a convex polyhedron for all times t) then $\left(\tilde{R}_t\right)_{t=0}^T$ is polyhedral and is closed

2.1 Recursive algorithm: Bellman's Principle

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!
- If $(R_t)_{t=0}^T$ is polyhedral (graph R_t is a convex polyhedron for all times t) then $\left(\tilde{R}_t\right)_{t=0}^T$ is polyhedral and is closed
- If $(R_t)_{t=0}^T$ is convex and convex upper continuous $(R_t^{-1}(D) = \left\{X \in L_d^p(\mathcal{F}_T) : R_t(X) \cap D \neq \emptyset\right\}$ is closed for every $D \in \mathcal{G}(M_t; M_{t,-}))$

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!
- If $(R_t)_{t=0}^T$ is polyhedral (graph R_t is a convex polyhedron for all times t) then $\left(\tilde{R}_t\right)_{t=0}^T$ is polyhedral and is closed
- If $(R_t)_{t=0}^T$ is convex and convex upper continuous $(R_t^{-1}(D) = \left\{X \in L_d^p(\mathcal{F}_T) : R_t(X) \cap D \neq \emptyset\right\}$ is closed for every $D \in \mathcal{G}(M_t; M_{t,-}))$ then $\left(\tilde{R}_t\right)_{t=0}^T$ is convex and convex upper continuous

- Want to find the multi-portfolio time consistent version of R_t
- If $(R_t)_{t=0}^T$ has conditionally convex images then $\left(\tilde{R}_t\right)_{t=0}^T$ does as well
- But, $\left(\tilde{R}_t\right)_{t=0}^T$ does not necessarily have closed images, even if $(R_t)_{t=0}^T$ does!
- If $(R_t)_{t=0}^T$ is polyhedral (graph R_t is a convex polyhedron for all times t) then $\left(\tilde{R}_t\right)_{t=0}^T$ is polyhedral and is closed
- If $(R_t)_{t=0}^T$ is convex and convex upper continuous $(R_t^{-1}(D) = \left\{X \in L_d^p(\mathcal{F}_T) : R_t(X) \cap D \neq \emptyset\right\}$ is closed for every $D \in \mathcal{G}(M_t; M_{t,-}))$ then $\left(\tilde{R}_t\right)_{t=0}^T$ is convex and convex upper continuous and is closed

Closed-Valued Composed Risk Measure

When $\left(\tilde{R}_t\right)_{t=0}^T$ does not have closed images, instead we need to consider $(\bar{R}_t)_{t=0}^T$:

$$\bar{R}_T(X) := \operatorname{cl} R_T(X); \qquad \bar{R}_t(X) := \operatorname{cl} \bigcup_{Z \in \bar{R}_{t+1}(X)} R_t(-Z)$$

- $\bar{R}_t$ is an arbitrarily close approximation for $\tilde{R}_t$

Closed-Valued Composed Risk Measure

When $\left(\tilde{R}_t\right)_{t=0}^T$ does not have closed images, instead we need to consider $(\bar{R}_t)_{t=0}^T$:

$$\bar{R}_T(X) := \operatorname{cl} R_T(X); \qquad \bar{R}_t(X) := \operatorname{cl} \bigcup_{Z \in \bar{R}_{t+1}(X)} R_t(-Z)$$

- $\bar{R}_t$ is an arbitrarily close approximation for $\tilde{R}_t$, for every $\delta > 0$:

$$\bar{R}_t(X) + \delta m \mathbf{1} \subseteq \tilde{R}_t(X) \subseteq \bar{R}_t(X)$$

Closed-Valued Composed Risk Measure

When $\left(\tilde{R}_t\right)_{t=0}^{T}$ does not have closed images, instead we need to consider $(\bar{R}_t)_{t=0}^{T}$:

$$\bar{R}_T(X) := \operatorname{cl} R_T(X); \qquad \bar{R}_t(X) := \operatorname{cl} \bigcup_{Z \in \bar{R}_{t+1}(X)} R_t(-Z)$$

- $\bar{R}_t$ is an arbitrarily close approximation for $\tilde{R}_t$, for every $\delta > 0$:

$$\bar{R}_t(X) + \delta m \mathbf{1} \subseteq \tilde{R}_t(X) \subseteq \bar{R}_t(X)$$

- When $\left(\tilde{R}_t\right)_{t=0}^{T}$ does **not** admit an ω-representation, $(\bar{R}_t)_{t=0}^{T}$ does

Pointwise Representation

$$\begin{aligned}\bar{R}_T(X)[\omega_T] &= \mathrm{cl}(R_T(X)[\omega_T])\\ \bar{R}_t(X)[\omega_t] &= \mathrm{cl}\bigcup\Big\{R_{t,t+1}(-Z)[\omega_t] : \forall \omega_{t+1} \in \mathrm{succ}(\omega_t) :\\ &\qquad\qquad Z(\omega_{t+1}) \in \bar{R}_{t+1}(X)[\omega_{t+1}]\Big\}\end{aligned}$$

- $\bar{R}_t(X)[\omega_t] = \inf_{Z \in \bar{\mathcal{Z}}_{t+1}[\omega_t]} R_{t,t+1}(-Z)[\omega_t]$ with

$$\bar{\mathcal{Z}}_{t+1}[\omega_t] = \big\{ Z \in L^p_d(\mathcal{F}_{t+1}) : \forall \omega_{t+1} \in \operatorname{succ}(\omega_t) : Z(\omega_{t+1}) \in \bar{R}_{t+1}(X)[\omega_{t+1}] \big\}$$

2.1 Recursive algorithm: Bellman's Principle

- $\bar{R}_t(X)[\omega_t] = \inf_{Z \in \bar{\mathcal{Z}}_{t+1}[\omega_t]} R_{t,t+1}(-Z)[\omega_t]$ with

$$\bar{\mathcal{Z}}_{t+1}[\omega_t] = \Big\{ Z \in L^p_d(\mathcal{F}_{t+1}) : \forall \omega_{t+1} \in \operatorname{succ}(\omega_t) : \\ Z(\omega_{t+1}) \in \bar{R}_{t+1}(X)[\omega_{t+1}] \Big\}$$

- Equivalent to vector optimization problem!

- $\bar{R}_t(X)[\omega_t] = \inf_{Z \in \bar{\mathcal{Z}}_{t+1}[\omega_t]} R_{t,t+1}(-Z)[\omega_t]$ with

$$\bar{\mathcal{Z}}_{t+1}[\omega_t] = \left\{ Z \in L^p_d(\mathcal{F}_{t+1}) : \forall \omega_{t+1} \in \operatorname{succ}(\omega_t) : \right. \\ \left. Z(\omega_{t+1}) \in \bar{R}_{t+1}(X)[\omega_{t+1}] \right\}$$

- Equivalent to vector optimization problem!

$$\bar{R}_t(X)[\omega_t] = \inf_{(Z,Y) \in \mathcal{Z}_{t+1}[\omega_t]} \Gamma(Z,Y)$$

for $\Gamma(Z,Y) = Y$ and
$\mathcal{Z}_{t+1}[\omega_t] = \left\{ (Z,Y) \in \bar{\mathcal{Z}}_{t+1}[\omega_t] \times M : Y \in R_{t,t+1}(-Z)[\omega_t] \right\}$

3.1 Computation: Benson's Algorithm

- For linear vector optimization, but has been extended to convex vector optimization
- Linear vector optimization problem:

$$\inf Px \text{ s.t. } Ax \leq b$$

3.1 Computation: Benson's Algorithm

- For linear vector optimization, but has been extended to convex vector optimization
- Linear vector optimization problem:

$$\inf Px \text{ s.t. } Ax \leq b$$

- Let $X = \left\{x \in \mathbb{R}^d : Ax \leq b\right\}$ and $P[X] = \{Px : x \in X\}$.

3.1 Computation: Benson's Algorithm

- For linear vector optimization, but has been extended to convex vector optimization
- Linear vector optimization problem:

$$\inf Px \text{ s.t. } Ax \leq b$$

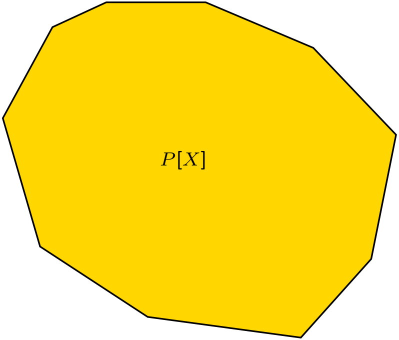
- Let $X = \left\{x \in \mathbb{R}^d : Ax \leq b\right\}$ and $P[X] = \{Px : x \in X\}$. Benson's algorithm finds $\mathcal{P} = P[X] + \mathbb{R}^d_+$

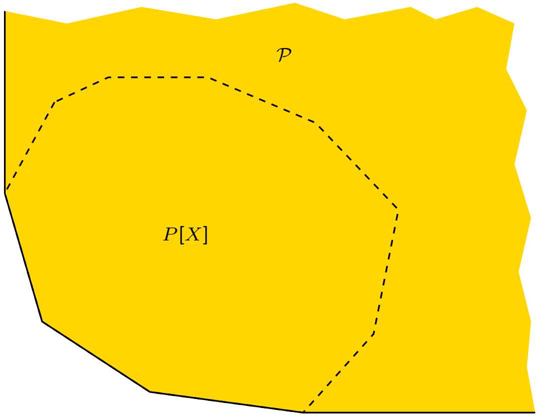
3.1 Computation: Benson's Algorithm

- For linear vector optimization, but has been extended to convex vector optimization
- Linear vector optimization problem:

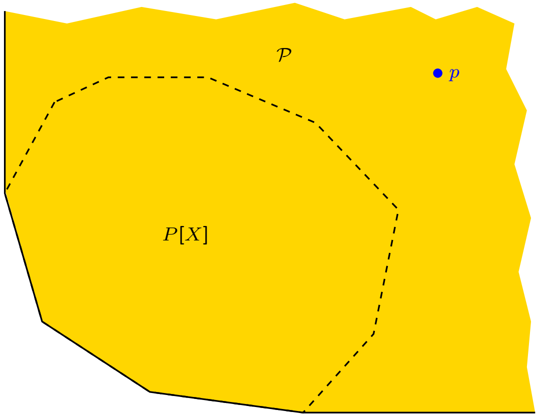
$$\inf Px \text{ s.t. } Ax \leq b$$

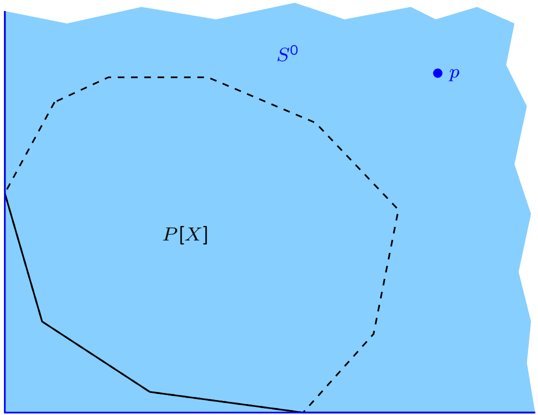
- Let $X = \left\{x \in \mathbb{R}^d : Ax \leq b\right\}$ and $P[X] = \{Px : x \in X\}$. Benson's algorithm finds $\mathcal{P} = P[X] + \mathbb{R}^d_+$
- Following slides were created by Andreas Löhne ("A dual variant of Benson's outer approximation algorithm" at the EURO XXII Conference, 2007, Prague, Czech Republic) and used with permission

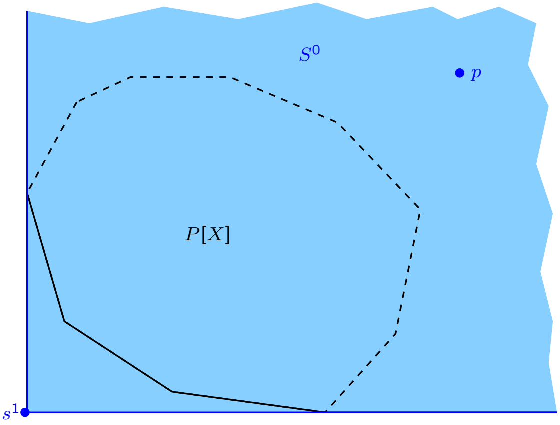

$P[X]$



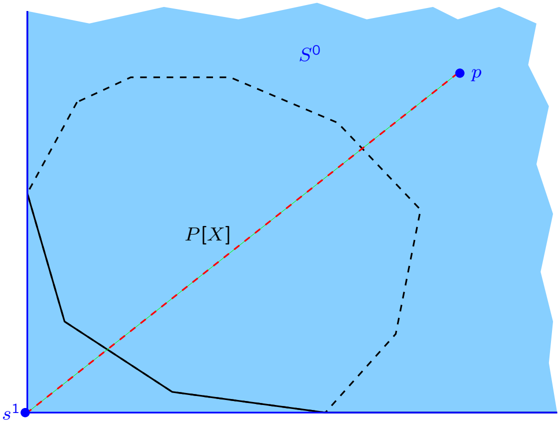
$\mathcal{P}$
$P[X]$

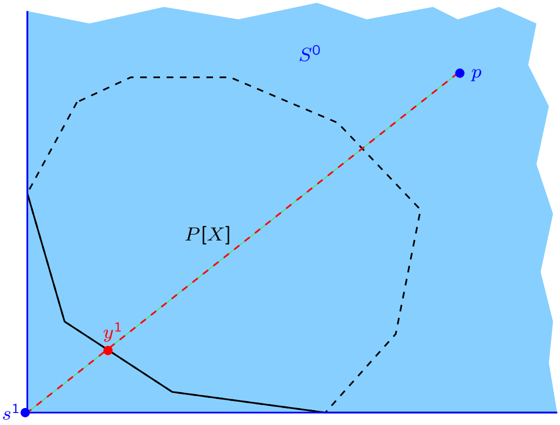

$\mathcal{P}$
p
$P[X]$

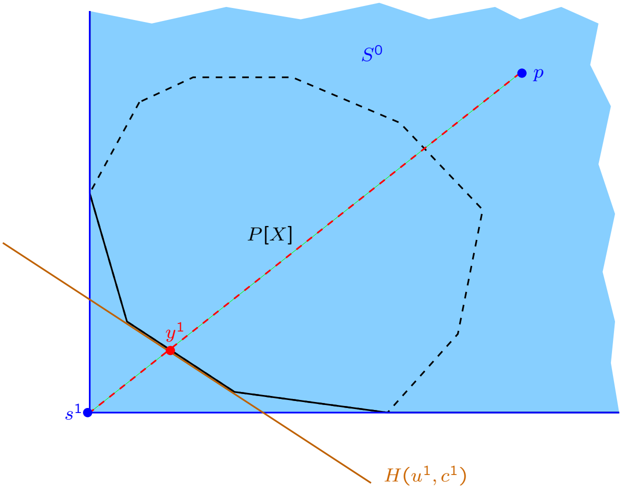

S^0
p
$P[X]$

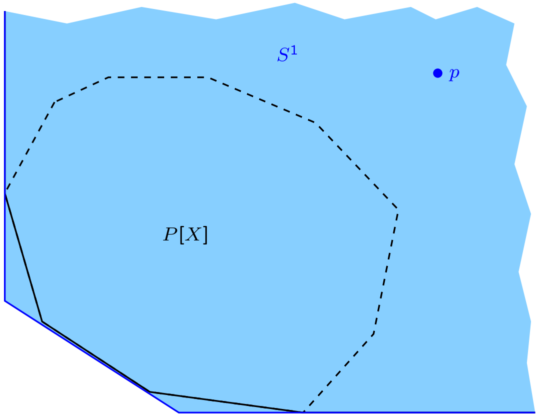


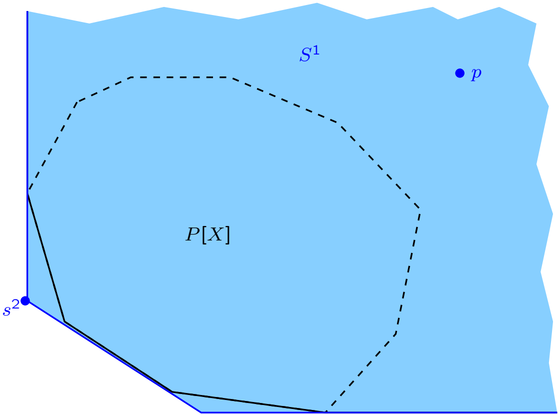
S^0
p
$P[X]$
s^1

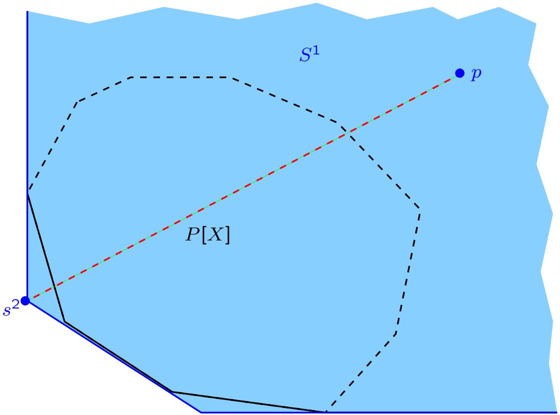

S^0
p
$P[X]$
s^1

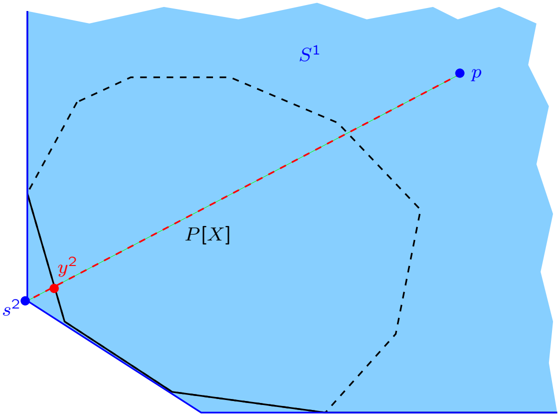

S^0
p
$P[X]$
y^1
s^1

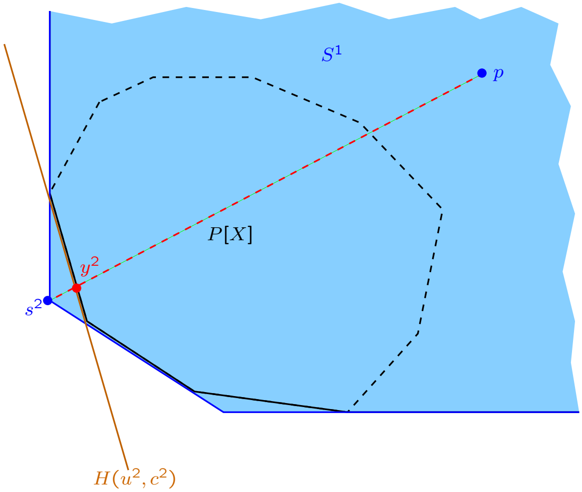

S^0
p
$P[X]$
y^1
s^1
$H(u^1, c^1)$

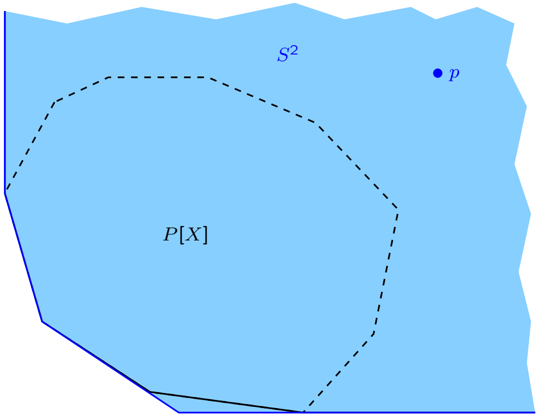

S^1
p
$P[X]$

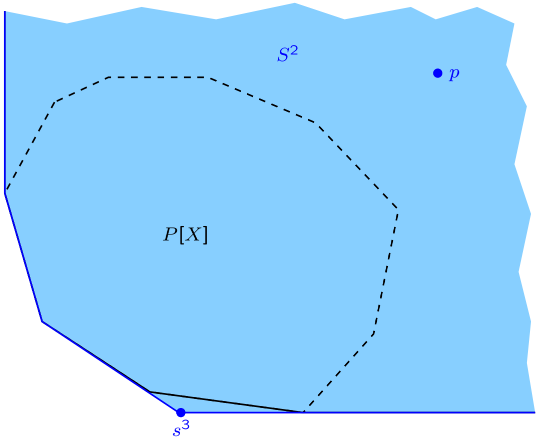

S^1
p
$P[X]$
s^2

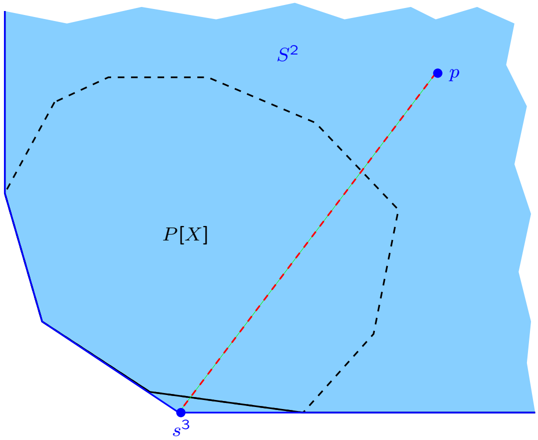

S^1
p
$P[X]$
s^2

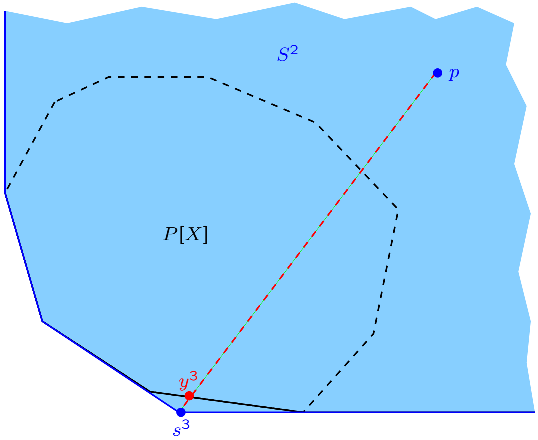

S^1
p
$P[X]$
y^2
s^2

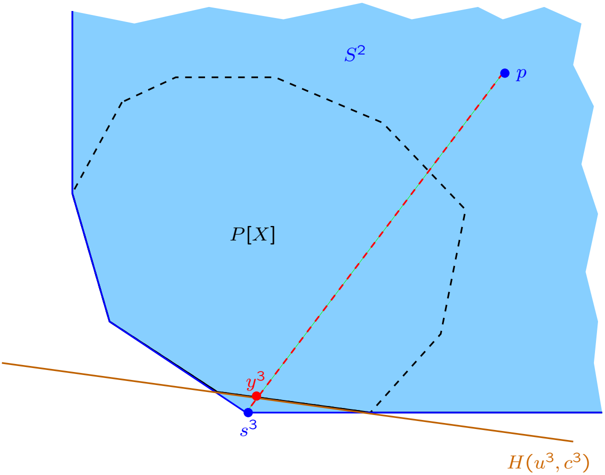

S^1
p
$P[X]$
y^2
s^2
$H(u^2, c^2)$

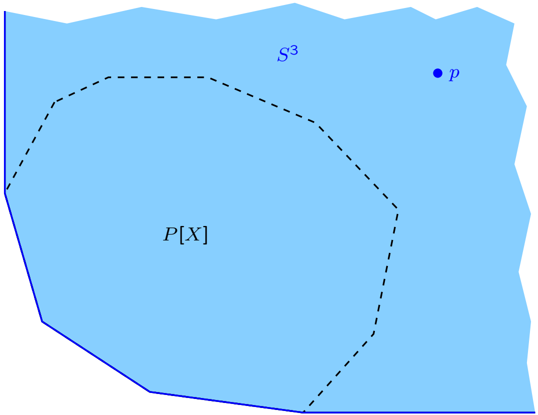

S^2
p
$P[X]$


S^2
p
$P[X]$
s^3


S^2
p
$P[X]$
s^3


S^2
p
$P[X]$
y^3
s^3


S^2
p
$P[X]$
y^3
s^3
$H(u^3, c^3)$


S^3
p
$P[X]$

3.2 Computation: Polyhedral Risk Measures

- Linear vector optimization

$$\begin{aligned}\bar{R}_T(X)[\omega_T] &= \mathrm{cl}(R_T(X)[\omega_T])\\ \bar{R}_t(X)[\omega_t] &= \mathrm{cl}\bigcup\{R_{t,t+1}(-Z)[\omega_t] : \forall \omega_{t+1} \in \mathrm{succ}(\omega_t) :\\ &\qquad Z(\omega_{t+1}) \in \bar{R}_{t+1}(X)[\omega_{t+1}]\}\end{aligned}$$

- $R_{t,t+1}(-Z)[\omega_t] = \{P_t z + M_{t,+} : A_t Z + B_t z \leq b_t\}$
- $\dim(M)$-dimensional problem with $d \times |\mathrm{succ}(\omega_t)| + |z|$-dimensional pre-image space
- Benson's algorithm can be applied directly

3.3 Computation: Conditionally Convex Risk Measures

- Convex vector optimization
- Modified Benson's algorithm provides an approximation

- Convex vector optimization
- Modified Benson's algorithm provides an approximation
- If $\bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}]$ is an ϵ-approximation of $\bar{R}_{t+1}(X)[\omega_{t+1}]$ $(\bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}] + \epsilon m\mathbf{1} \subseteq \bar{R}_{t+1}(X)[\omega_{t+1}] \subseteq \bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}])$ for every $\omega_{t+1} \in \operatorname{succ}(\omega_t)$ then

$$\bar{R}^{\epsilon}_t(X)[\omega_t] = \operatorname{cl} \bigcup \Big\{ R_{t,t+1}(-Z)[\omega_t] : \forall \omega_{t+1} \in \operatorname{succ}(\omega_t) : Z(\omega_{t+1}) \in \bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}] \Big\}$$

is an ϵ-approximation of $\bar{R}_t(X)[\omega_t]$!

- Convex vector optimization
- Modified Benson's algorithm provides an approximation
- If $\bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}]$ is an ϵ-approximation of $\bar{R}_{t+1}(X)[\omega_{t+1}]$ $(\bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}] + \epsilon m\mathbf{1} \subseteq \bar{R}_{t+1}(X)[\omega_{t+1}] \subseteq \bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}])$ for every $\omega_{t+1} \in \operatorname{succ}(\omega_t)$ then

$$\bar{R}^{\epsilon}_t(X)[\omega_t] = \operatorname{cl}\bigcup\{R_{t,t+1}(-Z)[\omega_t] : \forall \omega_{t+1} \in \operatorname{succ}(\omega_t) : Z(\omega_{t+1}) \in \bar{R}^{\epsilon}_{t+1}(X)[\omega_{t+1}]\}$$

  is an ϵ-approximation of $\bar{R}_t(X)[\omega_t]$!
- If $\bar{R}^{\epsilon,\gamma}_t(X)[\omega_t]$ is a γ-approximation of $\bar{R}^{\epsilon}_t(X)[\omega_t]$, then $\bar{R}^{\epsilon,\gamma}_t(X)[\omega_t]$ is an $(\epsilon+\gamma)$-approximation of $\bar{R}_t(X)[\omega_t]$!

- Convex vector optimization
- $R_{t,t+1}(-Z)[\omega_t] = \{P_t(z) + M_{t,+} : g(Z,z) \leq 0\}$
- To calculate: a polyhedral δ-approximation of $\bar{R}_t$ by recursively calculating backwards in time
- $\dim(M)$-dimensional problem with $d \times |\operatorname{succ}(\omega_t)| + |z|$-dimensional pre-image space
- Convex extension of Benson's algorithm can be applied directly with a given approximation error desired

4.1 Market Extension: Idea and Interpretation

- Want to consider market trading in calculation
- Convex transaction costs at time t: closed convex set $\mathbb{R}^d_+ \subseteq C_t[\omega] \subseteq \mathbb{R}^d$ (solvency region), positions transferable into non-negative portfolios

4.1 Market Extension: Idea and Interpretation

- Want to consider market trading in calculation
- Convex transaction costs at time t: closed convex set $\mathbb{R}^d_+ \subseteq C_t[\omega] \subseteq \mathbb{R}^d$ (solvency region), positions transferable into non-negative portfolios
- Idea 1: multi-portfolio time consistent version of the market-compatible version

4.1 Market Extension: Idea and Interpretation

- Want to consider market trading in calculation
- Convex transaction costs at time t: closed convex set $\mathbb{R}^d_+ \subseteq C_t[\omega] \subseteq \mathbb{R}^d$ (solvency region), positions transferable into non-negative portfolios
- Idea 1: multi-portfolio time consistent version of the market-compatible version
- Problem: result is not necessarily market-compatible! (works if $M = \mathbb{R}^d$)

4.1 Market Extension: Idea and Interpretation

- Want to consider market trading in calculation
- Convex transaction costs at time t: closed convex set $\mathbb{R}^d_+ \subseteq C_t[\omega] \subseteq \mathbb{R}^d$ (solvency region), positions transferable into non-negative portfolios
- Idea 1: multi-portfolio time consistent version of the market-compatible version
- Problem: result is not necessarily market-compatible! (works if $M = \mathbb{R}^d$)
- Idea 2: market-compatible version of multi-portfolio time consistent version

- Want to consider market trading in calculation
- Convex transaction costs at time t: closed convex set $\mathbb{R}^d_+ \subseteq C_t[\omega] \subseteq \mathbb{R}^d$ (solvency region), positions transferable into non-negative portfolios
- Idea 1: multi-portfolio time consistent version of the market-compatible version
- Problem: result is not necessarily market-compatible! (works if $M = \mathbb{R}^d$)
- Idea 2: market-compatible version of multi-portfolio time consistent version
- Problem: result is not necessarily multi-portfolio time consistent! (works if $M = \mathbb{R}^d$)

4.1 Market Extension: Idea and Interpretation

- Use idea 2: market-compatible version of multi-portfolio time consistent version

$$\tilde{R}_t(X) = \bigcup_{c \in C_t} \bigcup_{Z \in \tilde{R}_{t+1}(X-c)} R_{t,t+1}(-Z)$$

- Satisfies a modified multi-portfolio time consistency

4.1 Market Extension: Idea and Interpretation

- Use idea 2: market-compatible version of multi-portfolio time consistent version

$$\tilde{R}_t(X) = \bigcup_{c \in C_t} \bigcup_{Z \in \tilde{R}_{t+1}(X-c)} R_{t,t+1}(-Z)$$

- Satisfies a modified multi-portfolio time consistency
- Let $\mathbf{K}_t : L_d^\infty(\mathcal{F}_T) \to \mathcal{P}(L_d^\infty(\mathcal{F}_T))$ such that $\mathbf{K}_t(X)$ are the portfolios that can be reached by trading (including trading constraints)
- Modified multi-portfolio time consistency:

$$\tilde{R}_{t+1}(\mathbf{K}_t(X)) \subseteq \tilde{R}_{t+1}(\mathbf{K}_t(\mathbf{Y})) \Rightarrow \tilde{R}_t(X) \subseteq \tilde{R}_t(\mathbf{Y})$$

4.1 Market Extension: Idea and Interpretation

- Interpretation: the risk measure provides capital injections through time

4.1 Market Extension: Idea and Interpretation

- Interpretation: the risk measure provides capital injections through time
- $Z_0 \in \tilde{R}_0(X)$ if and only if there exists $(Z_t, \hat{c}_t)_{t=0}^T$ so that $Z_T \in R_T(X - \sum_{s=0}^T \hat{c}_s)$ and $Z_t \in R_t(-Z_{t+1})$

- Interpretation: the risk measure provides capital injections through time
- $Z_0 \in \tilde{R}_0(X)$ if and only if there exists $(Z_t, \hat{c}_t)_{t=0}^T$ so that $Z_T \in R_T(X - \sum_{s=0}^T \hat{c}_s)$ and $Z_t \in R_t(-Z_{t+1})$
- Equivalently stated: $u_0 \in \tilde{R}_0(X)$ if and only if there exists a sequence $(u_t, \hat{c}_t)_{t=0}^T$ so that

$$u_t \in \tilde{R}_t(X + \sum_{s=0}^{t-1}(u_s - \hat{c}_s) - \hat{c}_t)$$

with $u_t \in R_t(-u_{t+1})$ and $u_T \in R_T(X + \sum_{s=0}^{T-1}(u_s - \hat{c}_s) - \hat{c}_t)$.

4.2 Market Extension: Computation

- Higher dimensional problem because of modified multi-portfolio time consistency
- Need to find (a part of) the graph [instead of just 1 value in the graph]

4.2 Market Extension: Computation

- Higher dimensional problem because of modified multi-portfolio time consistency
- Need to find (a part of) the graph [instead of just 1 value in the graph]

$$\left\{(k,u) \in \mathbb{R}^d \times M : u \in \tilde{R}_t(X-k)[\omega_t]\right\}$$

4.2 Market Extension: Computation

- Higher dimensional problem because of modified multi-portfolio time consistency
- Need to find (a part of) the graph [instead of just 1 value in the graph]

$$\left\{(k, u) \in \mathbb{R}^d \times M : u \in \tilde{R}_t(X - k)[\omega_t]\right\}$$

- Problem is $d + |\dim(M)|$ dimensional

- Higher dimensional problem because of modified multi-portfolio time consistency
- Need to find (a part of) the graph [instead of just 1 value in the graph]

$$\left\{(k,u) \in \mathbb{R}^d \times M : u \in \tilde{R}_t(X-k)[\omega_t]\right\}$$

- Problem is $d + |\dim(M)|$ dimensional
- If $M = \mathbb{R}^d$ then idea 1 is the same as idea 2, and problem is reduced to d-dimensional

Thank you

Harold Benson.
An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem.
Journal of Global Optimization, 13(1):1–24, 1998.

Zachary Feinstein and Birgit Rudloff.
A recursive algorithm for multivariate risk measures and a set-valued bellman's principle.

Zachary Feinstein and Birgit Rudloff.
Time consistency of dynamic risk measures in markets with transaction costs.
Quantitative Finance, 13(9):1473–1489, 2013.

Andreas Löhne, Birgit Rudloff, and Firdevs Ulus.
Primal and dual approximation algorithms for convex vector optimization problems.
ArXiv e-prints, September 2013.