Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \ \geq \ b$$

- μ = vector of "returns", Q = "covariance" matrix
- x = vector of "asset weights"
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0$ = "risk-aversion" multiplier

Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \geq b$$

- $\mu =$ vector of “returns”, $Q =$ “covariance” matrix
- $x =$ vector of “asset weights”
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0 =$ “risk-aversion” multiplier

Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \geq b$$

- μ = vector of “returns”, Q = “covariance” matrix
- x = vector of “asset weights”
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0$ = “risk-aversion” multiplier

Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \geq b$$

- μ = vector of “returns”, Q = “covariance” matrix
- x = vector of “asset weights”
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0$ = “risk-aversion” multiplier

Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \geq b$$

- $\mu =$ vector of "returns", $Q =$ "covariance" matrix
- $x =$ vector of "asset weights"
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0 =$ "risk-aversion" multiplier

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Bertsimas et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient a_i is uncertain. We allow $a_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Bertsimas et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient a_i is uncertain. We allow $a_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Bertsimas et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient a_i is uncertain. We allow $a_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Bertsimas et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient a_i is uncertain. We allow $a_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Bertsimas et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient a_i is uncertain. We allow $a_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

Robust Optimization

- Optimization under parameter (data) uncertainty
- Ben-Tal and Nemirovsky, El Ghaoui et al
- Bertsimas et al
- Uncertainty is modeled by assuming that data is not known precisely, and will instead lie in known sets.
- Example: a coefficient a_i is uncertain. We allow $a_i \in [l_i, u_i]$.
- Typically, a **minimization** problem becomes a **min-max** problem.

What is the most common word in robust optimization literature?

"Tractable"

convex uncertainty models $\longrightarrow$ convex optimization techniques

$\longrightarrow$ polynomial-time algorithms

$\longrightarrow$ sacrifice model richness in favor of theoretical algorithm efficiency

$\longrightarrow$ in practice, SOCP not quite so "tractable"

What is the most common word in robust optimization literature?

"Tractable"

convex uncertainty models → convex optimization techniques

→ polynomial-time algorithms

→ sacrifice model richness in favor of theoretical algorithm efficiency

→ in practice, SOCP not quite so "tractable"

What is the most common word in robust optimization literature?

“Tractable”

convex uncertainty models $\longrightarrow$ convex optimization techniques

$\longrightarrow$ polynomial-time algorithms

$\longrightarrow$ sacrifice model richness in favor of theoretical algorithm efficiency

$\longrightarrow$ in practice, SOCP not quite so “tractable”

What is the most common word in robust optimization literature?

"Tractable"

convex uncertainty models $\longrightarrow$ convex optimization techniques

$\longrightarrow$ polynomial-time algorithms

$\longrightarrow$ sacrifice model richness in favor of theoretical algorithm efficiency

$\longrightarrow$ in practice, SOCP not quite so "tractable"

What is the most common word in robust optimization literature?

“Tractable”

convex uncertainty models $\longrightarrow$ convex optimization techniques

$\longrightarrow$ polynomial-time algorithms

$\longrightarrow$ sacrifice model richness in favor of theoretical algorithm efficiency

$\longrightarrow$ in practice, SOCP not quite so “tractable”

What is the most common word in robust optimization literature?

“Tractable”

convex uncertainty models $\longrightarrow$ convex optimization techniques

$\longrightarrow$ polynomial-time algorithms

$\longrightarrow$ sacrifice model richness in favor of theoretical algorithm efficiency

$\longrightarrow$ in practice, SOCP not quite so “tractable”

Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \geq b$$

- μ = vector of "returns", Q = "covariance" matrix
- x = vector of "asset weights"
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0$ = "risk-aversion" multiplier

$\longrightarrow$ want to model uncertainty in μ

Mean-Variance Portfolio Optimization

- $$\min \lambda x^T Q x \; - \; \mu^T x$$

  Subject to:

  $$Ax \geq b$$
- $\mu =$ vector of "returns", $Q =$ "covariance" matrix
- $x =$ vector of "asset weights"
- $Ax \geq b$: general linear constraints
- $\lambda \geq 0 =$ "risk-aversion" multiplier

$\longrightarrow$ want to model uncertainty in μ

The histogram model

- Parameters: $\mathbf{0} \le \gamma_1 \le \gamma_2 \le \ldots \le \gamma_K \le \mathbf{1}$,
  integers $\mathbf{0} \le n_i \le N_i$, $\mathbf{1} \le i \le K$
  for each asset j: $\bar{\mu}_j =$ expected return

- between n_i and N_i assets j satisfy:
  $(1 - \gamma_i)\bar{\mu}_j \le \mu_j \le (1 - \gamma_{i-1})\bar{\mu}_j$

- $\sum_j \mu_j \ge \Gamma \sum_j \bar{\mu}_j$; $\Gamma > 0$ a parameter

- (R. Tütüncü) For $1 \le h \le H$,
  - a **set** (“tier”) T_h of assets, and a parameter $\Gamma_h > 0$

  for each h, $\sum_{j \in T_h} \mu_j \ge \Gamma_h \sum_{j \in S_h} \bar{\mu}_j$

The histogram model

- Parameters: $\mathbf{0} \le \gamma_1 \le \gamma_2 \le \ldots \le \gamma_K \le \mathbf{1}$, integers $\mathbf{0} \le n_i \le N_i$, $\mathbf{1} \le i \le K$ for each asset j: $\bar{\mu}_j$ = expected return
- between n_i and N_i assets j satisfy: $(\mathbf{1} - \gamma_i)\bar{\mu}_j \le \mu_j \le (\mathbf{1} - \gamma_{i-1})\bar{\mu}_j$
- $\sum_j \mu_j \ge \Gamma \sum_j \bar{\mu}_j$; $\Gamma > \mathbf{0}$ a parameter
- (R. Tütüncü) For $\mathbf{1} \le h \le H$,
  - a **set** ("tier") T_h of assets, and a parameter $\Gamma_h > 0$

for each h, $\sum_{j \in T_h} \mu_j \ge \Gamma_h \sum_{j \in S_h} \bar{\mu}_j$

The histogram model

- Parameters: $0 \leq \gamma_1 \leq \gamma_2 \leq \ldots \leq \gamma_K \leq 1$, integers $0 \leq n_i \leq N_i$, $1 \leq i \leq K$ for each asset j: $\bar{\mu}_j =$ expected return
- between n_i and N_i assets j satisfy: $(1 - \gamma_i)\bar{\mu}_j \leq \mu_j \leq (1 - \gamma_{i-1})\bar{\mu}_j$
- $\sum_j \mu_j \geq \Gamma \sum_j \bar{\mu}_j$; $\Gamma > 0$ a parameter
- (R. Tütüncü) For $1 \leq h \leq H$,
  - a **set** ("tier") T_h of assets, and a parameter $\Gamma_h > 0$

  for each h, $\sum_{j \in T_h} \mu_j \geq \Gamma_h \sum_{j \in S_h} \bar{\mu}_j$

The histogram model

- Parameters: $\mathbf{0} \leq \gamma_{\mathbf{1}} \leq \gamma_{\mathbf{2}} \leq \ldots \leq \gamma_{\boldsymbol{K}} \leq \mathbf{1}$,
  integers $\mathbf{0} \leq \boldsymbol{n_i} \leq \boldsymbol{N_i}$, $\mathbf{1} \leq \boldsymbol{i} \leq \boldsymbol{K}$
  for each asset j: $\bar{\mu}_j = \text{expected return}$

- between $\boldsymbol{n_i}$ and $\boldsymbol{N_i}$ assets j satisfy:
  $(\mathbf{1} - \gamma_i)\bar{\mu}_j \leq \mu_j \leq (\mathbf{1} - \gamma_{i-1})\bar{\mu}_j$

- $\sum_j \mu_j \geq \boldsymbol{\Gamma} \sum_j \bar{\mu}_j$; $\boldsymbol{\Gamma} > \mathbf{0}$ a parameter

- (R. Tütüncü) For $\mathbf{1} \leq \boldsymbol{h} \leq \boldsymbol{H}$,
  - a **set** ("tier") $\boldsymbol{T_h}$ of assets, and a parameter $\Gamma_h > 0$

for each $\boldsymbol{h}$, $\sum_{j \in T_h} \mu_j \geq \boldsymbol{\Gamma_h} \sum_{j \in S_h} \bar{\mu}_j$

General methodology:

Benders' decomposition (= cutting-plane algorithm)

Generic problem: $\min_{x \in X} \max_{d \in \mathcal{D}} f(x, d)$

→ Maintain a **finite subset** $\tilde{D}$ of $\mathcal{D}$ (a **"model"**)

GAME

1. **Implementor:** solve $\min_{x \in X} \max_{d \in \tilde{D}} f(x, d)$, with solution x^*
2. **Adversary:** solve $\max_{d \in \mathcal{D}} f(x^*, d)$, with solution $\tilde{d}$
3. **Add** $\tilde{d}$ to $\tilde{D}$, and go to 1.

General methodology:

Benders' decomposition (= cutting-plane algorithm)

Generic problem: $\min_{x \in X} \max_{d \in \mathcal{D}} f(x, d)$

→ Maintain a **finite subset** $\tilde{D}$ of $\mathcal{D}$ (a **"model"**)

GAME

1. **Implementor:** solve $\min_{x \in X} \max_{d \in \tilde{D}} f(x, d)$, with solution x^*
2. **Adversary:** solve $\max_{d \in \mathcal{D}} f(x^*, d)$, with solution $\tilde{d}$
3. **Add** $\tilde{d}$ to $\tilde{D}$, and go to 1.

General methodology:

Benders' decomposition (= cutting-plane algorithm)

Generic problem: $\min_{x \in X} \max_{d \in \mathcal{D}} f(x,d)$

$\longrightarrow$ Maintain a **finite subset** $\tilde{D}$ of $\mathcal{D}$ (a **"model"**)

GAME

1. **Implementor:** solve $\min_{x \in X} \max_{d \in \tilde{D}} f(x,d)$, with solution x^*
2. **Adversary:** solve $\max_{d \in \mathcal{D}} f(x^*,d)$, with solution $\tilde{d}$
3. **Add** $\tilde{d}$ to $\tilde{D}$, and go to 1.

General methodology:

Benders' decomposition (= cutting-plane algorithm)

Generic problem: $\min_{x \in X} \max_{d \in \mathcal{D}} f(x, d)$

$\longrightarrow$ Maintain a **finite subset** $\tilde{D}$ of $\mathcal{D}$ (a **"model"**)

GAME

1. **Implementor:** solve $\min_{x \in X} \max_{d \in \tilde{D}} f(x, d)$, with solution x^*
2. **Adversary:** solve $\max_{d \in \mathcal{D}} f(x^*, d)$, with solution $\tilde{d}$
3. Add $\tilde{d}$ to $\tilde{D}$, and go to 1.

General methodology:

Benders' decomposition (= cutting-plane algorithm)

Generic problem: $\min_{x \in X} \max_{d \in \mathcal{D}} f(x, d)$

$\longrightarrow$ Maintain a **finite subset** $\tilde{D}$ of $\mathcal{D}$ (a **"model"**)

GAME

1. **Implementor:** solve $\min_{x \in X} \max_{d \in \tilde{D}} f(x, d)$, with solution x^*
2. **Adversary:** solve $\max_{d \in \mathcal{D}} f(x^*, d)$, with solution $\tilde{d}$
3. **Add** $\tilde{d}$ to $\tilde{D}$, and go to 1.

Why this approach

- Decoupling of implementor and adversary yields considerably simpler, and smaller, problems
- Decoupling allows us to use more sophisticated uncertainty models
- If number of iterations is small, implementor's problem is a small "convex" problem
- Most progress will be achieved in initial iterations – permits "soft" termination criteria

Why this approach

- Decoupling of implementor and adversary yields considerably simpler, and smaller, problems
- Decoupling allows us to use more sophisticated uncertainty models
- If number of iterations is small, implementor's problem is a small "convex" problem
- Most progress will be achieved in initial iterations – permits "soft" termination criteria

Why this approach

- Decoupling of implementor and adversary yields considerably simpler, and smaller, problems
- Decoupling allows us to use more sophisticated uncertainty models
- If number of iterations is small, implementor's problem is a small "convex" problem
- Most progress will be achieved in initial iterations – permits "soft" termination criteria

Why this approach

- Decoupling of implementor and adversary yields considerably simpler, and smaller, problems
- Decoupling allows us to use more sophisticated uncertainty models
- If number of iterations is small, implementor's problem is a small "convex" problem
- Most progress will be achieved in initial iterations – permits "soft" termination criteria

Why this approach

- Decoupling of implementor and adversary yields considerably simpler, and smaller, problems
- Decoupling allows us to use more sophisticated uncertainty models
- If number of iterations is small, implementor's problem is a small "convex" problem
- Most progress will be achieved in initial iterations – permits "soft" termination criteria

Implementor's problem

A convex quadratic program

At iteration m, solve

$$\min \ \lambda x^T Q x \ - \ r$$

Subject to:

$$Ax \geq b$$

$$r \leq \mu_{(i)}^T x, \ \ i = 1, \ldots, m$$

Here, $\mu_{(1)}, \ldots, \mu_{(m)}$ are given return vectors

Adversarial problem: A mixed-integer program

$x^* =$ given asset weights

$$\min \ \sum_j x_j^* \mu_j$$

Subject to:

$$\bar{\mu}_j (1 - \sum_i \gamma_{i-1} y_{ij}) \leq \mu_j \leq \bar{\mu}_j (1 - \sum_i \gamma_i y_{ij}) \quad \forall i \geq 1$$

$\sum_i y_{ij} \leq 1, \ \forall j$ (each asset in at most one segment)

$n_i \leq \sum_j y_{ij} \leq N_i, \ 1 \leq i \leq K$ (segment cardinalities)

$\sum_{j \in T_h} \mu_j \geq \Gamma_h \sum_{j \in T_h} \bar{\mu}_j, \ 1 \leq h \leq H$ (tier ineqs.)

μ_j free, $y_{ij} = 0$ or 1, all i, j

Why the adversarial problem is “easy”

(K = no. of segments, H = no. of tiers)

Theorem. For every fixed K and H, and for every $\epsilon > 0$, there is an algorithm that finds a solution to the adversarial problem with optimality relative error $\leq \epsilon$, in time polynomial in ϵ^{-1} and n (= no. of assets).

The simplest case

$$\max \sum_j x_j^* \delta_j$$

Subject to:

$$\sum_j \delta_j \leq \Gamma$$

$$0 \leq \delta_j \leq u_j y_j, \quad y_j = 0 \text{ or } 1, \quad \text{all j}$$

$$\sum_j y_j \leq N$$

$\cdots$ a *cardinality constrained knapsack problem*

B. (1995), DeFarias and Nemhauser (2004)

The LP relaxation $x^* =$ given asset weights

should (?) be tight

$$\min \sum_j x_j^* \mu_j$$

Subject to:

$$\bar{\mu}_j (1 - \sum_i \gamma_{i-1} y_{ij}) \le \mu_j \le \bar{\mu}_j (1 - \sum_i \gamma_i y_{ij}) \quad \forall i \ge 1$$

$\sum_i y_{ij} \le 1, \quad \forall j$ (each asset in at most one segment)

$n_i \le \sum_j y_{ij} \le N_i, \quad 1 \le i \le K$ (segment cardinalities)

$\sum_{j \in T_h} \mu_j \ge \Gamma_h \sum_{j \in T_h} \bar{\mu}_j, \quad 1 \le h \le H$ (tier ineqs.)

μ_j free, $0 \le y_{ij} \le 1$, all i, j

Robust problem:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$
Subject to: $Ax \geq b$

$r \leq \mu^T x, \ \forall \ \mu$ achievable by adversary

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$
Subject to: $Ax \geq b$

$r \leq \mu^T x, \ \forall \ \mu$ achievable by **relaxed** adversary

$V^* \geq V$, perhaps $V^* \approx V$,

Robust problem:

$$V \doteq \min \ \lambda x^T Qx \ - \ r$$
Subject to: $\quad Ax \geq b$
$r \ \leq \ \mu^T x, \ \ \forall \ \mu$ achievable by adversary

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Qx \ - \ r$$
Subject to: $\quad Ax \geq b$
$r \ \leq \ \mu^T x, \ \ \forall \ \mu$ achievable by **relaxed** adversary

$V^* \geq V$, perhaps $V^* \approx V$,

Robust problem:

$$V \doteq \min \; \lambda x^T Q x \; - \; r$$

Subject to: $Ax \geq b$

$r \; \leq \; \mu^T x, \;\; \forall \; \mu$ achievable by adversary

Robust problem for relaxed adversary:

$$V \doteq \min \; \lambda x^T Q x \; - \; r$$

Subject to: $Ax \geq b$

$r \; \leq \; \mu^T x, \;\; \forall \; \mu$ achievable by **relaxed** adversary

$V^* \geq V,$ perhaps $V^* \approx V,$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$

Subject to: $Ax \geq b$

$r \leq \mu^T x, \ \forall \ \mu$ achievable by relaxed adversary

or, $V \doteq \min \ \lambda x^T Q x \ - \ r$

Subject to: $Ax \geq b$

$r \leq$ minimum return(x)

but, minimum return(x) = **min** $\sum_j x_j^* \mu_j$

Subject to: $M_1 \mu + M_2 y \geq \Psi$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Qx - r$$

Subject to: $Ax \geq b$

$r \leq \mu^T x, \ \forall \ \mu$ achievable by relaxed adversary

or, $V \doteq \min \ \lambda x^T Qx - r$

Subject to: $Ax \geq b$

$r \leq$ minimum return(x)

but, minimum return(x) = $\min \sum_j x_j^* \mu_j$

Subject to: $M_1 \mu + M_2 y \geq \Psi$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$

Subject to: $Ax \geq b$

$r \ \leq \ \mu^T x, \ \ \forall \ \mu$ achievable by relaxed adversary

or, $V \doteq \min \ \lambda x^T Q x \ - \ r$

Subject to: $Ax \geq b$

$r \ \leq$ minimum return(x)

but, minimum return(x) = **min** $\sum_j x_j^* \mu_j$

Subject to: $M_1 \mu \ + \ M_2 y \ \geq \ \Psi$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$
Subject to: $Ax \geq b$
$r \ \leq \ \mu^T x, \ \ \forall \ \mu$ achievable by relaxed adversary

or, $$V \doteq \min \ \lambda x^T Q x \ - \ r$$
Subject to: $Ax \geq b$
$r \ \leq$ minimum return(x)

duality: minimum return(x) = $\mathbf{max} \ \ \Psi^T \alpha$

Subject to: $M_1^T \alpha \ = \ x, \ \ M_2^T \alpha \ = \ 0, \ \ \alpha \geq 0$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$
Subject to: $Ax \geq b$
$r \leq$ minimum return(x)

Robust problem:

$$\min \ \lambda x^T Q x \ - \ r$$
Subject to:
$Ax \geq b$

$$r \ - \ \Psi^T \alpha \ \leq 0$$

$$M_1^T \alpha \ - \ x \ = \ 0, \ \ M_2^T \alpha \ = \ 0, \ \ \alpha \geq 0$$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$
Subject to: $\quad Ax \geq b$

$r \ \leq$ minimum return(x)

Robust problem:

$$\min \ \lambda x^T Q x \ - \ r$$
Subject to:

$Ax \geq b$

$r - \Psi^T \alpha \leq 0$

$M_1^T \alpha - x = 0, \ M_2^T \alpha = 0, \ \alpha \geq 0$

Robust problem for relaxed adversary:

$$V \doteq \min \ \lambda x^T Q x \ - \ r$$

Subject to: $Ax \geq b$

$r \leq$ minimum return(x)

Robust problem:

$$\min \ \lambda x^T Q x \ - \ r$$

Subject to:

$$Ax \geq b$$

$$r \ - \ \Psi^T \alpha \ \leq 0$$

$$M_1^T \alpha \ - \ x \ = \ 0, \ \ M_2^T \alpha \ = \ 0, \ \ \alpha \geq 0$$

Robust problem for relaxed adversary:

$$V^* \doteq \min \ \lambda x^T Q x \ - \ r$$
$$\text{Subject to:} \quad Ax \geq b$$
$$r \ - \ \Psi^T \alpha \ \leq 0$$
$$(**) \ M_1^T \alpha \ - \ x \ = \ 0$$
$$M_2^T \alpha \ = \ 0, \quad \alpha \geq 0$$

Let $\hat{\mu}$ = optimal duals for (**)

$$V^* \ = \ \min \ \lambda x^T Q x \ - \ r$$
$$\text{Subject to:} \quad Ax \geq b$$
$$r \ - \ \hat{\mu}^T x \ \leq 0$$

($r \ - \ \mu^T x \ \leq 0, \ \ \forall \, \mu$ available to strict adversary)

Problem: Find μ available to strict adversary, and with $\mu \approx \hat{\mu}$

Robust problem for relaxed adversary:

$$
\begin{aligned}
& V^* \doteq \min \ \lambda x^T Q x \ - \ r \\
& \text{Subject to:} \quad Ax \geq b \\
& r \ - \ \Psi^T \alpha \ \leq 0 \\
(**) \ & M_1^T \alpha \ - \ x \ = \ 0 \\
& M_2^T \alpha \ = \ 0, \quad \alpha \geq 0
\end{aligned}
$$

Let $\hat{\mu} =$ optimal duals for (**)

$$
\begin{aligned}
& V^* \ = \min \ \lambda x^T Q x \ - \ r \\
& \text{Subject to:} \quad Ax \geq b \\
& r \ - \ \hat{\mu}^T x \ \leq 0
\end{aligned}
$$

($r \ - \ \mu^T x \ \leq 0, \ \ \forall \mu$ available to strict adversary)

Problem: Find μ available to strict adversary, and with $\mu \approx \hat{\mu}$

Robust problem for relaxed adversary:

$$V^* \doteq \min \ \lambda x^T Q x - r$$
Subject to: $Ax \geq b$
$$r - \Psi^T \alpha \leq 0$$
$$(**) \ M_1^T \alpha - x = 0$$
$$M_2^T \alpha = 0, \quad \alpha \geq 0$$

Let $\hat{\mu} =$ optimal duals for (**)

$$V^* = \min \ \lambda x^T Q x - r$$
Subject to: $Ax \geq b$
$$r - \hat{\mu}^T x \leq 0$$

$(r - \mu^T x \leq 0, \ \forall \mu$ available to strict adversary)

Problem: Find μ available to strict adversary, and with $\mu \approx \hat{\mu}$

Robust problem for relaxed adversary:

$$V^* \doteq \min \ \lambda x^T Q x - r$$
Subject to: $Ax \geq b$
$$r - \Psi^T \alpha \leq 0$$
$$(**) \ M_1^T \alpha - x = 0$$
$$M_2^T \alpha = 0, \quad \alpha \geq 0$$

Let $\hat{\mu} =$ optimal duals for (**)

$$V^* = \min \ \lambda x^T Q x - r$$
Subject to: $Ax \geq b$
$$r - \hat{\mu}^T x \leq 0$$

($r - \mu^T x \leq 0, \ \forall \mu$ available to strict adversary)

Problem: Find μ available to strict adversary, and with $\mu \approx \hat{\mu}$

Robust problem for relaxed adversary:

$$V^* \doteq \min \ \lambda x^T Q x - r$$
$$\text{Subject to:} \quad Ax \geq b$$
$$r - \Psi^T \alpha \leq 0$$
$$(**) \ M_1^T \alpha - x = 0$$
$$M_2^T \alpha = 0, \quad \alpha \geq 0$$

Let $\hat{\mu} =$ optimal duals for (**)

$$V^* = \min \ \lambda x^T Q x - r$$
$$\text{Subject to:} \quad Ax \geq b$$
$$r - \hat{\mu}^T x \leq 0$$

($r - \mu^T x \leq 0, \ \forall \mu$ available to strict adversary)

Problem: Find μ available to strict adversary, and with $\mu \approx \hat{\mu}$

Benders' algorithm with strengthening

Step 1. Solve relaxed robust problem; answer = $\hat{\mu}$

Step 2. Solve MIP to obtain vector $\breve{\mu}$ which is legal for histogram model, and with $\breve{\mu} \approx \hat{\mu}$

Step 3. Run Benders beginning with the cut $r - \breve{\mu}^T x \leq 0$

Alternate algorithm?

Step 1. Solve relaxed robust problem, let $\hat{\mu}$ be the min-max return vector

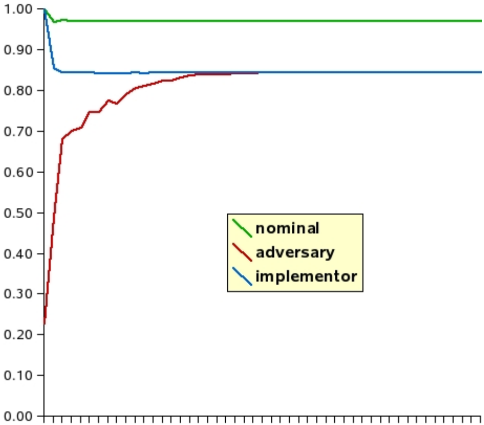
Step 2. Find a cut $\alpha^T \mu \leq \alpha_0$, that separates $\hat{\mu}$ from the convex hull of vectors available to the strict adversary

Step 3. Add $\alpha^T \hat{\mu} \leq \alpha_0$ to the definition of the adversarial problem, and go to 1.

Example: 2464 assets, 152-factor model. CPU time: 300 seconds No Strengthening – straight Benders

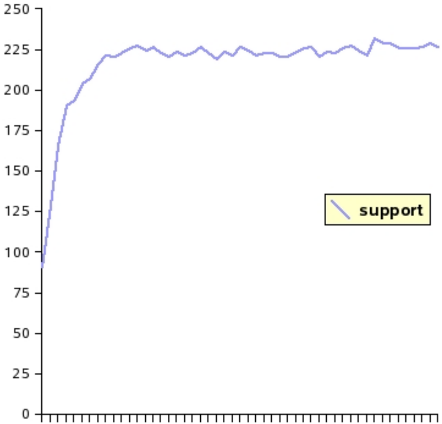
10 segments (a: “heavy tail”)

6 tiers: the top five deciles lose at most **10%** each, total loss $\leq$ **5%**



Same run

2464 assets, 152 factors;
10 segments, 6 tiers



Summary of average problems with 3-4 segments, 2-3 tiers

	columns	rows	**iterations**	**time (sec.)**	**imp. time**	**adv. time**
1	500	20	47	1.85	1.34	0.46
2	500	20	3	0.09	0.01	0.03
3	703	108	1	0.29	0.13	0.04
4	499	140	3	3.12	2.65	0.05
5	499	20	19	0.42	0.21	0.17
6	1338	81	7	0.45	0.17	0.08
7	2019	140	8	41.53	39.6	0.36
8	2443	153	2	12.32	9.91	0.07
9	2464	153	111	100.81	60.93	36.78

	time	bigQP	bigMIP	iters	impT	advT	01vars
A	327.04	2.52	211.72	135	12.27	100.24	5000
C	29.32	3.01	9.35	27	1.02	15.76	4990
F	74.06	13.57	15.96	27	2.47	41.42	13380
G *	681.12	–	–	19	64.7	615.54	20190
I	124.82	93.38	22.58	1	4.17	2.46	24640

Table: Heavy-tailed instances, 10 segments, 6 tiers, tol. = $1.0e^{-03}$

error	500 × 20	500 × 20	499 × 20	499 [b] × 140	703 * × 108	1338 × 81	2443 × 153
$5.0e^{-2}$	214.53	14.81	144.86	122.53	11.77	274.64	140.29
$1.0e^{-2}$	223.21	15.49	144.86	122.53	14.66	356.98	140.29
$5.0e^{-3}$	254.73	16.03	162.41	126.63	34.16	363.84	140.29
$1.0e^{-3}$	300.88	35.23	183.12	157.49	64.61	469.75	140.29
$5.0e^{-4}$	361.20	37.92	216.52	167.40	73.87	598.94	140.29

Table: Convergence time on heavy-tailed instances, 10 segments, 6 tiers

What is the impact of the uncertainty model

All runs on the same data set with 1338 columns and 81 rows

- 1 segment: $(200, 0.5)$
  robust random return = **4.57**, **157** assets
- 2 segments: $(200, 0.25)$, $(100, 0.5)$
  robust random return = **4.57**, **186** assets
- 2 segments: $(200, 0.2)$, $(100, 0.6)$
  robust random return = **3.25**, **213** assets
- 2 segments: $(200, 0.1)$, $(100, 0.8)$
  robust random return = **1.50**, **256** assets
- 1 segment: $(100, 1.0)$
  robust random return = **1.24**, **281** assets

Ambiguous chance-constrained models

1. The implementor chooses a vector x^* of assets
2. The adversary chooses a *probability distribution* P for the returns vector
3. A random returns vector μ is drawn from P

→ Implementor wants to choose x^* so as to minimize **value-at-risk** (conditional value at risk, etc.)

Erdogan and Iyengar (2004), Calafiore and Campi (2004)

→ We want to model *correlated* errors in the returns

Ambiguous chance-constrained models

1. The implementor chooses a vector x^* of assets
2. The adversary chooses a *probability distribution* P for the returns vector
3. A random returns vector μ is drawn from P

$\longrightarrow$ Implementor wants to choose x^* so as to minimize **value-at-risk** (conditional value at risk, etc.)

Erdogan and Iyengar (2004), Calafiore and Campi (2004)

$\longrightarrow$ We want to model *correlated* errors in the returns

Ambiguous chance-constrained models

1. The implementor chooses a vector x^* of assets
2. The adversary chooses a *probability distribution* P for the returns vector
3. A random returns vector μ is drawn from P

$\longrightarrow$ Implementor wants to choose x^* so as to minimize **value-at-risk** (conditional value at risk, etc.)

Erdogan and Iyengar (2004), Calafiore and Campi (2004)

$\longrightarrow$ We want to model *correlated* errors in the returns

Ambiguous chance-constrained models

1. The implementor chooses a vector x^* of assets
2. The adversary chooses a *probability distribution* P for the returns vector
3. A random returns vector μ is drawn from P

$\longrightarrow$ Implementor wants to choose x^* so as to minimize **value-at-risk** (conditional value at risk, etc.)

Erdogan and Iyengar (2004), Calafiore and Campi (2004)

$\longrightarrow$ We want to model *correlated* errors in the returns

Ambiguous chance-constrained models

1. The implementor chooses a vector x^* of assets
2. The adversary chooses a *probability distribution* P for the returns vector
3. A random returns vector μ is drawn from P

$\longrightarrow$ Implementor wants to choose x^* so as to minimize **value-at-risk** (conditional value at risk, etc.)

Erdogan and Iyengar (2004), Calafiore and Campi (2004)

$\longrightarrow$ We want to model *correlated* errors in the returns

Ambiguous chance-constrained models

1. The implementor chooses a vector x^* of assets
2. The adversary chooses a *probability distribution* P for the returns vector
3. A random returns vector μ is drawn from P

$\longrightarrow$ Implementor wants to choose x^* so as to minimize **value-at-risk** (conditional value at risk, etc.)

Erdogan and Iyengar (2004), Calafiore and Campi (2004)

$\longrightarrow$ We want to model *correlated* errors in the returns

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

→ Random return: $\mu_j = \bar{\mu}_j (1 - \delta w_j)$ ($\bar{\mu}$ = nominal returns).

Definition (Rockafellar and Uryasev): Given reals ν and $0 \leq \theta \leq 1$ the *value-at-risk* of x^* is the real $\rho \geq 0$ such that

$$\text{Prob}(\nu - \mu^T x^* \geq \rho) \geq \theta$$

→ The adversary wants to maximize VaR

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

→ Random return: $\mu_j = \bar{\mu}_j (1 - \delta w_j)$ ($\bar{\mu}$ = nominal returns).

Definition (Rockafellar and Uryasev): Given reals ν and $0 \leq \theta \leq 1$ the *value-at-risk* of x^* is the real $\rho \geq 0$ such that

$$\text{Prob}(\nu - \mu^T x^* \geq \rho) \geq \theta$$

→ The adversary wants to maximize VaR

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

→ Random return: $\mu_j = \bar{\mu}_j (1 - \delta w_j)$ ($\bar{\mu}$ = nominal returns).

Definition (Rockafellar and Uryasev): Given reals ν and $0 \leq \theta \leq 1$ the *value-at-risk* of x^* is the real $\rho \geq 0$ such that

$$\text{Prob}(\nu - \mu^T x^* \geq \rho) \geq \theta$$

→ The adversary wants to maximize VaR

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

$\rightarrow$ Random return: $\mu_j = \bar{\mu}_j (1 - \delta w_j)$ ($\bar{\mu}$ = nominal returns).

Definition (Rockafellar and Uryasev): Given reals ν and $0 \leq \theta \leq 1$ the *value-at-risk* of x^* is the real $\rho \geq 0$ such that

$$\text{Prob}(\nu - \mu^T x^* \geq \rho) \geq \theta$$

$\rightarrow$ The adversary wants to maximize VaR

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

$\longrightarrow$ Random return: $\mu_j = \bar{\mu}_j (1 - \delta w_j)$ ($\bar{\mu}$ = nominal returns).

Definition (Rockafellar and Uryasev): Given reals ν and $0 \leq \theta \leq 1$ the *value-at-risk* of x^* is the real $\rho \geq 0$ such that

$$\text{Prob}(\nu - \mu^T x^* \geq \rho) \geq \theta$$

$\longrightarrow$ The adversary wants to maximize VaR

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

→ Random return: $\mu_j = \bar{\mu}_j (1 - \delta w_j)$ ($\bar{\mu}$ = nominal returns).

Definition (Rockafellar and Uryasev): Given reals ν and $0 \leq \theta \leq 1$ the *value-at-risk* of x^* is the real $\rho \geq 0$ such that

$$\text{Prob}(\nu - \mu^T x^* \geq \rho) \geq \theta$$

→ The adversary wants to maximize VaR

Uncertainty set

Given a vector x^* of assets, the adversary

1. Chooses a vector $w \in R^n$ (n = no. of assets) with $0 \leq w_j \leq 1$ for all j.
2. Chooses a random variable $0 \leq \delta \leq 1$

→ Random return: $\mu_j = \bar{\mu}_j\,(1 - \delta w_j)$ ($\bar{\mu} =$ nominal returns).

Definition: Given reals ν and $0 \leq \theta \leq 1$ the *conditional value-at-risk* of x^* equals

$$E(\nu - \mu^T x^* \mid \nu - \mu^T x^* \geq \rho) \quad \text{where } \rho = \text{VaR}$$

→ The adversary wants to maximize CVaR

$\rightarrow$ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \le w_j \le 1 \ \forall j$, and $0 \le \delta \le 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \le \delta_2 \le \ldots \le \delta_K = 1$ example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \text{Prob}(\delta = \delta_i)$, $0 \le i \le K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \le \pi_i \le \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets ("tiers") T_h of assets, $1 \le h \le H$, we require:

  $E(\delta \sum_{j \in T_h} w_j) \le \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \le \Gamma_h$

$\rightarrow$ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \leq w_j \leq 1 \ \forall j$, and $0 \leq \delta \leq 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \leq \delta_2 \leq \ldots \leq \delta_K = 1$ example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \text{Prob}(\delta = \delta_i)$, $0 \leq i \leq K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \leq \pi_i \leq \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets ("tiers") T_h of assets, $1 \leq h \leq H$, we require:

  $E(\delta \sum_{j \in T_h} w_j) \leq \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$

$\rightarrow$ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \le w_j \le 1 \ \forall j$, and $0 \le \delta \le 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \le \delta_2 \le \ldots \le \delta_K = 1$ example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \text{Prob}(\delta = \delta_i)$, $0 \le i \le K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \le \pi_i \le \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets ("tiers") T_h of assets, $1 \le h \le H$, we require: $E(\delta \sum_{j \in T_h} w_j) \le \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \le \Gamma_h$

$\longrightarrow$ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \leq w_j \leq 1 \;\; \forall j$, and $0 \leq \delta \leq 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \leq \delta_2 \leq \ldots \leq \delta_K = 1$ example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \text{Prob}(\delta = \delta_i)$, $0 \leq i \leq K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \leq \pi_i \leq \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets ("tiers") T_h of assets, $1 \leq h \leq H$, we require:
  $E(\delta \sum_{j \in T_h} w_j) \leq \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$

$\rightarrow$ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \leq w_j \leq 1 \;\; \forall j$, and $0 \leq \delta \leq 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \leq \delta_2 \leq \ldots \leq \delta_K = 1$ example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \mathrm{Prob}(\delta = \delta_i)$, $0 \leq i \leq K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \leq \pi_i \leq \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets ("tiers") T_h of assets, $1 \leq h \leq H$, we require:
  $E(\delta \sum_{j \in T_h} w_j) \leq \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$

→ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \leq w_j \leq 1 \ \forall j$, and $0 \leq \delta \leq 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \leq \delta_2 \leq \ldots \leq \delta_K = 1$
  example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \text{Prob}(\delta = \delta_i)$, $0 \leq i \leq K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \leq \pi_i \leq \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets ("tiers") T_h of assets, $1 \leq h \leq H$, we require:
  $E(\delta \sum_{j \in T_h} w_j) \leq \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$

$\rightarrow$ Random return$_j$ $= \bar{\mu}_j(1 - \delta w_j)$ where $0 \leq w_j \leq 1 \ \forall j$, and $0 \leq \delta \leq 1$ is a random variable.

A *discrete distribution:*

- We are given **fixed** values $0 = \delta_0 \leq \delta_2 \leq \ldots \leq \delta_K = 1$ example: $\delta_i = \frac{i}{K}$
- Adversary *chooses* $\pi_i = \text{Prob}(\delta = \delta_i)$, $0 \leq i \leq K$
- The π_i are *constrained*: we have fixed bounds, $\pi_i^l \leq \pi_i \leq \pi_i^u$ (and possibly other constraints)
- Tier constraints: for sets (“tiers”) T_h of assets, $1 \leq h \leq H$, we require:
  $E(\delta \sum_{j \in T_h} w_j) \leq \Gamma_h$ (given)

  or, $(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$

Robust optimization problem (VaR case):

Given $0 < \epsilon$,

$$\min \; V$$

Subject to:

$$\lambda x^T Q x \; - \; \mu^T x \; \leq \; v * + \epsilon$$

$$Ax \geq b$$

$$V \; \geq \; \text{VaR}^{max}(x)$$

Here, $v^* \doteq \min \; \lambda x^T Q x \; - \; \mu^T x$

Subject to:

$$Ax \geq b$$

Robust optimization problem (VaR case):

Given $0 < \epsilon$,

$$\min \; V$$

Subject to:

$$\lambda x^T Q x \; - \; \mu^T x \;\; \leq \;\; v * + \epsilon$$

$$Ax \geq b$$

$$V \; \geq \; \text{VaR}^{max}(x)$$

Theorem: The problem can be reduced to K SOCPs.
K = number of points in discrete distribution

Adversarial problem – a nonlinear MIP

Recall: random return$_j$ $\mu_j = \bar{\mu}_j(1 - \delta w_j)$
where $\delta = \delta_i$ (given) with probability π_i (chosen by adversary),
$0 \leq \delta_0 \leq \delta_1 \leq \ldots \leq \delta_K = 1$ and $0 \leq w$

$\min_{\pi, w, V} \min_{1 \leq i \leq k} V_i$

Subject to

$0 \leq w_j \leq 1$, all j, $\pi_i^l \leq \pi_i \leq \pi_i^u$, all i,
$\sum_i \pi_i = 1$,

$V_i = \sum_j \bar{\mu}_j(1 - \delta_i w_j) x_j^*$, if $\pi_i + \pi_{i+1} + \ldots + \pi_K \geq \theta$
$V_i = M$ *(large)*, otherwise

$(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$, for each tier h

Adversarial problem – a nonlinear MIP

Recall: random return$_j$ $\mu_j = \bar{\mu}_j(1 - \delta w_j)$
where $\delta = \delta_i$ (given) with probability π_i (chosen by adversary),
$0 \leq \delta_0 \leq \delta_1 \leq \ldots \leq \delta_K = 1$ and $0 \leq w$

$$\min_{\pi, w, V} \ \min_{1 \leq i \leq k} V_i$$

Subject to

$0 \leq w_j \leq 1$, all j, $\pi_i^l \leq \pi_i \leq \pi_i^u$, all i,
$\sum_i \pi_i = 1$,

$V_i = \sum_j \bar{\mu}_j (1 - \delta_i w_j) x_j^*$, if $\pi_i + \pi_{i+1} + \ldots + \pi_K \geq \theta$
$V_i = M$ (*large*), otherwise

$(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h$, for each tier h

Adversarial problem – a nonlinear MIP

Recall: random return$_j$ $\mu_j = \bar{\mu}_j(1 - \delta w_j)$
where $\delta = \delta_i$ (given) with probability π_i (chosen by adversary),
$0 \le \delta_0 \le \delta_1 \le \ldots \le \delta_K = 1$ and $0 \le w$

$$\min_{\pi, w, V} \ \min_{1 \le i \le k} V_i$$

Subject to

$0 \le w_j \le 1$, all j, $\pi^l_i \le \pi_i \le \pi^u_i$, all i,
$\sum_i \pi_i = 1$,

$V_i = \sum_j \bar{\mu}_j (1 - \delta_i w_j) x^*_j$, if $\pi_i + \pi_{i+1} + \ldots + \pi_K \ge \theta$
$V_i = M$ (large), otherwise

$(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \le \Gamma_h$, for each tier h

Adversarial problem – a nonlinear MIP

Recall: random return$_j$ $\mu_j = \bar{\mu}_j(1 - \delta w_j)$
where $\delta = \delta_i$ (given) with probability π_i (chosen by adversary),
$0 \leq \delta_0 \leq \delta_1 \leq \ldots \leq \delta_K = 1$ and $0 \leq w$

$$\min_{\pi, w, V} \ \min_{1 \leq i \leq k} V_i$$

Subject to

$0 \leq w_j \leq 1,$ all j, $\pi_i^l \leq \pi_i \leq \pi_i^u,$ all i,
$\sum_i \pi_i = 1,$

$V_i = \sum_j \bar{\mu}_j(1 - \delta_i w_j) x_j^*,$ if $\pi_i + \pi_{i+1} + \ldots + \pi_K \geq \theta$
$V_i = M$ $(large),$ otherwise

$(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h,$ for each tier h

Approximation

$(\sum_i \delta_i \pi_i) \sum_{j \in T_h} w_j \leq \Gamma_h,$ for each tier h $(*)$

Let $N > 0$ be an integer. For $1 \leq k \leq N$, write

$\frac{k}{N} \sum_{j \in T_h} w_j \leq \Gamma_h + M(1 - z_{hk}),$ where

$z_{hk} = 1$ if $\frac{k-1}{N} < \sum_i \delta_i \pi_i \leq \frac{k}{N}$

$z_{hk} = 0$ otherwise

$\sum_k z_{hk} = 1$

and M is large

Lemma. Under reasonable conditions, replacing $(*)$ with this system creates an error of order $O(\frac{1}{N})$

Approximation

$$\left(\sum_i \delta_i \pi_i\right) \sum_{j \in T_h} w_j \le \Gamma_h, \quad \text{for each tier h} \qquad (*)$$

Let $N > 0$ be an integer. For $1 \le k \le N$, write

$\frac{k}{N} \sum_{j \in T_h} w_j \le \Gamma_h + M(1 - z_{hk})$, where

$z_{hk} = 1$ if $\frac{k-1}{N} < \sum_i \delta_i \pi_i \le \frac{k}{N}$

$z_{hk} = 0$ otherwise

$\sum_k z_{hk} = 1$

and M is large

Lemma. Under reasonable conditions, replacing $(*)$ with this system creates an error of order $O(\frac{1}{N})$

Approximation

$$\left(\sum_i \delta_i \pi_i\right) \sum_{j \in T_h} w_j \leq \Gamma_h, \quad \text{for each tier h} \qquad (*)$$

Let $N > 0$ be an integer. For $1 \leq k \leq N$, write

$\frac{k}{N} \sum_{j \in T_h} w_j \leq \Gamma_h + M\,(1 - z_{hk}),$ where

$z_{hk} = 1$ if $\frac{k-1}{N} < \sum_i \delta_i \pi_i \leq \frac{k}{N}$

$z_{hk} = 0$ otherwise

$\sum_k z_{hk} = 1$

and M is large

Lemma. Under reasonable conditions, replacing $(*)$ with this system creates an error of order $O(\frac{1}{N})$

Implementor's problem for Benders approach, at iteration r:

$\min \; V$

Subject to:

$\lambda x^T Q x \; - \; \mu^T x \; \leq \; (1 + \epsilon) v^*$

$Ax \geq b$

$V \; \geq \; \nu \; - \; \sum_j \bar{\mu}_j \left(1 - \delta_{i(t)} w_j^{(t)}\right) x_j, \quad t = 1, 2, \ldots, r - 1$

Here, $\delta_{i(t)}$ and $w^{(t)}$ are the adversary's output at iteration $t < r$.

Implementor's problem for Benders approach, at iteration r:

min V

Subject to:

$$\lambda x^T Q x \; - \; \mu^T x \;\; \leq \;\; (1+\epsilon) v^*$$

$$Ax \geq b$$

$$V \; \geq \; \nu \; - \; \sum_j \bar{\mu}_j \left(1 - \delta_{i(t)} w_j^{(t)}\right) x_j, \qquad t = 1, 2, \ldots, r-1$$

Here, $\delta_{i(t)}$ and $w^{(t)}$ are the adversary's output at iteration $t < r$.

But we can do better:

At iteration t, the adversary produces a probability distribution $\pi^{(t)}$ and a vector $w^{(t)}$

and the cut is: $V \geq \nu - \sum_j \bar{\mu}_j \left(1 - \delta_{i(t)} w_j^{(t)}\right) x_j$

here, $i(t)$ is smallest s.t. $\sum_{i \geq i(t)} \pi_i^{(t)} \geq \theta$

But we can do better:

At iteration t, the adversary produces a probability distribution $\pi^{(t)}$ and a vector $w^{(t)}$

and the cut is: $V \geq \nu - \sum_j \bar{\mu}_j \left(1 - \delta_{i(t)} w_j^{(t)}\right) x_j$

How about a cut that is valid for *every* w s.t. $(\pi^{(t)}, w)$ is feasible for the adversary?

We want an expression for

$$\min \sum_j \bar{\mu}_j (1 - \delta_{i(t)} w_j) x_j$$

Subject to

$$\left(\sum_i \delta_i \pi_i^{(t)}\right) \sum_{j \in T_h} w_j \leq \Gamma_h, \quad \text{for each tier h}$$

$\longrightarrow$ Use LP duality

$\longrightarrow$ The implementor's problem will gain new variables and rows at each iteration

We want an expression for

$$\min \sum_j \bar{\mu}_j (1 - \delta_{i(t)} w_j) x_j$$

Subject to

$$\left(\sum_i \delta_i \pi_i^{(t)}\right) \sum_{j \in T_h} w_j \leq \Gamma_h, \quad \text{for each tier h}$$

$\longrightarrow$ Use LP duality

$\longrightarrow$ The implementor's problem will gain new variables and rows at each iteration

We want an expression for

$$\min \sum_j \bar{\mu}_j (1 - \delta_{i(t)} w_j) x_j$$

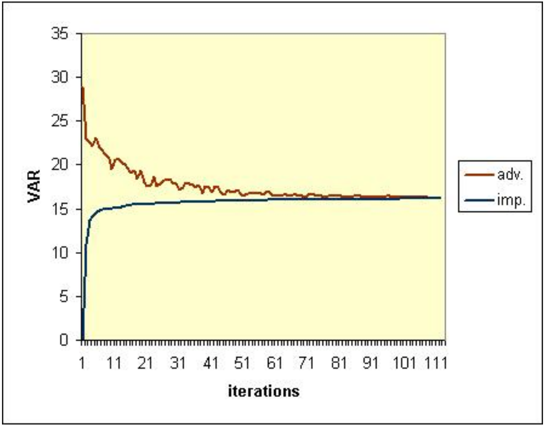
Subject to

$$\left(\sum_i \delta_i \pi_i^{(t)}\right) \sum_{j \in T_h} w_j \leq \Gamma_h, \quad \text{for each tier h}$$

→ Use LP duality

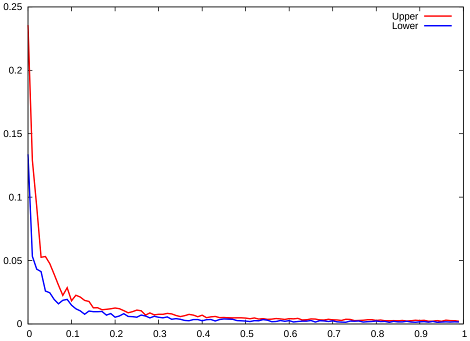
→ The implementor's problem will gain new variables and rows at each iteration

Typical convergence behavior – simple Benders



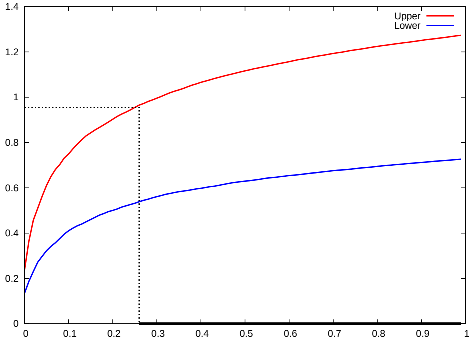
Heavy-tailed instances, $\theta = .05$

Heavy tail, proportional error (100 points):



Heavy-tailed instances, $\theta = .05$

Heavy tail, proportional error (100 points):



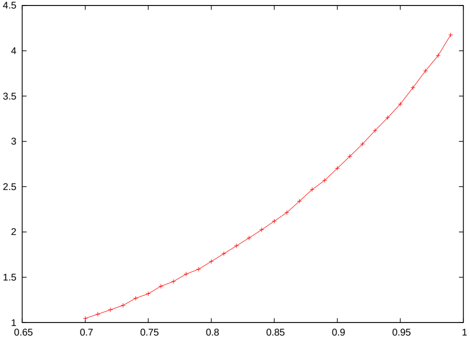
Heavy-tailed instances, $\theta = .05, \; K = 100$

VaR	A	D	E	F	G	I
time	1.98	5.02	2.47	2.03	26.51	38.32
iters	2	2	2	2	2	2
impt	0.25	2.25	0.54	1.07	14.09	19.90
advt	1.26	1.14	1.32	0.24	2.17	1.47
adj τ	$2.8e^{-04}$	$2.4e^{-04}$	$3.0e^{-04}$	$2.5e^{-04}$	$4.7e^{-05}$	$2.1e^{-04}$

CVaR	A	D	E	F	G	I
time	7.10	14.11	6.23	11.45	33.13	88.43
iters	2	2	2	2	2	3
impt	0.16	1.72	1.18	0.66	9.56	52.13
advt	6.72	10.67	4.74	10.33	12.2	23.85
gap	$9.8e^{-04}$	$2.2e^{-05}$	$7.3e^{-05}$	$5.1e^{-05}$	$3.2e^{-05}$	$1.3e^{-04}$
apperr	$2.3e^{-04}$	$2.2e^{-05}$	$2.4e^{-04}$	$1.6e^{-05}$	$1.0e^{-04}$	$2.2e^{-04}$

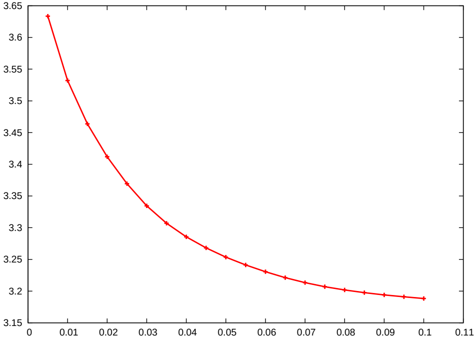
Impact of tail probability

"confidence level" = $1 - \theta$



Impact of suboptimality target

Fix $\theta = 0.95$ but vary suboptimality criterion



Experiment: sensitivity of model to parameters

Adversary chooses $\pi_i = P(\delta = \delta_i)$, $\pi_i^l \leq \pi_i \leq \pi_i^u$

Experiment: choose $\Delta \geq 0$, and solve robust problems for

$$\pi_i \leftarrow \max\{\pi_i^l - \Delta, 0\}, \quad \pi_i^l \leftarrow \pi_i^u + \Delta$$

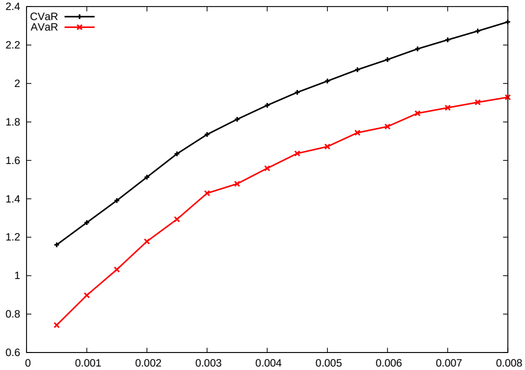
Experiment: sensitivity of model to parameters

Adversary chooses $\pi_i = P(\delta = \delta_i)$, $\pi_i^l \leq \pi_i \leq \pi_i^u$

Experiment: choose $\Delta \geq 0$, and solve robust problems for

$$\pi_i \leftarrow \max\{\pi_i^l - \Delta, 0\}, \ \pi_i^l \leftarrow \pi_i^u + \Delta$$

VaR and CVaR as a function of data errors:



($N = 10^4$ for VaR case)