



Scenario-Based Value-at-Risk Optimization

Oleksandr Romanko

Quantitative Research Group, Algorithmics Incorporated, an IBM Company

Joint work with **Helmut Mausser**

Fields Industrial Optimization Seminar
7 February 2012

Research supported by **Mitacs**

Quick facts about Algorithmics, an IBM Company

Algorithmics offers risk solutions, software and advisory services for

- Banking
- Insurance
- Asset Management
- Hedge Funds
- Pension Funds

Founded in 1989

Acquired by IBM in 2011

Over 800 employees worldwide

- 200 in Research and Development
- 250 in Professional Services
- 110 in Business Lines

Head Office in Toronto

- Primary offices in London and New York
- 23 offices globally in all major financial centers
- Clients in 55 countries

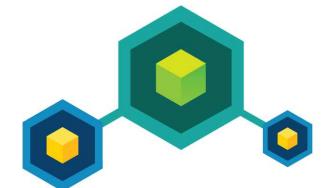
2010 revenue over \$163M USD



Some of our clients



& Modeling Framework



Mark-to-Future framework

- n Pre-compute asset values:

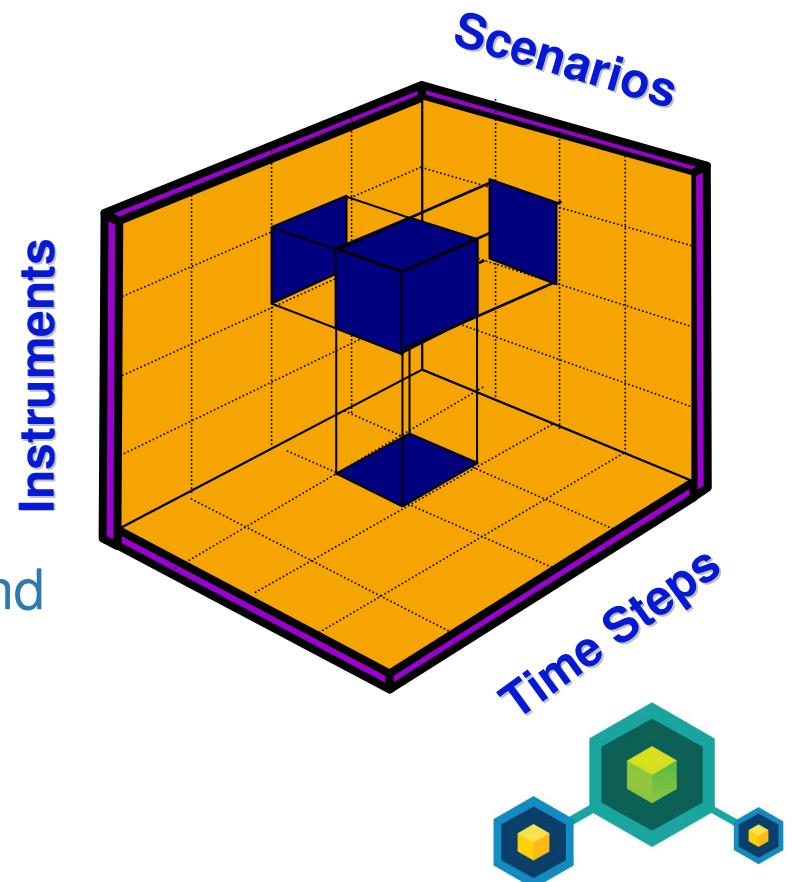
$$A_{j,i,t} = f_j(\text{market factors}_{i,t})$$

- n Compute portfolio value later for any portfolio:

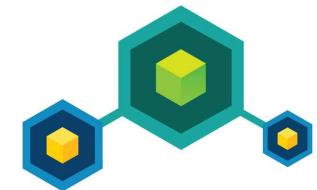
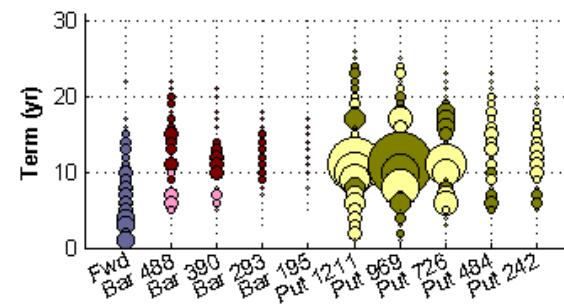
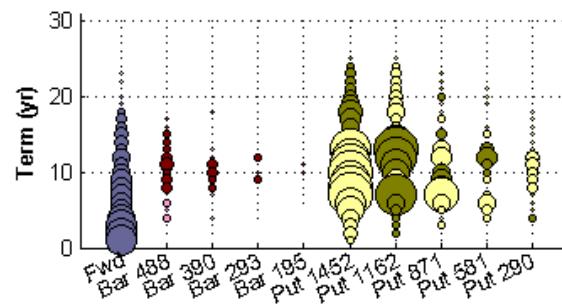
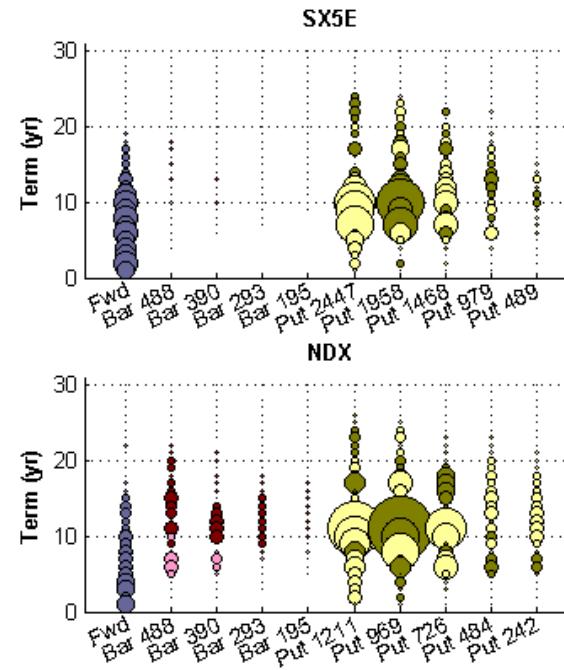
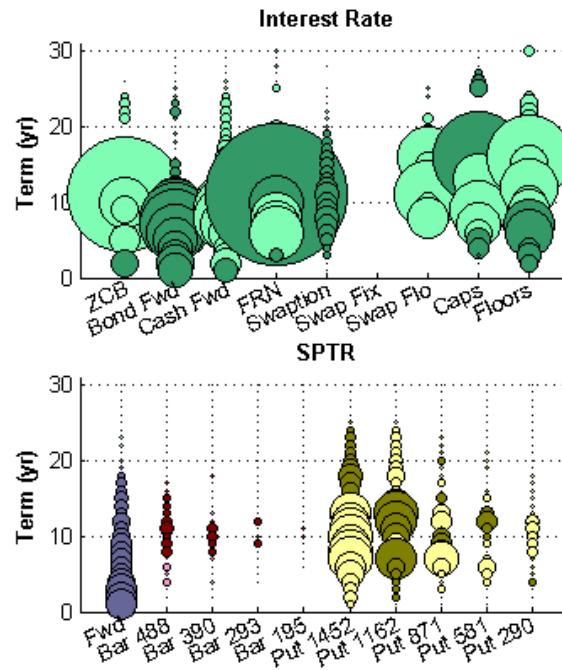
$$V_{i,t} = \sum_{j=1}^J w_j \cdot A_{j,i,t}$$

Each scenario represents a given realization of a state of the world
 Instruments are valued under each scenario at each time step
 Values are stored in an MtF Cube

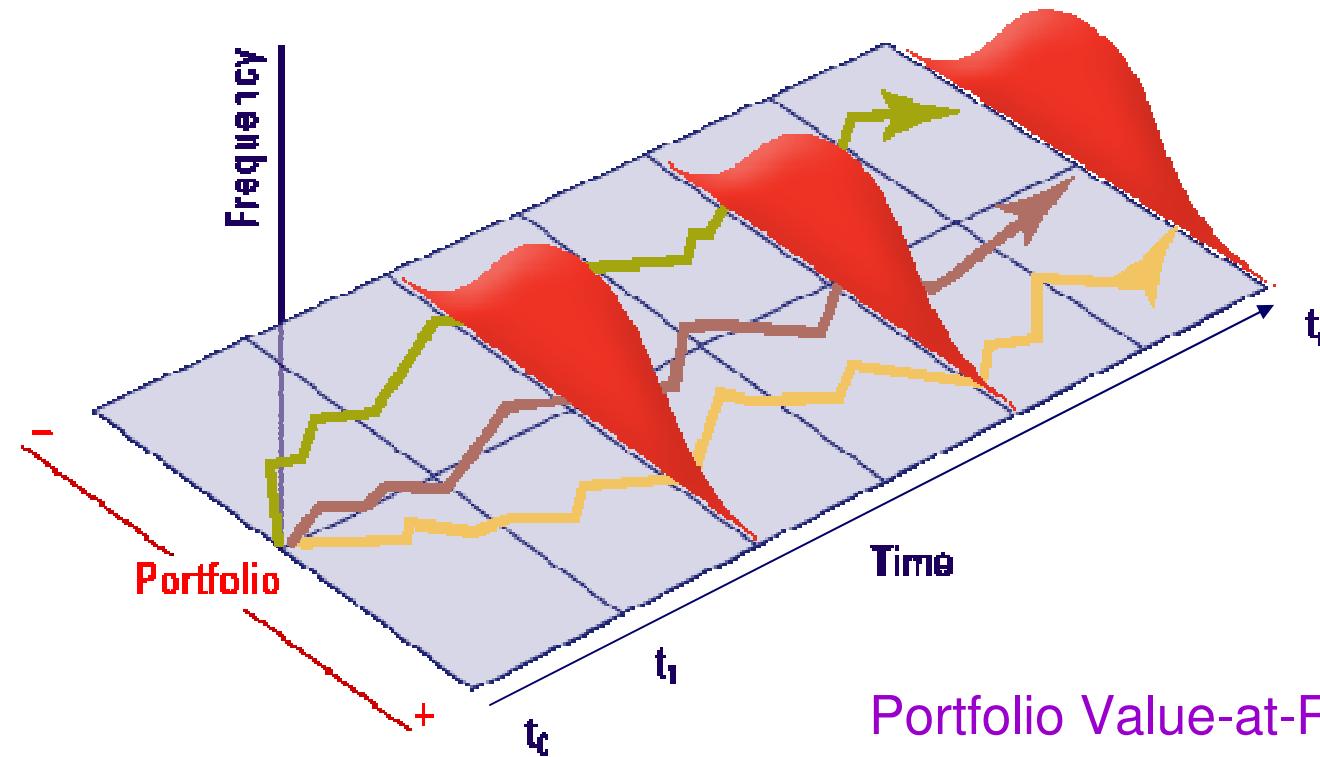
- n This allows new positions to be taken and portfolios to be recombined without re-valuation of the instruments



Portfolio



Computing risk measures from simulation



Portfolio Value-at-Risk

Based on empirical distribution
e.g. VaR 99% over 1000 scenarios
10th worst outcome



Mark-to-Future framework and sampling

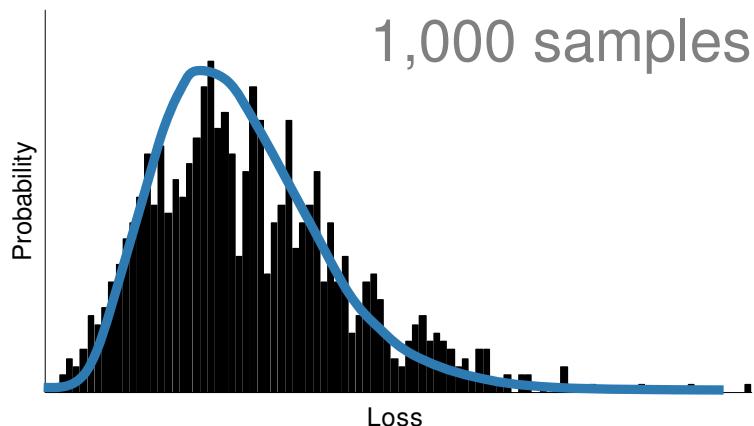
- n Pre-compute asset values:

$$A_{j,i,t} = f_j(\text{market factors}_{i,t})$$

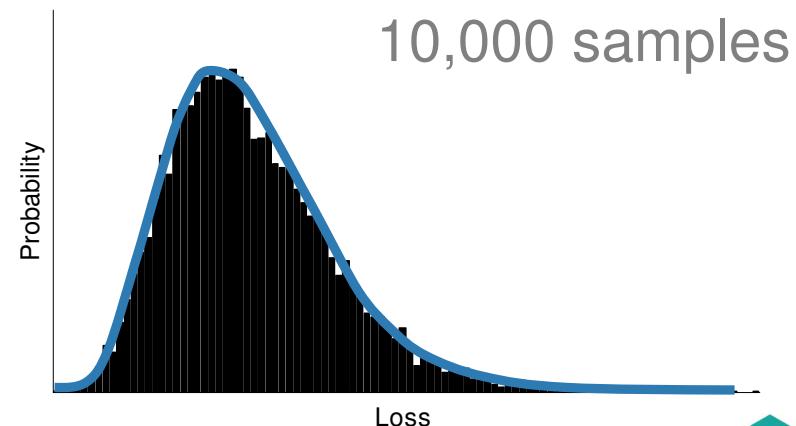
- n Compute portfolio value later for any portfolio:

$$V_{i,t} = \sum_{j=1}^J w_j \cdot A_{j,i,t}$$

- n Portfolio loss distribution:

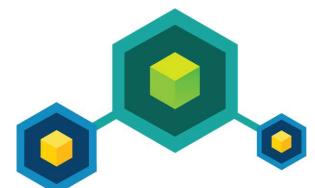


1,000 samples

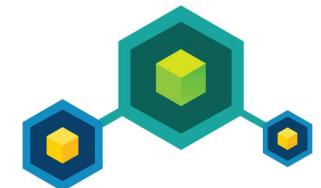


10,000 samples

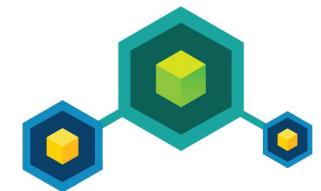
$$\ell_i = -(V_{i,1} - V_0)$$



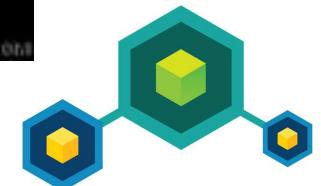
& Optimization Framework



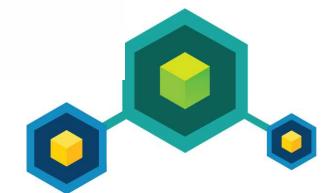
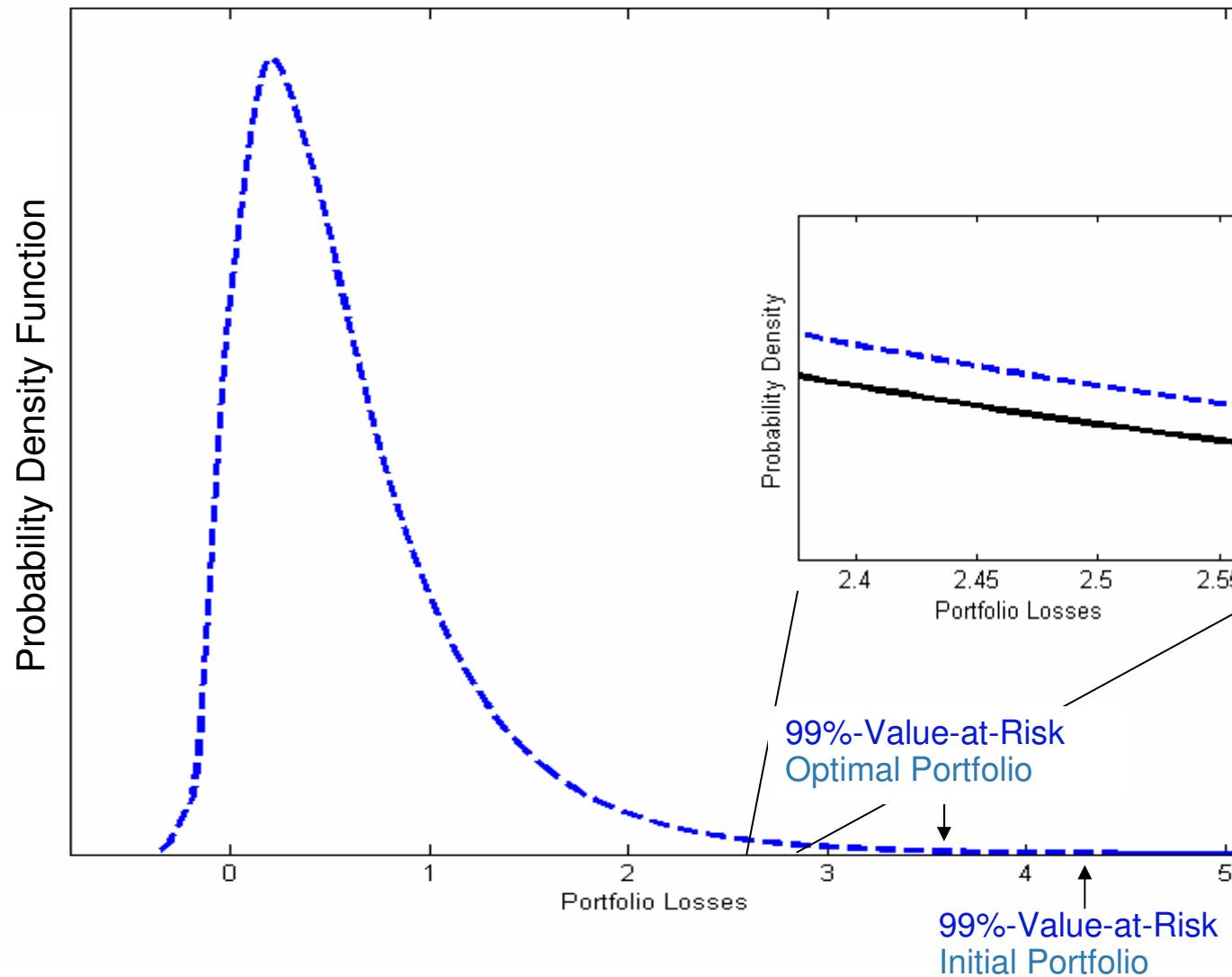
Which risks are worth taking?



Risk management

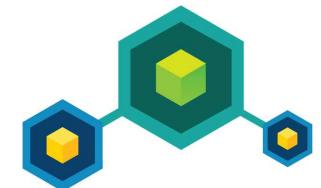


Portfolio risk optimization



&

Value-at-Risk Optimization



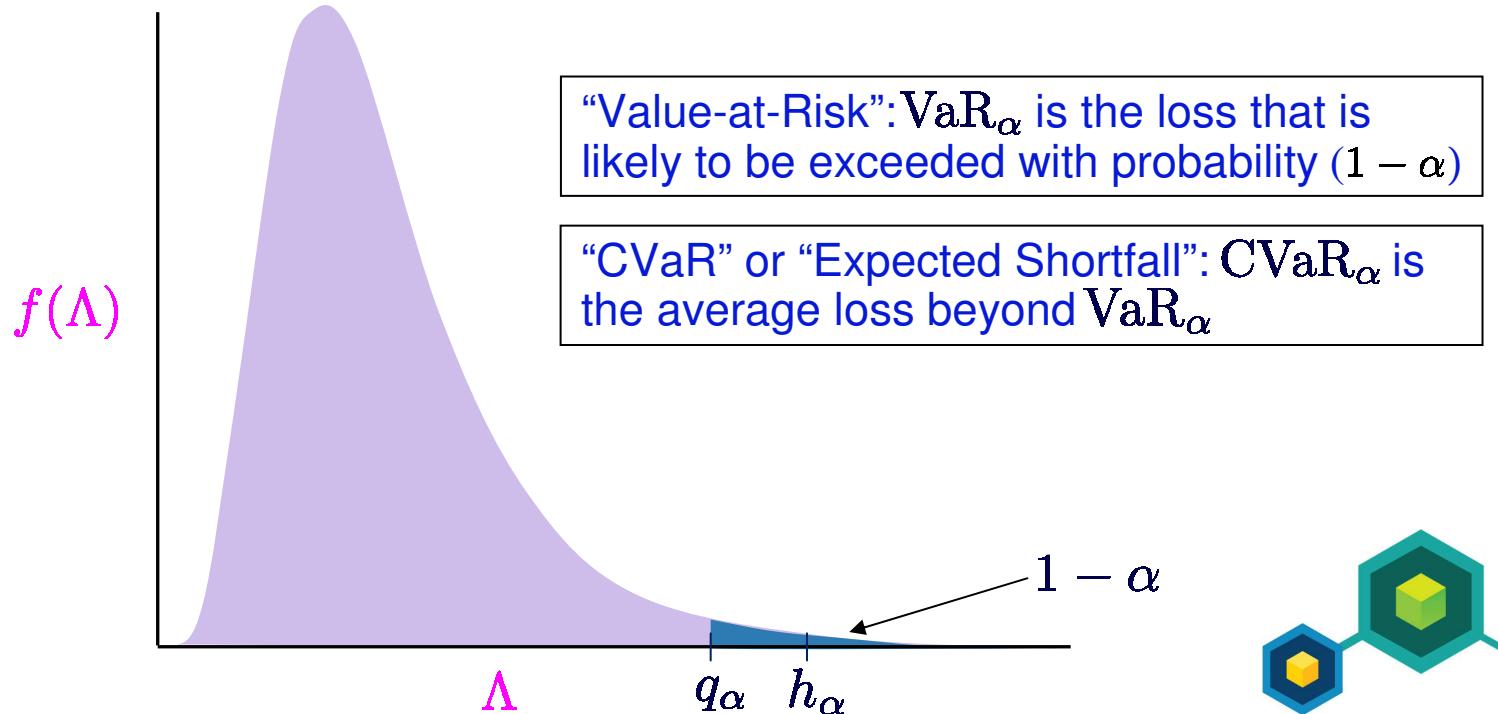
Tail-based risk measures

n Notation

- $\mathbf{w} \in \Omega \subseteq \mathbb{R}^J$ is a portfolio, where w_j is the weight of asset j
- F is the multivariate distribution of asset returns
- $q_\alpha(\mathbf{w})$ is the *actual risk* (out-of-sample VaR) of the portfolio \mathbf{w}
- $q_{\alpha,N}(\mathbf{w})$ is an *estimate* of $q_\alpha(\mathbf{w})$ based on a sample of size N from F

n Consider a continuous random variable Λ with distribution F

- the Value-at-Risk (VaR) at level α : $q_\alpha = F^{-1}(\alpha)$
- the Conditional Value-at-Risk (CVaR) at level α : $h_\alpha = \mathbb{E}[\Lambda | \Lambda > q_\alpha]$



Estimators

- Given a random sample of size N , let $\ell_{(k)}$ be the k^{th} order statistic, i.e.,

$$\ell_{(1)} \leq \ell_{(2)} \leq \dots \leq \ell_{(N)}$$

– An estimate of q_α is $q_{\alpha,N} = \ell_{(\lceil N\alpha \rceil)}$

– An estimate of h_α is $h_{\alpha,N} = \frac{1}{N(1-\alpha)} \left[(\lceil N\alpha \rceil - N\alpha) \ell_{(\lceil N\alpha \rceil)} + \sum_{k=\lceil N\alpha \rceil+1}^N \ell_{(k)} \right]$

...	$\ell_{(98)}$	$\ell_{(99)}$	$\ell_{(100)}$	$q_{0.98,100} = 0.42$	$h_{0.98,100} = 0.47$
...	0.42	0.44	0.50	$q_{0.975,100} = 0.42$	$h_{0.975,100} = 0.46$

More observations in the tail → less noise → more robust estimates

Rockafellar, R.T. and S. Uryasev (2002), “Conditional Value-at-Risk for General Loss Distributions,” *Journal of Banking and Finance* 26, 1443-1471.



Problem formulation

$\S \ell_i(\mathbf{w}) = \sum_{j=1}^J -r_{ij} w_j$ is the loss of portfolio $\mathbf{w} \in \Omega$ in scenario i

VaR minimization

$$\begin{array}{ll} \min_{\mathbf{w}, \mathbf{z}, q} & q \\ \text{s.t.} & \ell_i(\mathbf{w}) - q \leq M z_i, \quad i = 1, \dots, N \\ & \sum_{i=1}^N z_i \leq \lfloor N(1 - \alpha) \rfloor \\ & z_i \in \{0, 1\}, \quad i = 1, \dots, N \\ & \mathbf{w} \in \Omega \end{array}$$

MIP formulation

CVaR minimization

$$\begin{array}{ll} \min_{\mathbf{w}, q} & q + \frac{1}{N(1 - \alpha)} \sum_{i=1}^N [\ell_i(\mathbf{w}) - q]^+ \\ \text{s.t.} & \mathbf{w} \in \Omega \\ & a^+ = \max(0, a) \\ \\ \min_{\mathbf{w}, q, \mathbf{y}} & q + \frac{1}{N(1 - \alpha)} \sum_{i=1}^N y_i \\ \text{s.t.} & \ell_i(\mathbf{w}) - q - y_i \leq 0, \quad i = 1, \dots, N \\ & y_i \geq 0, \quad i = 1, \dots, N \\ & \mathbf{w} \in \Omega \end{array}$$

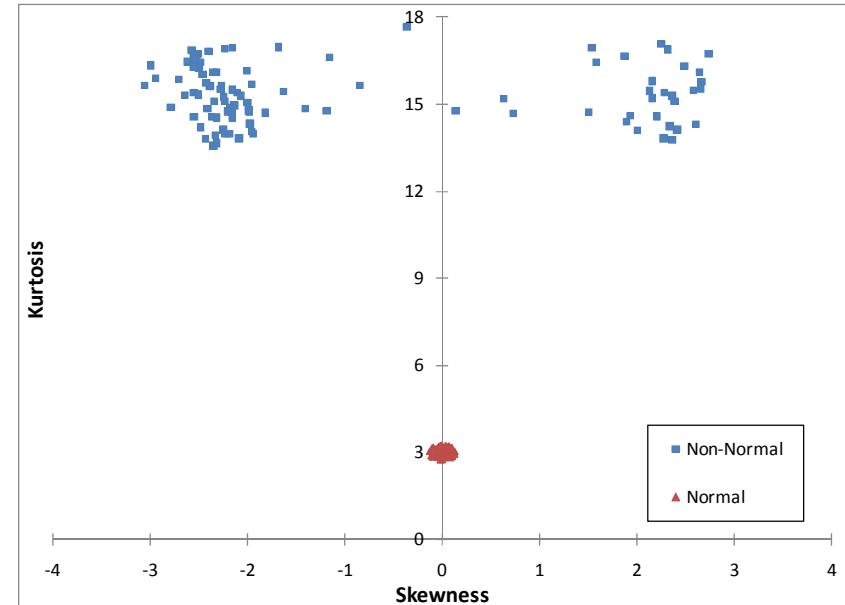

LP formulation

\S The larger $N(1 - \alpha)$, i.e., # tail observations, the longer the solution time
 – Better VaR estimates, in particular, come with high computational cost



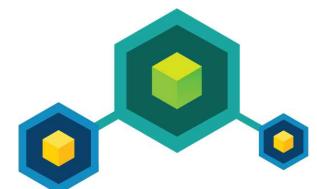
Asset return scenarios

- § International stocks selected based on monthly returns from January 2003 to June 2010 (90 months)
 - 100 stocks with Normal returns
 - 100 stocks with non-Normal returns (skewed, leptokurtic)



- § Generate return scenarios that match the first four moments and the correlations of the historical stock returns
 - One set of 1,000,000 scenarios represents the true return distribution
 - 25 sets of 20,000 scenarios for optimization

Høyland, K., Kaut, M. and S.W. Wallace (2003), “A Heuristic for Moment-Matching Scenario Generation,” *Computational Optimization and Applications* 24, 169-185



Experimental design

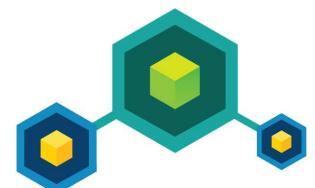
§ Consider the following parameters

- Multivariate asset return distributions: $\Phi = \{\text{Normal, Non-Normal}\}$
- Feasible region: $\Omega = \left\{ \mathbf{w} \in \mathbb{R}^J : \sum_{j=1}^J w_j = 1, w_j \geq 0 \text{ for } j = 1, \dots, J \right\}$
- Sample sizes: $\mathcal{N} = \{1000, 5000, 10000, 20000\}$
- Risk measures:
 - out-of-sample: $Q = \{q_{0.90}, q_{0.95}, q_{0.99}, q_{0.999}\}$
 - in-sample: $Q_N = \{q_{0.90,N}, q_{0.95,N}, q_{0.99,N}, q_{0.999,N}\}$

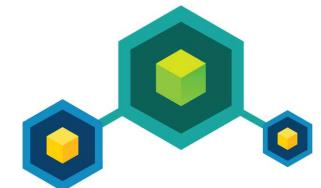
§ For each $F \in \Phi$, $N \in \mathcal{N}$, $q_N \in Q_N$

- Find \mathbf{w}^* that minimizes $q_N(\mathbf{w}^*)$
- For each $q \in Q$ and its estimator $q_N \in Q_N$
 - Record in-sample and out-of-sample VaR
 - Record problem statistics
 - Record computational time

Perform 25 trials with different samples and average the results



& **Algorithm for
Value-at-Risk
Optimization**



CVaR proxy for VaR

- n CVaR optimization problem

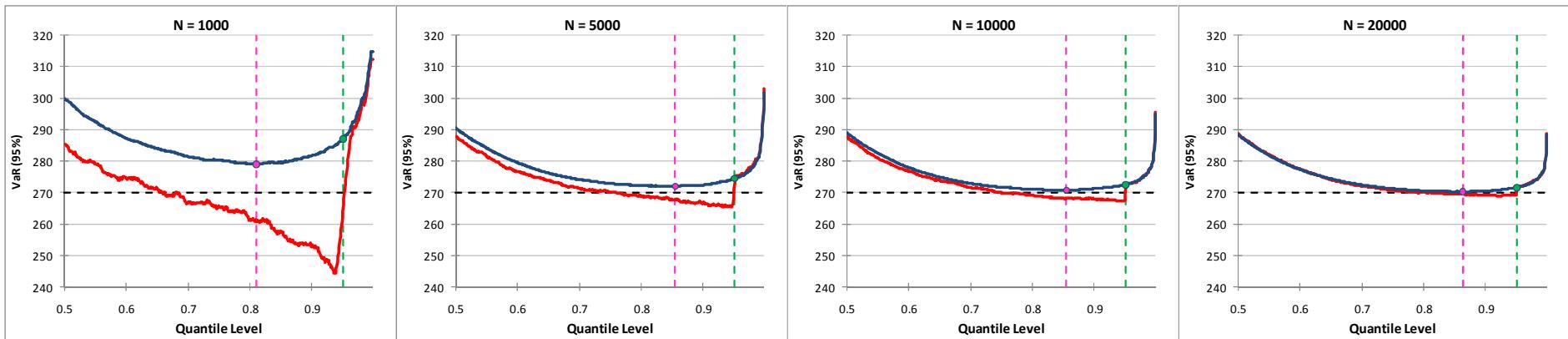
$$\begin{aligned} \min_{\boldsymbol{w}, q} \quad & q + \overbrace{\frac{1}{1-\alpha}}^1 \frac{1}{N} \sum_{i=1}^N [\ell_i(\boldsymbol{w}) - q]^+ \\ \text{s.t.} \quad & \boldsymbol{w} \in \Omega \end{aligned}$$

- n Simple $\text{VaR}_{0.95}$ minimization algorithm (solve CVaR optimization problem at different quantile levels to optimize VaR):
 1. Set $\alpha = 0.95 : -0.01 : 0.50$
 2. Minimize CVaR_α problem
 3. Compute $\text{VaR}_{0.95}$

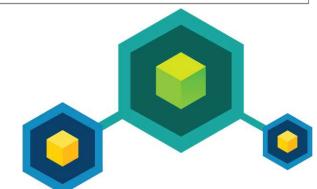
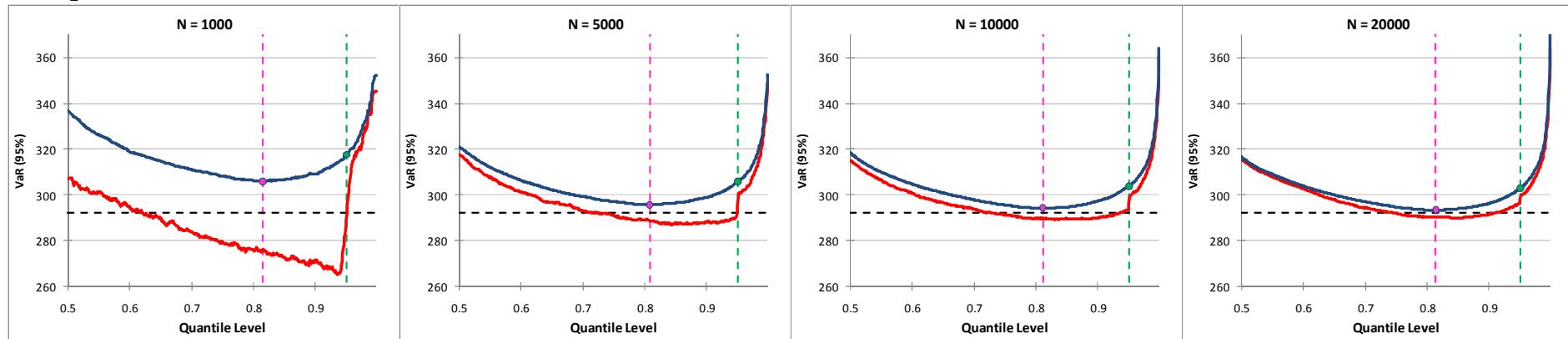


CVaR proxy frontiers for VaR 95%

q Normal

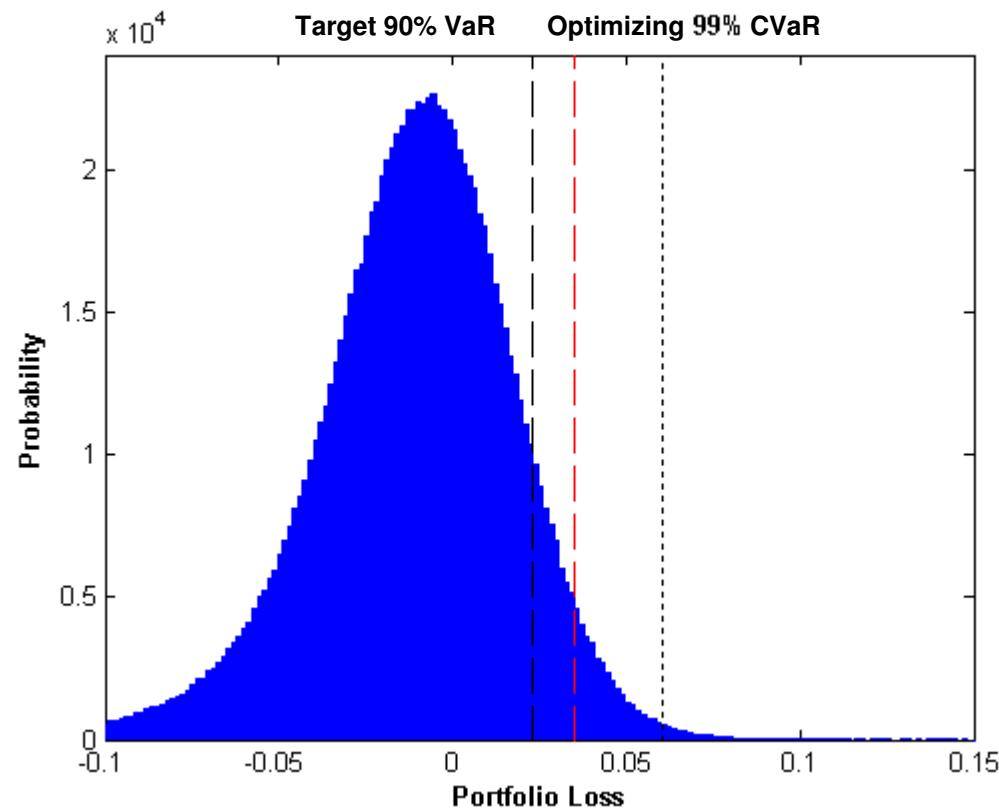


q Non-Normal



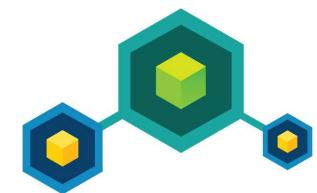
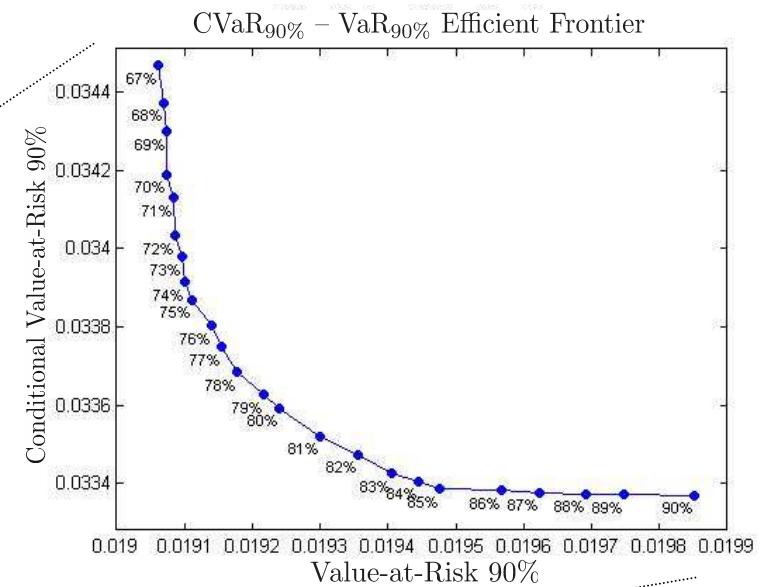
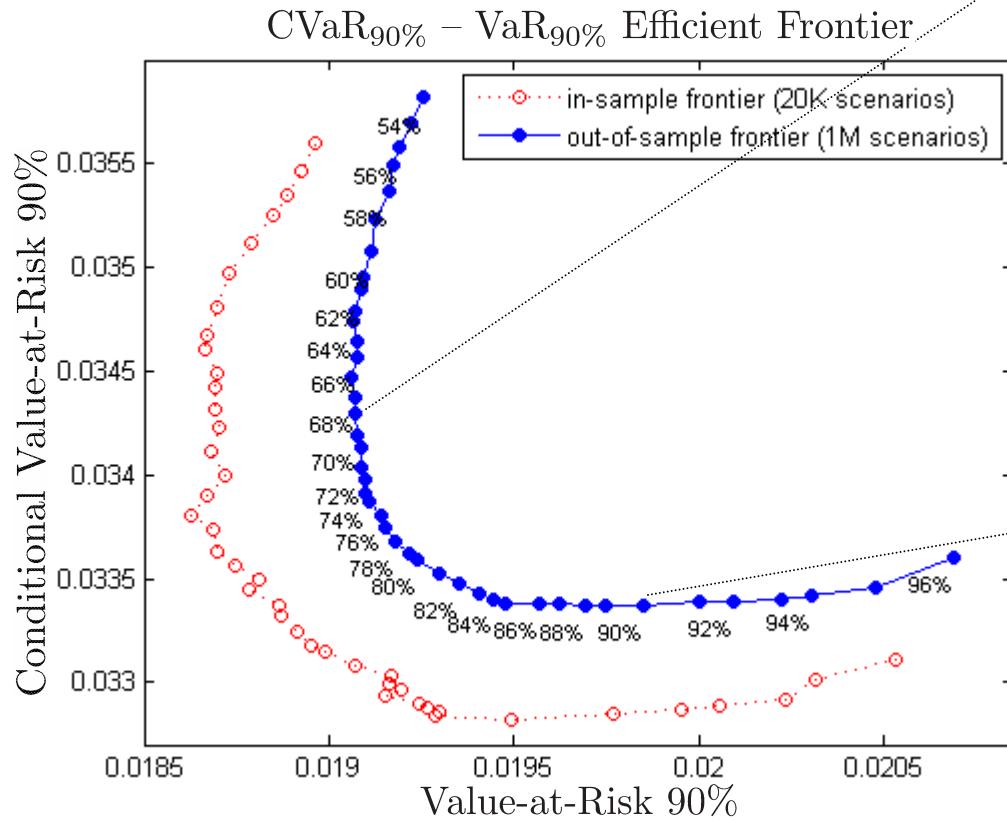
CVaR proxy for VaR

- n Solve CVaR optimization problem at different quantile levels α to optimize VaR_α



CVaR proxy frontiers for VaR

- Solve CVaR optimization problem at different quantile levels α to optimize $\text{VaR}_{\alpha'}$



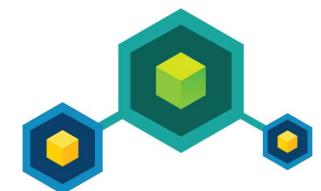
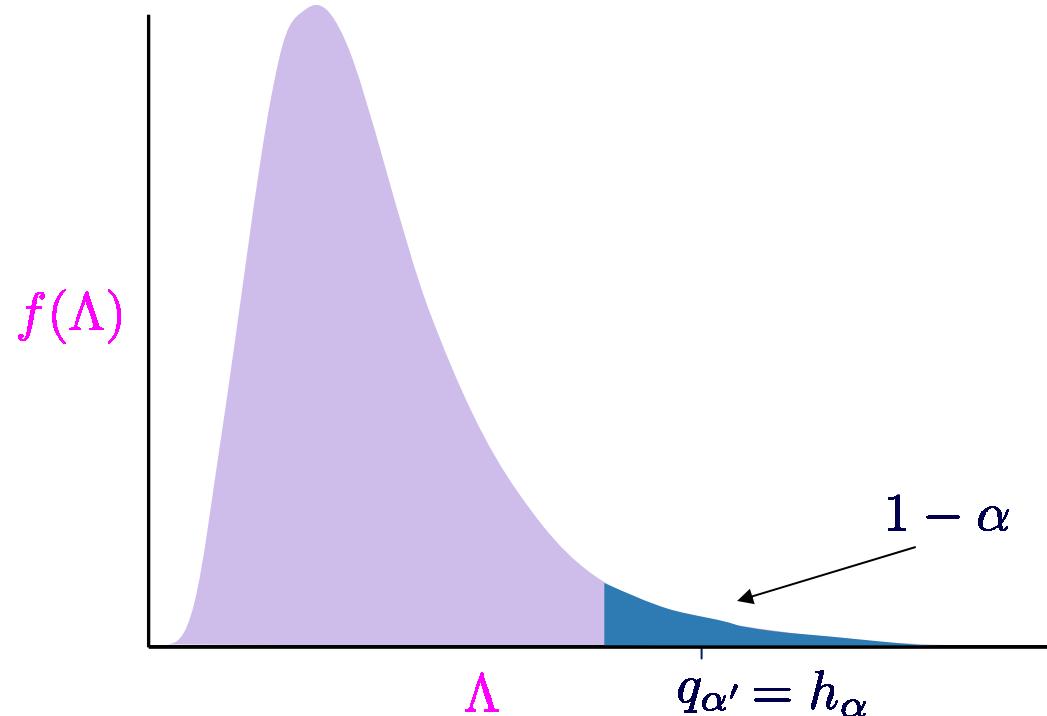
Algorithm for VaR minimization - idea

n Problem:

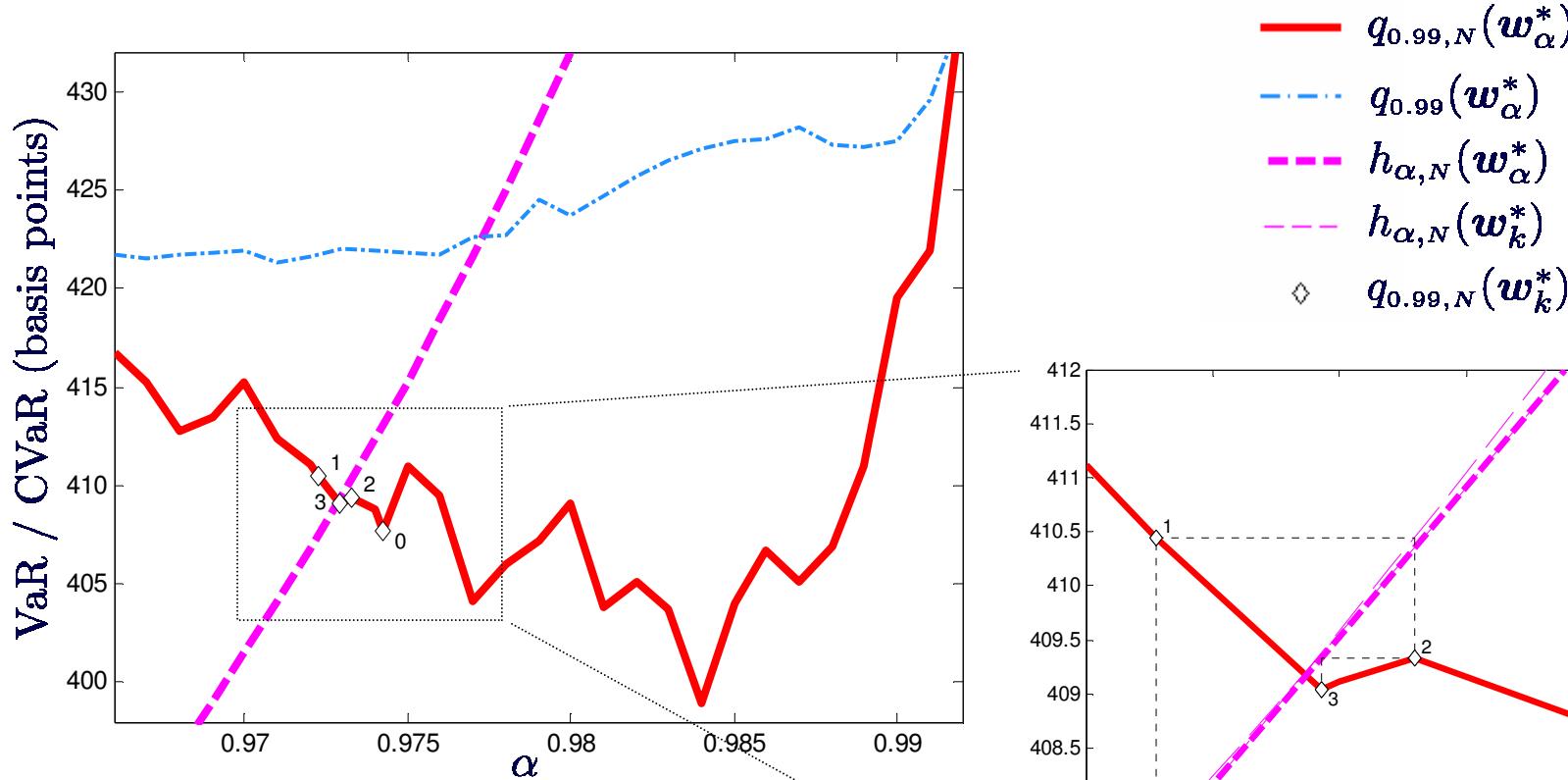
- q generating frontiers CVaR_α is computationally intensive
- q we need only one point on the frontier with the smallest value of $\text{VaR}_{\alpha'}$

n Idea:

- q find a point α on the frontier where $\text{VaR}_{\alpha'} \approx \text{CVaR}_\alpha$



Algorithm for VaR minimization - illustration



For Normal distributions:

$\text{CVaR}_\alpha = \text{VaR}_{\alpha'}$ if α satisfies

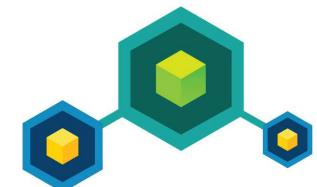
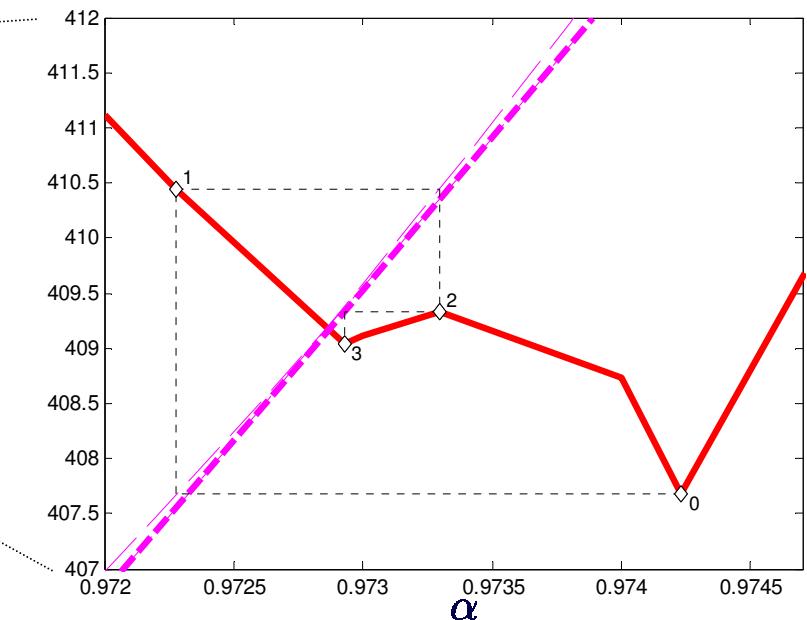
$$\frac{\phi(Z_\alpha)}{1 - \alpha} = Z_{\alpha'}$$

$\text{VaR } 90\% = \text{CVaR } 75.44\%$

$\text{VaR } 95\% = \text{CVaR } 87.45\%$

$\text{VaR } 99\% = \text{CVaR } 97.42\%$

$\text{VaR } 99.9\% = \text{CVaR } 99.74\%$



Algorithm for VaR minimization

Step 0. Initialization

Set $k = 0$, $\alpha_0 = \alpha_{\text{init}}$ (some appropriate initial quantile level).

Step 1. Optimize

Solve the optimization problem with $\text{CVaR}_{\alpha_k}(\mathbf{w})$ as a proxy for $\text{VaR}_\alpha(\mathbf{w})$:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \text{CVaR}_{\alpha_k}(\mathbf{w}) \\ \text{s.t.} \quad & \mathbf{w} \in \Omega. \end{aligned}$$

If an optimal solution exists then denote it by \mathbf{w}_k^* .

Step 2. Check for termination

- i) If optimization problem in Step 1 is infeasible or unbounded then stop.
- ii) If the termination criterion (convergence or iteration limit) is satisfied then stop and return \mathbf{w}_k^* .

Step 3. Sort relevant differences

- i) For $i = 1, \dots, N$, compute losses $\ell_i = -r_i^T \cdot \mathbf{w}_{k,i}^*$.
- ii) Order the ℓ_i , $i = 1, \dots, N$, in decreasing sequence, breaking ties arbitrarily. Let π index the scenarios with respect to this ordering, i.e., for $m = 1, \dots, N$, $\pi(m) = i$, where i is the scenario with the m^{th} -largest relevant difference.

Step 4. Estimate VaR

- i) Find m^* satisfying

$$\sum_{m=1}^{m^*-1} p_{\pi(m)} \leq 1 - \alpha < \sum_{m=1}^{m^*} p_{\pi(m)}$$

- ii) Set $\text{VaR}_\alpha^k = \ell_{\pi(m^*)}$.

- iii) Update the incumbent solution if necessary.

Step 5. Adjust CVaR quantile level so that $\text{CVaR}_{\alpha_{k+1}} = \text{VaR}_\alpha^k$

- i) Find m' satisfying

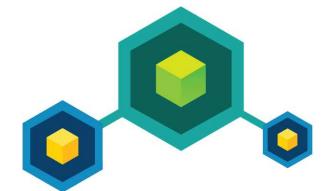
$$\sum_{m=1}^{m'} p_{\pi(m)} (\ell_{\pi(m)} - \text{VaR}_\alpha^k) \geq 0 > \sum_{m=1}^{m'+1} p_{\pi(m)} (\ell_{\pi(m)} - \text{VaR}_\alpha^k).$$

- ii) Set

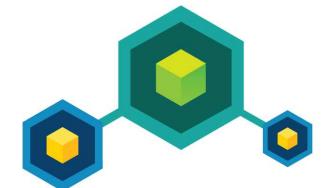
$$\alpha_{k+1} = 1 - \left(\sum_{m=1}^{m'} p_{\pi(m)} + \frac{\sum_{m=1}^{m'} p_{\pi(m)} (\ell_{\pi(m)} - \text{VaR}_\alpha^k)}{\text{VaR}_\alpha^k - \ell_{\pi(m'+1)}} \right).$$

Step 6. Continue

- i) Set $k = k + 1$.
- ii) Go to Step 1.



& Value-at-Risk
Minimization:
Computational Testing

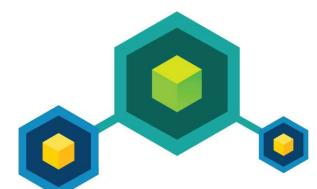
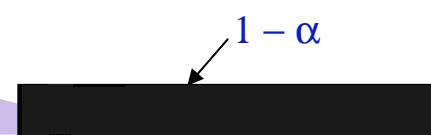


VaR minimization

- n Repeat previous computational experiments for VaR_α , $\alpha = 90\%, 95\%, 99\%, 99.9\%$
- n Proxies for VaR_α
 - q Algorithm (1 iteration)
 - q Algorithm (5 iterations)
 - q CVaR $_\alpha$
 - q MIP (30 minute time limit)
- n Additional proxies considered for VaR_α
 - q Heuristic of Larsen et al.
 - Minimize CVaR $_{\alpha'}$ then iteratively “discard” tail scenarios
 - We report results for $\xi = 0.1, 0.5, 1.0$

Larsen, N., Mausser H., and S. Uryasev (2002), “Algorithms for Optimization of Value-at-Risk,” in *Financial Engineering, e-commerce and Supply Chain*, P. Pardalos and V.K. Tsitsirigos (Eds.), Kluwer, 129-157.

$\xi = 0.5 \Rightarrow$ discard half the remaining tail scenarios in each iteration



Performance of algorithms for VaR minimization

Average in-sample VaR (%) relative to $q_{\alpha, N}$ of Heur ($\xi = 0.1$) - optimal portfolio

Returns	Algorithm \ N	VaR 90%				VaR 95%				VaR 99%				VaR 99.9%			
		1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000
Normal	Alg (5 iter)	23.5	8.3	5.2	3.2	16.5	6.6	4.5	3.0	11.0	7.3	5.4	3.5	0.6	1.6	5.1	5.2
	Alg (1 iter)	25.0	8.5	5.3	3.3	17.2	6.9	4.6	3.1	12.5	7.7	5.6	3.6	1.1	2.1	6.3	5.6
	CVaR $_{\alpha}$	30.3	9.9	6.4	4.0	21.9	8.4	5.3	3.5	19.1	9.0	6.4	4.3	1.1	7.3	9.7	7.7
	Heur ($\xi = 1$)	21.1	8.1	5.6	3.6	13.5	6.5	4.3	3.1	9.8	5.8	4.4	3.3	0.0	4.2	5.0	4.6
	Heur ($\xi = 0.5$)	4.7	3.1	2.6	1.9	3.3	1.7	1.4	1.2	5.3	1.2	0.9	0.9	0.0	2.4	2.9	1.1
	Heur ($\xi = 0.1$)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MIP	17.0	22.5	20.9	18.2	-1.1	14.0	15.6	14.4	-1.7	3.4	11.4	13.3	-4.8	-3.1	-0.3	2.4
Non-Normal	Alg (5 iter)	32.2	10.7	6.4	3.4	19.9	8.2	5.6	3.6	12.1	8.1	5.8	3.9	0.7	2.9	5.4	5.8
	Alg (1 iter)	33.5	11.4	6.7	3.7	21.0	8.4	5.8	3.7	13.5	8.7	6.0	4.1	1.0	3.1	6.0	6.1
	CVaR $_{\alpha}$	43.5	17.1	12.4	9.1	29.1	11.5	8.2	6.4	21.8	12.0	8.0	5.7	1.0	7.9	10.5	9.4
	Heur ($\xi = 1$)	30.9	15.0	11.4	8.6	17.6	9.4	7.2	5.8	10.7	7.9	5.9	4.6	0.0	4.0	5.4	5.1
	Heur ($\xi = 0.5$)	10.6	7.2	6.7	5.8	5.4	3.4	3.2	3.4	4.4	2.5	1.7	1.3	0.0	2.8	2.6	2.2
	Heur ($\xi = 0.1$)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MIP	17.9	37.9	43.5	55.7	-3.0	18.3	25.2	34.2	-3.1	3.6	14.3	18.1	-5.5	-3.0	-0.5	4.3

MIP performs better for smaller $N(1 - \alpha) \Leftarrow$ easier to solve

Alg cannot match *Heur* ($\xi = 0.5$, $\xi = 0.1$) since *Alg* never discards any scenarios



Performance of algorithms for VaR minimization

Average out-of-sample VaR relative to best known VaR_α , %

Returns	Algorithm \ N	VaR 90%				VaR 95%				VaR 99%				VaR 99.9%			
		1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000
Normal	Alg (5 iter)	3.6	0.6	0.2	0.0	3.8	0.8	0.3	0.0	8.2	1.4	0.6	0.1	11.9	5.3	2.1	0.6
	Alg (1 iter)	3.6	0.6	0.2	0.0	3.8	0.8	0.3	0.0	7.9	1.4	0.5	0.1	12.2	5.5	2.1	0.6
	CVaR $_\alpha$	5.4	1.5	1.1	0.8	6.2	1.6	0.9	0.5	11.3	2.7	1.4	0.7	12.2	7.9	5.1	2.8
	Heur ($\xi = 1$)	6.8	2.0	1.3	1.0	7.4	2.0	1.1	0.7	11.6	3.3	1.7	0.9	11.4	8.3	4.7	2.6
	Heur ($\xi = 0.5$)	8.0	2.1	1.2	0.7	7.5	2.0	1.1	0.6	10.7	3.1	1.9	0.9	11.4	7.1	4.2	1.9
	Heur ($\xi = 0.1$)	9.8	2.9	1.7	0.8	9.0	2.8	1.6	0.9	9.5	3.3	2.2	1.3	11.4	5.1	3.0	1.8
	MIP	13.0	11.7	13.7	14.1	9.8	7.6	9.5	10.7	9.9	4.9	5.1	8.2	10.2	4.3	3.0	1.9
Non-Normal	Alg (5 iter)	4.4	0.9	0.2	0.0	4.5	1.0	0.4	0.1	10.7	2.5	1.7	1.2	24.2	12.5	10.0	8.9
	Alg (1 iter)	4.6	1.1	0.4	0.2	4.5	1.1	0.4	0.2	10.6	2.5	1.7	1.2	24.6	12.5	10.0	9.1
	CVaR $_\alpha$	8.9	6.1	5.3	5.0	7.9	3.9	3.2	2.9	14.7	5.5	4.4	3.8	24.6	14.0	14.3	14.1
	Heur ($\xi = 1$)	10.4	6.4	5.5	5.1	9.4	4.1	3.4	3.0	15.3	5.7	4.1	3.8	26.2	14.3	12.6	14.0
	Heur ($\xi = 0.5$)	10.0	4.1	3.7	3.8	8.5	2.8	2.3	2.0	13.5	4.7	3.5	2.6	26.2	14.4	10.9	8.7
	Heur ($\xi = 0.1$)	12.5	3.1	1.3	0.6	10.1	3.2	1.7	0.8	12.5	5.1	3.4	2.3	26.2	12.4	9.2	8.4
	MIP	20.1	22.5	31.3	46.9	13.3	8.6	17.3	30.4	14.0	6.2	7.8	19.5	22.1	11.0	7.5	7.0

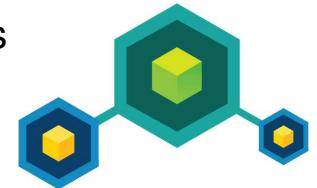
In most cases *Alg* (1 or 5 iter) outperforms other algorithms

One iteration of *Alg* is usually enough for the Normal case, while for non-Normal returns more than one iteration is needed

Heur does not perform well as it overfits to the in-sample scenarios

MIP solutions exhibit poor performance except for VaR 99.9%

For non-Normal VaR 99.9% none of the algorithms produces satisfactory results



Quantile levels computed by the Algorithm

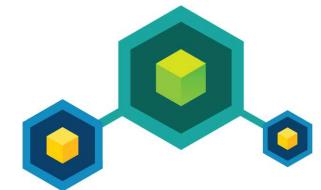
Solution times

Quantile levels α_k computed by the Algorithm

Returns	Algorithm	VaR 90%				VaR 95%				VaR 99%				VaR 99.9%			
		1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000
All	Alg (1 iter)	75.44	75.44	75.44	75.44	87.45	87.45	87.45	87.45	97.42	97.42	97.42	97.42	99.74	99.74	99.74	99.74
Normal	Alg (5 iter)	75.53	75.42	75.45	75.48	87.49	87.41	87.44	87.48	97.44	97.42	97.45	97.46	99.70	99.72	99.73	99.74
Non-Normal	Alg (5 iter)	74.60	74.02	74.43	74.00	87.18	87.13	87.00	86.86	97.39	97.34	97.34	97.29	99.70	99.73	99.74	99.73

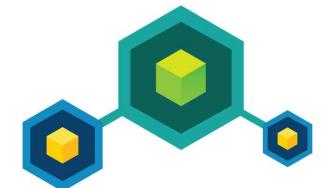
Solution times (in seconds) for all algorithms

Returns	Algorithm	VaR 90%				VaR 95%				VaR 99%				VaR 99.9%			
		1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000	1000	5000	10000	20000
Normal	Alg (5 iter)	0.9	19.5	55.8	100.1	0.8	14.7	40.6	84.8	0.8	7.6	24.0	51.5	0.6	12.1	19.5	24.7
	Alg (1 iter)	0.6	17.1	44.4	91.0	0.5	12.0	31.5	72.6	0.3	4.7	15.2	39.0	0.3	3.8	7.2	12.5
	CVaR $_{\alpha}$	0.4	10.3	28.0	53.0	0.3	7.0	17.7	47.5	0.3	4.0	14.6	23.8	0.3	3.2	4.7	10.0
	Heur ($\xi = 1$)	1.4	12.7	35.8	65.3	1.3	10.4	23.6	64.3	1.2	6.5	20.6	36.0	1.0	5.7	10.6	21.6
	Heur ($\xi = 0.5$)	4.6	75.9	151.8	388.0	3.5	62.2	107.3	279.2	2.3	37.3	71.8	163.1	1.1	15.2	30.8	81.2
	Heur ($\xi = 0.1$)	14.5	247.4	465.4	1167.1	11.8	192.7	367.5	911.0	7.0	126.0	223.1	565.0	1.1	48.4	102.4	263.9
	MIP	1805.7	1800.4	1801.1	1823.7	1807.3	1800.4	1800.9	1847.4	1812.7	1800.3	1800.7	1802.3	13.1	1396.0	1801.1	1801.3
Non-Normal	Alg (5 iter)	0.9	24.2	62.4	82.8	0.7	16.9	44.8	82.2	0.7	8.5	24.9	53.0	0.5	11.0	18.4	23.2
	Alg (1 iter)	0.6	19.1	41.9	47.0	0.4	13.5	30.9	59.4	0.3	5.4	15.2	41.4	0.3	3.4	5.3	11.8
	CVaR $_{\alpha}$	0.4	11.7	26.2	59.1	0.3	7.9	17.7	54.7	0.3	4.1	14.2	24.4	0.3	3.2	4.5	9.3
	Heur ($\xi = 1$)	1.4	15.0	31.8	71.1	1.2	10.7	26.0	72.0	1.2	7.1	20.1	41.2	1.1	5.5	10.2	21.1
	Heur ($\xi = 0.5$)	4.2	75.2	139.8	384.3	3.4	58.2	105.6	304.3	2.1	35.4	66.3	156.4	1.0	14.0	31.8	76.8
	Heur ($\xi = 0.1$)	14.0	252.1	453.2	1161.6	11.1	189.7	363.6	970.5	6.6	126.3	223.6	525.2	1.1	46.7	95.2	252.4
	MIP	1806.5	1800.5	1801.5	1837.1	1806.1	1800.4	1801.1	1820.5	1812.2	1800.3	1800.9	1803.0	9.3	1768.2	1800.5	1800.8



&

Value-at-Risk Constrained Optimization

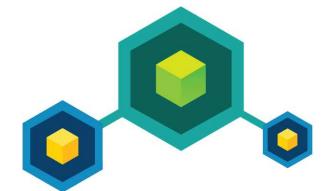


Value-at-Risk constrained optimization

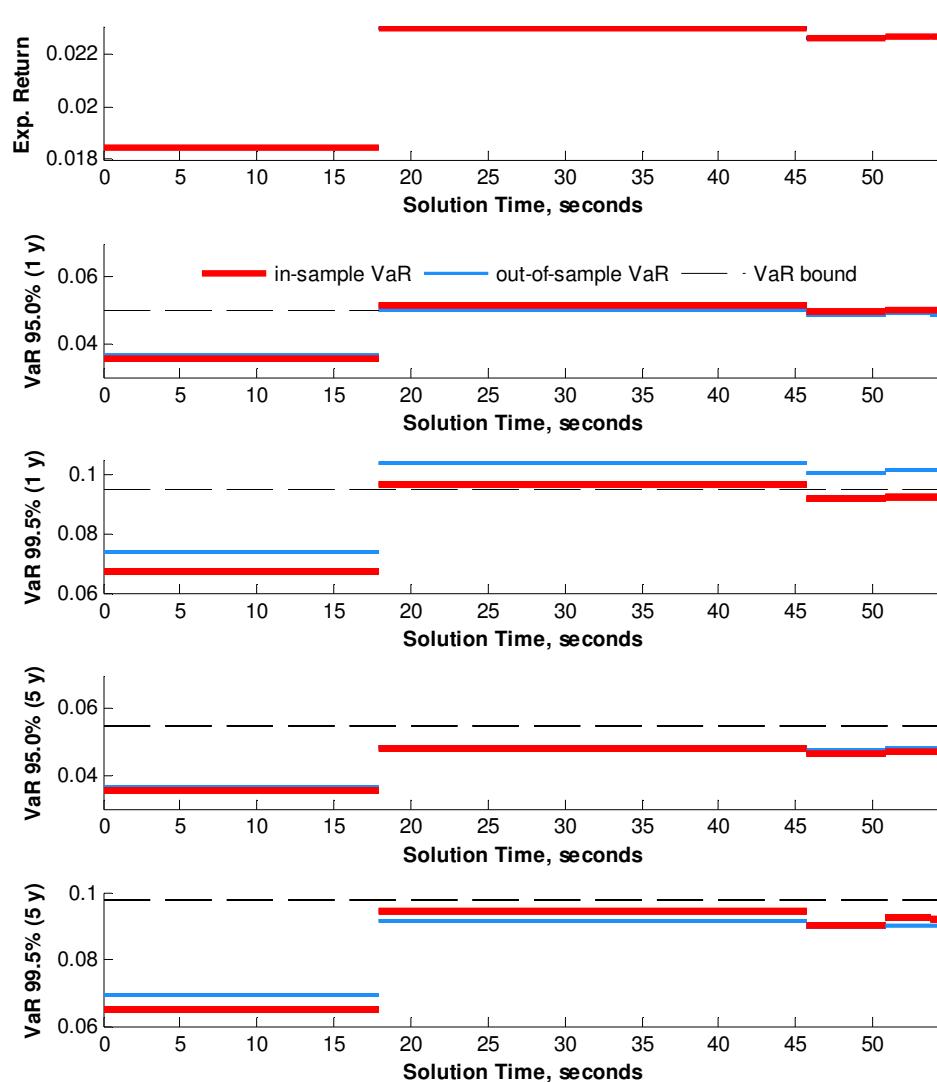
$$\begin{aligned}
 \max_{\boldsymbol{w}} \quad & \sum_{j=1}^J \mu_j w_j \\
 \text{s.t.} \quad & \text{VaR}_{0.950}^1(\boldsymbol{w}) \leq 0.050 \\
 & \text{VaR}_{0.995}^1(\boldsymbol{w}) \leq 0.095 \\
 & \text{VaR}_{0.950}^5(\boldsymbol{w}) \leq 0.055 \\
 & \text{VaR}_{0.995}^5(\boldsymbol{w}) \leq 0.098 \\
 & \sum_{j=1}^J w_j = 1 \\
 & \boldsymbol{w} \geq 0
 \end{aligned}$$

Iter No	Solution Objective	VaR 95.0% (1 y) 0.050	VaR 99.5% (1 y) 0.095	VaR 95.0% (5 y) 0.055	VaR 99.5% (5 y) 0.098
1	0.018445	0.03572 (95.00%)	0.06711 (99.50%)	0.03573 (95.00%)	0.06513 (99.50%)
2	0.022995	0.05138 (86.56%)	0.09638 (98.50%)	0.04831 (87.14%)	0.09461 (98.60%)
3	0.022645	0.04966 (87.42%)	0.09184 (98.70%)	0.04665 (87.08%)	0.09014 (98.84%)
4	0.022718	0.05003 (87.24%)	0.09227 (98.58%)	0.04701 (86.86%)	0.09231 (98.78%)
5	0.022702	0.04991 (87.28%)	0.09210 (98.58%)	0.04696 (86.92%)	0.09215 (98.88%)

$J = 100, N = 5000$



Value-at-Risk constrained optimization



Objective
function

$$\max_{\mathbf{w}} \sum_{j=1}^J \mu_j w_j$$

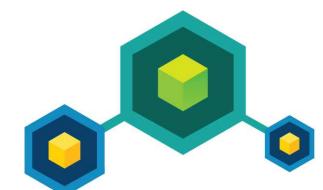
VaR
constraints

$$\text{VaR}_{0.950}^1(\mathbf{w}) \leq 0.050$$

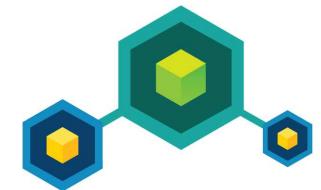
$$\text{VaR}_{0.995}^1(\mathbf{w}) \leq 0.095$$

$$\text{VaR}_{0.950}^5(\mathbf{w}) \leq 0.055$$

$$\text{VaR}_{0.995}^5(\mathbf{w}) \leq 0.098$$



& Conclusions



Optimization in Algo Risk Application

Objective Function

Add Function	Attribute	Scenario Set	Time	Benchmark	Relative Weight	Quantile	Constant
Minimize Expected Shortfall	PnL	Historical ALL Scenarios	1y	----- None -----		99	

Expected Shortfall Bound

Add Bound	Attribute	Relationship	Bound	Scenario Set	Time	Benchmark	Quantile	Constant
Expected Shortfall	PnL	=>	300	Historical ALL Scenarios	1d	----- None -----	95	

Trade List & Limits

Global Constraints

1 - Parameters

Aggregation: Asset Type

Group: Whole Portfolio Holdings: All

Attribute: Return Measure: Expectation

Scenario Set: Historical ALL Scenarios

Time: 1y

Trades: Net Current Holdings:

Scale: None

Soft Constraint: None

2 - Relationship

Less Than / Equal To

Greater Than / Equal To

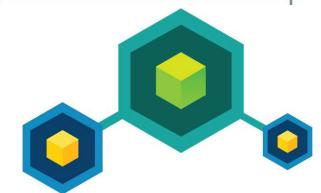
Equal To

3 - Factor

Adjusted By: Additive Multiple

4 - Target

Constant: 6% Relative To Benchmark Relative To Portfolio

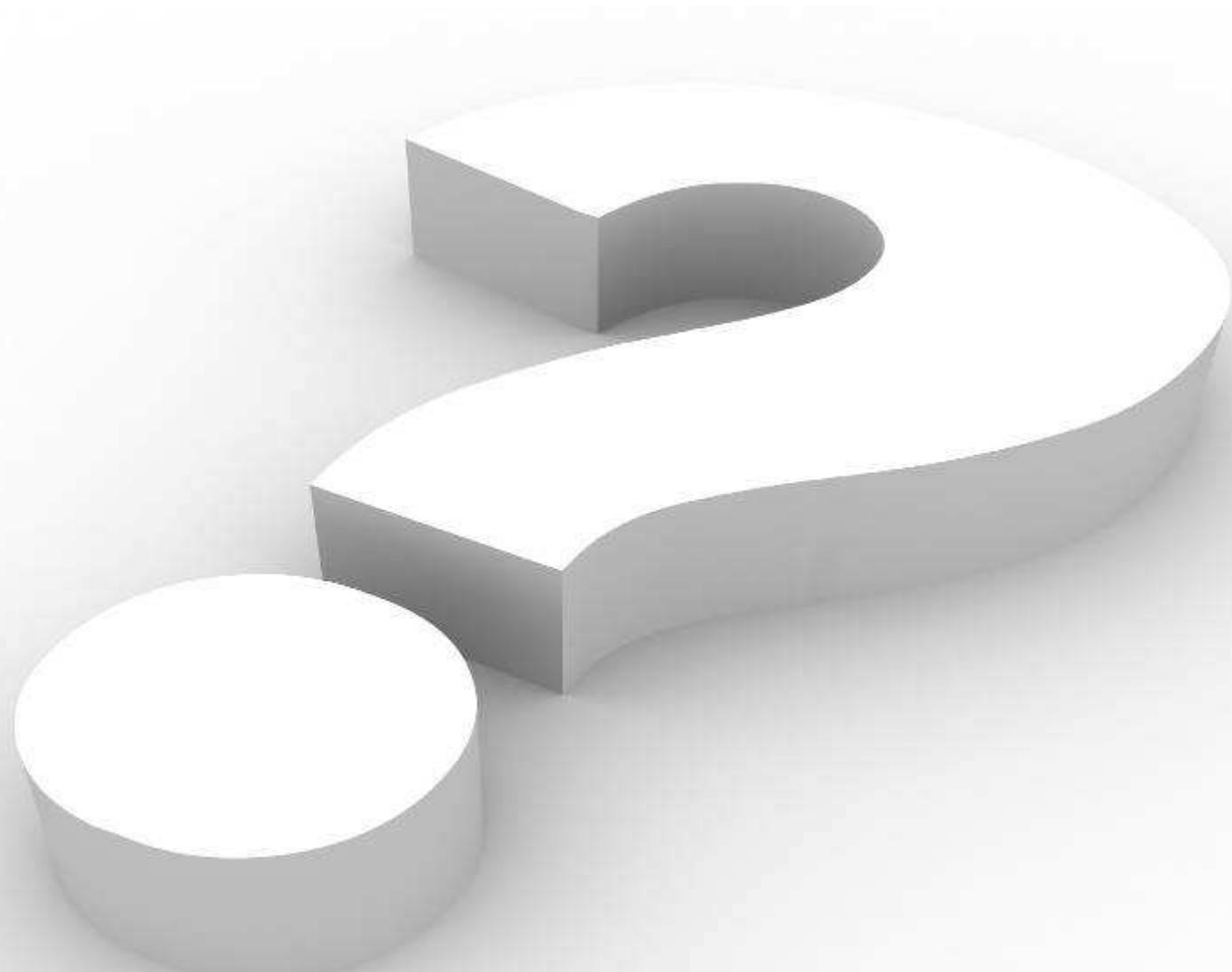


Conclusions and future work

- n We proposed a **simple Algorithm** for VaR optimization
 - n Algorithm works for both **minimizing VaR** and **constraining VaR**
 - n **Small number of iterations** (1-5) needed by the Algorithm allows solving relatively large problems
 - n **Warm-start** optimization solver (e.g., CPLEX) capabilities can be utilized
 - Performance-wise solving a **VaR optimization** problem is almost **equivalent** to solving a **CVaR optimization** problem of the same size
 - n Algorithm is useful not only for solving VaR optimization problems directly, but also for computing an **initial feasible solution** to start **MIP** optimization
 - n Algorithm is **robust** with respect to the **sampling error**
-
- n Many robust CVaR optimization techniques can directly utilize the Algorithm to perform **robust VaR optimization**
 - n Future work includes applying the Algorithm within **distributionally robust VaR optimization framework**



Questions



Thank you for your time

Contact Information:

Oleksandr Romanko

Research Fellow, Quantitative Research

Oleksandr.Romanko@algorithmics.com

<http://www.romanko.ca>

