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
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

- Pre-compute asset values:

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

- 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.

- This allows new positions to be taken and portfolios to be recomposed without revaluation of the instruments.
Portfolio

Interest Rate

SPTR

SXSE

NDX
Computing risk measures from simulation

Based on empirical distribution e.g. VaR 99% over 1000 scenarios

10th worst outcome

Portfolio Value-at-Risk

Based on empirical distribution e.g. VaR 99% over 1000 scenarios
10th worst outcome
Mark-to-Future framework and sampling

- Pre-compute asset values:

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

- Compute portfolio value later for any portfolio:

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

- Portfolio loss distribution:

\[ \ell_i = -(V_{i,1} - V_0) \]
& Optimization Framework
Which risks are worth taking?

Not all risks are worth taking.

Measuring risk along individual business lines can lead to a distorted picture of exposures. At Algorithmics, we help clients to see risk in its entirety. This unique perspective enables financial services companies to mitigate exposures, and identify new opportunities that maximize returns. Supported by a global team of risk professionals, our proven, enterprise risk solutions allow clients to master the art of risk-informed decision making through the science of knowing better.

Proven Enterprise Risk Solutions | algorithmics.com
Risk management

If you take too much risk, make sure you do it on a grand scale to ensure a government bailout.
Portfolio risk optimization

![Graph showing probability density function and 99% Value-at-Risk for initial and optimal portfolios.]

- **Probability Density Function**
- **Portfolio Losses** range from 0 to 5
- **99% Value-at-Risk**
  - Initial Portfolio
  - Optimal Portfolio
Value-at-Risk Optimization
Tail-based risk measures

Notation

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

Consider a continuous random variable \( \Lambda \) with distribution \( F \)
- the Value-at-Risk (VaR) at level \( \alpha \): \( q_\alpha = F^{-1}(\alpha) \)
- the Conditional Value-at-Risk (CVaR) at level \( \alpha \): \( h_\alpha = \mathbb{E}[\Lambda | \Lambda > q_\alpha] \)

“Value-at-Risk”: \( \text{VaR}_\alpha \) is the loss that is likely to be exceeded with probability \( (1 - \alpha) \)

“CVaR” or “Expected Shortfall”: \( \text{CVaR}_\alpha \) is the average loss beyond \( \text{VaR}_\alpha \)
Estimators

- Given a random sample of size $N$, let $\ell(k)$ be the $k^{th}$ order statistic, i.e.,
  \[ \ell(1) \leq \ell(2) \leq \ldots \leq \ell(N) \]
  - An estimate of $q_\alpha$ is
    \[ q_{\alpha,N} = \ell(\lceil N\alpha \rceil) \]
  - An estimate of $h_\alpha$ is
    \[ h_{\alpha,N} = \frac{1}{N(1 - \alpha)} \left( \lceil N\alpha \rceil - N\alpha \right) \ell(\lceil N\alpha \rceil) + \sum_{k=\lceil N\alpha \rceil+1}^{N} \ell(k) \]

\[
\begin{array}{ccc}
\ldots & \ell(98) & \ell(99) & \ell(100) \\
\ldots & 0.42 & 0.44 & 0.50 \\
\end{array}
\]

- More observations in the tail $\rightarrow$ less noise $\rightarrow$ more robust estimates

Problem formulation

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

<table>
<thead>
<tr>
<th>VaR minimization</th>
<th>CVaR minimization</th>
</tr>
</thead>
</table>
| \[ \begin{align*}
\min_{w,z,q} & \quad q \\
\text{s.t.} & \quad \ell_i(w) - q \leq Mz_i, \; i = 1, \ldots, N \\
& \quad \sum_{i=1}^{N} z_i \leq \lfloor N(1 - \alpha) \rfloor \\
& \quad z_i \in \{0, 1\}, \; i = 1, \ldots, N \\
& \quad w \in \Omega
\end{align*} \] | \[ \begin{align*}
\min_{w,q} & \quad q + \frac{1}{N(1 - \alpha)} \sum_{i=1}^{N} [\ell_i(w) - q]^+ \\
\text{s.t.} & \quad w \in \Omega \\
& \quad \alpha^+ = \max(0, \alpha)
\end{align*} \] |

<table>
<thead>
<tr>
<th>MIP formulation</th>
<th>LP formulation</th>
</tr>
</thead>
</table>
| \[ \begin{align*}
\min_{w,z,q} & \quad q \\
\text{s.t.} & \quad \ell_i(w) - q \leq Mz_i, \; i = 1, \ldots, N \\
& \quad \sum_{i=1}^{N} z_i \leq \lfloor N(1 - \alpha) \rfloor \\
& \quad z_i \in \{0, 1\}, \; i = 1, \ldots, N \\
& \quad w \in \Omega
\end{align*} \] | \[ \begin{align*}
\min_{w,q,y} & \quad q + \frac{1}{N(1 - \alpha)} \sum_{i=1}^{N} y_i \\
\text{s.t.} & \quad \ell_i(w) - q - y_i \leq 0, \; i = 1, \ldots, N \\
& \quad y_i \geq 0, \; i = 1, \ldots, N \\
& \quad w \in \Omega
\end{align*} \] |

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

Experimental design

Consider the following parameters

- Multivariate asset return distributions: $\Phi = \{\text{Normal, Non-Normal}\}$
- Feasible region: $\Omega = \left\{ w \in \mathbb{R}^J : \sum_{j=1}^J w_j = 1, w_j \geq 0 \text{ for } j = 1, \ldots, J \right\}$
- 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 $w^*$ that minimizes $q_N(w^*)$
- 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

CVaR optimization problem

\[
\min_{w,q} \quad q + \frac{1}{1 - \alpha} \frac{1}{N} \sum_{i=1}^{N} [\ell_i(w) - q]^+
\]

s.t. \quad w \in \Omega

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 \( \text{CVaR}_\alpha \) problem
3. Compute \( \text{VaR}_{0.95} \)
CVaR proxy frontiers for VaR 95%

- Normal

- Non-Normal
CVaR proxy for VaR

Solve CVaR optimization problem at different quantile levels $\alpha$ to optimize $\text{VaR}_\alpha$. 
CVaR proxy frontiers for VaR

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

![CVaR proxy frontiers for VaR chart](chart.png)
Algorithm for VaR minimization - idea

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

Idea:
- find a point $\alpha$ on the frontier where $\text{VaR}_\alpha' \approx \text{CVaR}_\alpha$
Algorithm for VaR minimization - illustration

For Normal distributions:

\[ \text{CVaR}_\alpha = \text{VaR}_{\alpha'} \text{ if } \alpha \text{ satisfies } \]

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

\[ \text{VaR } 90\% = \text{CVaR } 75.44\% \]
\[ \text{VaR } 95\% = \text{CVaR } 87.45\% \]
\[ \text{VaR } 99\% = \text{CVaR } 97.42\% \]
\[ \text{VaR } 99.9\% = \text{CVaR } 99.74\% \]
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 CVaR$_{\alpha_k}(\omega)$ as a proxy for VaR$_{\alpha_k}(\omega)$:

$$
\begin{align*}
\text{min} & \quad \text{CVaR}_{\alpha_k}(\omega) \\
\text{s.t.} & \quad \omega \in \Omega.
\end{align*}
$$

If an optimal solution exists then denote it by $\omega_k^*$.

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 $\omega_k^*$.

Step 3. Sort relevant differences

i) For $i = 1, \ldots, N$, compute losses $\ell_i = -\pi_i^T \cdot \omega_{k,i}^*$.

ii) Order the $\ell_i$, $i = 1, \ldots, N$, in decreasing sequence, breaking ties arbitrarily. Let $\pi$ index the scenarios with respect to this ordering, i.e., for $m = 1, \ldots, N$, $\pi(m) = i$, where $i$ is the scenario with the $m^\text{th}$-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 VaR$_{\alpha_k}^k = \ell_\pi(m^*)$.

iii) Update the incumbent solution if necessary.

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

i) Find $m'$ satisfying

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

ii) Set

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

Step 6. Continue

i) Set $k = k + 1$.

ii) Go to Step 1.
Value-at-Risk Minimization: Computational Testing
VaR minimization

- Repeat previous computational experiments for $\text{VaR}_\alpha$, $\alpha = 90\%, 95\%, 99\%, 99.9\%$

- Proxies for $\text{VaR}_\alpha$
  - Algorithm (1 iteration)
  - Algorithm (5 iterations)
  - $\text{CVaR}_\alpha$
  - MIP (30 minute time limit)

- Additional proxies considered for $\text{VaR}_\alpha$
  - Heuristic of Larsen et al.
    - Minimize $\text{CVaR}_{\alpha'}$ then iteratively “discard” tail scenarios
    - We report results for $\xi = 0.1, 0.5, 1.0$


$\xi = 0.5 \Rightarrow$ 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

<table>
<thead>
<tr>
<th>Returns</th>
<th>Algorithm</th>
<th>$N$</th>
<th>VaR 90%</th>
<th>VaR 95%</th>
<th>VaR 99%</th>
<th>VaR 99.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>5000</td>
<td>10000</td>
<td>20000</td>
<td>1000</td>
</tr>
<tr>
<td>Normal</td>
<td>Alg (5 iter)</td>
<td>23.5</td>
<td>8.3</td>
<td>5.2</td>
<td>3.2</td>
<td>16.5</td>
</tr>
<tr>
<td></td>
<td>Alg (1 iter)</td>
<td>25.0</td>
<td>8.5</td>
<td>5.3</td>
<td>3.3</td>
<td>17.2</td>
</tr>
<tr>
<td></td>
<td>CVaR$_\alpha$</td>
<td>30.3</td>
<td>9.9</td>
<td>6.4</td>
<td>4.0</td>
<td>21.9</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 1$)</td>
<td>21.1</td>
<td>8.1</td>
<td>5.6</td>
<td>3.6</td>
<td>13.5</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.5$)</td>
<td>4.7</td>
<td>3.1</td>
<td>2.6</td>
<td>1.9</td>
<td>3.3</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.1$)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MIP (5-iter)</td>
<td>17.0</td>
<td>22.5</td>
<td>20.9</td>
<td>18.2</td>
<td>-1.1</td>
</tr>
<tr>
<td>Non-Normal</td>
<td>Alg (5 iter)</td>
<td>32.2</td>
<td>10.7</td>
<td>6.4</td>
<td>3.4</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>Alg (1 iter)</td>
<td>33.5</td>
<td>11.4</td>
<td>6.7</td>
<td>3.7</td>
<td>21.0</td>
</tr>
<tr>
<td></td>
<td>CVaR$_\alpha$</td>
<td>43.5</td>
<td>17.1</td>
<td>12.4</td>
<td>9.1</td>
<td>29.1</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 1$)</td>
<td>30.9</td>
<td>15.0</td>
<td>11.4</td>
<td>8.6</td>
<td>17.6</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.5$)</td>
<td>10.6</td>
<td>7.2</td>
<td>6.7</td>
<td>5.8</td>
<td>5.4</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.1$)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>MIP (5-iter)</td>
<td>17.9</td>
<td>37.9</td>
<td>43.5</td>
<td>55.7</td>
<td>-3.0</td>
</tr>
</tbody>
</table>

MIP performs better for smaller $N(1 - \alpha) \iff$ 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$_\alpha$, %

<table>
<thead>
<tr>
<th>Returns</th>
<th>Algorithm</th>
<th>$N$</th>
<th>VaR 90%</th>
<th></th>
<th>VaR 95%</th>
<th></th>
<th>VaR 99%</th>
<th></th>
<th>VaR 99.9%</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>1000</td>
<td>5000</td>
<td>10000</td>
<td>20000</td>
<td>1000</td>
<td>5000</td>
<td>10000</td>
<td>20000</td>
</tr>
<tr>
<td>Normal</td>
<td>Alg (5 iter)</td>
<td></td>
<td>3.6</td>
<td>0.6</td>
<td>0.2</td>
<td>0.0</td>
<td>3.8</td>
<td>0.8</td>
<td>0.3</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>Alg (1 iter)</td>
<td></td>
<td><strong>3.6</strong></td>
<td><strong>0.6</strong></td>
<td><strong>0.2</strong></td>
<td>0.0</td>
<td>3.8</td>
<td>0.8</td>
<td><strong>0.3</strong></td>
<td><strong>0.0</strong></td>
</tr>
<tr>
<td></td>
<td>CVaR 90%</td>
<td></td>
<td>5.4</td>
<td>1.5</td>
<td>1.1</td>
<td>0.8</td>
<td>6.2</td>
<td>1.6</td>
<td>0.9</td>
<td>0.5</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 1$)</td>
<td></td>
<td>6.8</td>
<td>2.0</td>
<td>1.3</td>
<td>1.0</td>
<td>7.4</td>
<td>2.0</td>
<td>1.1</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.5$)</td>
<td></td>
<td>8.0</td>
<td>2.1</td>
<td>1.2</td>
<td>0.7</td>
<td>7.5</td>
<td>2.0</td>
<td>1.1</td>
<td>0.6</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.1$)</td>
<td></td>
<td>9.8</td>
<td>2.9</td>
<td>1.7</td>
<td>0.8</td>
<td>9.0</td>
<td>2.8</td>
<td>1.6</td>
<td>0.9</td>
</tr>
<tr>
<td></td>
<td>MIP</td>
<td></td>
<td>13.0</td>
<td>11.7</td>
<td>13.7</td>
<td>14.1</td>
<td>9.8</td>
<td>7.6</td>
<td>9.5</td>
<td>10.7</td>
</tr>
<tr>
<td>Non-Normal</td>
<td>Alg (5 iter)</td>
<td></td>
<td><strong>4.4</strong></td>
<td><strong>0.9</strong></td>
<td><strong>0.2</strong></td>
<td>0.0</td>
<td><strong>4.5</strong></td>
<td><strong>1.0</strong></td>
<td><strong>0.4</strong></td>
<td><strong>0.1</strong></td>
</tr>
<tr>
<td></td>
<td>Alg (1 iter)</td>
<td></td>
<td>4.6</td>
<td>1.1</td>
<td>0.4</td>
<td>0.2</td>
<td>4.5</td>
<td>1.1</td>
<td>0.4</td>
<td>0.2</td>
</tr>
<tr>
<td></td>
<td>CVaR 90%</td>
<td></td>
<td>8.9</td>
<td>6.1</td>
<td>5.3</td>
<td>5.0</td>
<td>7.9</td>
<td>3.9</td>
<td>3.2</td>
<td>2.9</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 1$)</td>
<td></td>
<td>10.4</td>
<td>6.4</td>
<td>5.5</td>
<td>5.1</td>
<td>9.4</td>
<td>4.1</td>
<td>3.4</td>
<td>3.0</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.5$)</td>
<td></td>
<td>10.0</td>
<td>4.1</td>
<td>3.7</td>
<td>3.8</td>
<td>8.5</td>
<td>2.8</td>
<td>2.3</td>
<td>2.0</td>
</tr>
<tr>
<td></td>
<td>Heur ($\xi = 0.1$)</td>
<td></td>
<td>12.5</td>
<td>3.1</td>
<td>1.3</td>
<td>0.6</td>
<td>10.1</td>
<td>3.2</td>
<td>1.7</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>MIP</td>
<td></td>
<td>20.1</td>
<td>22.5</td>
<td>31.3</td>
<td>46.9</td>
<td>13.3</td>
<td>8.6</td>
<td>17.3</td>
<td>30.4</td>
</tr>
</tbody>
</table>

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 $\alpha_k$ computed by the Algorithm

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

### Solution times (in seconds) for all algorithms

<table>
<thead>
<tr>
<th>Returns</th>
<th>Algorithm</th>
<th>VaR 90%</th>
<th>VaR 95%</th>
<th>VaR 99%</th>
<th>VaR 99.9%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1000 5000 10000 20000</td>
<td>1000 5000 10000 20000</td>
<td>1000 5000 10000 20000</td>
<td>1000 5000 10000 20000</td>
</tr>
<tr>
<td>Normal</td>
<td>Alg (5 iter)</td>
<td>0.9 19.5 55.8 100.1</td>
<td>0.8 14.7 40.6 84.8</td>
<td>0.8 7.6 24.0 51.5</td>
<td>0.6 12.1 19.5 24.7</td>
</tr>
<tr>
<td></td>
<td>Alg (1 iter)</td>
<td>0.6 17.1 44.4 91.0</td>
<td>0.5 12.0 31.5 72.6</td>
<td>0.3 4.7 15.2 39.0</td>
<td>0.3 3.8 7.2 12.5</td>
</tr>
<tr>
<td></td>
<td>CVaRα</td>
<td>0.4 10.3 28.0 53.0</td>
<td>0.3 7.0 17.7 47.5</td>
<td>0.3 4.0 14.6 23.8</td>
<td>0.3 3.2 4.7 10.0</td>
</tr>
<tr>
<td></td>
<td>Heur (ξ = 1)</td>
<td>1.4 12.7 35.8 65.3</td>
<td>1.3 10.4 23.6 43.1</td>
<td>1.2 6.5 20.6 36.0</td>
<td>1.0 5.7 10.6 21.6</td>
</tr>
<tr>
<td></td>
<td>Heur (ξ = 0.5)</td>
<td>4.6 75.9 151.8 388.0</td>
<td>3.5 62.2 107.3 279.2</td>
<td>2.3 37.3 71.8 163.1</td>
<td>1.1 15.2 30.8 81.2</td>
</tr>
<tr>
<td></td>
<td>Heur (ξ = 0.1)</td>
<td>14.5 247.4 465.4 1167.1</td>
<td>11.8 192.7 367.5 911.0</td>
<td>7.0 126.0 223.1 565.0</td>
<td>1.1 48.4 102.4 263.9</td>
</tr>
<tr>
<td></td>
<td>MIP</td>
<td>1805.7 1800.4 1801.1 1823.7</td>
<td>1807.3 1800.4 1800.9 1847.4</td>
<td>1812.7 1800.3 1800.7 1802.3</td>
<td>13.1 1396.0 1801.1 1801.3</td>
</tr>
<tr>
<td>Non-Normal</td>
<td>Alg (5 iter)</td>
<td>0.9 24.2 62.4 82.8</td>
<td>0.7 16.9 44.8 82.2</td>
<td>0.7 8.5 24.9 53.0</td>
<td>0.5 11.0 18.4 23.2</td>
</tr>
<tr>
<td></td>
<td>Alg (1 iter)</td>
<td>0.6 19.1 41.9 47.0</td>
<td>0.4 13.5 30.9 59.4</td>
<td>0.3 5.4 15.2 41.4</td>
<td>0.3 3.4 5.3 11.8</td>
</tr>
<tr>
<td></td>
<td>CVaRα</td>
<td>0.4 11.7 26.2 59.1</td>
<td>0.3 7.9 17.7 54.7</td>
<td>0.3 4.1 14.2 24.4</td>
<td>0.3 3.2 4.5 9.3</td>
</tr>
<tr>
<td></td>
<td>Heur (ξ = 1)</td>
<td>1.4 15.0 31.8 71.1</td>
<td>1.2 10.7 26.0 72.0</td>
<td>1.2 7.1 20.1 41.2</td>
<td>1.1 5.5 10.2 21.1</td>
</tr>
<tr>
<td></td>
<td>Heur (ξ = 0.5)</td>
<td>4.2 75.2 139.8 384.3</td>
<td>3.4 58.2 105.6 304.3</td>
<td>2.1 35.4 66.3 156.4</td>
<td>1.0 14.0 31.8 76.8</td>
</tr>
<tr>
<td></td>
<td>Heur (ξ = 0.1)</td>
<td>14.0 252.1 453.2 1161.6</td>
<td>11.1 189.7 363.6 970.5</td>
<td>6.6 126.3 223.6 525.2</td>
<td>1.1 46.7 95.2 252.4</td>
</tr>
<tr>
<td></td>
<td>MIP</td>
<td>1806.5 1800.5 1801.5 1837.1</td>
<td>1806.1 1800.4 1801.1 1820.5</td>
<td>1812.2 1800.3 1800.9 1803.0</td>
<td>9.3 1768.2 1800.5 1800.8</td>
</tr>
</tbody>
</table>
Value-at-Risk Constrained Optimization
Value-at-Risk constrained optimization

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

s.t. \[ \text{VaR}_{0.950}(\mathbf{w}) \leq 0.050 \]
\[ \text{VaR}_{0.995}(\mathbf{w}) \leq 0.095 \]
\[ \text{VaR}_{0.950}(\mathbf{w}) \leq 0.055 \]
\[ \text{VaR}_{0.995}(\mathbf{w}) \leq 0.098 \]
\[ \sum_{j=1}^{J} w_j = 1 \]

\[ \mathbf{w} \geq 0 \]

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

\[ J = 100, \quad N = 5000 \]
Value-at-Risk constrained optimization

Objective function

\[ \max_\omega \sum_{j=1}^{J} \mu_j w_j \]

VaR constraints

\[ \text{VaR}_{0.950}(\omega) \leq 0.050 \]
\[ \text{VaR}_{0.995}(\omega) \leq 0.095 \]
\[ \text{VaR}_{0.950}(\omega) \leq 0.055 \]
\[ \text{VaR}_{0.995}(\omega) \leq 0.098 \]
Conclusions
## Optimization in Algo Risk Application

### Objective Function

<table>
<thead>
<tr>
<th>Add Function</th>
<th>Attribute</th>
<th>Scenario Set</th>
<th>Time</th>
<th>Benchmark</th>
<th>Relative Weight</th>
<th>Quantile</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimize Expected Shortfall</td>
<td>PnL</td>
<td>Historical ALL Scenarios</td>
<td>1y</td>
<td>None</td>
<td>None</td>
<td>99</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Add Bound</th>
<th>Attribute</th>
<th>Relationship</th>
<th>Bound</th>
<th>Scenario Set</th>
<th>Time</th>
<th>Benchmark</th>
<th>Quantile</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Expected Shortfall</td>
<td>PnL</td>
<td>=&gt;</td>
<td>300</td>
<td>Historical ALL Scenarios</td>
<td>1d</td>
<td>None</td>
<td>95</td>
<td></td>
</tr>
</tbody>
</table>

### Trade List & Limits

### Global Constraints

<table>
<thead>
<tr>
<th>1 - Parameters</th>
<th>2 - Relationship</th>
<th>3 - Factor</th>
<th>4 - Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aggregation</td>
<td>Asset Type</td>
<td></td>
<td>Adjusted By</td>
</tr>
<tr>
<td>Group</td>
<td>Whole Portfolio</td>
<td>Holdings</td>
<td>All</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>Return</td>
<td>Measure</td>
<td>Expectation</td>
</tr>
<tr>
<td>Scenario Set</td>
<td>Historical ALL Scenarios</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time</td>
<td>1y</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trades</td>
<td>Net</td>
<td>Current Holdings</td>
<td></td>
</tr>
<tr>
<td>Scale</td>
<td>None</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Soft Constraint</td>
<td>None</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Conclusions and future work

- We proposed a **simple Algorithm** for VaR optimization.
- Algorithm works for both **minimizing VaR** and **constraining VaR**.
- **Small number of iterations** (1-5) needed by the Algorithm allows solving relatively large problems.
- **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.
- Algorithm is useful not only for solving VaR optimization problems directly, but also for computing an **initial feasible solution** to start MIP optimization.
- Algorithm is **robust** with respect to the **sampling error**.
- Many robust CVaR optimization techniques can directly utilize the Algorithm to perform **robust VaR optimization**.
- Future work includes applying the Algorithm within **distributionally robust VaR optimization framework**.
Thank you for your time

Contact Information:

Oleksandr Romanko
Research Fellow, Quantitative Research
Oleksandr.Romanko@algorithmics.com
http://www.romanko.ca