Nonlinear Programming Formulation of Chance-Constraints

Andreas Wächter
joint with Alejandra Peña-Ordieres and Jim Luedtke

Department of Industrial Engineering and Management Sciences
Northwestern University
Evanston, IL, USA
andreas.waechter@northwestern.edu

US & Mexico Workshop on Optimization and Its Applications
Huatulco, Mexico
January 9, 2018
Problem Statement

\[
\min_{x \in X} f(x) \\
\text{s.t. } \mathbb{P}_\xi [c(x, \xi) \leq 0] \geq 1 - \alpha
\]

- Random variable \( \xi \) with support \( \Xi \)
- Assume \( f(x) \) and \( c(x, \xi) \) are sufficiently smooth for all \( \xi \in \Xi \)
- \( X \subseteq \mathbb{R}^n \): Captures additional constraints

\[
f(x) : \mathbb{R}^n \rightarrow \mathbb{R} \\
c(x, \xi) : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}^m
\]
Problem Statement

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad \mathbb{P}_\xi[c(x, \xi) \leq 0] \geq 1 - \alpha
\end{align*}
\]

\[f(x) : \mathbb{R}^n \rightarrow \mathbb{R}\]
\[c(x, \xi) : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}^m\]

- Random variable \(\xi\) with support \(\Xi\)
- Assume \(f(x)\) and \(c(x, \xi)\) are sufficiently smooth for all \(\xi \in \Xi\)
- \(X \subseteq \mathbb{R}^n\): Captures additional constraints
- Goal: Formulate as continuous NLP
  - Do not want to assume particular probability distribution
  - Do not want to assume convexity (or convex approximations)
  - Want to avoid combinatorial approach
  - Use powerful NLP algorithms and techniques
Constraint in Probability Space

Impose \( p(x) = \mathbb{P}[c(x, \xi) \leq 0] \geq 1 - \alpha \)

\[
c(x, \xi) = x^2 - 2 + \xi
\]
\[\xi \sim \mathcal{N}(0, 1)\]

Issues

- Linearization is poor approximation
- Always nonconvex
- Used in [Hu, Hong, Zhang 13], [Bremer, Henrion, Möller 15]
- Let $Y$ be a real-valued random variable.
- $(1 - \alpha)$-quantile:

$$Q_{1-\alpha}^Y = \inf \{ y \in \mathbb{R} : \mathbb{P}[Y \leq y] \geq 1 - \alpha \}$$
Quantile Formulation

Let $Y$ be a real-valued random variable ($Y = c(x, \xi)$)

$(1 - \alpha)$-quantile:

$$Q_{1-\alpha}^{c(x,\xi)} = \inf\{y \in \mathbb{R} : \mathbb{P}[c(x, \xi) \leq y] \geq 1 - \alpha\}$$
Let $Y$ be a real-valued random variable ($Y = c(x, \xi)$)

$(1 - \alpha)$-quantile:

$$Q^{c(x, \xi)}_{1-\alpha} = \inf\{y \in \mathbb{R} : \mathbb{P}[c(x, \xi) \leq y] \geq 1 - \alpha\}$$

$$p(x) \geq 1 - \alpha \iff q_{1-\alpha}(x) = Q^{c(x, \xi)}_{1-\alpha} \leq 0$$
Choice of Formulation

\[ p(x) \geq 1 - \alpha \quad \iff \quad q_{1-\alpha}(x) \leq 0 \]

- Quantile formulation more suitable for NLP solver
Sample Average Approximation (SAA)

\[
\min_{x \in X} \ f(x) \\
\text{s.t. } P_{\xi}[c(x, \xi) \leq 0] \geq 1 - \alpha
\]

\[f(x) : \mathbb{R}^n \rightarrow \mathbb{R} \]
\[c(x, \xi) : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}\]

- Finite scenario set \(\hat{\Xi}_N = \{\hat{\xi}_1, \ldots, \hat{\xi}_N\}\) with \(\hat{\xi}_i \in \Xi\) chosen i.i.d.
- For simplicity: \(c_i(x) = c(x, \hat{\xi}_i)\)
Sample Average Approximation (SAA)

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\ 
\text{s.t.} & \quad \mathbb{P}_\xi [c(x, \xi) \leq 0] \geq 1 - \alpha 
\end{align*}
\]

\[
f(x) : \mathbb{R}^n \to \mathbb{R} \\
c(x, \xi) : \mathbb{R}^n \times \Xi \to \mathbb{R}
\]

- Finite scenario set \( \hat{\Xi}_N = \{\hat{\xi}_1, \ldots, \hat{\xi}_N\} \) with \( \hat{\xi}_i \in \Xi \) chosen i.i.d.
- For simplicity: \( c_i(x) = c(x, \hat{\xi}_i) \)

SAA approximation of quantile
- Ensure that \( M = \lceil (1 - \alpha)N \rceil \) constraints are satisfied
Sample Average Approximation (SAA)

\[
\min_{x \in X} f(x) \\
\text{s.t. } \mathbb{P}_\xi[c(x, \xi) \leq 0] \geq 1 - \alpha
\]

\[f(x) : \mathbb{R}^n \rightarrow \mathbb{R}\]
\[c(x, \xi) : \mathbb{R}^n \times \Xi \rightarrow \mathbb{R}\]

- Finite scenario set \(\hat{\Xi}_N = \{\hat{\xi}_1, \ldots, \hat{\xi}_N\}\) with \(\hat{\xi}_i \in \Xi\) chosen i.i.d.
- For simplicity: \(c_i(x) = c(x, \hat{\xi}_i)\)

SAA approximation of quantile

- Ensure that \(M = \lceil (1 - \alpha)N \rceil\) constraints are satisfied
- Scenario vector: \(C(x) = (c_1(x), \ldots, c_N(x))\)
Sample Average Approximation (SAA)

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad \mathbb{P}_\xi [c(x, \xi) \leq 0] \geq 1 - \alpha
\end{align*}
\]

\(f(x) : \mathbb{R}^n \to \mathbb{R}\)
\(c(x, \xi) : \mathbb{R}^n \times \Xi \to \mathbb{R}\)

- Finite scenario set \(\hat{\Xi}_N = \{\hat{\xi}_1, \ldots, \hat{\xi}_N\}\) with \(\hat{\xi}_i \in \Xi\) chosen i.i.d.
- For simplicity: \(c_i(x) = c(x, \hat{\xi}_i)\)

SAA approximation of quantile
- Ensure that \(M = \lceil (1 - \alpha)N \rceil\) constraints are satisfied
- Scenario vector: \(C(x) = (c_1(x), \ldots, c_N(x))\)
- Order constraint values: \(c_{[1]}(x) \leq c_{[2]}(x) \leq \ldots \leq c_{[N]}(x)\)
Sample Average Approximation (SAA)

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad \mathbb{P}_\xi [c(x, \xi) \leq 0] \geq 1 - \alpha
\end{align*}
\]

\[
f(x) : \mathbb{R}^n \to \mathbb{R}
\]
\[
c(x, \xi) : \mathbb{R}^n \times \Xi \to \mathbb{R}
\]

- Finite scenario set \( \widehat{\Xi}_N = \{\widehat{\xi}_1, \ldots, \widehat{\xi}_N\} \) with \( \widehat{\xi}_i \in \Xi \) chosen i.i.d.
- For simplicity: \( c_i(x) = c(x, \widehat{\xi}_i) \)

SAA approximation of quantile

- Ensure that \( M = \lceil (1 - \alpha)N \rceil \) constraints are satisfied
- Scenario vector: \( C(x) = (c_1(x), \ldots, c_N(x)) \)
- Order constraint values: \( c_{[1]}(x) \leq c_{[2]}(x) \leq \ldots \leq c_{[N]}(x) \)
- Empirical quantile: \( \tilde{q}_{1-\alpha}(C(x)) = c_{[M]}(x) \leq 0 \)
Empirical Quantile

- Feasible region for $c(x, \xi) = \xi_1 x_1 + \xi_2 x_2 - 1$  
  $\xi_1, \xi_1 \sim \mathcal{N}(0, 1)$

$N = 200$  
$N = 500$  
$N = 2000$
Empirical Quantile

- Feasible region for \( c(x, \xi) = \xi_1 x_1 + \xi_2 x_2 - 1 \)

\[ \xi_1, \xi_1 \sim \mathcal{N}(0, 1) \]

\( N = 200 \)

\( N = 500 \)

\( N = 2000 \)

Observations:

- Approximation improves as \( N \) increases
- Rough boundary of feasible region results in spurious local minima
- A lot of variance for small \( N \) as \( \xi_i \) are resampled
- \( \tilde{q}_{1-\alpha}(C(x)) \) is non-differentiable
Y[1], ..., Y[N] ordered realizations of a random variable (c(x, ξ_i))

Empirical quantiles: For \( p \in [0, 1] \) define
\[
\tilde{Q}_p^Y = Y_j, \text{ where } j \in \{1, \ldots, N\} \text{ with } \frac{j-1}{N} < p \leq \frac{j}{N}
\]
Variance Reduction using Kernels

- $Y_1, \ldots, Y_N$ ordered realizations of a random variable $(c(x, \xi))$
- Empirical quantiles: For $p \in [0, 1]$ define
  \[
  \tilde{Q}_p^Y = Y_j, \quad \text{where } j \in \{1, \ldots, N\} \text{ with } \frac{j-1}{N} < p \leq \frac{j}{N}
  \]
- Kernel smoothing [Parson 79]
  \[
  Q_{1-\alpha}^{Y,N,h} = \int_0^1 \tilde{Q}_p^Y \frac{1}{h} K \left( \frac{\alpha-p}{h} \right) dp
  \]
Variance Reduction using Kernels

- $Y_1, \ldots, Y_N$ ordered realizations of a random variable $(c(x, \xi_i))$

- Empirical quantiles: For $p \in [0, 1]$ define
  $$\tilde{Q}_p^Y = Y_{[j]}, \text{ where } j \in \{1, \ldots, N\} \text{ with } \frac{j-1}{N} < p \leq \frac{j}{N}$$

- Kernel smoothing [Parson 79]

\[
Q_{1-\alpha}^{Y,N,h} = \int_0^1 \tilde{Q}_p^Y \frac{1}{h} K \left( \frac{\alpha - p}{h} \right) \, dp = \sum_{j=1}^N Y_{[j]} \int_{\frac{j-1}{N}}^{\frac{j}{N}} \frac{1}{h} K \left( \frac{\alpha - p}{h} \right) \, dp
\]
Kernel Estimation of Quantiles

- Approximate \( q_{1-\alpha}(C(x)) = Q_{1-\alpha}^Y \) with \( Y_i = c(x; \xi_i) \) by kernel estimate

\[
q_{1-\alpha}^{N,h}(C(x)) := \sum_{i=1}^{N} w_{i}^{N,h} c[i](x) \leq 0
\]

- The weights depend only on \( N \) and \( h \) and can be precomputed
Kernel Estimation of Quantiles

- Approximate \( q_{1-\alpha}(C(x)) = Q^Y_{1-\alpha} \) with \( Y_i = c(x; \xi_i) \) by kernel estimate

\[
q^{N,h}_{1-\alpha}(C(x)) := \sum_{i=1}^{N} w^{N,h}_i c[i](x) \leq 0
\]

- The weights depend only on \( N \) and \( h \) and can be precomputed
- Epanechnikov kernel: \( K(u) = \frac{3}{4} (1 - u^2) \mathbb{1}_{|u|<1} \)
Feasible Region of Kernel-Quantile Formulation

\[ c(x, \xi) = \xi_1 x_1 + \xi_2 x_2 - 1 \quad \xi_1, \xi_1 \sim \mathcal{N}(0, 1) \quad h \in \{0.05, 0.01\} \]

\[ N = 200 \quad N = 500 \quad N = 2000 \]
Feasible Region of Kernel-Quantile Formulation

\[ c(x, \xi) = \xi_1 x_1 + \xi_2 x_2 - 1 \quad \xi_1, \xi_1 \sim \mathcal{N}(0, 1) \quad h \in \{0.05, 0.01\} \]

Observations:

- “Looks” smoother and convex
- Reduces variance as \( \xi_i \) are resampled
- Large \( h \) creates bias
- \( q_{1-\alpha}^{N,h}(C(x)) \) has more points of non-differentiability
- Usually “solved” well with Knitro, but no proper termination
Smoothed Empirical CDF

- Quantile $Q_p^Y = \inf\{ y \in \mathbb{R} : \Phi(y) \geq p \}$ where $\Phi(y)$ is cdf of $Y$
Smoothed Empirical CDF

- Quantile $Q_p^Y = \inf\{y \in \mathbb{R} : \Phi(y) \geq p\}$ where $\Phi(y)$ is cdf of $Y$
- $Y_1, \ldots, Y_N$ (unordered) realizations of a random variable
- Empirical cdf:
  $$\tilde{\Phi}(y) = \frac{1}{N} \sum_{i=1}^{N} 1_{\geq 0}(y - Y_i)$$
Quantile \( Q_p^Y = \inf \{ y \in \mathbb{R} : \Phi(y) \geq p \} \) where \( \Phi(y) \) is cdf of \( Y \)

\( Y_1, \ldots, Y_N \) (unordered) realizations of a random variable

Empirical cdf:
\[
\tilde{\Phi}(y) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\geq 0}(y - Y_i)
\]

Smoothed cdf [Azzalini 81]:
\[
\Phi^{N,\epsilon}(y) = \frac{1}{N} \sum_{i=1}^{N} K_\epsilon(y - Y_i)
\]

\( K_\epsilon(t) \approx \mathbb{1}_{\geq 0}(t) \) differentiable
Differentiable Quantile Estimates

- \( Y_i = c(x, \xi_i) \)
- Smoothed cdf: \( \Phi^{N,\epsilon}(y) = \frac{1}{N} \sum_{i=1}^{N} K_{\epsilon}(y - c_i(x)) \)
- Resulting quantile estimate \( \hat{q}_{p,\epsilon}^N(C(x)) \) is the solution \( y^* \) of
  \[
p = \Phi^{N,\epsilon}(y)
  \]
Differentiable Quantile Estimates

- $Y_i = c(x, \xi_i)$
- Smoothed cdf: $\Phi^{N, \epsilon}(y) = \frac{1}{N} \sum_{i=1}^{N} K_\epsilon(y - c_i(x))$
- Resulting quantile estimate $\hat{q}_p^{N, \epsilon}(C(x))$ is the solution $y^*$ of

$$p = \frac{1}{N} \sum_{i=1}^{N} K_\epsilon(y - c_i(x))$$

- Implicit function theorem guarantees that $\hat{q}_p^{N, \epsilon}(C(x))$ exists and is differentiable
Differentiable Quantile Estimates

\[ Y_i = c(x, \xi_i) \]

- Smoothened cdf: \[ \Phi_{N, \epsilon}(y) = \frac{1}{N} \sum_{i=1}^{N} K_{\epsilon}(y - c_i(x)) \]

- Resulting quantile estimate \( \hat{q}_{p, \epsilon}(C(x)) \) is the solution \( y^* \)

\[
p = \frac{1}{N} \sum_{i=1}^{N} K_{\epsilon}(y - c_i(x)) + \frac{1}{2N}
\]

- Implicit function theorem guarantees that \( \hat{q}_{p, \epsilon}(C(x)) \) exists and is differentiable (for \( p \in \{0, \frac{1}{N}, \frac{2}{N}, \ldots, \frac{N-1}{N}\} \))
Final Estimate

1. Kernel estimate using empirical quantiles $c_{[i]}(x)$

$$q_{1-\alpha}^{N,h}(C(x)) = \sum_{i=1}^{N} w_{i}^{N,h} c_{[i]}(x)$$
Final Estimate

1. Kernel estimate using empirical quantiles \( c_{[i]}(x) \)

\[
q_{1-\alpha}^{N,h}(C(x)) = \sum_{i=1}^{N} w_i^{N,h} c_{[i]}(x)
\]

2. Smoothed empirical quantile \( \hat{q}_{p}^{N,\epsilon}(C(x)) \), solution of

\[
p = \frac{1}{N} \sum_{i=1}^{N} K_\epsilon(y - c_i(x)) + \frac{1}{2N}
\]
Final Estimate

1. Kernel estimate using empirical quantiles $c_{[i]}(x)$

$$q_{1-\alpha}^{N,h}(C(x)) = \sum_{i=1}^{N} w_{i}^{N,h} c_{[i]}(x)$$

2. Smoothed empirical quantile $\hat{q}_{p}^{N,\epsilon}(C(x))$, solution of

$$p = \frac{1}{N} \sum_{i=1}^{N} K_{\epsilon}(y - c_{i}(x)) + \frac{1}{2N}$$

3. Final estimate

$$q_{1-\alpha}^{N,h,\epsilon}(C(x)) := \sum_{i=1}^{N} w_{i}^{N,h} \hat{q}_{i}^{N,\epsilon}(C(x))$$
Feasible Region with Smoothing ($N = 100$)

Andreas Wächter
Example: Portfolio Optimization

\[
\max_{x \in \mathbb{R}^n} \mathbb{E}[r^T x] = \mu^T x \\
\text{s.t. } \mathbb{P}_r[r^T x \geq -0.05] \geq 0.95 \\
\sum_{i=1}^{n} x_i = 1, \quad x \geq 0
\]

- \( r \sim \mathcal{N}(\mu, \Sigma) \), fixed \( \mu \in \mathbb{R}^{100} \) and \( \Sigma \in \mathbb{R}^{100 \times 100} \) pos. def.
- 100 replications
Example: Portfolio Optimization

\[
\begin{align*}
\max_{x \in \mathbb{R}^n} & \quad \mathbb{E}[r^T x] = \mu^T x \\
\text{s.t.} & \quad \mathbb{P}_r [r^T x \geq -0.05] \geq 0.95 \\
& \quad \sum_{i=1}^n x_i = 1, \quad x \geq 0
\end{align*}
\]

- \( r \sim \mathcal{N}(\mu, \Sigma) \), fixed \( \mu \in \mathbb{R}^{100} \) and \( \Sigma \in \mathbb{R}^{100 \times 100} \) pos. def.
- 100 replications

Iterations with Knitro

<table>
<thead>
<tr>
<th>( N )</th>
<th>( h )</th>
<th>min iter</th>
<th>mean iter</th>
<th>max iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>500</td>
<td>0.05</td>
<td>15</td>
<td>26.7</td>
<td>54</td>
</tr>
<tr>
<td>10,000</td>
<td>0.005</td>
<td>17</td>
<td>29.3</td>
<td>45</td>
</tr>
</tbody>
</table>
Portfolio Optimization

Andreas Wächter

Nonlinear Programming Formulation of Chance-Constraints
Portfolio Optimization

Andreas Wächter

Nonlinear Programming Formulation of Chance-Constraints
Portfolio Optimization

Andreas Wächter

Nonlinear Programming Formulation of Chance-Constraints
Portfolio Optimization

Andreas Wächter

Nonlinear Programming Formulation of Chance-Constraints
Joint Chance Constraints

- Original Problem

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad \mathbb{P}_\xi [c(x, \xi) \leq 0] \geq 1 - \alpha
\end{align*}
\]

- Scenario vector

\[C(x) = (c(x, \xi_1), \ldots, c(x, \xi_N))\]

- NLP formulation

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad q_{1-\alpha}^{N,h,\epsilon}(C(x)) \leq 0
\end{align*}
\]
Joint Chance Constraints

- Original Problem

\[
\min_{x \in X} f(x)
\]
\[
\text{s.t. } P_{\xi} \left[ c_j(x, \xi) \leq 0, \ j = 1, \ldots, m \right] \geq 1 - \alpha
\]

- Scenario vector

\[
C(x) = (c(x, \xi_1), \ldots, c(x, \xi_N))
\]

- NLP formulation

\[
\min_{x \in X} f(x)
\]
\[
\text{s.t. } q_{1-\alpha}^{N,h,\epsilon}(C(x)) \leq 0
\]
Joint Chance Constraints

▶ Original Problem
\[
\begin{align*}
\min_{x \in X} f(x) \\
\text{s.t. } \mathbb{P}_\xi[\hat{c}(x, \xi) \leq 0] \geq 1 - \alpha
\end{align*}
\]
\[
\hat{c}(x, \xi) := \max_j c_j(x, \xi_j)
\]

▶ Scenario vector
\[C(x) = (c(x, \xi_1), \ldots, c(x, \xi_N))\]

▶ NLP formulation
\[
\begin{align*}
\min_{x \in X} f(x) \\
\text{s.t. } q_{1-\alpha}^{N,h,\epsilon}(C(x)) \leq 0
\end{align*}
\]
Joint Chance Constraints

- **Original Problem**

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad \mathbb{P}_\xi[\hat{c}(x, \xi) \leq 0] \geq 1 - \alpha
\end{align*}
\]

\[\hat{c}(x, \xi) := \max_j c_j(x, \xi_i)\]

- **Scenario vector**

\[\hat{C}(x) = (\hat{c}(x, \xi_1), \ldots, \hat{c}(x, \xi_N))\]

- **NLP formulation**

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad q^{N,h,\epsilon}_{1-\alpha}(\hat{C}(x)) \leq 0
\end{align*}
\]

Note: Constraint no longer differentiable
Joint Chance Constraints

- **Original Problem**

  \[
  \min_{x \in X} f(x) \\
  \text{s.t. } \mathbb{P}_\xi[\hat{c}(x, \xi) \leq 0] \geq 1 - \alpha
  \]

- **Scenario vector**

  \[\hat{C}(x) = (\hat{c}(x, \xi_1), \ldots, \hat{c}(x, \xi_N))\]

- **NLP formulation**

  \[
  \min_{x \in X} f(x) \\
  \text{s.t. } q(\hat{C}(x)) \leq 0
  \]

  \[q(\hat{C}(x)) := q^N_{1-\alpha}(\hat{C}(x))\]

  Note: Constraint no longer differentiable
Exact Penalty Function

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad q(\hat{C}(x)) \leq 0
\end{align*}
\]

▶ Exact penalty function:

\[
\phi_\rho(x) = f(x) + \rho \max\{q(\hat{C}(x)), 0\} \quad (+\text{penalty for } X)
\]
Exact Penalty Function

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad q(\hat{C}(x)) \leq 0
\end{align*}
\]

- Exact penalty function:
  \[\phi_{\rho}(x) = f(x) + \rho \max\{q(\hat{C}(x)), 0\}\] (penalty for \(X\))

- Model of scenario vector \(\hat{C}(x) = (\ldots, \max_j c_j(x, \xi_i), \ldots)\)
  \[m_{\hat{C}}(x_k, d) = \left(\ldots, \max_j \left\{c_j(x_k, \xi_i) + \nabla c_j(x_k, \xi_i)^T d\right\}, \ldots\right)\]
Exact Penalty Function

\[
\begin{align*}
\min_{x \in X} & \quad f(x) \\
\text{s.t.} & \quad q(\hat{C}(x)) \leq 0
\end{align*}
\]

- Exact penalty function:
  \[
  \phi_\rho(x) = f(x) + \rho \max\{q(\hat{C}(x)), 0\} \quad (+\text{penalty for } X)
  \]

- Model of scenario vector
  \[
  \hat{C}(x) = (\ldots, \max_j c_j(x, \xi_i), \ldots)
  \]
  \[
  m_{\hat{C}}(x_k, d) = \left(\ldots, \max_j \{c_j(x_k, \xi_i) + \nabla c_j(x_k, \xi_i)^T d\}, \ldots\right)
  \]

- Model of penalty function
  \[
  m_\phi(x_k, d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d + \\
  \rho \max\{q(\hat{C}(x_k)) + \nabla q(\hat{C}(x_k))^T (m_{\hat{C}}(x_k, d) - \hat{C}(x_k)), 0\}
  \]
Trust Region $S\ell_1$QP Algorithm

\[ m_\phi(x_k, d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d + \rho \max \{ q(\hat{C}(x_k)) + \nabla q(\hat{C}(x_k))^T (m_{\hat{C}}(x_k, d) - \hat{C}(x_k)), 0 \} \]

- Can show: \[ |\phi_\rho(x_k) - m_\phi(x_k, d)| = O(\|d\|^2) \]
- Standard $S\ell_1$QP-type trust region algorithm can be applied
Trust Region $\ell_1$QP Algorithm

\[
m_\phi(x_k, d) = f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d + \rho \max\{ q(\hat{C}(x_k)) + \nabla q(\hat{C}(x_k))^T (m_{\hat{C}}(x_k, d) - \hat{C}(x_k)), 0 \}
\]

- Can show: \( |\phi_\rho(x_k) - m_\phi(x_k, d)| = O(\|d\|^2) \)
- Standard $\ell_1$QP-type trust region algorithm can be applied
- QP Subproblem

\[
\begin{align*}
\min_{d, z, t} & \quad f(x_k) + \nabla f(x_k)^T d + \frac{1}{2} d^T B_k d + \rho \, t \\
\text{s.t.} & \quad q(\hat{C}(x_k)) + \nabla q(\hat{C}(x_k))^T Z \leq t \quad [Z = (z_1, \ldots, z_n)] \\
& \quad c_j(x_k, \xi_i) + \nabla c_j(x_k, \xi_i)^T d \leq z_i \quad \text{for all } i, j \\
& \quad t \geq 0, \quad x_k + d \in \tilde{X}, \quad \|d\|_\infty \leq \Delta_k
\end{align*}
\]

- Size of QP proportional to number of nonzeros in $\nabla q(\hat{C}(x_k))$, not $N$
Goal: Formulate chance-constrained problem as NLP
- Constrain the quantile $q_{1-\alpha}(x) \leq 0$
  - Better linearization than $p(x) \geq (1 - \alpha)$
- Sample-based empirical quantile
  - No need to know probability distribution
- Variance reduction using kernel
  - Improves approximation of feasible region, “smoother boundary”
- Quantile estimates using smoothed empirical cdf
  - Constraint becomes differentiable
- Joint chance constraints
  - Exact merit function
  - $S\ell_1$QP-type trust region algorithm