A Sequential Algorithm for Solving Nonlinear Optimization Problems With Chance Constraints

Frank E. Curtis, Lehigh University

joint work with

Andreas Wächter, Northwestern University
Victor M. Zavala, University of Wisconsin–Madison

Modeling and Optimization: Theory and Applications (MOPTA)
Lehigh University, Bethlehem, Pennsylvania, USA

18 August 2016
Outline

Problem Statements

Penalty Function

Algorithm

Experiments

Conclusion
<table>
<thead>
<tr>
<th>Outline</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Problem Statements</strong></td>
</tr>
<tr>
<td>Penalty Function</td>
</tr>
<tr>
<td>Algorithm</td>
</tr>
<tr>
<td>Experiments</td>
</tr>
<tr>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Chance-constrained optimization

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

- Random variable \( \xi \) with associated space \((\Omega, \mathcal{F}, P)\)
- Assume \( f \) and \( c(\cdot, \xi) \) are \( C^1 \) for any realization of \( \xi \)
- Even if \( f \) and \( c \) linear, the feasible region is nonconvex
- Assume \( m = 1 \) (though \( m > 1 \) or multiple chance constraints also OK)
Cardinality-constrained optimization

Sample Average Approximation (SAA):

\[ \Omega = \{\xi_1, \ldots, \xi_N\} \text{ with equal probability} \]

\[
\min_{x \in \mathbb{R}^n} f(x) \\
\text{s.t. } |\{\xi_i \in \Omega : c(x, \xi_i) \leq 0\}| \geq \lceil (1 - \alpha)N \rceil
\]

(CCP)
Cardinality-constrained optimization

- Sample Average Approximation (SAA):

\[ \Omega = \{\xi_1, \ldots, \xi_N\} \text{ with equal probability} \]

\[
\min_{x \in \mathbb{R}^n} f(x) \\
\text{s.t. } |\{c_i(x) \leq 0\}| \geq M \quad \text{(CCP)}
\]

...arise in other applications as well
Propose an SQP-type method for solving problem CCP

- Viable approach even when, e.g., MINLP techniques are intractable
- Like penalty-SQP, sequential minimization of penalty function
- Novel penalty function with “exactness” properties
- Subproblems with linear chance constraints
  - Do not need to “predict” optimal subset of constraints
  - Efficient methods, e.g. [Luedtke 14], [Küçükyavuz 12]
- Global convergence guarantees
- Good performance on test problems
Outline

Problem Statements

Penalty Function

Algorithm

Experiments

Conclusion
Exact penalty function for deterministic optimization

$$\phi(x) = f(x) + \rho \| [c(x)]_+ \|_1$$

$$= f(x) + \rho \sum_{i=1}^{N} \max\{0, c_i(x)\}$$

Under CQ, local mins of $\phi$ correspond to local mins of optimization problem
Exact penalty function for deterministic optimization

\[ \phi(x) = f(x) + \rho \| [c(x)]_+ \|_1 = f(x) + \rho \sum_{i=1}^{N} \max\{0, c_i(x)\} \]

Under CQ, local mins of \( \phi \) correspond to local mins of optimization problem
Exact penalty function for deterministic optimization

\[ \phi(x) = f(x) + \rho \|[c(x)]_+\|_1 \]

\[ = f(x) + \rho \sum_{i=1}^{N} \max\{0, c_i(x)\} \]

Under CQ, local mins of \( \phi \) correspond to local mins of optimization problem
Constraint violation measure for CCP

- Order constraints: \( c_{i(1)}(x) \leq c_{i(2)}(x) \leq \ldots \)
- Violation measure: \( \langle [c(x)]_+ \rangle_M = \sum_{k=1}^{M} \max\{0, c_{i(k)}(x)\} \)
- Here: \( \langle [c(x)]_+ \rangle_M = 0 \)
Constraint violation measure for CCP

- Order constraints: \( c_i(1)(x) \leq c_i(2)(x) \leq \ldots \)
- Violation measure: \( \langle [c(x)]_+ \rangle_M = \sum_{k=1}^{M} \max\{0, c_{i(k)}(x)\} \)
- Here: \( \langle [c(x)]_+ \rangle_M = c_5(x) \)
Constraint violation measure for CCP

- Order constraints: \( c_{i(1)}(x) \leq c_{i(2)}(x) \leq \ldots \)
- Violation measure: \( \langle [c(x)]_+ \rangle_M = \sum_{k=1}^{M} \max\{0, c_{i(k)}(x)\} \)
- Here: \( \langle [c(x)]_+ \rangle_M = c_2(x) + c_5(x) \)
Constraint violation measure for CCP

- Order constraints: \( c_{i(1)}(x) \leq c_{i(2)}(x) \leq \ldots \)
- Violation measure: \( \langle\langle [c(x)]_+ \rangle \rangle_M = \sum_{k=1}^{M} \max\{0, c_{i(k)}(x)\} \)
- Here: \( \langle\langle [c(x)]_+ \rangle \rangle_M = 0 \)
Consider \( \| [c_S(x)]_+ \|_1 \) for different \( S \in \mathcal{G} \)

- Here: \( S = \{1, 3, 5\} \): \( \| [c_S(x)]_+ \|_1 = c_5(x) + c_3(x) \)
Equivalent formulation

\[ \mathcal{S} := \{ S \subseteq \{1, \ldots, N\} : |S| = M \} \quad \text{and} \quad c_S(x) := [c_i(x)]_{i \in S} \]

- Consider \( \| [c_S(x)]_+ \|_1 \) for different \( S \in \mathcal{S} \)
- Here: \( S = \{1, 2, 4\} \): \( \| [c_S(x)]_+ \|_1 = c_4(x) \)
Equivalent formulation

\[ S := \{ S \subseteq \{1, \ldots, N\} : |S| = M \} \quad c_S(x) := [c_i(x)]_{i \in S} \]

- Consider \( \|[c_S(x)]_+\|_1 \) for different \( S \in \mathcal{S} \)
- Here: \( S = \{1, 2, 5\} : \quad \|[c_S(x)]_+\|_1 = c_5(x) \)
Equivalent formulation

$$\mathcal{S} := \{S \subseteq \{1, \ldots, N\} : |S| = M\} \quad c_S(x) := [c_i(x)]_{i \in S}$$

- Consider \(\|c_S(x)\|_1\) for different \(S \in \mathcal{S}\)
- So \(\langle [c(x)]_+ \rangle_M = \min_{S \in \mathcal{S}} \|c_S(x)\|_1\)
Critical scenario selections

\[ \mathcal{S}(x) = \{ S \in \mathcal{S} : c_i(x) \leq V_c(x) \text{ for } i \in S \} \quad V_c(x) = c_i(M)(x) \]

- Then \( \langle [c(x)]_+ \rangle_M = \| [c_S(x)]_+ \|_1 \) for all \( S \in \mathcal{S}(x) \).
- Here: \( \mathcal{S}(x) = \{(1, 2, 5)\} \)
Critical scenario selections

\[ \mathcal{S}(x) = \{ S \in \mathcal{S} : c_i(x) \leq V_c(x) \text{ for } i \in S \} \quad V_c(x) = c_{i(M)}(x) \]

- Then \( \langle [c(x)]_+ \rangle_M = \| [c_S(x)]_+ \|_1 \) for all \( S \in \mathcal{S}(x) \).
- Here: \( \mathcal{S}(x) = \{(1, 2, 4)\} \)
Critical scenario selections

\[ \mathcal{S}(x) = \{ S \in \mathcal{S} : c_i(x) \leq V_c(x) \text{ for } i \in S \} \quad V_c(x) = c_{i(M)}(x) \]

- Then \( \langle [c(x)]_+ \rangle_M = \| [c_S(x)]_+ \|_1 \) for all \( S \in \mathcal{S}(x) \).
- Here: \( \mathcal{S}(x) = \{(1, 2, 5)\} \)
Critical scenario selections

\[ S(x) = \{ S \in S : c_i(x) \leq V_c(x) \text{ for } i \in S \} \quad V_c(x) = c_i(M)(x) \]

- Then \( \langle [c(x)]_+ \rangle_M = \| [c_S(x)]_+ \|_1 \) for all \( S \in S(x) \).
- Here: \( S(x) = \{(1, 2, 4), (1, 2, 5)\} \)
Exact penalty function for CCP

Penalty function for deterministic optimization with constraints in $S \in \mathcal{S}$:

$$\phi_S(x) := f(x) + \rho \|c_S(x)\|_1$$

Penalty function for CCP:

$$\phi(x) = f(x) + \rho \langle [c(x)]_+ \rangle M$$

$$= f(x) + \rho \min_{S \in \mathcal{S}} \|c_S(x)\|_1$$

$$= \min_{S \in \mathcal{S}} f(x) + \rho \|c_S(x)\|_1$$

$$= \min_{S \in \mathcal{S}} \phi_S(x)$$

$$= \phi_{\tilde{S}}(x) \quad \text{(for all } \tilde{S} \in \mathcal{S}(x))$$
Equivalence of minimizers

For \( \rho > 0 \) sufficiently large (and some CQ):

\[
x_\ast \text{ local min of CCP} \\
\updownarrow \\
\langle \langle [c(x_\ast)]_+ \rangle \rangle_M = 0 \text{ and } x_\ast \text{ local min of } \phi_S(x) \text{ for all } S \in \mathcal{S}(x_\ast)
\]

- Here \( \mathcal{S}(x_\ast) = \{(1, 2, 5)\} \) and \( x_\ast \) is local min of CCP
Equivalence of minimizers

For $\rho > 0$ sufficiently large (and some CQ):

$$x_* \text{ local min of CCP}$$

$$\Downarrow$$

$$\langle [c(x_*)^+]_+ \rangle_M = 0 \text{ and } x_* \text{ local min of } \phi_S(x) \text{ for all } S \in \mathcal{G}(x_*)$$

- Here $\mathcal{G}(x_*) = \{(1, 2, 4), (1, 2, 5)\}$ and $x_*$ is NOT local min of CCP
Outline

Problem Statements

Penalty Function

Algorithm

Experiments

Conclusion
Trust-region method for deterministic optimization ($S\ell_1$QP)

\[ \phi(x) = f(x) + \rho \|[c(x)]_+\|_1 \quad \text{ (} g_k = \nabla f(x_k), J_k = \nabla c(x_k)^T \text{)} \]

\[ m_k(d) = f_k + g_k^T d + \frac{1}{2} d^T H_k d + \rho \|[c_k + J_k d]_+\|_1 \]

At iterate $x_k$:

1. Compute step $d_k$ from trust region subproblem (QP):

   \[
   \min_{d \in \mathbb{R}^n} \ m_k(d) \quad \text{s.t.} \quad \|d\|_\infty \leq \delta_k
   \]

2. Define ratio

   \[
   r_k = \frac{\phi(x_k) - \phi(x_k + d_k)}{m_k(0) - m_k(d_k)}
   \]

3. Update iterate and trust region radius

   \[
   \text{If } r_k \geq \mu: \quad x_{k+1} \leftarrow x_k + d_k \quad \text{and} \quad \delta_{k+1} \leftarrow 2\delta_k
   \]

   \[
   \text{If } r_k < \mu: \quad x_{k+1} \leftarrow x_k \quad \text{and} \quad \delta_{k+1} \leftarrow 0.5\|d_k\|_\infty
   \]
Trust-region method for CCP

\[ \phi(x) = f(x) + \rho \langle \langle [c(x)]_+ \rangle \rangle_M \]
\[ m_k(d) = f_k + g_k^T d + \frac{1}{2} d^T H_k d + \rho \langle \langle [c_k + J_k d]_+ \rangle \rangle_M \]

At iterate \( x_k \):

1. Compute step \( d_k \) from trust region subproblem (QCCP):
   \[
   \min_{d \in \mathbb{R}^n} m_k(d) \quad \text{s.t.} \quad \|d\|_{\infty} \leq \delta_k
   \]

2. Define ratio
   \[ r_k = \frac{\phi(x_k) - \phi(x_k + d_k)}{m_k(0) - m_k(d_k)} \]

3. Update iterate and trust region radius
   - If \( r_k \geq \mu \): \( x_{k+1} \leftarrow x_k + d_k \) and \( \delta_{k+1} \leftarrow \max\{2\delta_k, \delta_{\text{reset}}\} \)
   - If \( r_k < \mu \): \( x_{k+1} \leftarrow x_k \) and \( \delta_{k+1} = 0.5\|d_k\|_{\infty} \)

\( \mu \in (0, 1) \) \quad \( \delta_{\text{reset}} > 0 \)
Solving the trust region subproblem

\[
m_k(d) = f_k + g_k^T d + \frac{1}{2} d^T H_k d + \rho \langle [c_k + J_k d] + \rangle_M
\]
\[
= f_k + g_k^T d + \frac{1}{2} d^T H_k d + \rho \min_S \| [c_{S,k} + J_{S,k} d] + \|_1
\]

MIQP formulation:

\[
\begin{align*}
\min_{d,s,z} & \quad f_k + g_k^T d + \frac{1}{2} d^T H_k d + \rho \sum_{i=1}^{N} s_i \\
\text{s.t.} & \quad c_i(x_k) + \nabla c_i(x_k)^T d \leq s_i + \text{bigM}(1 - z_i) \\
& \quad \sum_{i=1}^{N} z_i = M, \quad s \geq 0, \quad \|d\|_\infty \leq \delta_k, \quad z \in \{0,1\}^N
\end{align*}
\]

- big-M formulation is computationally expensive
- Better alternative: Use B&C method of [Luedtke 14]
- Can we reduce the number of scenarios to consider?
ε-critical scenario selections

\[ C(x, \epsilon) = \{ i \in \{1, \ldots, N\} : |c_i(x) - V_c(x)| \leq \epsilon \} \]
\[ \mathcal{N}(x, \epsilon) = \{ i \in \{1, \ldots, N\} : c_i(x) \leq V_c(x) - \epsilon \} \]
\[ \mathcal{S}(x, \epsilon) = \{ S \in \mathcal{S} : \mathcal{N}(x, \epsilon) \subseteq S \text{ and } S \subseteq \mathcal{N}(x, \epsilon) \cup C(x, \epsilon) \} \]

- Here: \( \mathcal{S}(x_*, \epsilon) = \{(1, 2, 4), (1, 2, 5)\} \)
\[ C(x, \epsilon) = \{ i \in \{1, \ldots, N\} : |c_i(x) - V_c(x)| \leq \epsilon \} \]

\[ N(x, \epsilon) = \{ i \in \{1, \ldots, N\} : c_i(x) \leq V_c(x) - \epsilon \} \]

\[ \mathcal{S}(x, \epsilon) = \{ S \in \mathcal{S} : N(x, \epsilon) \subseteq S \text{ and } S \subseteq N(x, \epsilon) \cup C(x, \epsilon) \} \]

- Here: \( \mathcal{S}(x_k, \epsilon) = \{ (1, 2, 4), (1, 2, 5) \} \)
Subproblem with $\epsilon$-critical scenario selections

\[
\begin{align*}
\min_{d \in \mathbb{R}^n} & \quad f_k + g_k^T d + \frac{1}{2} d^T H_k d + \rho \min_{S \in \mathcal{S}(x_k, \epsilon)} \|c_{S,k} + J_{S,k} d\|_1 \\
\text{s.t.} & \quad \|d\|_\infty \leq \delta_k
\end{align*}
\]

- Computational effort can be tuned with $\epsilon$
  - Small $\epsilon$: Easier subproblems
  - Large $\epsilon$: “Broader view” of feasible region

**Theorem 1**

Suppose $f \in C^1$, $c(\cdot, \xi) \in C^1$, $\{H_k\}$ bounded, and $\phi(x_k)$ bounded below. Then, every limit point $x_*$ of $\{x_k\}$ is a stationary point of $\phi_S$ for all $S \in \mathcal{S}(x_*)$. 

Joint chance constraints

Joint chance constraints: $m > 1$

Combined constraint satisfaction:

\[
V_i(x) = \begin{cases} 
\|c_i(x)\|_1 & \text{if } c_i(x) \not\leq 0 \\
\max\{c_{ij}(x) : j = 1, \ldots, m\} & \text{if } c_i(x) \leq 0 
\end{cases}
\]
Outline

Problem Statements

Penalty Function

Algorithm

Experiments

Conclusion
Numerical experiments

- Matlab implementation with AMPL interface (suffixes)
- MIQPs solved by CPLEX
  - take incumbent after 5 min time limit (parallel with 8 threads)
- $H_k$: Hessian of Lagrangian for current scenario selection
- $\gamma \in (0, 1]$: include $\lceil \gamma N \rceil$ scenarios below and above $V_c(x_k)$ in $C$
“Poor” local minimizers

\[
\begin{align*}
\min_{x,z} & \quad z \\
\text{s.t.} & \quad P[0.25x^4 - \frac{1}{3}x^3 - x^2 + 0.2x - 19.5 + \xi_1 x + \xi_1 \xi_0 \leq z] \geq 0.95
\end{align*}
\]

- \( \xi_0 \) from \( U[-12, 12] \) and \( \xi_1 \) from \( U[-3, 3] \)
- \( N = 5,000 \)
“Poor” local minimizers

<table>
<thead>
<tr>
<th>$x_0$</th>
<th>$N$</th>
<th>0.001</th>
<th>0.002</th>
<th>0.005</th>
<th>0.010</th>
<th>0.050</th>
<th>0.100</th>
<th>0.200</th>
<th>1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.5</td>
<td></td>
<td>1.776</td>
<td>1.776</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>-1.0</td>
<td></td>
<td>1.754</td>
<td>1.760</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>-0.5</td>
<td></td>
<td>2.049</td>
<td>1.951</td>
<td>1.814</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td>2.289</td>
<td>2.289</td>
<td>1.814</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>2.382</td>
<td>2.382</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.912</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
</tr>
<tr>
<td>1.5</td>
<td></td>
<td>0.579</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td>0.463</td>
<td>0.463</td>
<td>0.460</td>
<td>0.463</td>
<td>0.463</td>
<td>0.463</td>
<td>0.463</td>
<td>0.463</td>
</tr>
<tr>
<td>$x^{bad}$</td>
<td></td>
<td>2.310</td>
<td>2.289</td>
<td>1.814</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
</tbody>
</table>
“Poor” local minimizers

<table>
<thead>
<tr>
<th>$x_0$</th>
<th>$\gamma$</th>
<th>0.001</th>
<th>0.002</th>
<th>0.005</th>
<th>0.010</th>
<th>0.050</th>
<th>0.100</th>
<th>0.200</th>
<th>1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N$</td>
<td>11</td>
<td>21</td>
<td>51</td>
<td>101</td>
<td>501</td>
<td>751</td>
<td>1251</td>
<td>5000</td>
</tr>
<tr>
<td>-1.5</td>
<td></td>
<td>1.776</td>
<td>1.776</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>-1.0</td>
<td></td>
<td>1.754</td>
<td>1.760</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>-0.5</td>
<td></td>
<td>2.049</td>
<td>1.951</td>
<td>1.814</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>0.0</td>
<td></td>
<td>2.289</td>
<td>2.289</td>
<td>1.814</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
<tr>
<td>0.5</td>
<td></td>
<td>2.382</td>
<td>2.382</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
</tr>
<tr>
<td>1.0</td>
<td></td>
<td>0.912</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.463</td>
<td>0.463</td>
<td>0.463</td>
</tr>
<tr>
<td>1.5</td>
<td></td>
<td>0.579</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
<td>0.460</td>
</tr>
<tr>
<td>2.0</td>
<td></td>
<td>0.463</td>
<td>0.463</td>
<td>0.460</td>
<td>0.460</td>
<td>0.463</td>
<td>0.463</td>
<td>0.463</td>
<td>0.463</td>
</tr>
<tr>
<td>$x_\text{bad}$</td>
<td></td>
<td>2.310</td>
<td>2.289</td>
<td>1.814</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
<td>1.754</td>
</tr>
</tbody>
</table>
Cash flow problem [Dentcheva et al. 03]

- Invest capital over time horizon into different investment options
- Make sure we can pay random liabilities (with 95% prob.)
- Maximize final cash
- Interest rates depend on amount of investment
  - This makes the problem nonconvex
- 5 random instances per size
## Results

| $N$ | $\gamma$ | instances solved | $\gamma$ | $|C_k|$ | changes in $C_k$ | % improve over robust | time in secs |
|-----|----------|------------------|--------|--------|-----------------|----------------------|--------------|
| 500 | 0.001   | 5                | 9.84   | 9.40   | 6.14            | 5.7223 (5)           | 25.95       |
| 500 | 0.005   | 5                | 8.19   | 11.84  | 8.68            | 5.9602 (5)           | 37.45       |
| 500 | 0.050   | 5                | 7.04   | 51.00  | 13.06           | 6.0349 (5)           | 52.96       |
| 500 | 1.000   | 5                | 7.04   | 500.00 | 15.52           | 6.0349 (5)           | 260.08      |
| 1000 | 0.001 | 5                | 15.98  | 10.90  | 15.15           | 7.4040 (5)           | 117.13      |
| 1000 | 0.005 | 5                | 12.29  | 15.22  | 23.72           | 7.7296 (5)           | 147.30      |
| 1000 | 0.050 | 5                | 8.12   | 101.00 | 34.29           | 7.9134 (5)           | 313.17      |
| 1000 | 1.000 | 5                | 7.43   | 1000.00| 57.14           | 7.9134 (5)           | 819.24      |
| 2000 | 0.001 | 5                | 21.94  | 13.84  | 43.67           | 9.7347 (4)           | 652.19      |
| 2000 | 0.005 | 4                | 12.85  | 23.01  | 56.61           | 9.9029 (4)           | 513.20      |
| 2000 | 0.050 | 4                | 9.90   | 201.00 | 88.99           | 10.1009 (4)          | 1099.41     |
| 2000 | 1.000 | 5                | 14.33  | 2000.00| 129.18          | 10.0924 (4)          | 3519.53     |

- 96 vars, 20 chance constr, 150 non-chance constr
- Size for $N = 2000$ and $\gamma = 0.20$:
  - MIQP has about 40,000 constraints and 500 binary variables
Outline

Problem Statements

Penalty Function

Algorithm

Experiments

Conclusion
Conclusions

- SQP-type trust-region algorithm
  - Exact penalty function for chance-constrained NLP
- Computational effort of subproblem can be tuned
  - Small $\epsilon$: Reduce solution time
  - Large $\epsilon$: Escape spurious local solutions
- Potential improvements
  - Use efficient branch-and-cut subproblem (e.g., [Luedtke 12], ...)
  - Adaptive choices of $\rho$ and/or $\epsilon$ or $\gamma$
  - Adaptive sample size
    - At beginning, consider small subset of scenarios and choose large $\epsilon$
    - Refine SAA discretization with proximity to solution
- Heuristic solutions of subproblem
  - Avoid solution of mixed-integer problem
  - E.g., progress along successively improving scenario selections
- Ignore “very satisfied” constraints in subproblem
  - Do not include scenarios with $c_i(x) \ll 0$