A Sampling Method to Chance-constrained Semidefinite Optimization

Chuan XU$^1$, Jianqiang CHENG$^1$ and Abdel LISSE$^1$

$^1$Laboratoire de Recherche en Informatique, Paris

September 18, 2017
Table of Contents

1. Semidefinite programming
2. Chance-constrained semidefinite programming (CCSDP)
3. Sampling method to solve CCSDP
4. Experiment results
Semidefinite Programming

**SDP form**

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad F(x) \succeq 0
\end{align*}
\]

\(c \in \mathbb{R}^m\)

\(F(x) = F_0 + \sum_{i=1}^{m} x_i F_i\)

\(F^0, \ldots, F^m \in \mathbb{R}^{n \times n} \text{ symmetric}\)
Semidefinite Programming

**SDP form**

\[
\begin{align*}
\min & \quad c^T x \\
\text{s.t.} & \quad F(x) \succeq 0
\end{align*}
\]

- \(c \in \mathbb{R}^m\)
- \(F(x) = F_0 + \sum_{i=1}^{m} x_i F_i\)
- \(F_0, \ldots, F^m \in \mathbb{R}^{n \times n}\) symmetric

**positive-semidefinite**

\(F(x) \succeq 0 \Rightarrow \) for all \(z \in \mathbb{R}^n\), \(z^T F(x) z \geq 0\).
Semidefinite Programming

**SDP form**

\[
\begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad F(x) \succeq 0
\end{align*}
\]

- \(c \in \mathbb{R}^m\)
- \(F(x) = F_0 + \sum_{i=1}^{m} x_i F_i\)
- \(F_0, \ldots, F^m \in \mathbb{R}^{n \times n}\) symmetric
- convex, solved efficiently in polynomial time (via interior-point methods)

**positive-semidefinite**

- \(F(x) \succeq 0 \Rightarrow \)
- for all \(z \in \mathbb{R}^n\), \(z^T F(x) z \geq 0\).
Semidefinite Programming

SDP form

\[ \begin{align*}
\text{min} & \quad c^T x \\
\text{s.t.} & \quad F(x) \succeq 0
\end{align*} \]

- convex, solved efficiently in polynomial time (via interior-point methods)
- Usually requires about the same amount of computational resources as LP

\( c \in \mathbb{R}^m \)
\( F(x) = F_0 + \sum_{i=1}^{m} x_i F_i \)
\( F^0, \ldots , F^m \in \mathbb{R}^{n \times n} \) symmetric

positive-semidefinite

\( F(x) \succeq 0 \Rightarrow \)
for all \( z \in \mathbb{R}^n \), \( z^T F(x) z \geq 0 \)
(CCSDP) \( \min\{ f(x) : x \in X, \Pr\{ F(x, \xi) \succeq 0 \} \geq 1 - \epsilon \} \)

- \( x \subseteq \mathbb{R}^n \): a vector of decision variables
- \( X \): a deterministic feasible region
- \( \xi \): a random vector supported by a distribution \( \Xi \subseteq \mathbb{R}^d \)
- \( F : \mathbb{R}^n, \mathbb{R}^d \rightarrow \mathbb{R}^l \): a random vector-valued function
- \( \epsilon \): a risk parameter given by a decision maker

The probabilistic problem in our numerical tests is a bilinear semidefinite program with chance constraints:

\[
F(x, \xi) = A_0(x) + \sum_{i=1}^{m} \xi_i A_i(x) + \sum_{1 \leq j \leq k \leq m} \xi_j \xi_k B_{jk}(x)
\]

- \( A_i, B_{jk} \) are symmetric matrix
Methods to solve chance constrained problem (CCP):

1. Tractable approximation like convex approximation
   Nemirovski and Shapiro [2006a], Nemirovski [2012]

   - In Campi and Garatti [2011], the authors developed a sampling-and-discarding approach and a greedy algorithm to select the constraints to be removed is shown in Pagnoncelli et al. [2012].
   - In Garatti and Campi [2013], a precise procedure of applying this method on control design is presented.
Simulation-based approximation for CCSDP

Scenario Approach

\[(CCP − SA) \min \{ f(x)_{x \in X} : F(x, \xi^i) \succeq 0, \forall i = 1, \ldots, N \}\]

where $N$ is the number of sampling, $\xi^i$ is a random sample.

Big-M semidefinite sampling approach

\[(CCP − BM) \min f(x) \]

\[\begin{align*}
\text{s.t.} & \quad F(x, \xi^i) + y_i M I \succeq 0, \forall i \in 1, \ldots, N \\
& \sum_{i=1}^{N} y_i \leq \epsilon \times N \\
& x \in X, y \in \{0, 1\}^N
\end{align*}\]
Our sampling method starts by solving a relaxed $CCP - BM$ model. As we suppose that the relaxed values of $y$ could help select the constraints to be removed in sampling-and-discarding approach.

Then according to the sorted value of $y_i$, remove the corresponding constraints in $CCP - SA$ and solve the new reduced problem.
Numerical experiments

Discrete-time controlled dynamical system

\[ x(t + 1) = Ax(t) + bu(t) \quad t = 0, 1, \ldots \]
\[ x(0) = \bar{x} \]

Stability: the safe region for \( x \) could be an invariant ellipsoid.
An ellipsoid:

\[ E(Z) = \{ x \in \mathbb{R}^n : x^T Z x \leq 1 \} \]

\( Z \) is a symmetric positive definite matrix.

An invariant ellipsoid: if \( x \in E(Z) \), then \( A(x) + b \in E(Z) \).

Ellipsoid \( E(Z) \) is invariant if and only if there exists a \( \lambda \geq 0 \) such that

\[
\begin{bmatrix}
1 - b^T Z b - \lambda & -b^T Z A \\
-A^T Z b & \lambda Z - A^T Z A
\end{bmatrix} \succeq 0, \quad ||A|| < 1. (\text{Nemirovski [2001]})
\]
Minimum-volume invariant ellipsoid problem (Cheung et al. [2012])

\[
CCMVIE(\lambda) : \quad \max w \\
\text{s.t} \quad w \leq (\det Z)^{1/n} \\
\Pr \left\{ \begin{bmatrix} 1 - b^T Z b - \lambda & -b^T Z A \\
          -A^T Z b & \lambda Z - A^T Z A \end{bmatrix} \succeq 0 \right\} \geq 1 - \epsilon \\
Z \succeq 0
\]

We assume that the system could be disturbed by some random noise.

- \( b \) is corrupted and \( b_i = \bar{b}_i + \rho \xi_i, \forall i = 1, ..., N \) where \( \bar{b} \in \mathbb{R}^N \) is the nominal value.
- \( \rho \geq 0 \) is a fixed parameter to control the level of perturbation.
- \( \xi_i \) is a standard Gaussian random variable of sample \( i \).
Design of The Experiments

1. We test our method on the same instances as in (Cheung et al. [2012]).

2. The number of constraints to be removed is calculated as following (Campi and Garatti [2011]):

\[
k = \lfloor \epsilon N - d + 1 - \sqrt{2\epsilon \ln \frac{(\epsilon N)^{d-1}}{\beta}} \rfloor,
\]

where \(d\) is the dimension of variable \(Z\), \(\beta\) is a confidence parameter.

3. Sample sizes \(N\) are ranging from 400 up to 1400.

4. We vary the ratio of \(k/N\) from 0.03 to 0.05 to study the influence of \(k\) on the result.
We use the average linear size measure, which is defined as $ALS(E(Z)) = \left( Vol_n(E(Z))^{1/n} \right)$, to evaluate the volume of invariant ellipsoid $E(Z)$. The risk parameter $\epsilon$ is set to be 0.05.

**Figure**: Comparison of average linear size
**Comparison of violation ratio**

$Vio$ shows the violation ratio of each solution estimated under 100000 simulated random samples. The risk parameter $\epsilon$ is fixed to 0.05.

**Figure:** Comparison of violation ratio
Comparison of computing time

**Table:** Average CPU time in seconds of calculation

<table>
<thead>
<tr>
<th>CPU time</th>
<th>Data 1</th>
<th>Data 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SC 13.57</td>
<td>SC 251.75</td>
</tr>
<tr>
<td></td>
<td>Greedy 201.5</td>
<td>Greedy 4955.2</td>
</tr>
<tr>
<td></td>
<td>BMSP 23.29</td>
<td>BMSP 521.4</td>
</tr>
</tbody>
</table>

*BMSP* consumes much less CPU time than *Greedy* and almost twice CPU time than scenario approach. But as a counterpart of the CPU time, we obtain better solution than scenario approach.
We introduce a new simulation-based method to solve stochastic chance constrained program.

This method is a combination of Big-M relaxation and a sampling-and-discard method.

We apply this method to semidefinite programming problem in control theory.

The numerical results show that our method provides better solutions within a reasonable CPU time.


