Chubanov’s New Polynomial-Time for Linear Programming and Extensions

Takashi Tsuchiya
(National Graduate Institute for Policy Studies)

Joint work with Lourencó Bruno, Tomonari Kitahara, Masakazu Muramatsu and Takayuki Okuno

Piecewise smooth system and optimization with piecewise linearization
Via algorithmic differentiation
(Shonan Kokusai Mura, Hayama, June 24–28, 2018)
Introduction

- Recently, Chubanov developed another polynomial-time algorithm based on Oracle whose complexity is polynomial in the dimension of variables.
- In this talk, we also extend Chubanov’s algorithm to semi-infinite programming including SDP.
Introduction

- Pena and Soheili developed an interesting rescaling and projection algorithm for the symmetric cone interior feasibility problem combining Chubanov’s idea and their own development.

- While they take a view of “rescaling” of the system, our approach is regarded as the “cutting plane method.” Ours is directly connected to the ellipsoid method, and has better complexity (not all, but in many cases).
Symmetric Cone and Euclidean Jordan Algebra (EJA)

- Symmetric cone
  - Self-dual
  - Homogeneous

- Euclidean Jordan Algebra (EJA)
  - Vector space $V$ equipped with a product satisfying

\[
x \circ y = y \circ x
\]

\[
x \circ ((x^2) \circ y) = ((x^2) \circ (x \circ y)
\]

$\langle \ , \ \rangle$: inner product

\[
\langle x \circ y, z \rangle = \langle x, y \circ z \rangle
\]
Every symmetric cone $\mathcal{K}$ is written as the cone of squares in terms of an EJA:

$$\mathcal{K} = \{ x \mid x = y \circ y \}.$$  

**Example 1 (Second-order cone)**

$$(x_0; x_1), (y_0; y_1) \in V = \mathbb{R} \times \mathbb{R}^{k-1}$$

$$(x_0; x_1) \circ (y_0; y_1) = (x_0y_0 + x_1^T y_1; x_0y_1 + y_0x_1)$$

$$\mathcal{K} = \{ (x_0; x_1) \in \mathbb{R} \times \mathbb{R}^{k-1} \mid \|x_1\| \leq x_0 \}$$
Second-order Cone and Euclidean Jordan Algebra
Symmetric Cone and Euclidean Jordan Algebra

Example 2 (PSD cone)

\[ X, Y \in V = p \times p \text{ symmetric matrices} \]

\[ X \circ Y = \frac{1}{2} (XY + YX) \]

\[ \mathcal{K} = (\text{The set of } p \times p \text{ symmetric positive definite matrices}) \]
Symmetric Cone and Euclidean Jordan Algebra

- Eigenvalue decomposition (PSD case):

\[ X = \sum_{i=1}^{p} \lambda_i u_i u_i^T \]

\[ u_1 u_1^T, \ldots, u_p u_p^T \]: Called Jordan frame

- \( X \in \mathcal{K} \iff \lambda \geq 0 \)

- Rank of EJA: p
Symmetric Cone and Euclidean Jordan Algebra

- **Associated inner product**

  \[ \langle X, Y \rangle = \text{Tr}(XY) = \text{Tr}(X \circ Y) = \sum \lambda_i(X \circ Y) = \|\lambda(X \circ Y)\|_1. \]

- **Identity element**: I.

- \[ \|\lambda(X)\|_1 = \text{Tr}(X) \text{ if } X \in \mathcal{K}. \]
Symmetric Cone and Euclidean Jordan Algebra

- Eigenvalue decomposition (SOC case):

$$X = \lambda_1 c_1 + \lambda_2 c_2$$

$$\lambda_1 = x_0 + \|x_1\|, \quad \lambda_2 = x_0 - \|x_2\|$$

$c_1, c_2$: Called Jordan frame ($c_1 \perp c_2$).

- $X \in \mathcal{K} \iff \lambda \geq 0$

- Rank of EJA: 2 (independent of dim V)
Symmetric Cone and Euclidean Jordan Algebra

- **Associated inner product**

\[ \langle x, y \rangle = \text{Tr}(x \circ y) = \sum_i x_i y_i. \]

(Tr: sum of eigenvalues)

- **Identity element:** \((x_0; x_1) = (1; 0)\)

- \(\|\lambda(x)\|_1 = \text{Tr}(x) = 2x_0 \) if \(x \in \mathcal{K}\).
Second-order Cone and Euclidean Jordan Algebra

\[ c_1 + c_2 = e \]

\[ (1;0) \]
Direct Product Case

\[ V_i = \text{Vector space equipped with EJA} \]
\[ V = V_1 \otimes \ldots \otimes V_n \]
\[ \dim(V) = \dim(V_1) + \ldots + \dim(V_n). \]
\[ d = d_1 + \ldots + d_n, \quad d = \dim(V), \quad d_i = \dim(V_i). \]
Direct Product Case

\[ x, y \in V \implies x \circ y = (x_1 \circ y_1, \ldots, x_n \circ y_n) \]

Eigenvalue of \( x \):

\[ \lambda(x) = (\lambda_1(x), \ldots, \lambda_n(x)) \]

\[ = ((\lambda_{11}, \ldots, \lambda_{1r_1}), \ldots, (\lambda_{n1}, \ldots, \lambda_{nr_n})) \]
Direct Product Case

e_i = The idenity element of EJA associated with \(\mathcal{K}_i\)
\[e = (e_1; \ldots; e_n)\]
\[\mathcal{K} = \mathcal{K}_1 \otimes \ldots \otimes \mathcal{K}_n\]
\[u \in \mathcal{K} \iff u \geq 0.\]
\[\text{rank}(\mathcal{K}) = \text{rank}(\mathcal{K}_1) + \ldots + \text{rank}(\mathcal{K}_n).\]
\[r = r_1 + \ldots + r_n, \quad r = \text{rank}(\mathcal{K}), \quad r_i = \text{rank}(\mathcal{K}_i).\]
\( \lambda_{ij} \): the \( j \)th eigenvalue of \( V_i \) component.

\[
\| x \|_{1, \infty} = \| \lambda(x) \|_{1, \infty} = \max_{i=1}^n \sum_{j=1}^{r_i} | \lambda_{ij} | \\
\| x \|_{\infty, 1} = \| \lambda(x) \|_{\infty, 1} = \sum_{i=1}^n \max_{j=1}^{r_i} | \lambda_{ij} | \\
\| x \|_2 = \| \lambda(x) \|_2 = \sqrt{\sum_{i=1}^n \sum_{j=1}^{r_i} \lambda_{ij}^2} \\
\langle x, y \rangle \leq \| x \|_{1, \infty} \| y \|_{\infty, 1}.
\]

(Generalized Cauchy Shwartz inequality)
Setting

- Feasibility Problem (P)

\[ \text{Find } x \quad Ax = 0, \quad x \succeq 0. \]

- Dual Problem (D)

\[ \text{Find } (s, y) \quad s = A^T y, \quad s \succeq 0. \]
$\varepsilon$-interior feasible solution to (P)

$$Ax = 0, \quad x \succeq \varepsilon e.$$ 

The minimum eigenvalue is greater than or equal to $\varepsilon$ (instead of 0).
Goal

Given $\varepsilon \geq 0$, the algorithm either

- Finds an interior feasible solution to (P)
- Finds a nonzero feasible solution to (D) ((P) does not have an interior feasible solution.)
- Provides a certificate that there is no $\varepsilon$-interior feasible solution to (P). (All feasible solution exists very close to the boundary, or (P) is infeasible.)
Another Formulation

- $P_A$: Projection onto
  $$\text{Ker}(A) = \{x | Ax = 0\}$$

- (P)
  $$\text{Find}_u \quad x = P_A u, \quad x \geq 0.$$  

- (D)
  $$\text{Find}_s \quad 0 = P_A s, \quad s \geq 0.$$
Perform Basic Procedure. Then, one of the following three cases occurs

- Interior feasible solution to (P) is found.
- Nonzero feasible solution to (D) is found.
- A cut vector is found.

Based on the cut vector, we cut an area which does not contain a feasible solution. The volume of the cut area is large enough to ensure polynomiality.
Algorithm Outline

- By using the automorphism transformation of the cone, the problem is reshaped into the form so that the basic procedure can be applied again.

- Repeating this process, in iterations, one concludes that (1) (P) is interior feasible, (2) (P) is NOT interior feasible, i.e., (D) is nonzero feasible, or (3) There exists no $\varepsilon$ interior feasible solution.
Equivalent Formulation

- Define

\[ \mathcal{F} = \{ x \mid Ax = 0, \quad \|x\|_{1,\infty} \leq 1, \quad x \geq 0 \}. \]

- The original problem is interior feasible if and only if \( \mathcal{F} \) contains an interior point.
Truncated Symmetric Cones

$$H(w, v) = \{ x \mid w^T x \leq w^T v \}$$

Standard truncated Symmetric cone

$${\mathcal K}_k \cap H(e_k, e_k/r_k)$$

Obliquely truncated Symmetric cone

$${\mathcal K}_k \cap H(w, v)$$

w (normal)
Automorphism Transformation

- Given an Obliquely truncated symmetric cone, there always exists an automorphism $Q$ of the cone which maps the obliquely truncated symmetric cone to the standard one.
Volume of obliquely truncated symmetric cone and interior feasibility

- There exists minimum bound on the volume of obliquely truncated symmetric cone which contain $\varepsilon$-interior feasible solution. (Truncated symmetric cone cannot be “too small” if it contains $\varepsilon$-interior feasible solution.)
Algorithm Outline

Redline: feasible solution set

Block 1

Block 2

Block 3
Cut: Volume shrunked

Block 1

Block 2

Block 3
Scaled back to the standard form
Cut: Volume shrunked
Cut: Volume further shrinked (blue) in the original coordinate.
Pena and Soheili’s algorithm

Condition number is defined as a quantity of this surface of sphere (implicitly)

At each iteration, condition number of \( \mathcal{K} \cap \{x \mid AG_i x = 0 \} \) is improved or an interior feasible solution is found.
Key observation: If a “cut condition”

\[ 2r_{\text{max}} \sqrt{n} \| P_A y \| \leq \| y_k \|_1 \]

is satisfied for some \( 0 \neq y \in \mathcal{K} \) and \( k \), then, for any \( x \in \mathcal{F} \), we have

\[ x_k \in H(y_k, \frac{e_k}{\rho_k r_k}), \quad \rho_k = \frac{\| y_k \|_1}{r_k \| P_A y \|_2 \sqrt{n}} \]
Proof

\[ \langle x_k, y_k \rangle \leq \langle x, y \rangle = \langle y, P_A x \rangle \]

\[ \leq \|x\|_{1,\infty} \|P_A y\|_{\infty,1} \leq \sqrt{n} \|P_A y\|_2 \leq \frac{\|y_k\|_1}{\rho_k r_k} \]

\[ = \frac{\langle y_k, e_k \rangle}{\rho_k r_k}. \]
Initial Intuition

- It would be nice if the volume is reduced by a constant, like

$$\text{vol}(\mathcal{K}_k \cap H(y_k, \frac{e_k}{\rho_k r_k})) \leq 0.95 \text{vol}(\mathcal{K}_k \cap H(y_k, \frac{e_k}{r_k}))$$

- Unfortunately, we cannot expect this, but, can construct \((w_k, v_k)\) having the following properties suitable for our purpose.
Cut Vector

- $x_k \in H(w_k, v_k)$ if $x \in \mathcal{F}$.

$$\text{vol}(\mathcal{K}_k \cap H(w_k, v_k)) \leq 0.918 \text{vol}(\mathcal{K}_k \cap H(y_k, \frac{e_k}{r_k}))$$

By taking

$$w_k = \beta_1 y_k + \beta_2 e_k, \quad v_k = w_k^{-1}$$
Vol(COE) < Vol(AOB)
\[ \| y_{k1} \| \leq 0.6 y_{k0} \]

Vol(COE) < Vol(AOB)
\[ \| y_{k1} \| > 0.6 y_{k0} \]

\[ w = y_k + \alpha e_k \]

\[ \text{Vol(COE)} > \text{Vol(AOB)} > \text{Vol(FOG)} \]
We can show that the volume of the obliquely truncated symmetric cone cannot be too small if it contains an $\mathcal{E}$ interior feasible solution. So, after $O(r \log \varepsilon^{-1})$ cutting process, we conclude there exists no $\mathcal{E}$ interior feasible solution to this system.
Basic Procedure

- How to find a cut vector?
Basic Procedure

- Initial value: \( y = e/n > 0 \). (Throughout the iteration, \( y^T e = 1 \).)

- If \( P_A y > 0 \), then \( P_A y \) is an interior feasible solution to (P). If \( P_A y = 0 \), then \( y \) is a nonzero solution to the dual.

- If

\[
2r_{\max} \sqrt{n} \| P_A y \| \leq \| y_k \|_1
\]

for some \( k \), then a cut vector is found.
Basic Procedure

- If these conditions are not met,
  (a) Compute $\eta$
  (b) $y^+ = \alpha y + (1 - \alpha)\eta$, \((P_A y)_i \not= 0\).
and repeat.
  $$\alpha = \arg\min_{\alpha'} \| (\alpha' P_A y + (1 - \alpha') P_A \eta) \|$$
Basic Procedure

- In $O(n^3)$ iterations,
  1. An interior feasible solution (P) is found.
  2. Nonzero feasible solution to (D) is found.
  3. A cut vector $y$ is found.
More on the Basic Procedure

- How to construct $\eta$? Let $z = P_A y$.
- We want to have

\[
\langle \eta, e \rangle = 1, \quad \eta \circ \eta = \eta,
\]
\[
\langle \eta, z \rangle = \langle \eta, P_A y \rangle = \lambda_{\text{min}}(z) < 0.
\]

(This is a key relation!)
More on the Basic Procedure

- Since $\langle e, y \rangle = 1$, $\|y\|_{1,\infty} \geq 1/n$.

Therefore, any vector $y$ satisfying

$$2r_{\max}\sqrt{n}\|P_A y\| \leq \frac{1}{n}$$

becomes a cutting vector. This condition is equivalent to

$$\frac{1}{\|P_A y\|^2} \geq 4n^3 r_{\max}^2$$
More on the Basic Procedure

- By using that $\langle \eta, z \rangle \leq 0$, we can show that

\[
\frac{1}{\| P_A y^+ \|^2} - \frac{1}{\| P_A y \|^2} \geq 1.
\]

This implies that

\[
\frac{1}{\| P_A y \|^2} \geq 4n^3 r_{\text{max}}^2
\]

is attained in $O(n^3 r_{\text{max}}^2)$ iterations.
Overall Complexity

\[(A \in R^{m \times n}, \: r : \text{rank}K.)\]

- \# of basic procedure call
  \[O(r \log \varepsilon^{-1})\]

- \# iterations in basic procedure
  \[O(n^3 r_{\text{max}}^2)\]

- \# of arithmetic operations of required for the basic procedure
  \[O(m^2 + m^2 d + n^3 r_{\text{max}}^2 \max(md, c_{\text{min}}))\]

preprocessing (computation of projection matrix)  
cost for one inner iteration
Extension to Symmetric Cone Programming

- The algorithm so far was developed for SOCP (OMS, 2018) is extended to general symmetric cone programming including SDP (MP, 2017).
- So far, closeness to feasibility is measured in the space of cone variables.
- From now on, we develop another polynomial-time algorithm in the space of y-variables.
- This is also based on Chubanov’s work (2017).
Setting

- Feasibility Problem (D)
  \[
  \text{Find } y \quad a_t^T y > 0, \quad (t \in T).
  \]
- \(y: m\) dimensional, \(T\) can be infinite.
- We deal with Linear Semi-infinite Program (LSIP)
- W.O.L.G. \(\|a_t\| = 1\);
SDP Feasibility Problem

Find $y$ such that

$$\sum_{i=1}^{m} A_i y_i \succ 0.$$ 

Find $y$ such that

$$\sum_{i} (v^T A_i v) y_i > 0, \quad \forall 0 \neq v \in \mathbb{R}^n.$$
Setting

- Feasibility Problem (P)

\[
\text{Find } \sum_{t \in T_{sub}} x_t a_t = 0, \quad |T_{sub}| < \infty.
\]

- (P) and (D) cannot be feasible simultaneously.
Assumption

\[ \mathcal{F}_0 = \{ y \in \mathbb{R}^m | a_t^T y > 0, \ (t \in T), \ |y| \leq 1 \} \]

Assumption:
\[ \dim(\mathcal{F}_0) = m, \]
i.e., \( \mathcal{F}_0 \) have positive volume.
Setting

Oracle$(y)$

Input : $y \in R^m$

Output: $t \in T$ such that $a^T_t y \leq 0$, or declare $\forall t \in T \ a^T_t y > 0$. 
Setting (Scaled version)

\[ M : \text{ invertible } m \times m \text{ matrix} \]

\[ \mathcal{F}_0(M) = \{ \tilde{y} \in \mathbb{R}^m | a_t^T M \tilde{y} > 0, \ (t \in T), \ \|\tilde{y}\| \leq 1 \} \]

\[ \dim(\mathcal{F}_0(M)) = m, \]

i.e., \( \mathcal{F}_0(M) \) have positive volume.
Setting (scaled version)

**Oracle**($\tilde{y}, M$)

**Input:** $\tilde{y} \in \mathbb{R}^m$, $M \in \mathbb{R}^{m \times m}$

**Output:** $t \in T$ such that $a_t^T M \tilde{y} \leq 0$, or declare $\forall t \in T$ $a_t^T M \tilde{y} > 0$. 
Given $\varepsilon \geq 0$, the algorithm either

- Finds an interior feasible solution to (P)
- Finds a nonzero feasible solution to (D)
- Provides a certificate that there is no parallelopipede contained in $\mathcal{F}_0$ whose volume is greater than $\varepsilon$. (All feasible solution exists very close to the boundary, or (P) is infeasible.)
Outline

Direct extension of Chubanov’s oracle algorithm to LSIP.

- Basic procedure to find a direction in variable space along which feasible solution is flat or thin.
- Rescaling of variable space to make the feasible region “fat”.
Basic Procedure

- **Input:** Invertible (scaling) matrix $M$
- **Output:** Feasible Solution to (P) or (D), or the index set $T_{sub}$, $|T_{sub}| \leq 2m$ and weights $x_t$, $t \in T_{sub}$ satisfying

$$\left\| \sum_{t \in T_{sub}} x_t \tilde{a}_t \right\| \leq \frac{1}{2 \sqrt{3m^{3/2}}} \cdot \tilde{a}_t = \frac{Ma_t}{\|Ma_t\|}$$

- In particular, the index $\tilde{t}$ corresponds to the largest weight of $x_{\tilde{t}}$, and is exported to the main algorithm.
Consequence of Basic Procedure

- \( \tilde{a}_t^T \tilde{y} \leq \frac{1}{\sqrt{3m}} \) holds for all \( \tilde{y} \in \mathcal{F}_0(M) \).

\[
0 < \frac{\tilde{a}_t^T \tilde{y}}{2m} \leq x_t \tilde{a}_t^T \tilde{y} \leq \sum_{t \in T_{sub}} x_t \tilde{a}_t^T \tilde{y} \leq \frac{1}{2m} \frac{1}{\sqrt{3m}} \| \tilde{y} \|. 
\]

- Basic Procedure finishes in \( O(m^3) \) iterations.
- Provides a clue to rescale the system in the main algorithm.
Basic Procedure

- **Suppose that** \(|T_{sub}| < 2m\)

\[
z = \sum_{t \in T_{sub}} x_t \tilde{a}_t, \quad \sum_{t \in T_{sub}} x_t = 1, \quad x_t \geq 0.
\]

if \(z = 0\), (P) is solved.

- **Call Oracle(z, M).** If \(z\) is dual feasible, done. If not, the oracle returns a violating index \(\hat{t}\) such that \(\tilde{a}_{\hat{t}}^T z < 0\).

- \(T_{sub} := T_{sub} \cup \{\hat{t}\}\)
Basic Procedure

- **Solve**
  \[
  \min_{\alpha} \left\| \alpha z + (1 - \alpha)\tilde{a}_t \right\|
  \]
  \[
  (0 \leq \alpha \leq 1 \text{ since } \tilde{a}_t^T z < 0)
  \]

- **We let for** \( t \in T_{sub} \)
  \[
  x'_t = \alpha x_t \ (t \neq \hat{t}), \quad x'_\hat{t} = \alpha x_{\hat{t}} + 1 - \alpha \ (t = \hat{t})
  \]

- **Note that** \( x' \) satisfies the simplex condition as well.
Basic Procedure

- **If** \(|T_{sub}| > 2m\), we perform an index elimination procedure. (See next page.)

- \[
\frac{1}{\left\| \sum_{t \in T_{sub}} x'_t \tilde{a}_t \right\|}
\]
  is increased at least by 1 for every iteration.

- **If** \[\left\| \sum_{t \in T_{sub}} x'_t \tilde{a}_t \right\| \leq \frac{1}{2\sqrt{3m^{3/2}}}.
\]
  we are done. Let \(x := x'\) and exit.

- **If** not, let \(x := x'\) and go to next iteration.
Index Elimination Procedure

- If $|T_{sub}| > 2m$, we perform an index elimination procedure.
- We find a vertex solution $w$ in $O(m^3)$ operations to the system

$$
\sum_{t \in T_{sub}} w_t \tilde{a}_t = \sum_{t \in T_{sub}} x'_t \tilde{a}_t, \quad \sum_{t \in T_{sub}} w_t = 1, \quad w_t \geq 0.
$$
- We use $w$ instead of $x'$ to keep the system sparse. $T_{sub}$ is also updated as the support of $w$ so that $|T_{sub}| = m + 1$
Main Algorithm

- Execute Basic Procedure with scaling $M_s$
- Basic Procedure finds a feasible solution to (P) or (D), or returns the index $\tilde{t}$.
- Let $\tilde{a}_{\tilde{t}} = M_s^T a_{\tilde{t}} / \| M_s^T a_{\tilde{t}} \|

  $D_{\tilde{t}} = I - \frac{\tilde{a}_{\tilde{t}} \tilde{a}_{\tilde{t}}^T}{2 \| \tilde{a}_{\tilde{t}} \|^2}$

- $M_{s+1} := M_s D_{\tilde{t}}$ and return to scaling step.
Main Algorithm

- An upper bound on the maximum volume of the largest parallelogram contained in $\mathcal{F}_0$ is reduced by a factor of $\sqrt{e}/2$ per iteration of the main algorithm. (Polynomiality of the algorithm.)
Main Algorithm

- \( \text{vol}_P(\mathcal{F}_0(M)) \)
  The maximum volume of the largest parallelogram contained in \( \mathcal{F}_0(M) \)

- We have
  \[
  \text{vol}_P(\mathcal{F}_0(M)) \leq \frac{\sqrt{e}}{2} \text{vol}_P(\mathcal{F}_0(D_t \tilde{M}))
  \]

- Note that, no matter \( M' \) is,
  \[
  \text{vol}_P(\mathcal{F}_0(M')) \leq 1
  \]
Main Algorithm

- Repeating this estimate, we obtain

\[
\text{vol}_P(\mathcal{F}_0) = \text{vol}_P(\mathcal{F}_0(I)) \leq \left(\frac{\sqrt{e}}{2}\right)^k \text{vol}_P(\mathcal{F}_0(D_{\tilde{t}_k} \cdots D_{\tilde{t}_1} I))
\]
For $n \times n$ PSD matrix, the oracle is implemented in $O(n^3)$ arithmetic operations.

In $O((m^2 + mn^2 + n^3)m^3 \log \varepsilon^{-1})$ arithmetic operations, an interior feasible solution can be found, or declares that the volume of the feasible set is smaller than $\varepsilon$. 
In this talk, we extended two Chubanov’s LP algorithm to symmetric cone programming and linear semi-infinite programming.

Improvement of complexity and numerical experiment is an interesting topic for further study.
Acknowledgement

- Tomonari Kitahara (Kyushu University)
- Bruno F. Lourenco (University of Tokyo)
- Masakazu Muramatsu
  (University of Electro-Communications)
- Takayuki Okuno (RIKEN AIP)