Adaptive LP-Newton method for SOCP

Takayuki Okuno[1] Mirai Tanaka[1,2]

[1]RIKEN, [2]ISM

March 29th, 2018
The 4th ISM-ZIB-IMI MODAL Workshop on Mathematical Optimization and Data Analysis

T. Okuno and M. Tanaka:
Extension of the LP-Newton method to SOCPs via semi-infinite representation,
arXiv:1902.01004.

Table of contents

Second-order cone optimization problem (SOCP)

Linear optimization problem (LP)

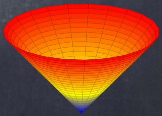
$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

is generalized to second-order cone optimization problem (SOCP)

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \in \mathcal{K} \end{array}$$

where $\mathcal{K} = \mathcal{K}^{n_1} \times \cdots \times \mathcal{K}^{n_p}$

$$\mathcal{K}^n = \{(x_1, \bar{\boldsymbol{x}}) \in \mathbb{R} \times \mathbb{R}^{n-1} : x_1 \geq \|\bar{\boldsymbol{x}}\|_2\}$$



In this slides, $p = 1$ for simplicity

Algorithms for solving SOCP

Basically generarization of algorithms for solving LP

Using Euclidian Jordan algebra

- Interior-point method [Monteiro and Tsuchiya, 2000]
- Chubanov's algorithm [Kitahara and Tsuchiya, 2018]

Using semi-infinite representation

- Simplex method [Hayashi et al., 2016, Muramatsu, 2006]
- LP-Newton method
  [Silvestri and Reinelt, 2017, Okuno and Tanaka, 2019]

Optimality condition

$$\begin{array}{ll} \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, & \boldsymbol{x} \geq \boldsymbol{0} \\ \boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{s} = \boldsymbol{c}, & \boldsymbol{s} \geq \boldsymbol{0} \\ \boldsymbol{x} \circ \boldsymbol{s} = \boldsymbol{0} & \text{(elementwise product)} \end{array}$$

Algorithms for solving SOCP

Basically generarization of algorithms for solving LP

Using Euclidian Jordan algebra

- Interior-point method [Monteiro and Tsuchiya, 2000]
- Chubanov's algorithm [Kitahara and Tsuchiya, 2018]

Using semi-infinite representation

- Simplex method [Hayashi et al., 2016, Muramatsu, 2006]
- LP-Newton method
  [Silvestri and Reinelt, 2017, Okuno and Tanaka, 2019]

Optimality condition

$$
\begin{array}{lll}
\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, & \cancel{\boldsymbol{x} \geq \boldsymbol{0}} & \boldsymbol{x} \in \mathcal{K} \\
\boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{s} = \boldsymbol{c}, & \cancel{\boldsymbol{s} \geq \boldsymbol{0}} & \boldsymbol{s} \in \mathcal{K} \\
\boldsymbol{x} \circ \boldsymbol{s} = \boldsymbol{0} & \cancel{\text{(elementwise product)}} & \text{(Jordan product)}
\end{array}
$$

Algorithms for solving SOCP

Basically generarization of algorithms for solving LP

Using Euclidian Jordan algebra

- Interior-point method [Monteiro and Tsuchiya, 2000]
- Chubanov's algorithm [Kitahara and Tsuchiya, 2018]

Using semi-infinite representation

- Simplex method [Hayashi et al., 2016, Muramatsu, 2006]
- LP-Newton method
  [Silvestri and Reinelt, 2017, Okuno and Tanaka, 2019]

Semi-infinite representation

$$
\begin{array}{ll}
\text{maximize} & \boldsymbol{c}^\top \boldsymbol{x} \\
\text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\
 & (1, \boldsymbol{v}^\top)\boldsymbol{x} \geq 0, \forall \boldsymbol{v} \in \mathbb{R}^{n-1} : \|\boldsymbol{v}\|_2 \leq 1 \\
 & \qquad (\Longleftrightarrow \boldsymbol{x} \in \mathcal{K}^n)
\end{array}
$$

Table of contents

LP-Newton method: Overview

LP-Newton method originally proposed for solving box-constrained LP

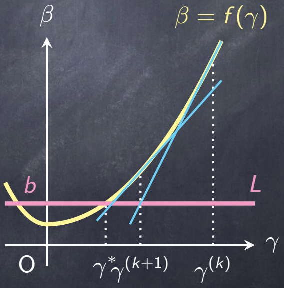
$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} \end{array}$$

by [Fujishige et al., 2009]

"LP" in "the LP-Newton method" means

- line and polytope
- rather than linear program

and "Newton method" means that for solving nonlinear equation $f(\gamma) = b$



Optimal value of LP is endpoint of $L \cap P$

$$\begin{array}{|ll|}\hline \underset{\boldsymbol{x}}{\text{maximize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u} \\ \hline \end{array} \iff \begin{array}{|ll|}\hline \underset{\boldsymbol{x},\gamma}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{c}^\top \boldsymbol{x} = \gamma \\ & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u} \\ \hline \end{array}$$

$$\iff \begin{array}{|ll|}\hline \underset{\gamma}{\text{maximize}} & \gamma \\ \text{subject to} & (\gamma, \boldsymbol{b}) \in P \\ \hline \end{array} \iff \begin{array}{|ll|}\hline \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & (\gamma, \boldsymbol{\beta}) \in L \cap P \\ \hline \end{array}$$

where

$$L := \{(\gamma, \boldsymbol{\beta}) \in \mathbb{R}^{1+m} : \boldsymbol{\beta} = \boldsymbol{b}\}$$
$$P := \{(\gamma, \boldsymbol{\beta}) \in \mathbb{R}^{1+m} : \boldsymbol{c}^\top \boldsymbol{x} = \gamma, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{\beta}, \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u}\}$$

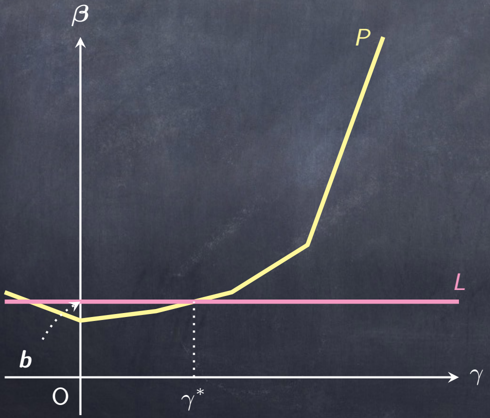
Optimal value γ^* is bounded from above, *i.e.*, $\gamma^* \le \boldsymbol{c}^\top \bar{\boldsymbol{x}}$, where

$$\bar{x}_j := \begin{cases} u_j & (c_j \ge 0) \\ l_j & (c_j < 0) \end{cases} \quad (j = 1, \ldots, n)$$

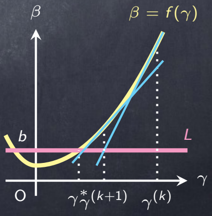
Geometric interpletation of LP-Newton method

LP-Newton method for solving box-constrained LP

$$\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{w} := (\gamma, \boldsymbol{\beta}) \in L \cap P \end{array}$$



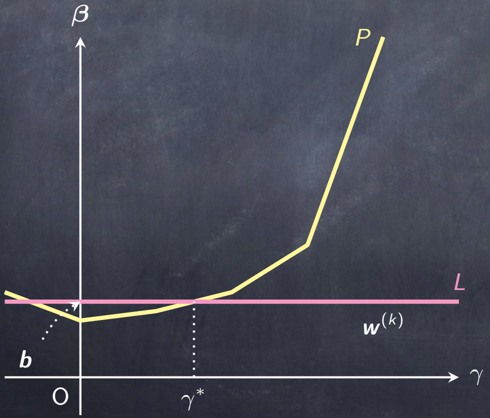
c.f.: Newton method for $f(\gamma) = b$



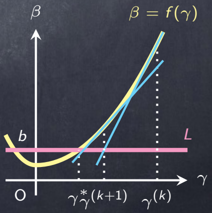
Geometric interpletation of LP-Newton method

LP-Newton method for solving box-constrained LP

$$\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{w} := (\gamma, \boldsymbol{\beta}) \in L \cap P \end{array}$$



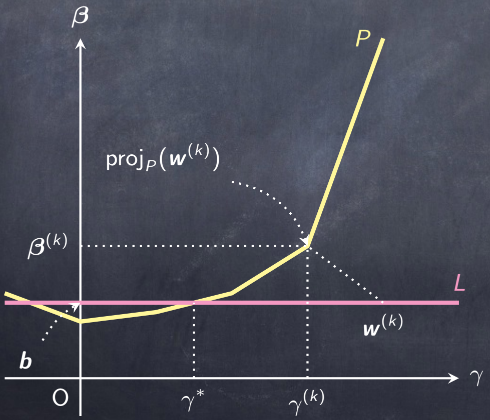
c.f.: Newton method for $f(\gamma) = b$



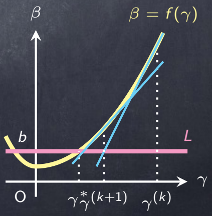
Geometric interpletation of LP-Newton method

LP-Newton method for solving box-constrained LP

$$\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{w} := (\gamma, \boldsymbol{\beta}) \in L \cap P \end{array}$$



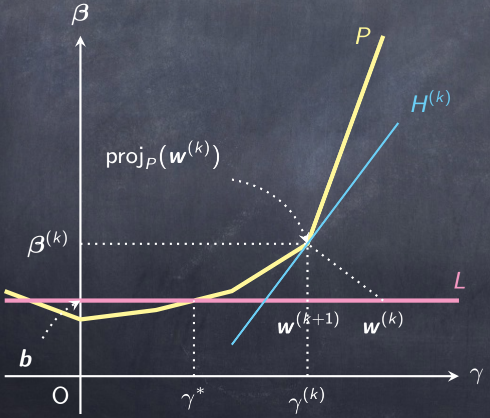
c.f.: Newton method for $f(\gamma) = b$



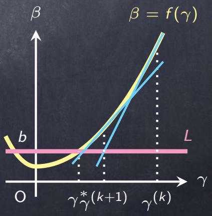
Geometric interpletation of LP-Newton method

LP-Newton method for solving box-constrained LP

$$\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{w} := (\gamma, \boldsymbol{\beta}) \in L \cap P \end{array}$$



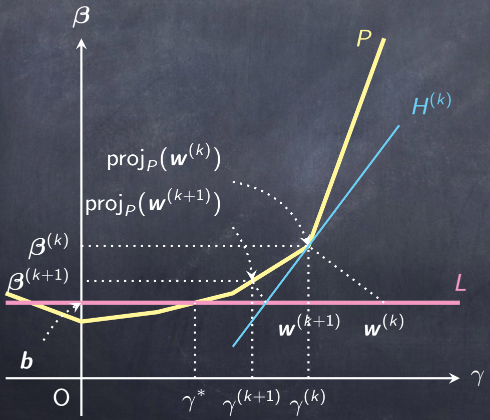
c.f.: Newton method for $f(\gamma) = b$



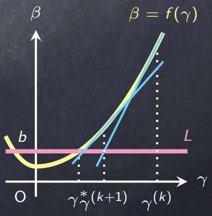
Geometric interpletation of LP-Newton method

LP-Newton method for solving box-constrained LP

$$\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{w} := (\gamma, \boldsymbol{\beta}) \in L \cap P \end{array}$$



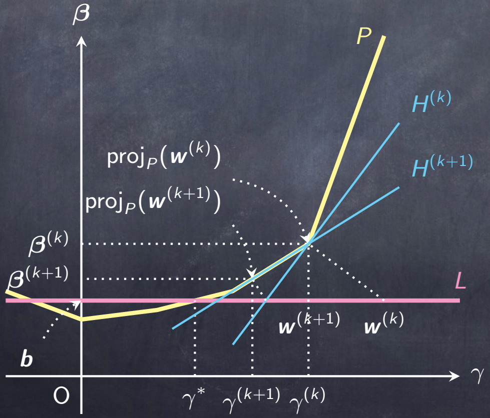
c.f.: Newton method for $f(\gamma) = b$



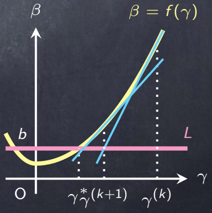
Geometric interpletation of LP-Newton method

LP-Newton method for solving box-constrained LP

$$\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & \boldsymbol{w} := (\gamma, \boldsymbol{\beta}) \in L \cap P \end{array}$$



c.f.: Newton method for $f(\gamma) = b$



LP-Newton method

Algorithm 1 LP-Newton method for solving box-constrained LP

1: Set $\boldsymbol{w}^{(0)} := (\gamma^{(0)}, \boldsymbol{b})$ for sufficiently large $\gamma^{(0)}$, *e.g.*, $\gamma^{(0)} := \boldsymbol{c}^\top \bar{\boldsymbol{x}}$

2: **for** $k = 0, 1, \dots$ (until convergence)

3: Compute the orthogonal projection $(\gamma^{(k)}, \beta^{(k)}) := \mathrm{proj}_P(\boldsymbol{w}^{(k)})$ of the current point $\boldsymbol{w}^{(k)}$ onto polytope P and find $\boldsymbol{x}^{(k)}$ satisfying $\boldsymbol{c}^\top \boldsymbol{x}^{(k)} = \gamma^{(k)}, \boldsymbol{A}\boldsymbol{x}^{(k)} = \beta^{(k)}, \boldsymbol{l} \le \boldsymbol{x}^{(k)} \le \boldsymbol{u}$

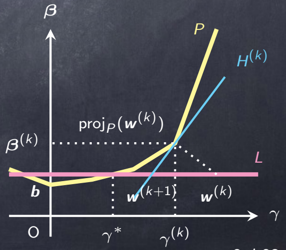
4: Compute the intersecting point $\boldsymbol{w}^{(k+1)}$ of the supporting hyperplane $H^{(k)}$ to P at $\mathrm{proj}_P(\boldsymbol{w}^{(k)})$ and line L

Theorem [Fujishige et al., 2009, Theorem 3.11]

LP-Newton method solves LP in a finite # steps

Remark

- Computation of the intersecting point $\boldsymbol{w}^{(k+1)}$ is easy because $\boldsymbol{w}^{(k)} - \mathrm{proj}_P(\boldsymbol{w}^{(k)}) \perp H^{(k)}$
- Computation of the projection is not trivial



Variants of LP-Newton method and their complexity

Literature	Outer algo	Inner algo (projection)
[Fujishige et al., 2009]	LP-Newton	[Wolfe, 1976]
[Kitahara et al., 2013]	LP-Newton	[Wilhelmsen, 1976]
[Kitahara and Sukegawa, 2019]	bisection	[Wolfe, 1976]

Note

[Kitahara et al., 2013] solves

$$\begin{array}{ll} \text{maximize} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{array}$$

Corresponding P is polyhedral cone

Overall complexity is roughly

$\#$ outer steps $\times$ complexity of inner algo

but still theoretically open because

- complexity of inner algo [Wolfe, 1976, Wilhelmsen, 1976] is unknown
- $\#$ outer LP-Newton steps is finite but unknown
- ($\#$ outer bisection steps is polynomial)

In practice, $\#$ outer LP-Newton step is small ($\lesssim 5$)

Table of contents

How extend LP-Newton method to SOCP?

[Silvestri and Reinelt, 2017] applied the LP-Newton method to conic-box-constrained SOCP

$$\boxed{\begin{array}{ll} \underset{\boldsymbol{x}}{\text{maximize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{l} \preceq \boldsymbol{x} \preceq \boldsymbol{u} \end{array}} \iff \boxed{\begin{array}{ll} \underset{\gamma,\boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & (\gamma, \boldsymbol{\beta}) \in L \cap P^* \end{array}}$$

where

$$\boldsymbol{a} \preceq \boldsymbol{b} \iff \boldsymbol{b} - \boldsymbol{a} \in \mathcal{K}$$

$$P^* := \{(\gamma, \boldsymbol{\beta}) \in \mathbb{R}^{1+m} : \boldsymbol{c}^\top \boldsymbol{x} = \gamma, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{\beta}, \boldsymbol{l} \preceq \boldsymbol{x} \preceq \boldsymbol{u}\}$$

P^* is not polyhedral ⇨ Computation of $\text{proj}_{P^*}(\boldsymbol{w}^{(k)})$ is more challenging

For computing $\text{proj}_{P^*}(\boldsymbol{w}^{(k)})$, [Silvestri and Reinelt, 2017] proposed Frank–Wolfe-type algorithm as inner algorithm ⇦ Time consuming?

Our approach: Polyhedral approximation

Polyhedral approximation via semi-infinite representation

$$\begin{array}{|ll|}\hline \underset{\boldsymbol{x}}{\text{maximize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \in \mathcal{K} \\ \hline \end{array} \iff \begin{array}{|ll|}\hline \underset{\boldsymbol{x}}{\text{maximize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & (1, \boldsymbol{v}^\top)\boldsymbol{x} \geq 0, \forall \boldsymbol{v} \in V^* \\ \hline \end{array}$$

where

$$V^* := \{\boldsymbol{v} \in \mathbb{R}^{n-1} : \|\boldsymbol{v}\|_2 \leq 1\}$$

Of course, $|V^*| = \infty$

Finite approximation, *i.e.*, LP relaxation using $V \subset V^*$ such that $|V| < \infty$

$$\begin{array}{|ll|}\hline \underset{\boldsymbol{x}}{\text{maximize}} & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \\ & (1, \boldsymbol{v}^\top)\boldsymbol{x} \geq 0, \forall \boldsymbol{v} \in V \\ \hline \end{array} \iff \begin{array}{|ll|}\hline \underset{\gamma, \boldsymbol{\beta}}{\text{maximize}} & \gamma \\ \text{subject to} & (\gamma, \boldsymbol{\beta}) \in L \cap P \\ \hline \end{array}$$

where

$$P := \{(\gamma, \boldsymbol{\beta}) \in \mathbb{R}^{1+m} : \boldsymbol{c}^\top \boldsymbol{x} = \gamma, \boldsymbol{A}\boldsymbol{x} = \boldsymbol{\beta}, (1, \boldsymbol{v}^\top)\boldsymbol{x}^{(k)} \geq 0, \forall \boldsymbol{v} \in V\}$$

LP-Newton method (for LP) can be applied to the resulting problem

Adaptive LP-Newton method

Idea: Apply LP-Newton method (for LP) adding cuts adaptively

Algorithm 2 LP-Newton method for box-constrained LP

2: Set $\boldsymbol{w}^{(0)} := (\gamma^{(0)}, \boldsymbol{b})$ for sufficiently large $\gamma^{(0)}$, *e.g.*, $\gamma^{(0)} := \boldsymbol{c}^\top \bar{\boldsymbol{x}}$

3: **for** $k = 0, 1, \ldots$ (until convergence)

4: Compute the orthogonal projection $(\gamma^{(k)}, \beta^{(k)}) = \mathrm{proj}_P(\boldsymbol{w}^{(k)})$ of the current point $\boldsymbol{w}^{(k)}$ onto polytope P and find $\boldsymbol{x}^{(k)}$ corresponding to $(\gamma^{(k)}, \beta^{(k)})$

5: Compute the intersecting point $\boldsymbol{w}^{(k+1)}$ of the supporting hyperplane $H^{(k)}$ to P at $\mathrm{proj}_P(\boldsymbol{w}^{(k)})$ and line L

Adaptive LP-Newton method

Idea: Apply LP-Newton method (for LP) adding cuts adaptively

Algorithm 3 Adaptive LP-Newton method for ~~box-constrained LP~~ SOCP

Generate initial finite approximation $V^{(0)}$ appropriately

Set $\boldsymbol{w}^{(0)} := (\gamma^{(0)}, \boldsymbol{b})$ for sufficiently large $\gamma^{(0)}$, ~~*e.g.*, $\gamma^{(0)} := \boldsymbol{c}^\top \bar{\boldsymbol{x}}$~~

for $k = 0, 1, \ldots$ (until convergence)

  Compute the orthogonal projection $(\gamma^{(k)}, \beta^{(k)}) = \mathrm{proj}_{P^{(k)}}(\boldsymbol{w}^{(k)})$ of the current point $\boldsymbol{w}^{(k)}$ onto polytope $P^{(k)}$ corresponding to $V^{(k)}$ and find $\boldsymbol{x}^{(k)}$ corresponding to $(\gamma^{(k)}, \beta^{(k)})$

  Compute the intersecting point $\boldsymbol{w}^{(k+1)}$ of the supporting hyperplane $H^{(k)}$ to $P^{(k)}$ at $\mathrm{proj}_{P^{(k)}}(\boldsymbol{w}^{(k)})$ and line L

  Compute $\boldsymbol{v}^{(k)} \in \mathrm{argmin}_{\boldsymbol{v} \in V^*} (1, \boldsymbol{v}^\top) \boldsymbol{x}^{(k)}$

  if $(1, \boldsymbol{v}^\top) \boldsymbol{x}^{(k)} < 0$ **then** $\triangleright$ Otherwise, $\boldsymbol{x}^{(k)} \in \mathcal{K}$

    Update $V^{(k+1)} := V^{(k)} \cup \{\boldsymbol{v}^{(k)}\}$

Global convergence

$$
\begin{array}{ll}
\text{maximize} & \boldsymbol{c}^\top \boldsymbol{x} \\
\text{subject to} & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \boldsymbol{x} \in \mathcal{K}
\end{array} \tag{1}
$$

Assumption

The optimal set of SOCP (1) is nonempty and compact

Theorem

Let $\{\boldsymbol{x}^{(k)}\}$ be a sequence generated by the adaptive LP-Newton method. Under the assumption above, any accumuration point of $\{\boldsymbol{x}^{(k)}\}$ is an optimal solution of SOCP (1)

Remark

The assumption above is satisfied if the dual problem

$$
\begin{array}{ll}
\text{minimize} & \boldsymbol{b}^\top \boldsymbol{y} \\
\text{subject to} & \boldsymbol{A}^\top \boldsymbol{y} - \boldsymbol{c} \in \mathcal{K}
\end{array}
$$

of SOCP (1) has an optimal solution and interior feasible solutions

Table of contents

Numerical experiments: Setting

- Performance of our adaptive LP-Newton method
- Comparison with an interiot-point method

Experiment environment

- CentOS 6.10 with 8 Intel Xeon CPUs (3.60 GHz) and 32 GB RAM
- MATLAB R2018a (9.4.0.813654)

Instances

Randomly generated well-conditioned instances

Initial finite approximation of $V^* = \{\boldsymbol{v} \in \mathbb{R}^{n-1} : \|\boldsymbol{v}\|_2 \leq 1\}$

$$V^{(0)} := \{\pm \boldsymbol{e}_j \in \mathbb{R}^{n-1} : j = 1, 2, \ldots, n-1\}$$

Numerical experiments: Setting (cont'd)

Computation of $\mathrm{proj}_{P^{(k)}}(\boldsymbol{w}^{(k)})$

Solve the following QP by using `lsqlin`

$$\begin{array}{ll} \underset{\boldsymbol{x}}{\text{minimize}} & \left\| \begin{pmatrix} \boldsymbol{c}^\top \\ \boldsymbol{A} \end{pmatrix} \boldsymbol{x} - \begin{pmatrix} \gamma^{(k)} \\ \beta^{(k)} \end{pmatrix} \right\|^2 \\ \text{subject to} & (1, \boldsymbol{v}^\top)\boldsymbol{x} \geq 0, \forall \boldsymbol{v} \in V^{(k)} \end{array}$$

Stopping criteria

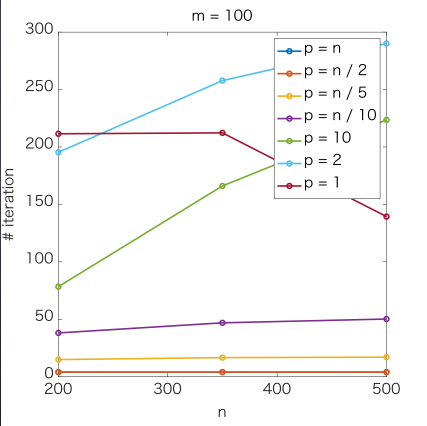
If $\boldsymbol{x}^{(k)}$ satisfies the following criteria,

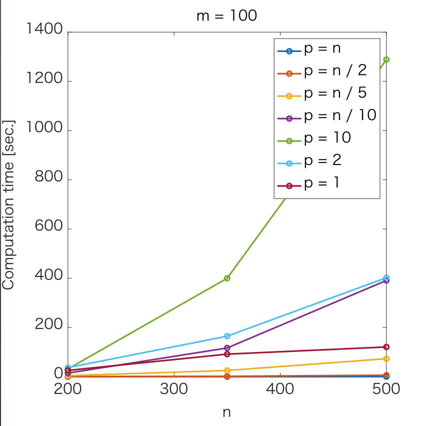
$$\max\{\|\boldsymbol{A}\boldsymbol{x}^{(k)} - \boldsymbol{b}\|, \|\bar{\boldsymbol{x}}^{1,(k)}\| - \boldsymbol{x}_1^{1,(k)}, \ldots, \|\bar{\boldsymbol{x}}^{p,(k)}\| - \boldsymbol{x}_1^{p,(k)}\} \leq 10^{-4}$$

where $\boldsymbol{x}^i \in \mathbb{R}^{n_i}$ denotes the i-th block of $\boldsymbol{x}$ partitioned along the Cartesian structure of $\mathcal{K} = \mathcal{K}^{n_1} \times \ldots \mathcal{K}^{n_p}$, *i.e.*, $\boldsymbol{x} = (\boldsymbol{x}^1, \ldots, \boldsymbol{x}^p)$

of variable vs # iteration & comput. time

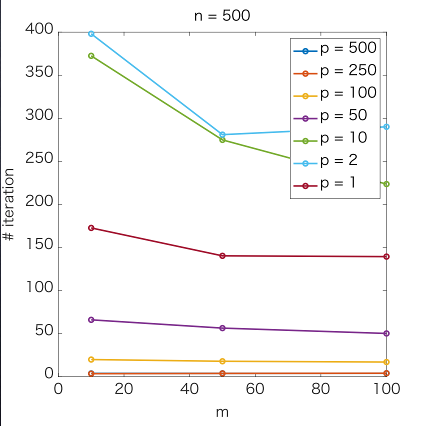
- $m = \#$ linear constr., $n = \#$ variables, $p = \#$ blocks
- Set $n_1 = \cdots = n_p = n/p$, where $(n_1, \ldots, n_p) =$ Cartesian struct. of $\mathcal{K}$
- Average of 10 trials is shown

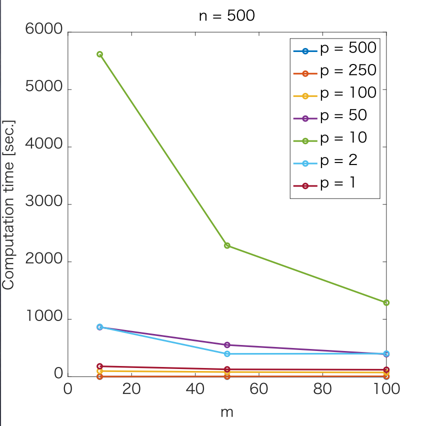




of linear constraints vs # iteration & comput. time

- $m = \#$ linear constr., $n = \#$ variables, $p = \#$ blocks
- Set $n_1 = \cdots = n_p = n/p$, where $(n_1, \ldots, n_p) =$ Cartesian struct. of $\mathcal{K}$
- Average of 10 trials is shown





Comparison with interior-point method

- Used SDPT3 with default setting
- Average of 10 trials is shown

# dimensions			time [sec.]	
m	n	$(n_1, n_2, \ldots, n_p)$	ALPN	SDPT3
1400	1500	$(3, 3, \ldots, 3)$	177.3	366.6
1700	1800	$(3, 3, \ldots, 3)$	260.4	638.4
2000	2100	$(3, 3, \ldots, 3)$	363.4	970.0

Table of contents

Concluding remarks

Contribution

- Proposed adaptive LP-Newton method for solving SOCP
- Used polyhedral approximation of SOC via semi-infinite representation
- Quickly solved instances with "low-dim" SOCs and many linear constr

Future work

- Efficient computation of the projection
- Complexity analysis

Preprint

T. Okuno and M. Tanaka:
Extension of the LP-Newton method to SOCPs via semi-infinite representation,
arXiv:1902.01004.

Thank you for your attention

References I

Fujishige, S., Hayashi, T., Yamashita, K., and Zimmermann, U. (2009).
Zonotopes and the LP-Newton method.
Optimization and Engineering, 10:193–205.

Hayashi, S., Okuno, T., and Ito, Y. (2016).
Simplex-type algorithm for second-order cone programmes via semi-infinite programming reformulation.
Optimization Methods and Software, 31:1272–1297.

Kitahara, T., Mizuno, S., and Shi, J. (2013).
The LP-Newton method for standard form linear programming problems.
Operations Research Letters, 41:426–429.

Kitahara, T. and Sukegawa, N. (2019).
A simple projection algorithm for linear programming problems.
Algorithmica, 81:167–178.

References II

Kitahara, T. and Tsuchiya, T. (2018).
An extension of Chubanov's polynomial-time linear programming algorithm to second-order cone programming.
Optimization Methods and Software, 33:1–25.

Monteiro, R. D. C. and Tsuchiya, T. (2000).
Polynomial convergence of primal-dual algorithms for the second-order cone program based on the MZ-family of directions.
Mathematical Programming, 88:61–83.

Muramatsu, M. (2006).
A pivoting procedure for a class of second-order cone programming.
Optimization Methods and Software, 21:295–315.

Okuno, T. and Tanaka, M. (2019).
Extension of the LP-Newton method to SOCPs via semi-infinite representation.
arXiv:1902.01004.

References III

Silvestri, F. and Reinelt, G. (2017).
The LP-Newton method and conic optimization.
arXiv:1611.09260v2.

Wilhelmsen, D. R. (1976).
A nearest point algorithm for convex polyhedral cones and applications to positive linear approximation.
Mathematics of Computation, 30:48–57.

Wolfe, P. (1976).
Finding the nearest point in a polytope.
Mathematical Programming, 11:128–149.