Iterative approximation schemes for SDPs using the factor width cone

Felix Kirschner, Etienne de Klerk

August 31, 2022

Tilburg University

Introduction

An SDP in primal standard form is given by

Primal SDP

$$\begin{aligned} v_P^* = \inf \ & \langle C, X \rangle \\ \text{s.t.} \ & \langle A_i, X \rangle = b_i \ \forall i \in [m] \\ & X \in \mathbb{S}^n_+. \end{aligned}$$

Introduction

An SDP in primal standard form is given by

Primal SDP

$$
\begin{aligned}
v_P^* = \inf\ & \langle C, X \rangle \\
\text{s.t.}\ & \langle A_i, X \rangle = b_i\ \forall i \in [m] \\
& X \in \mathbb{S}_+^n.
\end{aligned}
$$

- Semidefinite programming is more powerful and versatile than LP or SOCP,

Introduction

An SDP in primal standard form is given by

Primal SDP

$$v_P^* = \inf \langle C, X \rangle \quad \text{s.t.}\ \langle A_i, X \rangle = b_i \ \forall i \in [m] \quad X \in \mathbb{S}_+^n.$$

- Semidefinite programming is more powerful and versatile than LP or SOCP,
- but it is also more expensive (IPM scales as $O(mn^3 + m^2 n^2)$).

Introduction

An SDP in primal standard form is given by

Primal SDP

$$\begin{aligned} v_P^* = \inf \, & \langle C, X \rangle \\ \text{s.t.} \, & \langle A_i, X \rangle = b_i \; \forall i \in [m] \\ & X \in \mathbb{S}_+^n. \end{aligned}$$

- Semidefinite programming is more powerful and versatile than LP or SOCP,
- but it is also more expensive (IPM scales as $O(mn^3 + m^2n^2)$).
- We can solve LPs with millions of variables and constraints,
- while for SDPs we are limited to $\approx$ 10.000 constraints and matrices of size $\approx 1000 \times 1000$.

Let us solve something easier

- Suppose we have a simpler cone $\mathcal{K} \subseteq \mathbb{S}_+^n$.

Let us solve something easier

- Suppose we have a simpler cone $\mathcal{K} \subseteq \mathbb{S}^n_+$.

Solving any instance of

$$\begin{aligned} v_P^{(0)} = \inf \langle C, X \rangle \\ \text{s.t.} \langle A_i, X \rangle = b_i \; \forall i \in [m] \\ X \in \mathcal{K} \subseteq \mathbb{S}^n_+, \end{aligned}$$

yields an upper bound on v_P^*.

What cones to use as $\mathcal{K}$?

Ahmadi and Majumdar[1] considered the cones of diagonally dominant and scaled diagonally dominant matrices.

Definition

A symmetric matrix $S \in \mathbb{S}^n_+$ is said to be *diagonally dominant* if

$$S_{i,i} \geq \sum_{j \neq i} |S_{i,j}| \quad \forall i \in [n].$$

We call $S \in \mathbb{S}^n_+$ *scaled diagonally dominant* if $\exists$ diagonal matrix $D > 0$ such that DSD is diagonally dominant.

[1] [2] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization," 2014 48th Annual Conference on Information Sciences and Systems (CISS), 2014, pp. 1-5, doi: 10.1109/CISS.2014.6814141.

DD_n and SDD_n

We denote

$$DD_n = \{X \in \mathbb{S}^n : X \text{ is diagonally dominant}\} \subseteq \mathbb{S}^n_+$$

$$SDD_n = \{X \in \mathbb{S}^n : X \text{ is scaled diagonally dominant}\} \subseteq \mathbb{S}^n_+.$$

DD_n and SDD_n

We denote

$$DD_n = \{X \in \mathbb{S}^n : X \text{ is diagonally dominant}\} \subseteq \mathbb{S}^n_+$$

$$SDD_n = \{X \in \mathbb{S}^n : X \text{ is scaled diagonally dominant}\} \subseteq \mathbb{S}^n_+.$$

Optimizing over DD_n (resp. SDD_n) is equivalent to an LP (resp. $SOCP$).

DD_n and SDD_n

We denote

$$DD_n = \{X \in \mathbb{S}^n : X \text{ is diagonally dominant}\} \subseteq \mathbb{S}^n_+$$

$$SDD_n = \{X \in \mathbb{S}^n : X \text{ is scaled diagonally dominant}\} \subseteq \mathbb{S}^n_+.$$

Optimizing over DD_n (resp. SDD_n) is equivalent to an LP (resp. $SOCP$).

Example

Let $G = (V, E)$ be a random graph with $n = 50$ vertices. Consider the Lovász-ϑ number of G

$$\vartheta(G) = \min \left\{ t : tI + \sum_{\{i,j\} \in E} y_{i,j} E_{i,j} - J \succeq 0, t \in \mathbb{R}, y \in \mathbb{R}^{|E|} \right\} = 7.6311,$$

DD_n and SDD_n

We denote

$$DD_n = \{X \in \mathbb{S}^n : X \text{ is diagonally dominant}\} \subseteq \mathbb{S}^n_+$$

$$SDD_n = \{X \in \mathbb{S}^n : X \text{ is scaled diagonally dominant}\} \subseteq \mathbb{S}^n_+.$$

Optimizing over DD_n (resp. SDD_n) is equivalent to an LP (resp. $SOCP$).

Example

Let $G = (V, E)$ be a random graph with $n = 50$ vertices. Consider the Lovász-ϑ number of G

$$\vartheta(G) = \min \left\{ t : tI + \sum_{\{i,j\} \in E} y_{i,j} E_{i,j} - J \succeq 0, t \in \mathbb{R}, y \in \mathbb{R}^{|E|} \right\} = 7.6311,$$

Replacing the SDP constraint by a DD_n (resp. SDD_n) constraint lead to an objective value of 33 (resp. 26.298).

Update data matrices

- Ahmadi and Hall[2] first described an iterative method over the DD_n and SDD_n cone.

[2] [1] Ahmadi, A. A., and Hall, G. *Sum of squares basis pursuit with linear and second order cone programming*, in Algebraic and Geometric Methods in Discrete Mathematics, Contemp. Math. 685, Amer. Math. Soc., Providence, RI, 2017, pp. 27-53.

Update data matrices

- Ahmadi and Hall[2] first described an iterative method over the DD_n and SDD_n cone.
- Suppose $\mathcal{K} \in \{DD_n, SDD_n\}$

[2] [1] Ahmadi, A. A., and Hall, G. *Sum of squares basis pursuit with linear and second order cone programming*, in Algebraic and Geometric Methods in Discrete Mathematics, Contemp. Math. 685, Amer. Math. Soc., Providence, RI, 2017, pp. 27-53.

Update data matrices

- Ahmadi and Hall[2] first described an iterative method over the DD_n and SDD_n cone.
- Suppose $\mathcal{K} \in \{DD_n, SDD_n\}$
- and set $C^{(0)} = C$ and $A_i^{(0)} = A_i$ for all $i \in [m]$.

[2] [1] Ahmadi, A. A., and Hall, G. *Sum of squares basis pursuit with linear and second order cone programming*, in Algebraic and Geometric Methods in Discrete Mathematics, Contemp. Math. 685, Amer. Math. Soc., Providence, RI, 2017, pp. 27-53.

Update data matrices

- Ahmadi and Hall[2] first described an iterative method over the DD_n and SDD_n cone.
- Suppose $\mathcal{K} \in \{DD_n, SDD_n\}$
- and set $C^{(0)} = C$ and $A_i^{(0)} = A_i$ for all $i \in [m]$.
- Consider

0th iteration

$$\begin{aligned} v_P^{(0)} = \inf \langle C^{(0)}, X \rangle \\ \text{s.t.} \langle A_i^{(0)}, X \rangle = b_i \ \forall i \in [m] \\ X \in \mathcal{K} \subseteq \mathbb{S}_+^n. \end{aligned} \tag{1}$$

[2] [1] Ahmadi, A. A., and Hall, G. *Sum of squares basis pursuit with linear and second order cone programming*, in Algebraic and Geometric Methods in Discrete Mathematics, Contemp. Math. 685, Amer. Math. Soc., Providence, RI, 2017, pp. 27-53.

Iteration scheme

- Let X^* be optimal. Note that $X^* \in \mathcal{K} \subseteq \mathbb{S}^n_+$.

Iteration scheme

- Let X^* be optimal. Note that $X^* \in \mathcal{K} \subseteq \mathbb{S}^n_+$.
- Let $\mathrm{Chol(X^*)}$ be the Cholesky decomposition of X^*, i.e., $\mathrm{Chol(X^*)}\mathrm{Chol(X^*)}^T = X^*$.

Iteration scheme

- Let X^* be optimal. Note that $X^* \in \mathcal{K} \subseteq \mathbb{S}_+^n$.
- Let $\mathrm{Chol(X^*)}$ be the Cholesky decomposition of X^*, i.e., $\mathrm{Chol(X^*)}\mathrm{Chol(X^*)}^T = X^*$.
- Set
  - $C^{(1)} \leftarrow \mathrm{Chol(X^*)} C^{(0)} \mathrm{Chol(X^*)}$ and
  - $A_i^{(1)} \leftarrow \mathrm{Chol(X^*)} A_i^{(0)} \mathrm{Chol(X^*)}$ for all $i \in [m]$.

Iteration scheme

- Let X^* be optimal. Note that $X^* \in \mathcal{K} \subseteq \mathbb{S}^n_+$.
- Let $\mathrm{Chol(X^*)}$ be the Cholesky decomposition of X^*, i.e., $\mathrm{Chol(X^*)}\mathrm{Chol(X^*)}^T = X^*$.
- Set
  - $C^{(1)} \leftarrow \mathrm{Chol(X^*)} C^{(0)} \mathrm{Chol(X^*)}$ and
  - $A_i^{(1)} \leftarrow \mathrm{Chol(X^*)} A_i^{(0)} \mathrm{Chol(X^*)}$ for all $i \in [m]$.

Consider

1st iteration

$$\begin{aligned} v_P^{(1)} = \inf\ & \langle C^{(1)}, X \rangle \\ \text{s.t. } & \langle A_i^{(1)}, X \rangle = b_i \ \forall i \in [m] \\ & X \in \mathcal{K} \subseteq \mathbb{S}^n_+. \end{aligned} \tag{2}$$

Identity is feasible

Note that $X = I$ is feasible for (2) because

$$\langle A_i^{(1)}, I \rangle = \langle \mathrm{Chol}(\mathrm{X}^*)\, A_i^{(0)}\, \mathrm{Chol}(\mathrm{X}^*), I \rangle = \langle A_i^{(0)}, X^* \rangle = b_i.$$

Identity is feasible

Note that $X = I$ is feasible for (2) because

$$\langle A_i^{(1)}, I\rangle = \langle \mathrm{Chol(X^*)}\, A_i^{(0)}\, \mathrm{Chol(X^*)}, I\rangle = \langle A_i^{(0)}, X^*\rangle = b_i.$$

Moreover, $\langle C^{(1)}, I\rangle = v_P^{(0)}$. Hence,

$$v_P^{(0)} \geq v_P^{(1)} \geq v_P^*.$$

Identity is feasible

Note that $X = I$ is feasible for (2) because

$$\langle A_i^{(1)}, I \rangle = \langle \mathrm{Chol(X^*)} \, A_i^{(0)} \, \mathrm{Chol(X^*)}, I \rangle = \langle A_i^{(0)}, X^* \rangle = b_i.$$

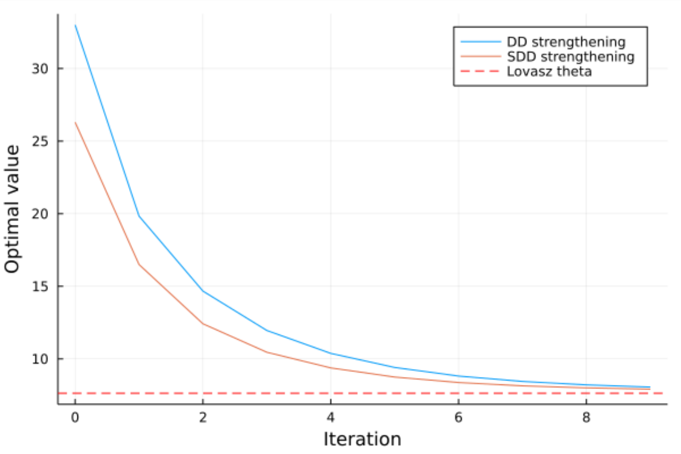
Moreover, $\langle C^{(1)}, I \rangle = v_P^{(0)}$. Hence,

$$v_P^{(0)} \geq v_P^{(1)} \geq v_P^*.$$

- Reiterating this procedure leads to a non-increasing sequence $\{v_P^{(k)}\}_{k \in \mathbb{N}}$.
- Let us compare the performance of the two cones.

SDD_n vs DD_n

Figure 1: Comparison of performance of the iterative scheme using the DD_n and the SDD_n cone on the problem of computing Lovász-ϑ on a random graph with 50 vertices.



Does it converge?

[3] [5] M. Sznaier, B. Roig-Solvas: A globally convergent LP and SOCP-based algorithm for semidefinite programming, arXiv:2202.12374

Does it converge?

- No, not in general.

[3] [5] M. Sznaier, B. Roig-Solvas: A globally convergent LP and SOCP-based algorithm for semidefinite programming, arXiv:2202.12374

Does it converge?

- No, not in general.
- But

Sznaier and Roig-Solvas[3] proposed a two phase interior point method algorithm and proved convergence.

[3] [5] M. Sznaier, B. Roig-Solvas: A globally convergent LP and SOCP-based algorithm for semidefinite programming, arXiv:2202.12374

Does it converge?

- No, not in general.
- But

Sznaier and Roig-Solvas[3] proposed a two phase interior point method algorithm and proved convergence.The two phases consist of

- a decrease phase, solving the relaxations iteratively and updating the data matrices (the procedure we just learned)

[3] [5] M. Sznaier, B. Roig-Solvas: A globally convergent LP and SOCP-based algorithm for semidefinite programming, arXiv:2202.12374

Does it converge?

- No, not in general.
- But

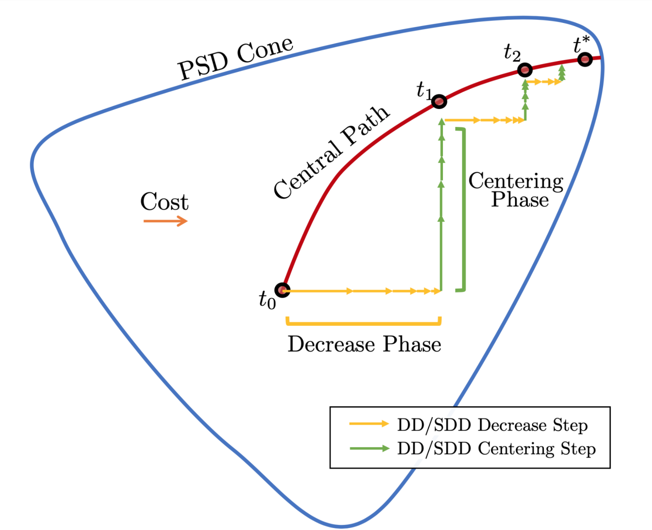
Sznaier and Roig-Solvas[3] proposed a two phase interior point method algorithm and proved convergence.The two phases consist of

- a decrease phase, solving the relaxations iteratively and updating the data matrices (the procedure we just learned)
- and a centering phase, producing feasible solutions with the same objective value that are closer to the central path of the SDP.

[3] [5] M. Sznaier, B. Roig-Solvas: A globally convergent LP and SOCP-based algorithm for semidefinite programming, arXiv:2202.12374

Visualisation

Figure 2: Figure from [5] visualizing the two phases of the algorithm



Generalizing the SDD cone

Lemma

The cone of scaled diagonally dominant matrices can be written as

$$SDD_n = \left\{ \sum_{i \in I} x_i x_i^T : x_i \in \mathbb{R}^n, \mathrm{supp}(x_i) \leq 2 \right\}.$$

This cone is known as the cone of matrices of factor-width 2.

Generalizing the SDD cone

Lemma

The cone of scaled diagonally dominant matrices can be written as

$$SDD_n = \left\{ \sum_{i \in I} x_i x_i^T : x_i \in \mathbb{R}^n, \mathrm{supp}(x_i) \leq 2 \right\}.$$

This cone is known as the cone of matrices of factor-width 2.

Factor-width k cone

We define the factor-width k cone as

$$\mathrm{FW}_n(k) := \left\{ \sum_{i \in I} x_i x_i^T : x_i \in \mathbb{R}^n, \mathrm{supp}(x_i) \leq k \right\}.$$

Clearly, $\mathrm{FW}_n(k) \subseteq \mathbb{S}^n_+$.

Optimizing over the $\mathrm{FW}_n(k)$ cone

- Ding and Lim[4] studied this class of cones and their duals and developed self-concordant barrier functions.

[4] [3] L. Ding, L.-H. Lim, *Higher-order cone programming*, arXiv:1811.05461

Optimizing over the $\mathrm{FW}_n(k)$ cone

- Ding and Lim[4] studied this class of cones and their duals and developed self-concordant barrier functions.
- An optimization problem over the $\mathrm{FW}_n(k)$ cone can be written as an SDP over the cone product

$$\underbrace{\mathbb{S}^k_+ \times \cdots \times \mathbb{S}^k_+}_{\binom{n}{k}}.$$

[4] [3] L. Ding, L.-H. Lim, *Higher-order cone programming*, arXiv:1811.05461

Optimizing over the $\mathrm{FW}_n(k)$ cone

- Ding and Lim[4] studied this class of cones and their duals and developed self-concordant barrier functions.
- An optimization problem over the $\mathrm{FW}_n(k)$ cone can be written as an SDP over the cone product

$$\underbrace{\mathbb{S}_+^k \times \cdots \times \mathbb{S}_+^k}_{\binom{n}{k}}.$$

- For $k = 3$ this involves $O(n^3)$ subcones $\mathbb{S}_+^3$ which could already be too much to handle.

[4] [3] L. Ding, L.-H. Lim, *Higher-order cone programming*, arXiv:1811.05461

Optimizing over the $\mathrm{FW}_n(k)$ cone

- Ding and Lim[4] studied this class of cones and their duals and developed self-concordant barrier functions.
- An optimization problem over the $\mathrm{FW}_n(k)$ cone can be written as an SDP over the cone product

$$\underbrace{\mathbb{S}^k_+ \times \cdots \times \mathbb{S}^k_+}_{\binom{n}{k}}.$$

- For $k = 3$ this involves $O(n^3)$ subcones $\mathbb{S}^3_+$ which could already be too much to handle.
- Algorithm by Sznaier and Roig-Solvas generalizes* to $\mathrm{FW}_n(k)$ for $k \leq n$.

[4] [3] L. Ding, L.-H. Lim, *Higher-order cone programming*, arXiv:1811.05461

Discussion

- It seems $\mathrm{FW}_n(2)$ (SDD_n) is the only practical case so far.

[5] [4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).

[6] [6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control.

Discussion

- It seems $\mathrm{FW}_n(2)$ (SDD_n) is the only practical case so far.
- But even here, each iteration (except the very first) often takes longer than solving the actual SDP.

[5] [4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).

[6] [6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control.

Discussion

- It seems $\mathrm{FW}_n(2)$ (SDD_n) is the only practical case so far.
- But even here, each iteration (except the very first) often takes longer than solving the actual SDP.
- Because after updating the matrices the problem is very dense.

[5] [4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).

[6] [6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control.

Discussion

- It seems $\mathrm{FW}_n(2)$ (SDD_n) is the only practical case so far.
- But even here, each iteration (except the very first) often takes longer than solving the actual SDP.
- Because after updating the matrices the problem is very dense.
- The cones are still interesting. Permenter and Parrilo[5] consider optimizing over $\mathrm{FW}_n(k)$ for facial reduction in SDP.

[5] [4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).

[6] [6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control.

Discussion

- It seems $\mathrm{FW}_n(2)$ (SDD_n) is the only practical case so far.
- But even here, each iteration (except the very first) often takes longer than solving the actual SDP.
- Because after updating the matrices the problem is very dense.
- The cones are still interesting. Permenter and Parrilo[5] consider optimizing over $\mathrm{FW}_n(k)$ for facial reduction in SDP.
- We expect these cones to get more attention in the coming years.

[5][4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).
[6][6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control.

Discussion

- It seems $\mathrm{FW}_n(2)$ (SDD_n) is the only practical case so far.
- But even here, each iteration (except the very first) often takes longer than solving the actual SDP.
- Because after updating the matrices the problem is very dense.
- The cones are still interesting. Permenter and Parrilo[5] consider optimizing over $\mathrm{FW}_n(k)$ for facial reduction in SDP.
- We expect these cones to get more attention in the coming years.
- Another promising idea by Zheng, Sootla[6] is using block-factor width 2 matrices.

[5] [4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).

[6] [6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control.

Block-factor width 2 matrices

The idea is as follows.

- Let $\gamma = \{k_1, \dots, k_p\}$ be a partition of n (i.e. $k_i \in \mathbb{N}$ and $k_1 + \cdots + k_p = n$).
- Let A be a block matrix

$$A = \begin{pmatrix} A_{1,1} & A_{1,2} & \dots & A_{1,p} \\ A_{2,1} & A_{2,2} & \dots & A_{2,p} \\ \vdots & \vdots & \ddots & \vdots \\ A_{p,1} & A_{p,2} & \dots & A_{p,p} \end{pmatrix},$$

  consisting of matrices $A_{i,j} \in \mathbb{R}^{k_i \times k_j}$.
- Define the lift operator $L_{i,j} = \mathbb{R}^{k_i + k_j \times k_i + k_j} \to \mathbb{R}^{n \times n}$

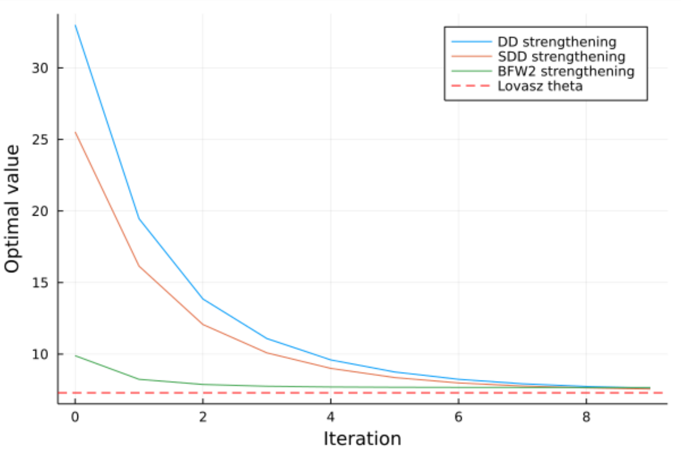
$$\left(L_{i,j} \begin{bmatrix} A_{i,i} & A_{i,j} \\ A_{j,i} & A_{jj} \end{bmatrix} \right)_{k,l} = \begin{cases} A_{i,j} & k \in \{i,j\}, l \in \{i,j\} \\ 0 & \text{otherwise.} \end{cases}$$

Block-factor width 2 matrices (ctd)

- Now replace the psd condition $X \succeq 0$ by
  - $X = \sum_{i<j} L_{i,j} \begin{bmatrix} A_{i,i} & A_{i,j} \\ A_{j,i} & A_{jj} \end{bmatrix}$,
  - such that $\begin{pmatrix} A_{i,i} & A_{i,j} \\ A_{j,i} & A_{j,j} \end{pmatrix} \in \mathbb{S}_+^{k_i+k_j}$.
  - We call X a block-factor width 2 matrix.
- Let $\mathrm{FW}_n(\gamma, 2) = \{X \in \mathbb{S}^n : X \text{ is a block-factor width 2 matrix wrt } \gamma\}$.
- Note that $\mathrm{FW}_n(2) \subseteq \mathrm{FW}_n(\gamma, 2)$ for all partitions γ of n.
- This cone is coarser that $\mathrm{FW}_n(2)$ and scales better than $\mathrm{FW}_n(3)$.

SDD_n vs DD_n vs $\mathrm{FW}_n((10, 10, \ldots, 10), 2)$

Figure 3: Comparison of performance of the iterative scheme using DD_n, SDD_n and $\mathrm{FW}_n((10, 10, \ldots, 10), 2)$ cone on the problem of computing Lovász-ϑ on a random graph with 50 vertices.



References

[1] A. A. Ahmadi, and G. Hall, *Sum of squares basis pursuit with linear and second order cone programming*, in Algebraic and Geometric Methods in Discrete Mathematics, Contemp. Math. 685, Amer. Math. Soc., Providence, RI, 2017, pp. 27-53.

[2] A. A. Ahmadi and A. Majumdar, "DSOS and SDSOS optimization: LP and SOCP-based alternatives to sum of squares optimization," 2014 48th Annual Conference on Information Sciences and Systems (CISS), 2014, pp. 1-5, doi: 10.1109/CISS.2014.6814141.

[3] L. Ding, L.-H. Lim, *Higher-order cone programming*, arXiv:1811.05461

[4] Permenter, F., Parrilo, P. Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone. Math. Program. 171, 1–54 (2018).

[5] M. Sznaier, B. Roig-Solvas: A globally convergent LP and SOCP-based algorithm for semidefinite programming, arXiv:2202.12374

[6] Y. Zheng, A. Sootla and A. Papachristodoulou, "Block Factor-width-two Matrices and Their Applications to Semidefinite and Sum-of-squares Optimization," in IEEE Transactions on Automatic Control, doi: 10.1109/TAC.2022.3151187.

Thank you for your attention

Questions?