An Application of a Facial Reduction Algorithm to Doubly Nonnegative Optimization Problems

Hayato Waki, Mirai Tanaka and Kazuhide Nakata

Institute of Mathematics for Industry, Kyushu University

2012-11-30

Motivation

Diagram on Optimization

Problem

$\Downarrow$

Formulation/Modeling

$\Downarrow$

Algorithm & Software

$\Downarrow$

Verification

Motivation

Diagram on Optimization

Problem

⇓

Formulation/Modeling

⇓

Algorithm & Software

⇓

Verification

- A relaxation technique which use a nonlinear programming

Motivation

Diagram on Optimization

Problem

⇓

Formulation/Modeling

⇓

Algorithm & Software

⇓

Verification

- A relaxation technique which use a nonlinear programming
- Not satisfy conditions for the algorithm & software

Motivation

Diagram on Optimization

Problem
⇓
Formulation/Modeling
⇓
Algorithm & Software
⇓
Verification

- A relaxation technique which use a nonlinear programming
- Not satisfy conditions for the algorithm & software
- Inaccurate solution & slower computation
- Motivation is to recover these difficulties

Relaxation approach

Original Problem

$$\theta^* := \inf_{\mathbf{x}}\{\mathbf{f}(\mathbf{x}) \mid \mathbf{x} \in \mathbf{F}\}$$

Relaxation Problem

$$\tilde{\theta}^* := \inf_{\mathbf{x}}\{\tilde{\mathbf{f}}(\mathbf{x}) \mid \mathbf{x} \in \tilde{\mathbf{F}}\}$$

Relaxation approach

Original Problem

$$\theta^* := \inf_{x}\{f(x) \mid x \in F\}$$

Relaxation Problem

$$\tilde{\theta}^* := \inf_{x}\{\tilde{f}(x) \mid x \in \tilde{F}\}$$

- Construct a relaxation problem
- Can obtain $\tilde{\theta}^*$ by an effective algorithm & software
- Has a nice structure $\Rightarrow \tilde{\theta}^* \leq \theta^*$ (Obtain a lower bound)

Relaxation approach

Original Problem

$$\theta^* := \inf_{x}\{f(x) \mid x \in F\}$$

Relaxation Problem

$$\tilde{\theta}^* := \inf_{x}\{\tilde{f}(x) \mid x \in \tilde{F}\}$$

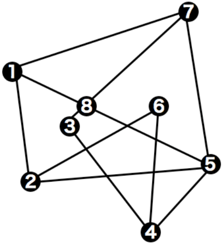
- Construct a relaxation problem
- Can obtain $\tilde{\theta}^*$ by an effective algorithm & software
- Has a nice structure $\Rightarrow \tilde{\theta}^* \leq \theta^*$ (Obtain a lower bound)

A simple observation from relaxation

Find $\hat{x} \in F$ s.t. $f(\hat{x}) - \tilde{\theta}^*$ is small $\Rightarrow$ $\hat{x}$: very close to the opt. sol.

Example from SDP relaxation for Graph Equipartition Problems

- **G(V, E)**: a weighted undirected graph $\Rightarrow$ Partition the vertex set **V** into **L** and **R**
- the minimum total weight of the cut subject to $|\mathbf{L}| = |\mathbf{R}|$
- Example: all weight $=1$



- QOP formulation

$$\inf_{x \in \mathbb{R}^n} \left\{ \frac{1}{2} \sum w_{ij}(1 - x_i x_j) : \sum_{i=1}^{n} x_i = 0, x_i^2 = 1 \ (i = 1, \ldots, n) \right\}$$

Example from SDP relaxation for Graph Equipartition Problems

- SDP relaxation problem: constant matrices $\mathbf{W}$, $\mathbf{E}$ and $\mathbf{E_i}$

$$\inf_{\mathbf{X} \in \mathbb{S}^n_+} \{\mathbf{W} \bullet \mathbf{X} \mid \mathbf{E} \bullet \mathbf{X} = 0, \mathbf{E_i} \bullet \mathbf{X} = 1\}$$

- Inaccurate value & many iterations

Table: SeDuMi 1.21 with ϵ=1.0e-8.

SDPLIB	iter	cpusec	duality gap
gpp124-1	33	19.16	-2.26e-05
gpp250-1	30	74.17	-1.41e-04
gpp500-1	36	304.65	-2.35e-04
gpp124-4	55	43.55	2.0e-08
gpp250-3	61	155.96	8.8e-09
gpp500-3	64	484.45	1.6e-09

Contents & Message

1. SDP relaxation
2. Facial Reduction Approach (FRA)
3. DNN relaxation

Contents & Message

1. SDP relaxation
2. Facial Reduction Approach (FRA)
3. DNN relaxation

Message

- Nonlinear Convex Programming = SDP and DNN
- Hopeless to get accurate solutions for such CPs
- FRA recovers the difficulty

SDP relaxation for Graph Partition Problems

Table: SeDuMi 1.21 with ϵ=1.0e-8.

SDPLIB	SDP iter	SDP cpusec	SDP d.gap	FRA d.gap	FRA cpusec	FRA iter
gpp100	33	16.29	-9.8e-08	-2.1e-09	5.29	17
gpp124-2	29	17.35	-1.2e-07	-1.1e-09	8.95	18
gpp124-3	33	27.97	-1.8e-07	-2.5e-09	11.75	17
gpp124-4	55	43.55	2.1e-08	-3.6e-09	11.98	17
gpp250-2	51	124.47	1.5e-08	-7.5e-09	38.10	18
gpp250-3	61	155.96	8.8e-09	-4.5e-10	39.19	18
gpp250-4	39	106.96	7.1e-08	-6.0e-09	37.30	18
gpp500-2	43	327.53	-1.0e-06	-2.2e-10	143.53	22
gpp500-3	64	484.45	1.6e-09	-6.3e-10	153.30	23
gpp500-4	21	157.55	-1.5e-06	-7.1e-10	170.76	25

SDP relaxation for Graph Partition Problems

Table: SeDuMi 1.21 with ϵ=1.0e-8.

	SDP			FRA		
SDPLIB	iter	cpusec	d.gap	d.gap	cpusec	iter
gpp100	33	16.29	-9.8e-08	-2.1e-09	5.29	17
gpp124-2	29	17.35	-1.2e-07	-1.1e-09	8.95	18
gpp124-3	33	27.97	-1.8e-07	-2.5e-09	11.75	17
gpp124-4	55	43.55	2.1e-08	-3.6e-09	11.98	17
gpp250-2	51	124.47	1.5e-08	-7.5e-09	38.10	18
gpp250-3	61	155.96	8.8e-09	-4.5e-10	39.19	18
gpp250-4	39	106.96	7.1e-08	-6.0e-09	37.30	18
gpp500-2	43	327.53	-1.0e-06	-2.2e-10	143.53	22
gpp500-3	64	484.45	1.6e-09	-6.3e-10	153.30	23
gpp500-4	21	157.55	-1.5e-06	-7.1e-10	170.76	25

Contents

1. **SDP relaxation**
2. Facial Reduction Approach (FRA)
3. DNN relaxation

SemiDefinite Programming (SDP)

SemiDefinite Programming (SDP)

$$\theta_P^* := \min_{X \in \mathbb{S}^n} \{A_0 \bullet X \mid A_j \bullet X = b_j \ (j = 1, \ldots, m), X \in \mathbb{S}^n_+\},$$

where

- $\mathbb{S}^n$: $n \times n$ symmetric cone, $\mathbb{S}^n_+$: $n \times n$ positive semidefinite cone and $V \bullet W = \sum_{i,j=1}^n V_{ij}W_{ij} = \mathrm{Tr}(VW)$
- $A_0, A_j \in \mathbb{S}^n$, $b_j \in \mathbb{R}$

SemiDefinite Programming (SDP)

SemiDefinite Programming (SDP)

$$\theta_P^* := \min_{X \in \mathbb{S}^n} \{A_0 \bullet X \mid A_j \bullet X = b_j \ (j = 1, \ldots, m), X \in \mathbb{S}^n_+\},$$

where

- $\mathbb{S}^n$: $n \times n$ symmetric cone, $\mathbb{S}^n_+$: $n \times n$ positive semidefinite cone and $V \bullet W = \sum_{i,j=1}^n V_{ij} W_{ij} = \mathrm{Tr}(VW)$
- $A_0, A_j \in \mathbb{S}^n$, $b_j \in \mathbb{R}$

- SDP has many applications: System and Control, Combinatorial Optimization, Statistics, Finance, Machine Learning, Computational Complexity etc
- Primal-Dual Interior-Point Method (PD-IPM) can find an approximated solution efficiently

SemiDefinite Programming (SDP)Slater condition – a constraint qualification

SemiDefinite Programming (SDP)

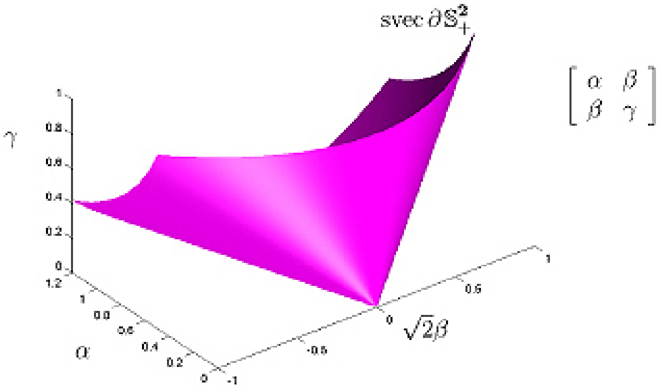
$$\theta_P^* := \min_{X \in \mathbb{S}^n} \{A_0 \bullet X \mid A_j \bullet X = b_j \ (j = 1, \ldots, m), X \in \mathbb{S}^n_+\},$$

where

- $\mathbb{S}^n$: $n \times n$ symmetric cone, $\mathbb{S}^n_+$: $n \times n$ positive semidefinite cone and $V \bullet W = \sum_{i,j=1}^n V_{ij} W_{ij} = \mathrm{Tr}(VW)$
- $A_0, A_j \in \mathbb{S}^n$, $b_j \in \mathbb{R}$

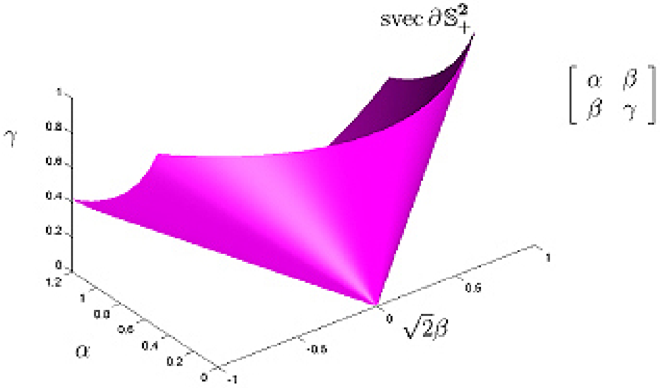
- SDP is a nonlinear optimization – Difficulties is packaged in $\mathbb{S}^n_+$
- Theoretical results for SDP required Slater condition

Figure of $\mathbb{S}^2_+$ from www.convexoptimization.com/wikimization



- Observation – $\alpha, \gamma \geq \mathbf{0}$ & $\alpha\gamma - \beta^2 \geq \mathbf{0}$

SDP is nonlinear problem... example



$$\min \left\{ \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \bullet \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} \middle| \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \bullet \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} = 2, \begin{pmatrix} \alpha & \beta \\ \beta & \gamma \end{pmatrix} \in \mathbb{S}^2_+ \right\}$$

- $\beta = 1$

Slater condition – a constrained qualification

Slater condition

$\exists \hat{\mathbf{X}}$ such that $\mathbf{A_j} \bullet \hat{\mathbf{X}} = \mathbf{b_j}$ $(\mathbf{j} = 1, \ldots, \mathbf{m})$ and $\hat{\mathbf{X}}$ positive definite

- SDP has interior feasible solutions $\Rightarrow$ SDP has optimal solutions

Slater condition – a constrained qualification

Slater condition

$\exists \hat{X}$ such that $A_j \bullet \hat{X} = b_j \ (j = 1, \ldots, m)$ and $\hat{X}$ positive definite

- SDP has interior feasible solutions $\Rightarrow$ SDP has optimal solutions
- PD-IPM converges to an optimal solution with $O(\sqrt{n}\log(1/\epsilon))$
- PD-IPM converges superlinealy under a mild assumption

Slater condition – a constrained qualification

Slater condition

$\exists \hat{X}$ such that $A_j \bullet \hat{X} = b_j$ $(j = 1, \dots, m)$ and $\hat{X}$ positive definite

- SDP has interior feasible solutions $\Rightarrow$ SDP has optimal solutions
- PD-IPM converges to an optimal solution with $O(\sqrt{n} \log(1/\epsilon))$
- PD-IPM converges superlinealy under a mild assumption
- Failed in Slater condition $\Rightarrow$ Return an inaccurate solution and/or convergence slowly
- SDP relaxation for Graph Partition Problems

Contents

Facial Reduction Algorithm

- Proposed by Borwein & Wolkowicz in 1980
- Generate an equivalent SDP problem that contains an interior feasible solution
- Obtain an accurate solution with $20 \sim 40$ iterations
- $\Rightarrow$ Use a pre-conditioner for solving SDP problems

Applications of FRA

SDP relaxation problems for Graph Partition Problem [Wolkowicz-Zhao 1999], Quadratic Assignment Problem [Zhao-Karisch-Rendl-Wolkowicz 1998] and Sensor Network Localization [Krislock-Wolkowicz 2009]

Facial Reduction Algorithm

SDP that does not have any interior feasible solutions

$$\begin{cases} \min & A_0 \bullet X \\ \text{subject to} & A_j \bullet X = b_j \ (j = 1, \ldots, m), \\ & X \in \left\{ Q \begin{pmatrix} \tilde{X} & O \\ O & O \end{pmatrix} Q^T \,\middle|\, \tilde{X} \in \mathbb{S}^k_+ \right\} \subset \mathbb{S}^n_+ \end{cases}$$

- Q is an orthogonal matrix and k is the max. rank of feasible solutions.
- FRA finds Q and k in finite iterations
- Need to solve the following SDP:

$$\text{Find } y \in \mathbb{R}^m \text{ s.t. } b^T y = 0, -\sum_{j=1}^{m} y_j A_j \in \mathbb{S}^n_+$$

Facial Reduction Algorithm

SDP that does not have any interior feasible solutions

$$\begin{cases} \min & A_0 \bullet X \\ \text{subject to} & A_j \bullet X = b_j \ (j = 1, \ldots, m), \\ & X \in \left\{ Q \begin{pmatrix} \tilde{X} & O \\ O & O \end{pmatrix} Q^T \,\middle|\, \tilde{X} \in \mathbb{S}^k_+ \right\} \subset \mathbb{S}^n_+ \end{cases}$$

- $W := -\sum_{j=1}^m y_j A_j$

$$\text{Find } y \in \mathbb{R}^m \text{ s.t. } b^T y = 0, -\sum_{j=1}^m y_j A_j \in \mathbb{S}^n_+$$

- $W \bullet X = 0$ for all feasible solutions X
- Q is obtained from W

$$W = Q \Lambda Q^T, \text{ and } \Lambda = \begin{pmatrix} D & O \\ O & O \end{pmatrix}, \text{ where } D \in \mathbb{S}^{n-k}$$

Facial Reduction Algorithm

SDP that does not have any interior feasible solutions

$$\begin{cases} \min & A_0 \bullet X \\ \text{subject to} & A_j \bullet X = b_j \ (j = 1, \ldots, m), \\ & X \in \left\{ Q \begin{pmatrix} \tilde{X} & O \\ O & O \end{pmatrix} Q^T \,\middle|\, \tilde{X} \in \mathbb{S}^k_+ \right\} \subset \mathbb{S}^n_+ \end{cases}$$

- Construct a smaller SDP

$$\begin{cases} \min & \tilde{A}_0 \bullet \begin{pmatrix} \tilde{X} & O \\ O & O \end{pmatrix} \\ \text{subject to} & \tilde{A}_j \bullet \begin{pmatrix} \tilde{X} & O \\ O & O \end{pmatrix} = b_j \ (j = 1, \ldots, m), \\ & \tilde{X} \in \mathbb{S}^k_+ \end{cases}$$

Facial Reduction Algorithm – Difficult!

SDP that does not have any interior feasible solutions

$$\begin{cases} \min & \mathbf{A_0} \bullet \mathbf{X} \\ \text{subject to} & \mathbf{A_j} \bullet \mathbf{X} = \mathbf{b_j}\ (\mathbf{j} = 1, \ldots, \mathbf{m}), \\ & \mathbf{X} \in \left\{ \mathbf{Q} \begin{pmatrix} \tilde{\mathbf{X}} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix} \mathbf{Q}^T \,\middle|\, \tilde{\mathbf{X}} \in \mathbb{S}^k_+ \right\} \subset \mathbb{S}^n_+ \end{cases}$$

- Need to solve the following SDP:

$$\text{Find } \mathbf{y} \in \mathbb{R}^m \text{ s.t. } \mathbf{b}^T\mathbf{y} = 0, -\sum_{j=1}^{m} \mathbf{y_j}\mathbf{A_j} \in \mathbb{S}^n_+$$

- $\mathbf{y} = \mathbf{0}$ is unique solution $\Leftrightarrow$ $\exists$ interior feasible sol.
- Compute exact solutions $\mathbf{y}$ by floating-point computation
- E.g., $\mathbf{y} = (1.0\text{-e}05, -2.0\text{-e}06, 3.0\text{-e}04)$ is the zero vector?

Facial Reduction Algorithm – Difficult!

SDP that does not have any interior feasible solutions

$$\begin{cases} \min & \mathbf{A_0} \bullet \mathbf{X} \\ \text{subject to} & \mathbf{A_j} \bullet \mathbf{X} = \mathbf{b_j} \ (\mathbf{j} = 1, \ldots, \mathbf{m}), \\ & \mathbf{X} \in \left\{ \mathbf{Q} \begin{pmatrix} \tilde{\mathbf{X}} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{pmatrix} \mathbf{Q}^T \,\middle|\, \tilde{\mathbf{X}} \in \mathbb{S}^{\mathbf{k}}_+ \right\} \subset \mathbb{S}^{\mathbf{n}}_+ \end{cases}$$

- Existing applications of FRA – finding $\mathbf{Q}$ and/or $\mathbf{k}$ by using a structure/property
- Density may become larger; $\mathbf{Q}$ may be fully dense
- Computation time may increase $\Rightarrow$ Need to keep the density in smaller SDP

FRA to construct sparse SDP [Waki-Muramatsu, 2009]

- $W = RR^T \Rightarrow R^T X R = O_{n-k}$

FRA to construct sparse SDP [Waki-Muramatsu, 2009]

- $\mathbf{W} = \mathbf{R}\mathbf{R}^T \Rightarrow \mathbf{R}^T\mathbf{X}\mathbf{R} = \mathbf{O}_{n-k}$
- Choose a suitable matrix $\mathbf{L}$ s.t. $\mathbf{V} = (\mathbf{L}, \mathbf{R})$: non-singular

$$
\left\{
\begin{array}{ll}
\inf & (\mathbf{V}^{-1}\mathbf{C}\mathbf{V}^{-T}) \bullet (\mathbf{V}^T\mathbf{X}\mathbf{V}) \\
\text{subject to} & (\mathbf{V}^{-1}\mathbf{A}_i\mathbf{V}^{-T}) \bullet (\mathbf{V}^T\mathbf{X}\mathbf{V}) = b_i \ (i = 1, \ldots, m), \\
 & \mathbf{V}^T\mathbf{X}\mathbf{V} \in \mathbb{S}^n_+, \mathbf{R}^T\mathbf{X}\mathbf{R} = \mathbf{O}_{n-k}
\end{array}
\right.
$$

FRA to construct sparse SDP [Waki-Muramatsu, 2009]

- $\mathbf{W} = \mathbf{R}\mathbf{R}^T \Rightarrow \mathbf{R}^T\mathbf{X}\mathbf{R} = \mathbf{O}_{n-k}$
- Choose a suitable matrix $\mathbf{L}$ s.t. $\mathbf{V} = (\mathbf{L}, \mathbf{R})$: non-singular

$$\left\{\begin{array}{cl} \inf & (\mathbf{V}^{-1}\mathbf{C}\mathbf{V}^{-T}) \bullet (\mathbf{V}^T\mathbf{X}\mathbf{V}) \\ \text{subject to} & (\mathbf{V}^{-1}\mathbf{A}_i\mathbf{V}^{-T}) \bullet (\mathbf{V}^T\mathbf{X}\mathbf{V}) = b_i \ (i = 1, \dots, m), \\ & \mathbf{V}^T\mathbf{X}\mathbf{V} \in \mathbb{S}^n_+, \mathbf{R}^T\mathbf{X}\mathbf{R} = \mathbf{O}_{n-k} \end{array}\right.$$

- Obtain small SDP

$$\mathbf{V}^T\mathbf{X}\mathbf{V} = \begin{pmatrix} \mathbf{L}^T\mathbf{X}\mathbf{L} & \mathbf{L}^T\mathbf{X}\mathbf{R} \\ \mathbf{R}^T\mathbf{X}\mathbf{L} & \mathbf{R}^T\mathbf{X}\mathbf{R} \end{pmatrix} \in \mathbb{S}^n_+$$

$$\Rightarrow \mathbf{V}^T\mathbf{X}\mathbf{V} = \begin{pmatrix} \mathbf{L}^T\mathbf{X}\mathbf{L} & \mathbf{O} \\ \mathbf{O} & \mathbf{O}_{n-k} \end{pmatrix}, \mathbf{L}^T\mathbf{X}\mathbf{L} \in \mathbb{S}^k_+$$

FRA to construct sparse SDP [Waki-Muramatsu, 2009]

- $\mathbf{W} = \mathbf{R}\mathbf{R}^T \Rightarrow \mathbf{R}^T\mathbf{X}\mathbf{R} = \mathbf{O}_{n-k}$
- Choose a suitable matrix $\mathbf{L}$ s.t. $\mathbf{V} = (\mathbf{L}, \mathbf{R})$: non-singular

$$\left\{ \begin{array}{cl} \inf & (\mathbf{V}^{-1}\mathbf{C}\mathbf{V}^{-T}) \bullet (\mathbf{V}^T\mathbf{X}\mathbf{V}) \\ \text{subject to} & (\mathbf{V}^{-1}\mathbf{A}_i\mathbf{V}^{-T}) \bullet (\mathbf{V}^T\mathbf{X}\mathbf{V}) = b_i \ (i = 1, \dots, m), \\ & \mathbf{V}^T\mathbf{X}\mathbf{V} \in \mathbb{S}^n_+, \mathbf{R}^T\mathbf{X}\mathbf{R} = \mathbf{O}_{n-k} \end{array} \right.$$

- Obtain small SDP

$$\begin{aligned} \mathbf{V}^T\mathbf{X}\mathbf{V} &= \begin{pmatrix} \mathbf{L}^T\mathbf{X}\mathbf{L} & \mathbf{L}^T\mathbf{X}\mathbf{R} \\ \mathbf{R}^T\mathbf{X}\mathbf{L} & \mathbf{R}^T\mathbf{X}\mathbf{R} \end{pmatrix} \in \mathbb{S}^n_+ \\ &\Rightarrow \mathbf{V}^T\mathbf{X}\mathbf{V} = \begin{pmatrix} \mathbf{L}^T\mathbf{X}\mathbf{L} & \mathbf{O} \\ \mathbf{O} & \mathbf{O}_{n-k} \end{pmatrix}, \mathbf{L}^T\mathbf{X}\mathbf{L} \in \mathbb{S}^k_+ \end{aligned}$$

- $\mathbf{L}$ consists of some suitable unit vectors

FRA for GPP

$$(\text{QOP}): \quad \inf_{x \in \mathbb{R}^n} \{x^T C x \mid x_1 + \cdots + x_n = 0, x_i^2 = 1 \ (i = 1, \ldots, n)\}$$

$$(\text{SDP}): \quad \inf_{X \in \mathbb{S}^n_+} \{C \bullet X \mid E \bullet X = 0, E_i \bullet X = 1 \ (i = 1, \ldots, n)\}$$

- Easy to apply FRA

FRA for GPP

$$(\text{QOP}): \quad \inf_{x \in \mathbb{R}^n} \{x^T C x \mid x_1 + \cdots + x_n = 0, x_i^2 = 1 \ (i = 1, \ldots, n)\}$$

$$(\text{SDP}): \quad \inf_{X \in \mathbb{S}^n_+} \{C \bullet X \mid E \bullet X = 0, E_i \bullet X = 1 \ (i = 1, \ldots, n)\}$$

- Easy to apply FRA
- Relationship

(QOP)		
↓ SDP relax.		
(SDP)	$\xrightarrow[\text{equiv.}]{\text{FRA}}$	(SDP')

FRA for GPP

$$(\text{QOP}): \quad \inf_{x\in\mathbb{R}^n}\{x^TCx \mid x_1+\cdots+x_n=0, x_i^2=1\ (i=1,\ldots,n)\}$$

$$(\text{SDP}): \quad \inf_{X\in\mathbb{S}^n_+}\{C\bullet X \mid E\bullet X=0, E_i\bullet X=1\ (i=1,\ldots,n)\}$$

- Easy to apply FRA
- Relationship

(QOP)		
$\downarrow$ SDP relax.		
(SDP)	$\xrightarrow[\text{equiv.}]{\text{FRA}}$	(SDP')

- (SDP')$\leftrightarrow$(QOP').

FRA for GPP

$$(\mathrm{QOP}): \quad \inf_{x \in \mathbb{R}^n} \{x^T C x \mid x_1 + \cdots + x_n = 0, x_i^2 = 1 \ (i = 1, \ldots, n)\}$$

$$(\mathrm{SDP}): \quad \inf_{X \in \mathbb{S}^n_+} \{C \bullet X \mid E \bullet X = 0, E_i \bullet X = 1 \ (i = 1, \ldots, n)\}$$

- Easy to apply FRA
- Relationship

(QOP)	$\xrightarrow{\text{equiv.}}$	(QOP')
$\downarrow$ SDP relax.		SDP relax. $\downarrow$
(SDP)	$\xrightarrow[\text{equiv.}]{\text{FRA}}$	(SDP')

- (SDP')$\leftrightarrow$(QOP').

FRA for GPP

$$(\text{QOP}): \quad \inf_{x\in\mathbb{R}^n} \{x^T C x \mid x_1 + \cdots + x_n = 0, x_i^2 = 1 \ (i = 1, \ldots, n)\}$$

$$(\text{SDP}): \quad \inf_{X\in\mathbb{S}^n_+} \{C \bullet X \mid E \bullet X = 0, E_i \bullet X = 1 \ (i = 1, \ldots, n)\}$$

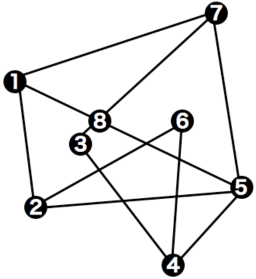
- Easy to apply FRA
- Relationship

(QOP)	$\xrightarrow{\text{equiv.}}$	(QOP')
$\downarrow$ SDP relax.		SDP relax. $\downarrow$
(SDP)	$\xrightarrow[\text{equiv.}]{\text{FRA}}$	(SDP')

- (SDP')$\leftrightarrow$(QOP').
- (QOP') = QOP by removing x_1 from (QOP)

Numerical result for maximum stable set

- For G(V, E), $\mathbf{W} \subset \mathbf{V}$: stable set if and only if no two vertices in **W** are adjacent
- $\mathbf{W} = \{1, 3, 5, 6\}$, $\mathbf{W} = \{2, 4, 7\}$ etc
- Find the maximum cardinality $\alpha(\mathbf{G})$ of possible stable sets in G(V, E)



- Motzkin-Strus formula & apply Lasserre's SDP hierarchy

Numerical result for maximum stable set

- Motzkin-Strus 1965

$$\frac{1}{\alpha(G)} = \min_{x \in \mathbb{R}^n} \left\{ x^T (A_G + I_n) x : \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \right\}$$

- SDP relaxation problems: Slater condition fails for primal & dual $\Rightarrow$ Apply FRA to both

Table: SDPT3 4.0 with ϵ=1.0e-8.

	SDP			FRA		
n	iter	cpusec	d.gap	d.gap	cpusec	iter
30	33	124.2	1.4e-4	4.8e-9	98.1	36
30	35	163.3	2.5e-4	1.3e-9	101.2	37
40	31	606.8	5.5e-5	6.2e-10	722.0	39
40	29	501.1	2.6e-3	3.0e-9	726.6	40

Numerical result for maximum stable set

- Motzkin-Strus 1965

$$\frac{1}{\alpha(G)} = \min_{x \in \mathbb{R}^n} \left\{ x^T (A_G + I_n) x : \sum_{i=1}^{n} x_i = 1, x_i \geq 0 \right\}$$

- SDP relaxation problems: Slater condition fails for primal & dual $\Rightarrow$ Apply FRA to both

Table: SDPT3 4.0 with ϵ=1.0e-8.

	SDP			FRA		
n	iter	cpusec	d.gap	d.gap	cpusec	iter
30	33	124.2	1.4e-4	4.8e-9	98.1	36
30	35	163.3	2.5e-4	1.3e-9	101.2	37
40	31	606.8	5.5e-5	6.2e-10	722.0	39
40	29	501.1	2.6e-3	3.0e-9	726.6	40

Contents

Another example – DNN relaxation

$$
(\mathrm{MINQP})\theta^* := \begin{cases} \inf_{x \in \mathbb{R}^n} & x^T Q x + 2c^T x \\ \text{s.t.} & Ax = b, x \geq 0, \\ & x_j \in \{0, 1\} (\forall j \in \mathcal{I}) \end{cases}
$$

- Burer (2009) reformulated it to Completely Positive Problem (CPP)
- (CPP) can be exploited for many combinatorial problems, e.g., Stability number, Stochastic mixed-binary linear optimization problems with objective uncertainty, Graph partition, Quadratic assignment problem etc.

Another example – DNN relaxation

$$(\text{MINQP})\theta^* := \begin{cases} \inf_{x\in\mathbb{R}^n} & x^T Q x + 2c^T x \\ \text{s.t.} & Ax = b, x \geq 0, \\ & x_j \in \{0,1\} (\forall j \in \mathcal{I}) \end{cases}$$

- Burer (2009) reformulated it to Completely Positive Problem (CPP)
- (CPP) can be exploited for many combinatorial problems, e.g., Stability number, Stochastic mixed-binary linear optimization problems with objective uncertainty, Graph partition, Quadratic assignment problem etc.

CPP

$$\theta^* = \begin{cases} \inf_{X\in\mathbb{S}^{n+1}} & P \bullet X \\ \text{s.t.} & P_j \bullet X = d_j, \ X \in (\mathcal{C}_{n+1})^* \end{cases}$$

where $(\mathcal{C}_{n+1})^* = \{X \in \mathbb{S}^n : X = KK^T, K \in \mathbb{R}_+^{n\times k}\}$

CPP is also difficult

CPP

$$\theta^* = \begin{cases} \inf_{X \in \mathbb{S}^{n+1}} & P \bullet X \\ \text{s.t.} & P_j \bullet X = d_j, \ X \in (\mathcal{C}_{n+1})^* \end{cases}$$

where $(\mathcal{C}_{n+1})^* = \{X \in \mathbb{S}^n : X = KK^T, K \in \mathbb{R}_+^{n \times k}\}$

- Difficulties in MINQP is packaged at $(\mathcal{C}_{n+1})^*$
- CPP is intractable!
- $(\mathcal{C}_{n+1})^* \subseteq \mathbb{S}_+^{n+1} \cap \mathbb{R}_+^{(n+1)\times(n+1)} \Rightarrow$ DNN relaxation
- $\exists$Many approximations for $(\mathcal{C}_{n+1})^*$

DNN relaxation is also difficult

$$(\mathrm{DNN})\tilde{\theta}^* := \begin{cases} \inf_{\mathbf{X} \in \mathbb{S}^n} & \mathbf{P} \bullet \mathbf{X} \\ \text{s.t.} & \mathbf{P_j} \bullet \mathbf{X} = \mathbf{d_j} \\ & \mathbf{X} \in \mathbb{S}_+^{\mathbf{n+1}}, \mathbf{X} \in \mathbb{R}_+^{\mathbf{(n+1)\times(n+1)}} \end{cases}$$

- $\tilde{\theta}^* \leq \theta^*$
- No $\mathbf{X} \in \mathbb{R}_+^{\mathbf{(n+1)\times(n+1)}}$ constraints = SDP relaxation
- No $\mathbf{X} \in \mathbb{S}_+^{\mathbf{n+1}}$ constraints = LP relaxation
- DNN is stronger than LP & SDP
- DNN can be reformulated to SDP ⇒ SDP software is available!

Difficulties in DNN relaxation

Difficulties in DNN relaxation

- Size of SDP is too large to solve
- Slater condition fails!

Size

- Size of SDP is too large to solve

$$\mathbf{X} \in \mathbb{S}^{\mathbf{N}}_{+} \cap \mathbb{R}^{\mathbf{N}\times\mathbf{N}}_{+} \Leftrightarrow \mathbf{X} = \mathbf{Y}, \mathbf{Z} = \begin{pmatrix} \mathbf{X} & \mathbf{O} \\ \mathbf{O} & \mathrm{Diag}(\mathbf{Y}) \end{pmatrix} \in \mathbb{S}^{\mathbf{N}+\mathbf{N}^2}_{+}$$

Slater condition fails

- For some $\mathbf{j} = \mathbf{1}, \ldots, \mathbf{m}$, $(\mathbf{P}_0 + \mathbf{P}_j + \mathbf{P}_{m+j}) \bullet \mathbf{X} = \mathbf{0}$ & $\mathbf{P}_0 + \mathbf{P}_j + \mathbf{P}_{m+j} \in \mathbb{S}^{\mathbf{n}+\mathbf{1}}_{+}$

Applying FRA to DNN relaxation

Good News

- Can find $\mathbf{y}$ & $\mathbf{W}$ from $\mathbf{A}, \mathbf{b}$ in the original MIQP

$$\text{Find } \mathbf{y} \in \mathbb{R}^m \text{ s.t. } \mathbf{d}^T\mathbf{y} = 0, \mathbf{W} = -\sum_{j=1} y_j P_j \in \mathbb{S}^n_+$$

- Also easy to construct $\mathbf{R}$ s.t. $\mathbf{W} = \mathbf{R}\mathbf{R}^T$
- Size : $\mathbb{S}^{n+1}_+ \Rightarrow \mathbb{S}^{n-m+1}_+$ & $\mathbb{R}^{(n+1)\times(n+1)}_+ \Rightarrow \mathbb{R}^{(n-m+1)\times(n-m+1)}_+$

Bad News

- It's not complete FRA
- Need to solve the homogeneous system

Numerical result – Randomly generated problems

$$(\text{MINQP})\theta^* := \begin{cases} \inf_{x\in\mathbb{R}^n} & x^TQx + 2c^Tx \\ \text{s.t.} & a_i^Tx = b\ (i = 1, \ldots, m), x \geq 0, \\ & x_j \in \{0, 1\} (\forall j \in B) \end{cases}$$

Table: SDPA 7.3.1 with ϵ =1.0e-7

$(m, n, \lvert B \rvert)$	SDP iter.	SDP cpusec	SDP d.gap	FRA d.gap	FRA cpusec	FRA iter.
(50, 10, 0)	25	2.5	1.8e-5	1.5e-7	29.8	29
(50, 20, 0)	29	3.4	1.5e-3	1.4e-5	36.4	30
(50, 30, 0)	30	3.9	1.5e-3	1.3e-6	17.8	27
(50, 10, 30)	29	3.4	3.3e-3	2.7e-6	43.8	36
(50, 20, 30)	34	4.5	9.8e-4	2.5e-6	53.0	39
(50, 30, 30)	50	7.5	3.1e-4	9.5e-7	28.7	41

Numerical result – Randomly generated problems

$$(\mathrm{MINQP})\theta^* := \begin{cases} \inf_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{x}^T\mathbf{Q}\mathbf{x} + 2\mathbf{c}^T\mathbf{x} \\ \text{s.t.} & \mathbf{a}_i^T\mathbf{x} = \mathbf{b}\ (\mathbf{i} = 1, \dots, \mathbf{m}), \mathbf{x} \geq \mathbf{0}, \\ & \mathbf{x}_j \in \{0, 1\}(\forall \mathbf{j} \in \mathbf{B}) \end{cases}$$

Table: SDPA 7.3.1 with ϵ =1.0e-7

	SDP			FRA		
$(\mathbf{m}, \mathbf{n}, \lvert\mathbf{B}\rvert)$	iter.	cpusec	d.gap	d.gap	cpusec	iter.
(50, 10, 0)	25	2.5	1.8e-5	1.5e-7	29.8	29
(50, 20, 0)	29	3.4	1.5e-3	1.4e-5	36.4	30
(50, 30, 0)	30	3.9	1.5e-3	1.3e-6	17.8	27
(50, 10, 30)	29	3.4	3.3e-3	2.7e-6	43.8	36
(50, 20, 30)	34	4.5	9.8e-4	2.5e-6	53.0	39
(50, 30, 30)	50	7.5	3.1e-4	9.5e-7	28.7	41

Numerical result – Portfolio selection problems

$$\theta^* := \begin{cases} \inf_{\mathbf{x}\in\mathbb{R}^n} & \mathbf{x}^T\Sigma\mathbf{x} - \lambda(\mu^T\mathbf{x})^2 \\ \text{s.t.} & \mathbf{e}^T\mathbf{x} = 1, \mathbf{a}_i^T\mathbf{x} \le \mathbf{b}_i, \mathbf{x} \ge \mathbf{0} \end{cases}$$

Table: SDPA 7.3.1 with ϵ =1.0e-7

	SDP		FRA		FRA+α	
(n, m)	cpusec	d.gap	cpusec	d.gap	cpusec	d.gap
(100, 0)	78.1	8.1e-7	112.9	3.9e-7	78.2	9.8e-7
(100, 20)	334.3	5.4e-6	6798.3	7.5e-7	101.9	8.1e-7
(100, 40)	774.4	3.6e-6	–	–	176.0	9.8e-7
(150, 0)	704.5	887.4	–	–	325.3	9.3e-7
(150, 30)	3436.1	1.7e-5	–	–	558.0	6.6e-7
(150, 60)	–	–	–	–	960.1	5.9e-7

Summary

- SDPs in which Slater condition fails $\Rightarrow$ Hopless to get accurate results by PD-IPM
- Facial Reduction Algorithm resolves the difficulty
- But, may spend much computation time to solve the resulting problem, *e.g.*, DNN relax.

Summary

- SDPs in which Slater condition fails $\Rightarrow$ Hopless to get accurate results by PD-IPM
- Facial Reduction Algorithm resolves the difficulty
- But, may spend much computation time to solve the resulting problem, *e.g.*, DNN relax.
- **Thank you very much for your attention**