Conic programming bounds for structured combinatorial problems

Juan Vera♯**, Cristian Dobre**♮

Tilburg University, The Netherlands♯
RWTH Aachen University, Germany♮

Cones at a glance

Examples of cones

$\mathbb{R}^n_+$

$$\{x \in \mathbb{R}^n : x \geq 0\}$$

Cones at a glance

Examples of cones

$\mathbb{R}^n_+$ $\{x \in \mathbb{R}^n : x \geq 0\}$

$\mathcal{S}_n^+$ $\{X \in \mathbb{R}^{n \times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n\}$

Cones at a glance

Examples of cones

$\mathbb{R}^n_+$ $$\{x \in \mathbb{R}^n : x \geq 0\}$$

$\mathcal{S}_n^+$ $$\{X \in \mathbb{R}^{n\times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n\}$$

$\mathcal{C}$ $$\{X \in \mathbb{R}^{n\times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n_+\}.$$

Cones at a glance

Examples of cones

$\mathbb{R}^n_+$ $\{x \in \mathbb{R}^n : x \geq 0\}$

$\mathcal{S}_n^+$ $\{X \in \mathbb{R}^{n \times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n\}$

$\mathcal{C}$ $\{X \in \mathbb{R}^{n \times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n_+\}.$

Primal cone ($\mathcal{K}$) - dual cone ($\mathcal{K}^*$)

$$\mathcal{K}^* := \{s \in \mathbb{R}^n : \langle s, x \rangle := s^T x \geq 0 \text{ for all } x \in \mathcal{K}\}$$

$$\mathcal{K}^* := \{S \in \mathbb{R}^{n \times n} : \langle S, X \rangle := \mathrm{trace}(SX) \geq 0 \text{ for all } X \in \mathcal{K}\}$$

Cones at a glance

Examples of cones

$\mathbb{R}^n_+$ $\quad \{x \in \mathbb{R}^n : x \geq 0\}$

$\mathcal{S}_n^+$ $\quad \{X \in \mathbb{R}^{n\times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n\}$

$\mathcal{C}$ $\quad \{X \in \mathbb{R}^{n\times n} : u^T X u \geq 0 \text{ for all } u \in \mathbb{R}^n_+\}.$

Primal cone ($\mathcal{K}$) - dual cone ($\mathcal{K}^*$)

$$\mathcal{K}^* := \{s \in \mathbb{R}^n : \langle s, x\rangle := s^T x \geq 0 \text{ for all } x \in \mathcal{K}\}$$

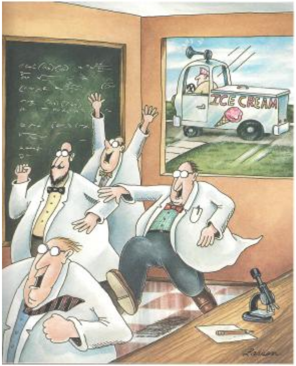
$$\mathcal{K}^* := \{S \in \mathbb{R}^{n\times n} : \langle S, X\rangle := \mathrm{trace}(SX) \geq 0 \text{ for all } X \in \mathcal{K}\}$$

"Famous" dual cones

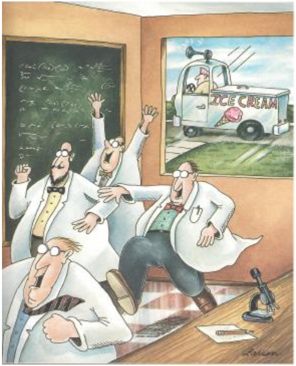
$$(\mathbb{R}^n_+)^* = \mathbb{R}^n_+ \quad (\mathcal{S}_n^+)^* = \mathcal{S}_n^+$$

$$\mathcal{C}^* := \mathrm{cone}\{xx^T : x \geq 0\}$$

Linearization over suitable cones



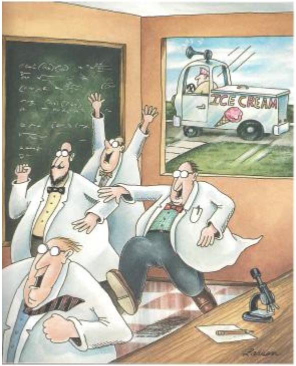
Linearization over suitable cones



Linearization -Technique: $X := xx^T$

$$x^T Qx = \mathrm{trace}(x^T Qx) = \mathrm{trace}(Qxx^T) = \mathrm{trace}(QX);$$

Linearization over suitable cones



Linearization -Technique: $X := xx^T$

$$x^T Qx = \mathrm{trace}(x^T Qx) = \mathrm{trace}(Qxx^T) = \mathrm{trace}(QX);$$

$$X = xx^T \quad \Leftrightarrow \quad \mathrm{rank}(X) = 1 \text{ and } X \succeq 0.$$

Standard Quadratic Optimization Problem

$$\begin{aligned} \min \quad & x^T A x + 2c^T x \\ \text{s.t.} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

Standard Quadratic Optimization Problem

$$\begin{aligned} \min \quad & x^T A x + 2c^T x \\ \text{s.t.} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

- $x^T A x + 2c^T x = x^T (A + ce^T + ec^T) x = x^T Q x = \mathrm{trace}(Q x x^T) = \mathrm{trace}(QX)$

Standard Quadratic Optimization Problem

$$\begin{aligned} \min \quad & x^T A x + 2c^T x \\ \text{s.t.} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

- $x^T A x + 2c^T x = x^T (A + ce^T + ec^T) x = x^T Q x = \mathrm{trace}(Q x x^T) = \mathrm{trace}(QX)$
- $e^T x = (e^T x)^2 = \mathrm{trace}(e^T x x^T e) = \mathrm{trace}(e e^T x x^T) = \mathrm{trace}(JX)$

Standard Quadratic Optimization Problem

$$\begin{aligned} \min \quad & x^T A x + 2c^T x \\ \text{s.t.} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

- $x^T A x + 2c^T x = x^T(A + ce^T + ec^T)x = x^T Q x = \mathrm{trace}(Qxx^T) = \mathrm{trace}(QX)$
- $e^T x = (e^T x)^2 = \mathrm{trace}(e^T x x^T e) = \mathrm{trace}(ee^T xx^T) = \mathrm{trace}(JX)$

... because xx^T with $x \geq 0$ are the extreme rays of $\mathcal{C}^*$...

$$\begin{aligned} \min \quad & \langle Q, X \rangle \\ \text{s.t.} \quad & \langle J, X \rangle = 1 \\ & X \in \mathcal{C}^*. \end{aligned}$$

Standard Quadratic Optimization Problem

$$\begin{aligned} \min \quad & x^T A x + 2c^T x \\ \text{s.t.} \quad & e^T x = 1 \\ & x \geq 0. \end{aligned}$$

- $x^T A x + 2c^T x = x^T(A + ce^T + ec^T)x = x^T Q x = \mathrm{trace}(Qxx^T) = \mathrm{trace}(QX)$
- $e^T x = (e^T x)^2 = \mathrm{trace}(e^T x x^T e) = \mathrm{trace}(ee^T xx^T) = \mathrm{trace}(JX)$

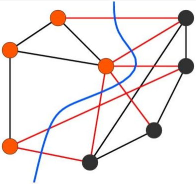
... because xx^T with $x \geq 0$ are the extreme rays of $\mathcal{C}^*$...

$$\begin{aligned} \min \quad & \langle Q, X \rangle \\ \text{s.t.} \quad & \langle J, X \rangle = 1 \\ & X \in \mathcal{C}^*. \end{aligned}$$

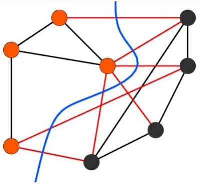
First paper to describe relaxations of NP-hard combinatorial optimization problems

A.J. Quist, E. de Klerk, C. Roos and T. Terlaky. Copositive relaxation for general quadratic programming. *Optimization Methods and Software*, **9**:185–208, 1998.

Max-cut problem



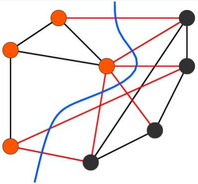
Max-cut problem



$V = [S_1 S_2]$

$$\mathrm{MC} := \max \quad \sum_{i,j=1}^{n} W_{ij} \left(\frac{1 - x_i x_j}{4} \right)$$
$$\text{s.t.} \quad x_i \in \{-1, 1\}, \quad i \in V.$$

Max-cut problem



$V = [S_1 S_2]$

$$\mathrm{MC} := \max \quad \sum_{i,j=1}^{n} W_{ij} \left(\frac{1 - x_i x_j}{4} \right)$$
$$\text{s.t.} \quad x_i \in \{-1, 1\}, \quad i \in V.$$

$x \in \{-1,1\}^n$ and $X := xx^T$

$$\begin{aligned} \mathrm{MC} = \max \quad & \frac{1}{4}\mathrm{trace}(W(J - X)) \\ \text{s.t.} \quad & \mathrm{diag}(\mathrm{X}) = \mathrm{e}, \\ & X \succeq 0, \\ & \mathrm{rank}(X) = 1. \end{aligned}$$

Conic programming

- For given symmetric $n \times n$ matrices $A_0, \ldots, A_m$ and $b \in \mathbb{R}^m$, the standard conic programing problem is defined as:

$$\begin{aligned} \min \quad & \langle A_0, X \rangle \\ \text{s.t.} \quad & \langle A_i, X \rangle = b_i, \quad i = 1, ..., m \\ & X \in \mathcal{K}. \end{aligned}$$

Conic programming

- For given symmetric $n \times n$ matrices $A_0, \ldots, A_m$ and $b \in \mathbb{R}^m$, the standard conic programing problem is defined as:

$$\begin{array}{ll} \min & \langle A_0, X\rangle \\ \text{s.t.} & \langle A_i, X\rangle = b_i, \quad i = 1, ..., m \\ & X \in \mathcal{K}. \end{array}$$

- define the dual as:

$$\begin{array}{ll} \max & \langle b, y\rangle \\ \text{s.t.} & A_0 - \sum_{i=1}^{m} y_i A_i = S \\ & y \in \mathbb{R}^m,\ S \in \mathcal{K}^*. \end{array}$$

Conic programming

- For given symmetric $n \times n$ matrices $A_0, \ldots, A_m$ and $b \in \mathbb{R}^m$, the standard conic programing problem is defined as:

$$\begin{array}{ll} \min & \langle A_0, X \rangle \\ \text{s.t.} & \langle A_i, X \rangle = b_i, \quad i = 1, ..., m \\ & X \in \mathcal{K}. \end{array}$$

- define the dual as:

$$\begin{array}{ll} \max & \langle b, y \rangle \\ \text{s.t.} & A_0 - \sum_{i=1}^{m} y_i A_i = S \\ & y \in \mathbb{R}^m, \ S \in \mathcal{K}^*. \end{array}$$

- if $\mathcal{K} = \mathcal{S}_n^+$ we talk about semidefinite programming
- if $\mathcal{K} = \mathcal{C}$ we talk about copositive programming

Copositivity and combinatorial problems

Reformulation of the standard quadratic optimization problem

I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, **18**: 301–320, 2000.

Copositivity and combinatorial problems

Reformulation of the standard quadratic optimization problem

I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, **18**: 301–320, 2000.

Reformulation of the stability number of a graph

E. de Klerk and D.V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, **12**:875–892, 2002.

Copositivity and combinatorial problems

Reformulation of the standard quadratic optimization problem

I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, **18**: 301–320, 2000.

Reformulation of the stability number of a graph

E. de Klerk and D.V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, **12**:875–892, 2002.

Reformulation of the quadratic assignment problem

J. Povh, and F. Rendl. Copositive and semidefinite relaxations of the Quadratic Assignment Problem. *Discrete Optimization* **6**:231–241, 2009.

Copositivity and combinatorial problems

Reformulation of the standard quadratic optimization problem

I.M. Bomze, M. Dür, E. de Klerk, C. Roos, A.J. Quist, and T. Terlaky. On copositive programming and standard quadratic optimization problems. *Journal of Global Optimization*, **18**: 301–320, 2000.

Reformulation of the stability number of a graph

E. de Klerk and D.V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, **12**:875–892, 2002.

Reformulation of the quadratic assignment problem

J. Povh, and F. Rendl. Copositive and semidefinite relaxations of the Quadratic Assignment Problem. *Discrete Optimization* **6**:231–241, 2009.

Reformulation of the binary and continuous nonconvex quadratic programs

S. Burer. On the copositive representation of binary and continuous nonconvex quadratic programs. *Mathematical Programming A*, **120(2)**:479–495, 2009.

Optimizing over $\mathcal{C}$

- all the combinatorial problems mentioned before are "hard"
- reformulating them as optimization problems over $\mathcal{C}$ moves the difficulty in the structure of $\mathcal{C}$

Optimizing over $\mathcal{C}$

- all the combinatorial problems mentioned before are "hard"
- reformulating them as optimization problems over $\mathcal{C}$ moves the difficulty in the structure of $\mathcal{C}$

Example: stability number

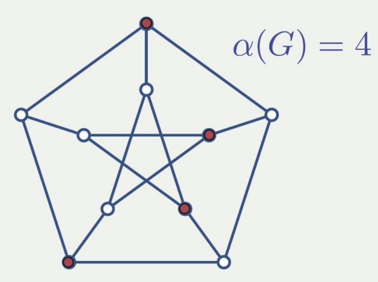
$$\alpha(G) = \min \{\lambda \ : \ \lambda(A + I) - J \in \mathcal{C}\}.$$

Optimizing over $\mathcal{C}$

- all the combinatorial problems mentioned before are "hard"
- reformulating them as optimization problems over $\mathcal{C}$ moves the difficulty in the structure of $\mathcal{C}$

Example: stability number

$$\alpha(G) = \min \{\lambda \ : \ \lambda(A + I) - J \in \mathcal{C}\}.$$



Approximation hierarchies

- approximate $\mathcal{C}$ via a hierarchy of semidefinite programs (SDP). Why ?

Approximation hierarchies

- approximate $\mathcal{C}$ via a hierarchy of semidefinite programs (SDP). Why ?
- SDP may be solved in polynomial time with fixed precision in the real number model. (Ellipsoid algorithm of Nemirovski-Yudin)
- Interior point methods (IPM s) for LP were extended to SDP in the early 1990 s (Nesterov-Nemirovski, Alizadeh, ...)

Approximation hierarchies

- approximate $\mathcal{C}$ via a hierarchy of semidefinite programs (SDP). Why ?
- SDP may be solved in polynomial time with fixed precision in the real number model. (Ellipsoid algorithm of Nemirovski-Yudin)
- Interior point methods (IPM s) for LP were extended to SDP in the early 1990 s (Nesterov-Nemirovski, Alizadeh, ...)

Drawback

Such approaches come at the expense of increasing the size of the data in the resulting SDP problems, rendering them numerically insolvable.

Approximation hierarchies

- approximate $\mathcal{C}$ via a hierarchy of semidefinite programs (SDP). Why ?
- SDP may be solved in polynomial time with fixed precision in the real number model. (Ellipsoid algorithm of Nemirovski-Yudin)
- Interior point methods (IPM s) for LP were extended to SDP in the early 1990 s (Nesterov-Nemirovski, Alizadeh, ...)

Drawback

Such approaches come at the expense of increasing the size of the data in the resulting SDP problems, rendering them numerically insolvable.

Good news :)

Exploit the symmetry coming from the structure of the problem.

Approximation hierarchies (ctd...)

Parrilo

P. Parrilo. *Structured Semidefinite programming and Algebraic Geometry Methods in Robustness and Optimization. PhD thesis, California Institute of Technology, Pasadena, CA, USA, 2000.*

de Klerk - Pasechnik

E. de Klerk and D.V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, **12**:875–892, 2002.

Approximation hierarchies (ctd...)

Parrilo

P. Parrilo. *Structured Semidefinite programming and Algebraic Geometry Methods in Robustness and Optimization. PhD thesis, California Institute of Technology, Pasadena, CA, USA, 2000.*

de Klerk - Pasechnik

E. de Klerk and D.V. Pasechnik. Approximation of the stability number of a graph via copositive programming. *SIAM Journal on Optimization*, **12**:875–892, 2002.

Pena - Vera - Zuluaga

J. Pena, J. Vera, and L.F. Zuluaga. Computing the stability number of a graph via linear and semidefinite programming. *SIAM Journal on Optimization*, **18(1)**:87–105, 2007.

Pena - Vera - Zuluaga hierarchy

Given $\beta \in \mathbb{N}^n$, $|\beta| := \beta_1 + ... + \beta_n$, $x^\beta := x^{\beta_1} \cdots x^{\beta_n}$

$$\mathcal{E}^r := \left\{ \sum_{\beta \in \mathbb{N}^n, |\beta| = r} x^\beta x^T (P_\beta + N_\beta) x \ : \ P_\beta \in \mathcal{S}_n^+, N_\beta \in \mathcal{N} \right\}$$

Pena - Vera - Zuluaga hierarchy

Given $\beta \in \mathbb{N}^n$, $|\beta| := \beta_1 + ... + \beta_n$, $x^\beta := x^{\beta_1} \cdots x^{\beta_n}$

$$\mathcal{E}^r := \left\{ \sum_{\beta \in \mathbb{N}^n, |\beta| = r} x^\beta x^T (P_\beta + N_\beta) x \ : \ P_\beta \in \mathcal{S}_n^+, N_\beta \in \mathcal{N} \right\}$$

The following sequence of cones approximate $\mathcal{C}$

$$\mathcal{K}^r := \left\{ M \in \mathcal{S} \ : \ x^T M x \left(\sum_{i=1}^n x_i \right)^r \in \mathcal{E}^r \right\}.$$

Pena - Vera - Zuluaga hierarchy

Given $\beta \in \mathbb{N}^n$, $|\beta| := \beta_1 + ... + \beta_n$, $x^\beta := x^{\beta_1} \cdots x^{\beta_n}$

$$\mathcal{E}^r := \left\{ \sum_{\beta \in \mathbb{N}^n, |\beta| = r} x^\beta x^T (P_\beta + N_\beta) x \;:\; P_\beta \in \mathcal{S}_n^+, N_\beta \in \mathcal{N} \right\}$$

The following sequence of cones approximate $\mathcal{C}$

$$\mathcal{K}^r := \left\{ M \in \mathcal{S} \;:\; x^T M x \left(\sum_{i=1}^n x_i \right)^r \in \mathcal{E}^r \right\}.$$

$\mathcal{K}^0$ - dual of doubly nonnegative matrices / $\mathcal{K}^1$

$$\mathcal{K}^0 := \left\{ M \in \mathcal{S} \;:\; x^T M x = x^T (P_\beta + N_\beta) x \right\},$$

$$\mathcal{K}^1 := \left\{ M \in \mathcal{S} \;:\; x^T M x (\sum_{i=1}^n x_i) = \sum_{i=1}^n x_i x^T (P_i + N_i) x \right\}.$$

Example revisited

Recall..

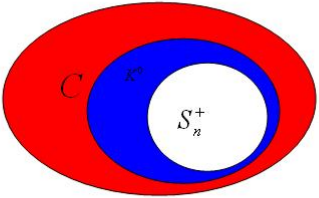
$$\alpha(G) = \min \{\lambda \ : \ \lambda(A + I) - J \in \mathcal{C}\}.$$

Example revisited

Recall..

$$\alpha(G) = \min\{\lambda \,:\, \lambda(A + I) - J \in \mathcal{C}\}.$$

- $\mathcal{S}_n^+ + \mathcal{N} = \mathcal{K}^0 \subset \mathcal{K}^1 \subset \ldots \subset \mathcal{C}$

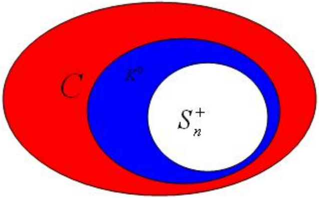


Example revisited

Recall..

$$\alpha(G) = \min \{\lambda \ : \ \lambda(A+I) - J \in \mathcal{C}\}.$$

- $\mathcal{S}_n^+ + \mathcal{N} = \mathcal{K}^0 \subset \mathcal{K}^1 \subset ... \subset \mathcal{C}$



Relaxation

$$\alpha(G) \leq \min \{\lambda \ : \ \lambda(A+I) - J \in \mathcal{K}^r\}.$$

$\lambda(A + I) - J \in \mathcal{K}^1$

- denote $\lambda(A + I) - J := M$

Recall...

$$\mathcal{K}^1 := \left\{ M \in \mathcal{S} \ : \ x^T M x (\sum_{i=1}^{n} x_i) = \sum_{i=1}^{n} x_i x^T (P_i + N_i) x \right\}.$$

$\lambda(A+I)-J \in \mathcal{K}^1$

- denote $\lambda(A+I)-J := M$

Recall...

$$\mathcal{K}^1 := \left\{ M \in \mathcal{S} \ : \ x^T M x (\sum_{i=1}^{n} x_i) = \sum_{i=1}^{n} x_i x^T (P_i + N_i) x \right\}.$$

Equivalent to:

$$\mathcal{K}^1 := \left\{ M \in \mathcal{S} \ : \ x^T M x (\sum_{i=1}^{n} x_i) \geq \sum_{i=1}^{n} x_i x^T P_i x \right\}.$$

$\lambda(A+I)-J\in\mathcal{K}^1$

- denote $\lambda(A+I)-J := M$

Recall...

$$\mathcal{K}^1 := \left\{ M\in\mathcal{S} \;:\; x^TMx(\sum_{i=1}^n x_i) = \sum_{i=1}^n x_i x^T(P_i+N_i)x \right\}.$$

Equivalent to:

$$\mathcal{K}^1 := \left\{ M\in\mathcal{S} \;:\; x^TMx(\sum_{i=1}^n x_i) \geq \sum_{i=1}^n x_i x^T P_i x \right\}.$$

Thus:

$$\alpha(G) \leq \min\left\{ \lambda \;:\; x^TMx(\sum_{i=1}^n x_i) \geq \sum_{i=1}^n x_i x^T P_i x,\; P_i\succeq 0 \right\}.$$

ALGEBRAIC DETOUR !!!

Matrix algebras

Definition

A set $\mathcal{A} \subseteq \mathbb{C}^{n\times n}$ (resp. $\mathbb{R}^{n\times n}$) is called a matrix $*$-algebra over $\mathbb{C}$ (resp. $\mathbb{R}$) if, for all $X, Y \in \mathcal{A}$:

- $\alpha X + \beta Y \in \mathcal{A}\ \forall \alpha, \beta \in \mathbb{C}$ (resp. $\mathbb{R}$);
- $X^* \in \mathcal{A}$;
- $XY \in \mathcal{A}$.

Matrix algebras

Definition

A set $\mathcal{A} \subseteq \mathbb{C}^{n \times n}$ (resp. $\mathbb{R}^{n \times n}$) is called a matrix $*$-algebra over $\mathbb{C}$ (resp. $\mathbb{R}$) if, for all $X, Y \in \mathcal{A}$:

- $\alpha X + \beta Y \in \mathcal{A}\ \forall \alpha, \beta \in \mathbb{C}$ (resp. $\mathbb{R}$);
- $X^* \in \mathcal{A}$;
- $XY \in \mathcal{A}$.

Denote the basis of $\mathcal{A}$ by $\{B_1, B_2, ..., B_d\}$.

Example

- The circulant matrices form a commutative matrix $*$-algebra.

Example

- The circulant matrices form a commutative matrix $*$-algebra.

Form of a circulant matrix R

$$R = \begin{pmatrix} r_0 & r_1 & r_2 & . & . & r_{n-1} \\ r_{n-1} & r_0 & r_1 & r_2 & . & . \\ . & . & r_0 & r_1 & r_2 & . \\ . & . & . & . & . & . \\ r_2 & . & . & . & r_0 & r_1 \\ r_1 & r_2 & . & . & . & r_0 \end{pmatrix}$$

Example

- The circulant matrices form a commutative matrix $*$-algebra.

Form of a circulant matrix R

$$R = \begin{pmatrix} r_0 & r_1 & r_2 & . & . & r_{n-1} \\ r_{n-1} & r_0 & r_1 & r_2 & . & . \\ . & . & r_0 & r_1 & r_2 & . \\ . & . & . & . & . & . \\ r_2 & . & . & . & r_0 & r_1 \\ r_1 & r_2 & . & . & . & r_0 \end{pmatrix}$$

- Each row is a cyclic shift of the row above it, i.e:

$$R_{ij} := r_{(j-i) \ mod \ n}.$$

Example

- The circulant matrices form a commutative matrix $*$-algebra.

Form of a circulant matrix R

$$R = \begin{pmatrix} r_0 & r_1 & r_2 & . & . & r_{n-1} \\ r_{n-1} & r_0 & r_1 & r_2 & . & . \\ . & . & r_0 & r_1 & r_2 & . \\ . & . & . & . & . & . \\ r_2 & . & . & . & r_0 & r_1 \\ r_1 & r_2 & . & . & . & r_0 \end{pmatrix}$$

- Each row is a cyclic shift of the row above it, i.e:

$$R_{ij} := r_{(j-i)\ mod\ n}.$$

Further reading:

R.M. Gray. Toeplitz and Circulant Matrices: A review. *Foundation and Trends in Comunications and Information Theory*, **2**(3):155-239, 2006. Available online.

Example (ctd.)

- A basis for the symmetric 4×4 circulant matrices is

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; B_2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix},$$

$$B_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Algebraic Symmetry for SDP Problems

Assumption

$M := \lambda(A + I) - J$ belongs to some matrix $\mathbb{C}*$-algebra $\mathcal{A}$ of dimension d, having the basis: $\{B_1, B_2, ..., B_d\}$.

Algebraic Symmetry for SDP Problems

Assumption

$M := \lambda(A + I) - J$ belongs to some matrix $\mathbb{C}*$-algebra $\mathcal{A}$ of dimension d, having the basis: $\{B_1, B_2, ..., B_d\}$.

Theorem

If the SDP problem has a optimal primal-dual solution then there exists an optimal primal-dual solution (X^*, y^*, S^*) such that $X^* \in \mathcal{A}$, $y \in \mathbb{R}^m$ and $S^* \in \mathcal{A}$.

Algebraic Symmetry for SDP Problems

Assumption

$M := \lambda(A + I) - J$ belongs to some matrix $\mathbb{C}*$-algebra $\mathcal{A}$ of dimension d, having the basis: $\{B_1, B_2, ..., B_d\}$.

Theorem

If the SDP problem has a optimal primal-dual solution then there exists an optimal primal-dual solution (X^*, y^*, S^*) such that $X^* \in \mathcal{A}$, $y \in \mathbb{R}^m$ and $S^* \in \mathcal{A}$.

- We may restrict optimization to feasible points in $\mathcal{A}$.
- $M = \sum_{i=1}^{d} m_i B_i$, for some $m_i \in \mathbb{C}$ (resp. $\mathbb{R}$).

Algebraic Symmetry for SDP Problems

Assumption

$M := \lambda(A+I) - J$ belongs to some matrix $\mathbb{C}*$-algebra $\mathcal{A}$ of dimension d, having the basis: $\{B_1, B_2, ..., B_d\}$.

Theorem

If the SDP problem has a optimal primal-dual solution then there exists an optimal primal-dual solution (X^*, y^*, S^*) such that $X^* \in \mathcal{A}$, $y \in \mathbb{R}^m$ and $S^* \in \mathcal{A}$.

- We may restrict optimization to feasible points in $\mathcal{A}$.
- $M = \sum_{i=1}^{d} m_i B_i$, for some $m_i \in \mathbb{C}$ (resp. $\mathbb{R}$).
- One obtain significant reductions when $\mathcal{A}$ is low dimensional.
- For the 4×4 circulant matrices we have: $n = 4$ and $d = 3$.

Canonical block diagonalization of a matrix *-algebra $\mathcal{A}$

Theorem (Wedderburn)

Assume $\mathcal{A}$ is a matrix *-algebra over $\mathbb{C}$ that contains I. Then there is a unitary Q ($Q^*Q = I$) and some integer s such that

$$Q^*\mathcal{A}Q = \begin{pmatrix} \mathcal{A}_1 & 0 & \dots & 0 \\ 0 & \mathcal{A}_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \mathcal{A}_s \end{pmatrix},$$

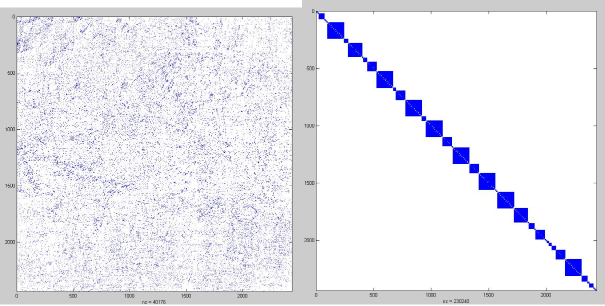
where each $\mathcal{A}_i$ takes the form

$$\mathcal{A}_i = \left\{ \begin{pmatrix} A & 0 & \dots & 0 \\ 0 & A & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & A \end{pmatrix} \mid A \in \mathbb{C}^{n_i \times n_i} \right\},$$

for some integers n_i, $i = 1, ..., s$.

Smaller data matrices

- smaller data matrices yield numerical tractability of the original problem



The algebraic detour is over!

K. Murota, Y. Kanno, M. Kojima and S. Kojima: A numerical algorithm for block-diagonal decomposition of matrix *-algebras with application to semidefinite programming. *Japan Journal of Industrial and Applied Mathematics*, **27(1)**:125–160, 2010.

E. de Klerk, C. Dobre, D.V. Pasechnik: Numerical block diagonalization of matrix $*$-algebras with application to semidefinite programming. *Mathematical Programming B*, **129**:91–111, 2011.

Symmetry reduction

Recall...

$$\alpha(G) \leq \min \left\{ \lambda \ : \ x^T M x (\sum_{i=1}^{n} x_i) \geq \sum_{i=1}^{n} x_i x^T P_i x, \ P_i \succeq 0 \right\}.$$

Symmetry reduction

Recall...

$$\alpha(G) \leq \min\left\{\lambda \ : \ x^T M x(\sum_{i=1}^{n} x_i) \geq \sum_{i=1}^{n} x_i x^T P_i x, \ P_i \succeq 0\right\}.$$

- restrict optimization to $\mathcal{A} := \mathrm{cent}\ \mathrm{aut}(A)$
- $M = \sum_{i=1}^{d} m_i B_i$ and $P_j = \sum_{i=1}^{d} p_i^j B_i$.

Symmetry reduction

Recall...

$$\alpha(G) \leq \min\left\{\lambda \ : \ x^T M x(\sum_{i=1}^n x_i) \geq \sum_{i=1}^n x_i x^T P_i x, \ P_i \succeq 0\right\}.$$

- restrict optimization to $\mathcal{A} := \mathrm{cent\ aut}(A)$
- $M = \sum_{i=1}^d m_i B_i$ and $P_j = \sum_{i=1}^d p_i^j B_i$.

If aut(A) is transitive then the problem is equivalent to:

$$\alpha(G) \leq \min\left\{\lambda \ : \ x^T M x(\sum_{i=1}^n x_i) \geq x_1 x^T P x, \ P \succeq 0\right\},$$

where $P \in \mathrm{cent\ stab}_{\mathrm{aut}(A)}(1)$.

- $M = \sum_{i=1}^d m_i B_i$ and $P = \sum_{i=1}^{d_1} p_i D_i$.

Symmetry reduction (ctd...)

$$\alpha(G) \leq \min \left\{ \lambda \ : \ x^T M x (\sum_{i=1}^{n} x_i) \geq x_1 x^T P x, \ \sum_{i=1}^{d_1} p_i D_i \succeq 0 \right\}.$$

Symmetry reduction (ctd...)

$$\alpha(G) \leq \min\left\{\lambda \ : \ x^T M x(\sum_{i=1}^{n} x_i) \geq x_1 x^T P x, \ \sum_{i=1}^{d_1} p_i D_i \succeq 0\right\}.$$

- construct the linear operator Lone mapping degree two polynomials invariant under aut(A) into degree three polynomials invariant under aut(A).

Symmetry reduction (ctd...)

$$\alpha(G) \leq \min\left\{\lambda \ : \ x^T M x(\sum_{i=1}^{n} x_i) \geq x_1 x^T P x, \ \sum_{i=1}^{d_1} p_i D_i \succeq 0\right\}.$$

- construct the linear operator Lone mapping degree two polynomials invariant under aut(A) into degree three polynomials invariant under aut(A).
- construct the linear operator Lmany mapping degree two polynomials invariant under $\mathrm{stab}_{\mathrm{aut}(A)}(1)$ into degree three polynomials invariant under aut(A).

Symmetry reduction (ctd...)

$$\alpha(G) \leq \min\left\{\lambda \ : \ x^T M x(\sum_{i=1}^{n} x_i) \geq x_1 x^T P x, \ \sum_{i=1}^{d_1} p_i D_i \succeq 0\right\}.$$

- construct the linear operator Lone mapping degree two polynomials invariant under aut(A) into degree three polynomials invariant under aut(A).
- construct the linear operator Lmany mapping degree two polynomials invariant under $\mathrm{stab}_{\mathrm{aut}(A)}(1)$ into degree three polynomials invariant under aut(A).

$M = \sum_{i=1}^{d} m_i B_i$ and $P = \sum_{i=1}^{d_1} p_i D_i$

$$\alpha(G) \leq \min\left\{\lambda \ : \ \mathrm{Lone} * m \geq \mathrm{Lmany} * p, \ \sum_{i=1}^{d_1} p_i \oplus_{k=1}^{s} D_i^k \succeq 0\right\}.$$

Toy-example: 4-cycle (square)

Adjacency matrix - A

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Toy-example: 4-cycle (square)

Adjacency matrix - A

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Basis cent aut(A)

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \; B_2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, \; B_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Toy-example: 4-cycle (square)

Adjacency matrix - A

$$A = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}.$$

Basis cent aut(A)

$$B_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, B_2 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{pmatrix}, B_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Basis $\mathrm{cent\ stab}_{\mathrm{aut}(A)}(1)$

$$D_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, D_2 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, D_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, D_4 = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$D_5 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, D_6 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}, D_7 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \ldots$ - 10

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \ldots$ - 10
- degree three polynomials: $x_1^3,\ x_1^2x_2,\ x_1^2x_3,\ x_1^2x_4,\ x_1x_2x_3, \ldots$ - 20

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \dots$ - 10
- degree three polynomials: $x_1^3,\ x_1^2x_2,\ x_1^2x_3,\ x_1^2x_4,\ x_1x_2x_3, \dots$ - 20
- degree two polynomials invariant under aut(A):
  $x_1^2 + x_2^2 + x_3^2 + x_4^2,\ x_1x_2 + x_1x_4 + x_2x_3 + x_3x_4,\ x_1x_3 + x_2x_4.$

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \ldots$ - 10
- degree three polynomials: $x_1^3,\ x_1^2x_2,\ x_1^2x_3,\ x_1^2x_4,\ x_1x_2x_3, \ldots$ - 20
- degree two polynomials invariant under aut(A):
  $x_1^2 + x_2^2 + x_3^2 + x_4^2,\ x_1x_2 + x_1x_4 + x_2x_3 + x_3x_4,\ x_1x_3 + x_2x_4.$
- degree three polynomials invariant under aut(A):
  $x_1^3 + x_2^3 + x_3^3 + x_4^3,\ (x_1^2 + x_3^2)(x_2 + x_4) + (x_1 + x_3)(x_2^2 + x_4^2),\ x_1x_3(x_1 + x_3) + x_2x_4(x_2 + x_4),\ x_1x_3(x_2 + x_4) + x_2x_4(x_1 + x_3).$

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \dots$ - 10
- degree three polynomials: $x_1^3,\ x_1^2x_2,\ x_1^2x_3,\ x_1^2x_4,\ x_1x_2x_3, \dots$ - 20
- degree two polynomials invariant under aut(A): $x_1^2+x_2^2+x_3^2+x_4^2,\ x_1x_2+x_1x_4+x_2x_3+x_3x_4,\ x_1x_3+x_2x_4.$
- degree three polynomials invariant under aut(A): $x_1^3+x_2^3+x_3^3+x_4^3,\ (x_1^2+x_3^2)(x_2+x_4)+(x_1+x_3)(x_2^2+x_4^2),\ x_1x_3(x_1+x_3)+x_2x_4(x_2+x_4),\ x_1x_3(x_2+x_4)+x_2x_4(x_1+x_3).$
- degree two polynomials invariant under $\mathrm{stab}_{\mathrm{aut}(A)}(1)$: $x_1^2,\ x_2^2+x_4^2,\ x_3^2,\ x_1x_2+x_1x_4,\ x_1x_3,\ x_2x_4,\ x_2x_3+x_3x_4.$

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \ldots$ - 10
- degree three polynomials: $x_1^3,\ x_1^2x_2,\ x_1^2x_3,\ x_1^2x_4,\ x_1x_2x_3, \ldots$ - 20
- degree two polynomials invariant under aut(A): $x_1^2+x_2^2+x_3^2+x_4^2,\ x_1x_2+x_1x_4+x_2x_3+x_3x_4,\ x_1x_3+x_2x_4.$
- degree three polynomials invariant under aut(A): $x_1^3+x_2^3+x_3^3+x_4^3,\ (x_1^2+x_3^2)(x_2+x_4)+(x_1+x_3)(x_2^2+x_4^2),\ x_1x_3(x_1+x_3)+x_2x_4(x_2+x_4),\ x_1x_3(x_2+x_4)+x_2x_4(x_1+x_3).$
- degree two polynomials invariant under $\mathrm{stab}_{\mathrm{aut}(A)}(1)$: $x_1^2,\ x_2^2+x_4^2,\ x_3^2,\ x_1x_2+x_1x_4,\ x_1x_3,\ x_2x_4,\ x_2x_3+x_3x_4.$

$$Lone = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{pmatrix},\ Lmany = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}.$$

Square - n=4 variables

- degree two polynomials: $x_1^2,\ x_1x_2,\ x_1x_3,\ x_1x_4,\ x_2^2, \dots$ - 10
- degree three polynomials: $x_1^3,\ x_1^2x_2,\ x_1^2x_3,\ x_1^2x_4,\ x_1x_2x_3, \dots$ - 20
- degree two polynomials invariant under aut(A): $x_1^2+x_2^2+x_3^2+x_4^2,\ x_1x_2+x_1x_4+x_2x_3+x_3x_4,\ x_1x_3+x_2x_4.$
- degree three polynomials invariant under aut(A): $x_1^3+x_2^3+x_3^3+x_4^3,\ (x_1^2+x_3^2)(x_2+x_4)+(x_1+x_3)(x_2^2+x_4^2),\ x_1x_3(x_1+x_3)+x_2x_4(x_2+x_4),\ x_1x_3(x_2+x_4)+x_2x_4(x_1+x_3).$
- degree two polynomials invariant under $\mathrm{stab}_{\mathrm{aut}(A)}(1)$: $x_1^2,\ x_2^2+x_4^2,\ x_3^2,\ x_1x_2+x_1x_4,\ x_1x_3,\ x_2x_4,\ x_2x_3+x_3x_4.$

$$Lone = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 2 & 1 \end{pmatrix},\ Lmany = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{pmatrix}.$$

$n = 4,\ d = 3;\ d_1 = 7$

$$\alpha(C_4) \le \min\left\{\lambda\ :\ \mathrm{Lone} * m \ge \mathrm{Lmany} * p,\ \sum_{i=1}^{d_1} p_i D_i \succeq 0\right\}.$$

Example - The Crossing Number

- example follows — for crossing number computation problem in $\mathcal{K}_{7,n}$

Example - The Crossing Number

- example follows — for crossing number computation problem in $\mathcal{K}_{7,n}$

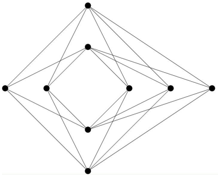


Figure : Drawing of $\mathcal{K}_{4,5}$ with 8 crossings

Definition

The crossing number $cr(G)$ of a graph G is the minimum number of pairwise intersection of edges in a drawing of G in the plane.

The Crossing Number of Complete Bipartite Graphs

- Turán posed the problem of finding $cr(K_{m,n})$ while being in a forced labor camp in WWII.

The Crossing Number of Complete Bipartite Graphs

- Turán posed the problem of finding $cr(K_{m,n})$ while being in a forced labor camp in WWII.
- Zarankiewicz (1954) proved: $cr(K_{m,n}) = Z(m,n)$,
  where $Z(m,n) := \lfloor \frac{m-1}{2} \rfloor \lfloor \frac{m}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor \lfloor \frac{n}{2} \rfloor$.

The Crossing Number of Complete Bipartite Graphs

- Turán posed the problem of finding $cr(K_{m,n})$ while being in a forced labor camp in WWII.
- Zarankiewicz (1954) proved: $cr(K_{m,n}) = Z(m,n)$, where $Z(m,n) := \lfloor \frac{m-1}{2} \rfloor \lfloor \frac{m}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor \lfloor \frac{n}{2} \rfloor$.
- A flaw in Zarankiewicz argument for $\geq$ was found by Guy in 1969, hence $cr(K_{m,n}) \leq Z(m,n)$.

The Crossing Number of Complete Bipartite Graphs

- Turán posed the problem of finding $cr(K_{m,n})$ while being in a forced labor camp in WWII.
- Zarankiewicz (1954) proved: $cr(K_{m,n}) = Z(m,n)$, where $Z(m,n) := \lfloor \frac{m-1}{2} \rfloor \lfloor \frac{m}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor \lfloor \frac{n}{2} \rfloor$.
- A flaw in Zarankiewicz argument for $\geq$ was found by Guy in 1969, hence $cr(K_{m,n}) \leq Z(m,n)$.

Zarankiewicz conjecture - if $m, n \geq 7$

$$cr(K_{m,n}) \leq Z(m,n)$$

The Crossing Number of Complete Bipartite Graphs

- Turán posed the problem of finding $cr(K_{m,n})$ while being in a forced labor camp in WWII.
- Zarankiewicz (1954) proved: $cr(K_{m,n}) = Z(m,n)$, where $Z(m,n) := \lfloor \frac{m-1}{2} \rfloor \lfloor \frac{m}{2} \rfloor \lfloor \frac{n-1}{2} \rfloor \lfloor \frac{n}{2} \rfloor$.
- A flaw in Zarankiewicz argument for $\geq$ was found by Guy in 1969, hence $cr(K_{m,n}) \leq Z(m,n)$.

Zarankiewicz conjecture - if $m, n \geq 7$

$$cr(K_{m,n}) \leq Z(m,n)$$

de Klerk et al. - 2008

$$cr(K_{m,n}) \geq \frac{n}{2} \left(n \max\{t \ : \ Q_{(m-1)!} - tJ_{(m-1)!} \in \mathcal{C}\} - \lfloor \frac{m-1}{2} \rfloor \lfloor \frac{m}{2} \rfloor \right).$$

$cr(K_{7,n})$

- Upper bound (conjecture): $cr(K_{7,n}) \leq 2.25n^2$.

$cr(K_{7,n})$

- Upper bound (conjecture): $cr(K_{7,n}) \leq 2.25n^2$.

de Klerk et al. - 2008

$$cr(K_{7,n}) \geq \frac{n}{2}\left(n \max\{t \ : \ Q_{6!} - tJ_{6!} \in \mathcal{K}^0\} - 9\right).$$

$cr(K_{7,n})$

- Upper bound (conjecture): $cr(K_{7,n}) \leq 2.25n^2$.

de Klerk et al. - 2008

$$cr(K_{7,n}) \geq \frac{n}{2}\left(n \max\{t \ : \ Q_{6!} - tJ_{6!} \in \mathcal{K}^0\} - 9\right).$$

- Lower bound optimizing over $\mathcal{K}^0$: $cr(K_{7,n}) \geq 2.1796n^2$.

$cr(K_{7,n})$

- Upper bound (conjecture): $cr(K_{7,n}) \leq 2.25n^2$.

de Klerk et al. - 2008

$$cr(K_{7,n}) \geq \frac{n}{2}\left(n \max\{t \ : \ Q_{6!} - tJ_{6!} \in \mathcal{K}^0\} - 9\right).$$

- Lower bound optimizing over $\mathcal{K}^0$: $cr(K_{7,n}) \geq 2.1796n^2$.

Vera, Dobre - 2013

$$cr(K_{7,n}) \geq \frac{n}{2}\left(n \max\{t \ : \ Q_{6!} - tJ_{6!} \in \mathcal{K}^1\} - 9\right).$$

$cr(K_{7,n})$

- Upper bound (conjecture): $cr(K_{7,n}) \leq 2.25n^2$.

de Klerk et al. - 2008

$$cr(K_{7,n}) \geq \frac{n}{2}\left(n \max\{t \;:\; Q_{6!} - tJ_{6!} \in \mathcal{K}^0\} - 9\right).$$

- Lower bound optimizing over $\mathcal{K}^0$: $cr(K_{7,n}) \geq 2.1796n^2$.

Vera, Dobre - 2013

$$cr(K_{7,n}) \geq \frac{n}{2}\left(n \max\{t \;:\; Q_{6!} - tJ_{6!} \in \mathcal{K}^1\} - 9\right).$$

- Lower bound optimizing over $\mathcal{K}^1$: $cr(K_{7,n}) \geq 2.2030n^2$.

Computing the Symmetry Reduction over $\mathcal{K}^1$

$n = 720,\ d = 78;\ d_1 = 19305$

$$cr(K_{7,n}) \geq \max\left\{ t \ :\ \text{Lone} * m \geq \text{Lmany} * p,\ \sum_{i=1}^{d_1} p_i D_i \succeq 0 \right\}.$$

Computing the Symmetry Reduction over $\mathcal{K}^1$

$n = 720,\ d = 78;\ d_1 = 19305$

$$cr(K_{7,n}) \geq \max\left\{ t \ :\ \mathrm{Lone} * m \geq \mathrm{Lmany} * p,\ \sum_{i=1}^{d_1} p_i D_i \succeq 0 \right\}.$$

- The number of linear equations reduces, many of these equations are repeated. Goes from 62.5 million linear equations down to 19305

Computing the Symmetry Reduction over $\mathcal{K}^1$

$n = 720,\ d = 78;\ d_1 = 19305$

$$cr(K_{7,n}) \geq \max\left\{ t \ :\ \text{Lone} * m \geq \text{Lmany} * p,\ \sum_{i=1}^{d_1} p_i D_i \succeq 0 \right\}.$$

- The number of linear equations reduces, many of these equations are repeated. Goes from 62.5 million linear equations down to 19305
- A more "standard" approach to reduce problem further from one 720×720 SDP constraint, to five 30×30, 78×78, 102×102, 102×102 and 102×102 SDP constrains. (de Klerk, Dobre, Pasechnik - 2011).

Computing the Symmetry Reduction over $\mathcal{K}^1$

$n = 720$, $d = 78$; $d_1 = 19305$

$$cr(K_{7,n}) \geq \max\left\{ t \ : \ \text{Lone} * m \geq \text{Lmany} * p, \ \sum_{i=1}^{d_1} p_i D_i \succeq 0 \right\}.$$

- The number of linear equations reduces, many of these equations are repeated. Goes from 62.5 million linear equations down to 19305
- A more "standard" approach to reduce problem further from one 720×720 SDP constraint, to five 30×30, 78×78, 102×102, 102×102 and 102×102 SDP constrains. (de Klerk, Dobre, Pasechnik - 2011).
- The routines are implemented in Matlab.
- The problem is solved in less than four hours using SDPT3 in Coral lab at Lehigh University - 32 GB of RAM and 16 AMD Opteron 2.0 GHz Processor.

Summary, conclusions and future work

- we have seen which "hard" combinatorial problems can be reformulated via copositive programs

Summary, conclusions and future work

- we have seen which "hard" combinatorial problems can be reformulated via copositive programs
- we have seen how one can approximate copositive programming via semidefinite programming hierarchies

Summary, conclusions and future work

- we have seen which "hard" combinatorial problems can be reformulated via copositive programs
- we have seen how one can approximate copositive programming via semidefinite programming hierarchies
- we showed how to perform symmetry reduction in the SDP hierarchies

Summary, conclusions and future work

- we have seen which "hard" combinatorial problems can be reformulated via copositive programs
- we have seen how one can approximate copositive programming via semidefinite programming hierarchies
- we showed how to perform symmetry reduction in the SDP hierarchies
- we exemplified the advantages of this approach by providing new lower bounds on crossing number instances

Summary, conclusions and future work

- we have seen which "hard" combinatorial problems can be reformulated via copositive programs
- we have seen how one can approximate copositive programming via semidefinite programming hierarchies
- we showed how to perform symmetry reduction in the SDP hierarchies
- we exemplified the advantages of this approach by providing new lower bounds on crossing number instances

- Adapt the techniques for other combinatorial problems.

Thank you for your attention !