Conic Optimization

The Basics, some Fundamental Results, and Recent Developments

Miguel F. Anjos

Professor and Canada Research Chair

MINO-COST Spring School, April 2014

Semidefinite Optimization

Semidefinite optimization, or semidefinite programming (SDP), refers to the problem of optimizing a linear function over the intersection of the set of symmetric positive semidefinite matrices with an affine space.

The simplest example of semidefinite optimization is the familiar linear programming (LP) problem:

$$\begin{array}{lll|lll} \max & \langle \mathbf{c}, \mathbf{x} \rangle & & \min & \langle \mathbf{b}, \mathbf{y} \rangle & \\ \text{s.t.} & \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i, \; i = 1, \ldots, m & & \text{s.t.} & \mathbf{z} = \sum_{i=1}^{m} y_i \mathbf{a}_i - \mathbf{c} & \\ & \mathbf{x} \geq 0 & & & \mathbf{z} \geq 0 & \end{array}$$

where $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T \mathbf{b}$.

Semidefinite Optimization (ctd)

The general SDP problem has the form:

$$\begin{array}{lll|lll} \max & \langle \mathbf{C}, \mathbf{X} \rangle & & \min & \langle \mathbf{b}, \mathbf{y} \rangle & \\ \text{s.t.} & \langle \mathbf{A}_i, \mathbf{X} \rangle = b_i, \; i = 1, \ldots, m & & \text{s.t.} & \mathbf{Z} = \sum_{i=1}^{m} y_i \mathbf{A}_i - \mathbf{C} & \\ & \mathbf{X} \succeq 0 & & & \mathbf{Z} \succeq 0 & \end{array}$$

where $\langle \mathbf{A}, \mathbf{B} \rangle = \mathbf{A} \bullet \mathbf{B} = \operatorname{trace} \mathbf{AB} = \sum_{i,j} A_{ij} B_{ij}$,
all matrices are square and symmetric ($\mathcal{S}^n$),
and $\mathbf{X} \succeq 0$ denotes that $\mathbf{X}$ is positive semidefinite.

Importance of SDP

SDP is an important class of optimization problems for several reasons:

1. Because SDP problems are solvable in polynomial time, any problem that can be expressed using SDP is also solvable in polynomial time.
2. SDP problems can be solved efficiently in practice. This can be done by using one of the software packages available, or alternatively by implementing a suitable algorithm.
3. SDP can be used to obtain tight approximations for hard problems in integer and global optimization.
4. SDP problems are useful for a wide range of practical applications in areas such as control theory, portfolio optimization, truss topology design, and principal component analysis.

Conic Terminology

Both the non-negative orthant

$$\mathcal{N} = \{\mathbf{x} \in \mathbb{R}^n : x_i \geq 0 \text{ for all } i\}$$

and the set of symmetric positive semidefinite (psd) matrices

$$\mathcal{P} = \{\mathbf{X} \in \mathcal{S}^n : \lambda_i(\mathbf{X}) \geq 0 \text{ for all } i\}$$

are pointed closed convex cones.

A set $\mathcal{K}$ is

- a *cone* if $x \in \mathcal{K}$, then $\alpha x \in \mathcal{K}$ for all $\alpha \geq 0$
- *convex* if $\overline{x}, \overline{y} \in \mathcal{K}$, then $\alpha\overline{x} + (1-\alpha)\overline{y} \in \mathcal{K}$ for all $\alpha \in (0, 1)$
- *closed* if it contains its boundary
- *pointed / proper* if $\mathcal{K} \cap -\mathcal{K} = \{0\}$

Interior and Boundary of $\mathcal{P}$

The interior of $\mathcal{P}$ is the set of positive definite (pd) matrices:

$$\{\mathbf{X} \in \mathcal{S}^n : \lambda_i(\mathbf{X}) > 0 \text{ for all } i\}$$

and the boundary of $\mathcal{P}$ are the singular psd matrices.

Algebraic Characterizations

There are many ways to express the psd (pd) condition on a matrix $\mathbf{X}$. A few of them are:

$$\mathbf{X} \succeq 0 \Leftrightarrow \lambda_i(\mathbf{X}) \geq 0 \text{ for all } i \quad (\lambda_i(\mathbf{X}) > 0)$$

$$\Leftrightarrow \mathbf{v}^T\mathbf{X}\mathbf{v} \geq 0 \text{ for all } \mathbf{v} \in \mathbb{R}^n \quad (\mathbf{v}^T\mathbf{X}\mathbf{v} > 0)$$

$$\Leftrightarrow \exists\ \mathbf{X}^{\frac{1}{2}} \in \mathcal{S}^n \text{ s.t. } \mathbf{X}^{\frac{1}{2}}\mathbf{X}^{\frac{1}{2}} = \mathbf{X} \quad (\& \ \mathbf{X}^{\frac{1}{2}} \text{ is invertible})$$

$$\Leftrightarrow \exists\ \mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_n \text{ s.t. } X_{ij} = \mathbf{w}_i^T\mathbf{w}_j \quad (\& \ \mathbf{w}_1, \ldots, \mathbf{w}_n \text{ lin. indep.})$$

If $\mathbf{X}_1 \succ \mathbf{0}$ then

$$\begin{bmatrix} \mathbf{X}_1 & \mathbf{X}_2 \\ \mathbf{X}_2^T & \mathbf{X}_3 \end{bmatrix} \succeq \mathbf{0} \quad (\succ \mathbf{0})$$

$$\Leftrightarrow \mathbf{X}_3 - \mathbf{X}_2^T\mathbf{X}_1^{-1}\mathbf{X}_2 \succeq \mathbf{0} \quad (\mathbf{X}_3 - \mathbf{X}_2^T\mathbf{X}_1^{-1}\mathbf{X}_2 \succ \mathbf{0})$$

Sufficient (but not necessary) condition for psd (pd):

$$X_{ii} \geq \sum_{j \neq i} |X_{ij}| \text{ for all } i \Rightarrow \mathbf{X} \succeq \mathbf{0} \qquad \left(X_{ii} > \sum_{j \neq i} |X_{ij}| \Rightarrow \mathbf{X} \succ \mathbf{0}\right)$$

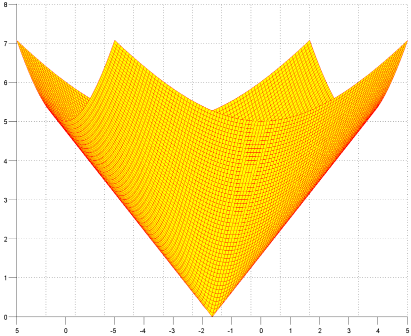
The Second-Order Cone (or Lorentz Cone)

An $(n+1)$-dimensional second-order cone (SOC) is the set of all vectors $(x_0, x_1, \ldots, x_n)$ that satisfy $x_0 \geq \sqrt{x_1^2 + \ldots + x_n^2}$, or equivalently

$$SOC = \{\mathbf{x} \in \mathbb{R}^{n+1} : x_0^2 - x_1^2 - \ldots - x_n^2 \geq 0, x_0 \geq 0\}.$$

The SOC is also a pointed closed convex cone, and second-order cone programming (SOCP) consists of optimizing a linear function subject to linear equality constraints and one or more SOC constraints.

The Lorentz Cone in 3 Dimensions



Relationships Between $\mathcal{N}$, $\mathcal{P}$, and the SOC

The SOC constraint

$$x_0^2 - x_1^2 - \ldots - x_n^2 \geq 0, x_0 \geq 0$$

is equivalent to the positive semidefinite constraint

$$\begin{pmatrix} x_0 & & & & x_1 \\ & x_0 & & & x_2 \\ & & x_0 & & x_3 \\ & & & \ddots & \vdots \\ x_1 & x_2 & x_3 & \cdots & x_0 \end{pmatrix} \succeq 0$$

Furthermore, the k-dimensional non-negative orthant is the direct (Cartesian) product of k 1-dimensional SOC cones.

Hence, SOCP is a special case of SDP;
and LP is a special case of SOCP.

A Fundamental Structure: The Elliptope

The $\binom{n}{2}$-dimensional elliptope (or spectrahedron) is the feasible set of the SDP problem

$$\begin{array}{ll} \min & \langle \mathbf{C}, \mathbf{X} \rangle \\ \text{s.t.} & \text{diag}(\mathbf{X}) = \mathbf{e} \\ & \mathbf{X} \succeq \mathbf{0}. \end{array}$$

where $\text{diag}(\mathbf{X})$ denotes a vector with the diagonal elements of X, and $\mathbf{e}$ is the vector of all ones.

In other words, the elliptope of dimension $\binom{n}{2}$ is the set of all symmetric $n \times n$ matrices that are psd and have ones on the diagonal.

This special set comes up in many applications of SDP.

Small Elliptopes

If $\mathbf{X} \in \mathcal{S}^2$, we obtain the elliptope in $\mathbb{R}^1$:

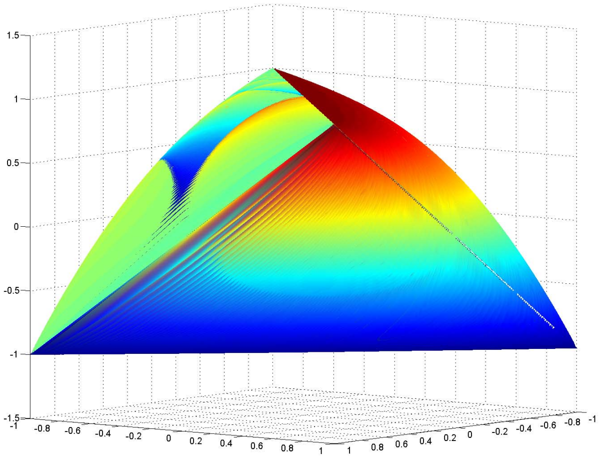
$$\begin{pmatrix} 1 & x \\ x & 1 \end{pmatrix} \succeq \mathbf{0} \Rightarrow x \in [-1, 1].$$

If $\mathbf{X} \in \mathcal{S}^3$, we obtain the elliptope in $\mathbb{R}^3$:

$$\left\{ \begin{pmatrix} x \\ y \\ z \end{pmatrix} \in \mathbb{R}^3 : \begin{pmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{pmatrix} \succeq \mathbf{0} \right\}.$$

We can visualize this set in $\mathbb{R}^3$.

The Elliptope in 3 Dimensions



Geometry of the Elliptope

The vertices of the elliptope correspond to the psd matrices with all entries equal to ± 1.

For $n = 2$, we have two vertices:

$$\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

For $n = 3$, there are four vertices:

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 1 & -1 \\ 1 & 1 & -1 \\ -1 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & -1 & -1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{pmatrix}.$$

Geometry of the Elliptope

Unlike a polyhedron, the elliptope has extreme points that are *not* vertices.

This occurs first when $n = 3$: the matrix

$$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix}$$

is not a vertex, but it is an extreme point of the elliptope since it cannot be expressed as a convex combination of the four vertices.

The Dual Cone

For any convex cone $\mathcal{K}$, the dual cone $\mathcal{K}^*$ is defined as

$$\mathcal{K}^* := \{\mathbf{y} \in \mathbb{R}^n : \langle \mathbf{x}, \mathbf{y} \rangle \geq 0 \quad \forall \mathbf{x} \in \mathcal{K}\}.$$

We have that:

- For any cone $\mathcal{K}$, $\mathcal{K}^*$ is a closed convex cone;
- The non-negative orthant is self-dual (obvious);
- The SOC is self-dual, by the Cauchy-Schwarz inequality;
- The psd cone is self-dual, by Fejer's Theorem:

$$\mathbf{X} \succeq 0 \Leftrightarrow \mathbf{X} \bullet \mathbf{Z} \geq 0 \text{ for all } \mathbf{Z} \succeq 0$$

Other Important Examples of Cones of Matrices

- The completely positive cone:

$$\mathcal{C} := \{\mathbf{X} \in \mathcal{S}^n : \mathbf{X} = \sum_{i=1}^{k} \mathbf{v}_i \mathbf{v}_i^T, \mathbf{v}_i \geq 0\}$$

- The copositive cone:

$$\mathcal{C}^* := \{\mathbf{X} \in \mathcal{S}^n : \mathbf{v}^T \mathbf{X} \mathbf{v} \geq 0 \text{ for all } \mathbf{v} \geq 0\}$$

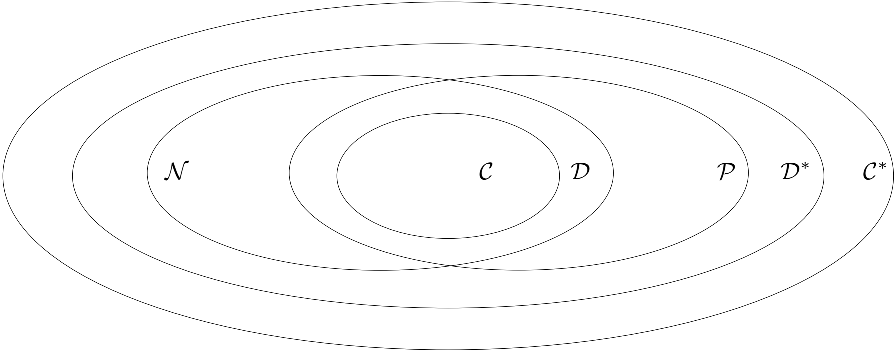
- The doubly non-negative cone:

$$\mathcal{D} := \mathcal{P} \cap \mathcal{N}$$

- and its dual

$$\mathcal{D}^* = \mathcal{P} \oplus \mathcal{N}$$

Relationships Between the Cones for $n \geq 5$



Conic Optimization Duality

Consider the general conic optimization problem

$$\begin{array}{ll} \min & \langle \mathbf{c}, \mathbf{x} \rangle \\ \text{s.t.} & \langle \mathbf{a}_i, \mathbf{x} \rangle = b_i, \; i = 1, \ldots, m \quad \leftarrow y_i \\ & \mathbf{x} \in \mathcal{K} \end{array}$$

The Lagrangian dual is:

$$\max_y \left\{ \min_{\mathbf{x} \in \mathcal{K}} \langle \mathbf{c}, \mathbf{x} \rangle + \sum_{i=1}^m y_i \left(b_i - \langle \mathbf{a}_i, \mathbf{x} \rangle \right) \right\} = \max_y \left\{ \sum_{i=1}^m b_i y_i + \min_{\mathbf{x} \in \mathcal{K}} \langle \mathbf{c} - \sum_{i=1}^m y_i \mathbf{a}_i, \mathbf{x} \rangle \right\}$$

The inner minimization is unbounded below unless

$$\langle \mathbf{c} - \sum_{i=1}^m y_i \mathbf{a}_i, \mathbf{x} \rangle \geq 0 \text{ for all } \mathbf{x} \in \mathcal{K}$$

in which case the minimum is zero.

Conic Optimization Duality (ctd)

Since the outer problem is a maximization problem, it is therefore equivalent to

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} b_i y_i \\ \text{s.t.} \quad & \mathbf{c} - \sum_{i=1}^{m} y_i \mathbf{a}_i \in \mathcal{K}^* \end{aligned}$$

which is equivalent to

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} b_i y_i \\ \text{s.t.} \quad & \sum_{i=1}^{m} y_i \mathbf{a}_i + \mathbf{z} = \mathbf{c} \\ & \mathbf{z} \in \mathcal{K}^* \end{aligned}$$

This is the dual cone optimization problem.

Why Focus on Semidefinite Optimization?

Convexity is of paramount importance in optimization.

Convex optimization problems have many of the advantageous properties of LP, including:

- an elegant and powerful duality theory, and
- polynomial-time solvability using interior-point methods (IPMs) – but with a major caveat.

Self-Concordant Barrier Functions

Use of an IPM requires a self-concordant barrier function for the cone underlying the feasible set.

Although such a function (the Universal Barrier Function) exists for a large variety of convex cones, it is very hard to compute in general.

However, efficient self-concordant barriers exist for symmetric cones.

Symmetric Cones

Symmetric cones arise from direct products of the following five types of cones:

- SOC
- symmetric psd matrices over the reals (psd cone)
- Hermitian psd matrices over the complex numbers (can be expressed as a psd cone of 2 times the size);
- Hermitian psd matrices over the quaternions (can be expressed as a psd cone of 4 times the size);
- One exceptional 27-dimensional cone (3×3 Hermitian psd matrices over the octonions).

Thus, SDP is (basically) the most general class of symmetric cones, and these are the cones over which we know how to optimize in polynomial time.

Solving SDP Problems

Numerous algorithms have been proposed (and, in most cases, implemented) for solving SDP problems:

- Ellipsoid method
- Interior-point methods (IPMs)
- Spectral bundle method
- Low-rank method
- Augmented Lagrangian methods
- Semi-infinite LP methods
- Boundary point method
- and more...

Solving SDP problems

Numerous algorithms have been proposed (and, in most cases, implemented) for solving SDP problems:

- Ellipsoid method
- Interior-point methods (IPMs)
- Spectral bundle method
- Low-rank method
- Augmented Lagrangian methods
- Semi-infinite LP methods
- Boundary point method
- and more...

The first two are the only ones with provable polynomial-time convergence.

IPMs are the only ones that are polynomial-time *and* efficient in practice.

IPM variants

Within the framework of IPMs, many variants have been proposed, analyzed, and implemented:

- Path-following
- Infeasible
- Potential reduction
- Dual scaling
- Primal-dual completion-based
- and more...

Modern LP software contains both simplex and interior-point solvers, often several variants for each.

Angelika Wiegele will speak about algorithms for SDP this afternoon.

SDP Duality

Consider a primal-dual SDP pair in the following form:

$$\begin{array}{lll|lll} \min & \langle \mathbf{C}, \mathbf{X} \rangle & & \max & \langle \mathbf{b}, \mathbf{y} \rangle \\ \text{s.t.} & \langle \mathbf{A}_i, \mathbf{X} \rangle = b_i, \ i = 1, \ldots, m & & \text{s.t.} & \sum_{i=1}^{m} y_i \mathbf{A}_i + \mathbf{Z} = \mathbf{C} \\ & \mathbf{X} \succeq \mathbf{0} & & & \mathbf{Z} \succeq \mathbf{0} \end{array}$$

Like in LP, we have a weak duality theorem.

Theorem

If $\tilde{\mathbf{X}}$ is primal feasible and $(\tilde{\mathbf{y}}, \tilde{\mathbf{Z}})$ is dual feasible then $\langle \mathbf{C}, \tilde{\mathbf{X}} \rangle \geq \langle \mathbf{b}, \tilde{\mathbf{y}} \rangle$.

The proof is just like for LP:

$$\langle \mathbf{C}, \tilde{\mathbf{X}} \rangle - \langle \mathbf{b}, \tilde{\mathbf{y}} \rangle = \langle \tilde{\mathbf{Z}}, \tilde{\mathbf{X}} \rangle + \sum_{i=1}^{m} \tilde{y}_i \langle \mathbf{A}_i, \tilde{\mathbf{X}} \rangle - \sum_{i=1}^{m} \tilde{y}_i \langle \mathbf{A}_i, \tilde{\mathbf{X}} \rangle = \langle \tilde{\mathbf{Z}}, \tilde{\mathbf{X}} \rangle \geq 0.$$

The difference between the primal and dual objective values for feasible solutions $\tilde{\mathbf{X}}$ and $(\tilde{\mathbf{y}}, \tilde{\mathbf{Z}})$ is called the duality gap.

Beyond weak duality, however, the picture differs. For example, for LP,

- if the primal is feasible and bounded, or
- if the dual is feasible and bounded,

then both primal and dual have optimal solutions, and the duality gap is zero at optimality.

For SDP, the situation is more complicated, as the following two examples demonstrate.

Zero Duality Gap Without Attainment

Consider the SDP problem

$$\begin{array}{ll} \inf & X_{11} \\ \text{s.t.} & \begin{pmatrix} X_{11} & 1 \\ 1 & X_{22} \end{pmatrix} \succeq 0. \end{array}$$

It is feasible and bounded yet the optimal value zero cannot be attained because $\begin{pmatrix} 0 & 1 \\ 1 & X_{22} \end{pmatrix}$ is not psd for any value of X_{22}.

The dual problem is

$$\begin{array}{ll} \max & y_1 \\ \text{s.t.} & \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} - y_1 \begin{pmatrix} 0 & \frac{1}{2} \\ \frac{1}{2} & 0 \end{pmatrix} \succeq 0 \end{array}$$

or equivalently

$$\begin{array}{ll} \max & y_1 \\ \text{s.t.} & \begin{pmatrix} 1 & -\frac{y_1}{2} \\ -\frac{y_1}{2} & 0 \end{pmatrix} \succeq 0 \end{array}$$

with $y_1^* = 0$ optimal. (In fact, $y_1 = 0$ is the only feasible solution.)

Positive Duality Gap

The SDP problem

$$\begin{array}{ll} \min & X_{11} \\ \text{s.t.} & X_{11} + 2X_{23} = 1 \\ & X_{22} = 0 \\ & \begin{pmatrix} X_{11} & X_{12} & X_{13} \\ X_{12} & X_{22} & X_{23} \\ X_{13} & X_{23} & X_{33} \end{pmatrix} \succeq 0 \end{array}$$

has optimal value 1.
The dual SDP problem is

$$\begin{array}{ll} \max & y_1 \\ \text{s.t.} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} - y_1 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & \frac{1}{2} & 0 \end{pmatrix} - y_2 \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 1 - y_1 & 0 & 0 \\ 0 & -y_2 & -\frac{y_1}{2} \\ 0 & \frac{y_1}{2} & 0 \end{pmatrix} \end{array}$$

and the psd constraint implies $y_1 = 0$ for every feasible solution, hence the optimal value is 0. (Take e.g. $y_1^* = 0, y_2^* = 0$.)

Constraint Qualification

To avoid this kind of difficulty and obtain a strong duality result for SDP, we must require that the primal-dual SDP pair satisfy a *constraint qualification (CQ)*.

This is a standard concept in non-linear optimization. Arguably the most commonly used CQ is Slater's CQ.

Definition

Slater's CQ holds if both primal and dual have feasible positive definite matrices.

We then have the following result.

Theorem

Under Slater's CQ, both primal and dual have optimal solutions, and the duality gap is zero at optimality.

Verifying Slater's CQ

Example

$$\begin{array}{lll|lll} \min & \langle \mathbf{C}, \mathbf{X} \rangle & & \max & \langle \mathbf{e}, \mathbf{y} \rangle \\ s.t. & \langle \mathbf{e}_i \mathbf{e}_i^T, \mathbf{X} \rangle = 1, \; i = 1, \ldots, m & & s.t. & \sum_{i=1}^{m} y_i \mathbf{e}_i \mathbf{e}_i^T + \mathbf{Z} = \mathbf{C} \\ & \mathbf{X} \succeq \mathbf{0} & & & \mathbf{Z} \succeq \mathbf{0} \end{array}$$

Optimality Conditions

From the weak duality of SDP, we have that the duality gap equals

$$\langle \mathbf{C}, \tilde{\mathbf{X}} \rangle - \mathbf{b}^T \tilde{\mathbf{y}} = \langle \tilde{\mathbf{Z}}, \tilde{\mathbf{X}} \rangle \geq 0.$$

Since both $\mathbf{X}$ and $\mathbf{Z}$ are psd, $\langle \mathbf{X}, \mathbf{Z} \rangle = 0$ implies $\mathbf{XZ} = \mathbf{ZX} = 0$, thus we obtain the sufficient optimality conditions:

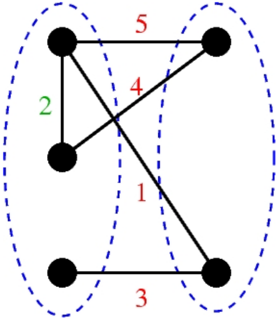
$$\begin{array}{ll}
\langle \mathbf{A}_i, \mathbf{X} \rangle = b_i, i = 1, \ldots, m, & \text{(primal feasibility)} \\
\mathbf{X} \succeq 0 & \\
\mathbf{Z} + \sum_{i=1}^{m} y_i \mathbf{A}_i = \mathbf{C} & \text{(dual feasibility)} \\
\mathbf{Z} \succeq 0 & \\
\mathbf{XZ} = 0 & \text{(complementarity)}
\end{array}$$

If Slater's CQ holds, they are also necessary for optimality.
These optimality conditions can be used as the starting point for defining IPMs to solve SDP problems.

A Key Result in the History of SDP: The Goemans-Williamson Approximation Algorithm for the Maximum-Cut (Max-Cut) Problem

The Max-Cut Problem

Given a graph $G = (V, E)$ and weights w_{ij} for all edges $(i, j) \in E$, find an edge-cut of maximum weight, i.e. find a set $S \subseteq V$ s.t. the sum of the weights of the edges with one end in S and the other in $V \setminus S$ is maximum.
We assume wlog that $w_{ii} = 0$ for all $i \in V$, and that G is complete (assign $w_{ij} = 0$ if edge $ij \notin E$).



Standard Integer LP Formulation

$$
\begin{array}{ll}
\max & \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} y_{ij} \\
\text{s.t.} & y_{ij} + y_{ik} + y_{jk} \le 2, 1 \le i < j < k \le n \\
& y_{ij} - y_{ik} - y_{jk} \le 0, 1 \le i < j \le n, k \ne i, j \\
& y_{ij} \in \{0, 1\}, 1 \le i < j \le n
\end{array}
$$

where

$$
y_{ij} = \begin{cases} 1 & \text{if edge } ij \text{ is cut} \\ 0 & \text{otherwise,} \end{cases}
$$

$y_{ij} = y_{ji}$, and w_{ij} denotes the weight of edge ij.

This formulation is the basis for a highly successful branch-and-cut algorithm for solving spin glass problems in physics (Liers, Jünger, Reinelt and Rinaldi (2005)).

The solver can be accessed online at the Spin Glass Server:

`http://www.informatik.uni-koeln.de/spinglass/`

Quadratic Formulation of Max-Cut

Whereas the ILP formulation is edge-based, we use a node-based quadratic formulation.

- Let the vector $\mathbf{v} \in \{-1, +1\}^n$ represent any cut in the graph via the interpretation that the sets $\{i|v_i = +1\}$ and $\{i|v_i = -1\}$ specify the partition.
- Then max-cut may be formulated as:

$$\begin{array}{ll} \max & \sum_{i=1}^{n} \sum_{j=i+1}^{n} w_{ij} \left(\frac{1-v_i v_j}{2} \right) \\ \text{s.t.} & v_i^2 = 1, i = 1, \ldots, n. \end{array}$$

The Basic Semidefinite Relaxation of Max-Cut

Consider the change of variable $\mathbf{X} = \mathbf{v}\mathbf{v}^T, \mathbf{v} \in \{\pm 1\}^n$.
Then $X_{ij} = v_i v_j$ and max-cut is equivalent to

$$\begin{array}{ll} \max & \langle \mathbf{Q}, \mathbf{X} \rangle \\ \text{s.t.} & \operatorname{diag}(\mathbf{X}) = \mathbf{e} \\ & \operatorname{rank}(\mathbf{X}) = 1 \\ & \mathbf{X} \succeq \mathbf{0}, \end{array}$$

where $\mathbf{Q} = \frac{1}{4}\left(\operatorname{Diag}(\mathbf{We}) - \mathbf{W}\right)$.

Removing the rank constraint, we obtain the basic SDP relaxation of max-cut.

Question: How good is this SDP relaxation?

Goemans and Williamson (1995): 0.878-approximation algorithm

Theorem (Goemans and Williamson (1995))

If $w_{ij} \geq 0$ for all edges ij, then

$$\frac{\textit{max-cut opt value}}{\textit{SDP relax opt value}} \geq \alpha$$

where $\alpha := \min_{0 \leq \xi \leq \pi} \frac{2}{\pi} \frac{\xi}{1 - \cos \xi} \approx 0.87856.$

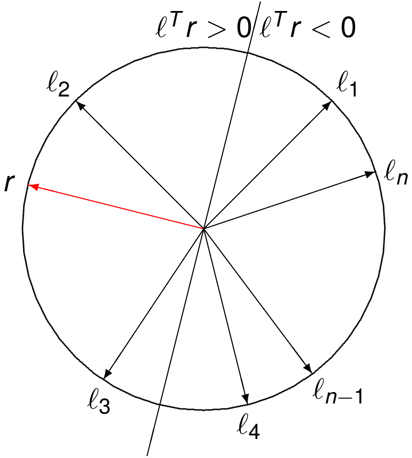
This performance ratio is best-possible if the Unique Games Conjecture is true.

Goemans and Williamson (1995): 0.878-approximation algorithm (ctd)

In fact, Goemans and Williamson proved a stronger result:
they described a randomized algorithm that

- from an optimal solution $\mathbf{X}^*$ of the SDP relaxation
- *generates a cut* with expected weight $\geq \alpha$ (SDP relax opt value).

Using the fact that $\mathbf{X}^* \succeq 0 \Rightarrow \exists\ \ell_1, \ell_2, \ldots, \ell_n$ s.t. $X_{ij} = \ell_i^T \ell_j$



Since the optimal max-cut value is at least the expected value of this cut, the theorem follows.

Constant Relative Accuracy Estimate

With no restriction on the edge weights, Nesterov (1998) proved constant relative accuracy estimates for the basic SDP bound. Define

$$\mu^* = \begin{array}{ll} \max & \mathbf{v}^T \mathbf{Q} \mathbf{v} \\ \text{s.t.} & \mathbf{v} \in \{-1, 1\}^n \end{array} \qquad \mu_* = \begin{array}{ll} \min & \mathbf{v}^T \mathbf{Q} \mathbf{v} \\ \text{s.t.} & \mathbf{v} \in \{-1, 1\}^n \end{array}$$

$$\psi^* = \begin{array}{ll} \max & \langle \mathbf{Q}, \mathbf{X} \rangle \\ \text{s.t.} & \operatorname{diag}(\mathbf{X}) = \mathbf{e},\ \mathbf{X} \succeq 0 \end{array} \qquad \psi_* = \begin{array}{ll} \min & \langle \mathbf{Q}, \mathbf{X} \rangle \\ \text{s.t.} & \operatorname{diag}(\mathbf{X}) = \mathbf{e}, \mathbf{X} \succeq 0 \end{array}$$

and

$$s(\beta) := \beta\, \psi^* + (1 - \beta)\, \psi_*,\ \ \beta \in [0, 1].$$

Theorem (Nesterov (1998))

Without any assumption on the matrix $\mathbf{Q}$,

$$\frac{|s(\frac{2}{\pi}) - \mu^*|}{\mu^* - \mu_*} \leq \frac{\pi}{2} - 1 < \frac{4}{7}.$$

Some Recent Highlights

There has been tremendous activity on SDO and max-cut since 1995. Some recent computational highlights:

- The SDP relaxation, augmented with selected inequalities, is a key ingredient of the max-cut solver *Biqmac* (Rendl, Rinaldi and Wiegele (2007)):

  `http://biqmac.uni-klu.ac.at/`

- The generalization of the SDP relaxation to max-k-cut works very well on dense graphs with branch-and-cut and the bundle method (Anjos, Ghaddar, Hupp, Liers, Wiegele (2013)).

This basic relaxation of max-cut is also the basis for solution approaches to other problems. Two recent examples are:

- Min-bisection problems (Armbruster, Helmberg, Fügenschuh and Martin (2011))
- Single- and multi-row facility layout problems (Anjos and several co-authors, particularly Hungerländer)

A Very Brief Introduction to Polynomial Optimization...

... with a focus on binary problems.

Polynomial Optimization

Polynomial optimization problems (POPs) consist of optimizing a multivariate polynomial subject to multivariate polynomial constraints:

$$\begin{aligned} z = \quad & \sup \quad f(x) \\ & \text{s.t.} \quad g_i(x) \geq 0 \qquad i = 1, \ldots, m. \end{aligned}$$

Numerous classes of problems can be modelled as POPs, including:

- Linear Problems
- Quadratic Problems (Convex / Non-convex)
- Mixed-Binary Problems

$$x_i \in \{0, 1\} \quad \Leftrightarrow \quad x_i^2 - x_i = 0$$

Thus, solving POPs is in general NP-hard.

General POP Perspective

Given a general POP problem:

$$\text{(POP)} \quad z = \begin{array}{ll} \sup & f(x) \\ \text{s.t.} & g_i(x) \geq 0 \qquad i = 1, \ldots, m. \end{array}$$

If λ is the optimal value of POP, then POP is equivalent to

$$\begin{array}{ll} \inf & \lambda \\ \text{s.t.} & \lambda - f(x) \geq 0 \quad \forall x \in S := \{x : g_i(x) \geq 0,\ i = 1, \ldots, m\} \end{array}$$

which we rewrite as

$$\begin{array}{ll} \inf & \lambda \\ \text{s.t.} & \lambda - f(x) \in \mathcal{P}_d(S) \end{array}$$

where

$$\mathcal{P}_d(S) = \{p(x) \in \mathbf{R}_d[x] : p(s) \geq 0 \text{ for all } s \in S\}$$

is the cone of polynomials of degree $\leq d$ that are non-negative over S.

Main question:

How to relax the condition $\lambda - f(x) \in \mathcal{P}_d(S)$?

Conic Relaxations of POP

We relax $\lambda - f(x) \in \mathcal{P}_d(S)$ to

$$\lambda - f(x) \in \mathcal{K} \text{ for a suitable cone } \mathcal{K} \subseteq \mathcal{P}_d(S).$$

Then the conic optimization problem

$$\begin{array}{ll} \inf & \lambda \\ \text{s.t.} & \lambda - f(x) \in \mathcal{K} \end{array}$$

provides an upper bound for the original problem.

- The choice of $\mathcal{K}$ is key to obtaining good bounds on the problem
- Optimizing over $\mathcal{K}$ should (must?) be tractable

Example: Application to Max-cut

The formulation

$$\begin{aligned} & \max \ \mathbf{x}^T \mathbf{Q} \mathbf{x} \\ & \text{s.t. } x_i^2 = 1, i = 1, \ldots, n \end{aligned}$$

can be recast as

$$\begin{aligned} & \min \ \lambda \\ & \text{s.t. } \lambda - \mathbf{x}^T \mathbf{Q} \mathbf{x} \in \mathcal{P}_2 \left(\{ \mathbf{x} : x_i^2 = 1, i = 1, \ldots, n \} \right) \end{aligned}$$

The $r = 2$ SOS relaxation is precisely the dual SDP problem of the basic SDP relaxation.

SOS Approach - Lasserre (2001), Parrilo (2000)

For each $r > 0$, define the approximation $\mathcal{K}_r \subseteq \mathcal{P}_d(S)$ as

$$\mathcal{K}_r := \left(\mathrm{SOS}_r + \sum_{i=1}^{m} g_i(x) \mathrm{SOS}_{r-\deg(g_i)} \right) \cap \mathbf{R}_d[x]$$

where SOS_d denotes the cone of real polynomials of degree at most d that are SOSs of polynomials, and $\mathbf{R}_d[x]$ denotes the set of polynomials in the variables x of degree at most d.

The corresponding relaxation can be written as

$$\begin{array}{llll} (\mathrm{L}_r) & z_r = & \inf\limits_{\lambda,\sigma_i} & \lambda \\ & & \text{s.t.} & \lambda - f(x) = \sigma_0(x) + \sum_{i=1}^{m} \sigma_i(x) g_i(x) \\ & & & \sigma_0(x) \text{ is SOS of degree} \leq r \\ & & & \sigma_i(x) \text{ is SOS of degree} \leq r - \deg(g_i(x)), i = 1, \ldots, m. \end{array}$$

Solving the SOS Relaxation

For each r, the relaxation (L_r) can be cast as an SDP problem, since $\sigma(x)$ is a SOS of degree $2k$ if and only if

$$\sigma(x) = \begin{pmatrix} 1 \\ \vdots \\ x_i \\ \vdots \\ x_i x_j \\ \vdots \\ \prod_{|k|} x \end{pmatrix}^T M \begin{pmatrix} 1 \\ \vdots \\ x_i \\ \vdots \\ x_i x_j \\ \vdots \\ \prod_{|k|} x \end{pmatrix} \quad \text{with} \quad M \succeq 0.$$

Note that $\mathrm{SOS}_d = \mathrm{SOS}_{d-1}$ for every odd degree d.

Lasserre's Hierarchy for a Small Example

To solve

$$\begin{array}{ll} \sup_{x,y} & -(x-1)^2-(y-1)^2 \\ \text{s.t.} & x^2-4xy-1 \geq 0 \\ & yx-3 \geq 0 \\ & y^2-4 \geq 0 \\ & 12^2-(x-2)^2-4(y-1)^2 \geq 0 \end{array}$$

r	2	4	6
# vars	14	73	245
# constraints	6	15	28
Bound	9.40	36.06	51.73

There is no need to run relaxations for $r > 6$, because an optimal solution (and optimality certificate) can be extracted from the solution to the SDP problem L_6.

Assessing the SOS Relaxation

Good news: Under mild conditions, $z_r \to z$.

Bad news: For a problem with n variables and m inequality constraints, the size of the relaxation is:

- One psd matrix of dimension $\binom{n+r}{r}$;
- m psd matrices, each of dimension $\binom{n+r-\deg(g_i)}{r-\deg(g_i)}$
- $\binom{n+r}{r}$ linear constraints.

To overcome the size blow-up one may exploit the structure (sparsity, symmetry, convexity) to get smaller SDP programs.
Many authors have contributed here: Gatermann, Helton, Kim, Kojima, Lasserre, Netzer, Nie, Parrilo, Pasechnik, Riener, Schweighofer, Sotirov, Theobald, etc.

One idea: Improve the bound without growing r

Recall

$$(\text{POP}) \quad z = \sup \; f(x) \quad \text{s.t.} \; x \in S := \{x : g_i(x) \geq 0, \; i = 1, \ldots, m\}$$

$$(\text{L}_r(G)) \quad z_r(G) = \inf_{\lambda, \sigma_i} \; \lambda$$
$$\text{s.t.} \quad \lambda - f(x) = \sigma_0(x) + \textstyle\sum_{i=1}^m \sigma_i(x) g_i(x)$$
$$\sigma_0(x) \text{ is SOS of degree} \leq r$$
$$\sigma_i(x) \text{ is SOS of degree} \leq r - \deg(g_i(x)),$$
$$i = 1, \ldots, m.$$

Observe that

- (L_r) is defined in terms of the functions used to describe S
- Call this set $G = \{g_i(x) : i = 1, \ldots, m\}$

Ghaddar, Vera, Anjos (2011): Improve the description of S by growing G in such a way that the bound obtained from L_r improves for fixed r.

Another idea: Pure SOC Relaxations

For binary quadratic POPs, it is possible to obtain purely second-order cone (SOC) relaxations.

Let us separate the various types of constraints as follows:

$$
\begin{array}{lll}
\max & x^T Q x + p^T x & \\
\text{s.t.} & a_j^T x = b_j & \forall j \in \{1, \ldots, t\} \\
& c_j^T x \leq d_j & \forall j \in \{1, \ldots, u\} \\
& x^T F_j x + e_j^T x = k_j & \forall j \in \{1, \ldots, v\} \\
& x^T G_j x + h_j^T x \leq l_j & \forall j \in \{1, \ldots, w\} \\
& x_i \in \{-1, 1\} & \forall i \in \{1, \cdots, n\}
\end{array}
$$

Useful Lemma

$x \in \{-1, 1\}^n \Rightarrow \|x\|^2 = n$.

This leads to the following specialized lemma that we will use:

Lemma

If $f(x)$ is a polynomial of degree one and $\mathcal{B}' := \{x : \|x\|^2 = n\}$, then

$$f(x) \in \mathcal{P}_1(\mathcal{B}') \text{ if and only if } f(x) = f^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix}$$

with $f \in SOC^{n+1}$.

First Relaxation

Using the previous lemma, we obtain **(BQPP$_{\text{ss}}$)**:

$$
\begin{aligned}
&\min \lambda \\
&\text{s.t. } \lambda - (x^T Q x + p^T x) = \begin{pmatrix} 1 & x^T \end{pmatrix} M \begin{pmatrix} 1 \\ x \end{pmatrix} \\
&\qquad + \sum_i (1 + x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1 - x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \gamma_i (1 - x_i^2) \\
&\qquad + \sum_j \delta_j(x)(b_j - a_j^T x) + \sum_j (d_j - c_j^T x)\eta_j^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
&\qquad + \sum_j \theta_j (k_j - x^T F_j x - e_j^T x) + \sum_j \xi_j (l_j - x^T G_j x - h_j^T x), \\
&M \in \mathcal{S}_+^{n+1}, \quad \alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \quad \delta_j \in \mathbb{R}_1[x], \quad \gamma_i, \theta_j \in \mathbb{R}, \quad \xi_j \in \mathbb{R}_+
\end{aligned}
$$

Second (Improved) Relaxation

Adding products of linear constraints strengthens further: $\mathbf{(BQPP_{SS^+})}$

$$\min \lambda \quad \text{s.t. } \lambda - (x^T Q x + p^T x) = \begin{pmatrix} 1 & x^T \end{pmatrix} M \begin{pmatrix} 1 \\ x \end{pmatrix}$$

$$+ \sum_i (1 + x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1 - x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \gamma_i (1 - x_i^2)$$

$$+ \sum_j \delta_j(x)(b_j - a_j^T x) + \sum_j (d_j - c_j^T x)\eta_j^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix}$$

$$+ \sum_j \theta_j (k_j - x^T F_j x - e_j^T x) + \sum_j \xi_j (l_j - x^T G_j x - h_j^T x)$$

$$+ \sum_{i,k} \sigma_{ik} (d_k - c_k^T x)(1 + x_i) + \sum_{i,k} \mu_{ik} (d_k - c_k^T x)(1 - x_i)$$

$$+ \sum_{k \le l} \nu_{kl} (d_k - c_k^T x)(d_l - c_l^T x) + \sum_{i \le j} \tau_{ij} (1 - x_i)(1 - x_j)$$

$$+ \sum_{i \le j} \omega_{ij} (1 + x_i)(1 + x_j) + \sum_{i,j} \phi_{ij} (1 - x_i)(1 + x_j)$$

$$M \in \mathcal{S}_+^{n+1}, \qquad \alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \qquad \gamma_i, \theta_j \in \mathbb{R}, \qquad \xi_j, \sigma_{ik}, \mu_{ik}, \nu_{kl}, \tau_{ij}, \omega_{ij}, \phi_{ij} \in \mathbb{R}_+$$

Pure SOC Relaxation

We can relax (BQPP_{SS}) by removing the SOS term.
In the absence of the SOS term, the valid inequalities

$$-1 \le x_i x_j \le 1$$

are no longer satisfied, and may strengthen the relaxation **(BQPP_{soc})**:

$$\begin{aligned}
\min \lambda \quad & \text{s.t. } \lambda - (x^T Q x + p^T x) = \\
& \sum_i (1 + x_i)\alpha_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i (1 - x_i)\beta_i^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} + \sum_i \gamma_i (1 - x_i^2) \\
& + \sum_j \delta_j(x)(b_j - a_j^T x) + \sum_j (d_j - c_j^T x)\eta_j^T \begin{pmatrix} \sqrt{n} \\ x \end{pmatrix} \\
& + \sum_j \theta_j (k_j - x^T F_j x - e_j^T x) + \sum_j \xi_j (l_j - x^T G_j x - h_j^T x) \\
& + \sum_{i<j} \nu_{ij}^+ (1 + x_i x_j) + \sum_{i<j} \nu_{ij}^- (1 + x_i x_j)
\end{aligned}$$

$$\alpha_i, \beta_i, \eta_j \in \mathcal{L}^{n+1}, \quad \delta_j \in \mathbb{R}_1[x], \quad \gamma_i, \theta_j \in \mathbb{R}, \quad \xi_j, \nu_{ij}^+, \nu_{ij}^- \in \mathbb{R}_+$$

Lasserre Relaxation for BQPPs

Lasserre introduced SDP relaxations for binary polynomial programs of the form:

$$\Gamma_r := \Big(\Psi_{r+2} + \sum_i (1 - x_i^2)\Psi_r + \sum_i (x_i^2 - 1)\Psi_r + \sum_i (b_i - a_i^T x)\Psi_r$$
$$+ \sum_i (a_i^T x - b_i)\Psi_r + \sum_i (d_i - c_i^T x)\Psi_r + \sum_i (k_i - x^T F_i x - e_i^T x)\Psi_r$$
$$+ \sum_i (x^T F_i x + e_i^T x - k_i)\Psi_r + \sum_i (l_i - x^T G_i x - h_i^T x)\Psi_r \Big) \cap \mathbb{R}_2[x],$$

for even $r \geq 0$. Taking $r = 0$, we obtain a relaxation for BQPP:

$$\textbf{(BQPP}_{\textbf{Las}}\textbf{)} \text{ min } \lambda \text{ s.t. } \lambda - q(x) \in \Gamma_0$$

Theorem

$$\lambda^*_{BQPP_{Las}} \geq \lambda^*_{BQPP_{SS}} \geq \lambda^*_{BQPP_{SS^+}} \geq z^*_{BQPP}$$

Computational Setup

- All relaxations were implemented in Matlab 7.9.0
- SeDuMi 1.3 was used to solve the conic programming problems
- All computations were carried out on a 1200 MHz Sun Sparc machine

Computational Results for General BQPP Problems

For each size n

- 100 randomly generated instances
- density (number of nonzero coefficients in the objective function) between 20% to 100%.
- the number of linear and quadratic constraints (m) varies from 1 to $\frac{n}{2}$.

We implemented Lasserre's relaxation using our code and the cone definition above.

The gaps (in %) are calculated as follows:

$$gap = 100 \times \frac{ub_{\text{relaxation}} - ub_{\text{best}}}{ub_{\text{best}}},$$

where the best upper bound is the one obtained by the ($BQPP_{SS^+}$) relaxation.

Computational Results for General BQPP

n	m	$(BQPP_{SS^+})$	$(BQPP_{SS})$		$(BQPP_{Las})$		$(BQPP_{SOC})$	
		Time	Gap	Time	Gap	Time	Gap	Time
40	1	78.18	1.66	56.30	34.89	34.59	**29.97**	33.07
	5	122.37	2.33	67.54	36.38	44.66	**28.18**	37.47
	20	306.31	5.71	88.80	50.60	48.27	**38.60**	44.11
50	1	268.93	0.68	179.74	**5.12**	112.49	15.16	48.72
	5	397.34	3.44	193.86	**17.71**	122.32	39.05	117.75
	25	1245.49	12.27	258.77	94.54	142.29	**43.08**	190.33
60	1	970.00	3.15	626.87	**19.61**	375.24	65.83	94.16
	5	1169.37	3.69	663.09	40.75	397.93	**39.75**	183.34
	30	5637.18	9.42	850.83	58.95	473.50	**52.10**	650.46
70	1	2793.31	0.93	2515.31	29.44	1214.23	31.51	165.63
	5	3848.18	2.50	2532.18	53.64	1245.09	**26.98**	549.22
	35	15420.53	14.85	2429.09	47.51	1446.99	**46.99**	1818.69

Summary of Computational Results for General BQPP

- ($BQPP_{SOC}$) is the most computationally efficient relaxation in most cases.
- ($BQPP_{SOC}$) frequently has better gaps than ($BQPP_{Las}$)
- With more linear constraints, ($BQPP_{Las}$) is slightly more efficient but its bounds are weaker than those provided by ($BQPP_{SOC}$).

In conclusion...

- Semidefinite, conic and polynomial optimization is a very active and exciting research area.
- There are still numerous open questions, both theoretical and algorithmic.
- Furthermore, this area is ripe for real-world applications.

Angelika Wiegele will cover algorithms for SDP in this afternoon's lectures.