Conic Optimization: Relaxing at the Cutting Edge

Miguel F. Anjos

Professor and Canada Research Chair

GERAD POLYTECHNIQUE MONTRÉAL NSERC CRSNG

Various parts are joint work with
E. Adams (Poly Mtl),
A. Engau (U. Colorado-Denver),
F. Rendl and A. Wiegele (Klagenfurt),
and
A. Vannelli (U. Guelph)

CanaDAM 2013 – Memorial U., NL, Canada – June 11, 2013

Outline

Background

Conic Optimization

Conic optimization refers to the problem of optimizing a linear (or possibly convex quadratic) function over the intersection of an affine space and a (pointed) closed convex cone:

$$\text{(P)} \quad \begin{array}{ll} \inf & \langle c, x\rangle \\ \text{s.t.} & \langle a_i, x\rangle = b_i, i = 1, \ldots, m \\ & x \in \mathcal{K} \end{array} \qquad \text{(D)} \quad \begin{array}{ll} \sup & b^T y \\ \text{s.t.} & \sum_{i=1}^{m} y_i a_i + s = c \\ & s \in \mathcal{K}^* \end{array}$$

where the dual cone $\mathcal{K}^*$ is defined as

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

- If $\mathcal{K} = \mathbb{R}^n_+$ then we have linear programming (LP)
- If $\mathcal{K} = \mathcal{S}^n_+$ then we have semidefinite programming (SDP)
- If $\mathcal{K} = SOC^n = \{\mathbf{x} \in \mathbb{R}^{n+1} : x_0 \geq \sqrt{x_1^2 + \ldots + x_n^2}\}$ then we have second-order cone programming (SOCP)

These cones are self-dual: $\mathcal{K} = \mathcal{K}^*$.

The SOC and the psd Cone

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

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

where $\succeq 0$ denotes positive semidefiniteness.
Hence, SOCP is a special case of SDP,
and LP is a special case of SOCP.

Why Conic Optimization?

Conic optimization problems share 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: 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 general convex sets, 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:

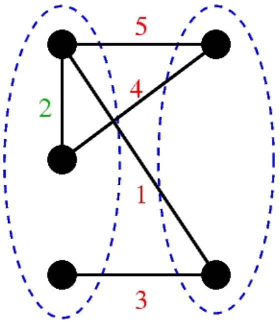
- second-order cones
- symmetric psd matrices over the reals (psd cone)
- Hermitian psd matrices over the complex numbers (can be expressed as a psd cone of twice the size);
- Hermitian psd matrices over the quaternions (can be expressed as a psd cone of four times the size);
- One exceptional 27-dimensional cone (3×3 Hermitian psd matrices over the octonions).

Thus, $\mathcal{S}_+^n$ is (basically) the most general class of symmetric cones.

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

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

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 $v \in \{-1, +1\}^n$ 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 $X = vv^T, v \in \{\pm 1\}^n$.
Then $X_{ij} = v_i v_j$ and max-cut is equivalent to

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

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

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

Goemans and Williamson (1995): 0.878-approximation algorithm

Theorem

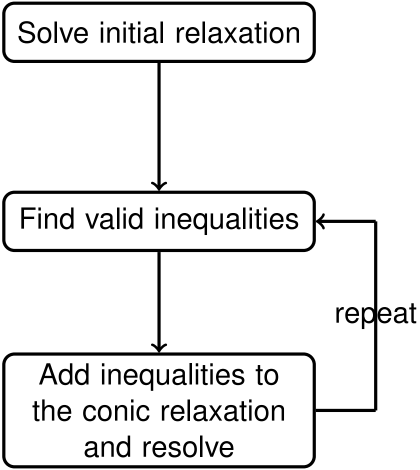
If $w_{ij} \geq 0$ for all edges ij, then

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

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

This result is much stronger than any similar result known for linear optimization relaxations.

Framework for a Practical Cutting-Plane Algorithm



$$\begin{array}{ll} \max & Q \bullet X \\ \text{s.t.} & \operatorname{diag}(X) = e \\ & X \succeq 0 \end{array}$$

Triangle Inequalities:

$$\begin{aligned} X_{ij} + X_{ik} + X_{jk} &\geq -1 \\ -X_{ij} + X_{ik} + X_{jk} &\geq -1 \\ X_{ij} - X_{ik} + X_{jk} &\geq -1 \\ X_{ij} + X_{ik} - X_{jk} &\geq -1 \end{aligned}$$

Selected Extensions

The basic 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/`

This basic relaxation of max-cut is also the basis for successful solution approaches to other problems, including:

- Max-k-cut problems (Ghaddar, A. and Liers (2007); A., Ghaddar, Hupp, Liers, Wiegele (2013))
- Min-bisection problems (Armbruster, Helmberg, Fügenschuh and Martin (2011))
- Single-row facility layout problems

Single-Row Facility-Layout Problem (SRFLP)

Problem Data:

- n one-dimensional facilities with positive lengths ℓ_i, $i = 1, \ldots, n$
- c_{ij} pairwise interaction costs

Decision Variables:

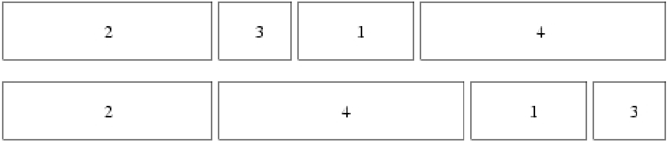
- permutation π

Problem Objective:

- minimize the total weighted sum of the center-to-center distances

$$\min_{\pi \in \Pi} \sum_{i<j} c_{ij}\left(\tfrac{1}{2}\ell_i + D_\pi(i,j) + \tfrac{1}{2}\ell_j\right) = \sum_{i<j} \tfrac{1}{2} c_{ij}(\ell_i + \ell_j) + \sum_{i<j} c_{ij} D_\pi(i,j)$$

where $D_\pi(i,j)$ is the sum of the lengths of the facilities between i and j.



Binary Quadratic Model (A., Kennings, Vannelli (2005))

$$R_{ij} = \begin{cases} -1 & \text{if facility } i \text{ is placed to the left of facility } j \\ 1 & \text{if facility } i \text{ is placed to the right of facility } j \end{cases}$$

- Facility k is between i and j if and only if $R_{ki}R_{kj} = -1$ so that

$$D(i,j) = \sum_{k \neq i,j} l_k \frac{(1 - R_{ki}R_{kj})}{2}$$

- If $R_{ik} = R_{kj}$, then $R_{ij} = R_{ik}$ which gives the necessary constraint

$$R_{ik}R_{kj} - R_{ik}R_{ij} - R_{ij}R_{kj} = -1 \text{ for all triples } i < k < j$$

$$\min \text{ const} - \sum_{i<j} \frac{c_{ij}}{2} \left(\sum_{k<i} l_k R_{ki}R_{kj} - \sum_{i<k<j} l_k R_{ik}R_{kj} + \sum_{k>j} l_k R_{ik}R_{jk} \right)$$
$$\text{s.t. } R_{ik}R_{kj} - R_{ik}R_{ij} - R_{ij}R_{kj} = -1 \text{ and } R_{ij}^2 = 1 \text{ for all } i < k < j$$

SDP Relaxation

The decision variable is

$$X = xx^T \text{ where } x = (R_{12}, \dots, R_{(n-1)n})^T \in \mathbb{R}^{\binom{n}{2}}$$

$$\min \text{ const} - \sum_{i<j} \frac{c_{ij}}{2} \left(\sum_{k<i} l_k X_{ki,kj} - \sum_{i<k<j} l_k X_{ik,kj} + \sum_{k>j} l_k X_{ik,jk} \right)$$

$$\text{s.t. } X_{ij,jk} - X_{ij,ik} + X_{ik,jk} = -1, \ \text{diag}(X) = e, \ X \succeq 0$$

- A cutting-plane algorithm using triangle inequalities can be applied and solve instances with up to 30 facilities to global optimality (A. and Vannelli (2008)).
- We can reduce the inequalities by summing over k

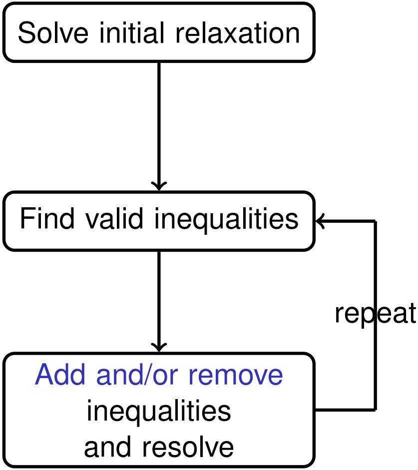
$$\sum_{k \neq i,j} (X_{ij,jk} - X_{ij,ik} + X_{ik,jk}) = -(n-2)$$

In this way, bounds for instances with up to 100 facilities can be obtained (A. and Yen (2009)).

Integrated Interior-Point Method with Cutting Planes

Recall: Cutting-Plane Algorithm Framework

$$\begin{array}{ll} \min & C \bullet X \\ \text{s.t.} & \mathcal{A}(X) = b \\ & X \succeq 0 \end{array}$$



This is challenging in practice as the number of inequalities increases.

Indicators: Active Set Detection in LP

Let $I_x = \{i : x_i^* > 0\}$ and $I_s = \{i : s_i^* > 0\}$ with (x^*, y^*, s^*) optimal to

$$\text{(LP)} \quad \min c^T x \text{ s.t. } Ax = b,\ x \geq 0$$
$$\text{(LD)} \quad \max b^T y \text{ s.t. } A^T y + s = c,\ s \geq 0$$

- Variable Indicator (folklore)

$$\lim_{k\to\infty} x_i^{(k)} \begin{cases} > 0 \\ = 0 \end{cases} \quad \text{and} \quad \lim_{k\to\infty} s_i^{(k)} \begin{cases} = 0 & \text{for } i \in I_x \\ > 0 & \text{for } i \in I_s \end{cases}$$

- Primal-Dual Indicator (Ye (1990), Gay (1991))

$$\lim_{k\to\infty} \frac{x_i^{(k)}}{s_i^{(k)}} = \begin{cases} \infty \\ 0 \end{cases} \quad \text{and} \quad \lim_{k\to\infty} \frac{s_i^{(k)}}{x_i^{(k)}} = \begin{cases} 0 & \text{for } i \in I_x \\ \infty & \text{for } i \in I_s \end{cases}$$

- Tapia Indicator (Tapia (1980), El-Bakry, Tapia and Zhang (1994))

$$\lim_{k\to\infty} \frac{x_i^{(k+1)}}{x_i^{(k)}} = \begin{cases} 1 \\ 0 \end{cases} \quad \text{and} \quad \lim_{k\to\infty} \frac{s_i^{(k+1)}}{s_i^{(k)}} = \begin{cases} 0 & \text{for } i \in I_x \\ 1 & \text{for } i \in I_s \end{cases}$$

Separation

Let $v_i^k = q_i - P_i \bullet X^k$ be the violation of inequality i by the k^{th} iterate (X^k, y^k, z^k, S^k)

(SDP_I)	$\min C \bullet X$	(SDD_I)	$\max b^T y + q_I^T z_I$
	$\text{s.t. } \mathcal{A}(X) = b$		$\text{s.t. } \mathcal{A}^T(y) + \mathcal{P}_I^T(z_I) + S = C$
	$\mathcal{P}_I(X) \geq q_I$		$z_I \geq 0$
	$X \succeq 0$		$S \succeq 0$

- Inequality added if triggered by one of variable or Tapia indicators

$$I_+ = \left\{ i \notin I : v_i^k \geq \min\left\{ \nu^+,\ \nu_v^+ v_i^{k-1} \right\} \right\}$$

(no primal-dual indicator because we have no dual variable z_i yet)

- Inequality removed if detected by all of variable, Tapia, and primal-dual indicators

$$I_- = \left\{ i \in I : v_i^k \leq \min\left\{ \nu^-,\ \nu_v^- v_i^{k-1},\ \nu_z^- z_i^k \right\} \right\}$$

Integrated Algorithm

Start from the initial relaxation without any cuts $\mathcal{P}(X) \geq q$

$$\begin{aligned} \min\ & C \bullet X \\ \text{s.t. } & \mathcal{A}X = b,\ X \succeq 0 \\ & \mathcal{P}_I X - r = q_I,\ r - \xi = 0,\ \xi \geq 0 \end{aligned}$$

$$\begin{aligned} \max\ & b^T y + q_I^T z \\ \text{s.t. } & S = C - \mathcal{A}^T y - \mathcal{P}_I^T z \succeq 0 \\ & z - \psi = 0,\ \psi \geq 0 \end{aligned}$$

1. Initialize: (X^0, y^0, S^0), $\mu^0 = (X^0 \bullet S^0)/N$, $v^0 = q - \mathcal{P}(X^0)$, $I^0 = \emptyset$, $k = 0$.
2. Stop if $\max\{\mu^k, \|b - \mathcal{A}(X^k)\|, \|C - S^k - \mathcal{A}^T(y^k)\|\} < \varepsilon$, $v^k \leq 0$.
3. Update $I^{k+1} = I^k \cup I_+^k \setminus I_-^k$ (using indicators).
4. Warm start by slacking new variables r_i^k and $z_i^k = 0$
5. Compute new iterate $(X^{k+1}, y^{k+1}, z^{k+1}, r^{k+1}, S^{k+1}, \xi^{k+1}, \psi^{k+1})$.
6. Set $k = k + 1$, $\mu^k = (X^k \bullet S^k)/n$, $v^k = q - \mathcal{P}(X^k)$; go to Step 2.

Practical Implementation and Testing

SDP Solver: SDPT3 Version 3.02 (Toh, Todd & Tütüncü (1999, 2003))

- options.maxit = 2: stops for cut separation every other iteration
- additional routines for handling of free variables (A. & Burer 2008))
- routines for primal-dual warmstarting (Engau, A. & Vannelli 2009))

Each problem is solved in three different ways:

- NOCUT solves only the initial SDP relaxation without any additional cuts
- INTCUT applies our algorithm adding up to 200 cuts every other iteration
- COPYCUT solves only the final relaxation with active cuts at optimality

Results for SRFLP Instances

	NOCUT		INTCUT			COPYCUT		
Problem	time	(iter)	cuts	time	(iter)	time	(iter)	γ
BeHa82_4	0.5	(9)	60	0.1	(8)	0.1	(9)	0.87
LoWo76_5	0.1	(10)	317	1.6	(12)	3.6	(11)	0.45
HeKu91_5	0.1	(9)	332	1.9	(13)	3.4	(10)	0.57
HeKu91_6	0.1	(11)	517	4.2	(15)	8.7	(10)	0.49
HeKu91_7	0.2	(11)	801	3.3	(16)	5.1	(11)	0.65
HeKu91_8	0.3	(11)	1356	15.4	(23)	20.6	(13)	0.75
Si69_8	0.3	(12)	1231	10.2	(19)	16.5	(13)	0.62
Si69_9	0.4	(11)	2068	51.0	(31)	74.0	(17)	0.69
Si69_10	0.6	(14)	1702	27.0	(25)	31.1	(11)	0.87
Si69_11	0.9	(14)	2207	59.6	(30)	70.4	(14)	0.85
HeKu91_12	1.2	(12)	3175	179.5	(46)	244.8	(20)	0.73
HeKu91_15	5.1	(15)	5503	957.8	(56)	1269.7	(24)	0.75
Am06_15	4.7	(14)	5940	1583.5	(83)	1967.5	(30)	0.80
Am08_17	11.5	(15)	6225	1634.9	(87)	1919.8	(26)	0.85
Am08_18	14.7	(14)	5606	734.4	(85)	776.3	(27)	0.95
HeKu91_20	28.6	(15)	8094	2478.6	(112)	3050.0	(38)	0.81
AnVa08_25	54.9	(15)	8199	3708.5	(96)	4313.8	(43)	0.86

What About Polynomiality?

The computational results show that the integrated algorithm benefits from the use of indicators and interior-point warmstarts.

The common idea in this algorithm and in others mentioned earlier are:

- to try to predict and add relevant inequalities before they are violated, and
- to resume the algorithm from the current iterate.

To the best of our knowledge, the literature contains no supporting theoretical analysis or proof of convergence and worst-case complexity for such methods.

What Can Reasonably Be Expected

To obtain such an analysis, Engau and A. (2011) propose a conceptual primal-dual IPM for linear optimization with equality and inequality constraints that adds relevant inequalities “on the fly”.

- We do not expect the conceptual algorithm to be an improvement in practice.
- Without assumptions on the added inequalities, we do not expect better worst-case complexity than a standard method because extra work is inevitable to choose and properly integrate selected inequalities.
- Moreover, in the worst case, all the inequalities are necessary.

Main objective: Obtain insights into the conditions under which an algorithm of this kind is polynomial or may be exponential.

Primal-Dual Form With Equalities and Inequalities

We consider the following non-standard primal-dual form:

$$\begin{aligned} \min\ & c^T x \\ \text{s.t. } & Ax = b \\ & Px \geq q \\ & x \geq 0, \end{aligned} \qquad \begin{aligned} \max\ & b^T y + q^T z \\ \text{s.t. } & A^T y + P^T z + s = c \\ & z \geq 0 \\ & s \geq 0 \end{aligned}$$

with n primal variables, m primal equality constraints, ℓ primal inequalities $Px \geq q$, and the following assumptions:

Assumption 1: There exists an optimal solution (x^*, y^*, z^*, s^*).

Assumption 2: There exists a feasible solution (x, y, z, s) that satisfies $(x, s) > 0$, $z = 0$, $\|Xs - \mu e\| \leq (1/4)\mu$ for $\mu = x^T s/n$, and $Px > q$, or equivalently, $Px - q \geq (1/\tau)\mu$ for some $\tau > 0$.

Assumption 3: There exists a suff. large $M < \infty$ such that the primal residual $r = Px - q$ at any feasible point is bounded above by M.

General Worst-Case Complexity Result

Theorem (Engau and A. (2011))

Let (x, y, s) be a strictly feasible point that satisfies $x^T s \leq (1/\epsilon)^\kappa$ with $\kappa > 0$ and Assumption 2 with $\tau > 0$. If the problem satisfies Assumptions 1 and 3, then the new algorithm finds an ϵ-optimal solution in

$$O(((\kappa + \tau + 1)/\epsilon) l (n + l)^{3/2} e^{\theta/11})$$

iterations, where $\theta = O(l/\sqrt{n + l})$ and l is the number of inequalities that are added to the problem.

In particular, if $l = O(\sqrt{n})$ or $l \leq \ell = O(\sqrt{n})$,
then $\theta = O(1)$ and the bound is polynomial:

$$O(((\kappa + \tau + 1)/\epsilon) l (n + l)^{3/2}).$$

Additional Remarks

1. A more detailed analysis shows that the algorithm remains polynomial even for an arbitrarily large number l, as long as the centering and feasiblilty-restoring steps applied after adding an inequality do not run into another inequality too often.
2. We are not able to affirm polynomial time complexity if large numbers of inequalities must be added very close to optimality. This confirms the well-known observation in practice that if the barrier parameter becomes very small then every step can also be very small and IPMs tend to jam.

A New Hierarchy of Relaxations

A New Hierarchy of Relaxations

This is ongoing joint work with E. Adams, F. Rendl, and A. Wiegele.
Recall that we can express all the feasible solutions for max-cut in the form $X = vv^T, v \in \{\pm 1\}^n$.
The convex hull of these 2^{n-1} points is called the cut polytope, denoted by CUT_n.

We can take any subset $I \subseteq \{1, 2, \dots, n\}$ with $|I| = k$ and consider X_I, the principal submatrix of X indexed by I.

Key Observation

If $X \in CUT_n$ then $X_I \in CUT_k$.

This can be expressed as

$$X_I = \sum_j \lambda_j \bar{v}_j \bar{v}_j^T, \quad \lambda \geq 0, \quad \sum_j \lambda_j = 1,$$

where $\bar{v}_j \in \{\pm 1\}^k$ runs through the 2^{k-1} cuts in CUT_k.

A New Hierarchy of Relaxations

This idea leads to a new hierarchy of relaxations for max-cut indexed by k:

$$
\begin{aligned}
z_k = \quad & \max \quad Q \bullet X \\
& \text{s.t.} \quad \text{diag}(X) = e \\
& \qquad X \succeq 0 \\
& \qquad \text{triangle inequalities on } X \\
& \qquad X_I \in CUT_k \text{ for all } k \text{ with } |I| = k.
\end{aligned}
$$

- For k fixed the relaxation is solvable in polynomial time.
- As k approaches n, we get better and better bounds, and if $k = n$ we get the exact solution.
- There is no improvement for $k \leq 4$ because the triangle inequalities give an exact description of the cut polytope.

Illustrative Example (Laurent 2004)

$$Q = -\frac{1}{2}\begin{pmatrix} 0 & 14 & 13 & 14 & 12 \\ 14 & 0 & 13 & 15 & 17 \\ 13 & 13 & 0 & 13 & 11 \\ 14 & 15 & 13 & 0 & 14 \\ 12 & 17 & 11 & 14 & 0 \end{pmatrix}$$

Relaxation	Bound
Basic SDP	38.263
Basic SDP plus triangles	36.143
Basic SDP plus triangles and $k = 5$ constraints	34.000
Optimal value	34.000

Observations

- This idea is similar to the motivation for target cuts in Buchheim, Liers and Oswald (2008), and the lifting and separation of Bonato et al. (2011)
- However, this approach uses an inner description of CUT_k
- This is different from cutting-plane approaches that add constraints valid for CUT_n
- This can be interpreted as a variant of column generation (the new variables are the λ_j)
- For each I we add $2k - 1$ nonnegative variables and $\binom{k}{2}$ new equations
- Adding the constraints for all I at once is computationally inefficient, so the challenge is to identify good choices for I

Generality of this Approach

This approach works for graph optimization problems with the property that restriction to node induced subgraphs results in a similar optimization problem.

It is therefore applicable to problems such as

- max-cut;
- max-stable-set, max-clique;
- graph coloring;
- ordering problems (e.g. SRFLP).

It does not work for

- assignment problems;
- traveling salesman problems;
- max-k-cut, equicut.

More results coming soon...

Time to wrap up...

The (Second To) Last Slide

Main challenges in this area

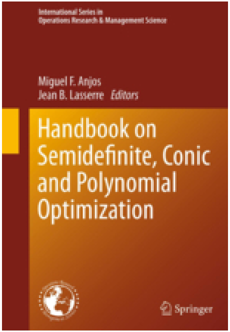
1. Solve (very) large SDP relaxations
2. Exploit structure in the conic relaxations

The presentation today was on one recent contribution per challenge.

- The match of semidefinite/conic optimization with discrete optimization spawned an exciting and active research area.
- The use of conic optimization is expanding to more and more applications and will surely remain fruitful for years to come.

For Plenty More...

Handbook of Semidefinite, Conic and Polynomial Optimization: Theory, Algorithms, Software and Applications



For papers, references, questions, you are welcome to contact me:

miguel-f.anjos@polymtl.ca