Part III
A new convex quadratic programming relaxation

Vienna, January 2012

A New Approach

Recall the non-convex quadratically-constrained program:

$$\begin{array}{rll} \max & \sum_{i \in V} x_i & \\ \text{s.t.} & x_i^2 = x_i & (i \in V) \\ & x_i x_j = 0 & (\{i,j\} \in E). \end{array}$$

$\forall \lambda \in \mathbb{R}^{|V|}$, $\forall \mu \in \mathbb{R}^{|E|}$ the quadratic inequality

$$\sum_{i \in V} \lambda_i (x_i^2 - x_i) + \sum_{\{i,j\} \in E} \mu_{ij} x_i x_j \leq 0$$

is satisfied by the incidence vectors of all stable sets in G

Ellipsoids

rewrite the inequality in a compact form:

$$x^T(\text{Diag}(\lambda) + M(\mu))x \leq \lambda^T x$$

$M(\mu)$ symmetric matrix with $\mu_{ij}/2$ in the entry ij for $\{i,j\} \in E$, and zeroes elsewhere.

Theorem

Let $\lambda \in \mathbb{R}^{|V|}$ and $\mu \in \mathbb{R}^{|E|}$. If the matrix $\text{Diag}(\lambda) + M(\mu)$ *is psd, the set*

$$E(\lambda, \mu) = \left\{x \in \mathbb{R}^n : \; x^T(\text{Diag}(\lambda) + M(\mu))x \leq \lambda^T x\right\}$$

defines an ellipsoid that contains STAB(G)

Description of $TH(G)$ by ellipsoids

this infinite family of ellipsoids indeed describes $TH(G)$:

Lemma (Fujie & Tamura, 2002)

For any graph G, we have:

$$TH(G) = \bigcap_{\lambda,\mu:\text{Diag}(\lambda)+M(\mu)\succeq 0} E(\lambda, \mu)$$

this does not give a straightforward algorithm, but there is one special ellipsoid showing a very nice property

A nice ellipsoid

Let's look back at the Lovász theta relaxation:

$$\begin{aligned} \theta(G) = \max \quad & tr(X) \\ \lambda] \quad & Y_{0i} = X_{ii} \quad (i \in V) & (1) \\ \mu] \quad & X_{ij} = 0 \quad (\{i,j\} \in E) & (2) \\ & Y \in \mathcal{S}^{+}_{|V|+1} \end{aligned}$$

Theorem (GLRS, 2011)

Let $\lambda^* \in \mathbb{R}^{|V|}$, $\mu^* \in \mathbb{R}^{|E|}$ be the optimal dual vectors. Then

$$\theta(G) = \max \left\{ \sum_{i \in V} x_i : x \in E(-\lambda^*, -\mu^*) \right\}.$$

Semidefinite and Lagrangian relaxations

Relax the quadratic constraints by multipliers $p \in \mathbb{R}^n$ and $q \in \mathbb{R}^{|E|}$:

$$f(x,p,q) = \sum_{i \in V} x_i + x^T(\mathrm{Diag}(p) + M(q))x - p \cdot x$$

Lemma (A well known equivalence result)

$$\theta(G) = \min_{p \in \mathbb{R}^n,\, q \in \mathbb{R}^{|E|}} \max_{x \in \mathbb{R}^n} f(x,p,q).$$

Also, $(p^* = -\lambda^*, q^* = -\mu^*)$*. Moreover,* $f(x, -\lambda^*, -\mu^*)$ *is concave (therefore* $-\mathrm{Diag}(\lambda^*) - M(\mu^*)$ *is psd).*

Sketch of the proof

by the previous lemma we have

$$\theta(G) = \max\left\{\sum_{i\in V} x_i - x^T(\mathsf{Diag}(\lambda^*) + M(\mu^*))x + \lambda^*\cdot x :\ x\in\mathbb{R}^n\right\}$$

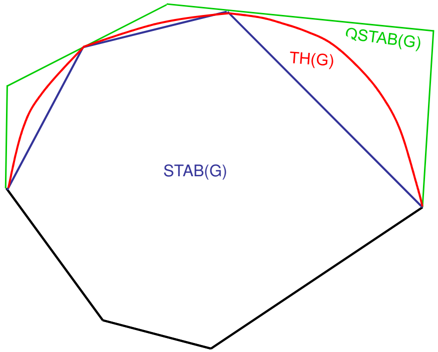
This is a concave maximisation problem and $(-\lambda^*, -\mu^*)$ is an optimal pair of multipliers. Therefore,

$$\theta(G) = \max\left\{\sum_{i\in V} x_i : -x^T(\mathsf{Diag}(\lambda^*) + M(\mu^*))x \le -\lambda^*\cdot x,\ x\in\mathbb{R}^n\right\}$$

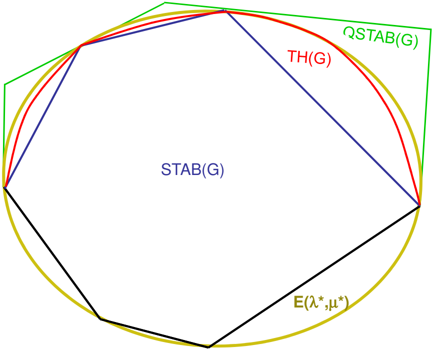
$$= \max\left\{\sum_{i\in V} x_i : x\in E(-\lambda^*, -\mu^*)\right\}.$$

□

The 'optimal' ellipsoid

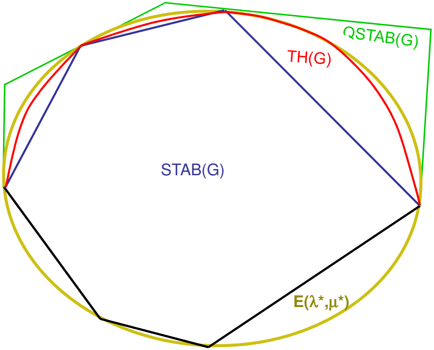


The 'optimal' ellipsoid



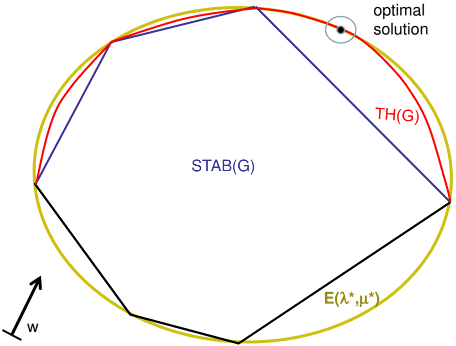
$$TH(G) \subseteq E(-\lambda^*, -\mu^*) \not\subseteq \mathrm{QSTAB}(G)$$
$$\mathrm{QSTAB}(G) \not\subseteq E(-\lambda^*, -\mu^*)$$

The 'optimal' ellipsoid



the incidence vectors of stable sets lie on the boundary

The 'optimal' ellipsoid



$$\theta(G) = \max\{\sum_{i \in V} w_i x_i : x \in TH(G)\} =$$
$$= \max\{\sum_{i \in V} w_i x_i : x \in E(-\lambda^*, -\mu^*)\}$$

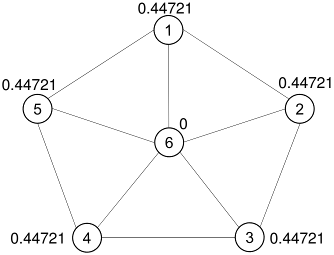
Example

5-wheel: optimal ellipsoid

$$0.3727\sum_{i=1}^{5} x_i^2 + 0.3884x_6^2 + 0.4606\sum_{ij\in E(C)} x_i x_j + 0.3404\sum_{j\in C} x_6 x_j +$$

$$-0.3727\sum_{i=1}^{5} x_i - 0.3884x_6 \leq 0$$

optimal solution value
$= 2.2362 = \sqrt{5}$



Branch-and-cut I: QCP

How to use the 'optimal' ellipsoid $E(-\lambda^*, -\mu^*)$ algorithmically?

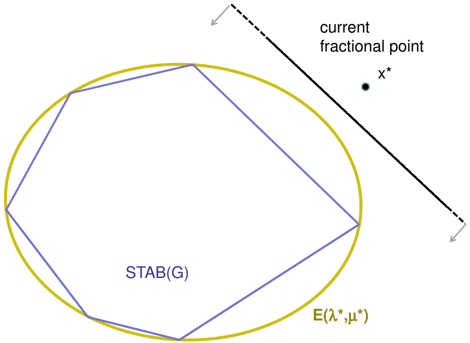
First attempt: branch-and-cut in which each subproblem has a *single convex quadratic constraint*:

$$
\begin{array}{lcr}
\max & \sum_{i \in V} x_i & \\
\text{s.t.} & \sum_{i \in C} x_i \leq 1 & (\forall C \in \mathcal{C}) \\
& x \in E(-\lambda^*, -\mu^*) \cap \mathbb{R}^n_+ &
\end{array}
$$

where $\mathcal{C}$ is a collection of maximal cliques in G.

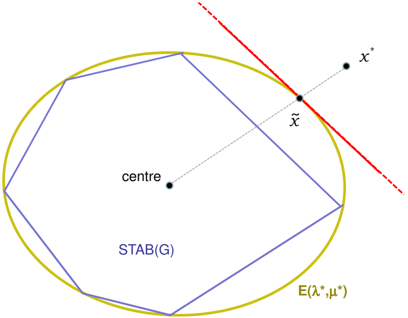
performed badly: solving subproblems too time-consuming

Separating hyperplane



any hyperplane separating x^* from $E(-\lambda^*, -\mu^*)$ defines a valid inequality for $\mathrm{STAB(G)}$

Separation algorithm



1. Perform a line-search to find a point $\tilde{x}$, convex combination of x^* and the centre which lies on the boundary of $E(-\lambda^*, -\mu^*)$.
2. Generate a linear inequality $a^T x \leq b$ that defines a tangent hyperplane to $E(-\lambda^*, -\mu^*)$ at $\tilde{x}$

Branch-and-cut II (LP): clique + ellipsoid cuts

Preprocessing: compute the optimal ellipsoid $E(-\lambda^*, -\mu^*)$
Initial formulation: collection of clique inequalities

Cut generation:

1. Generate an inequality $a^T x \leq b$ that defines a tangent hyperplane to $E(-\lambda^*, -\mu^*)$ in $\tilde{x}$.
2. Scale the vector a and scalar b by a constant factor Δ and round them down
3. **Strengthen** the cut (and obtain multiple cuts)
4. Add the cut and reoptimize
5. Run heuristic for separation of clique inequalities

Cut strengthening

- for any variable x_j in a given set S:
  - compute the "best" rhs $\bar{b}$ for $x_j = 0$:

    $$\gamma_j^0 = \max\{\sum_{i \neq j} a_i x_i : \mathbf{x} \in \mathbf{E}(-\lambda^*, -\mu^*), x_j = 0\}$$

    $$\bar{b} := \min\{b, \lfloor \gamma_j^0 \rfloor\}$$

  - compute the "best" coefficient $\bar{a}_j$ by lifting x_j :

    $$\gamma_j^1 = \max\{\sum_{i \neq j} a_i x_i : \mathbf{x} \in \mathbf{E}(-\lambda^*, -\mu^*), x_j = 1\}$$

    $$\bar{a}_j = \bar{b} - \lfloor \gamma_j^1 \rfloor$$

- Different sets S and different sequences yield different cuts

Example (continued)

- Solve the LP with all cliques and get the otimal fractional point $x^* = (0.5, \dots, 0.5, 0)$
- compute the tangent hyperlane:

$$0.372900182 \sum_{i=1}^{5} x_i + 0.372833116 x_6 \leq 0.833912948$$

- scale and round down the cut

$$3729 \sum_{i=1}^{5} x_i + 3728 x_6 \leq 8339$$

- strengthen the cut ($S = \{x_6\}$)

$$3729 \sum_{i=1}^{5} x_i + 8339 x_6 \leq 8339 \qquad \approx \sum_{i=1} x_i + \sqrt{5} x_6 \leq \sqrt{5}$$

- add the cut and solve LP: optimal value $= \sqrt{5} = \theta(G)$

Experiments

Graphs DIMACS challenge benchmark graphs
(http://www.dimacs.rutgers.edu/pub/challenge/graph)

Computer 2 Intel Xeon 5150 processors clocked at 2.66 GHz with 4GB of RAM

MIP/QCP solver CPLEX 11.2

SDP solver Malick, Povh, Rendl and Wiegle '07
(http://www.math.uni-klu.ac.at/or/Software)

Upper bounds

Graph name	n	m	$\alpha(G)$	$\theta(G)$	$\mathrm{UB}_{\mathrm{ellips}}$
brock200_1	200	5,066	21	27.50	27.79
brock200_2	200	10,024	12	14.22	14.32
brock200_3	200	7,852	15	18.82	19.00
brock200_4	200	6,811	17	21.29	21.52
C.125.9	125	787	34	37.89	38.05
C.250.9	250	3,141	44	56.24	57.41
c-fat200-5	200	11,427	58	60.34	60.36
DSJC125.1	125	736	34	38.39	38.44
DSJC125.5	125	3,891	10	11.47	11.48
mann_a27	378	702	126	132.76	132.88
keller4	171	5,100	11	14.01	14.09
p_hat300-1	300	33,917	8	10.10	10.15
p_hat300-2	300	22,922	25	27.00	27.14
p_hat300-3	300	11,460	36	41.16	41.66
san200_0.7-2	200	5,970	18	18.00	18.10
sanr200_07	200	6,032	18	23.80	24.00
sanr200_09	200	2,037	42	49.30	49.77

Cut-and-branch results

	Clique cut-and-branch		Ellipsoid cut-and-branch	
Graph	#sub.	time	#sub.	time
brock200_1	270,169	1,784.78	122,387	**1,432.76**
brock200_2	6,102	83.16	2,689	209.39
brock200_3	52,173	459.02	7,282	**252.17**
brock200_4	85,134	735.52	18,798	**468.52**
C.125.9	3,049	5.63	2,514	6.52
C.250.9	—	—	—	—
c-fat200-5	47	12.06	47	25.16
DSJC125.1	4,743	6.13	2,981	8.09
DSJC125.5	1,138	6.97	369	9.66
mann a27	4,552	2.06	1,278	2.46
keller4	4,856	21.88	3,274	36.55
p hat300-1	4,518	124.36	4,238	132.54
p hat300-2	7,150	194.27	1,522	204.88
p hat300-3	398,516	10,270.53	107,240	**8,756.76**
san200 0.7-2	86	8.58	0	0.76
sanr200_07	88,931	827.52	42,080	**733.47**
sanr200_09	635,496	2,650.55	274,261	**1,281.52**

savings: 54% subproblems on average, $35-50\%$ CPU time in hard instances

Final remarks and open issues

- A preliminary experience shows that approximate dual multipliers, computed at early iterations of the SDP solver, still give strong (and valid) cuts
- current research:
  - extend the ellipsoid to stronger relaxations
  - extend the ellipsoid to CO other problems
- Ellipsoid cuts are denser than lift-and-proj cuts and require more careful cut management
- Overall lesson: cutting planes obtained by methods which disdain the structure of $\mathrm{STAB}(G)$ do help in practice