Cutting-Planes for the Max-Cut Problem

Laura Galli [1] Konstantinos Kaparis Adam N. Letchford [2]

University of Warwick, July 2011

[1] DEIS, University of Bologna
[2] Lancaster University

Outline

- Literature review
  - Max-Cut problem
  - ILP and IQP formulations
  - Product-type inequalities
- Separation algorithms
  - Triangle inequalities
  - Odd-clique and rounded-psd inequalities
  - Gap inequalities
- Some further ‘tricks’
  - Strengthening of gap inequalities
  - Stabilisation
  - A ‘special’ triangle packing
- Computational results
- Conclusions

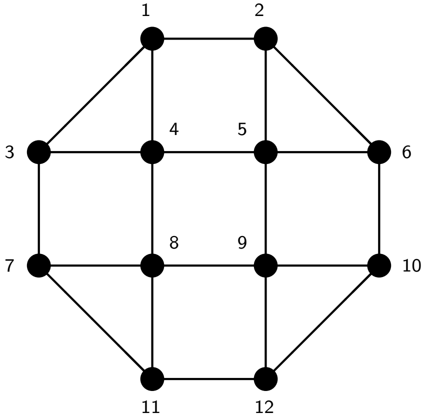
The Max-Cut problem

The Max-cut problem is a well-known, fundamental and strongly $\mathcal{NP}$-hard combinatorial optimisation problem.

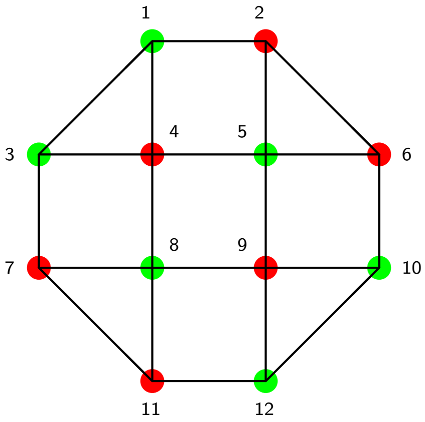
Definition

Let $G = (V, E)$ be an edge-weighted undirected graph. Max–cut calls for the vertex set to be partitioned into two subsets $(S, V \setminus S)$, in such a way that the sum of the weights on the edges crossing from S to its complement $V \setminus S$ is maximised (i.e., looks for the cut $\delta(S)$ of maximal weight).

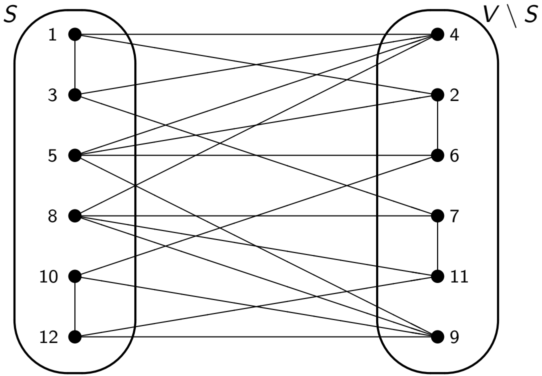
Example



Example (cont.)



Example (cont.)



Some well known facts

- Surprisingly many applications (physics, VLSI, scheduling)
- Strongly $\mathcal{NP}$-hard (Karp *et al.* 1972, Johnson & Stockmeyer 1975)
- Polynomial solvable cases are known (e.g., planar graphs)
- 0.878 approximation-algorithm by Goemans & Williamson 1995
- W.l.o.g. one can always assume the graph is complete

A 0-1 Linear Programming formulation

$$
\begin{array}{rll}
\max & \sum_{1 \le i < j \le n} w_{ij} x_{ij} & \\
\text{s.t.} & x_{ij} + x_{ik} + x_{jk} \le 2 & (1 \le i < j < k \le n) \\
& x_{ij} - x_{ik} - x_{jk} \le 0 & (1 \le i < j \le n;\, k \ne i, j) \\
& x_{ij} \in \{0, 1\} & (1 \le i < j \le n)
\end{array}
$$

The convex hull in $\mathbb{R}^{\binom{n}{2}}$ of solutions is called the cut polytope and often denoted by CUT_n.

Product-type Inequalities

Remark

It turns out that many of the known valid inequalities are of product type:

$$\sum_{1 \leq i < j \leq n} b_i b_j x_{ij} \leq r$$

for some reals $b_1, \ldots, b_n$ and some real r.

Triangle inequalities

$$
\begin{array}{rll}
\max & \sum_{1\le i<j\le n} w_{ij}x_{ij} & \\
\text{s.t.} & x_{ij}+x_{ik}+x_{jk}\le 2 \quad (1\le i<j<k\le n) & (1)\\
& x_{ij}-x_{ik}-x_{jk}\le 0 \quad (1\le i<j\le n;\, k\ne i,j) & (2)\\
& x_{ij}\in\{0,1\} \quad (1\le i<j\le n) &
\end{array}
$$

- Inequalities (1)-(2) are called triangle inequalities

Triangle inequalities

$$\begin{array}{lll} \max & \sum_{1\le i<j\le n} w_{ij}x_{ij} & \\ \text{s.t.} & x_{ij}+x_{ik}+x_{jk}\le 2 \quad (1\le i<j<k\le n) & (1) \\ & x_{ij}-x_{ik}-x_{jk}\le 0 \quad (1\le i<j\le n;\, k\ne i,j) & (2) \\ & x_{ij}\in\{0,1\} \quad (1\le i<j\le n) & \end{array}$$

- Inequalities (1)-(2) are called triangle inequalities
- Triangle inequalities induce facets of CUT_n for all n

Triangle inequalities

$$
\begin{aligned}
\max \quad & \textstyle\sum_{1\le i<j\le n} w_{ij}x_{ij} & & \\
\text{s.t.} \quad & x_{ij}+x_{ik}+x_{jk} \le 2 & (1 \le i < j < k \le n) & \quad (1)\\
& x_{ij}-x_{ik}-x_{jk} \le 0 & (1 \le i < j \le n;\, k \ne i,j) & \quad (2)\\
& x_{ij} \in \{0,1\} & (1 \le i < j \le n) &
\end{aligned}
$$

- Inequalities (1)-(2) are called triangle inequalities
- Triangle inequalities induce facets of CUT_n for all n
- The polytope defined by triangle inequalities is called metric polytope and denoted by MET_n

Triangle inequalities

$$
\begin{aligned}
\max \quad & \textstyle\sum_{1\le i<j\le n} w_{ij}x_{ij} & & \\
\text{s.t.} \quad & x_{ij}+x_{ik}+x_{jk}\le 2 & (1\le i<j<k\le n) & \quad (1)\\
& x_{ij}-x_{ik}-x_{jk}\le 0 & (1\le i<j\le n;\, k\ne i,j) & \quad (2)\\
& x_{ij}\in\{0,1\} & (1\le i<j\le n) &
\end{aligned}
$$

- Inequalities (1)-(2) are called triangle inequalities
- Triangle inequalities induce facets of CUT_n for all n
- The polytope defined by triangle inequalities is called metric polytope and denoted by MET_n
- Triangle inequalities provide a complete description for $n \le 4$
- Triangle inequalities are of product-type

An Integer Quadratic Programming formulation

Let $y_i = \begin{cases} 1 & \text{if } i \in S \\ -1 & \text{if } i \in V \setminus S \end{cases}$

So, y_i takes the value 1 if vertex i is on one shore of the cut, and -1 if it is on the other. Then the max-cut problem can be formulated as the following integer quadratic program:

$$\begin{array}{lll} \max & \frac{1}{2} \sum_{1 \le i < j \le n} w_{ij}(1 - y_i y_j) & \\ \text{s.t.} & y_i \in \{-1, +1\} & (i \in V) \end{array}$$

SDP relaxation

If we define an $n \times n$ matrix Y in which, for all i, j, the entry Y_{ij} represents the product $y_i y_j$, then the max-cut problem can be formulated as a semidefinite program:

$$
\begin{array}{lcr}
\max & \frac{1}{2}\sum_{1 \le i < j \le n} w_{ij}(1 - Y_{ij}) & \\
\text{s.t.} & Y_{ii} = 1 & (i \in V) \\
 & rank(Y) = 1 & \\
 & Y \in \mathcal{S}_+^n &
\end{array}
$$

The SDP relaxation is derived by dropping the $rank(Y) = 1$ constraint.

Positive Semidefinite (psd) inequalities

The feasible region of the SDP relaxation called the elliptope is convex, but not polyhedral.

Positive Semidefinite (psd) inequalities

The feasible region of the SDP relaxation called the elliptope is convex, but not polyhedral.

Laurent & Poljak 1995 showed that it can be projected onto the space of the x_{ij} variables via the mapping $Y_{ij} = 1 - 2x_{ij}$.

Positive Semidefinite (psd) inequalities

The feasible region of the SDP relaxation called the elliptope is convex, but not polyhedral.

Laurent & Poljak 1995 showed that it can be projected onto the space of the x_{ij} variables via the mapping $Y_{ij} = 1 - 2x_{ij}$.

The projection of the elliptope is defined by the following positive semidefinite (psd) inequalities:

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \sigma(b)^2/4 \qquad (\forall b \in \mathbb{R}^n), \tag{3}$$

where $\sigma(b)$ denotes $\sum_{i=1}^n b_i$.

Positive Semidefinite (psd) inequalities

The feasible region of the SDP relaxation called the elliptope is convex, but not polyhedral.

Laurent & Poljak 1995 showed that it can be projected onto the space of the x_{ij} variables via the mapping $Y_{ij} = 1 - 2x_{ij}$.

The projection of the elliptope is defined by the following positive semidefinite (psd) inequalities:

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \sigma(b)^2/4 \qquad (\forall b \in \mathbb{R}^n), \tag{3}$$

where $\sigma(b)$ denotes $\sum_{i=1}^n b_i$.

Psd-inequalities are of product-type

Rounded-psd and Odd-Clique inequalities

If we choose integer values for $b_1, \ldots, b_n$, clearly, the left-hand side will then be an integer in any feasible integer solution.

Rounded-psd and Odd-Clique inequalities

If we choose integer values for $b_1, \ldots, b_n$, clearly, the left-hand side will then be an integer in any feasible integer solution.

Thus, in this case, we can round down the right hand side of the inequality without cutting off any integer solution.

Rounded-psd and Odd-Clique inequalities

If we choose integer values for $b_1, \ldots, b_n$, clearly, the left-hand side will then be an integer in any feasible integer solution.

Thus, in this case, we can round down the right hand side of the inequality without cutting off any integer solution.

(In fact, we can assume that $\sigma(b)$ is odd without losing any important inequalities.) This leads to the following rounded-psd inequalities:

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \lfloor \sigma(b)^2/4 \rfloor \qquad (\forall b \in \mathbb{Z}^n : \sigma(b) \text{ odd}). \tag{4}$$

Rounded-psd and Odd-Clique inequalities

If we choose integer values for $b_1, \dots, b_n$, clearly, the left-hand side will then be an integer in any feasible integer solution.

Thus, in this case, we can round down the right hand side of the inequality without cutting off any integer solution.

(In fact, we can assume that $\sigma(b)$ is odd without losing any important inequalities.) This leads to the following rounded-psd inequalities:

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \lfloor \sigma(b)^2/4 \rfloor \qquad (\forall b \in \mathbb{Z}^n : \sigma(b) \text{ odd}). \tag{4}$$

In the special case where $b \in \{-1, 0, +1\}^n$, the rounded-psd inequalities reduce to the odd-clique inequalities (Barahona & Mahjoub) .

Gap Inequalities

The rounded psd inequalities are formed by taking a psd inequality and reducing the right hand side.

Taking this idea further, we can reduce the right hand side until the constraint becomes supporting.

This leads to the gap inequalities of Laurent & Poljak 1996, which take the form:

$$\sum_{1 \leq i < j \leq n} b_i b_j x_{ij} \leq \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n), \tag{5}$$

where $\gamma(b) := \min \left\{ |z^T b| : z \in \{\pm 1\}^n \right\}$ is the so-called gap of b.

Gap Inequalities: good news

Gap inequalities are remarkably general...

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2\right)/4 \qquad (\forall b \in \mathbb{Z}^n)$$

gap

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \lfloor \sigma(b)^2 / 4 \rfloor \text{ and } \sigma(b) \text{ odd } (\forall b \in \mathbb{Z}^n)$$

gap $\longrightarrow$ rounded psd

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \sigma(b)^2 / 4 \qquad (\forall b \in \mathbb{R}^n)$$

gap $\longrightarrow$ rounded psd $\longrightarrow$ psd

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \sigma(b)^2 / 4 \text{ and } \gamma(b) = 0 \quad (\forall b \in \mathbb{Z}^n)$$

gap $\longrightarrow$ rounded psd $\longrightarrow$ psd $\longrightarrow$ gap-0

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) /4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le 0 \text{ and } \sigma(b) = 0 \quad (\forall b \in \mathbb{Z}^n)$$

gap $\longrightarrow$ rounded psd $\longrightarrow$ psd $\longrightarrow$ gap-0 $\longrightarrow$ negative-type

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2\right)/4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2\right)/4 \text{ and } \gamma(b) = 1 \quad (\forall b \in \mathbb{Z}^n)$$

$$\begin{tikzcd}
\text{gap} \arrow[r] & \text{rounded psd} \arrow[r] \arrow[d] & \text{psd} \arrow[r] & \text{gap-0} \arrow[r] & \text{negative-type} \\
& \text{gap-1} & & &
\end{tikzcd}$$

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \text{ and } \gamma(b) = 1 \quad (\forall b \in \mathbb{Z}^n)$$

$$\begin{tikzcd}
\text{gap} \arrow[r] & \text{rounded psd} \arrow[r] \arrow[d] & \text{psd} \arrow[r] & \text{gap-0} \arrow[r] & \text{negative-type} \\
& \text{gap-1} \arrow[urr] & & &
\end{tikzcd}$$

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le 0 \text{ and } \sigma(b) = 1 \quad (\forall b \in \mathbb{Z}^n)$$

$$\begin{tikzcd}
\text{gap} \arrow[r] & \text{rounded psd} \arrow[r] \arrow[d] & \text{psd} \arrow[r] & \text{gap-0} \arrow[r] & \text{negative-type} \\
& \text{gap-1} \arrow[urr] \arrow[rr] & & \text{hypermetric} &
\end{tikzcd}$$

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le 0 \text{ and } \sigma(b) = 1 \quad (\forall b \in \mathbb{Z}^n)$$

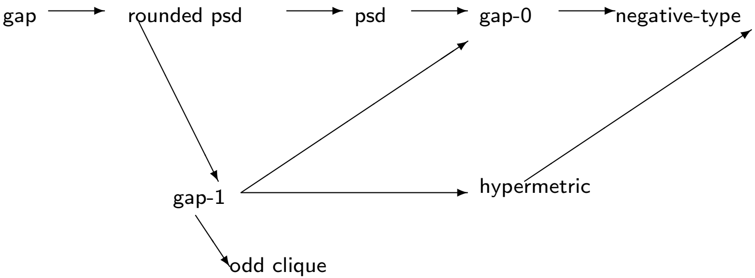
$$\begin{tikzcd}
\text{gap} \arrow[r] & \text{rounded psd} \arrow[r] \arrow[d] & \text{psd} \arrow[r] & \text{gap-0} \arrow[r] & \text{negative-type} \\
 & \text{gap-1} \arrow[urr] \arrow[rr] & & \text{hypermetric} \arrow[ur] &
\end{tikzcd}$$

Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \lfloor \sigma(b)^2 / 4 \rfloor \text{ and } \sigma(b) \text{ odd } (\forall b \in \{0, \pm 1\}^n)$$



Gap Inequalities: good news

Gap inequalities are remarkably general...

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \left(\sigma(b)^2 - \gamma(b)^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$$

$$\sum_{1 \le i < j \le n} b_i b_j x_{ij} \le \lfloor \sigma(b)^2 / 4 \rfloor \text{ and } \sigma(b) \text{ odd } (\forall b \in \{0, \pm 1\}^n)$$

$$\begin{tikzcd}
\text{gap} \arrow[r] & \text{rounded psd} \arrow[r] \arrow[d] & \text{psd} \arrow[r] & \text{gap-0} \arrow[r] & \text{negative-type} \\
& \text{gap-1} \arrow[urr] \arrow[rr] \arrow[dr] & & \text{hypermetric} \arrow[ur] & \\
& & \text{odd clique} \arrow[r] & \text{triangle} &
\end{tikzcd}$$

Gap Inequalities: bad news

Gap Inequalities: bad news

- They are infinite in number
- Is not known if they define a polyhedral set
- Computing the gap is $\mathcal{NP}$ - hard (Laurent & Poljak, 1997)

Gap Inequalities: bad news

- They are infinite in number
- Is not known if they define a polyhedral set
- Computing the gap is $\mathcal{NP}$ - hard (Laurent & Poljak, 1997)

Thus they have received little attention so far...

Separation of Triangle Inequalities

Triangle inequalities can be separated in $\mathcal{O}(n^3)$ time by mere enumeration.

The following Algorithm 1 still runs in $\mathcal{O}(n^3)$ time, but has the advantage of generating only $\mathcal{O}(n^2)$ inequalities.

Algorithm 1: Heuristic separation algorithm for Triangle inequalities.

Input: A point $x^* \in \mathbb{R}^{\binom{n}{2}}$ to be separated.
Output: Failure or a violated triangle inequality.

for $1 \leq i < j \leq n$ **do**
  if $x^*_{ij} > (2+\epsilon)/3$ **then**
    Find $k : x^*_{ik} + x^*_{jk} = \max_{h=j+1\dots n}\{x^*_{ih} + x^*_{jh}\}$;
    if $x^*_{ij} + x^*_{ik} + x^*_{jk} \geq 2 + \epsilon$ **then**
      Return violated inequality $x_{ij} + x_{ik} + x_{jk} \leq 2$;
    end
  end
  if $x^*_{ij} > \epsilon$ **then**
    Find $k : x^*_{ik} + x^*_{jk} = \min_{h=1\dots n}\{x^*_{ih} + x^*_{jh}\}$;
    if $x^*_{ij} - x^*_{ik} - x^*_{jk} \geq \epsilon$ **then**
      Return violated inequality $x_{ij} - x_{ik} - x_{jk} \leq 0$;
    end
  end
end

Separation of Odd-Clique Inequalities

Using the mapping, the odd clique inequalities can be written as:

$$b^T Y b \geq 1 \qquad (\forall b \in \{0, \pm 1\}^n : \sigma(b) \text{ odd}).$$

If Y^* (the point to be separated) is psd, then the separation problem reduces to the following Convex Integer Quadratic Programme (CIQP):

$$\min \left\{ b^T Y^* b : \ \sigma(b) = 2k + 1, \ b \in \{0, \pm 1\}^n, \ k \in \mathbb{Z}_+ \right\}.$$

There is good software available for solving CIQPs.

If Y^* is not psd, then one must use a slightly different approach...

Separation of Odd-Clique Inequalities (cont.)

The odd-clique heuristic of Helmberg & Rendl 1998 is a simple greedy:

- The basic idea consists in keeping collection of odd clique (including triangle) inequalities from which new odd clique inequalities can be derived
- For each inequality in the collection, the algorithm tries to extend the clique by inserting two nodes in the hope of obtaining a new violated odd clique inequality, that is added to the current pool.
- The advantage of this heuristic is that the inequalities start out very sparse and get denser as the algorithm progresses.

Separation of Rounded-psd Inequalities

Similarly to the case of odd-clique inequalities, rounded-psd inequalities can be separated exactly via CIQP.

Another approach is to use a greedy heuristic, similar to the one of Helmberg & Rendl 1998 for odd-clique inequalities.

- Take a previously-generated round psd inequality (which could be a triangle or an odd-clique inequality)
- Test whether a violated rounded psd inequality can be obtained by incrementing or decrementing two of the components of the b vector.

Heuristic separation of Gap Inequalities

We present now an $\mathcal{O}(n^2 u)$ (where u is an UB on $||b||_1 = \sum_{i=1}^{n} |b_i|$) heuristic separation scheme for gap inequalities.

It is based on the spectral decomposition of the current solution Y^*.

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$
- Construct the matrix Y^*

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$
- Construct the matrix Y^*
- Let $b^* \in \mathbb{R}^n$ be an eigenvector of Y^* corresponding to the minimum eigenvalue

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$
- Construct the matrix Y^*
- Let $b^* \in \mathbb{R}^n$ be an eigenvector of Y^* corresponding to the minimum eigenvalue
- Compute $u^* = ||b^*||_1$

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$
- Construct the matrix Y^*
- Let $b^* \in \mathbb{R}^n$ be an eigenvector of Y^* corresponding to the minimum eigenvalue
- Compute $u^* = ||b^*||_1$
- Scale b^* by multiplying it by u/u^*

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$
- Construct the matrix Y^*
- Let $b^* \in \mathbb{R}^n$ be an eigenvector of Y^* corresponding to the minimum eigenvalue
- Compute $u^* = ||b^*||_1$
- Scale b^* by multiplying it by u/u^*
- Round the components of b^* to integers

Heuristic separation of Gap Inequalities (cont.)

- Let u be a prespecified upper bound on $||b||_1$
- Construct the matrix Y^*
- Let $b^* \in \mathbb{R}^n$ be an eigenvector of Y^* corresponding to the minimum eigenvalue
- Compute $u^* = ||b^*||_1$
- Scale b^* by multiplying it by u/u^*
- Round the components of b^* to integers
- Compute the gap of b^* and check the correspondng gap inequality for violation.

Heuristic separation of Gap Inequalities (cont.)

Note that

$$\gamma(b) = ||b||_1 - 2\ \mathrm{SSP},$$

where SSP is the solution to the following subset-sum problem:

$$\max\left\{ \sum_{i=1}^{n} |b_i| y_i :\ \sum_{i=1}^{n} |b_i| y_i \leq \left\lfloor \frac{||b||_1}{2} \right\rfloor,\ y \in \{0,1\}^n \right\}.$$

This subset-sum problem can be solved in $\mathcal{O}(n||b||_1)$ time by dynamic programming.

Strengthening of Gap Inequalities

1. Recall that psd inequalities constitute a relaxation of gap inequalities (i.e., $\gamma(b)^2$ is removed)

Strengthening of Gap Inequalities

1. Recall that psd inequalities constitute a relaxation of gap inequalities (i.e., $\gamma(b)^2$ is removed)
2. Obtaining the gap $\gamma(b)$, calls for solving an instance of the *subset sum problem* (SSP) and setting $\gamma(b) = ||b||_1 - 2SSP$

Strengthening of Gap Inequalities

1. Recall that psd inequalities constitute a relaxation of gap inequalities (i.e., $\gamma(b)^2$ is removed)
2. Obtaining the gap $\gamma(b)$, calls for solving an instance of the *subset sum problem* (SSP) and setting $\gamma(b) = ||b||_1 - 2SSP$ ...can we do better than that?

Strengthening of Gap Inequalities

1. Recall that psd inequalities constitute a relaxation of gap inequalities (i.e., $\gamma(b)^2$ is removed)
2. Obtaining the gap $\gamma(b)$, calls for solving an instance of the *subset sum problem* (SSP) and setting $\gamma(b) = ||b||_1 - 2SSP$ ...can we do better than that?
3. We can adjust the vector b^* and try to maximise the violation of the resulting gap inequality (strengthening).

Strengthening of Gap Inequalities (cont.)

1. For $r = 0, \ldots, u$, let

$$f(r) = \max \left\{ \sum_{j \neq i} |b_j| y_j : \sum_{j \neq i} |b_j| y_j \leq r, \ y \in \{0,1\}^n \right\}$$

2. Set $b' = \begin{cases} b_j & \text{if } j \neq i \\ b_i + \Delta & \text{if } j = i \end{cases}$ note that $||b'||_1 = ||b||_1 + |b_i + \Delta_i| - b_i$

   $\text{GAP}' : \sum_{1 \leq i < j \leq n} b_i b_j x_{ij} + \Delta \sum_{j \neq i} b_j x_{ij} \leq \left(\sigma(b')^2 - \gamma(b')^2 \right) / 4 \qquad (\forall b \in \mathbb{Z}^n)$

Strengthening of Gap Inequalities (cont.)

1. For $r = 0, \ldots, u$, let

$$f(r) = \max \left\{ \sum_{j \neq i} |b_j| y_j : \sum_{j \neq i} |b_j| y_j \leq r, \; y \in \{0,1\}^n \right\}$$

2. Set $b' = \begin{cases} b_j & \text{if } j \neq i \\ b_i + \Delta & \text{if } j = i \end{cases}$ note that $||b'||_1 = ||b||_1 + |b_i + \Delta_i| - b_i$

   $\text{GAP}' : \sum_{1 \leq i < j \leq n} b_i b_j x_{ij} + \Delta \sum_{j \neq i} b_j x_{ij} \leq \left(\sigma(b')^2 - \gamma(b')^2 \right) /4 \qquad (\forall b \in \mathbb{Z}^n)$

$$\begin{aligned} SSP' = \max \quad & |b_i + \Delta| y_i + \textstyle\sum_{k \neq i} |b_k| y_k \\ \text{s.t.} \quad & |b_i + \Delta| y_i + \textstyle\sum_{k \neq i} |b_k| y_k \leq \left\lfloor \frac{||b'||_1}{2} \right\rfloor \\ & y \in \{0,1\}^n \end{aligned}$$

   That is $SSP' = \max \left\{ f\left(\left\lfloor \frac{||b'||_1}{2} \right\rfloor \right), \; |b_i + \Delta| + f\left(\left\lfloor \frac{||b'||_1}{2} \right\rfloor - |b_i + \Delta| \right) \right\}$

3. Set $\gamma(b') = ||b'||_1 - 2SSP'$ and export the stronger gap inequality

Primal Stabilisation

The tailing-off phenomenon leads to very long running times for large instances. One way around it is using primal stabilisation as follows:

1. Solve the SDP relaxation, yielding a solution x^*

Primal Stabilisation

The tailing-off phenomenon leads to very long running times for large instances. One way around it is using primal stabilisation as follows:

1. Solve the SDP relaxation, yielding a solution x^*
2. Add to the LP a lower and upper bound for each variable, to force the LP solution to stay near x^*

Primal Stabilisation

The tailing-off phenomenon leads to very long running times for large instances. One way around it is using primal stabilisation as follows:

1. Solve the SDP relaxation, yielding a solution x^*
2. Add to the LP a lower and upper bound for each variable, to force the LP solution to stay near x^*
3. Run a cutting-plane algorithm

Primal Stabilisation

The tailing-off phenomenon leads to very long running times for large instances. One way around it is using primal stabilisation as follows:

1. Solve the SDP relaxation, yielding a solution x^*
2. Add to the LP a lower and upper bound for each variable, to force the LP solution to stay near x^*
3. Run a cutting-plane algorithm
4. If no more psd inequalities are violated (within some tolerance), or if none of the lower and upper bounds are binding, remove the lower and upper bounds

Edge-disjoint Triangle Packing

To improve the initial LP relaxation we generate a collection of special triangle inequalities.
Special means:

1. likely to substantially improve the initial upper bound
2. not likely to slow down the LP solver

Edge-disjoint Triangle Packing

To improve the initial LP relaxation we generate a collection of special triangle inequalities.
Special means:

1. likely to substantially improve the initial upper bound
2. not likely to slow down the LP solver

Property 1 suggests that the more TIs we use, the better. But property 2 suggests that it would be a good idea to ensure that no variable appears in more than one of the chosen TIs.

Edge-disjoint Triangle Packing

To improve the initial LP relaxation we generate a collection of special triangle inequalities.
Special means:

1. likely to substantially improve the initial upper bound
2. not likely to slow down the LP solver

Property 1 suggests that the more TIs we use, the better. But property 2 suggests that it would be a good idea to ensure that no variable appears in more than one of the chosen TIs.

This leads to the idea of finding a maximum cardinality packing of edge-disjoint triangles in the complete graph K_n.

Edge-disjoint Triangle Packing

To improve the initial LP relaxation we generate a collection of special triangle inequalities.
Special means:

1. likely to substantially improve the initial upper bound
2. not likely to slow down the LP solver

Property 1 suggests that the more TIs we use, the better. But property 2 suggests that it would be a good idea to ensure that no variable appears in more than one of the chosen TIs.

This leads to the idea of finding a maximum cardinality packing of edge-disjoint triangles in the complete graph K_n.

For each triangle in the packing, we have four TIs to choose from. So we select the best one according to the signs of the weights of the corresponding edges.

Computational results

We summarise the results obtained from the application of two cutting plane schemes on a subset of Max-Cut instances taken from the Biq Mac Library [3].

[3] http://BiqMac.uni-klu.ac.at

Computational results

We summarise the results obtained from the application of two cutting plane schemes on a subset of Max-Cut instances taken from the Biq Mac Library [3].

- Among these instances, g_05_n are unweighted with edge probability 0.5 and $n = 60,\ 80,\ 100$. Instances pm1d_80.0, have edge probability 0.99, $w_{ij} = \{-1,\ 0,\ 1\}$ for $\{1 \leq i \leq j \leq n\}$ and $n = 80$.

[3] http://BiqMac.uni-klu.ac.at

Computational results

We summarise the results obtained from the application of two cutting plane schemes on a subset of Max-Cut instances taken from the Biq Mac Library [3].

- Among these instances, g_05_n are unweighted with edge probability 0.5 and $n = 60,\ 80,\ 100$. Instances pm1d_80.0, have edge probability 0.99, $w_{ij} = \{-1,\ 0,\ 1\}$ for $\{1 \leq i \leq j \leq n\}$ and $n = 80$.
- All our algorithms were implemented in ANSI C and tested on a PC Intel Xeon, 2.4 GHz, with a 12 GB RAM, using ILOG CPLEX 12.1 as LP solver.

[3] http://BiqMac.uni-klu.ac.at

Computational results

We summarise the results obtained from the application of two cutting plane schemes on a subset of Max-Cut instances taken from the Biq Mac Library [3].

- Among these instances, g_05_n are unweighted with edge probability 0.5 and $n = 60,\ 80,\ 100$. Instances pm1d_80.0, have edge probability 0.99, $w_{ij} = \{-1,\ 0,\ 1\}$ for $\{1 \leq i \leq j \leq n\}$ and $n = 80$.
- All our algorithms were implemented in ANSI C and tested on a PC Intel Xeon, 2.4 GHz, with a 12 GB RAM, using ILOG CPLEX 12.1 as LP solver.
- Computing times are expressed in seconds.

[3] http://BiqMac.uni-klu.ac.at

Computational results

We summarise the results obtained from the application of two cutting plane schemes on a subset of Max-Cut instances taken from the `Biq Mac Library` [3].

- Among these instances, `g_05_n` are unweighted with edge probability 0.5 and $n = 60,\ 80,\ 100$. Instances `pm1d_80.0`, have edge probability 0.99, $w_{ij} = \{-1,\ 0,\ 1\}$ for $\{1 \leq i \leq j \leq n\}$ and $n = 80$.
- All our algorithms were implemented in ANSI C and tested on a PC Intel Xeon, 2.4 GHz, with a 12 GB RAM, using ILOG CPLEX 12.1 as LP solver.
- Computing times are expressed in seconds.
- Table 1 presents integrality gaps for different Max-Cut relaxations.

[3] `http://BiqMac.uni-klu.ac.at`

Computational results

We summarise the results obtained from the application of two cutting plane schemes on a subset of Max-Cut instances taken from the Biq Mac Library [3].

- Among these instances, g_05_n are unweighted with edge probability 0.5 and $n = 60,\ 80,\ 100$. Instances pm1d_80.0, have edge probability 0.99, $w_{ij} = \{-1,\ 0,\ 1\}$ for $\{1 \leq i \leq j \leq n\}$ and $n = 80$.
- All our algorithms were implemented in ANSI C and tested on a PC Intel Xeon, 2.4 GHz, with a 12 GB RAM, using ILOG CPLEX 12.1 as LP solver.
- Computing times are expressed in seconds.
- Table 1 presents integrality gaps for different Max-Cut relaxations.
- Each row is the average of 10 instances of the given class.

[3] http://BiqMac.uni-klu.ac.at

Computational results (cont.)

	% IG TIs	% IG SDP	Mult.GAPs		Mult.GAPs+TIs	
			% IG	Time (scs)	% IG	Time (scs)
g_05_60	10.83	2.52	5.31	60	1.84	275
g_05_80	13.26	2.18	4.81	180	1.59	2685
g_05_100	15.28	2.23	5.01	600	1.93	15383
pmd1_80	102.1	22.3	35.55	180	15.12	3650

Table: Integrality gaps of various relaxations of the max-cut problem.

Conclusions

- Our preliminary computations (table 1) show that our cutting-plane scheme based on the heuristic separation of GAP and triangle inequalities is capable of beating the SDP relaxation on all instances

Conclusions

- Our preliminary computations (table 1) show that our cutting-plane scheme based on the heuristic separation of GAP and triangle inequalities is capable of beating the SDP relaxation on all instances
- Odd clique and rounded psd inequalities generally do not help in terms of bound quality, but can be useful to reduce the number of triangle inequalities in the LP

Conclusions

- Our preliminary computations (table 1) show that our cutting-plane scheme based on the heuristic separation of GAP and triangle inequalities is capable of beating the SDP relaxation on all instances
- Odd clique and rounded psd inequalities generally do not help in terms of bound quality, but can be useful to reduce the number of triangle inequalities in the LP
- Finally, the results presented refer to GAP inequalities without strengthening. In fact, after many experiments, we noticed that the best set-up for our cutting-plane scheme, is separating 'rounds' of GAP inequalities, by generating one inequality for each negative eigenvalue in the Y matrix.

Ta!