Lifting and Separation Procedures for the Cut Polytope

T. Bonato, M. Jünger, G. Reinelt, G. Rinaldi

Saarbrücken, February 21, 2014

Outline

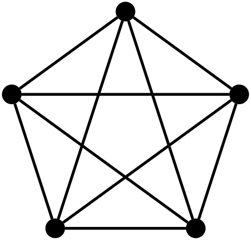
1. Max-Cut Problem
2. Contraction-based Separation
3. Computational Results

Outline

Max-Cut Problem

Definition

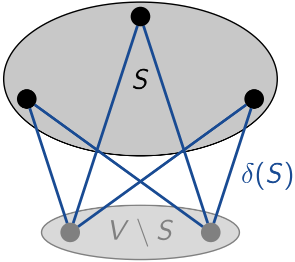
Let $G = (V, E)$ be an undirected weighted graph.



Definition

Let $G = (V, E)$ be an undirected weighted graph.

Any $S \subseteq V$ induces a set $\delta(S)$ of edges with exactly one end node in S. The set $\delta(S)$ is called a cut of G with shores S and $V \setminus S$.

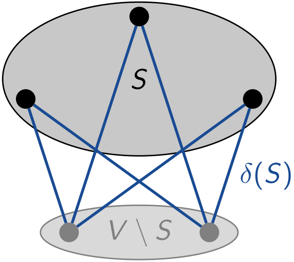


Definition

Let $G = (V, E)$ be an undirected weighted graph.

Any $S \subseteq V$ induces a set $\delta(S)$ of edges with exactly one end node in S. The set $\delta(S)$ is called a cut of G with shores S and $V \setminus S$.

Finding a cut with maximum aggregate edge weight is known as max-cut problem.



Complexity

- NP-hard for:
  - general graphs with arbitrary edge weights,
  - almost planar graphs.
- Polynomial for e. g.:
  - graphs with exclusively negative edge weights,
  - planar graphs,
  - graphs not contractible to K_5.

Complexity

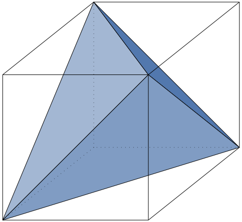
- NP-hard for:
  - general graphs with arbitrary edge weights,
  - almost planar graphs.
- Polynomial for e. g.:
  - graphs with exclusively negative edge weights,
  - planar graphs,
  - graphs not contractible to K_5.

Applications

- Unconstrained quadratic $+/-1$- resp. $0/1$-optimization.
- Computation of ground states of Ising spin glasses.
- Via minimization in VLSI circuit design.

Cut polytope CUT(G)

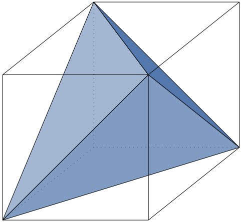
Convex hull of all incidence vectors of cuts of G.



CUT(K_3)

Cut polytope CUT(G)

Convex hull of all incidence vectors of cuts of G.



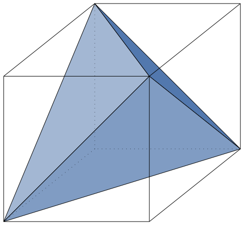
Semimetric polytope MET(G)

Relaxation of the max-cut IP formulation described by two inequality classes:

Odd-cycle:	$x(F) - x(C \setminus F) \leq \lvert F \rvert - 1,$	for each cycle C of G, for all $F \subseteq C, \lvert F \rvert$ odd.
Trivial:	$0 \leq x_e \leq 1,$	for all $e \in E$.

Cut polytope CUT(G)

Convex hull of all incidence vectors of cuts of G.



CUT(K_3)

Semimetric polytope MET(G)

Relaxation of the max-cut IP formulation described by two inequality classes:

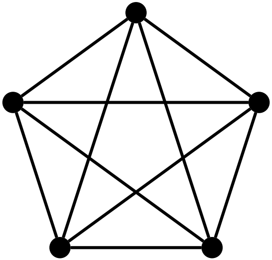
Odd-cycle:	$x(F) - x(C \setminus F) \leq \lvert F \rvert - 1,$	for each cycle C of G, for all $F \subseteq C, \lvert F \rvert$ odd.
Trivial:	$0 \leq x_e \leq 1,$	for all $e \in E$.

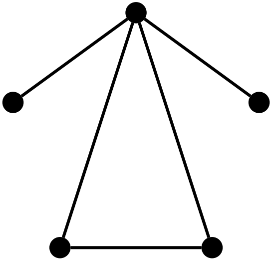
CUT(G) and MET(G) have exactly the same integral points.

Algorithms

- **Branch & Cut**,
- Branch & Bound using SDP relaxations.

Certain separation procedures only work for dense/complete graphs.





Algorithms

- **Branch & Cut**,
- Branch & Bound using SDP relaxations.

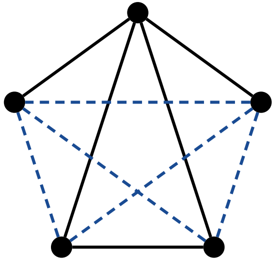
Certain separation procedures only work for dense/complete graphs.

How to handle sparse graphs?

Algorithms

- **Branch & Cut**,
- Branch & Bound using SDP relaxations.

Certain separation procedures only work for dense/complete graphs.



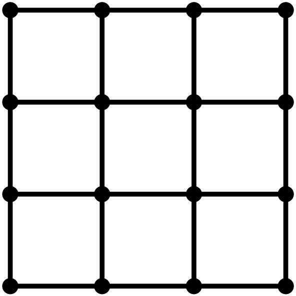
How to handle sparse graphs

- **Trivial approach**:
  artificial completion using edges with weight 0.
- **Drawback**:
  increases number of variables and thus the computational difficulty.

Outline

An Example

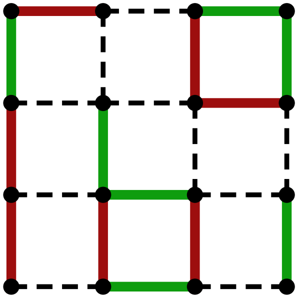
Quadratic (4×4)-grid with 16 nodes and 24 edges.



An Example

Quadratic (4×4)-grid with 16 nodes and 24 edges.

W.r.t. LP solution $z \in \text{MET}(G) \setminus \text{CUT}(G)$, the edge set decomposes into:

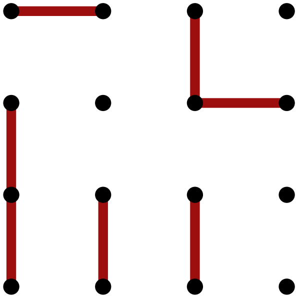


An Example

Quadratic (4×4)-grid with 16 nodes and 24 edges.

W.r.t. LP solution $z \in \text{MET}(G) \setminus \text{CUT}(G)$, the edge set decomposes into:

- 0-edges,

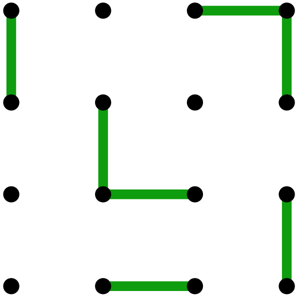


An Example

Quadratic (4×4)-grid with 16 nodes and 24 edges.

W.r.t. LP solution $z \in \text{MET}(G) \setminus \text{CUT}(G)$, the edge set decomposes into:

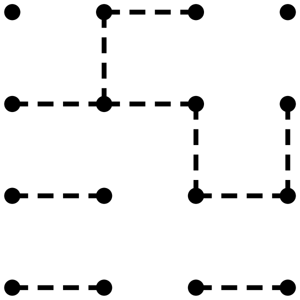
- 0-edges,
- 1-edges,



Quadratic (4×4)-grid with 16 nodes and 24 edges.

W.r.t. LP solution $z \in \mathrm{MET}(G) \setminus \mathrm{CUT}(G)$, the edge set decomposes into:

- 0-edges,
- 1-edges,
- fractional edges.

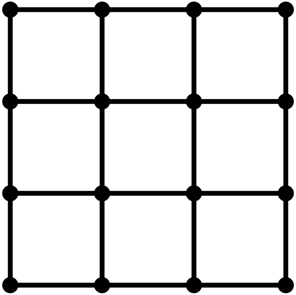


An Example

Quadratic (4×4)-grid with 16 nodes and 24 edges.

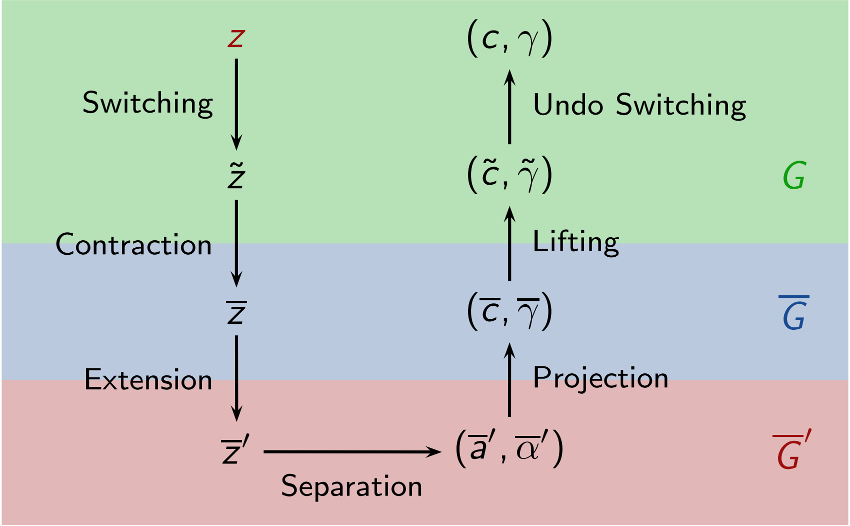
W.r.t. LP solution $z \in \text{MET}(G) \setminus \text{CUT}(G)$, the edge set decomposes into:

- 0-edges,
- 1-edges,
- fractional edges.

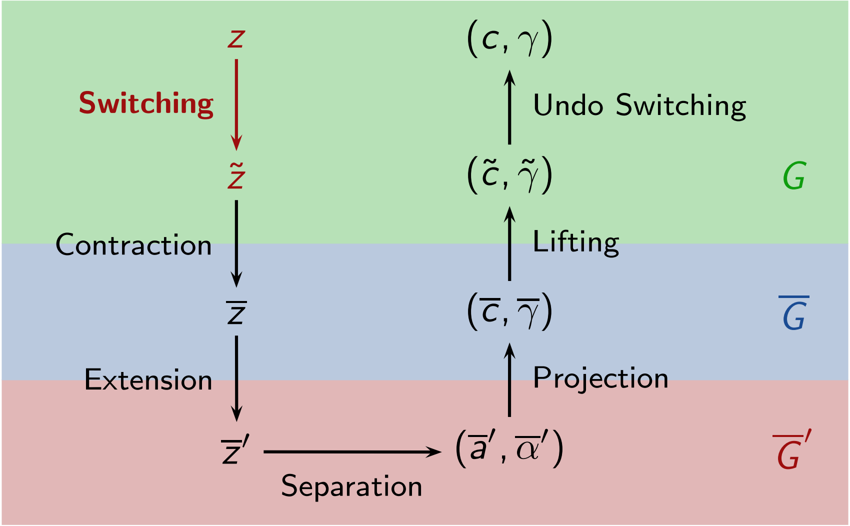


Artificial completion would require 96 additional edges!

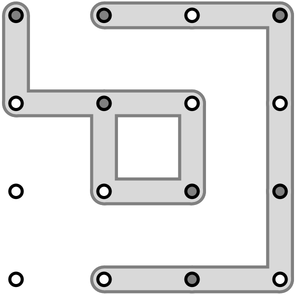
Input: LP solution $z \in \text{MET}(G) \setminus \text{CUT}(G)$.



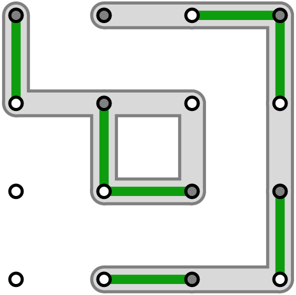
Transform 1-edges into 0-edges.



$z \in \text{MET}(G)$ implies existence of a cut $\boxed{\delta(S)}$ that contains

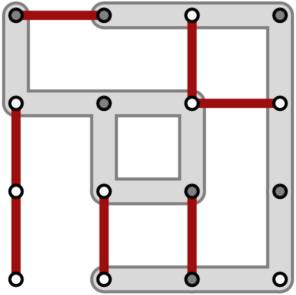


$z \in \mathrm{MET}(G)$ implies existence of a cut $\boxed{\delta(S)}$ that contains all 1-edges

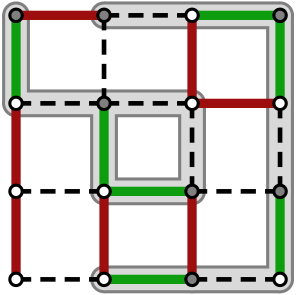


Switching

$z \in \mathrm{MET}(G)$ implies existence of a cut $\delta(S)$ that contains all 1-edges but no 0-edges.



$z \in \mathrm{MET}(G)$ implies existence of a cut $\boxed{\delta(S)}$ that contains all 1-edges but no 0-edges.



Switching z alongside the cut $\delta(S)$:

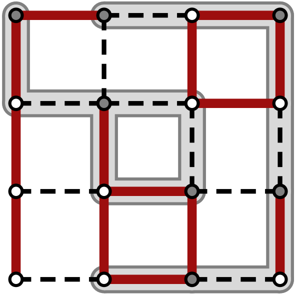
- only affects edges of $\delta(S)$,

Switching

$z \in \mathrm{MET}(G)$ implies existence of a cut $\boxed{\delta(S)}$ that contains all 1-edges but no 0-edges.

Switching z alongside the cut $\delta(S)$:

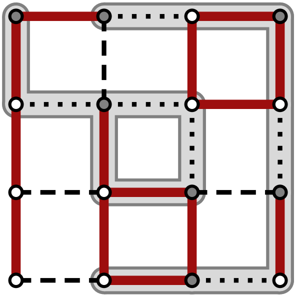
- only affects edges of $\delta(S)$,
- transforms all 1-edges into 0-edges,



$z \in \text{MET}(G)$ implies existence of a cut $\delta(S)$ that contains all 1-edges but no 0-edges.

Switching z alongside the cut $\delta(S)$:

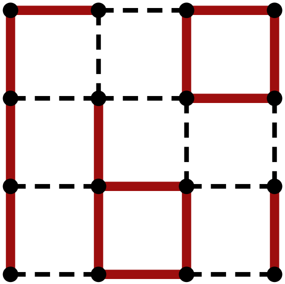
- only affects edges of $\delta(S)$,
- transforms all 1-edges into 0-edges,
- may alter values of fractional edges.



$z \in \mathrm{MET}(G)$ implies existence of a cut $\boxed{\delta(S)}$ that contains all 1-edges but no 0-edges.

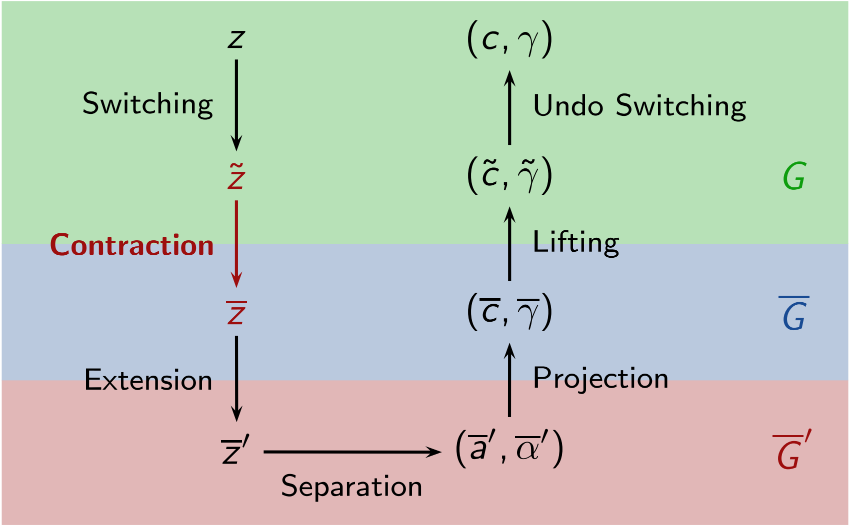
Switching z alongside the cut $\delta(S)$:

- only affects edges of $\delta(S)$,
- transforms all 1-edges into 0-edges,
- may alter values of fractional edges.



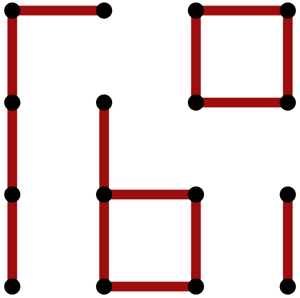
Values in the switched LP solution $\tilde{z}$ are either fractional or zero.

Contract 0-edges.



Contraction

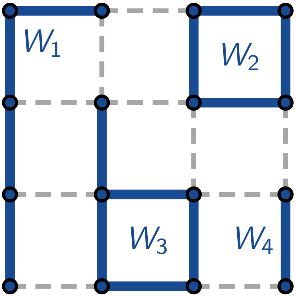
Let G_0 be the graph induced by the 0-edges of the switched LP solution $\tilde{z}$.



Contraction

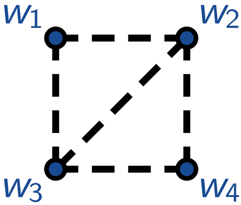
Let G_0 be the graph induced by the 0-edges of the switched LP solution $\tilde{z}$.

1. Determine connected components of G_0.



Let G_0 be the graph induced by the 0-edges of the switched LP solution $\tilde{z}$.

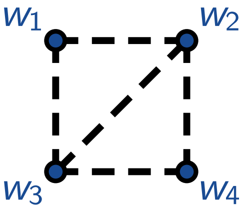
1. Determine connected components of G_0.
2. Contract each component to a supernode.



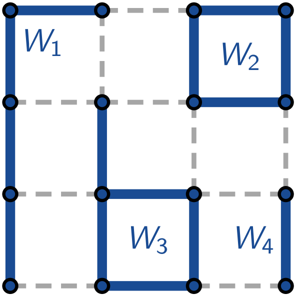
Contraction

Let G_0 be the graph induced by the 0-edges of the switched LP solution $\tilde{z}$.

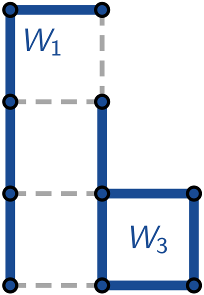
1. Determine connected components of G_0.
2. Contract each component to a supernode.



- Contracted LP solution $\overline{z}$ has only fractional values.
- Associated contracted graph $\overline{G}$ may not be complete.

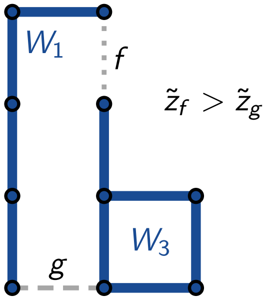

W_1
W_2
W_3
W_4

Contraction as Heuristic Odd-Cycle Separator



Contraction as Heuristic Odd-Cycle Separator

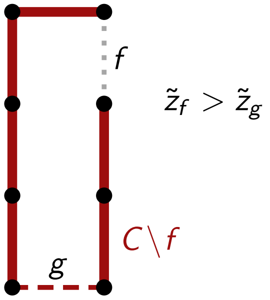
Assume two components of G_0 are joined by two fractional edges f, g with different switched LP values. W.l.o.g. let $\tilde{z}_f > \tilde{z}_g$.



Assume two components of G_0 are joined by two fractional edges f, g with different switched LP values. W.l.o.g. let $\tilde{z}_f > \tilde{z}_g$.

Then the switched LP solution $\tilde{z}$ violates the odd-cycle inequality

$$x_f - x(C \setminus f) \leq 0.$$

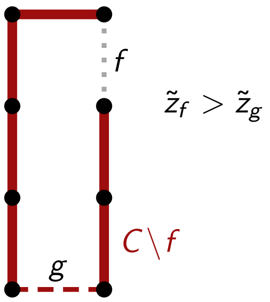


Contraction as Heuristic Odd-Cycle Separator

Assume two components of G_0 are joined by two fractional edges f, g with different switched LP values. W.l.o.g. let $\tilde{z}_f > \tilde{z}_g$.

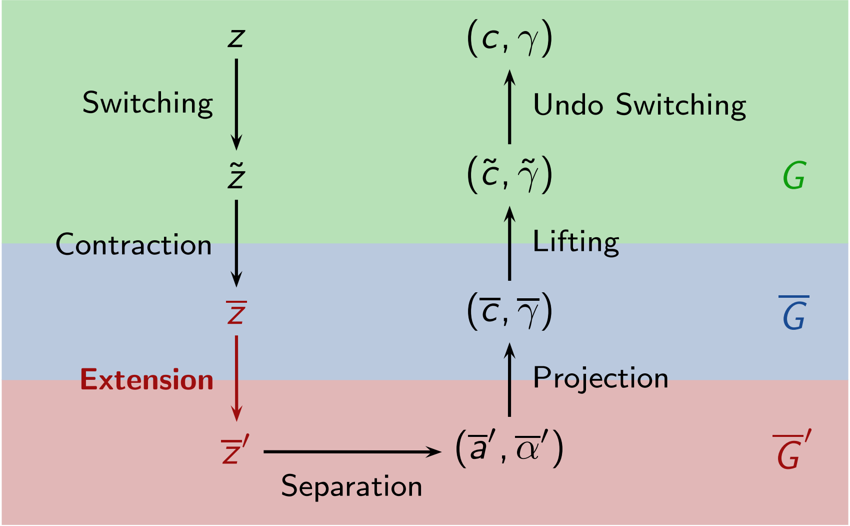
Then the switched LP solution $\tilde{z}$ violates the odd-cycle inequality

$$x_f - x(C \setminus f) \leq 0.$$



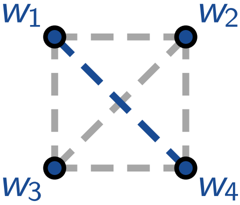
Contraction allows heuristic odd-cycle separation.

Introduce artificial LP values for non-edges.



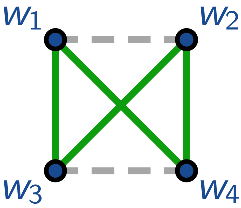
Extension

Given a contracted LP solution $\overline{z} \in \mathrm{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \mathrm{MET}(\overline{G}')$.



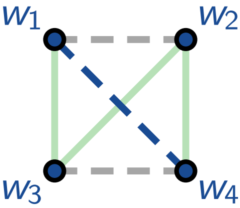
Given a contracted LP solution $\overline{z} \in \mathrm{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \mathrm{MET}(\overline{G}')$.

New cycles in the extended graph



Given a contracted LP solution $\overline{z} \in \text{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \text{MET}(\overline{G}')$.

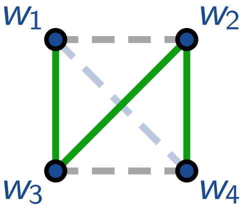
New cycles in the extended graph consist of a former non-edge



Extension

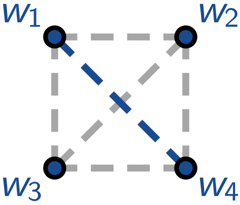
Given a contracted LP solution $\overline{z} \in \mathrm{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \mathrm{MET}(\overline{G}')$.

New cycles in the extended graph consist of a former non-edge and a connecting path.



Given a contracted LP solution $\overline{z} \in \mathrm{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \mathrm{MET}(\overline{G}')$.

New cycles in the extended graph consist of a former non-edge and a connecting path.

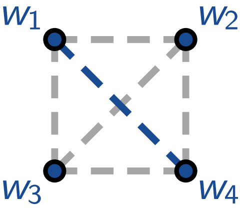


Feasible artificial LP values of non-edge ht

Range: $[\max\{0, L_{ht}\}, \min\{U_{ht}, 1\}] \subseteq [0, 1]$ with

$$L_{ht} := \max\{\ \overline{z}(F) - \overline{z}(P \setminus F) - |F| + 1 \mid P\ (h,t)\text{-path},\ F \subseteq P,\ |F| \text{ odd}\},$$
$$U_{ht} := \min\{-\overline{z}(F) + \overline{z}(P \setminus F) + |F| \mid P\ (h,t)\text{-path},\ F \subseteq P,\ |F| \text{ even}\}.$$

Given a contracted LP solution $\overline{z} \in \mathrm{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \mathrm{MET}(\overline{G}')$.



New cycles in the extended graph consist of a former non-edge and a connecting path.

Feasible artificial LP values of non-edge ht

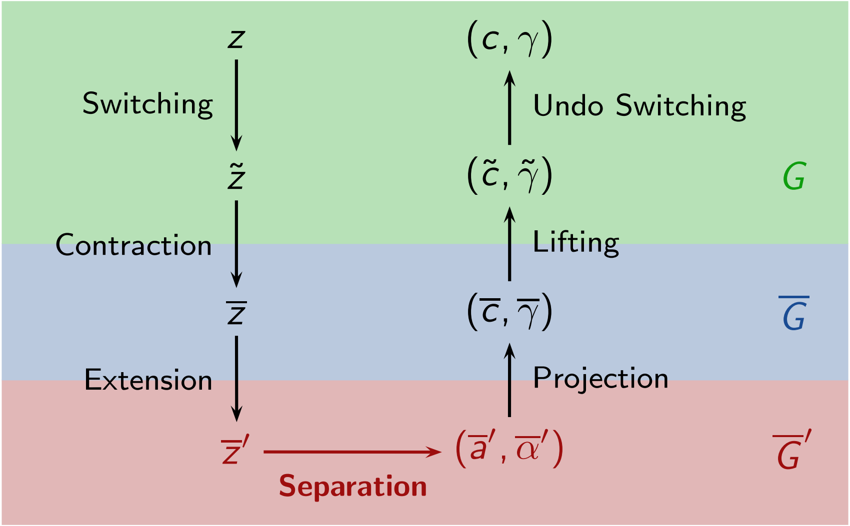
Range: $[\max\{0, L_{ht}\}, \min\{U_{ht}, 1\}] \subseteq [0, 1]$ with

$$L_{ht} := \max\{\ \overline{z}(F) - \overline{z}(P \setminus F) - |F| + 1 \mid P\ (h,t)\text{-path},\ F \subseteq P,\ |F| \text{ odd}\},$$
$$U_{ht} := \min\{-\overline{z}(F) + \overline{z}(P \setminus F) + |F| \mid P\ (h,t)\text{-path},\ F \subseteq P,\ |F| \text{ even}\}.$$

Odd-cycle / trivial inequality derived from arg max (resp. arg min) is called a lower (resp. upper) inequality of ht.

Outline of the Contraction-based Separation

Separate extended LP solution.



Target Cuts vs. Standard Separation

Target cuts were introduced by Buchheim, Liers, and Oswald (2008).

Given:

- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$.

Target Cuts vs. Standard Separation

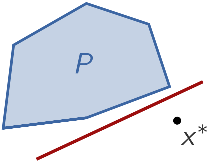
Target cuts were introduced by Buchheim, Liers, and Oswald (2008).

Given:

- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$.

Standard separation

Find a valid inequality $a^T x \leq \alpha$
that separates x^* from P.



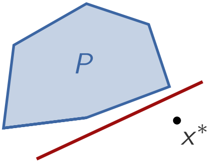
Target cuts were introduced by Buchheim, Liers, and Oswald (2008).

Given:

- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$.

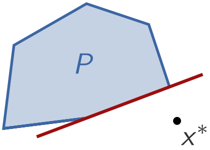
Standard separation

Find a valid inequality $a^T x \leq \alpha$ that separates x^* from P.



Target cut separation

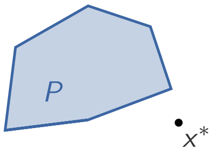
Find a facet defining inequality $a^T x \leq \alpha$ that separates x^* from P.



Target Cuts in a Nutshell

Given:

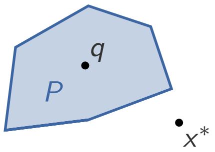
- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$,



Target Cuts in a Nutshell

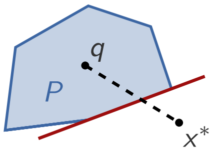
Given:

- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$,
- Point $q \in \text{relint}(P)$.



Given:

- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$,
- Point $q \in \text{relint}(P)$.

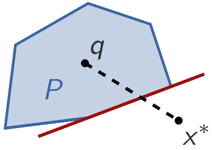


Goal

Find the inequality $a^T x \leq \alpha$ which defines the facet of P that is intersected by the line $\overline{qx^*}$.

Given:

- Polytope $P := \text{conv}\{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$,
- Point $x^* \notin P$,
- Point $q \in \text{relint}(P)$.



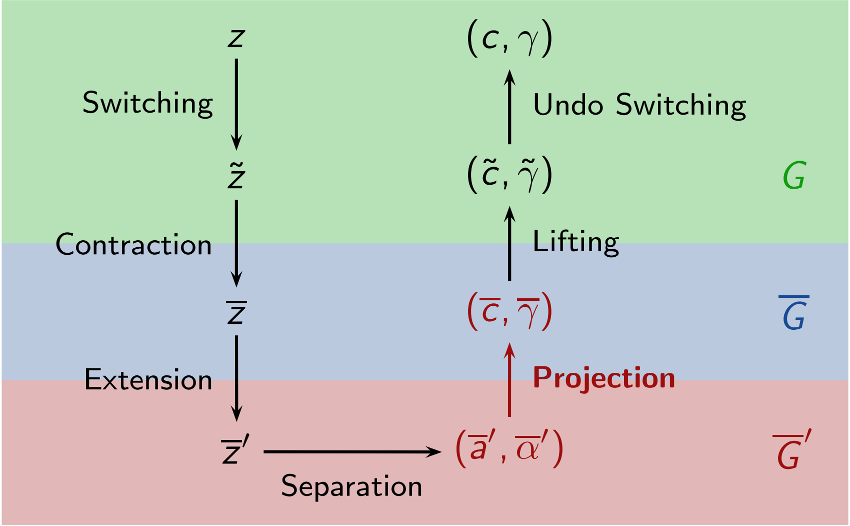
Goal

Find the inequality $a^T x \leq \alpha$ which defines the facet of P that is intersected by the line $\overline{qx^*}$.

Target Cut LP

$$\begin{aligned} \max \;\; & a^T(x^* - q) \\ \text{s.t.} \;\; & a^T(x_i - q) \leq 1, \;\; \text{for } i = 1, \ldots, n \\ & a \in \mathbb{R}^d. \end{aligned}$$

Project out nonzero coefficients related to non-edges.



Projection

Consider a valid inequality $\overline{a}'^T \overline{x}' \leq \overline{\alpha}'$ violated by the extended LP solution $\overline{z}'$.

Non-edges may have nonzero coefficients!

$$(\cdots \quad \overline{\mathbf{a}}'_{\mathbf{uv}} \; \cdots \quad \overline{\mathbf{a}}'_{\mathbf{st}} \; \cdots, \overline{\boldsymbol{\alpha}}')$$

Projection

Consider a valid inequality $\overline{a}'^T \overline{x}' \leq \overline{\alpha}'$ violated by the extended LP solution $\overline{z}'$.

Non-edges may have nonzero coefficients!

Project out coefficient of non-edge uv

Add a lower inequality if $\overline{a}'_{uv} > 0$ resp. an upper inequality if $\overline{a}'_{uv} < 0$.

$$
\begin{array}{rcccccl}
(\cdots & \overline{\mathbf{a}}'_{\mathbf{uv}} & \cdots & \overline{\mathbf{a}}'_{\mathbf{st}} & \cdots, & \overline{\boldsymbol{\alpha}}') \\
+\,(\cdots & -\overline{\mathbf{a}}'_{\mathbf{uv}} & \cdots & \cdots & \cdots, & \overline{\boldsymbol{\beta}}'_{\mathbf{1}}) \\
+\,(\cdots & \cdots & \cdots & -\overline{\mathbf{a}}'_{\mathbf{st}} & \cdots, & \overline{\boldsymbol{\beta}}'_{\mathbf{2}})
\end{array}
$$

Projection

Consider a valid inequality $\overline{a}'^T \overline{x}' \leq \overline{\alpha}'$ violated by the extended LP solution $\overline{z}'$.

Non-edges may have nonzero coefficients!

Project out coefficient of non-edge uv

Add a lower inequality if $\overline{a}'_{uv} > 0$ resp. an upper inequality if $\overline{a}'_{uv} < 0$.

$$
\begin{array}{rcccccl}
 & (\cdots & \overline{\mathbf{a}}'_{\mathbf{uv}} & \cdots & \overline{\mathbf{a}}'_{\mathbf{st}} & \cdots, & \overline{\boldsymbol{\alpha}}') \\
+ & (\cdots & -\overline{\mathbf{a}}'_{\mathbf{uv}} & \cdots & \cdots & \cdots, & \overline{\boldsymbol{\beta}}'_{\mathbf{1}}) \\
+ & (\cdots & \cdots & \cdots & -\overline{\mathbf{a}}'_{\mathbf{st}} & \cdots, & \overline{\boldsymbol{\beta}}'_{\mathbf{2}}) \\
\hline
= & (\cdots & \mathbf{0} & \cdots & \mathbf{0} & \cdots, & \overline{\gamma})
\end{array}
$$

In the projected inequality, all non-edge coefficients are 0 and can be truncated.

Projection

Consider a valid inequality $\overline{a}'^T \overline{x}' \leq \overline{\alpha}'$ violated by the extended LP solution $\overline{z}'$.
Non-edges may have nonzero coefficients!

Project out coefficient of non-edge uv

Add a lower inequality if $\overline{a}'_{uv} > 0$ resp. an upper inequality if $\overline{a}'_{uv} < 0$.

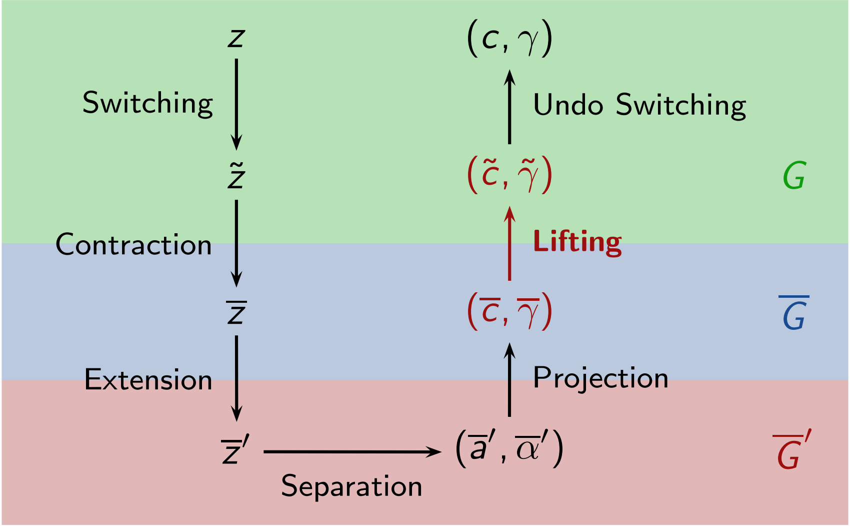
$$
\begin{array}{rlllll}
 & (\cdots \quad \overline{\mathbf{a}}'_{\mathbf{uv}} & \cdots & \overline{\mathbf{a}}'_{\mathbf{st}} & \cdots, & \overline{\alpha}') \\
+ & (\cdots \; -\overline{\mathbf{a}}'_{\mathbf{uv}} & \cdots & \cdots & \cdots, & \overline{\beta}'_1) \\
+ & (\cdots \quad \cdots & \cdots & -\overline{\mathbf{a}}'_{\mathbf{st}} & \cdots, & \overline{\beta}'_2) \\
\hline
= & (\cdots \quad \mathbf{0} & \cdots & \mathbf{0} & \cdots, & \overline{\gamma})
\end{array}
$$

In the projected inequality, all non-edge coefficients are 0 and can be truncated.

Drawback

- If the added inequalities are not tight at $\overline{z}'$ then the projection reduces the initial violation.
- Question of facet preservation partially answered by Avis et al. (2008), but conditions seldom satisfied by presented approach.

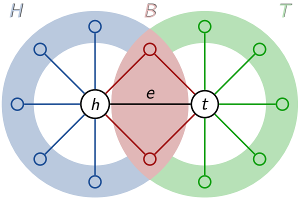
Lift inequality.



Required information

When contracting edge $e = ht$, store partition (H, T, B) of the adjacent nodes:

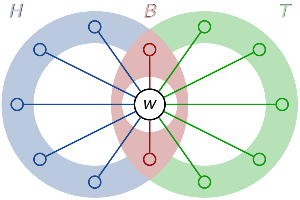
- $H = \{$ exclusive neighbors of $h\}$,
- $T = \{$ exclusive neighbors of $t\}$,
- $B = \{$ common neighbors of h and $t\}$.



Required information

When contracting edge $e = ht$, store partition (H, T, B) of the adjacent nodes:

- $H = \{$ exclusive neighbors of $h \}$,
- $T = \{$ exclusive neighbors of $t \}$,
- $B = \{$ common neighbors of h and $t \}$.



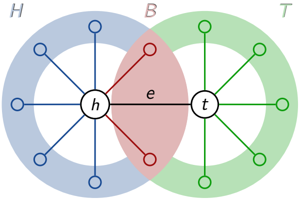
Lift inequality

- Assign coefficients of edges of the shrunk graph

Required information

When contracting edge $e = ht$, store partition (H, T, B) of the adjacent nodes:

- $H = \{$ exclusive neighbors of $h\}$,
- $T = \{$ exclusive neighbors of $t\}$,
- $B = \{$ common neighbors of h and $t\}$.



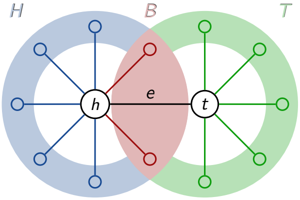
Lift inequality

- Assign coefficients of edges of the shrunk graph to edges of the original graph w.r.t. (H, T, B).

Required information

When contracting edge $e = ht$, store partition (H, T, B) of the adjacent nodes:

- $H = \{$ exclusive neighbors of $h \}$,
- $T = \{$ exclusive neighbors of $t \}$,
- $B = \{$ common neighbors of h and $t \}$.



Lift inequality

- Assign coefficients of edges of the shrunk graph to edges of the original graph w.r.t. (H, T, B).
- Edge e gets coefficient $-\min\left\{ \sum_{v \in T} |\overline{c}_{wv}|, \sum_{v \in H} |\overline{c}_{wv}| \right\}$.
  W.l.o.g. we assume that the edges $(w : T)$ yield the arg min.

Theorem

An inequality $(\hat{c}, \gamma)$ *obtained from a facet* $(\overline{c}, \gamma)$ *of* $\mathrm{CUT}(\overline{G})$ *by splitting node* w *w.r.t.* (H, T, B) *defines a facet of* $\mathrm{CUT}(\hat{G})$ *if there exists a node set* $S \subseteq \overline{V}$ *with* $w \in S$ *that satisfies:*

(i) $(\overline{c}, \gamma)$ *is tight at* $\chi^{\delta(S)}$.

(ii) $\overline{c}$ *is nonnegative on* $(w : T \cap S)$ *and nonpositive on* $(w : T \setminus S)$.

(iii) $\overline{c}(v : S) = \overline{c}(v : \overline{V} \setminus S)$, *for all* $v \in B$.

Lifting: Facet Preservation

Theorem

An inequality $(\hat{c}, \gamma)$ *obtained from a facet* $(\overline{c}, \gamma)$ *of* $\mathrm{CUT}(\overline{G})$ *by splitting node* w *w.r.t.* (H, T, B) *defines a facet of* $\mathrm{CUT}(\hat{G})$ *if there exists a node set* $S \subseteq \overline{V}$ *with* $w \in S$ *that satisfies:*

(i) $(\overline{c}, \gamma)$ *is tight at* $\chi^{\delta(S)}$.

(ii) $\overline{c}$ *is nonnegative on* $(w : T \cap S)$ *and nonpositive on* $(w : T \setminus S)$.

(iii) $\overline{c}(v : S) = \overline{c}(v : \overline{V} \setminus S)$, *for all* $v \in B$.

Proof sketch

Show: face $(\hat{c}, \gamma) \subseteq$ facet $(\hat{b}, \beta) \Rightarrow$ IEQs identical up to pos. scaling.

Theorem

An inequality $(\hat{c}, \gamma)$ *obtained from a facet* $(\overline{c}, \gamma)$ *of* $\mathrm{CUT}(\overline{G})$ *by splitting node* w *w.r.t.* (H, T, B) *defines a facet of* $\mathrm{CUT}(\hat{G})$ *if there exists a node set* $S \subseteq \overline{V}$ *with* $w \in S$ *that satisfies:*

(i) $(\overline{c}, \gamma)$ *is tight at* $\chi^{\delta(S)}$.

(ii) $\overline{c}$ *is nonnegative on* $(w : T \cap S)$ *and nonpositive on* $(w : T \setminus S)$.

(iii) $\overline{c}(v : S) = \overline{c}(v : \overline{V} \setminus S)$, *for all* $v \in B$.

Proof sketch

Show: face $(\hat{c}, \gamma) \subseteq$ facet $(\hat{b}, \beta) \Rightarrow$ IEQs identical up to pos. scaling.

- $\exists \lambda > 0$ with $\hat{b}_e = \lambda \hat{c}_e$, for $e \notin (\{h, t\} : B) \cup ht$ and $\hat{b}_{hv} + \hat{b}_{tv} = \lambda \hat{c}_{hv}$, for $v \in B$. (due to tight cut correspondence)

Lifting: Facet Preservation

Theorem

An inequality $(\hat{c}, \gamma)$ obtained from a facet $(\overline{c}, \gamma)$ of $\mathrm{CUT}(\overline{G})$ by splitting node w w.r.t. (H, T, B) defines a facet of $\mathrm{CUT}(\hat{G})$ if there exists a node set $S \subseteq \overline{V}$ with $w \in S$ that satisfies:

(i) *$(\overline{c}, \gamma)$ is tight at $\chi^{\delta(S)}$.*

(ii) *$\overline{c}$ is nonnegative on $(w : T \cap S)$ and nonpositive on $(w : T \setminus S)$.*

(iii) *$\overline{c}(v : S) = \overline{c}(v : \overline{V} \setminus S)$, for all $v \in B$.*

Proof sketch

Show: face $(\hat{c}, \gamma) \subseteq$ facet $(\hat{b}, \beta) \Rightarrow$ IEQs identical up to pos. scaling.

- $\exists \lambda > 0$ with $\hat{b}_e = \lambda \hat{c}_e$, for $e \notin (\{h, t\} : B) \cup ht$ and $\hat{b}_{hv} + \hat{b}_{tv} = \lambda \hat{c}_{hv}$, for $v \in B$. (due to tight cut correspondence)
- $S' := S \setminus w \cup h$ and $S'' := S' \cup t$ induce tight cuts of $(\hat{c}, \gamma)$.

Theorem

An inequality $(\hat{c}, \gamma)$ *obtained from a facet* $(\overline{c}, \gamma)$ *of* $\mathrm{CUT}(\overline{G})$ *by splitting node* w *w.r.t.* (H, T, B) *defines a facet of* $\mathrm{CUT}(\hat{G})$ *if there exists a node set* $S \subseteq \overline{V}$ *with* $w \in S$ *that satisfies:*

(i) $(\overline{c}, \gamma)$ *is tight at* $\chi^{\delta(S)}$.

(ii) $\overline{c}$ *is nonnegative on* $(w : T \cap S)$ *and nonpositive on* $(w : T \setminus S)$.

(iii) $\overline{c}(v : S) = \overline{c}(v : \overline{V} \setminus S)$, *for all* $v \in B$.

Proof sketch

Show: face $(\hat{c}, \gamma) \subseteq$ facet $(\hat{b}, \beta) \Rightarrow$ IEQs identical up to pos. scaling.

- $\exists \lambda > 0$ with $\hat{b}_e = \lambda \hat{c}_e$, for $e \notin (\{h, t\} : B) \cup ht$ and $\hat{b}_{hv} + \hat{b}_{tv} = \lambda \hat{c}_{hv}$, for $v \in B$. (due to tight cut correspondence)
- $S' := S \setminus w \cup h$ and $S'' := S' \cup t$ induce tight cuts of $(\hat{c}, \gamma)$.
- Consider $\delta(S')$, let $v \in B$ swap shores $\rightsquigarrow$ $\hat{b}_{tv} = \lambda \hat{c}_{tv} = 0$.

Theorem

An inequality $(\hat{c}, \gamma)$ *obtained from a facet* $(\overline{c}, \gamma)$ *of* $\mathrm{CUT}(\overline{G})$ *by splitting node* w *w.r.t.* (H, T, B) *defines a facet of* $\mathrm{CUT}(\hat{G})$ *if there exists a node set* $S \subseteq \overline{V}$ *with* $w \in S$ *that satisfies:*

(i) $(\overline{c}, \gamma)$ *is tight at* $\chi^{\delta(S)}$.

(ii) $\overline{c}$ *is nonnegative on* $(w : T \cap S)$ *and nonpositive on* $(w : T \setminus S)$.

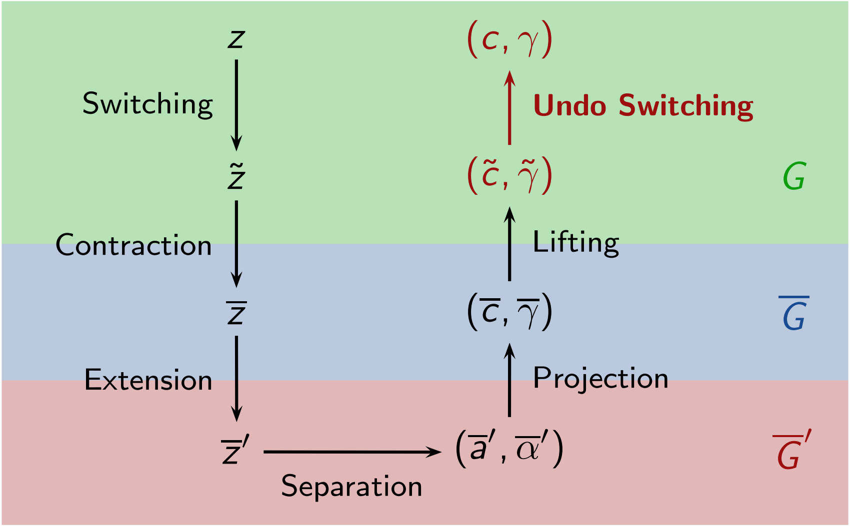
(iii) $\overline{c}(v : S) = \overline{c}(v : \overline{V} \setminus S)$, *for all* $v \in B$.

Proof sketch

Show: face $(\hat{c}, \gamma) \subseteq$ facet $(\hat{b}, \beta) \Rightarrow$ IEQs identical up to pos. scaling.

- $\exists\, \lambda > 0$ with $\hat{b}_e = \lambda \hat{c}_e$, for $e \notin (\{h, t\} : B) \cup ht$ and $\hat{b}_{hv} + \hat{b}_{tv} = \lambda \hat{c}_{hv}$, for $v \in B$. (due to tight cut correspondence)
- $S' := S \setminus w \cup h$ and $S'' := S' \cup t$ induce tight cuts of $(\hat{c}, \gamma)$.
- Consider $\delta(S')$, let $v \in B$ swap shores $\rightsquigarrow$ $\hat{b}_{tv} = \lambda \hat{c}_{tv} = 0$.
- Compare $\delta(S')$ and $\delta(S'')$ $\rightsquigarrow$ $\hat{b}_{ht} = \lambda \hat{c}_{ht}$.

Switch back lifted inequality.



Undo Switching: Facet Preservation

Theorem (Barahona and Mahjoub (1986))

An inequality

$$\sum_{e \notin \delta(S)} c_e x_e + \sum_{e \in \delta(S)} c_e (1 - x_e) \leq \gamma$$

obtained from a facet (c, γ) *of* $\mathrm{CUT}(G)$ *by switching alongside a cut* $\delta(S)$ *defines a facet of* $\mathrm{CUT}(G)$.

Theorem (Barahona and Mahjoub (1986))

An inequality

$$\sum_{e \notin \delta(S)} c_e x_e + \sum_{e \in \delta(S)} c_e (1 - x_e) \leq \gamma$$

obtained from a facet (c, γ) *of* CUT(G) *by switching alongside a cut* $\delta(S)$ *defines a facet of* CUT(G).

Proof sketch

For cuts induced by a single node v:

- $\left\{\chi^{\delta(U_i)}\right\}_{i=1,\ldots,|E|}$ are affinely independent and satisfy the original inequality with equality. W.l.o.g. $v \in U_i$, for all i.
- Thus, $\left\{\chi^{\delta(U_i \setminus v)}\right\}_{i=1,\ldots,|E|}$ are affinely independent and satisfy the switched inequality with equality.

Undo Switching: Facet Preservation

Theorem (Barahona and Mahjoub (1986))

An inequality

$$\sum_{e \notin \delta(S)} c_e x_e + \sum_{e \in \delta(S)} c_e (1 - x_e) \leq \gamma$$

obtained from a facet (c, γ) *of* $\mathrm{CUT}(G)$ *by switching alongside a cut* $\delta(S)$ *defines a facet of* $\mathrm{CUT}(G)$*.*

Proof sketch

For cuts induced by a single node v:

- $\left\{\chi^{\delta(U_i)}\right\}_{i=1,\ldots,|E|}$ are affinely independent and satisfy the original inequality with equality. W.l.o.g. $v \in U_i$, for all i.
- Thus, $\left\{\chi^{\delta(U_i \setminus v)}\right\}_{i=1,\ldots,|E|}$ are affinely independent and satisfy the switched inequality with equality.

For arbitrary cuts, iterate over all nodes in one of the cut's shores.

Outline

Computational Experiments

Used max-cut solver based on B&C framework ABACUS.

Used max-cut solver based on B&C framework ABACUS.

Problem classes

- Quadratic 0/1-optimization resp. max-cut problems taken from Biq Mac Library.
- Spin glass problems on toroidal grid graphs with uniformly distributed ± 1 resp. Gaussian distributed integral weights.

Used max-cut solver based on B&C framework ABACUS.

Problem classes

- Quadratic 0/1-optimization resp. max-cut problems taken from Biq Mac Library.
- Spin glass problems on toroidal grid graphs with uniformly distributed ± 1 resp. Gaussian distributed integral weights.

Separation schemes

- **Standard (CYC)**:
  Odd-cycles (spanning-tree heuristic, 3- / 4-cycles, exact separation).

Used max-cut solver based on B&C framework ABACUS.

Problem classes

- Quadratic 0/1-optimization resp. max-cut problems taken from Biq Mac Library.
- Spin glass problems on toroidal grid graphs with uniformly distributed ± 1 resp. Gaussian distributed integral weights.

Separation schemes

- **Standard (CYC)**:
  Odd-cycles (spanning-tree heuristic, 3-/4-cycles, exact separation).
- **Contraction (CON)**:
  "Standard" + contraction-based OC-separation prior to exact OC-separation.

Used max-cut solver based on B&C framework ABACUS.

Problem classes

- Quadratic 0/1-optimization resp. max-cut problems taken from Biq Mac Library.
- Spin glass problems on toroidal grid graphs with uniformly distributed ± 1 resp. Gaussian distributed integral weights.

Separation schemes

- **Standard (CYC)**:
  Odd-cycles (spanning-tree heuristic, 3- / 4-cycles, exact separation).
- **Contraction (CON)**:
  "Standard" + contraction-based OC-separation prior to exact OC-separation.
- **Extension (CLQ, TC)**:
  "Contraction" + additional separation on extended LP solution.

Comparison of CPU Times

Class	#Files	#Limit	#Wins				#Add. rejects			Avg. CPU time red. [%]		
			CYC	CON	CLQ	TC	CON	CLQ	TC	CON	CLQ	TC
bqp50	10	0	0	6	**8**	**8**	0	0	0	92	**97**	97
bqp100	10	0	0	**6**	**6**	4	0	0	0	**97**	97	96
bqp250	10	5	0	**5**	0	0	**2(0)**	3(3)	**2(0)**	**59**	9	41
be120.3	10	0	0	**9**	1	0	**1(0)**	1(1)	1(1)	**57**	−85	16
be250	10	6	0	**4**	0	0	0	1(1)	0	**54**	−23	47
gka.a	8	0	0	4	**5**	3	0	0	0	93	**94**	91
gka.b	10	0	0	**10**	0	0	0	0	4(4)	**58**	19	−40215
gka.c	7	0	0	2	3	**5**	0	0	0	95	95	**96**
gka.d	10	4	0	**6**	1	1	**1(0)**	1(1)	1(1)	**67**	−1	41
tpm.2d	120	8	0	**112**	—[b]	—[b]	**18(0)**	—[b]	—[b]	**96**	—[b]	—[b]
tg.2d	320	0	0	**320**	—[b]	—[b]	**1(0)**	—[b]	—[b]	**90**	—[b]	—[b]
tpm.3d	40	5	5	**25**	4	7	**1(0)**	5(5)	1(1)	**23**	−159	−56
tg.3d	70	7	2	**39**	17	20	4(2)	5(5)	**3(0)**	**44**	−40	26
g05_60[c]	10	0	0	—[a]	**10**	0	—[a]	0	3(3)	—[a]	$\geq$ **87**	$\geq$ 62
pm1s	20	0	**14**	6	0	0	0	0	0	**−16**	−309	−62
w01	10	0	**8**	1	0	1	0	0	0	**−37**	−458	−73
pw01	10	0	**8**	2	0	0	0	0	0	**−24**	−685	−102
man	4	0	0	**3**	1	0	**2(0)**	**2(0)**	**2(0)**	**71**	70	61
pman	57	0	7	23	**27**	22	9(7)	**9(0)**	10(3)	67	**68**	64

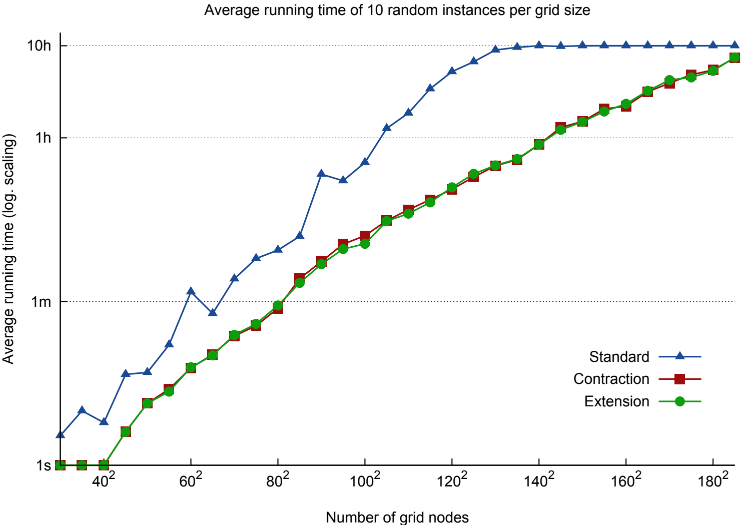
[a] CON ran out of memory for every instance.

[b] Equivalent to CON.

[c] CYC exceeded the 10 hour time limit on all instances. We used the limit as lower bound on the CPU time of CYC.

[Intel Xeon 2.8 GHz, 8GB shared RAM. **Running time capped to 10h per instance.**]

Spin Glass Problems with Gaussian Distributed Integral Weights

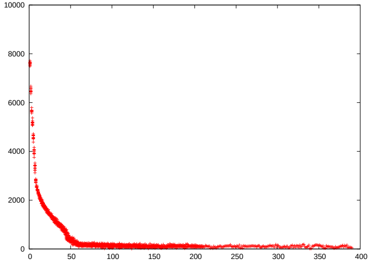


[Intel Xeon 2.8 GHz, 8GB shared RAM. **Running time capped to 10h per instance.**]

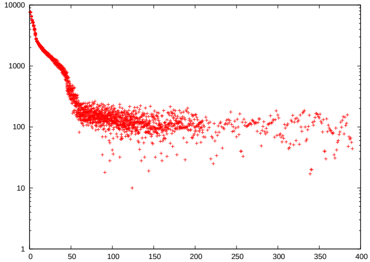
Data origin

- Performed optimization runs on ten toroidal (100×100)-grids with Gaussian distributed edge weights.
- Recorded the final sizes of those graphs that could be contracted without encountering violated odd-cycles.

Figures show the superimposed data of the optimization runs.



Standard scaling



Logarithmic scaling

Conclusion

Contraction-based Separation

- Enables the use of separation techniques for dense / complete graphs on sparse graphs.

Contraction-based Separation

- Enables the use of separation techniques for dense / complete graphs on sparse graphs.
- Use of target cut separation can produce facet defining inequalities not available through techniques that follow the classical template paradigm.

Contraction-based Separation

- Enables the use of separation techniques for dense / complete graphs on sparse graphs.
- Use of target cut separation can produce facet defining inequalities not available through techniques that follow the classical template paradigm.
- In our experiments, using the contraction as heuristic odd-cycle separator lead to an average CPU time reduction of almost 55% with peak values of up to 97% for selected problem classes.

- Avis, Imai, Ito. *Generating facets for the cut polytope of a graph by triangular elimination*. 2008.
- Barahona, Mahjoub. *On the cut polytope*. Mathematical Programming 36:157–173. 1986.
- Bonato, Jünger, Reinelt, Rinaldi. *Lifting and separation procedures for the cut polytope*. Mathematical Programming A. 2013.
- Buchheim, Liers, Oswald. *Local cuts revisited*. Operations Research Letters 36:430–433. 2008.
- Jünger, Thienel. *ABACUS - A Branch-And-CUt System*. `http://www.informatik.uni-koeln.de/abacus/`.
- Wiegele. *Biq Mac Library - A collection of Max-Cut and quadratic 0-1 programming instances of medium size*. `http://biqmac.uni-klu.ac.at/biqmaclib.html`.