Separation for the Max-Cut Problem on General Graphs

Thorsten Bonato

Research Group Discrete and Combinatorial Optimization
University of Heidelberg

Joint work with:
Michael Jünger (University of Cologne)
Gerhard Reinelt (University of Heidelberg)
Giovanni Rinaldi (IASI, Rome)

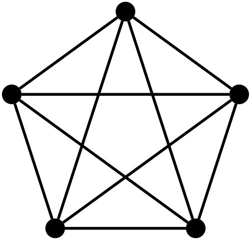
14th Combinatorial Optimization Workshop
Aussois, January 6, 2010

Outline

Outline

Definition

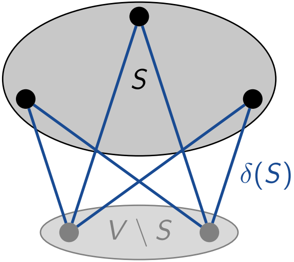
Let $G = (V, E, c)$ be an undirected weighted graph.



Definition

Let $G = (V, E, c)$ 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 $V \setminus S$.

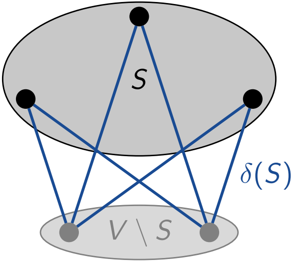


Definition

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

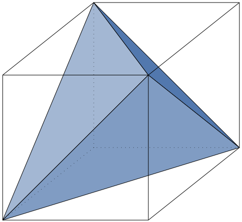
Any $S \subseteq V$ induces a set $\delta(S)$ of edges with exactly one end node in S. The set $\delta(S)$ is called a cut of G with shores S and $V \setminus S$.

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



Cut polytope CUT(G)

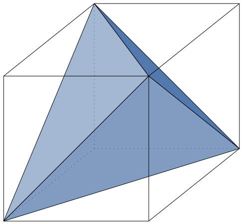
Convex hull of all incidence vectors of cuts of G.



$\text{CUT}(K_3)$

Cut polytope CUT(G)

Convex hull of all incidence vectors of cuts of G.



$\mathrm{CUT}(K_3)$

Semimetric polytope MET(G)

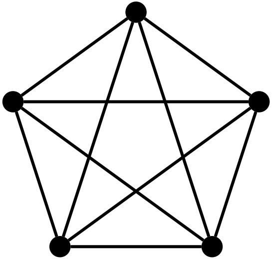
Relaxation of the max-cut IP formulation described by two inequality classes:

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 $e \in E$.

Algorithms

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

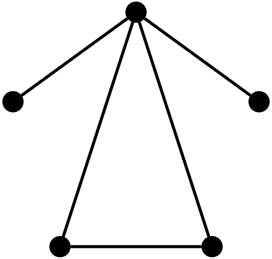
Certain separation procedures only work for dense/complete graphs.



Algorithms

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

Certain separation procedures only work for dense/complete graphs.

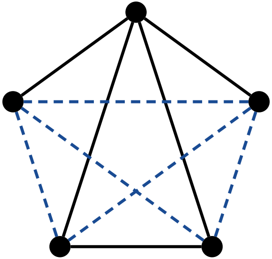


How to handle sparse graphs?

Algorithms

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

Certain separation procedures only work for dense/complete graphs.

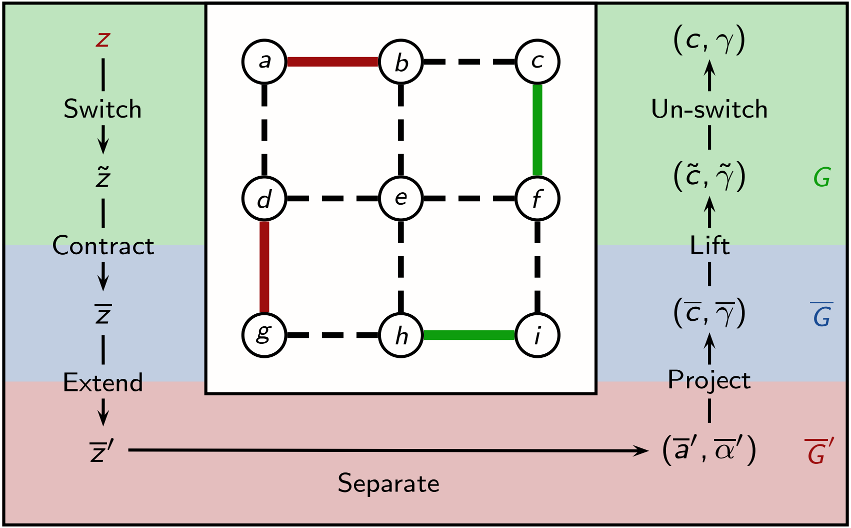


How to handle sparse graphs

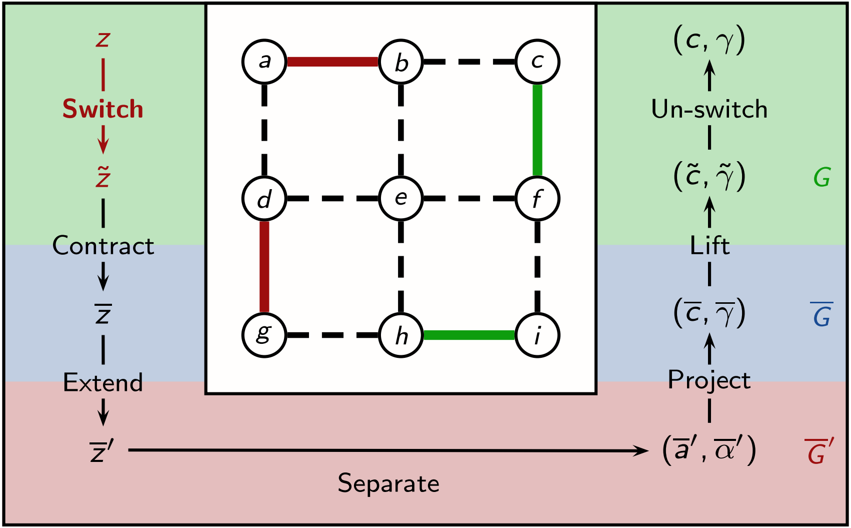
- **Trivial approach**:
  artificial completion using edges with weight 0.
- **Drawback**:
  increases number of variables and thus the computational difficulty.

Outline

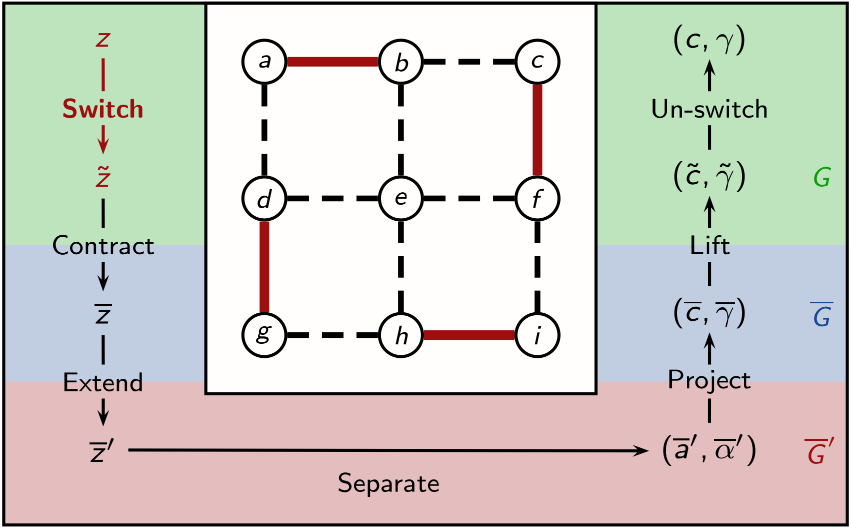
Input: LP solution $z \in \mathrm{MET}(G) \setminus \mathrm{CUT}(G)$.



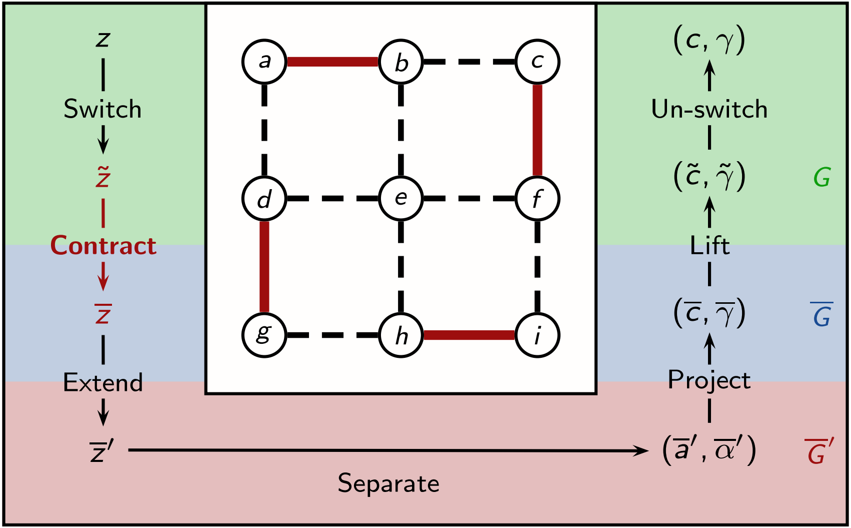
Transform 1-edges into 0-edges.



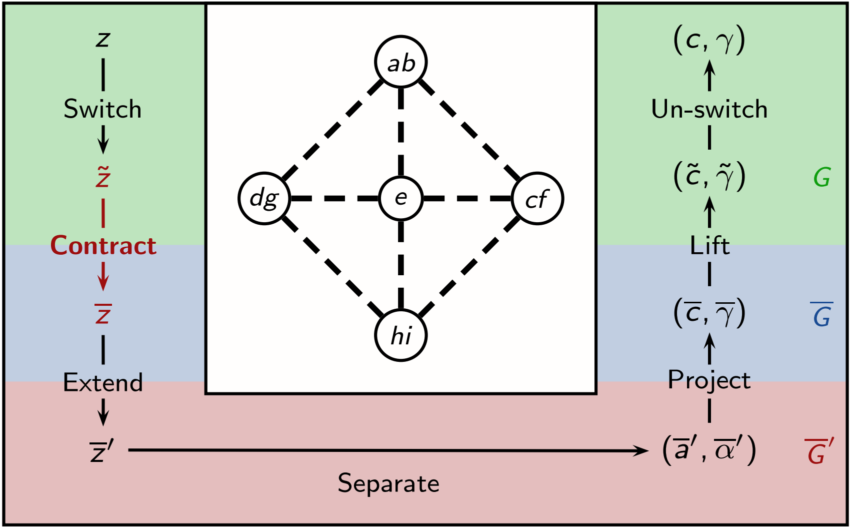
Transform 1-edges into 0-edges.



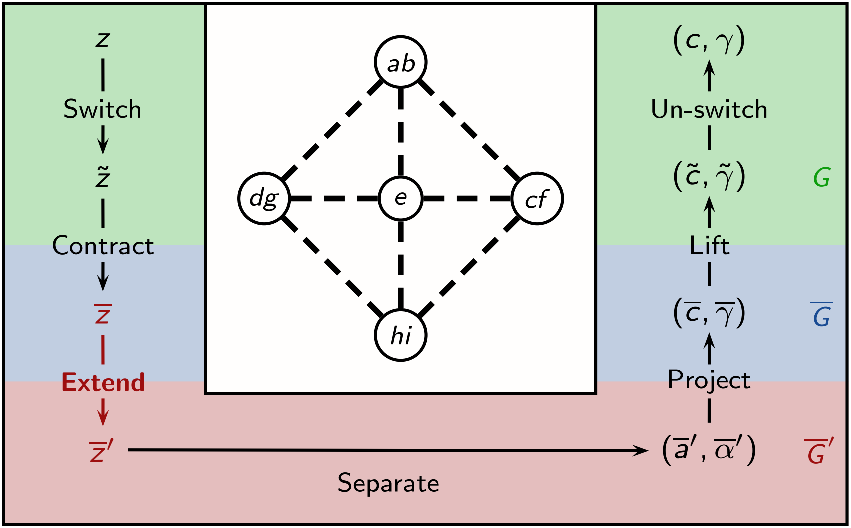
Contract 0-edges.



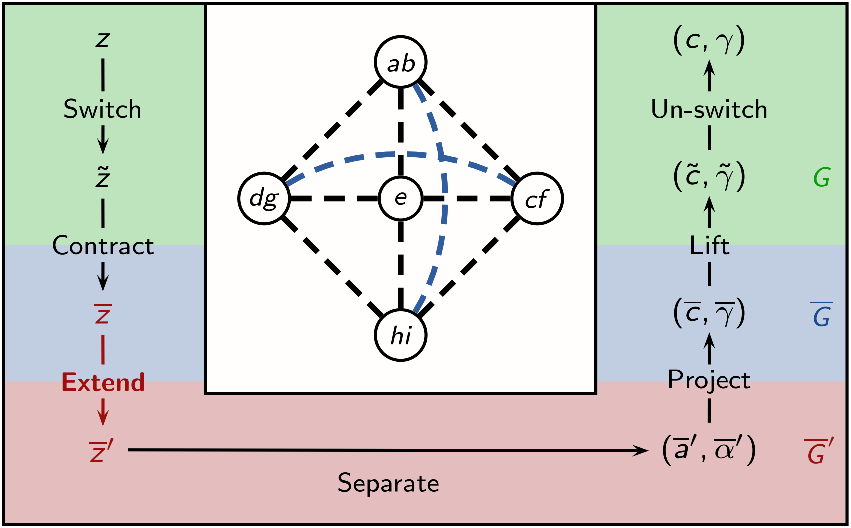
Contract 0-edges.



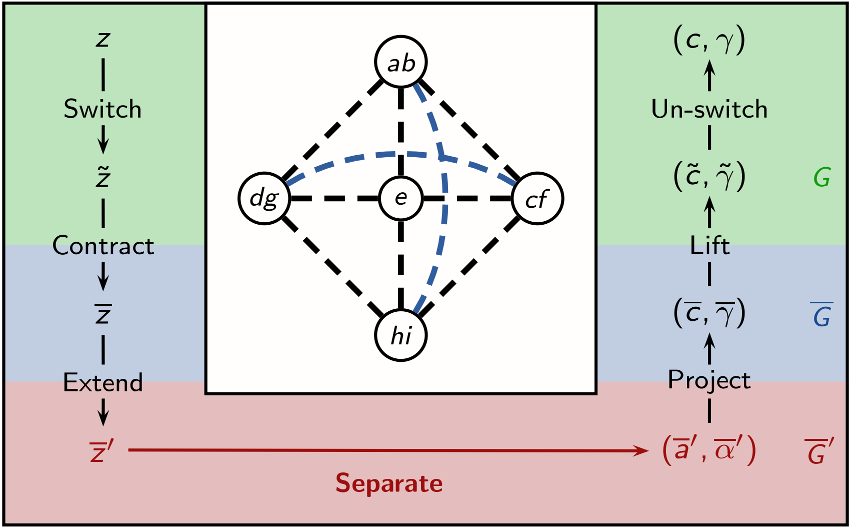
Introduce artificial LP values for non-edges.



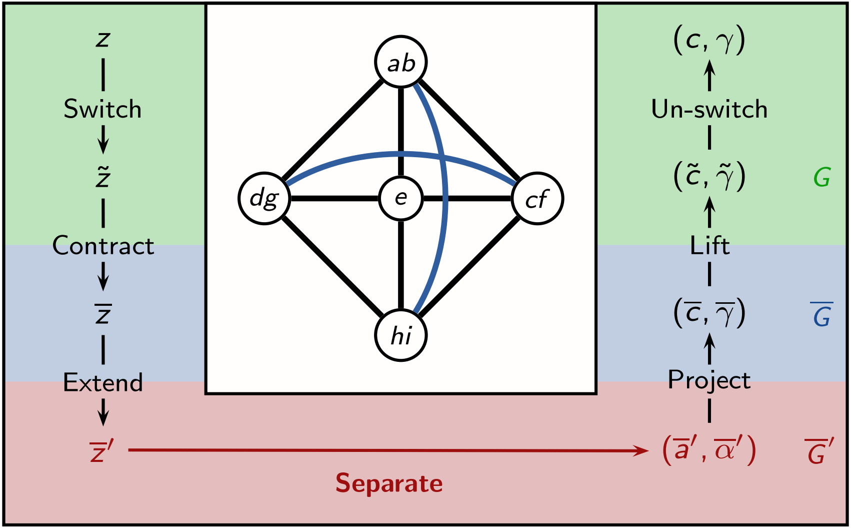
Introduce artificial LP values for non-edges.



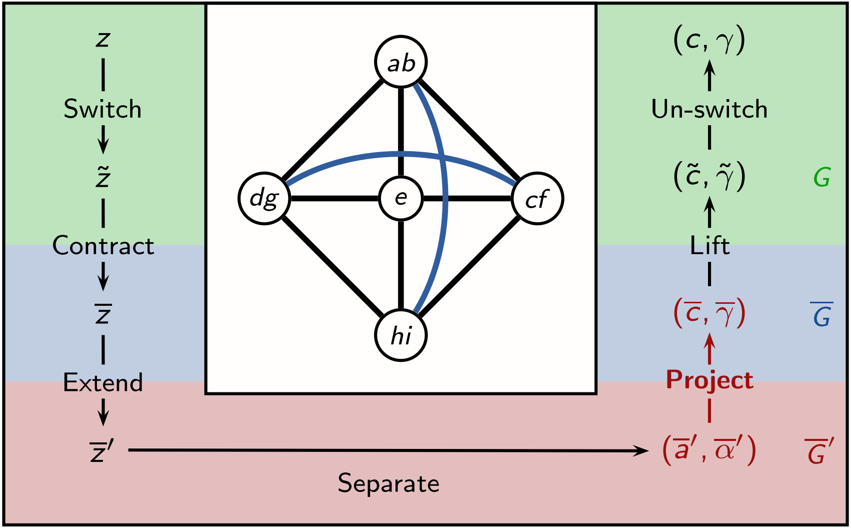
Separate extended LP solution.



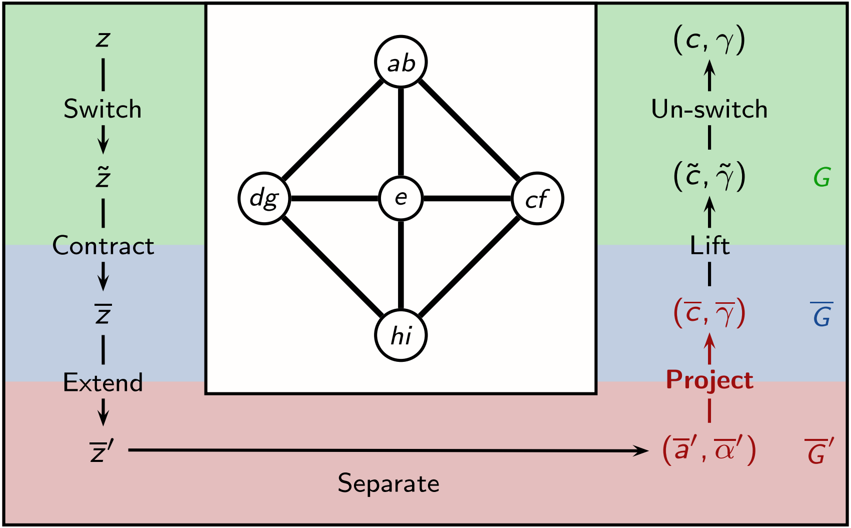
Separate extended LP solution.



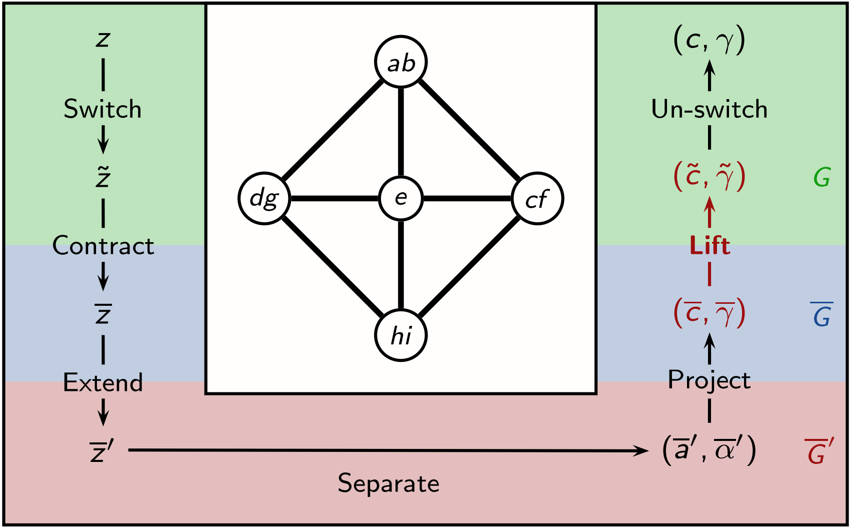
Project out nonzero coefficients related to non-edges.



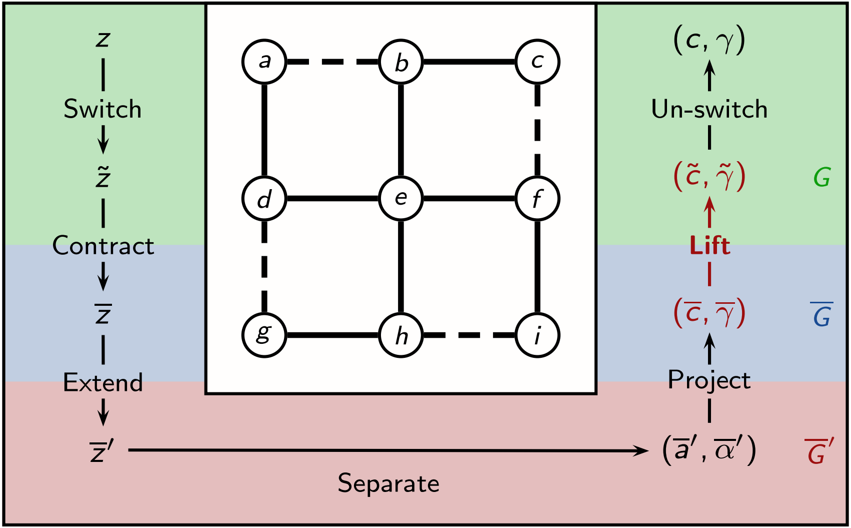
Project out nonzero coefficients related to non-edges.



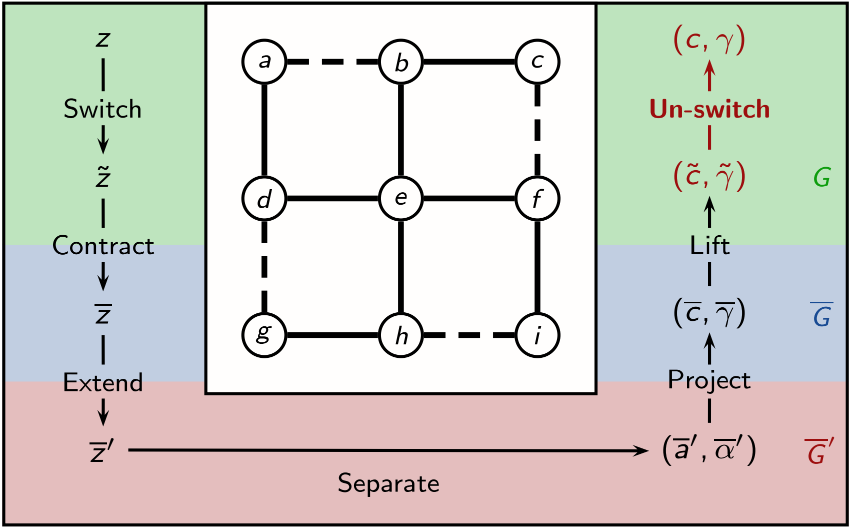
Lift inequality.



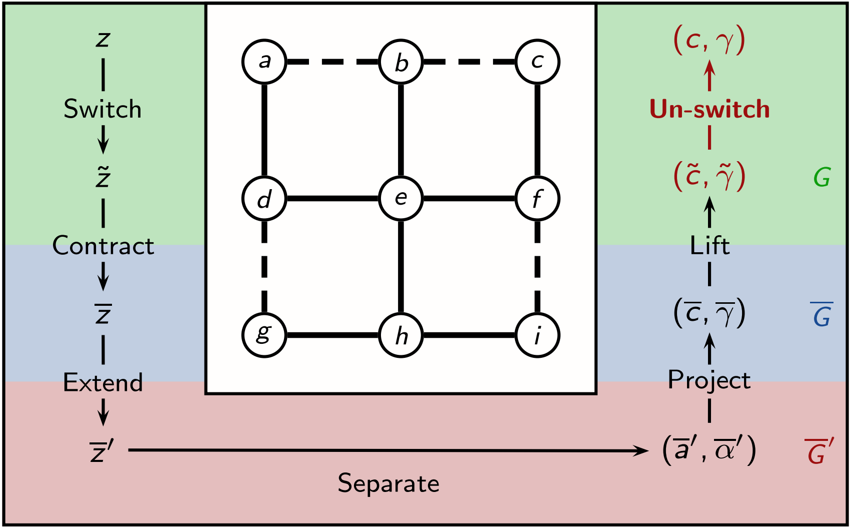
Lift inequality.

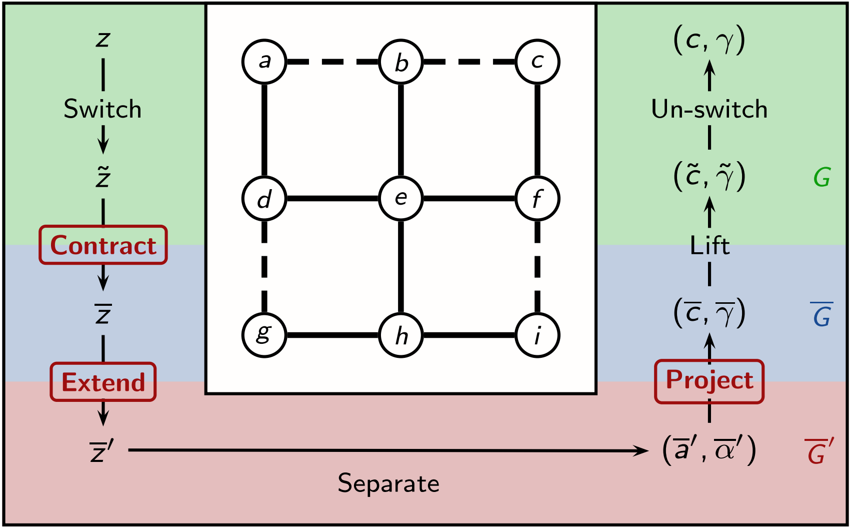


Switch lifted inequality.

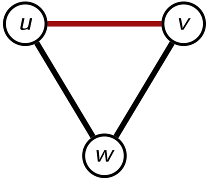


Switch lifted inequality.



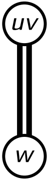

z
Switch
z̃
Contract
z̄
Extend
z̄′
Separate
(ā′, ᾱ′)
Project
(c̄, γ̄)
Lift
(c̃, γ̃)
Un-switch
(c, γ)
G
Ḡ
Ḡ′
a
b
c
d
e
f
g
h
i

Assume the end nodes of a 0-edge uv share a common neighbor w.



Assume the end nodes of a 0-edge uv share a common neighbor w.

Contraction of uv merges the edges uw and vw.



Assume the end nodes of a 0-edge uv share a common neighbor w.

Contraction of uv merges the edges uw and vw.

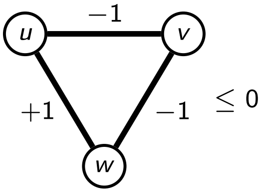
If the LP values of the merged edges differ, e. g., $z_{uw} > z_{vw}$

Assume the end nodes of a 0-edge uv share a common neighbor w.

Contraction of uv merges the edges uw and vw.

If the LP values of the merged edges differ, e. g., $z_{uw} > z_{vw}$ then z violates the odd-cycle inequality

$$x_{uw} - x_{vw} - x_{uv} \leq 0.$$

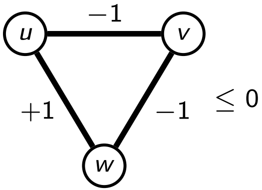


Assume the end nodes of a 0-edge uv share a common neighbor w.

Contraction of uv merges the edges uw and vw.

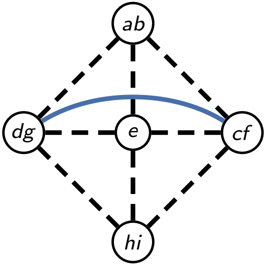
If the LP values of the merged edges differ, e. g., $z_{uw} > z_{vw}$ then z violates the odd-cycle inequality

$$x_{uw} - x_{vw} - x_{uv} \leq 0.$$



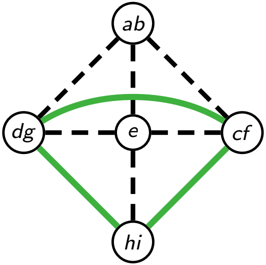
Contraction allows heuristic odd-cycle separation.

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



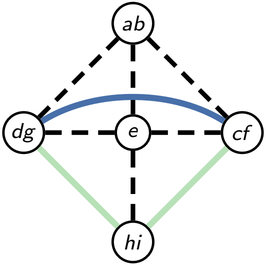
Given a contracted LP solution $\overline{z} \in \mathrm{MET}(\overline{G})$, assign artificial LP values to the non-edges.
Goal: extended LP solution $\overline{z}' \in \mathrm{MET}(\overline{G}')$.

New cycles in the extended graph



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

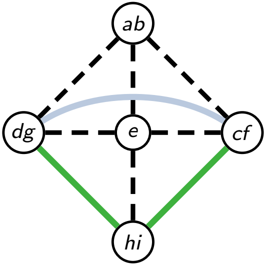
New cycles in the extended graph consist of a former non-edge



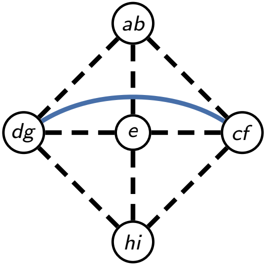
Extension

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 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.

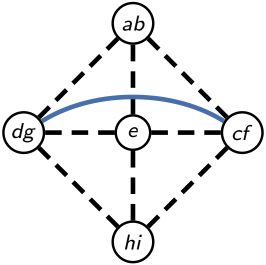
Feasible artificial LP values of non-edge uv

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

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

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 uv

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

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

Odd-cycle inequality derived from arg max (resp. arg min) is called a lower (resp. upper) inequality of *uv*.

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!

$$
\begin{aligned}
&(\cdots \quad \overline{\mathbf{a}}'_{\mathbf{uv}} \quad \cdots \quad \overline{\mathbf{a}}'_{\mathbf{st}} \quad \cdots, \ \overline{\alpha}') \\
+&(\cdots \ -\overline{\mathbf{a}}'_{\mathbf{uv}} \quad \cdots \quad \cdots \quad \cdots, \ \overline{\beta}'_1) \\
+&(\cdots \quad \cdots \quad \cdots \ -\overline{\mathbf{a}}'_{\mathbf{st}} \quad \cdots, \ \overline{\beta}'_2)
\end{aligned}
$$

Project out coefficient of non-edge uv

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

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!

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

Project out coefficient of non-edge uv

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

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!

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

Project out coefficient of non-edge uv

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

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

Problem

If the added inequalities are not tight at $\overline{z}'$ then the projection reduces the initial violation $\overline{a}'^T \overline{z}' - \overline{\alpha}'$.

Adaptive Extension

Artificial LP values $\overline{z}'_{uv}$ adapt to the sign of the corresponding coefficient in a given inequality $\overline{a}'^T \overline{x}' \leq \overline{\alpha}'$, i. e.,

$$\overline{z}'_{uv} = \begin{cases} L_{uv} & \text{if } \overline{a}'_{uv} > 0, \\ U_{uv} & \text{otherwise.} \end{cases}$$

Adaptive Extension

Artificial LP values $\overline{z}'_{uv}$ adapt to the sign of the corresponding coefficient in a given inequality $\overline{a}'^T\overline{x}' \leq \overline{\alpha}'$, i. e.,

$$\overline{z}'_{uv} = \begin{cases} L_{uv} & \text{if } \overline{a}'_{uv} > 0, \\ U_{uv} & \text{otherwise.} \end{cases}$$

Advantage: Violation remains unchanged during projection.
Drawback: Separation procedures may need to be modified.

Artificial LP values $\overline{z}'_{uv}$ adapt to the sign of the corresponding coefficient in a given inequality $\overline{a}'^T\overline{x}' \leq \overline{\alpha}'$, i. e.,

$$\overline{z}'_{uv} = \begin{cases} L_{uv} & \text{if } \overline{a}'_{uv} > 0, \\ U_{uv} & \text{otherwise.} \end{cases}$$

Advantage: Violation remains unchanged during projection.
Drawback: Separation procedures may need to be modified.

Trivial modification case

For a given class of inequalities, all nonzero coefficients have identical sign.

Artificial LP values $\overline{z}'_{uv}$ adapt to the sign of the corresponding coefficient in a given inequality $\overline{a}'^T\overline{x}' \leq \overline{\alpha}'$, i. e.,

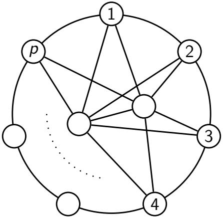
$$\overline{z}'_{uv} = \begin{cases} L_{uv} & \text{if } \overline{a}'_{uv} > 0, \\ U_{uv} & \text{otherwise.} \end{cases}$$

Advantage: Violation remains unchanged during projection.
Drawback: Separation procedures may need to be modified.

Trivial modification case

For a given class of inequalities, all nonzero coefficients have identical sign.

E. g., bicycle-p-wheel inequalities: $x(B) \leq 2p$ (set $\overline{z}'_{uv} = L_{uv}$ for all non-edges uv).



Adaptive Extension: Target Cuts (1/2)

Input for separation framework [Buchheim, Liers, and Oswald]

- Associated polyhedron $Q = \text{conv}\{x_1, \ldots, x_s\} + \text{cone}\{y_1, \ldots, y_t\}$,
- Interior point $q \in Q$,
- Point $z \notin Q$ to be separated.

Input for separation framework [Buchheim, Liers, and Oswald]

- Associated polyhedron $Q = \text{conv}\{x_1, \ldots, x_s\} + \text{cone}\{y_1, \ldots, y_t\}$,
- Interior point $q \in Q$,
- Point $z \notin Q$ to be separated.

Obtain facet defining inequality $a^T(x - q) \leq 1$ by solving the LP

$$\begin{aligned} \max \quad & a^T(z - q) \\ \text{s.t.} \quad & a^T(x_i - q) \leq 1, \quad \text{for all } i = 1, \ldots, s \\ & a^T y_j \leq 0, \quad \text{for all } j = 1, \ldots, t \\ & a \in \mathbb{R}^m \end{aligned}$$

Input for separation framework [Buchheim, Liers, and Oswald]

- Associated polyhedron $Q = \text{conv}\{x_1, \ldots, x_s\} + \text{cone}\{y_1, \ldots, y_t\}$,
- Interior point $q \in Q$,
- Point $z \notin Q$ to be separated.

Obtain facet defining inequality $a^T(x - q) \leq 1$ by solving the LP

$$\begin{aligned} \max\ & a^T(z - q) \\ \text{s.t.}\ & a^T(x_i - q) \leq 1, \quad \text{for all } i = 1, \ldots, s \\ & a^T y_j \leq 0, \quad \text{for all } j = 1, \ldots, t \\ & a \in \mathbb{R}^m \end{aligned}$$

For max-cut we set $Q = \text{CUT}\big(G(W)\big)$ for a subset $W \subseteq V$.

Adaptive Extension: Target Cuts (2/2)

Modified input

W.l.o.g. let the last ℓ vector entries correspond to the non-edges.

Modified input

W.l.o.g. let the last ℓ vector entries correspond to the non-edges.

$$
\begin{aligned}
z' &:= (z_1, \ldots, z_{m-\ell}, \quad L_1, \ldots, L_\ell, \quad U_1, \ldots, U_\ell),\\
x_i' &:= (x_{i1}, \ldots, x_{i,m-\ell}, \quad x_{i,m-\ell+1}, \ldots, x_{im}, \quad x_{i,m-\ell+1}, \ldots, x_{im}),\\
q' &:= (q_1, \ldots, q_{m-\ell}, \quad q_{m-\ell+1}, \ldots, q_m, \quad q_{m-\ell+1}, \ldots, q_m),
\end{aligned}
$$

$$Q' := \operatorname{conv}\{x_1', \ldots, x_s'\} + \operatorname{cone}\{-e_{m-\ell+k}, e_{m+k} \mid k = 1, \ldots, \ell\}.$$

Modified input

W.l.o.g. let the last ℓ vector entries correspond to the non-edges.

$$
\begin{aligned}
z' &:= (z_1, \ldots, z_{m-\ell}, \quad L_1, \quad \ldots, L_\ell, \quad U_1, \quad \ldots, U_\ell),\\
x_i' &:= (x_{i1}, \ldots, x_{i,m-\ell}, \quad x_{i,m-\ell+1}, \ldots, x_{im}, \quad x_{i,m-\ell+1}, \ldots, x_{im}),\\
q' &:= (q_1, \ldots, q_{m-\ell}, \quad q_{m-\ell+1}, \ldots, q_m, \quad q_{m-\ell+1}, \ldots, q_m),
\end{aligned}
$$

$$
Q' := \operatorname{conv}\{x_1', \ldots, x_s'\} + \operatorname{cone}\{-e_{m-\ell+k}, e_{m+k} \mid k = 1, \ldots, \ell\}.
$$

Resulting target cut separation LP

$$
\begin{aligned}
\max \quad & a'^T(z' - q') \\
\text{s.t.} \quad & a'^T(x_i' - q') \le 1, \quad \text{for all } i = 1, \ldots, s \\
& -a'_{m-\ell+k},\ a'_{m+k} \le 0, \quad \text{for all } k = 1, \ldots, \ell \\
& a' \in \mathbb{R}^{m+\ell}
\end{aligned}
$$

Modified input

W.l.o.g. let the last ℓ vector entries correspond to the non-edges.

$$\begin{aligned}
z' &:= (z_1, \dots, z_{m-\ell}, \quad L_1, \dots, L_\ell, \quad U_1, \dots, U_\ell), \\
x'_i &:= (x_{i1}, \dots, x_{i,m-\ell}, \quad x_{i,m-\ell+1}, \dots, x_{im}, \quad x_{i,m-\ell+1}, \dots, x_{im}), \\
q' &:= (q_1, \dots, q_{m-\ell}, \quad q_{m-\ell+1}, \dots, q_m, \quad q_{m-\ell+1}, \dots, q_m),
\end{aligned}$$

$$Q' := \operatorname{conv}\{x'_1, \dots, x'_s\} + \operatorname{cone}\{-e_{m-\ell+k}, e_{m+k} \mid k = 1, \dots, \ell\}.$$

Resulting target cut separation LP

$$\begin{aligned}
\max \quad & a'^T(z' - q') \\
\text{s.t.} \quad & a'^T(x'_i - q') \leq 1, \quad \text{for all } i = 1, \dots, s \\
& -a'_{m-\ell+k},\ a'_{m+k} \leq 0, \quad \text{for all } k = 1, \dots, \ell \\
& a' \in \mathbb{R}^{m+\ell}
\end{aligned}$$

In an optimum solution a'^* at most one of $a'^*_{m-\ell+k}$ and a'^*_{m+k} can be nonzero for each $k = 1, \dots, \ell$.

Outline

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

Computational Experiments

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

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

Problem classes

1. Unconstrained quadratic 0/1-optimization problems.
2. Spin glass problems on toroidal grid graphs with:
   - Uniformly distributed ± 1-weights.
   - Gaussian distributed integral weights.

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

Problem classes

1. Unconstrained quadratic 0/1-optimization problems.
2. Spin glass problems on toroidal grid graphs with:
   - Uniformly distributed ± 1-weights.
   - Gaussian distributed integral weights.

Separation schemes

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

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

Problem classes

1. Unconstrained quadratic 0/1-optimization problems.
2. Spin glass problems on toroidal grid graphs with:
   - Uniformly distributed ± 1-weights.
   - Gaussian distributed integral weights.

Separation schemes

- **Standard**:
  odd-cycles (spanning-tree heuristic, 3-/4-cycles, exact separation).
- **Contraction**:
  standard scheme + contraction as heuristic OC-separator.

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

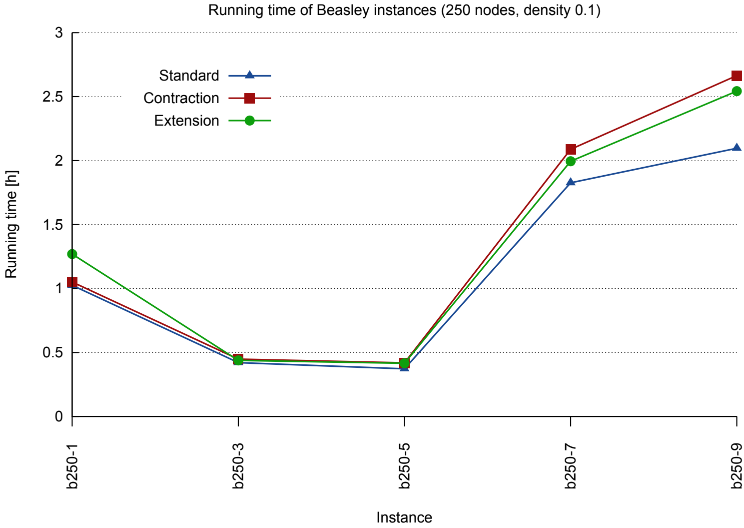
Problem classes

1. Unconstrained quadratic 0/1-optimization problems.
2. Spin glass problems on toroidal grid graphs with:
   - Uniformly distributed ± 1-weights.
   - Gaussian distributed integral weights.

Separation schemes

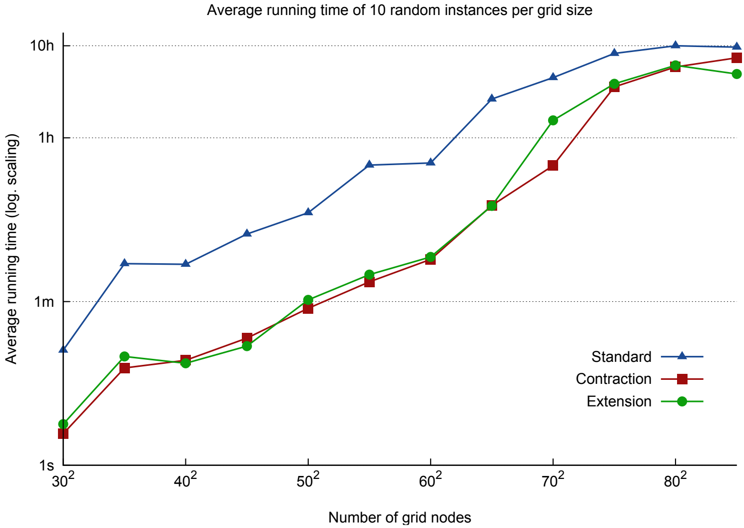
- **Standard**:
  odd-cycles (spanning-tree heuristic, 3-/4-cycles, exact separation).
- **Contraction**:
  standard scheme + contraction as heuristic OC-separator.
- **Extension**:
  contraction scheme + separation of bicycle-p-wheels, hypermetric inequalities and target cuts on the extended LP solution.

Unconstrained Quadratic 0/1-Optimization Problems



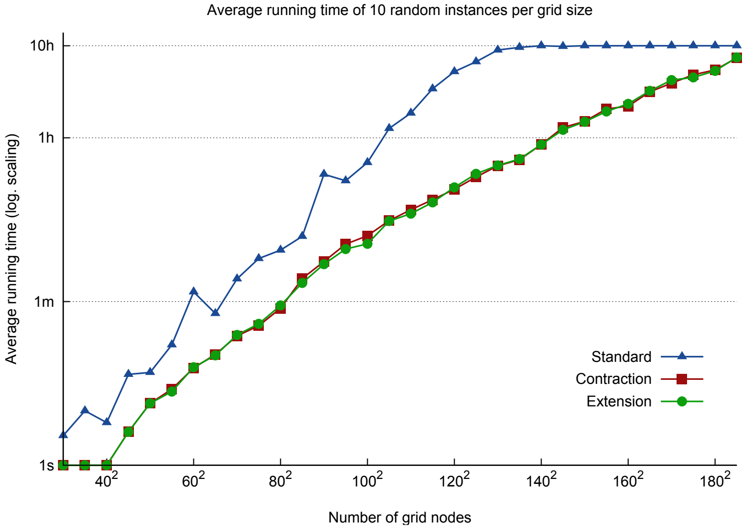
[Intel Xeon 2.8 GHz, 8GB shared RAM.]

Spin Glass Problems with Uniformly Distributed ±1-Weights



[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.**]

Separation using graph contraction

- Enables the use of separation techniques for dense / complete graphs on sparse graphs.
- Accelerates the exact solution of the max-cut problem for the examined classes of spin glass problems.
- Acceleration is mainly due to the use of contraction as heuristic odd-cycle separator.

Conclusion and Future Work

Separation using graph contraction

- Enables the use of separation techniques for dense / complete graphs on sparse graphs.
- Accelerates the exact solution of the max-cut problem for the examined classes of spin glass problems.
- Acceleration is mainly due to the use of contraction as heuristic odd-cycle separator.

Future work

- Develop special branching rules.
- Determine good parameter settings.
- Further computational experiments.

Separation using graph contraction

- Enables the use of separation techniques for dense / complete graphs on sparse graphs.
- Accelerates the exact solution of the max-cut problem for the examined classes of spin glass problems.
- Acceleration is mainly due to the use of contraction as heuristic odd-cycle separator.

Future work

- Develop special branching rules.
- Determine good parameter settings.
- Further computational experiments.

Thank you for your attention!