UNIVERSITY OF TWENTE.

Cutting Planes for Multilinear Optimization

Matthias Walter (Uni Twente)
Joint work with Emily Schutte (Uni Luxembourg)
and Alberto del Pia (Uni Wisconsin-Madison)

Funding support from the Dutch Research Council (NWO) on grant number OCENW.M20.151.

UNIVERSITY OF TWENTE.

Cutting Planes for Multilinear Optimization

Matthias Walter (Uni Twente)
Joint work with Emily Schutte (Uni Luxembourg)
and Alberto del Pia (Uni Wisconsin-Madison)

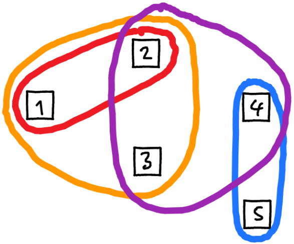
Funding support from the Dutch Research Council
(NWO) on grant number OCENW.M20.151.

Definition – Multilinear polytope [Del Pia, Khajavirad '16; Buchheim, Crama, Rodríguez-Heck '16]

Let $G = (V, E)$ be a hypergraph. The **multilinear polytope** of G is the polytope

$$\text{ML}(G) := \text{conv}\left\{(x, y) \in \{0,1\}^V \times \{0,1\}^E : y_e = \prod_{v \in e} x_v \text{ for each hyperedge } \{u, v\} \in E\right\}.$$

Example:



$y_{\{1,2\}} = x_1 \cdot x_2$

$y_{\{1,2,3\}} = x_1 \cdot x_2 \cdot x_3$

$y_{\{2,3,4\}} = x_2 \cdot x_3 \cdot x_4$

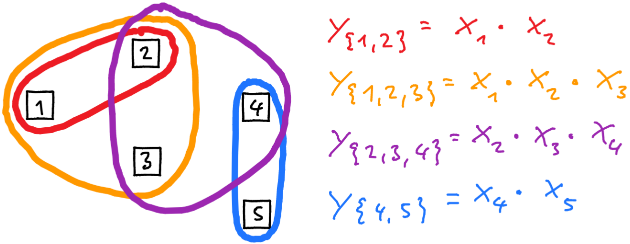
$y_{\{4,5\}} = x_4 \cdot x_5$

Definition – Multilinear polytope [Del Pia, Khajavirad '16; Buchheim, Crama, Rodríguez-Heck '16]

Let $G = (V, E)$ be a hypergraph. The **multilinear polytope** of G is the polytope

$$\text{ML}(G) := \text{conv}\left\{(x, y) \in \{0,1\}^V \times \{0,1\}^E : y_e = \prod_{v \in e} x_v \text{ for each hyperedge } \{u, v\} \in E\right\}.$$

Example:



Remarks:

- For binary $x \in \{0,1\}^V$, the original optimization problem (without auxiliary variables) is called pseudo-boolean optimization problem. [Hammer, Hansen, Simeone '84; Boros, Hammer '91]
- For each hyperedge $e = \{v_1, v_2, \ldots, v_k\}$, we have the logic AND constraint $y_e = x_{v_1} \wedge x_{v_2} \wedge \cdots \wedge x_{v_k}$.

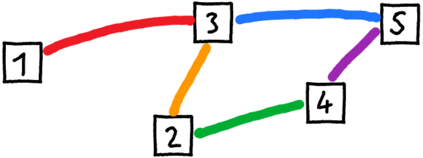
Special Case: Boolean Quadric Polytopes

Definition – Boolean quadric polytope **[Padberg '88]**

Let $G = (V, E)$ be a graph. The **boolean quadric polytope** of G is the polytope

$$\mathrm{BQP}(G) := \mathrm{conv}\left\{(x,y) \in \{0,1\}^V \times \{0,1\}^E : y_{\{u,v\}} = x_u \cdot x_v \text{ for each edge } \{u,v\} \in E\right\}.$$

Example:



$y_{\{1,3\}} = x_1 \cdot x_3$ $\quad$ $y_{\{3,5\}} = x_3 \cdot x_5$

$y_{\{2,3\}} = x_2 \cdot x_3$ $\quad$ $y_{\{4,5\}} = x_4 \cdot x_5$

$y_{\{2,4\}} = x_2 \cdot x_4$ $\quad$ $x_i \in \{0,1\} \quad i = 1, \dots, 5$

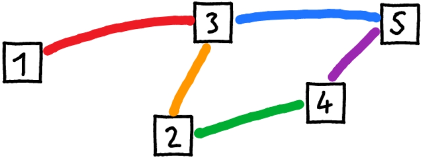
Special Case: Boolean Quadric Polytopes

Definition – Boolean quadric polytope **[Padberg '88]**

Let $G = (V, E)$ be a graph. The **boolean quadric polytope** of G is the polytope

$$\text{BQP}(G) := \text{conv}\left\{(x, y) \in \{0,1\}^V \times \{0,1\}^E : y_{\{u,v\}} = x_u \cdot x_v \text{ for each edge } \{u,v\} \in E\right\}.$$

Example:



$y_{\{1,3\}} = x_1 \cdot x_3$ $\quad$ $y_{\{3,5\}} = x_3 \cdot x_5$

$y_{\{2,3\}} = x_2 \cdot x_3$ $\quad$ $y_{\{4,5\}} = x_4 \cdot x_5$

$y_{\{2,4\}} = x_2 \cdot x_4$ $\quad$ $x_i \in \{0,1\} \quad i = 1, \ldots, 5$

Remarks:

- Equivalent to CUT polytope of related graph. [Barahona, Mahjoub '86; De Simone '90]
- Can be used to minimize a quadratic function $q(x)$ over $x \in \{0,1\}^n$, also known as "quadratic unconstrained binary optimization" (QUBO).
- Optimization over BQP is NP-hard in general.

Standard Relaxation

Proposition – Standard relaxation [Fortet '60; Glover, Woolsey '74]

Let $G = (V, E)$ be a hypergraph. The polytope $\text{SR}(G)$ defined by

$$0 \leq y_e \leq x_v \leq 1 \qquad \forall v \in e \in E \qquad (1a)$$

$$y_e + \sum_{v \in e} (1 - x_v) \geq 1 \qquad \forall e \in E \qquad (1b)$$

yields an IP formulation, i.e., $\text{SR}(G) \cap \mathbb{Z}^{V+E} = \text{ML}(G) \cap \mathbb{Z}^{V+E}$.

Standard Relaxation

Proposition – Standard relaxation [Fortet '60; Glover, Woolsey '74]

Let $G = (V, E)$ be a hypergraph. The polytope $\mathrm{SR}(G)$ defined by

$$0 \leq y_e \leq x_v \leq 1 \qquad \forall v \in e \in E \qquad (1a)$$

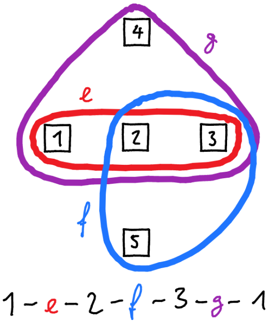
$$y_e + \sum_{v \in e} (1 - x_v) \geq 1 \qquad \forall e \in E \qquad (1b)$$

yields an IP formulation, i.e., $\mathrm{SR}(G) \cap \mathbb{Z}^{V+E} = \mathrm{ML}(G) \cap \mathbb{Z}^{V+E}$.

Berge cycle:

$v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_1$ with:

- $v_i \in V$ are distinct nodes.
- $e_i \in E$ are distinct edges.
- $v_i \in e_{i-1} \cap e_i$ for each i



Proposition – Standard relaxation [Fortet '60; Glover, Woolsey '74]

Let $G = (V, E)$ be a hypergraph. The polytope $\mathrm{SR}(G)$ defined by

$$0 \leq y_e \leq x_v \leq 1 \qquad \forall v \in e \in E \qquad (1a)$$

$$y_e + \sum_{v \in e} (1 - x_v) \geq 1 \qquad \forall e \in E \qquad (1b)$$

yields an IP formulation, i.e., $\mathrm{SR}(G) \cap \mathbb{Z}^{V+E} = \mathrm{ML}(G) \cap \mathbb{Z}^{V+E}$.

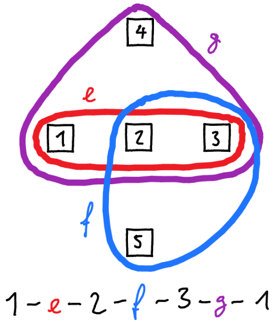
Theorem – Perfect formulation [Del Pia, Khajavirad '16; Buchheim, Crama, Rodríguez-Heck '16]

$\mathrm{SR}(G) = \mathrm{ML}(G)$ holds if and only if G is **Berge-acyclic**.

Berge cycle:

$v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_1$ with:

- $v_i \in V$ are distinct nodes.
- $e_i \in E$ are distinct edges.
- $v_i \in e_{i-1} \cap e_i$ for each i



Proposition – Standard relaxation [Fortet '60; Glover, Woolsey '74]

Let $G = (V, E)$ be a hypergraph. The polytope $\text{SR}(G)$ defined by

$$0 \leq y_e \leq x_v \leq 1 \qquad \forall v \in e \in E \qquad \text{(1a)}$$

$$y_e + \sum_{v \in e} (1 - x_v) \geq 1 \qquad \forall e \in E \qquad \text{(1b)}$$

yields an IP formulation, i.e., $\text{SR}(G) \cap \mathbb{Z}^{V+E} = \text{ML}(G) \cap \mathbb{Z}^{V+E}$.

Theorem – Perfect formulation [Del Pia, Khajavirad '16; Buchheim, Crama, Rodríguez-Heck '16]

$\text{SR}(G) = \text{ML}(G)$ holds if and only if G is **Berge-acyclic**.

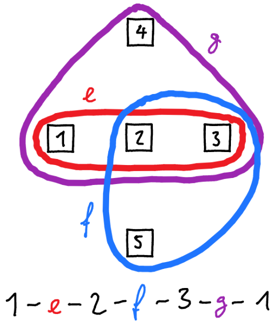
Computational experience:

- Bounds obtained from $\text{SR}(G)$ are often very weak.
  [Luedtke, Namazifar, Linderoth '12; Bao, Khajavirad, Sahinidis, Tawarmalani '14]

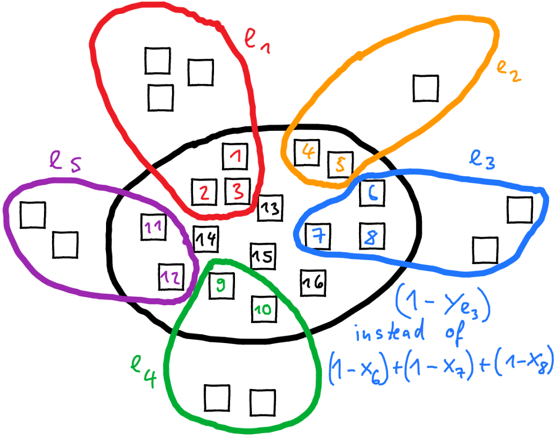
Berge cycle:

$v_1, e_1, v_2, e_2, \ldots, v_k, e_k, v_1$ with:

- $v_i \in V$ are distinct nodes.
- $e_i \in E$ are distinct edges.
- $v_i \in e_{i-1} \cap e_i$ for each i



Flower Relaxation



Definition – Flower relaxation [Del Pia, Khajavirad '18]

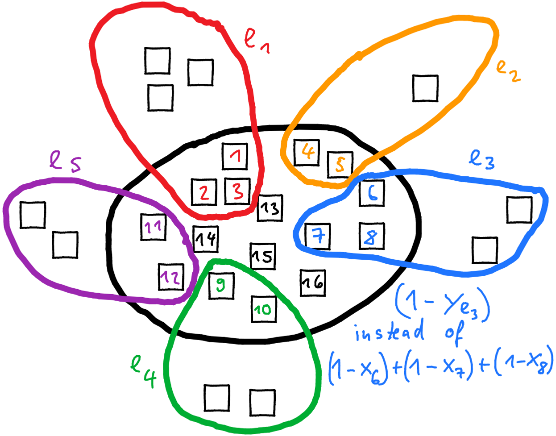
The k-**flower inequality centered at** f **with neighbors** $e_1, e_2, \ldots, e_k$ is

$$y_f + \sum_{i=1}^{k}(1 - y_{e_i}) + \sum_{v \in R}(1 - x_v) \geq 1, \qquad (2)$$

where $R := f \setminus \bigcup_{i=1}^{k} e_i$ contains all nodes of v that are not in a leaf. We denote by FR(G) the standard relaxation SR(G), augmented by all 1-flower and all 2-flower inequalities.

For comparison: $y_f + \sum_{v \in f}(1 - x_v) \geq 1 \qquad (1b)$

Flower Relaxation



Definition – Flower relaxation [Del Pia, Khajavirad '18]

The k-**flower inequality centered at** f **with neighbors** $e_1, e_2, \dots, e_k$ is

$$y_f + \sum_{i=1}^{k} (1 - y_{e_i}) + \sum_{v \in R} (1 - x_v) \geq 1, \qquad (2)$$

where $R := f \setminus \bigcup_{i=1}^{k} e_i$ contains all nodes of v that are not in a leaf. We denote by $\text{FR}(G)$ the standard relaxation $\text{SR}(G)$, augmented by all 1-flower and all 2-flower inequalities.

For comparison: $y_f + \sum_{v \in f} (1 - x_v) \geq 1 \qquad (1b)$

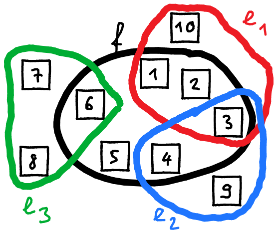
Special case:

- 1-flower inequalities were independently introduced as **2-link inequalities**. [Crama, Rodríguez-Heck '17]

Flower vs. Recursive McCormick Relaxations

Extended flower inequalities:

- Inequality is valid even if leaves overlap!

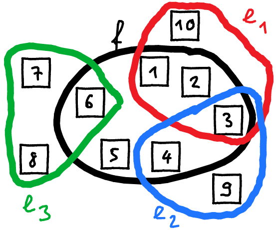


$$y_f + (1 - y_{e_1}) + (1 - y_{e_2}) + (1 - y_{e_3}) + (1 - x_5) \geq 1$$

Flower vs. Recursive McCormick Relaxations

Extended flower inequalities:

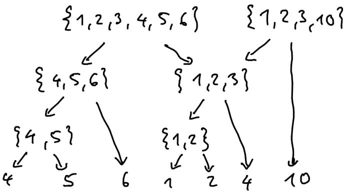
- Inequality is valid even if leaves overlap!



$$y_f + (1 - y_{e_1}) + (1 - y_{e_2}) + (1 - y_{e_3}) + (1 - x_5) \geq 1$$

Recursive McCormick formulations:

- Add auxiliary variables y_I for intermediate sets I.



$$y_{I \cup J} = y_I \cdot y_J$$

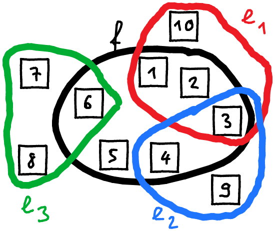
Linearization:

$$y_{I \cup J} \leq y_I$$
$$y_{I \cup J} \leq y_J$$
$$y_{I \cup J} \geq y_I + y_J - 1$$
$$y_{I \cup J} \geq 0$$

$$y_{\{1,2,3,4,5,6\}} = y_{\{4,5,6\}} \cdot y_{\{1,2,3\}}$$
$$y_{\{1,2,3\}} = y_{\{1,2\}} \cdot x_3$$
$$y_{\{1,2\}} = x_1 \cdot x_2$$

Extended flower inequalities:

- Inequality is valid even if leaves overlap!



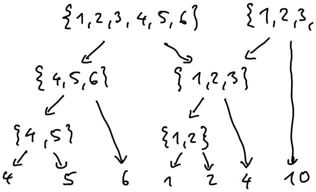
$$y_f + (1 - y_{e_1}) + (1 - y_{e_2}) + (1 - y_{e_3}) + (1 - x_5) \geq 1$$

Theorem – Flower power [Khajavirad '23]

The **extended flower relaxation** is at least as strong as any **recursive McCormick relaxation**.

Recursive McCormick formulations:

- Add auxiliary variables y_I for intermediate sets I.



$$y_{I \cup J} = y_I \cdot y_J$$

Linearization:

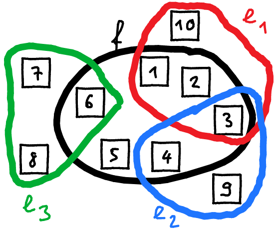
$$y_{I \cup J} \leq y_I$$
$$y_{I \cup J} \leq y_J$$
$$y_{I \cup J} \geq y_I + y_J - 1$$
$$y_{I \cup J} \geq 0$$

$$y_{\{1,2,3,4,5,6\}} = y_{\{4,5,6\}} \cdot y_{\{1,2,3\}}$$
$$y_{\{1,2,3\}} = y_{\{1,2\}} \cdot x_3$$
$$y_{\{1,2\}} = x_1 \cdot x_2$$

Flower vs. Recursive McCormick Relaxations

Extended flower inequalities:

- Inequality is valid even if leaves overlap!



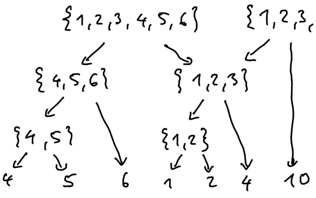
$$y_f + (1 - y_{e_1}) + (1 - y_{e_2}) + (1 - y_{e_3}) + (1 - x_5) \geq 1$$

Theorem – Flower power [Khajavirad '23]

The **extended flower relaxation** is at least as strong as any **recursive McCormick relaxation**.

Recursive McCormick formulations:

- Add auxiliary variables y_I for intermediate sets I.



$$y_{I \cup J} = y_I \cdot y_J$$

Linearization:

$$y_{I \cup J} \leq y_I$$
$$y_{I \cup J} \leq y_J$$
$$y_{I \cup J} \geq y_I + y_J - 1$$
$$y_{I \cup J} \geq 0$$

$$y_{\{1,2,3,4,5,6\}} = y_{\{4,5,6\}} \cdot y_{\{1,2,3\}}$$
$$y_{\{1,2,3\}} = y_{\{1,2\}} \cdot x_3$$
$$y_{\{1,2\}} = x_1 \cdot x_2$$

Theorem – McCormick strikes back [Schutte, Walter '24]

The **extended flower relaxation** is equal to the intersection of all **recursive McCormick relaxations**.

Faster Separation of Flower Inequalities

A useful parameter:

- By $r \in \mathbb{N}$ we denote the **rank** of G, defined as the maximum size of an edge.

Theorem – Separation via Matching **[Del Pia, Khajavirad '18]**

For γ-acyclic hypergraphs, the separation problem can be solved in time $\mathcal{O}(r|E|^2(|V|+|E|))$.

A useful parameter:

- By $r \in \mathbb{N}$ we denote the **rank** of G, defined as the maximum size of an edge.

Theorem – Separation via Matching **[Del Pia, Khajavirad '18]**

For γ-acyclic hypergraphs, the separation problem can be solved in time $\mathcal{O}(r|E|^2(|V| + |E|))$.

Theorem – Easy **[Del Pia, Khajavirad, Sahinidis '20]**

For fixed r, separation can be solved in time $\mathcal{O}(|E|^2)$.

Faster Separation of Flower Inequalities

A useful parameter:

- By $r \in \mathbb{N}$ we denote the **rank** of G, defined as the maximum size of an edge.

Theorem – Separation via Matching [Del Pia, Khajavirad '18]

For γ-acyclic hypergraphs, the separation problem can be solved in time $\mathcal{O}(r|E|^2(|V| + |E|))$.

Theorem – Easy [Del Pia, Khajavirad, Sahinidis '20]

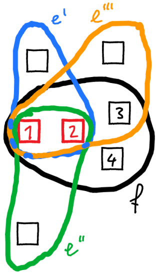
For fixed r, separation can be solved in time $\mathcal{O}(|E|^2)$.

Proof idea:

- For each center edge f, enumerate sets of neighbors.

Recap: k-flower inequality:

$$y_f + \sum_{i=1}^{k} (1 - y_{e_i}) + \sum_{v \in R} (1 - x_v) \geq 1$$



A useful parameter:

- By $r \in \mathbb{N}$ we denote the **rank** of G, defined as the maximum size of an edge.

Theorem – Separation via Matching **[Del Pia, Khajavirad '18]**

For γ-acyclic hypergraphs, the separation problem can be solved in time $\mathcal{O}(r|E|^2(|V| + |E|))$.

Theorem – Easy **[Del Pia, Khajavirad, Sahinidis '20]**

For fixed r, separation can be solved in time $\mathcal{O}(|E|^2)$.

Proof idea:

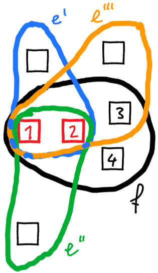
- For each center edge f, enumerate sets of neighbors.

Theorem – New algorithm **[Del Pia, Walter '24+]**

For fixed r, the separation problem can be solved in time $\mathcal{O}(|E|)$.

Recap: k-flower inequality:

$$y_f + \sum_{i=1}^{k} (1 - y_{e_i}) + \sum_{v \in R} (1 - x_v) \geq 1$$



Faster Separation of Flower Inequalities

A useful parameter:

- By $r \in \mathbb{N}$ we denote the **rank** of G, defined as the maximum size of an edge.

Theorem – Separation via Matching **[Del Pia, Khajavirad '18]**

For γ-acyclic hypergraphs, the separation problem can be solved in time $\mathcal{O}(r|E|^2(|V| + |E|))$.

Theorem – Easy **[Del Pia, Khajavirad, Sahinidis '20]**

For fixed r, separation can be solved in time $\mathcal{O}(|E|^2)$.

Proof idea:

- For each center edge f, enumerate sets of neighbors.

Theorem – New algorithm **[Del Pia, Walter '24+]**

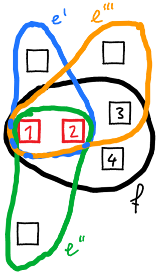
For fixed r, the separation problem can be solved in time $\mathcal{O}(|E|)$.

Proof idea:

- Per **overlap set** $U \coloneqq e \cap f$, store $\min\{(1 - y_e) \mid e \supseteq U\}$.
- For each center edge f, enumerate contained overlap sets. Minimizing edges are the neighbors.

Recap: k-flower inequality:

$$y_f + \sum_{i=1}^{k} (1 - y_{e_i}) + \sum_{v \in R} (1 - x_v) \geq 1$$



Definition – Odd β-cycle inequalities **[Del Pia, Di Gregorio '19]**

Definition 2. Consider a hypergraph $G = (V, E)$, let $C = v_1, e_1, v_2, \dots, v_m, e_m, v_1$ be a β-cycle in G, and let E^-, E^+ be a partition of $E(C)$ such that $k := |E^-|$ is odd and $e_1 \in E^-$. Let $D := \{e_{p+1}, e_{p+2}, \dots, e_m\}$, where e_p is the last edge in C that belongs to E^-. We denote by $f_1, \dots, f_k$ the subsequence of $e_1, \dots, e_m$ of the edges in E^-. Let $S_1 := (\cup_{e \in E^-} e) \setminus \bigcup_{e \in E^+} e$ and $S_2 := V(C) \setminus \bigcup_{e \in E^-} e$. With this notation in place, we make the following assumptions:

(a) Every node $v \in \bigcup_{i=1}^m e_i$ is contained in at most two edges among $e_1, \dots, e_m$.
(b) For every edge $e_i \in E^+ \setminus D$, every edge in E^- adjacent to e_i (if any) is either e_{i-1} or e_{i+1}.
(c) No edge in D is adjacent to an edge f_i with i even.
(d) At least one of the following two conditions holds:
(d1) For every $v \in S_1$, either v is contained in just one edge $e \in E^-$, or it is contained in two edges f_i, f_j with i odd and j even.
(d2) For every $e' \in E^-$ and $e'' \in D$ such that $e' \cap e'' \neq \emptyset$, then either $e' = e_1$, $e'' = e_m$ or $e' = e_p$, $e'' = e_{p+1}$.

We then define the odd β-cycle inequality corresponding to C and E^- as

$$\sum_{v \in S_1} z_v - \sum_{e \in E^-} z_e - \sum_{v \in S_2} z_v + \sum_{e \in E^+} z_e \leq |S_1| - |\{i \in \{1, \dots, m\} : e_i, e_{i+1} \in E^-\}| + \left\lfloor \frac{k}{2} \right\rfloor. \tag{2}$$

Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities [Del Pia, Di Gregorio '19]

Definition 2. Consider a hypergraph $G = (V, E)$, let $C = v_1, e_1, v_2, \ldots, v_m, e_m, v_1$ be a β-cycle in G, and let E^-, E^+ be a partition of $E(C)$ such that $k := |E^-|$ is odd and $e_1 \in E^-$. Let $D := \{e_{p+1}, e_{p+2}, \ldots, e_m\}$, where e_p is the last edge in C that belongs to E^-. We denote by $f_1, \ldots, f_k$ the subsequence of $e_1, \ldots, e_m$ of the edges in E^-. Let $S_1 := (\cup_{e \in E^-} e) \setminus \bigcup_{e \in E^+} e$ and $S_2 := V(C) \setminus \bigcup_{e \in E^-} e$. With this notation in place, we make the following assumptions:

(a) Every node $v \in \bigcup_{i=1}^m e_i$ is contained in at most two edges among $e_1, \ldots, e_m$.

(b) For every edge $e_i \in E^+ \setminus D$, every edge in E^- adjacent to e_i (if any) is either e_{i-1} or e_{i+1}.

(c) No edge in D is adjacent to an edge f_i with i even.

(d) At least one of the following two conditions holds:

(d1) For every $v \in S_1$, either v is contained in just one edge $e \in E^-$, or it is contained in two edges f_i, f_j with i odd and j even.

(d2) For every $e' \in E^-$ and $e'' \in D$ such that $e' \cap e'' \neq \emptyset$, then either $e' = e_1$, $e'' = e_m$ or $e' = e_p$, $e'' = e_{p+1}$.

We then define the odd β-cycle inequality corresponding to C and E^- as

$$\sum_{v \in S_1} z_v - \sum_{e \in E^-} z_e - \sum_{v \in S_2} z_v + \sum_{e \in E^+} z_e \leq |S_1| - |\{i \in \{1, \ldots, m\} : e_i, e_{i+1} \in E^-\}| + \left\lfloor \frac{k}{2} \right\rfloor. \quad (2)$$

Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities [Del Pia, Di Gregorio '19]

Definition 2. Consider a hypergraph $G = (V, E)$, let $C = v_1, e_1, v_2, \ldots, v_m, e_m, v_1$ be a β-cycle in G, and let E^-, E^+ be a partition of $E(C)$ such that $k := |E^-|$ is odd and $e_1 \in E^-$. Let $D := \{e_{p+1}, e_{p+2}, \ldots, e_m\}$, where e_p is the last edge in C that belongs to E^-. We denote by $f_1, \ldots, f_k$ the subsequence of $e_1, \ldots, e_m$ of the edges in E^-. Let $S_1 := (\cup_{e \in E^-} e) \setminus \bigcup_{e \in E^+} e$ and $S_2 := V(C) \setminus \bigcup_{e \in E^-} e$. With this notation in place, we make the following assumptions:

(a) Every node $v \in \bigcup_{i=1}^m e_i$ is contained in at most two edges among $e_1, \ldots, e_m$.

(b) For every edge $e_i \in E^+ \setminus D$, every edge in E^- adjacent to e_i (if any) is either e_{i-1} or e_{i+1}.

(c) No edge in D is adjacent to an edge f_i with i even.

(d) At least one of the following two conditions holds:

(d1) For every $v \in S_1$, either v is contained in just one edge $e \in E^-$, or it is contained in two edges f_i, f_j with i odd and j even.

(d2) For every $e' \in E^-$ and $e'' \in D$ such that $e' \cap e'' \neq \emptyset$, then either $e' = e_1$, $e'' = e_m$ or $e' = e_p$, $e'' = e_{p+1}$.

We then define the odd β-cycle inequality corresponding to C and E^- as

$$\sum_{v \in S_1} z_v - \sum_{e \in E^-} z_e - \sum_{v \in S_2} z_v + \sum_{e \in E^+} z_e \leq |S_1| - |\{i \in \{1, \ldots, m\} : e_i, e_{i+1} \in E^-\}| + \left\lfloor \frac{k}{2} \right\rfloor. \quad (2)$$

Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities [Del Pia, Di Gregorio '19]

Definition 2. Consider a hypergraph $G = (V, E)$, let $C = v_1, e_1, v_2, \ldots, v_m, e_m, v_1$ be a β-cycle in G, and let E^-, E^+ be a partition of $E(C)$ such that $k := |E^-|$ is odd and $e_1 \in E^-$. Let $D := \{e_{p+1}, e_{p+2}, \ldots, e_m\}$, where e_p is the last edge in C that belongs to E^-. We denote by $f_1, \ldots, f_k$ the subsequence of $e_1, \ldots, e_m$ of the edges in E^-. Let $S_1 := (\cup_{e \in E^-} e) \setminus \bigcup_{e \in E^+} e$ and $S_2 := V(C) \setminus \bigcup_{e \in E^-} e$. With this notation in place, we make the following assumptions:

(a) Every node $v \in \bigcup_{i=1}^m e_i$ is contained in at most two edges among $e_1, \ldots, e_m$.

(b) For every edge $e_i \in E^+ \setminus D$, every edge in E^- adjacent to e_i (if any) is either e_{i-1} or e_{i+1}.

(c) No edge in D is adjacent to an edge f_i with i even.

(d) At least one of the following two conditions holds:

(d1) For every $v \in S_1$, either v is contained in just one edge $e \in E^-$, or it is contained in two edges f_i, f_j with i odd and j even.

(d2) For every $e' \in E^-$ and $e'' \in D$ such that $e' \cap e'' \neq \emptyset$, then either $e' = e_1$, $e'' = e_m$ or $e' = e_p$, $e'' = e_{p+1}$.

We then define the odd β-cycle inequality corresponding to C and E^- as

$$\sum_{v \in S_1} z_v - \sum_{e \in E^-} z_e - \sum_{v \in S_2} z_v + \sum_{e \in E^+} z_e \leq |S_1| - |\{i \in \{1, \ldots, m\} : e_i, e_{i+1} \in E^-\}| + \left\lfloor \frac{k}{2} \right\rfloor. \quad (2)$$

Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities **[Del Pia, Di Gregorio '19]**

Let $G = (V, E)$ be a hypergraph. If there is a β-cycle C with a certain edge bipartition and some extra definitions satisfying some extra properties, then

$$\langle \text{some inequality with complicated coefficients and complicated right-hand side} \rangle$$

is called **odd β-cycle inequality** and valid for $\text{ML}(G)$.

Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities **[Del Pia, Di Gregorio '19]**

Let $G = (V, E)$ be a hypergraph. If there is a β-cycle C with a certain edge bipartition and some extra definitions satisfying some extra properties, then

$$\langle \text{some inequality with complicated coefficients and complicated right-hand side} \rangle$$

is called **odd β-cycle inequality** and valid for $\text{ML}(G)$.

Theorem – CG rank **[Del Pia, Di Gregorio '19]**

Odd β-cycle inequalities have Chvátal rank 2 (w.r.t. SR).

Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities [Del Pia, Di Gregorio '19]

Let $G = (V, E)$ be a hypergraph. If there is a β-cycle C with a certain edge bipartition and some extra definitions satisfying some extra properties, then

$$\langle \text{some inequality with complicated coefficients and complicated right-hand side} \rangle$$

is called **odd β-cycle inequality** and valid for $\text{ML}(G)$.

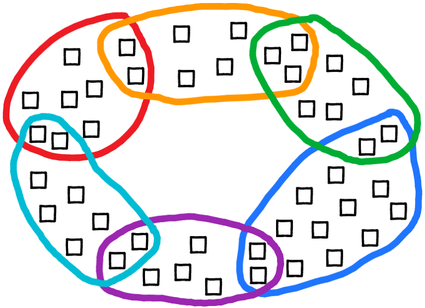
Theorem – CG rank [Del Pia, Di Gregorio '19]

Odd β-cycle inequalities have Chvátal rank 2 (w.r.t. SR).

Theorem – Perfection [Del Pia, Di Gregorio '19]

For **cyclic hypergraphs** G, $\text{ML}(G)$ is completely described by $\text{FR}(G)$ plus odd β-cycle inequalities.

Cyclic hypergraph:



Odd β-Cycle Inequalities

Definition – Odd β-cycle inequalities **[Del Pia, Di Gregorio '19]**

Let $G = (V, E)$ be a hypergraph. If there is a β-cycle C with a certain edge bipartition and some extra definitions satisfying some extra properties, then

$$\langle\text{some inequality with complicated coefficients and complicated right-hand side}\rangle$$

is called **odd β-cycle inequality** and valid for $\text{ML}(G)$.

Theorem – CG rank **[Del Pia, Di Gregorio '19]**

Odd β-cycle inequalities have Chvátal rank 2 (w.r.t. SR).

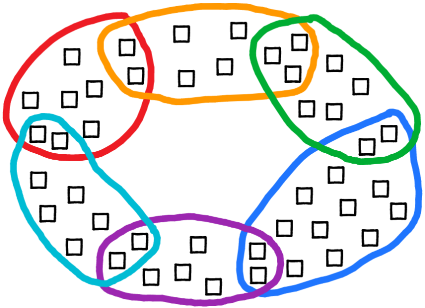
Theorem – Perfection **[Del Pia, Di Gregorio '19]**

For **cyclic hypergraphs** G, $\text{ML}(G)$ is completely described by $\text{FR}(G)$ plus odd β-cycle inequalities.

Theorem – Separation **[Del Pia, Di Gregorio '19]**

For **cyclic hypergraphs**, separation of odd β-cycle inequalities can be done in polynomial time.

Cyclic hypergraph:



Simple / Relaxed Odd β-Cycle Inequalities

Definitions – Simple / Relaxed Odd β-cycle inequalities **[Del Pia, Walter '22/24+]**

Also skipped in this talk!

Remark:

- Both new definitions yields weaker inequalities in general!

Theorem – CG rank **[Del Pia, Walter '22/24+]**

Simple / relaxed odd β-cycle inequalities have Chvátal rank 2 (with respect to the standard relaxation SR).

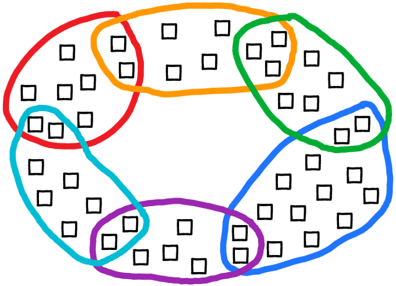
Theorem – Perfection **[Del Pia, Walter '22/24+]**

For **cyclic hypergraphs** G, $\text{ML}(G)$ is completely described by $\text{FR}(G)$ plus **simple / relaxed** odd β-cycle inequalities.

Theorem – Separation **[Del Pia, Walter '22/24+]**

For **arbitrary hypergraphs**, separation of **simple / relaxed** odd β-cycle inequalities can done in polynomial time.

Cyclic hypergraph:



Simple vs. Relaxed Odd β-Cycle Inequalities

Common idea:

- Patch together 2-flower, 1-flower and standard inequalities in cyclic fashion.
- Some coefficients in overlap cancel out, some add up.
- Scale by $1/2$ and round up the resulting inequality à la Chvátal.

Simple vs. Relaxed Odd β-Cycle Inequalities

Common idea:

- Patch together 2-flower, 1-flower and standard inequalities in cyclic fashion.
- Some coefficients in overlap cancel out, some add up.
- Scale by 1/2 and round up the resulting inequality à la Chvátal.

Separation algorithm for simple:

- Need to consider all edge pairs as nodes in auxiliary graph and run minimum-weight odd cycle algorithm.
- Running time: $\mathcal{O}(|E|^5 + |V|^2|E|)$

Simple vs. Relaxed Odd β-Cycle Inequalities

Common idea:

- Patch together 2-flower, 1-flower and standard inequalities in cyclic fashion.
- Some coefficients in overlap cancel out, some add up.
- Scale by 1/2 and round up the resulting inequality à la Chvátal.

Separation algorithm for simple:

- Need to consider all edge pairs as nodes in auxiliary graph and run minimum-weight odd cycle algorithm.
- Running time: $\mathcal{O}(|E|^5 + |V|^2|E|)$

Separation algorithm for relaxed:

- Need to consider overlap sets as nodes in auxiliary digraph and run minimum-weight odd cycle algorithm.
- Running time for fixed rank: $\mathcal{O}(|E|^2 \log |E|)$

Computational Results – Implementation

Implementation:

- SCIP plugin `sepa_multilinear`, inspecting all AND constraints:
- 1- and 2-flowers will be in SCIP 10.
- Simple/relaxed odd β-cycles are not mature, yet.

Computational Results – Implementation

Implementation:

- SCIP plugin `sepa_multilinear`, inspecting all AND constraints:
- 1- and 2-flowers will be in SCIP 10.
- Simple/relaxed odd β-cycles are not mature, yet.

Experiments:

- Time limit: 3600 s
- Default settings of SCIP, i.e., including other cutting planes, heuristics, presolve, etc..

Computational Results – Implementation

Implementation:

- SCIP plugin `sepa_multilinear`, inspecting all AND constraints:
- 1- and 2-flowers will be in SCIP 10.
- Simple/relaxed odd β-cycles are not mature, yet.

Experiments:

- Time limit: 3600 s
- Default settings of SCIP, i.e., including other cutting planes, heuristics, presolve, etc..

First insights:

- Fast 1- and 2-flower separation is orders of magnitude faster, so results for slow are skipped.
- Number of overlaps typically lies in ballpark of $|E|$.
- Odd β-cycles are turned off if auxiliary (di)graph is too large.

Low Autocorrelated Binary Sequences

Problem: [used by Del Pia, Di Gregorio '21]

$$\min_{x \in \{-1,1\}^N} \sum_{i=1}^{N-R+1} \sum_{d=1}^{R-1} \left(\sum_{j=i}^{i+R-1-d} x_j x_{j+d} \right)^2 \quad \leftrightsquigarrow \quad \text{nonnegative degree-4 polynomial}$$

Parameters:

- $N \in \{10, 15, 20, \ldots, 50\}$ and $R \in \{\lceil \frac{k}{8} N \rceil : k = 1, 2, \ldots, 8\}$ ($9 \cdot 8 = 72$ instances)

	#solved	Closed gap mean	Running time geo.mean	Sepa time s.geo.mean	B&B nodes s.geo.mean
Old default (no k-flowers)	28	75 %	503.58 s	–	2026.39
1-flowers, no 2-flowers	30	90 %	410.96 s	2.15 s	640.28
New default (1- and 2-flowers)	29	89 %	419.40 s	3.15 s	651.42
Simple oβc's	26	84 %	[6 fails]	25.71 s	[6 fails]
Simple oβc's, delayed	27	88 %	550.20 s	116.67 s	269.37
Simple oβc's, delayed, only root	27	89 %	501.38 s	60.79 s	398.44
Rlxd oβc's	29	85 %	467.81 s	145.70 s	262.78
Rlxd oβc's, delayed	29	89 %	428.86 s	52.67 s	576.51
Rlxd oβc's, delayed, only root	30	89 %	417.87 s	14.61 s	619.18
Both oβc's, delayed	26	87 %	573.21 s	268.15 s	157.41

Unconstrained Random High-Degree

Problem: [used by Crama, Rodríguez-Heck '17]

N binaries, M monomials, degree $d \in \{m, m+1, m+2, \dots\}$ with probability $\approx 2^{m-d-1}$, coefs in $\mathcal{U}\{-10, 10\}$

Parameters:

- 5 instances per $N \in \{50, 100, 200\}$, $M \in \{5N, 10N, 20N\}$ and $m \in \{2, 3, 4\}$ $(5 \cdot 3 \cdot 3 \cdot 3 = 135$ instances)

	#solved	Closed gap mean	Running time geo.mean	Sepa time s.geo.mean	B&B nodes s.geo.mean
Old default (no k-flowers)	53	0.66 %	1041.98 s	–	4411.10
1-flowers, no 2-flowers	52	71 %	1011.22 s	0.59 s	2375.78
New default (1- and 2-flowers)	52	72 %	1027.10 s	1.76 s	1927.02
Simple oβc's	53	77 %	1206.02 s	369.70 s	145.70
Simple oβc's, delayed	54	78 %	1002.96 s	145.11 s	475.23
Simple oβc's, delayed, only root	52	78 %	982.39 s	96.42 s	435.67
Rlxd oβc's	50	70 %	1454.68 s	558.20 s	343.25
Rlxd oβc's, delayed	51	71 %	1220.32 s	190.24 s	1399.01
Rlxd oβc's, delayed, only root	51	72 %	1127.21 s	42.69 s	1629.70
Both oβc's, delayed	54	78 %	1067.71 s	240.03 s	401.20

Image Restoration

Problem: [used by Crama, Rodríguez-Heck '17]

Image restoration on w-by-h black/white image of which p pixels were perturbed; penalty per 2-by-2 patch

Parameters:

- 2 instances per $(w, h) \in \{(10,10), (10,15), (15,15), (15,20), (20,20), (20,25), (25,25),$ $p \in \{0, 5\,\%, 50\,\%\}$, three motifs (105 distinct instances)

	#solved	Closed gap mean	Running time geo.mean	Sepa time s.geo.mean	B&B nodes s.geo.mean
Old default (no k-flowers)	37	77 %	1596.01 s	–	1393.43
1-flowers, no 2-flowers	105	100 %	91.55 s	0.06 s	50.40
New default (1- and 2-flowers)	99	99 %	183.94 s	0.11 s	158.41
Simple oβc's	?	?	?	?	
Simple oβc's, delayed	101	99 %	74.58 s	7.56 s	5.96
Simple oβc's, delayed, only root	?	?	?	?	
Rlxd oβc's	78	92 %	328.84 s	133.20 s	150.36
Rlxd oβc's, delayed	97	99 %	220.01 s	19.30 s	148.93
Rlxd oβc's, delayed, only root	97	99 %	217.95 s	14.07 s	149.93
Both oβc's, delayed	?	?	?	?	

Image Restoration

Problem: [used by Crama, Rodríguez-Heck '17]

Image restoration on w-by-h black/white image of which p pixels were perturbed; penalty per 2-by-2 patch

Parameters:

- 2 instances per $(w, h) \in \{(10,10), (10,15), (15,15), (15,20), (20,20), (20,25), (25,25),$ $p \in \{0, 5\,\%, 50\,\%\}$, three motifs (105 distinct instances)

	#solved	**Closed gap mean**	**Running time geo.mean**
Old default (no k-flowers)	37	77 %	1596.01 s
1-flowers, no 2-flowers	105	100 %	91.55 s
New default (1- and 2-flowers)	99	99 %	183.94 s
Simple oβc's	?	?	?
Simple oβc's, delayed	101	99 %	74.58 s
Simple oβc's, delayed, only root	?	?	?
Rlxd oβc's	78	92 %	328.84 s
Rlxd oβc's, delayed	97	99 %	220.01 s
Rlxd oβc's, delayed, only root	97	99 %	217.95 s
Both oβc's, delayed	?	?	?

Conclusions

Insights:

- Flower inequalities are now really cheap and effective.
- Separation problem for relaxed odd β-cycle inequalities is faster than for simple, but only in theory.

Conclusions

Insights:

- Flower inequalities are now really cheap and effective.
- Separation problem for relaxed odd β-cycle inequalities is faster than for simple, but only in theory.

Future work:

- Strengthen cuts that act on one edge and its neighbors further!
- Speed up odd β-cycle cut generation (incl. underlying minium odd cycle search)!
- Strengthen odd β-cycle cuts (lift coefficients of nodes)!
- Integrate into SCIP (if it pays off)?!

Conclusions

Insights:

- Flower inequalities are now really cheap and effective.
- Separation problem for relaxed odd β-cycle inequalities is faster than for simple, but only in theory.

Future work:

- Strengthen cuts that act on one edge and its neighbors further!
- Speed up odd β-cycle cut generation (incl. underlying minium odd cycle search)!
- Strengthen odd β-cycle cuts (lift coefficients of nodes)!
- Integrate into SCIP (if it pays off)?!

Thank you! – Questions?