\( \nu \)-polyhedral disjunctive cuts

Aleksandr M. Kazachkov

Based on joint work with Egon Balas
New cutting plane method for mixed-integer linear programming

Maximize something good subject to constraints*

*Linear constraints
New *cutting plane* method for mixed-integer linear programming

*Linear constraints*

Maximize *something good* subject to constraints* and integrality

*Integer points*

$Feasible integer points \ (P_1)$

$conv(P_1)$
New cutting plane method for mixed-integer linear programming

Maximize something good subject to constraints* and integrality

Generally cannot efficiently optimize over $P_1$, but can over $P$

Idea: Optimize over $P$

*Linear constraints
New cutting plane method for mixed-integer linear programming

*Linear constraints

Maximize something good subject to constraints* and integrality

Generally cannot efficiently optimize over \( P_1 \), but can over \( P \)

Idea: Optimize over \( P \) then tighten the relaxation by valid cuts
Setting: mixed-integer linear programming

Optimize over mixed-integer feasible region in $\mathbb{R}^n$

\[
\begin{align*}
(\text{IP}) & \quad \min_{x} c^T x \\
(\text{LP}) & \quad \begin{bmatrix}
Ax & \geq & b \\
x_j & \in & \mathbb{Z} \quad \text{for all } j \in \mathcal{I}
\end{bmatrix} P
\end{align*}
\]

Start with solution $\overline{x}$ to (LP), apply valid general-purpose cuts to tighten the relaxation
Cutting planes from disjunctions

We focus on valid cuts derived from disjunctions

$$\bigvee_{t \in \mathcal{T}} \{x \in \mathbb{R}^n : D^t x \geq D_0^t\}$$
Cutting planes from disjunctions

We focus on valid cuts derived from disjunctions.

\[ \bigvee_{t \in T} \{ x \in \mathbb{R}^n : D^t x \geq D_0^t \} \]

\[ x_k \leq \lfloor x_k \rfloor \]

\[ x_k \geq \lceil x_k \rceil \]
Cutting planes from disjunctions

Valid disjunction: partitions the search space such that

\[ P_I \subseteq \bigcup_{t \in T} \{ x \in P : D^t x \geq D^t_0 \} \]

\[ x_k \leq \lfloor x_k \rfloor \]

\[ x_k \geq \lceil x_k \rceil \]
Cutting planes from disjunctions

**Disjunctive cuts:** inequalities valid for the **disjunctive hull**

\[
\text{conv} \left( \bigcup_{t \in T} P^t \right)
\]

but not for \( P \)
Goals: added strength, faster solving time, better numerical properties

Existing cuts:
- Relatively simple
- Already critical to solver performance
- Require recursion to reach strong cuts
- May lead to numerical problems and "tailing off"*

Goal: Efficiently and non-recursively generate strong cuts

In one round

Non-simple disjunctive sets can lead to stronger cuts.
Existing work on “stronger cuts” (partial list)

Balas (1979) – *disjunctive programming*
Andersen, Louveaux, Weismantel, Wolsey (2007) – *sparked renewed interest*

**Simple disjunctive cuts**
Balas, Ceria, Cornuéjols (1993, 1996)
   – L&P cuts (only tested with splits)
Espinoza (2010)
Basu, Bonami, Cornuéjols, Margot (2011)x2
Balas, Margot (2013)
Balas, Qualizza (2013)
Dey, Lodi, Tramontani, Wolsey (2014)

**Non-simple disjunctive cuts**
Perregaard, Balas (2001)
Chvátal, Cook, Espinoza (2013)
Dash, Günlük, Vielma (2014)
Louveaux, Poirrier, Salvagnin (2015)

*Simple: one disjunctive inequality per term*
Generating “stronger cuts” is challenging

“Stronger cuts” often require substantially more computational effort (than Gomory cuts)

E.g., if number of axis-parallel split disjunctions is $\mathcal{O}(n)$, then the number of two-row options is $\mathcal{O}(n^2)$ (already impractical)

Number of possible cuts also grows unmanageably large

Expensive, and ultimately may not yield better results within branch-and-cut
Contributions

Development of strong, non-recursive cutting plane method and supporting theoretical results

Evaluation and investigation via computational experiments with multiterm general disjunctions and within branch-and-cut

Ongoing research on cut strengthening in our framework
Lift-and-project cuts
Lift-and-project is a commonly-used framework for generating disjunctive cuts

\[
\alpha^T x \geq \beta \text{ valid for } \text{conv}(U_{t \in T} P^t) \iff \\
\alpha^T x \geq \beta \text{ for all } x \in P^t, t \in T
\]

Cut is valid if and only if there exists a certificate of validity \(v^t\) for each \(P^t := \{x \in \mathbb{R}^n : A^tx \geq b^t\}, t \in T\)

\[
\alpha^T = v^t A^t \\
\beta \leq v^t b^t \\
v^t \geq 0
\]
Lift-and-project cuts are generated through a cut-generating linear program (CGLP)

\[
\begin{align*}
\min_{\alpha, \beta, \{v^t\}_{t \in T}} & \quad \alpha^T \bar{x} - \beta \\
\alpha^T &= v^t A^t & \text{for all } t \in \mathcal{T} \\
\beta &\leq v^t b^t & \text{for all } t \in \mathcal{T} \\
v^t &\geq 0 & \text{for all } t \in \mathcal{T} \\
+ &\text{ normalization}
\end{align*}
\]
Taking a $\mathcal{V}$-polyhedral perspective
\( \mathcal{V} \)-polyhedral cuts: a different perspective on generating disjunctive cuts

\[ \alpha^T x \geq \beta \text{ valid for } \overline{\text{conv}}(U_{t \in T} P^t) \]

\[ \iff \]

\[ \alpha^T x \geq \beta \text{ for all } x \in P^t, t \in T \]

**Lift-and-project cuts**
Cut is valid if and only if there exists a Farkas certificate \( v^t \) for each \( P^t := \{ x \in \mathbb{R}^n : A^t x \geq b^t \} \)

\[ \alpha^T = v^t A^t \]

\[ \beta \leq v^t b^t \]

\[ v^t \geq 0 \]

**\( \mathcal{V} \)-polyhedral cuts (VPCs)**
Cut is valid if and only if it is satisfied by the extreme points and rays of each \( P^t \)

\[ \alpha^T p \geq \beta \text{ for all } p \in \text{vertices}(P^t) \]

\[ \alpha^T r \geq 0 \text{ for all } r \in \text{rays}(P^t) \]

**\( \mathcal{H} \)-polyhedral description**
\[ \begin{align*}
\min_{\alpha, \beta} \quad & \alpha^T w \\
\alpha^T p & \geq \beta \quad \text{for all } p \in \mathcal{P} \\
\alpha^T r & \geq 0 \quad \text{for all } r \in \mathcal{R} 
\end{align*} \]
Barrier to using $\mathcal{V}$-polyhedral perspective is the exponential number of constraints

Issue is that the number of points and rays of $P^t$ may be exponential (in the number of inequalities)

Perregaard and Balas (2001) and Louveaux et al. (2015) use row generation to overcome this difficulty (this is expensive)

We contribute a compact formulation that directly yields valid cuts
Solve for different objectives

\[ \begin{aligned}
\min_{\alpha, \beta} & \quad \alpha^T \mathbf{w} \\
\alpha^T \mathbf{p} & \geq \beta \quad \text{for all } \mathbf{p} \in \mathcal{P} \\
\alpha^T \mathbf{r} & \geq 0 \quad \text{for all } \mathbf{r} \in \mathcal{R}
\end{aligned} \]

Choose disjunction

Point-ray linear program (PRLP)

Obtain points and rays, \((\mathcal{P}, \mathcal{R})\)
Which objectives? \[ \min_{\alpha, \beta} \alpha^T w \]

Which disjunction?
\[ \alpha^T p \geq \beta \quad \text{for all } p \in \mathcal{P} \]
\[ \alpha^T r \geq 0 \quad \text{for all } r \in \mathcal{R} \]
Which objectives?\[\begin{align*}
\min_{\alpha, \beta} & \quad \alpha^T w \\
\alpha^T p & \geq \beta \quad \text{for all } p \in \mathcal{P} \\
\alpha^T r & \geq 0 \quad \text{for all } r \in \mathcal{R}
\end{align*}\]Which disjunction? Which points rays?
Instead of, e.g., splits and crosses, expend effort to get one strong disjunction

Existing approaches generate many shallow disjunctions

Computationally expensive, difficult to target useful cuts

Idea: Generate one strong disjunction

Leaf nodes of a partial branch-and-bound tree
Which objectives?

\[
\min_{\alpha, \beta} \alpha^T w
\]

\[
\alpha^T p \geq \beta \quad \text{for all } p \in \mathcal{P}
\]

\[
\alpha^T r \geq 0 \quad \text{for all } r \in \mathcal{R}
\]

Which disjunction?

Which points/rays?
Full $\mathcal{V}$-polyhedral description is impractical

Impractical to use the complete $\mathcal{V}$-polyhedral description of each disjunctive term

**Goal:** Find a **compact** collection of points and rays such that all cuts (from PRLP) are **valid**
Theorem: Extreme ray solutions to the PRLP correspond to facets of $\text{conv}(\mathcal{P}) + \text{cone}(\mathcal{R})$.

Corollary: If $\mathcal{P}$ and $\mathcal{R}$ are sets of points and rays such that, for all $t \in \mathcal{T}$,

$$P^t \subseteq \text{conv}(\mathcal{P}) + \text{cone}(\mathcal{R}),$$

then PRLP from $(\mathcal{P}, \mathcal{R})$ yields valid VPCs.

Sufficient to use a $\mathcal{V}$-polyhedral relaxation to guarantee valid cuts.
Goal: find compact $\mathcal{V}$-polyhedral relaxation

\[
P^1 = \{ x \in P : x_k \leq \lceil \bar{x}_k \rceil \}
\]
\[
P^2 = \{ x \in P : x_k \geq \lfloor \bar{x}_k \rfloor \}
\]

**Need:**
\[
P^1 \cup P^2 \subseteq \text{conv}(P) + \text{cone}(\mathcal{R})
\]
Goal: find compact $\mathcal{V}$-polyhedral relaxation

\[ P^1 = \{ x \in P : x_k \leq \lceil \bar{x}_k \rceil \} \]
\[ P^2 = \{ x \in P : x_k \geq \lfloor \bar{x}_k \rfloor \} \]

Need:
\[ P^1 \cup P^2 \subseteq \text{conv}(P) + \text{cone}(\mathcal{R}) \]

Use LP basis cone for each disjunctive term
Goal: find compact \( V \)-polyhedral relaxation

**Need:**
\[ P^1 \cup P^2 \subseteq \text{conv}(P) + \text{cone}(R) \]

Use LP basis cone for each disjunctive term

Any **cut** valid for each of the relaxations will be valid for \( P_I \).
Goal: find compact $\mathcal{V}$-polyhedral relaxation

**Need:**

$$P^1 \cup P^2 \subseteq \text{conv}(\mathcal{P}) + \text{cone}(\mathcal{R})$$

Use LP basis cone for each disjunctive term

Any **cut** valid for each of the relaxations will be valid for $P_I$
Goal: find compact \( \mathcal{V} \)-polyhedral relaxation

**Need:**
\[ P^1 \cup P^2 \subseteq \text{conv}(\mathcal{P}) + \text{cone}(\mathcal{R}) \]

Use LP basis cone for each disjunctive term

Any cut valid for each of the relaxations will be valid for \( P_I \)
Simple point-ray relaxation and resulting simple PRLP

Let $p^t \in \text{argmin}\{c^T x : x \in P^t\}$ and $C^t$ denote the associated basis cone (corresponding to a basis of $p^t$)

**Simple point-ray collection**

$(\mathcal{P}_0, \mathcal{R}_0) \subseteq \left( \bigcup_{t \in \mathcal{T}} p^t, \bigcup_{t \in \mathcal{T}} \text{rays}(C^t) \right)$

**Constraints of the simple PRLP**

$\alpha^T p^t \geq \beta$ for all $t \in \mathcal{T}$

$\alpha^T r \geq 0$ for all $r \in \text{rays}(C^t)$, $t \in \mathcal{T}$
The PRLP avoids the use of an extended formulation as in the CGLP

Cut-generating linear program for lift-and-project:*  
Constraints: \((n + 1) \cdot |\mathcal{T}|\)  
\((+ \text{ nonnegativity})\)  
Variables: \(n + (m + m_t) \cdot |\mathcal{T}|\)  
\((m_t: \# \text{ rows of } D^t x \geq D^t_0)\)

Point-ray linear program for VPCs:*  
Constraints: \(|\mathcal{P} \cup \mathcal{R}| \cdot (n + 1) \cdot |\mathcal{T}|\)  
Variables: \(n\)

VPCs offer an efficient alternative to get disjunctive cuts

*Assuming fixed \(\beta \in \{-1, +1\}\)
Surprisingly, the simple point-ray collection includes strong facets of the disjunctive hull

**Theorem:**
Suppose that the optimal basis of $p^t$ is unique for all $t \in T$

For a split disjunction, every facet of $\text{conv}(\mathcal{P}_0) + \text{cone}(\mathcal{R}_0)$ that is tight on both terms is also a facet of $P_D$

A slightly weaker version holds for general disjunctions
Which objectives?

\[
\begin{bmatrix}
\min_{\alpha, \beta} & \alpha^T w \\
\alpha^T p \geq \beta & \text{for all } p \in \mathcal{P} \\
\alpha^T r \geq 0 & \text{for all } r \in \mathcal{R}
\end{bmatrix}
\]
To get good cuts, start with good objectives

Choice of objectives $w$ for PRLP is crucial in determining the strength of the cuts obtained

Two perspectives:

- **Maximize violation** (for point not in disjunctive hull)
- **Minimize slack** (for point in disjunctive hull)
Target the disjunctive lower bound to attain the same objective value from cuts

**Idea:** Target cuts that are tight at the disjunctive optimal solution \( p \), an optimal solution to
\[
\min_{p \in P_0} c^T p = \min_{x \in \mathcal{P}_t} c^T x
\]

Yields strategy for objectives that are **structured, bounded, and likely to be distinct**

Pursues a **diverse** set of facet-defining inequalities of \( \text{conv}(\mathcal{P}_0) + \text{cone}(\mathcal{R}_0) \)
Key theoretical takeaway: framework for an effective disjunctive cut generator

$\mathcal{N}$-polyhedral perspective enables separating disjunctive cuts in the original dimension

**Compact** $\mathcal{N}$-polyhedral relaxation can be found with only $(n + 1) \cdot |\mathcal{T}|$ points and rays

Many strong disjunctive facets are already captured

Under mild conditions, all VPCs from this simple relaxation define facets of $P_D$
Computational results with VPCs
Computational setup

Evaluated effect of VPCs on percent gap closed and branch-and-bound time

Implemented cut generation in COIN-OR framework and branch-and-bound tests by adding as user cuts in Gurobi 7.5.1

195 preprocessed instances from MIPLIB, COR@L, and NEOS
# rows, # cols ≤ 5000; IP optimal value is known; partial tree does not find IP optimal solution but does close some gap
Computational setup

Disjunctions: leaf nodes of a partial branch-and-bound tree

Partial tree strategy: strong branching for variable selection, minimum objective value for node selection

Partial tree sizes: \( 2^\ell \) leaf nodes, \( \ell \in \{1, \ldots, 6\} \)

Cut limit: \# fractional integer variables at \( \bar{x} \)
Average percent gap closed (all numbers %)

<table>
<thead>
<tr>
<th>GMIC</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>17.3</td>
</tr>
</tbody>
</table>


**Average percent gap closed (all numbers %)**

<table>
<thead>
<tr>
<th></th>
<th>GMIC</th>
<th>VPC (V)</th>
<th>V+GMIC</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>17.3</td>
<td>15.6</td>
<td>27.0</td>
</tr>
</tbody>
</table>
### Average percent gap closed (all numbers %)

<table>
<thead>
<tr>
<th></th>
<th>GMIC</th>
<th>VPC (V)</th>
<th>V+GMIC</th>
<th>GurF</th>
<th>V+GurF</th>
<th>GurL</th>
<th>V+GurL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>All</strong></td>
<td>17.3</td>
<td>15.6</td>
<td>27.0</td>
<td>26.0</td>
<td>33.0</td>
<td>46.5</td>
<td>52.1</td>
</tr>
</tbody>
</table>

- Gurobi after one round of cuts at the root
- Gurobi after last round of cuts at the root
### Average percent gap closed (all numbers %)

<table>
<thead>
<tr>
<th></th>
<th>GMIC</th>
<th>VPC (V)</th>
<th>V+GMIC</th>
<th>GurF</th>
<th>V+GurF</th>
<th>GurL</th>
<th>V+GurL</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>17.3</td>
<td>15.6</td>
<td>27.0</td>
<td>26.0</td>
<td>33.0</td>
<td>46.5</td>
<td>52.1</td>
</tr>
<tr>
<td>≥10%</td>
<td>14.4</td>
<td>29.6</td>
<td>33.5</td>
<td>20.0</td>
<td>32.6</td>
<td>38.8</td>
<td>50.0</td>
</tr>
</tbody>
</table>

Instances for which VPCs close at least 10% of the integrality gap

Gurobi after one round of cuts at the root

Gurobi after last round of cuts at the root
### Branch-and-bound results [time]

<table>
<thead>
<tr>
<th>Bin</th>
<th># inst</th>
<th>Time (shifted geomean)</th>
<th>Wins</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Gurobi</td>
<td>VPC</td>
</tr>
<tr>
<td>All &lt; 3600s</td>
<td>159</td>
<td>81.5</td>
<td>63.8</td>
</tr>
<tr>
<td>&gt; 10s</td>
<td>81</td>
<td>247.7</td>
<td>180.6</td>
</tr>
<tr>
<td>&gt; 100s</td>
<td>37</td>
<td>869.7</td>
<td>652.8</td>
</tr>
<tr>
<td>&gt; 1000s</td>
<td>14</td>
<td>2156.1</td>
<td>1840.7</td>
</tr>
</tbody>
</table>

At least 10% faster solution time

Counting cut generation time
Conclusions & future research
VPCs provide a computationally tractable way to generate disjunctive cuts

\textbf{\textit{\nu}-polyhedral cuts:} computationally tractable way to generate strong disjunctive cuts that can be helpful when used with branch-and-bound and utilize \textbf{structural properties}

However, missing strength with respect to Gomory cuts: \textbf{coefficient modularization}

Our ongoing research uses polarity concepts to enable this \textbf{cut strengthening} to be applied to VPCs
Extensions and future outlook

Disjunctions from partial branch-and-bound trees: tighter integration between cutting planes and branch-and-bound, and a pathway to better understanding their interaction

VPCs provide a framework for investigating cut selection:

*Which cutting planes help most for branch-and-cut solve time?*

Other extensions: nonlinear settings
Thank you for your attention

Questions?
Additional results
VPC framework has computational advantages over lift-and-project cuts

Theoretically, all facets of the disjunctive hull can be obtained through either the lift-and-project or VPC framework.

In practice, lift-and-project cuts may not even be supporting for the disjunctive hull due to the normalization and the extended formulation.

VPCs do not suffer from this drawback, but using a relaxation will produce only a subset of the valid disjunctive inequalities.

Theorem: Cuts define facets of the convex hull of the points and rays

Given $\mathcal{P}$ and $\mathcal{R}$ (points and rays), every extreme ray $(\alpha, \beta)$ of

$$\alpha^T p \geq \beta \quad \text{for all } p \in \mathcal{P}$$
$$\alpha^T r \geq 0 \quad \text{for all } r \in \mathcal{R}$$

defines a facet $\alpha^T x \geq \beta$ of $\text{conv}(\mathcal{P}) + \text{cone}(\mathcal{R})$
Strength evaluated based on percent integrality gap closed

Let $\hat{x}$ be an optimal solution after adding cuts

Let $x^I$ be an optimal solution over $P_I$

Define the \textbf{percent integrality gap closed} as

$$100 \times \frac{c^T \hat{x} - c^T \bar{x}}{c^T x^I - c^T \bar{x}}$$
Effect of varying number leaf nodes

- $V$
- $V+G$
- $V+GurL$
- $V+GurF$
- VPC upper bound

Average % gap closed

- 0 leaves
- 2 leaves
- 4 leaves
- 8 leaves
- 16 leaves
- 32 leaves
- 64 leaves
- Best
Gurobi run with one random seed vs min of up to 7 VPC runs from different trees

Below the line is better for VPCs

Gurobi time w/VPCs (min 7 runs)

Gurobi time w/o VPCs
Min 7 Gurobi runs with different random seeds vs exactly 7 VPC runs from different trees.
Branch-and-bound results [nodes] (all 6 partial trees successfully tested)

<table>
<thead>
<tr>
<th>Bin</th>
<th># inst</th>
<th>Gurobi7</th>
<th>VPC</th>
<th>Gurobi7</th>
<th>VPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>All &lt; 3600s</td>
<td>97</td>
<td>5,588</td>
<td>5,239</td>
<td>32</td>
<td>51</td>
</tr>
<tr>
<td>&gt; 10s</td>
<td>41</td>
<td>34,449</td>
<td>31,386</td>
<td>5</td>
<td>17</td>
</tr>
<tr>
<td>&gt; 100s</td>
<td>19</td>
<td>139,998</td>
<td>135,861</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>&gt; 1000s</td>
<td>8</td>
<td>314,438</td>
<td>261,187</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Cut density increases with disjunction size and may be useful for cut selection

<table>
<thead>
<tr>
<th></th>
<th>V (2)</th>
<th>V (4)</th>
<th>V (8)</th>
<th>V (16)</th>
<th>V (32)</th>
<th>V (64)</th>
</tr>
</thead>
<tbody>
<tr>
<td># inst</td>
<td>155</td>
<td>141</td>
<td>134</td>
<td>131</td>
<td>118</td>
<td>109</td>
</tr>
<tr>
<td># wins (by time)</td>
<td>46</td>
<td>26</td>
<td>37</td>
<td>39</td>
<td>37</td>
<td>36</td>
</tr>
<tr>
<td>Avg cut density</td>
<td>0.363</td>
<td>0.371</td>
<td>0.432</td>
<td>0.491</td>
<td>0.516</td>
<td>0.525</td>
</tr>
<tr>
<td>Avg density (win)</td>
<td>0.356</td>
<td>0.316</td>
<td>0.352</td>
<td>0.435</td>
<td>0.508</td>
<td>0.496</td>
</tr>
<tr>
<td>Avg density (non-win)</td>
<td>0.366</td>
<td>0.383</td>
<td>0.462</td>
<td>0.515</td>
<td>0.520</td>
<td>0.540</td>
</tr>
</tbody>
</table>