$\mathcal{V}$-polyhedral disjunctive cuts

Aleksandr M. Kazachkov

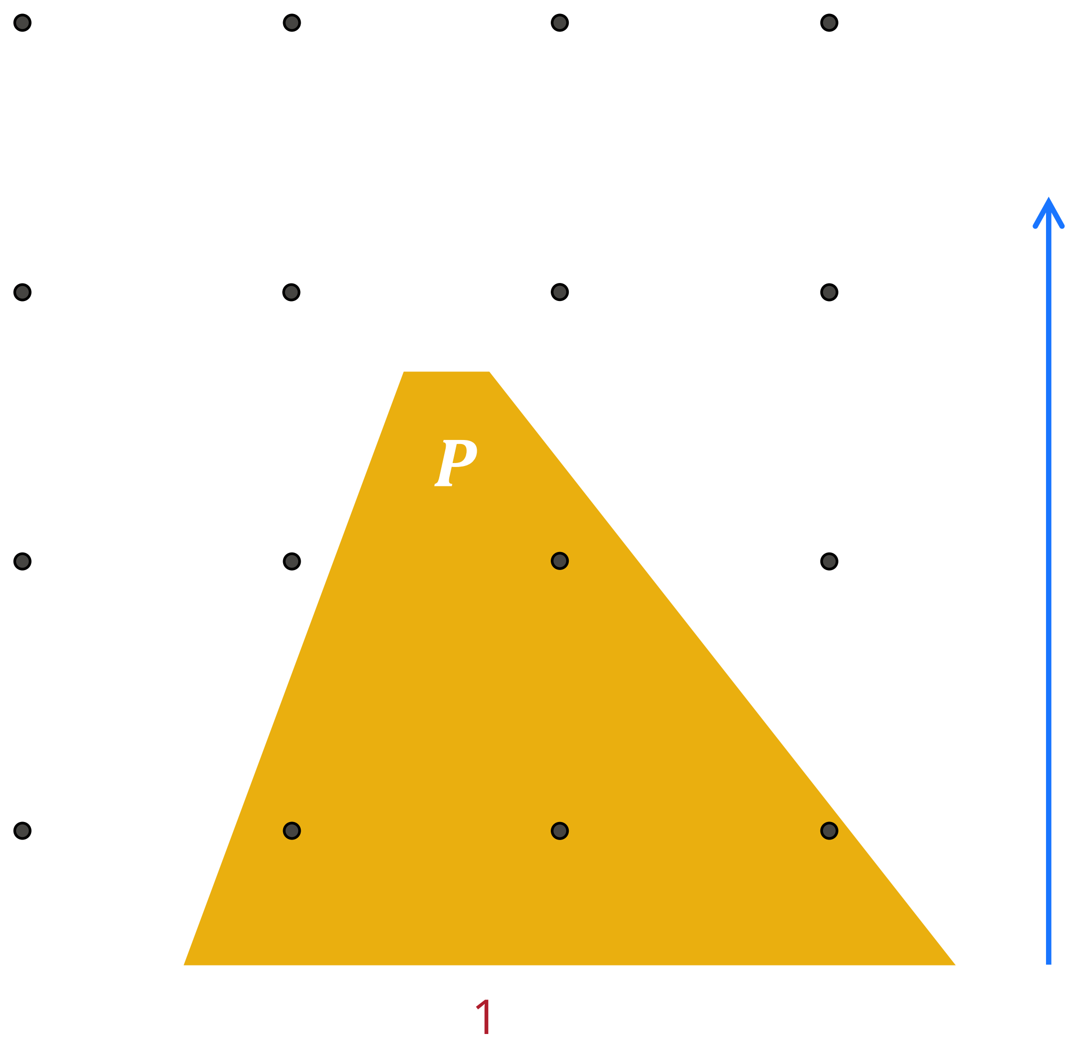
Based on joint work with Egon Balas

INFORMS 2018

Carnegie Mellon

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

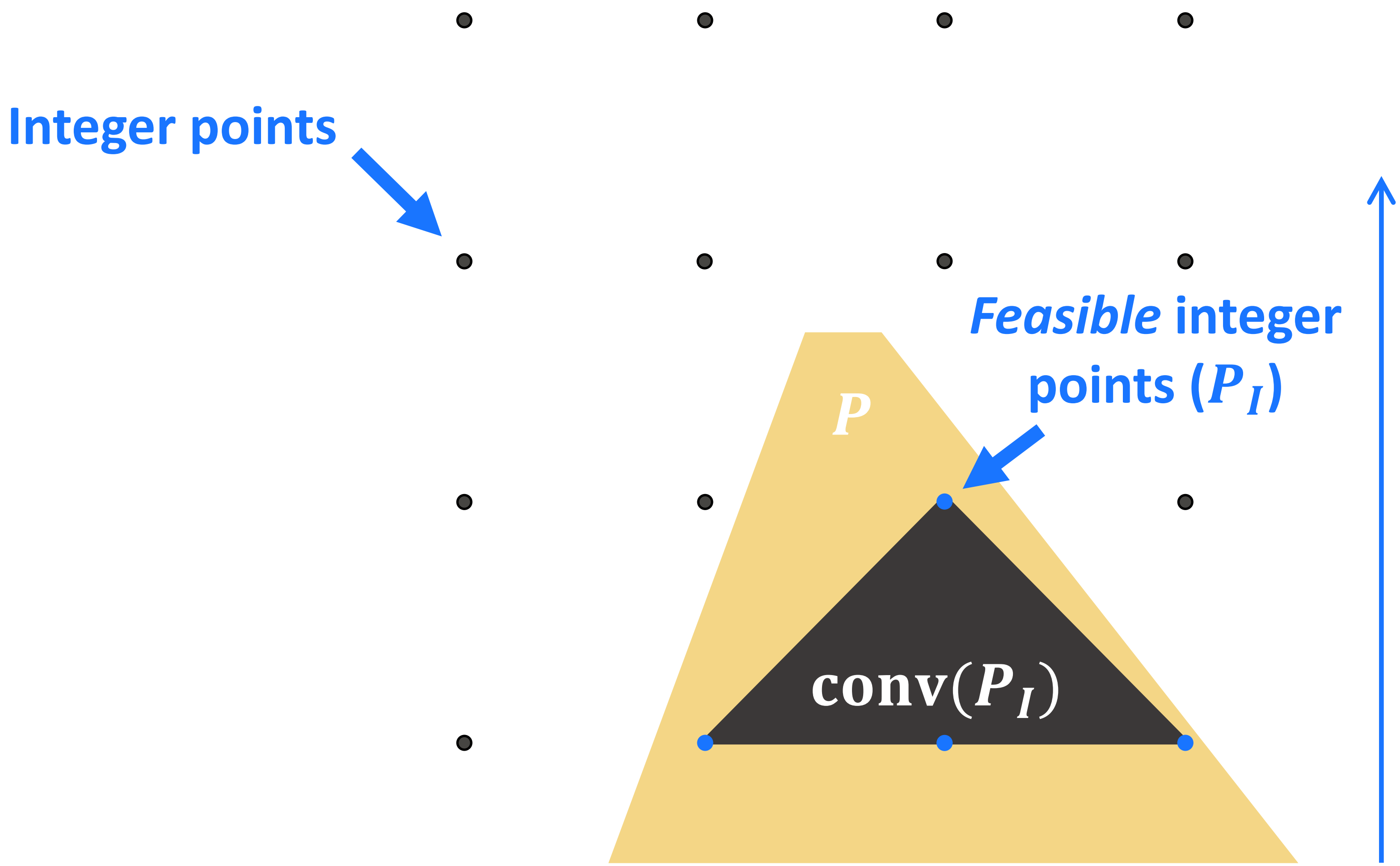
New *cutting plane* method for mixed-integer linear programming



Maximize
something good
subject to **constraints***

**Linear* constraints

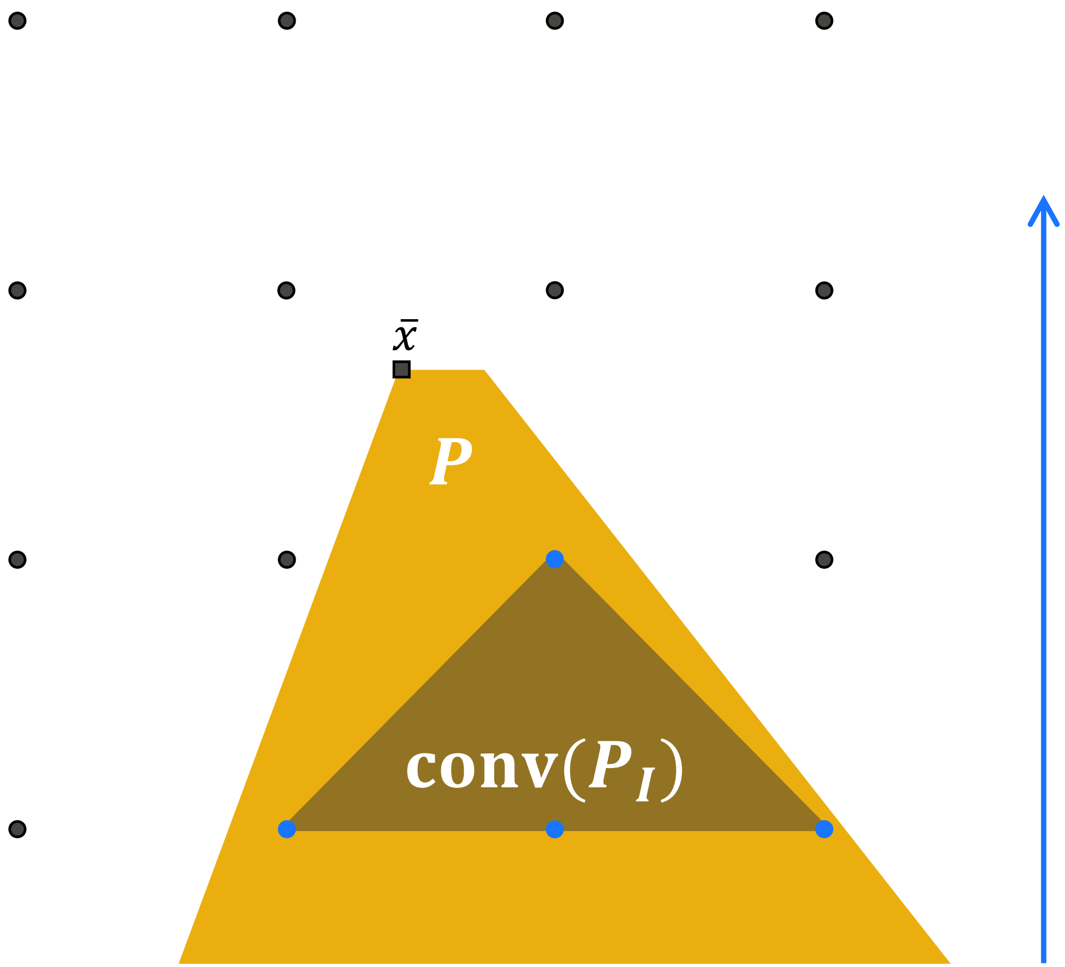
New *cutting plane* method for mixed-integer linear programming



Maximize
something good
subject to **constraints***
and **integrality**

***Linear* constraints

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



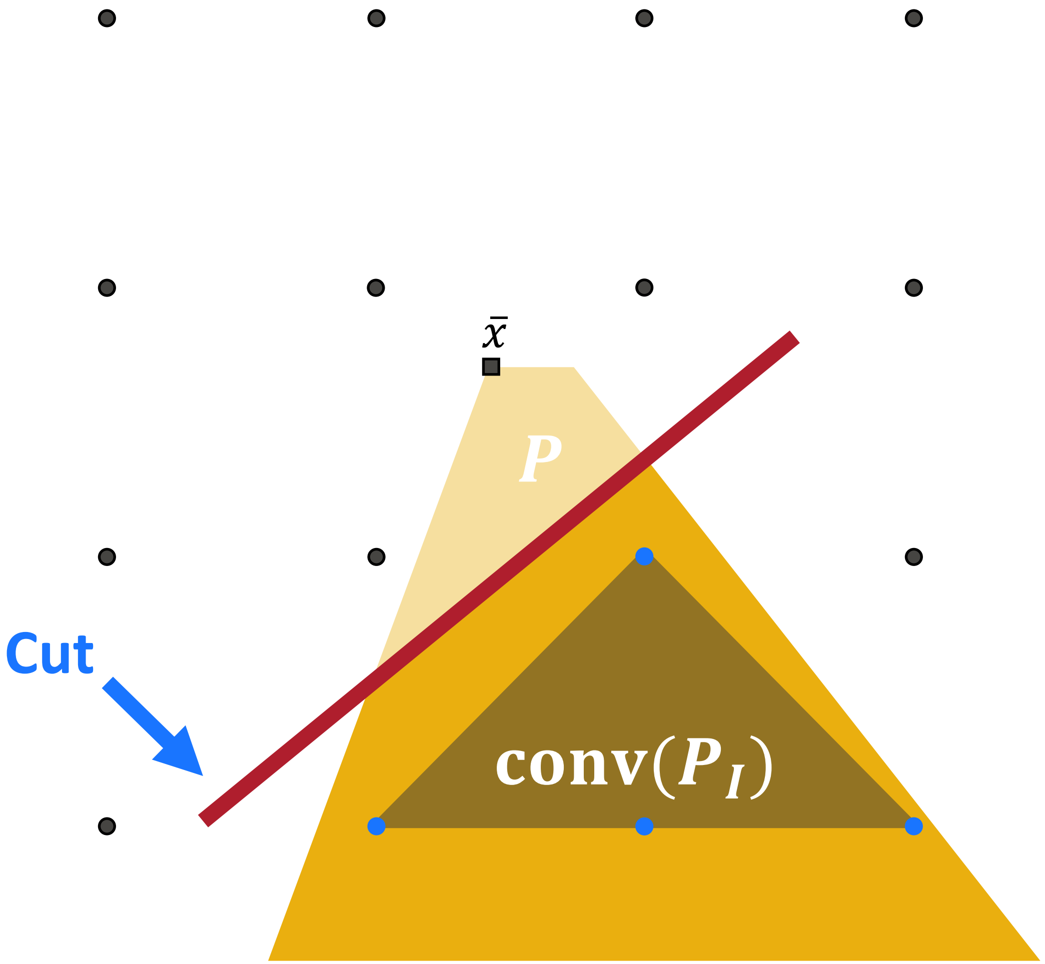
Maximize
something good
subject to **constraints***
and **integrality**

Generally cannot **efficiently** optimize over P_I, but can over P

Idea: Optimize over P

**Linear* constraints

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



Maximize
something good
subject to **constraints***
and **integrality**

Generally cannot **efficiently** optimize over P_I, but can over P

Idea: Optimize over P then tighten the relaxation by valid **cuts**

*****Linear*** **constraints**

Setting: mixed-integer linear programming

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

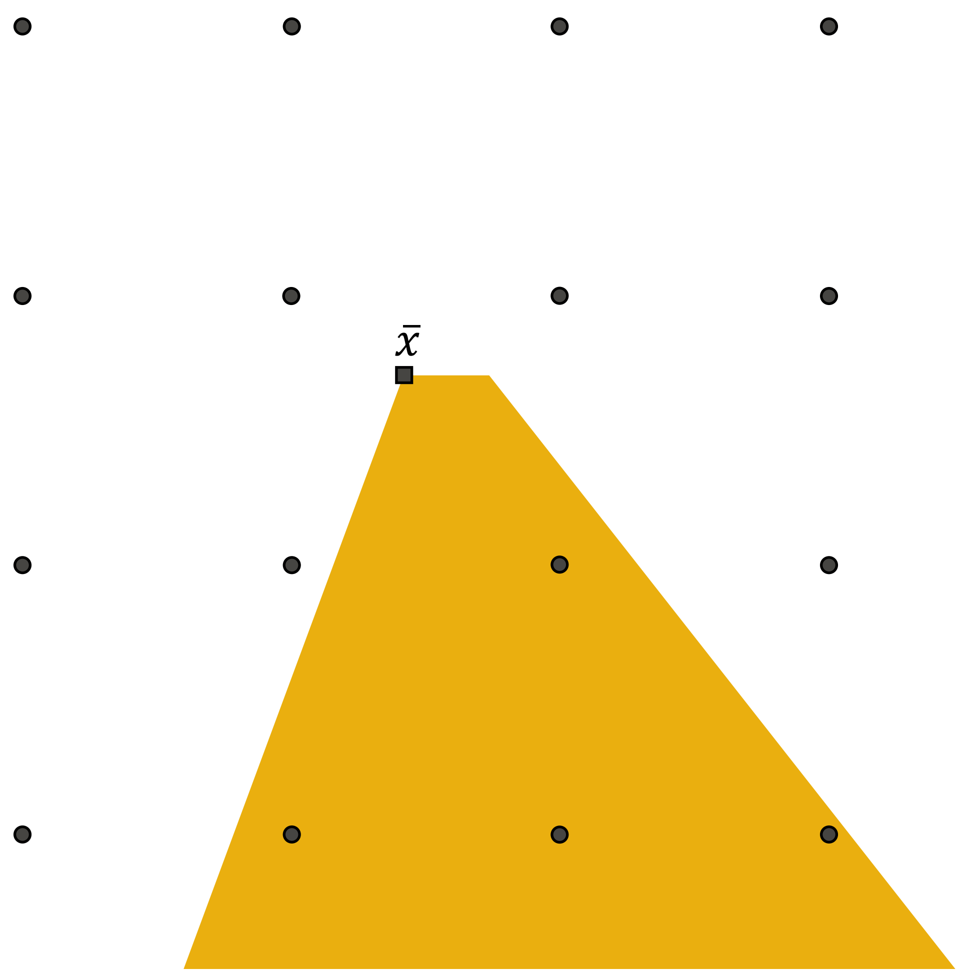
$$\textbf{(IP)} \left[\; \textbf{(LP)} \left[\begin{array}{ll} \min\limits_x & c^\intercal x \\ & \left. Ax \geq b \;\right] \boldsymbol{P} \\ & x_j \in \mathbb{Z} \text{ for all } j \in \mathcal{I} \end{array} \right. \right] \boldsymbol{P_I}$$

Start with solution $\bar{x}$ to (LP), apply **valid *general-purpose* cuts** to tighten the relaxation

Cutting planes from disjunctions

We focus on valid cuts derived from **disjunctions**

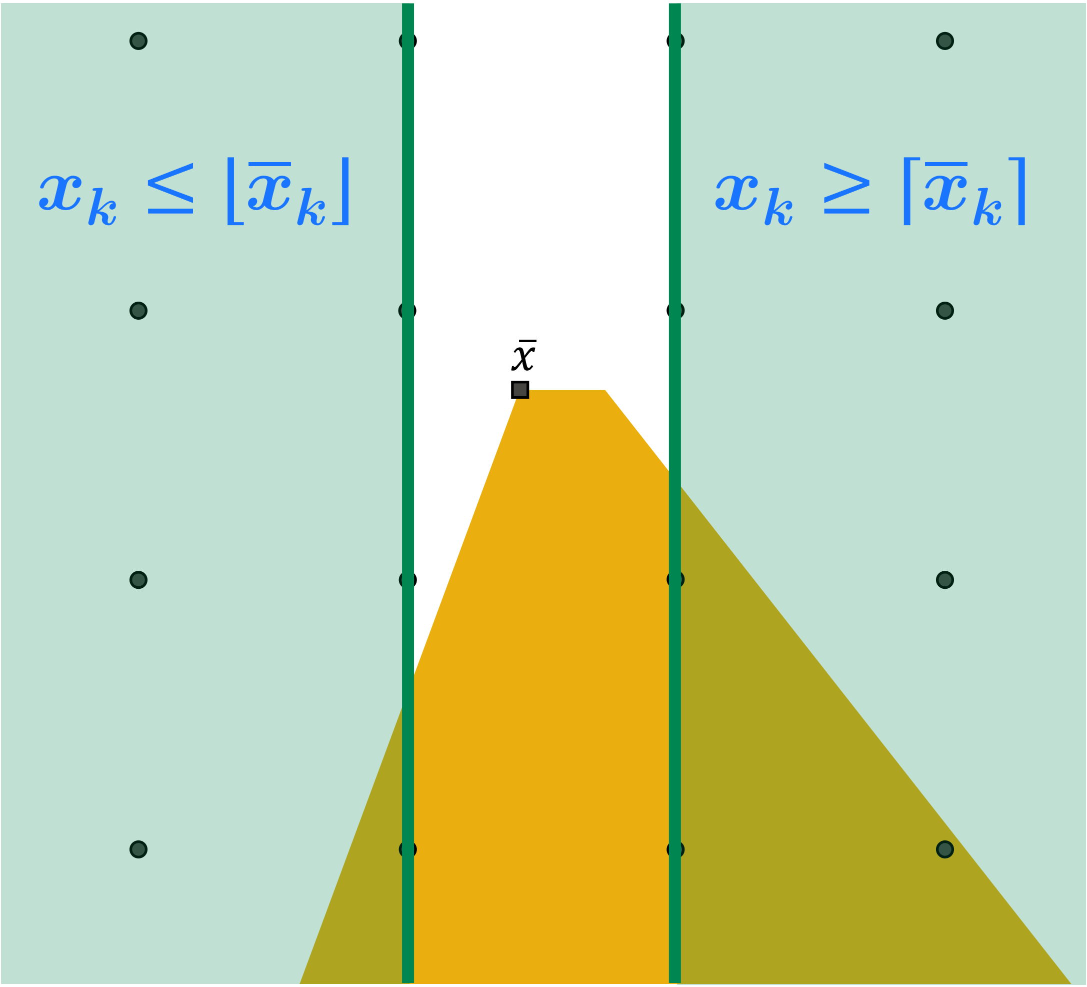
$$\bigvee_{t\in\mathcal{T}}\{x\in\mathbb{R}^n\colon D^t x \geq D_0^t\}$$



Cutting planes from disjunctions

We focus on valid cuts derived from **disjunctions**

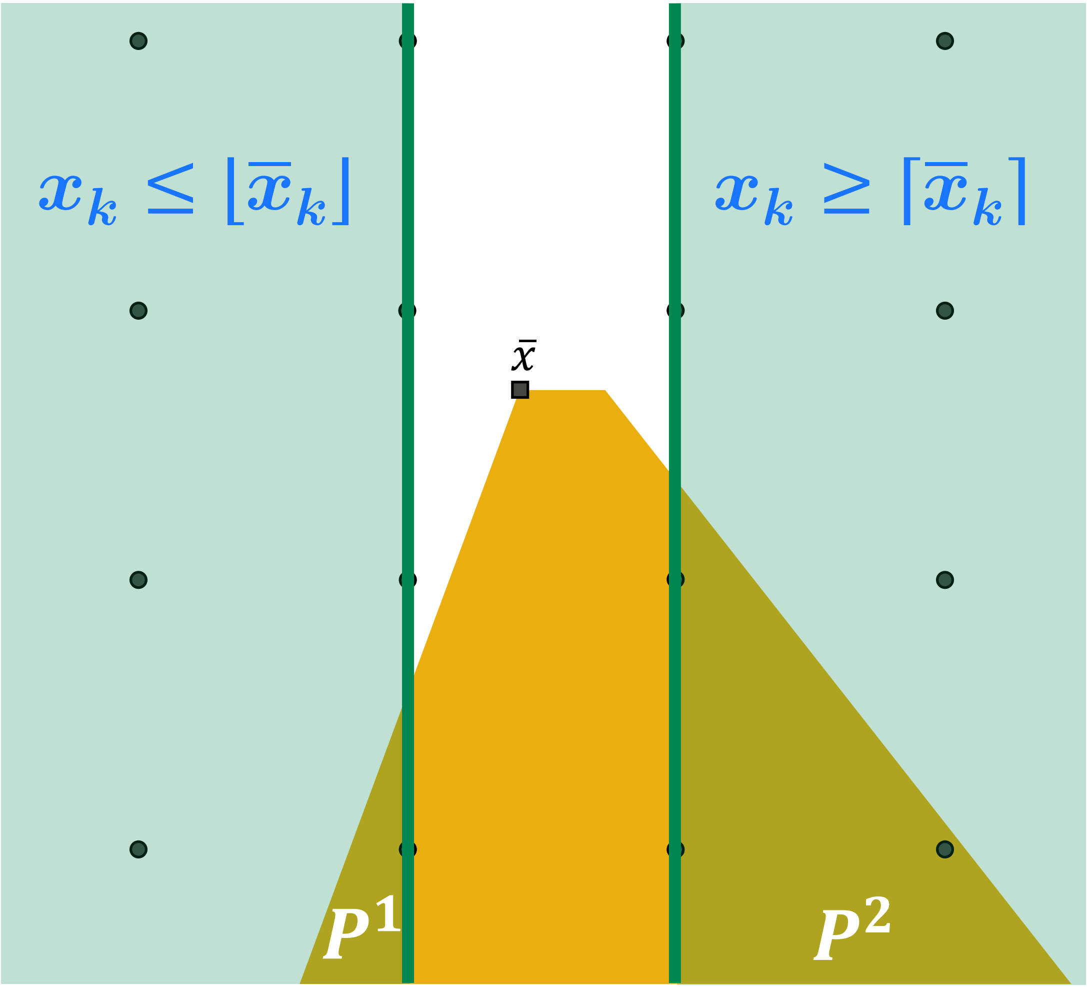
$$\bigvee_{t \in \mathcal{T}} \{x \in \mathbb{R}^n : D^t x \geq D_0^t\}$$


$x_k \leq \lfloor \bar{x}_k \rfloor$
$x_k \geq \lceil \bar{x}_k \rceil$
$\bar{x}$

Cutting planes from disjunctions

Valid disjunction: partitions the search space such that

$$P_I \subseteq \bigcup_{t \in \mathcal{T}} \underbrace{\{x \in P : D^t x \geq D_0^t\}}_{P^t\text{, disjunctive term } t}$$

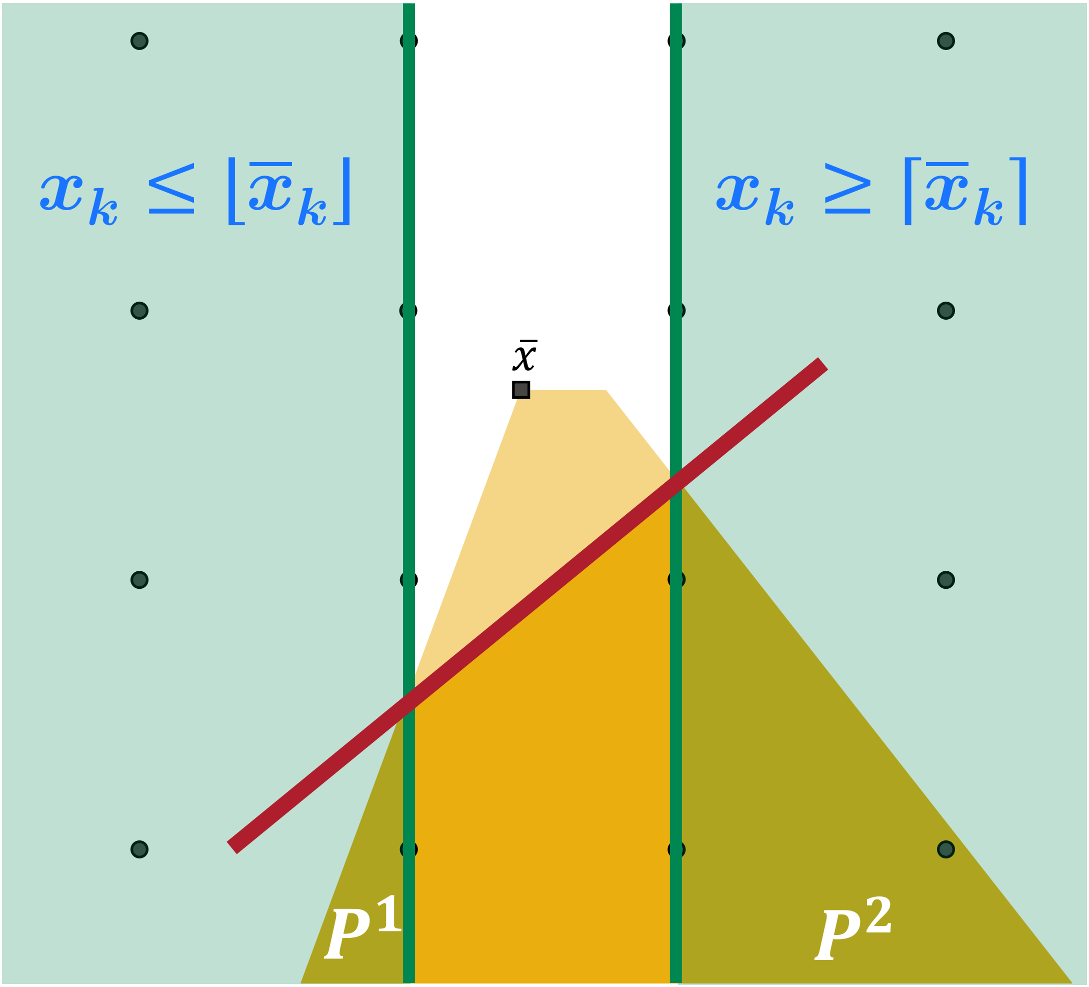

$x_k \leq \lfloor \bar{x}_k \rfloor$
$x_k \geq \lceil \bar{x}_k \rceil$
$\bar{x}$
P^1
P^2

Cutting planes from disjunctions

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

$$\overline{\textbf{conv}}\left(\bigcup_{t\in\mathcal{T}} P^t\right)$$

but not for P



$x_k \leq \lfloor \bar{x}_k \rfloor$
$x_k \geq \lceil \bar{x}_k \rceil$
$\bar{x}$
P^1
P^2

Goals: added strength, faster solving time, better numerical properties

Existing cuts:

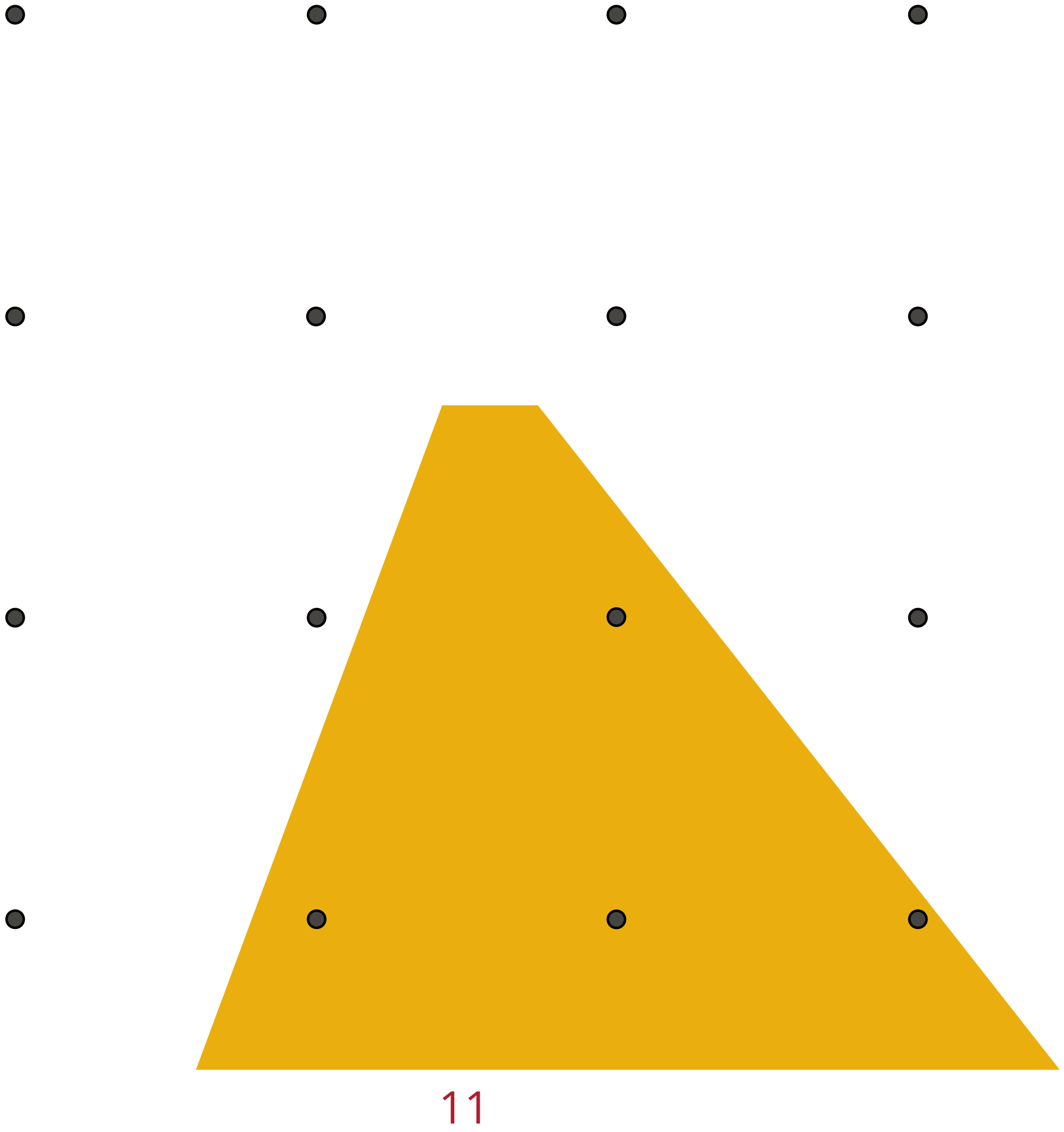
- Relatively **simple**
- Already **critical** to solver performance
- Require **recursion** to reach strong cuts
- May lead to **numerical problems** and **“tailing off”***

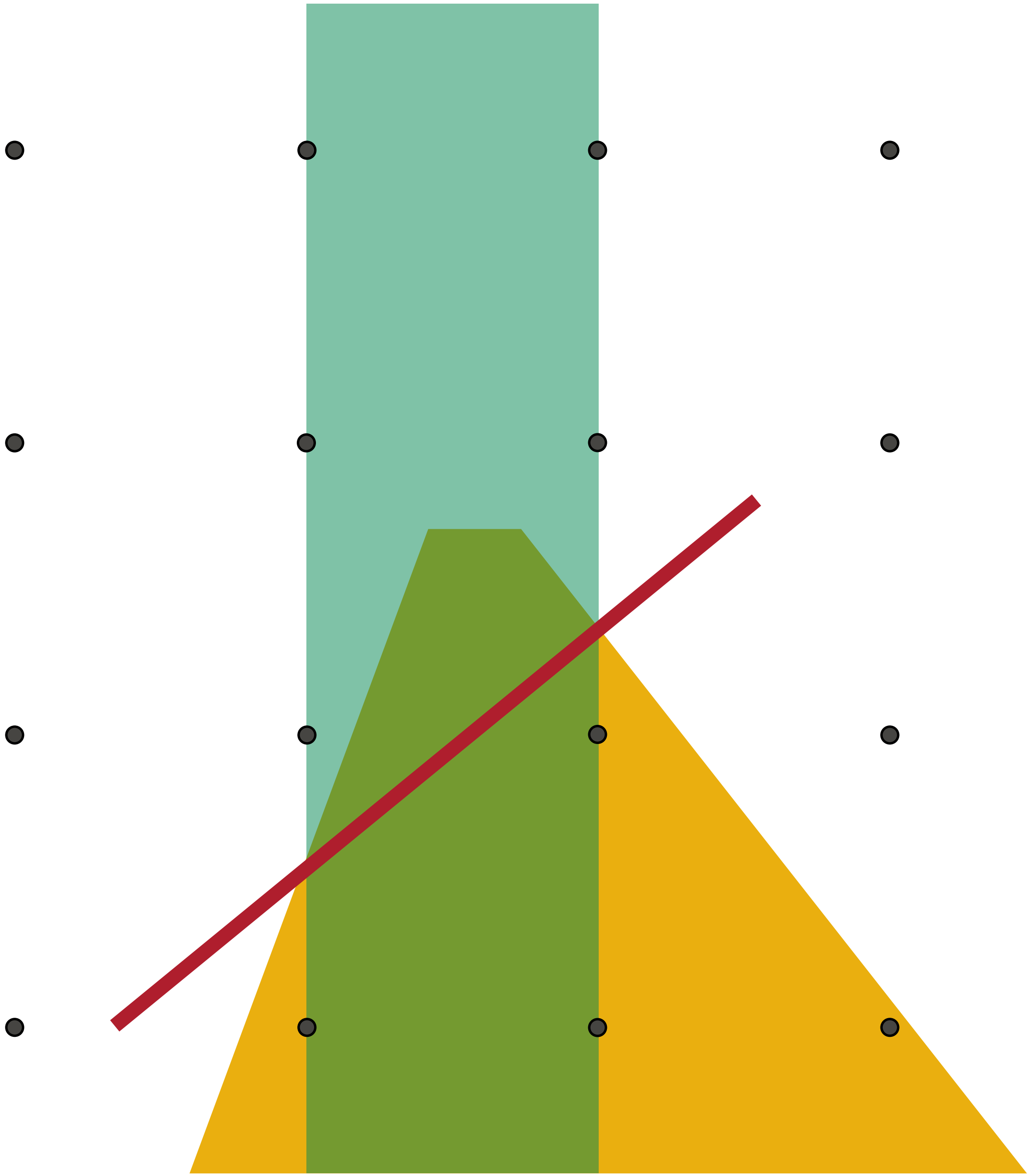
Goal: **Efficiently** and **non-recursively** generate **strong** cuts

↑

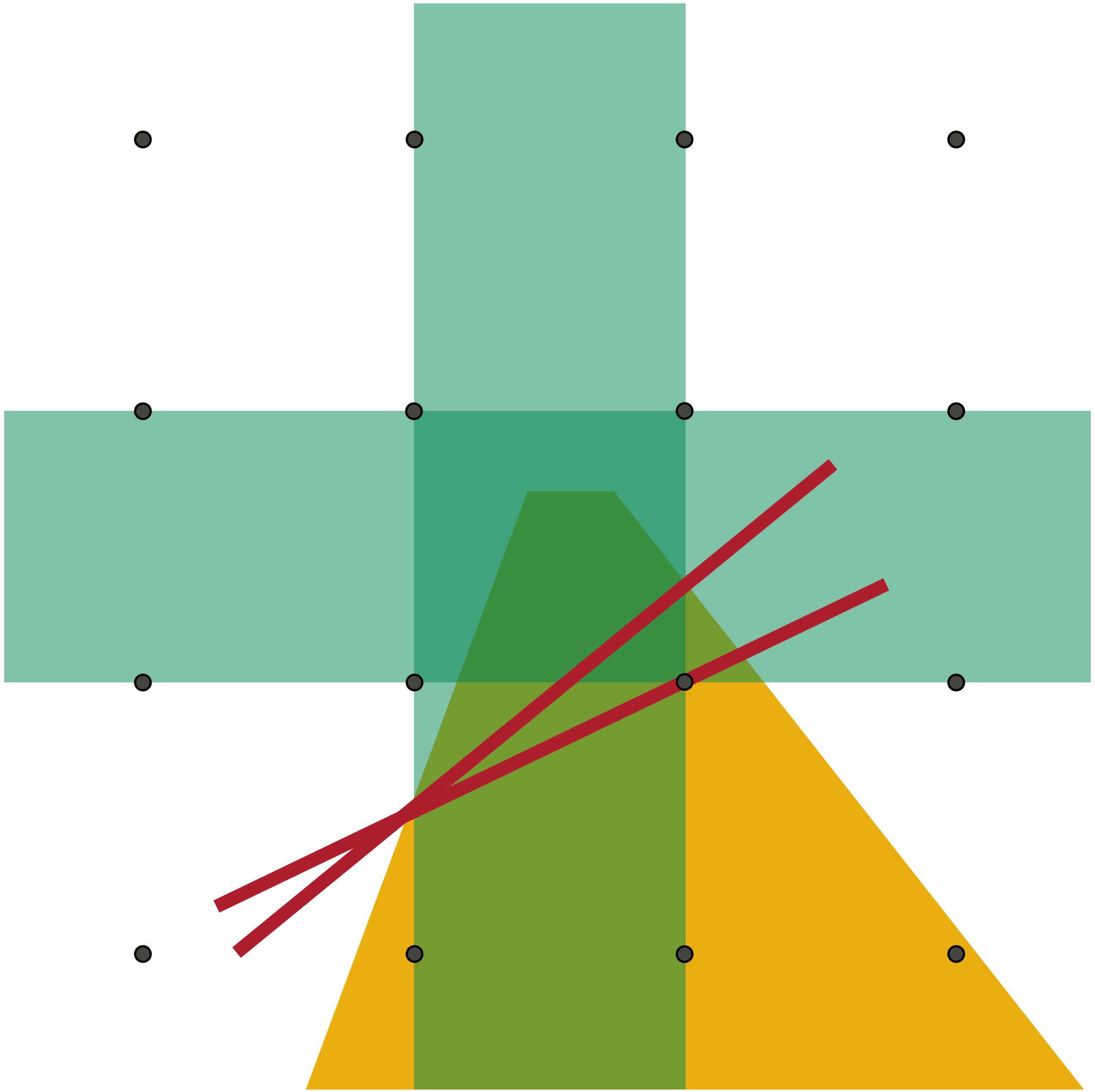
In one round

*See, e.g., Miltenberger, Ralphs, Steffy. “Exploring the numerics of branch-and-cut”. 2018.


11



12



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

$$\alpha^{\mathrm{T}} x \geq \beta \text{ valid for } \overline{\mathrm{conv}}(\cup_{t \in \mathcal{T}} P^t)$$

$$\Leftrightarrow$$

$$\alpha^{\mathrm{T}} x \geq \beta \text{ for all } x \in P^t,\ t \in \mathcal{T}$$

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

$$\begin{aligned} \alpha^{\mathsf{T}} &= v^t A^t \\ \beta &\leq v^t b^t \\ v^t &\geq 0 \end{aligned}$$

Lift-and-project cuts are generated through a *cut-generating linear program*

Cut-generating linear program (CGLP)

$$\left[\begin{array}{lll} \displaystyle\min_{\alpha,\beta,\{v^t\}_{t\in\mathcal{T}}} & \alpha^\intercal \bar{x} - \beta & \\ & \alpha^\intercal = v^t A^t & \text{for all } t \in \mathcal{T} \\ & \beta \leq v^t b^t & \text{for all } t \in \mathcal{T} \\ & v^t \geq 0 & \text{for all } t \in \mathcal{T} \\ & + \text{ normalization} & \end{array}\right.$$

Taking a $\mathcal{V}$-polyhedral perspective

$\mathcal{V}$-polyhedral cuts: a different perspective on generating disjunctive cuts

$$\alpha^{\mathrm{T}} x \geq \beta \text{ valid for } \overline{\operatorname{conv}}\left(\bigcup_{t \in \mathcal{T}} P^t\right)$$

$$\Leftrightarrow$$

$$\alpha^{\mathrm{T}} x \geq \beta \text{ for all } x \in P^t,\ t \in \mathcal{T}$$

Lift-and-project cuts

Cut is valid if and only if there exists a Farkas certificate v^t for each $P^t := \{x \in \mathbb{R}^n : A^t x \geq b^t\}$

$$\begin{aligned} \alpha^{\mathsf{T}} &= v^t A^t \\ \beta &\leq v^t b^t \\ v^t &\geq 0 \end{aligned}$$

$\mathcal{H}$-polyhedral description

$\mathcal{V}$-polyhedral cuts (*VPCs*)

Cut is valid if and only if it is satisfied by the extreme points and rays of each P^t

$\mathcal{V}$-polyhedral description

$$\begin{aligned} \alpha^{\mathsf{T}} p &\geq \beta && \text{for all } p \in \operatorname{vertices}(P^t) \\ \alpha^{\mathsf{T}} r &\geq 0 && \text{for all } r \in \operatorname{rays}(P^t) \end{aligned}$$

$$\left.\begin{array}{lll} \min\limits_{\alpha,\beta} & \alpha^{\mathsf{T}} w & \\ & & \\ & \alpha^{\mathsf{T}} p \geq \beta & \text{for all } p \in \mathcal{P} \\ & \alpha^{\mathsf{T}} r \geq 0 & \text{for all } r \in \mathcal{R} \end{array}\right] \begin{array}{c}\textbf{Point-ray}\\ \textbf{linear}\\ \textbf{program}\\ \textbf{(PRLP)}\end{array}$$

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

Choose disjunction

Obtain points and rays, $(\mathcal{P}, \mathcal{R})$

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

Point-ray linear program (PRLP)

Which objectives?

$$\min_{\alpha,\beta} \quad \alpha^{\intercal} w$$

Which disjunction?

Which points/rays?

$$\begin{aligned} \alpha^{\intercal} p &\geq \beta && \text{for all } p \in \mathcal{P} \\ \alpha^{\intercal} r &\geq 0 && \text{for all } r \in \mathcal{R} \end{aligned}$$

Which objectives?

Which disjunction?

Which points/rays?

$$\begin{aligned} \min_{\alpha,\beta} \quad & \alpha^\intercal w \\ & \alpha^\intercal p \geq \beta \qquad \text{for all } p \in \mathcal{P} \\ & \alpha^\intercal r \geq 0 \qquad \text{for all } r \in \mathcal{R} \end{aligned}$$

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?

Which disjunction?

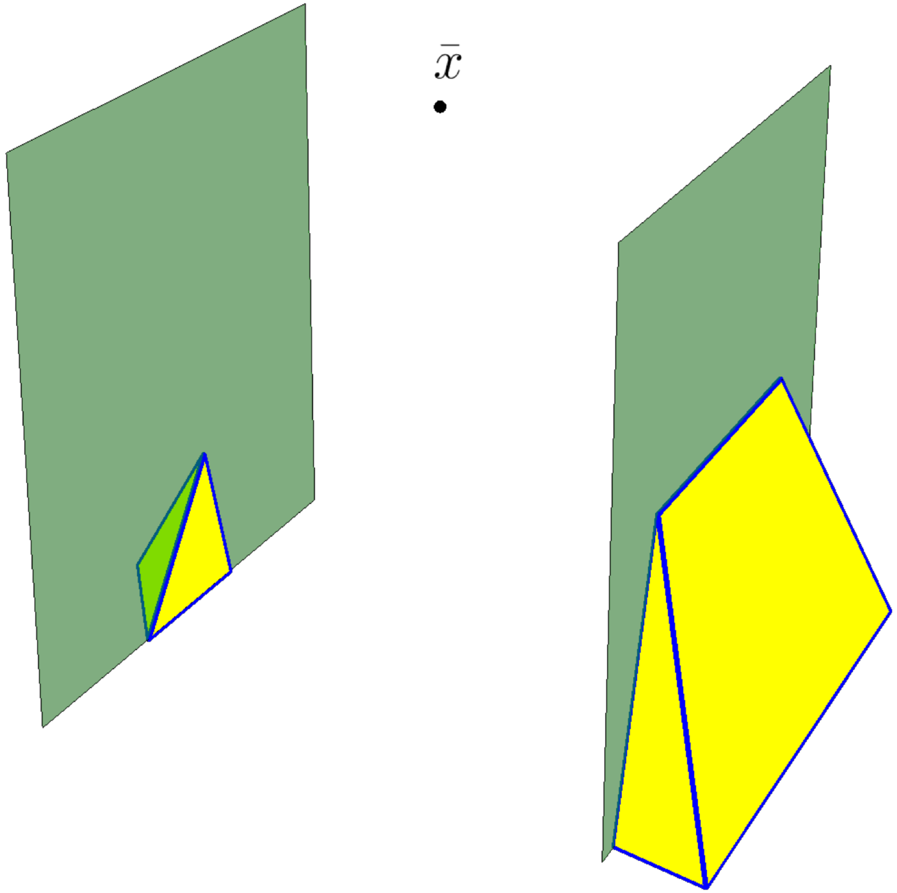
Which points/rays?

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

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**



Sufficient to use a $\mathcal{V}$-polyhedral *relaxation* to guarantee valid cuts

Theorem: Extreme ray solutions to the PRLP correspond to facets of $\operatorname{conv}(\mathcal{P}) + \operatorname{cone}(\mathcal{R})$

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

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

$\mathcal{V}$-polyhedral relaxation of each $\boldsymbol{P^t}$

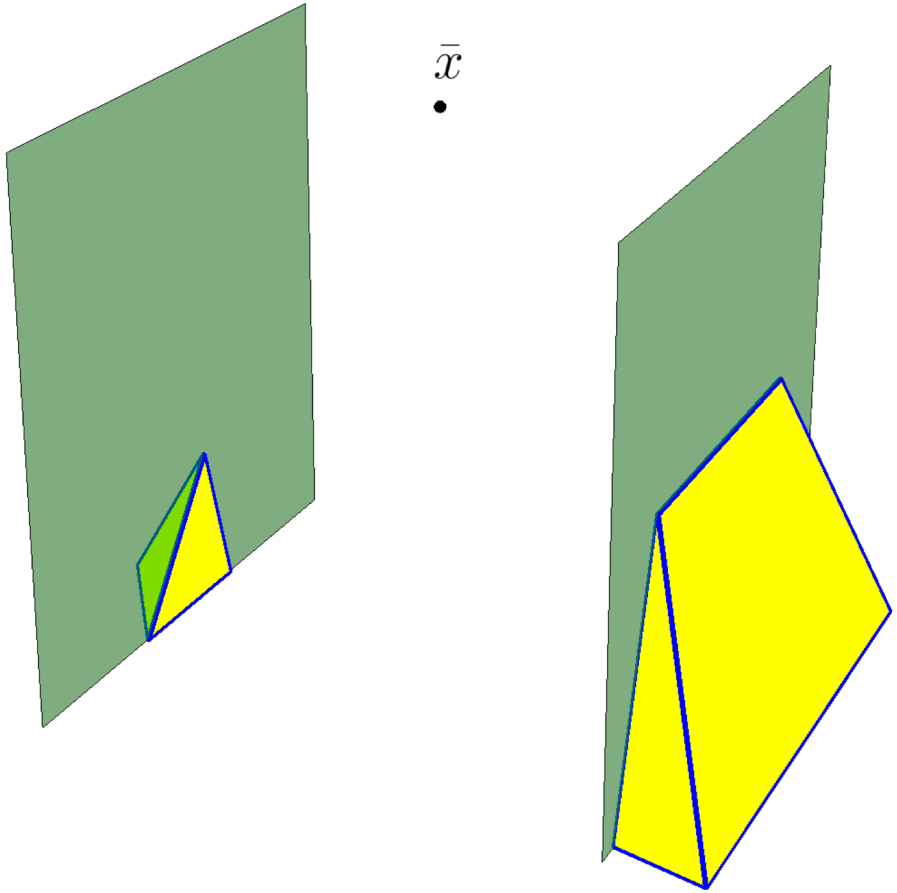
then PRLP from $(\mathcal{P}, \mathcal{R})$ yields valid VPCs

Goal: find compact $\mathcal{V}$-polyhedral relaxation

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

Need:

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



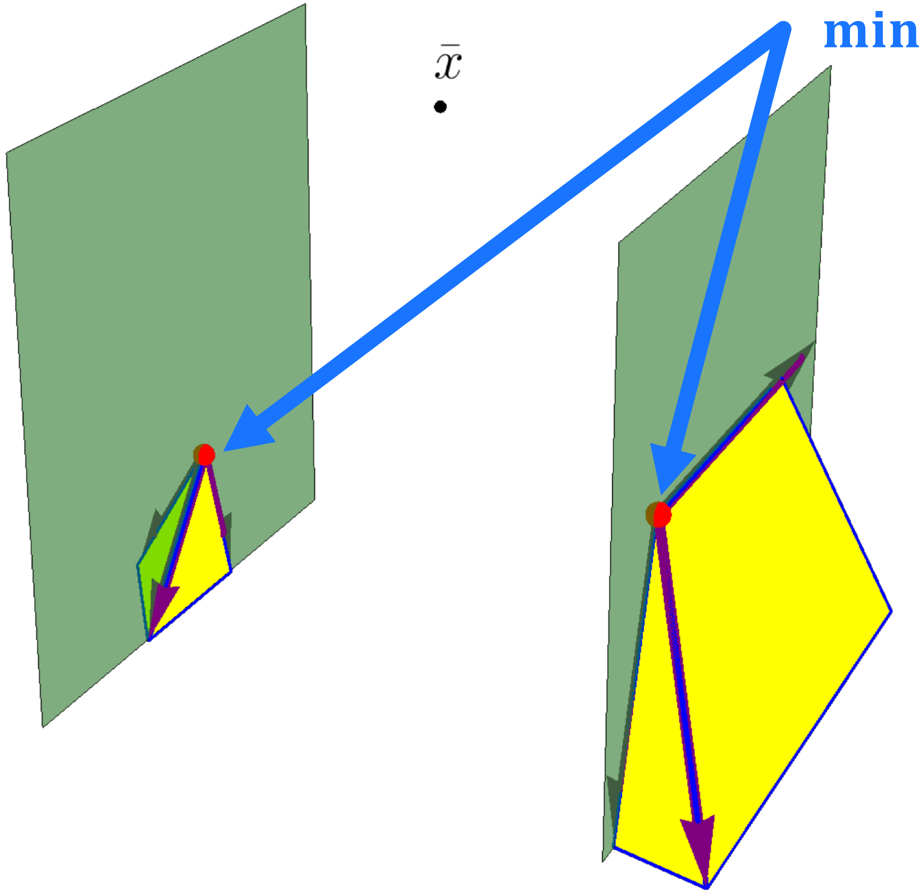
Goal: find compact $\mathcal{V}$-polyhedral relaxation

$$P^1 = \{x \in P \colon x_k \le \lfloor \bar{x}_k \rfloor\}$$
$$P^2 = \{x \in P \colon x_k \ge \lceil \bar{x}_k \rceil\}$$

Need:

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

Use **LP basis cone** for each disjunctive term



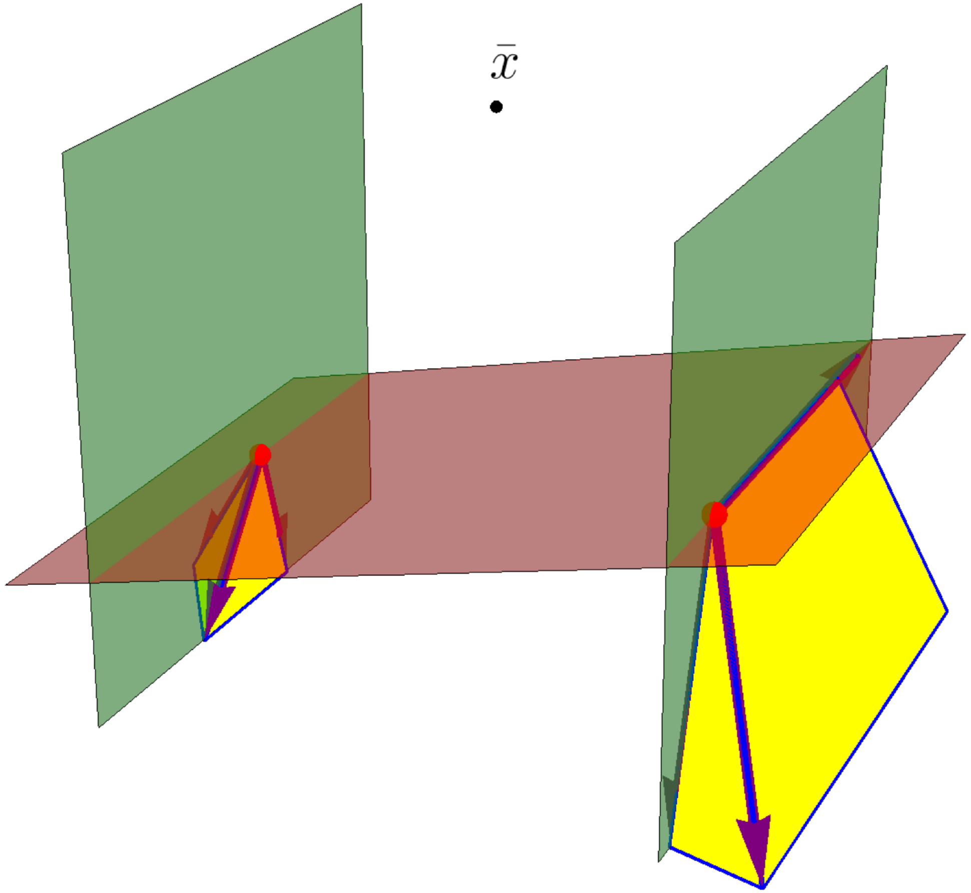
Optimal solution to $\min\{c^{\mathrm{T}}x \colon x \in P^t\}$ for $t \in \mathcal{T}$

Goal: find compact $\mathcal{V}$-polyhedral relaxation

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

Use LP basis cone for each disjunctive term

Any **cut** valid for each of the relaxations will be valid for P_I



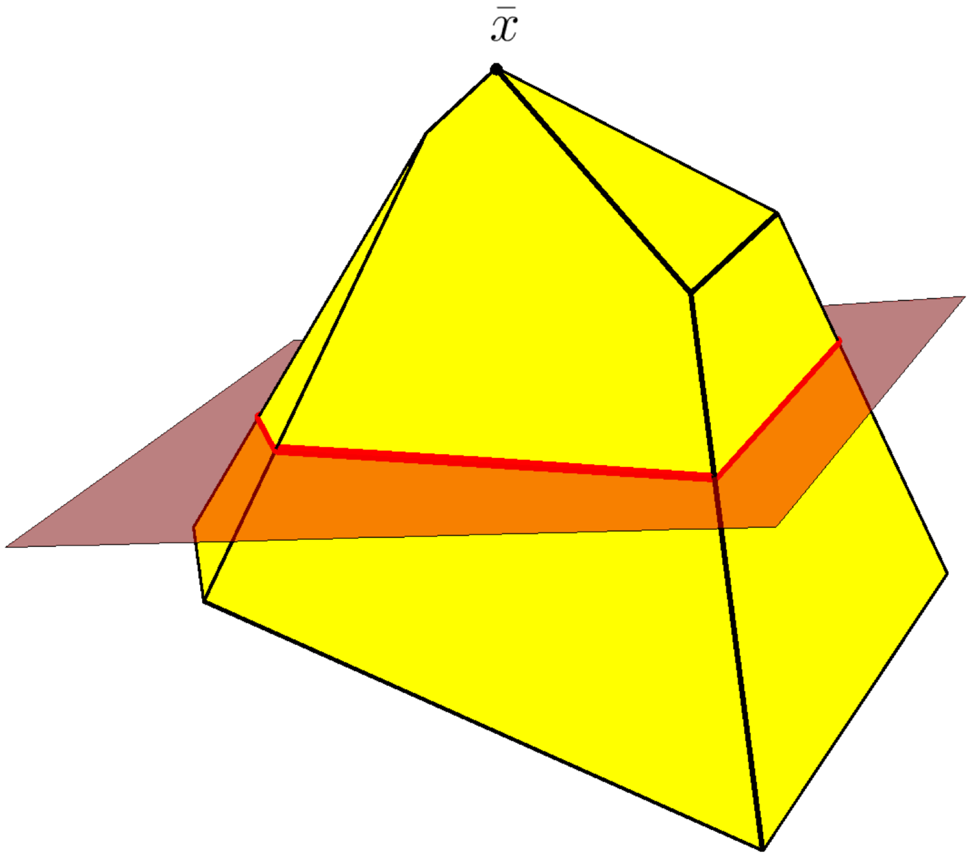
Goal: find compact $\mathcal{V}$-polyhedral relaxation

Need:

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

Use LP basis cone for each disjunctive term

Any **cut** valid for each of the relaxations will be valid for P_I



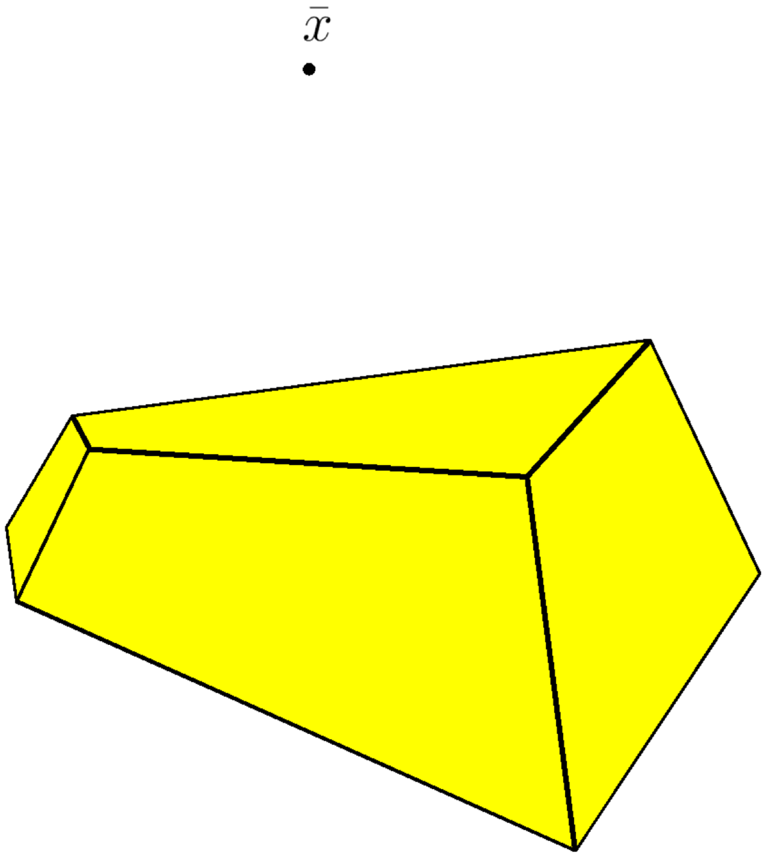
Goal: find compact $\mathcal{V}$-polyhedral relaxation

Need:

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

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

***Simple* point-ray collection**
$(\mathcal{P}_\mathbf{0}, \mathcal{R}_\mathbf{0})$

$$\left[\left(\bigcup_{t \in \mathcal{T}} p^t, \bigcup_{t \in \mathcal{T}} \text{rays}(C^t)\right)\right.$$

Constraints of the *simple* PRLP

$$\left[\begin{array}{ll} \alpha^\mathsf{T} p^t \geq \beta & \text{for all } t \in \mathcal{T} \\ \alpha^\mathsf{T} r \geq 0 & \text{for all } r \in \text{rays}(C^t), t \in \mathcal{T} \end{array}\right.$$

The PRLP avoids the use of an extended formulation as in the CGLP

Cut-generating linear program for lift-and-project:*

Constraints: $(n+1) \cdot |\mathcal{T}|$ (+ nonnegativity)

Variables: $n + \underbrace{(m + m_t) \cdot |\mathcal{T}|}_{\text{Polynomial but too large}}$ (m_t: # rows of $D^t x \geq D_0^t$)

Point-ray linear program for VPCs:*

Constraints: ~~$|\mathcal{P} \cup \mathcal{R}|$~~ $(n+1) \cdot |\mathcal{T}|$

Variables: n

VPCs offer an **efficient** alternative to get disjunctive cuts

*Assuming fixed $\boldsymbol{\beta} \in \{-1, +1\}$

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 $t \in \mathcal{T}$

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

A slightly weaker version holds for general disjunctions

Which objectives?

$$\min_{\alpha,\beta} \quad \alpha^\intercal w$$

Which disjunction?
Which points/rays?

$$\begin{aligned} \alpha^\intercal p &\geq \beta && \text{for all } p \in \mathcal{P} \\ \alpha^\intercal r &\geq 0 && \text{for all } r \in \mathcal{R} \end{aligned}$$

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** $\underline{p}$, an optimal solution to $\min_{p \in \mathcal{P}_0} c^{\mathrm{T}} p = \min_{\substack{x \in P^t \\ t \in \mathcal{T}}} c^{\mathrm{T}} x$

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

Pursues a **diverse** set of facet-defining inequalities of $\mathrm{conv}(\mathcal{P}_0) + \mathrm{cone}(\mathcal{R}_0)$

Key theoretical takeaway: framework for an effective disjunctive cut generator

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

Compact $\mathcal{V}$-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^{ℓ} leaf nodes, $\ell \in \{1, \dots, 6\}$

Cut limit: # fractional integer variables at $\bar{x}$

Average percent gap closed (all numbers %)

	GMIC
All	17.3

Average percent gap closed (all numbers %)

	GMIC	VPC (V)	V+GMIC
All	17.3	15.6	27.0

Average percent gap closed (all numbers %)

Gurobi after one round of cuts at the root

Gurobi after last round of cuts at the root

	GMIC	VPC (V)	V+GMIC	GurF	V+GurF	GurL	V+GurL
All	17.3	15.6	27.0	26.0	33.0	46.5	52.1

Average percent gap closed (all numbers %)

Gurobi after one round of cuts at the root

Gurobi after last round of cuts at the root

	GMIC	VPC (V)	V+GMIC	GurF	V+GurF	GurL	V+GurL
All	17.3	15.6	27.0	26.0	33.0	46.5	52.1
≥10%	14.4	29.6	33.5	20.0	32.6	38.8	50.0

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

Branch-and-bound results [time]

At least 10% faster solution time

		Time (shifted geomean)			Wins	
Bin	# inst	Gurobi	VPC	w/PRLP	VPC	w/PRLP
All < 3600s	159	81.5	63.8	68.4	89	45
> 10s	81	247.7	180.6	195.8	44	33
> 100s	37	869.7	652.8	713.8	20	17
> 1000s	14	2156.1	1840.7	1853.5	5	5

Counting cut generation time

Conclusions & future research

VPCs provide a computationally tractable way to generate disjunctive cuts

$\mathcal{V}$-polyhedral cuts: computationally tractable way to generate strong disjunctive cuts that can be helpful when used with branch-and-bound and utilize **structural properties**

However, missing strength with respect to Gomory cuts: **coefficient modularization**

Our ongoing research uses polarity concepts to enable this **cut strengthening** to be applied to VPCs

Extensions and future outlook

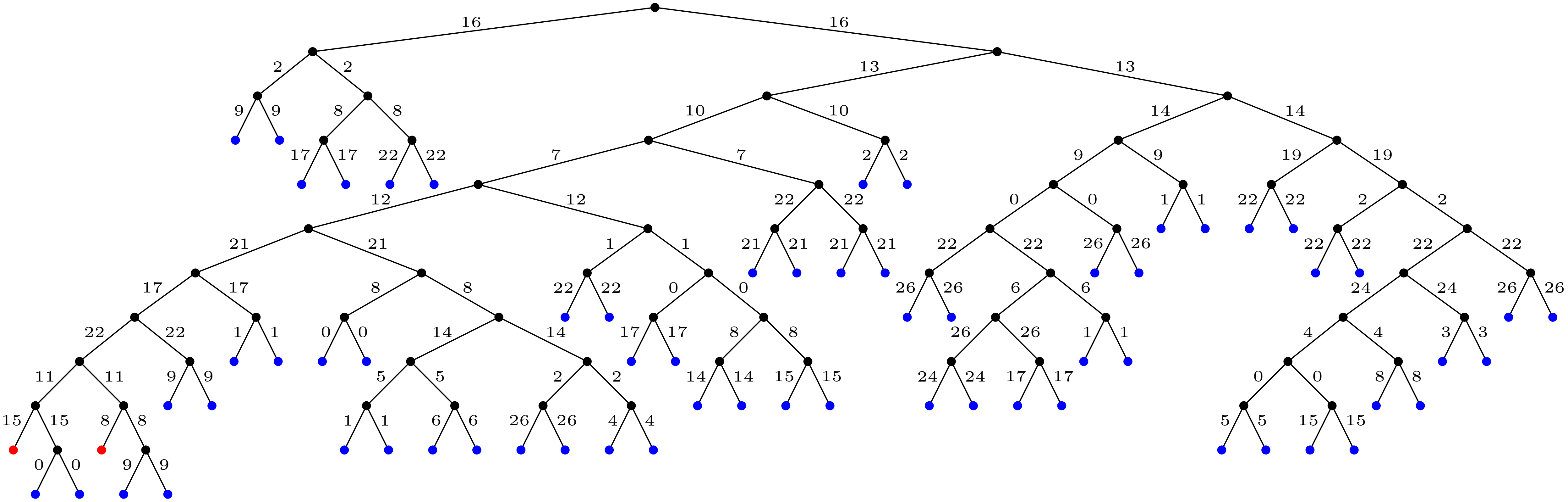
Disjunctions from partial branch-and-bound trees: tighter integration between cutting planes and branch-and-bound, and a pathway to better understanding their interaction

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

*See, e.g., Fischetti, Lodi, Tramontani. “On the separation of disjunctive cuts”. 2011.

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 $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ of

$$\begin{aligned} \alpha^\mathsf{T} p &\geq \beta && \text{for all } p \in \mathcal{P} \\ \alpha^\mathsf{T} r &\geq 0 && \text{for all } r \in \mathcal{R} \end{aligned}$$

defines a facet $\boldsymbol{\alpha}^T \boldsymbol{x} \geq \boldsymbol{\beta}$ of $\operatorname{conv}(\mathcal{P}) + \operatorname{cone}(\mathcal{R})$

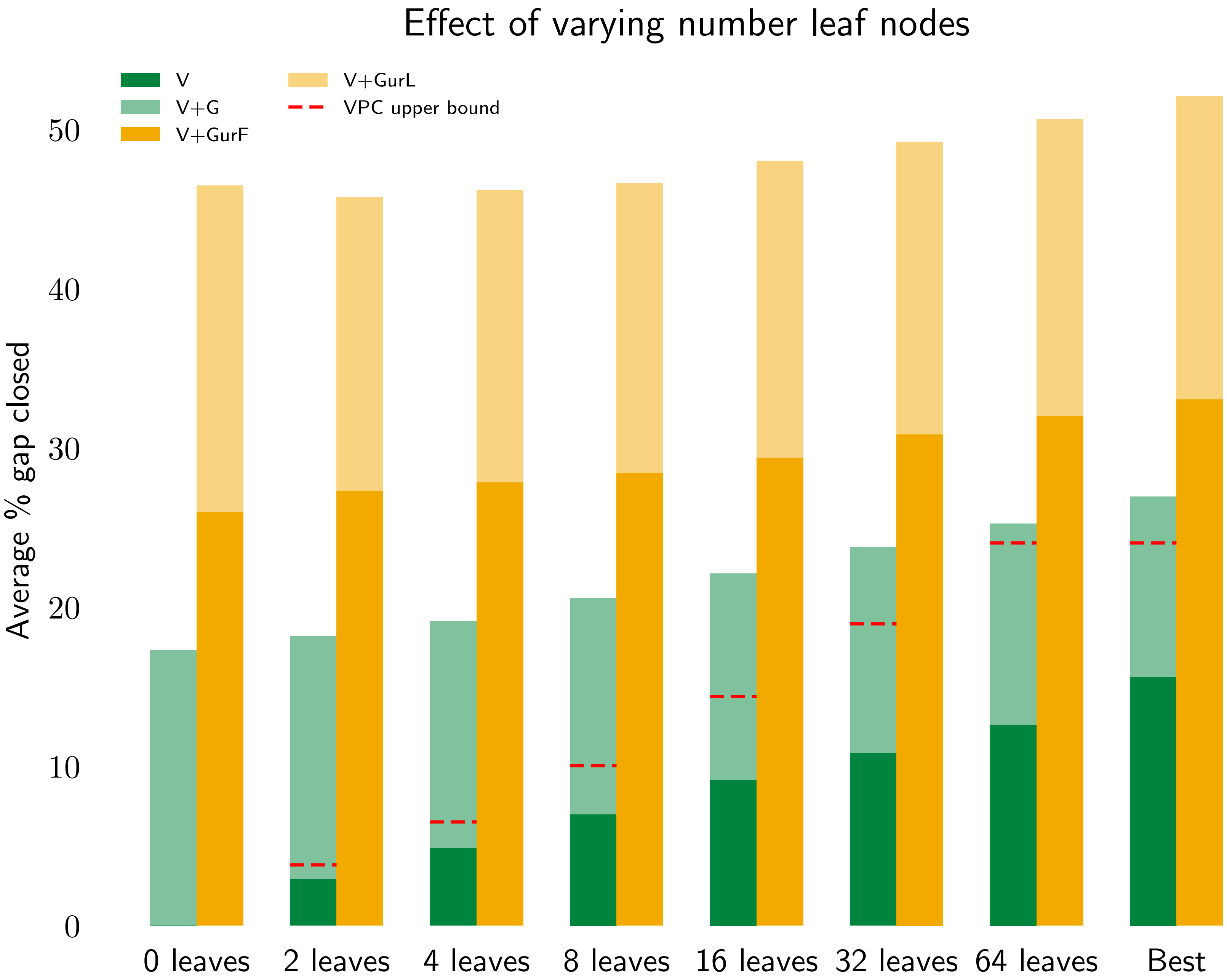
Strength evaluated based on percent integrality gap closed

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

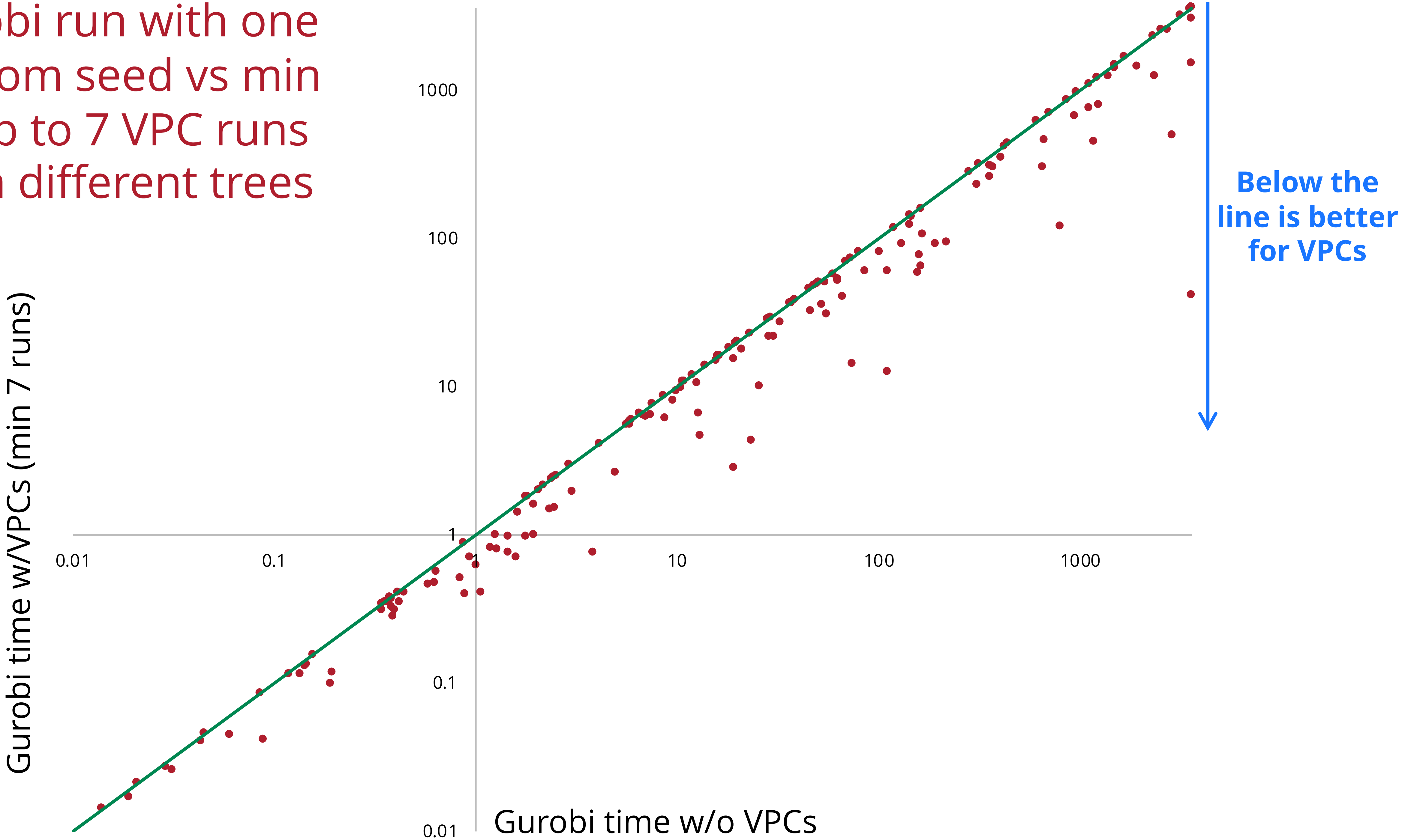
Let $\boldsymbol{x^I}$ be an optimal solution over P_I

Define the **percent integrality gap closed** as

$$100 \times \frac{c^T \hat{x} - c^T \bar{x}}{c^T x^I - c^T \bar{x}}$$

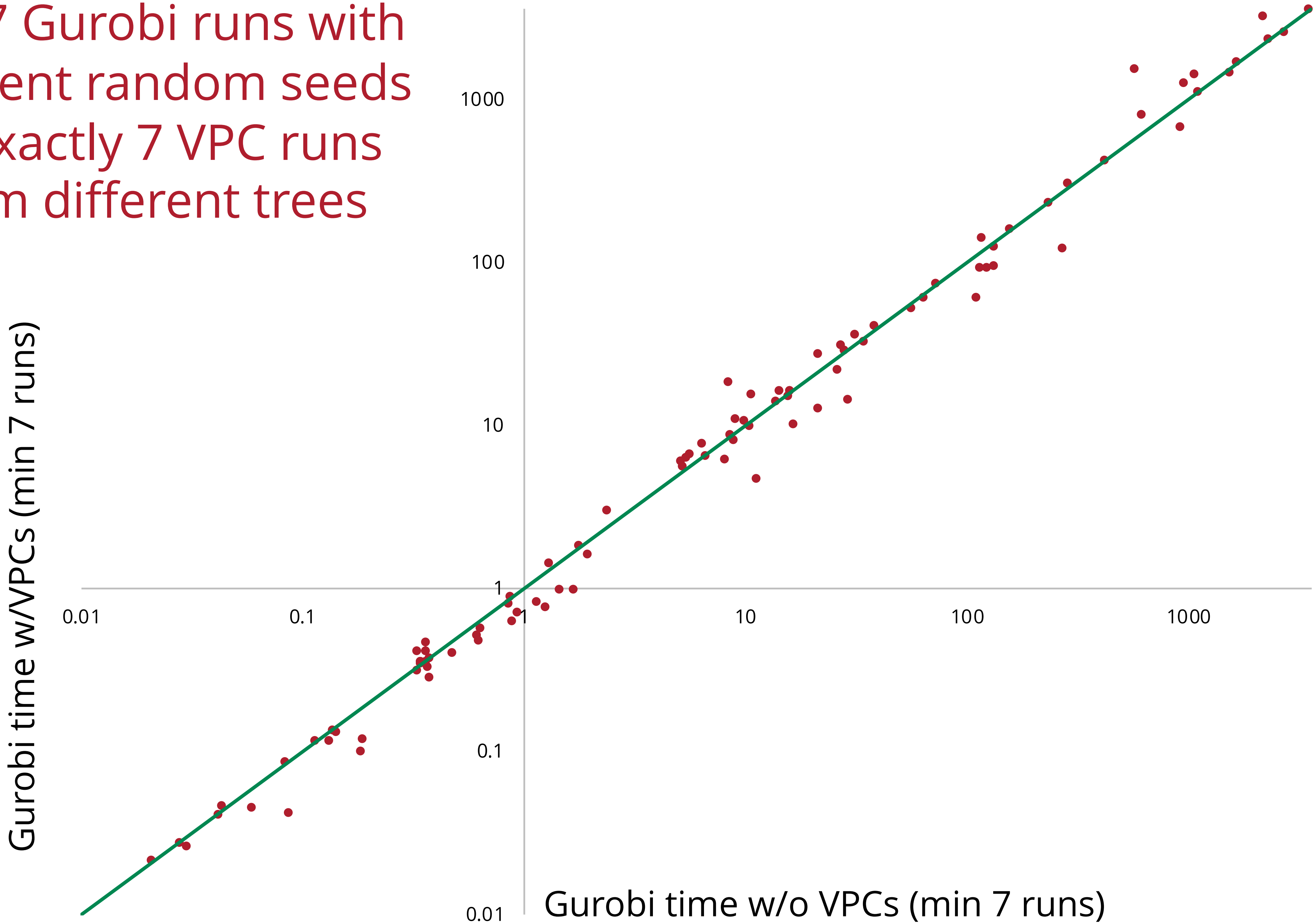
Effect of varying number leaf nodes
V
V+G
V+GurF
V+GurL
VPC upper bound
Average % gap closed
0
10
20
30
40
50
0 leaves
2 leaves
4 leaves
8 leaves
16 leaves
32 leaves
64 leaves
Best

Gurobi run with one random seed vs min of up to 7 VPC runs from different trees



Below the line is better for VPCs

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)

		Nodes (shifted geomean)		Wins	
Bin	# inst	Gurobi7	VPC	Gurobi7	VPC
All < 3600s	97	5,588	5,239	32	51
> 10s	41	34,449	31,386	5	17
> 100s	19	139,998	135,861	3	4
> 1000s	8	314,438	261,187	2	1

Cut density increases with disjunction size and may be useful for cut selection

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