Strengthening Parametric Disjunctive Cuts

Shannon Kelley[1], Aleksandr M. Kazachkov[2], Ted Ralphs[1]

[1]Lehigh University [2]University of Florida

Parameterization can weaken disjunctive cuts

Why parametric disjunctive cuts need strengthening

The challenge:

- Disjunctive cuts are strong but costly to compute.
- Parameterization enables faster reuse across MILP sequences. [3]
- However, parameterization can weaken disjunctive cuts.

A solution is strengthening parametric disjunctive cuts when

- an incumbent solution is known (Figure 1),
- the coefficient matrix changes (Figure 2),
- an infeasible disjunctive term becomes feasible (Figure 3), or
- the supporting basis becomes infeasible (Figure 4).

Our approach:

- **Prune** terms made irrelevant by incumbent solution bound (Algorithm 2).
- **Recalculate** supporting Farkas multipliers and **reapply** parameterization (Algorithm 3).

Results include **stronger root node relaxations** (Table 1) and **faster overall solve times** (Table 3).

When disjunctive cuts support relaxations

Input: A sequence of mixed integer linear optimization problems (MILPs), IP-1, ..., sharing same variables, a **disjunction** $\{\mathcal{X}^t\}_{t\in T}$, where $\mathcal{X}^t := \{x \in \mathbb{R}^n : D^t x \geq D_0^t\}$, and the inequality (α, β).

Denote the following for each problem and disjunctive term:

$$\min_{x\in\mathcal{S}^k} \quad c^k x \qquad \text{(IP-}k\text{)}$$

where

$$\mathcal{S}^k = \{x \in \mathcal{P}^k : x_j \in \mathbb{Z} \text{ for all } j \in \mathcal{I}\},$$
$$\mathcal{P}^k = \{x \in \mathbb{R}^n : A^k x \geq b^k\},$$

$$A^{kt} = \begin{bmatrix} A^k \\ D^t \end{bmatrix}, \quad b^{kt} = \begin{bmatrix} b^k \\ D_0^t \end{bmatrix},$$

and $\mathcal{Q}^{kt} = \mathcal{P}^k \cap \mathcal{X}^t$.

Definition 1: The inequality (α, β) valid for $\mathcal{Q}^{kt}$ **supports** $\mathcal{Q}^{kt}$ if there exists $x' \in \mathcal{Q}^{kt}$ such that $\alpha^\mathsf{T} x' = \beta$.

Solve LP-$kt(\alpha)$ to find β such that (α, β) supports $\mathcal{Q}^{kt}$.

$$\begin{aligned} \min \quad & \alpha^\mathsf{T} x \\ & A^{kt} x \geq b^{kt} \end{aligned} \qquad \text{(LP-}kt(\alpha)\text{)}$$

$$\begin{aligned} \max \quad & {b^{kt}}^\mathsf{T} v \\ & \left.\begin{aligned} {A^{kt}}^\mathsf{T} v &= \alpha \\ v &\geq 0 \end{aligned}\right\} (\mathcal{D}^{kt}(\alpha)) \end{aligned} \qquad \text{(Dual-}kt(\alpha)\text{)}$$

Lemma 1: Let $t \in T$. If $\mathcal{Q}^{kt} \neq \emptyset$ and $(x^t, v^t) = \left(\arg\min_{x\in\mathcal{Q}^{kt}}\{\alpha^\mathsf{T} x\}, \arg\max_{v\in\mathcal{D}^{kt}(\alpha)}\{{b^{kt}}^\mathsf{T} v\}\right)$, the cut $(\alpha, {b^{kt}}^\mathsf{T} v^t)$ supports $\mathcal{Q}^{kt}$ at x^t. We refer to $\{v^t\}_{t\in[T]}$ as **Farkas multipliers**. [2]

How we use parametric disjunctive cuts

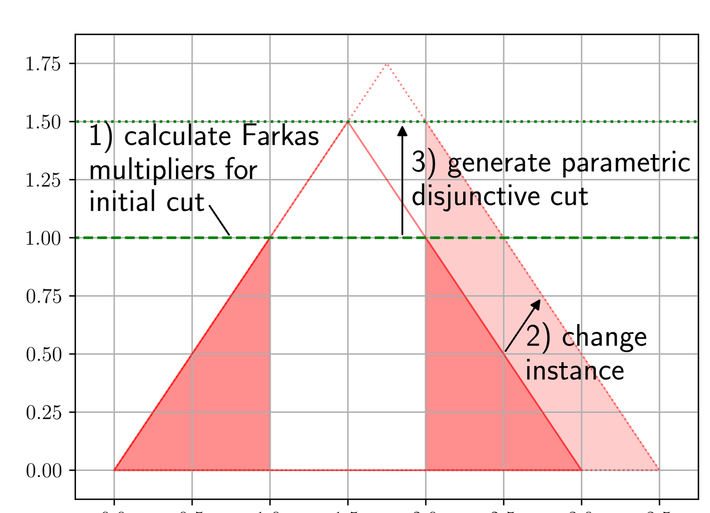
Overview:

1) **Calculate** Farkas multipliers for an inequality valid for $\mathcal{S}^k$.
2) **Change** IP-k to IP-ℓ.
3) **Generate** a parametric disjunctive cut with PDCG$(\ell, \{\mathcal{X}^t\}_{t\in T}, \{v^t\}_{t\in T})$. [3]

Algorithm 1:
Parametric Disjunctive Cut Generator (PDCG)

Require: $k, \{\mathcal{X}^t\}_{t\in T}, \{v^t\}_{t\in T}$

1: **return** $\left(\max_{t\in T}\{{A^{kt}}^\mathsf{T} v^t\}, \min_{t\in T}\{{b^{kt}}^\mathsf{T} v^t\}\right)$



Try it yourself

Code and examples are available on GitHub:
`github.com/spkelle2/vws/tree/dev`

Prune disjunction and reparameterize to strengthen

When can we strengthen parametric disjunctive cuts

Answer: When an incumbent solution exists (Figure 1) or when disjunctive hull support is lost (Figures 2 - 4). In practice, this looks like the following:

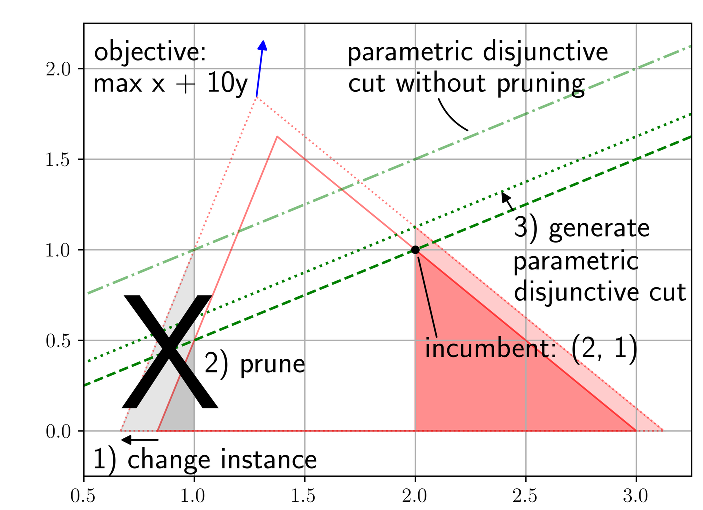


Figure 1. Pruning the disjunction with a known solution

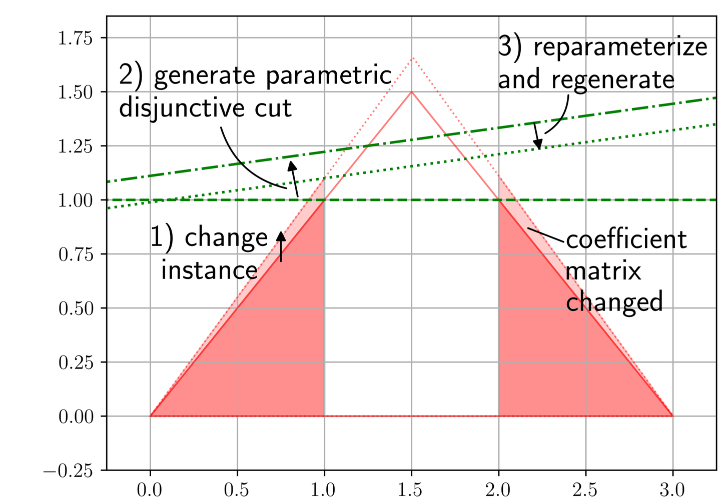


Figure 2. Restoring disjunctive hull support under matrix perturbation

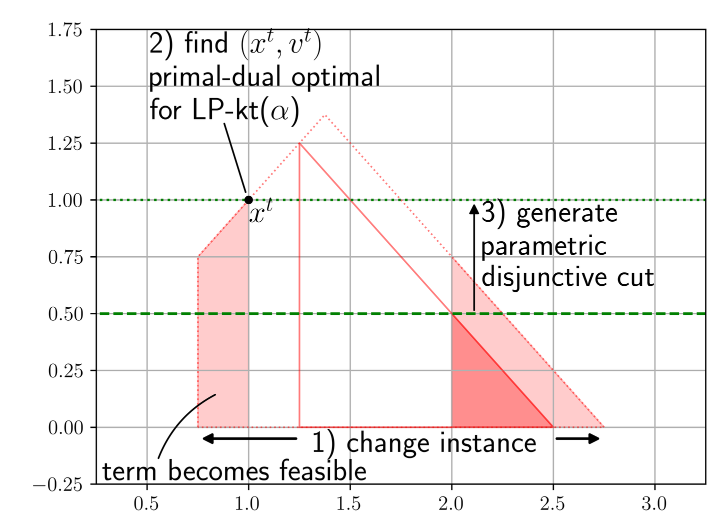


Figure 3. Finding a Farkas multiplier, v^t, for perturbation-induced feasible terms

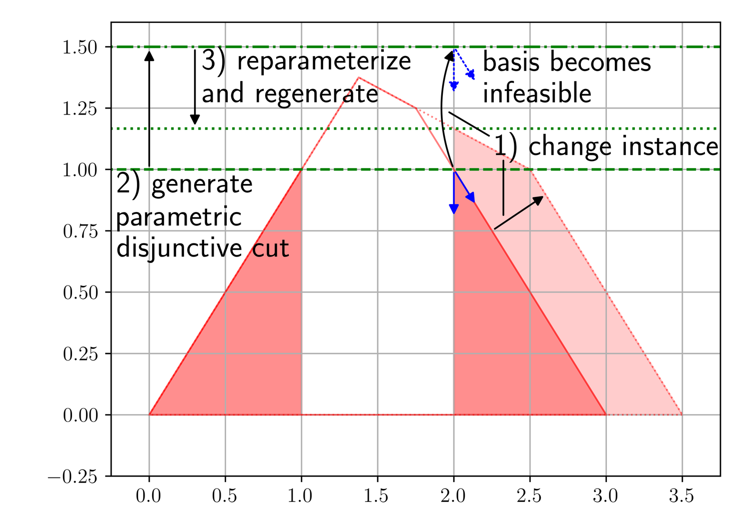


Figure 4. Restoring disjunctive hull support for perturbation-induced infeasible bases

How to strengthen parametric disjunctive cuts

Solution: Tighten parametric disjunctive cuts by

- **(PD)** pruning $\{\mathcal{X}^t\}_{t\in T}$ with an incumbent solution, x^*, (Figure 1) and
- **(SPDCG)** recalculating Farkas multipliers and reapplying parameterization (Figures 2 - 4).

Algorithm 2: Prune Disjunction (PD)

Require: $k, \{\mathcal{X}^t\}_{t\in T}, x^*$

1: **return** $\{t \in T : \min_{x\in\mathcal{Q}^{kt}}\{{c^k}^\mathsf{T} x\} < {c^k}^\mathsf{T} x^*\}$ ▷ Drop terms that can't improve upon x^*

Algorithm 3: Supporting Parametric Disjunctive Cut Generator (SPDCG)

Require: $k, \{\mathcal{X}^t\}_{t\in T}, \{v^t\}_{t\in T}$

1: $(\alpha, \beta) \leftarrow$ PDCG$(k, \{\mathcal{X}^t\}_{t\in T}, \{v^t\}_{t\in T})$ ▷ Find an initial α
2: **for all** $t \in T$ such that $(\alpha, {b^{kt}}^\mathsf{T} v^t)$ does not support $\mathcal{Q}^{kt}$ **do**
3: $\quad v^t \leftarrow \arg\max_{v\in\mathcal{D}^{\ell t}(\alpha)}\{v^\mathsf{T} b^{\ell t}\}$ ▷ Recalculate Farkas multipliers (**reparameterize**)
4: **end for**
5: **return** PDCG$(\ell, \{\mathcal{X}^t\}_{t\in T}, \{v^t\}_{t\in T})$ ▷ Reapply parameterization (**regenerate**)

Why our fix always supports the disjunctive hull

Claim: SPDCG supports the disjunctive hull. [2]

Proof Sketch:

- **Generate an initial cut** (α, β) in Line 1.
- **Recalculate Farkas multipliers** in Line 3 such that ${A^{\ell t}}^\mathsf{T} v^t = \alpha$ for all $t \in T$.
- **Reapply PDGC** yielding $(\alpha, \min_{t\in T}\{{b^{kt}}^\mathsf{T} v^t\})$. This cut supports $\mathcal{Q}^{kt}$ for some $t \in T$ (Lemma 1); thus, it supports the disjunctive hull.

Tighter root relaxations and faster solve times ensue

How does strengthening improve performance

We compare 4 different ways of generating disjunctive cuts in Gurobi:

- **Default**: no disjunctive cuts
- **PRLP**: via Point-Ray LP relaxation [1]
- **PDCG**: via PDCG
- **Strength**: via PD + SPDCG

While varying:

- **degree** of perturbation (1, 4)*
- number of disjunctive **terms** (4, 64)
- **type** of perturbation (objective, coefficient matrix, right hand side (rhs))
- MILP **instance** randomly perturbed
- **count** of random perturbations for the combination of degree, terms, instance, and type (2-10, 20)

Legend: Higher is better | Lower is better | Percentage-corresponding values

Strengthened parametric disjunctive cuts can close 10% more root optimality gap

Degree	Terms	Type	Default	PRLP	PDPC	Strength	Count
1	64	matrix	39.90	68.28	62.78	66.12	20
		rhs	39.78	70.24	66.37	68.93	20
4	64	matrix	39.61	65.62	39.46	49.92	20
		rhs	42.15	69.10	52.15	56.45	20

Table 1. Root optimality gap closed (%) averaged across 20 perturbations of bm23.**†

While reducing root cut generation time 80%

Degree	Terms	Type	Default	PRLP	PDPC	Strength	Count
1	64	matrix	0.108	0.547	0.117	0.170	20
		rhs	0.121	0.543	0.111	0.119	20
4	64	matrix	0.117	0.552	0.168	0.198	20
		rhs	0.098	0.554	0.130	0.146	20

Table 2. Root node processing time (s) averaged across 20 perturbations of bm23.**‡

And producing 55% faster solves overall

Degree	Terms	Type	Instance	Default	PRLP	PDCG	Strength	Count
1	4	matrix	neos-860300	129.2	97.85	70.47	56.30	10
	64	objective	neos18	26.50	51.96	24.83	20.32	3
4	4	rhs	blp-ir98	12.69	143.3	8.673	7.341	2
	64	matrix	ran13x13	54.50	66.37	77.42	45.97	10

Table 3. Instance solve time (in seconds) averaged across randomly perturbed MILP sequences.

* Degree is the angle between two flattened coefficient vectors.
** PRLP and PDCG bound Strength from above and below.
† 4-term and objective cases excluded: PRLP ≈ PDCG.
‡ Matched exclusions from Table 1.

What we learned and are planning next

When to strengthen:

- An incumbent solution is available
- Disjunctive hull support is lost

Benefits:

- Stronger root relaxations
- Faster overall solves

Next steps:

- Extend experiments to full MIPLIB 2017.
- Compare vs. warm-started node pools. [4]
- Train a neural network to choose disjunction reuse strategies.

References

[1] Egon Balas and Aleksandr M. Kazachkov. $\mathcal{V}$-polyhedral disjunctive cuts, 2022.
[2] Shannon Kelley, Aleksandr Kazachkov, and Ted Ralphs. Strengthening parametric disjunctive cuts. 2025.
[3] Shannon Kelley, Aleksandr Kazachkov, and Ted Ralphs. Warm starting of mixed integer linear optimization problems via parametric disjunctive cuts. 2025.
[4] T.K. Ralphs and M. Güzelsoy. Duality and Warm Starting in Integer Programming. In *The Proceedings of the 2006 NSF Design, Service, and Manufacturing Grantees and Research Conference*, 2006.

sek519@lehigh.edu
This material is based upon work supported by the Air Force Office of Scientific Research under award number FA9550-23-1-0340.
2025 MIP Workshop, Minneapolis, Minnesota