Theoretical Foundations and Challenges in Machine Learning for Branch-and-Cut

Hongyu Cheng and Amitabh Basu

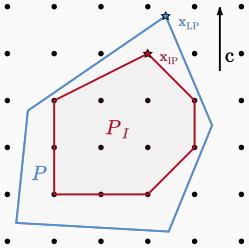
IOS 2026

Department of Applied Mathematics and Statistics,
Johns Hopkins University

Branch-and-cut lives and dies by local choices

Global performance = tree size. Control happens through **local decisions**.

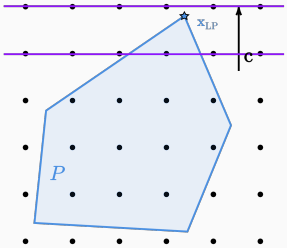
The Gap



$\mathcal{P}$ vs. $\mathcal{P}_I$

B&C closes the gap between LP relaxation and integer hull.

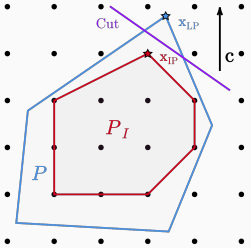
Branching



Which variable to branch on?

A single branching choice shapes the entire future tree.

Cutting

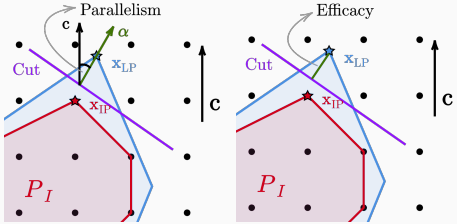


Which cuts to add?

Cut selection tightens the relaxation—or fails to.

Classical solvers already score candidate cuts

In SCIP, cut selection already follows the template: **features** → **score** → **argmax**.



Parallelism **Efficacy**

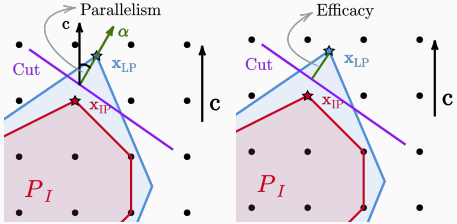
$$\phi_{\text{obj}}(s, a) = \frac{|\boldsymbol{\alpha}^\top \mathbf{c}|}{\|\boldsymbol{\alpha}\|_2 \|\mathbf{c}\|_2} \qquad \phi_{\text{eff}}(s, a) = \frac{\boldsymbol{\alpha}^\top \mathbf{x}_{\text{LP}} - \beta}{\|\boldsymbol{\alpha}\|_2}$$

SCIP augments these with cutoff distance,

integral support, and other features.

Classical solvers already score candidate cuts

In SCIP, cut selection already follows the template: **features** → **score** → **argmax**.



$$\phi_{\text{obj}}(s,a) = \frac{|\boldsymbol{\alpha}^\top \mathbf{c}|}{\|\boldsymbol{\alpha}\|_2 \|\mathbf{c}\|_2} \qquad \phi_{\text{eff}}(s,a) = \frac{\boldsymbol{\alpha}^\top \mathbf{x}_{\text{LP}} - \beta}{\|\boldsymbol{\alpha}\|_2}$$

SCIP augments these with cutoff distance, integral support, and other features.

SCIP score-based cut selection

- Candidate cut: $a = (\boldsymbol{\alpha}, \beta)$
- Feature vector: $\phi(s,a) \in \mathbb{R}^\ell$

$$f_{\text{SCIP}}(s,a,\mathbf{w}) = \mathbf{w}^\top \phi(s,a)$$

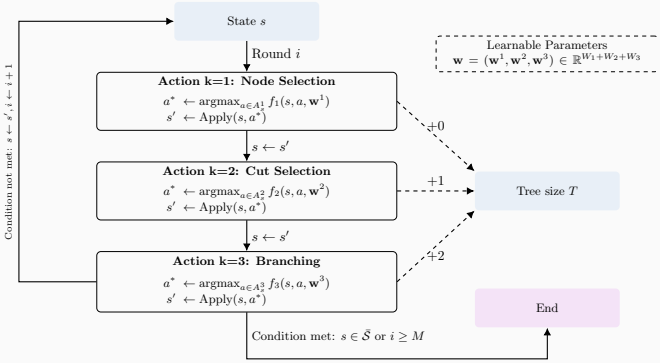
$$a^* \in \arg\max_{a \in \mathcal{A}^s} f_{\text{SCIP}}(s,a,\mathbf{w})$$

- $\mathbf{w}$ is hand-tuned by developers.

Key point: before ML, B&C already used a *parameterized score family*.

B&C as a sequential decision process

- State $s \in \mathcal{S}$: current B&C configuration
- Action $a \in \mathcal{A}^s$: candidate cut / branching variable / node
- Score and policy: $f(s, a, \mathbf{w})$, choose $a^* \in \arg\max_a f(s, a, \mathbf{w})$
- Performance: $V(I, \mathbf{w})$ = tree size on instance I



The true goal vs. what we can optimize

Goal: $\min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w}) = \mathbb{E}_{I \sim \mathcal{D}}[V(I, \mathbf{w})]$

The true goal vs. what we can optimize

Goal: $\min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w}) = \mathbb{E}_{I \sim \mathcal{D}}[V(I, \mathbf{w})]$

$$F(\mathbf{w}) \xrightarrow{\text{finite data}} F_N(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} V(I_i, \mathbf{w})$$

The true goal vs. what we can optimize

Goal: $\min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w}) = \mathbb{E}_{I \sim \mathcal{D}}[V(I, \mathbf{w})]$

$$F(\mathbf{w}) \xrightarrow{\text{finite data}} F_N(\mathbf{w}) = \frac{1}{N} \sum_{i=1}^{N} V(I_i, \mathbf{w})$$

Problem: $V(I, \mathbf{w})$ is **expensive** to evaluate and **non-differentiable**.

The true goal vs. what we can optimize

Goal: $\min_{\mathbf{w}\in\mathcal{W}} F(\mathbf{w}) = \mathbb{E}_{I\sim\mathcal{D}}[V(I,\mathbf{w})]$

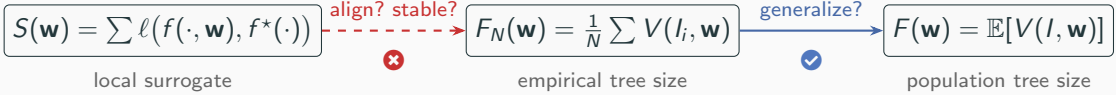
$$F(\mathbf{w}) \xrightarrow{\text{finite data}} F_N(\mathbf{w}) = \frac{1}{N}\sum_{i=1}^{N} V(I_i,\mathbf{w})$$

Problem: $V(I,\mathbf{w})$ is **expensive** to evaluate and **non-differentiable**.

Workaround: replace tree-size ERM by **local imitation**.

- **Local signal:** $f(s,a,\mathbf{w}) \approx f^\star(s,a)$
- **Training loss:** $S(\mathbf{w}) := \sum_{(s,a)\in\mathcal{D}_{s,a}} \ell\big(f(s,a,\mathbf{w}),\, f^\star(s,a)\big)$
- **Examples of $f^\star$:** strong branching for variable selection,
  LP bound improvement for cut selection

Learning B&C policies



Direct route: optimize F_N

- **Generalization:** $N \gtrsim L \cdot W$ suffices for ReLU MLP policies
- **Optimization:** F_N is piecewise constant, non-differentiable (open)

Surrogate route: optimize S

- **Optimization:** easy (differentiable)
- **But:** does *not* control tree size — alignment and stability can both fail

Positive result: Generalization | Negative result: Local$\not\Rightarrow$Global

Training performance $\approx$ test performance?

Uniform convergence

We want to show that the empirical average converges to the true expectation, *uniformly* over all policies $\mathbf{w} \in \mathcal{W}$:

$$\sup_{\mathbf{w} \in \mathcal{W}} \left| \frac{1}{N} \sum_{i=1}^{N} V(I_i, \mathbf{w}) - \mathbb{E}_{I \sim \mathcal{D}}[V(I, \mathbf{w})] \right| \leq \varepsilon_N.$$

Why uniform convergence?

If $\sup_{\mathbf{w}\in\mathcal{W}} \left| \frac{1}{N}\sum_{i=1}^{N} V(I_i, \mathbf{w}) - \mathbb{E}_{I\sim\mathcal{D}}[V(I, \mathbf{w})] \right| \leq \varepsilon_N$, then,

Why uniform convergence?

If $\sup_{\mathbf{w}\in\mathcal{W}} \left| \frac{1}{N}\sum_{i=1}^{N} V(I_i, \mathbf{w}) - \mathbb{E}_{I\sim\mathcal{D}}[V(I, \mathbf{w})] \right| \leq \varepsilon_N$, then,

1. Train/test gap is controlled:

$$\left| \frac{1}{N}\sum_{i=1}^{N} V(I_i^{\text{train}}, \mathbf{w}) - \frac{1}{N}\sum_{j=1}^{N} V(I_j^{\text{test}}, \mathbf{w}) \right| \leq 2\varepsilon_N, \quad \forall\, \mathbf{w} \in \mathcal{W}$$

Why uniform convergence?

If $\sup_{\mathbf{w}\in\mathcal{W}} \left| \frac{1}{N}\sum_{i=1}^{N} V(I_i,\mathbf{w}) - \mathbb{E}_{I\sim\mathcal{D}}[V(I,\mathbf{w})] \right| \le \varepsilon_N$, then,

1. Train/test gap is controlled:

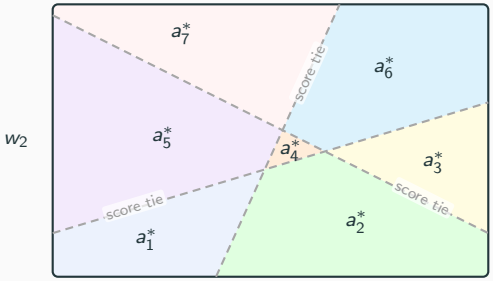
$$\left| \frac{1}{N}\sum_{i=1}^{N} V(I_i^{\text{train}},\mathbf{w}) - \frac{1}{N}\sum_{j=1}^{N} V(I_j^{\text{test}},\mathbf{w}) \right| \le 2\varepsilon_N, \quad \forall\, \mathbf{w}\in\mathcal{W}$$

2. ERM is near-optimal: let $\hat{\mathbf{w}} \in \arg\min_{\mathbf{w}} \frac{1}{N}\sum_i V(I_i,\mathbf{w})$.

$$\mathbb{E}_{I\sim\mathcal{D}}\big[V(I,\hat{\mathbf{w}})\big] - \min_{\mathbf{w}\in\mathcal{W}} \mathbb{E}_{I\sim\mathcal{D}}\big[V(I,\mathbf{w})\big] \le 2\varepsilon_N$$

So bounding ε_N is the key — it controls both generalization and optimization.

The hidden structure: piecewise behavior in parameter space



$\mathcal{W}$ (parameter space)

If scores are piecewise polynomial in $\mathbf{w}$, then every argmax is stable within each region—so $V(I, \mathbf{w})$ is **piecewise constant** in $\mathbf{w}$.

- Linear scores: single polynomial piece $\rightarrow$ trivially piecewise ✓
- ReLU MLPs: finitely many polynomial pieces (activation patterns) ✓

Main result: a scaling law for learning B&C policies

Theorem (Generalization for ReLU MLP Policies)

$$\varepsilon_N = \mathcal{O}\left(H \sqrt{\frac{L \cdot W \cdot \log(\rho, M, \ldots) + \log(1/\delta)}{N}} \right)$$

- L = depth (number of layers) of the neural scoring function
- W = total number of learnable parameters

Scaling law: $N \gtrsim L \cdot W$

Scale training instances **linearly** with model size $\times$ depth to maintain generalization.

Locally close $\not\Rightarrow$ globally close.

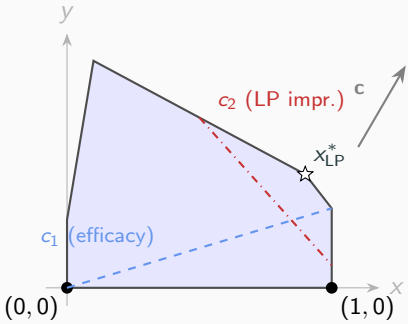
Two broken links in local supervision

Standard pipeline: train on local expert scores $\rightarrow$ deploy argmax at test time.

	"Expert" can be bad	**Stability problem**
Cut selection	LP improvement selects cuts yielding $2^{\Omega(n)}$ larger tree than efficacy	ε RHS change in root cuts: tree size $1 \rightarrow 2^{\Omega(n)}$
Branching	Strong branching itself can be $2^{\Omega(n)}$ suboptimal [Dey+ '24]	ε-close scores $\Rightarrow$ $2^{\Omega(n)}$ blowup; k deviations $\Rightarrow$ k times blowup

All results are worst-case: they identify structural failure modes, not empirical predictions.

1. The expert signal can be exponentially suboptimal



One block of the product instance

LP improvement: ranks $\mathcal{C}_2$ higher ↑

Efficacy: ranks $\mathcal{C}_1$ higher ↑

$$|\mathcal{T}_{\pi_{\mathrm{SB}}}(I;\mathcal{C}_1)| \le 2m+1$$
$$|\mathcal{T}_{\pi_{\mathrm{SB}}}(I;\mathcal{C}_2)| \ge 2^{\Omega(m)}$$

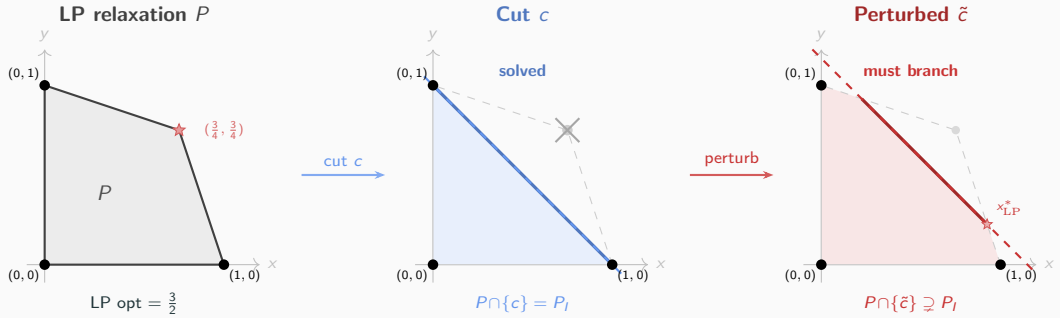
The more expensive expert signal leads to the worse tree.

2. ε-perturbation of cuts $\Rightarrow$ exponential tree-size gap

$$P = \operatorname{conv}\{(0,0),(1,0),(\tfrac{3}{4},\tfrac{3}{4}),(0,1)\}, \quad P_I = P \cap \{0,1\}^2 = \{(0,0),(1,0),(0,1)\}$$

$$I_m := \max\Big\{\textstyle\sum_{i=1}^{m}(x_i + y_i) : (x_i, y_i) \in P,\ \forall i \in [m]\Big\}$$

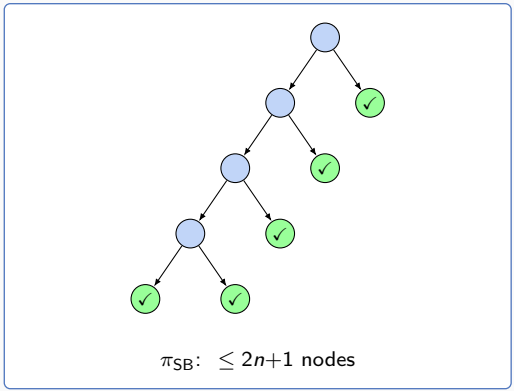
$$\mathcal{C} = \{x_i + y_i \le 1 : i \in [m]\}, \quad \widetilde{\mathcal{C}} = \{x_i + y_i \le 1 + \varepsilon : i \in [m]\}.$$

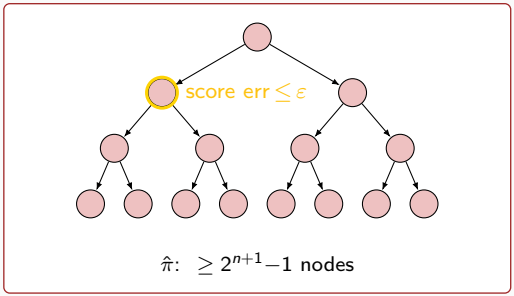


Product of m gadgets $\Longrightarrow$ $|\mathcal{T}(I;\mathcal{C})| = 1$ vs. $|\mathcal{T}(I;\tilde{\mathcal{C}})| = 2^{m+1}-1$

3. ε-close branching scores $\Rightarrow$ exponential tree-size gap

$$\|\widehat{\text{Score}} - \text{Score}_{\text{SB}}\|_\infty \leq \varepsilon \text{ at every node.}$$

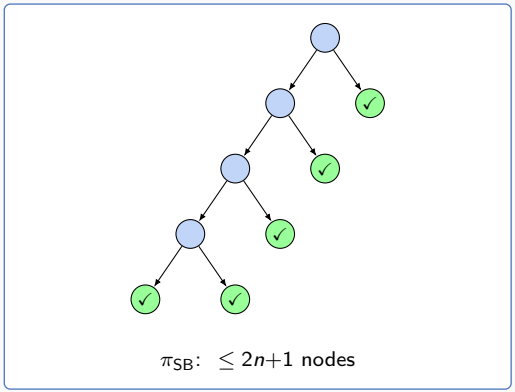


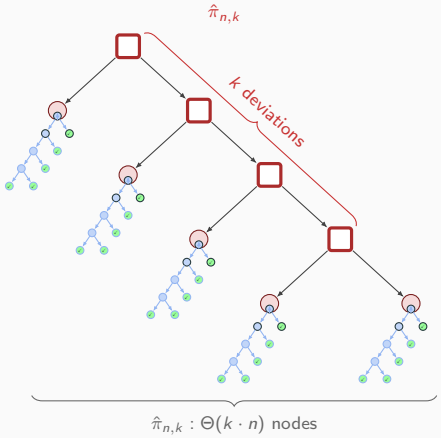


- Tree size can still jump from $\leq 2n{+}1$ to $\geq 2^{n+1}{-}1$
- Even **identical** scores with different **tie-breaking** can separate exponentially

4. k deviations from expert $\Rightarrow$ k times tree-size blowup

Red squares are the deviations; below each frontier node, the policy follows π_{SB}.





$$\left|\mathcal{T}_{\hat{\pi}_{n,k}}(I_n)\right| \geq \Omega(k)\ \left|\mathcal{T}_{\pi_{\mathrm{SB}}}(I_n)\right|$$

Takeaways

- $F_N(\mathbf{w}) \to F(\mathbf{w})$: $N \gtrsim L \cdot W$ **training instances suffice.**
- $S(\mathbf{w}) \not\to F(\mathbf{w})$: **Locally close $\not\Rightarrow$ globally close.**
- $\min_{\mathbf{w}} F_N(\mathbf{w})$: **Computationally open.**

Thank you.

Joint work with Amitabh Basu | hongyucheng@jhu.edu