We represent **B&B search trees** to learn **branching** policies that generalize across **heterogeneous MILPs**

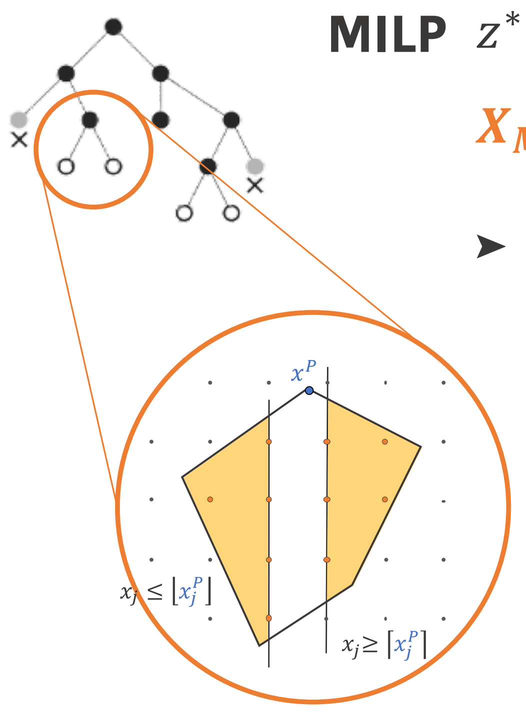
35th AAAI Conference on Artificial Intelligence · February 2-9, 2021

Parameterizing B&B search trees to learn branching policies

Giulia Zarpellon, Jason Jo, Andrea Lodi, Yoshua Bengio

MILPs and B&B: a primer

- Mixed-Integer Linear Problems encode discrete decisions in a variety of real-world settings
- Branch and bound (B&B) is an exact tree search method that sequentially solves relaxations



MILP $z^* = \min_x \{c^T x : x \in X_{MIP}\}$

$X_{MIP} = \{x \in \mathbb{R}^n : Ax \leq b,\ x_i \in \mathbb{Z}\ \forall\, i \in \mathcal{J}\}$

➤ If $x^P \notin X_{MIP}$, **candidates for branching**

$\mathcal{C} := \{i \in \mathcal{J} : x_i^P \notin \mathbb{Z}\}$

BRANCHING aka VARIABLE SELECTION:

➤ Select $j \in \mathcal{C}$ to split the node

$x_j \leq \lfloor x_j^P \rfloor \ \vee\ x_j \geq \lceil x_j^P \rceil$

Rethink *learning to branch*

Currently in the literature

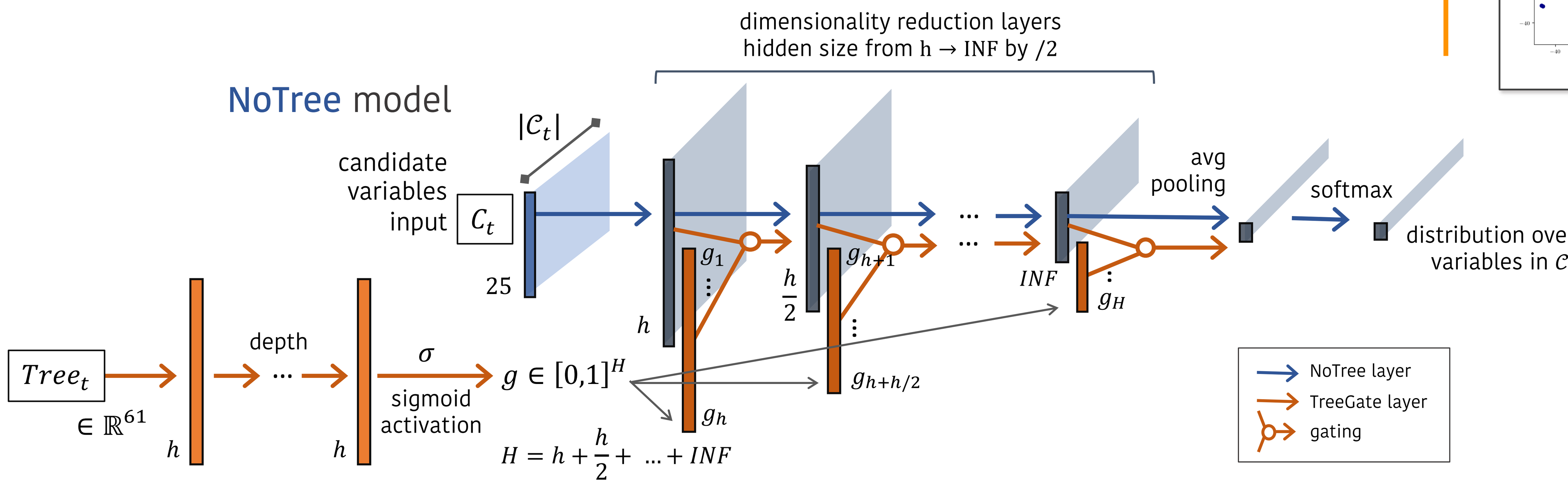
> (c, A, b)-dependent models and features (e.g., GCNN), with focus on special combinatorial classes

> generalization measured on synthetic, bigger-size problems

⟶ We seek broader generalization scope, across generic MILPs no restrictions on structure/size

How?

New **representation paradigm**: combine input about variables' roles in the B&B search ($\mathcal{C}_t$) and the tree exploration itself ($Tree_t$) to enable generalization across heterogeneous instances



- **Modulation** (gating) of variables' representations provides context over branching via learned tree-based signal
- **Systematic generalization** by better inference + composition of high-level branching factors (less prone to overfitting to superficial regularities)

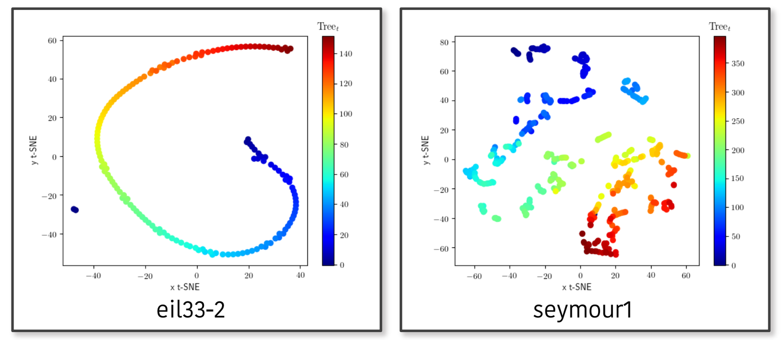
Experiments

- 27 instances from MILP benchmarks: highly diverse, manageable trees ~ 85K / 14K / 28K samples
- Imitation learning expert is `relpscost` (SCIP default)
- Proper solver setting to fairly compare branching rules
- Test on samples from never seen instances and larger branching sets

Take-home

- **TreeGate** better than **NoTree** in all aggregated metrics: **contextual signal** allows broader generalization scope, w/o need of analogs (vs. GCNN)
- Scale up to highly **non-uniform data and action spaces:** tree-related parameterization useful for future (reinforcement) learning approaches

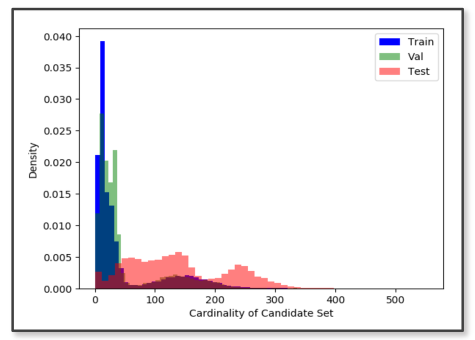
Is $Tree_t$ *static or capturing* **B&B search dynamics***?*



t-SNE plots to summarize the evolution of $Tree_t$ throughout different B&B searches can look very diverse

How do **branching sets** $\mathcal{C}_t$ *look like?*

Longer tail distribution of $|\mathcal{C}_t|$ for test set



Learned policies **comparison***: accuracy and B&B nodes*

- **TreeGate** improves on **NoTree**
  +19% test accuracy, -27% B&B nodes (test)
- Both policies compare well with SCIP rules

Policy	Test acc@1 (@5)	Val acc@1 (@5)
NoTree	64.02 (88.51)	77.69 (95.88)
TreeGate	**83.70** (95.83)	**84.33** (96.60)
GCNN	15.28 (44.16)	19.28 (38.44)

Imitation learning accuracy

GCNN struggles to generalize and often hits time-limit

Compare to SCIP default in terms of fair number of nodes

Set	NoTree	TreeGate	% diff	GCNN	random	pscost	relpscost (fair)
ALL	1241.79	1056.79	**-14.90**	*3660.32	*6580.79	*1471.61	286.15 (719.20)
TRAIN	834.40	759.94	**-8.92**	*1391.41	*2516.04	884.37	182.27 (558.34)
TEST	3068.96	2239.47	**-27.03**	*33713.63	*61828.29	*4674.34	712.77 (1276.76)

B&B nodes across multiple SCIP rollouts (shifted geometric means)

Check out our paper and code!

https://arxiv.org/abs/2002.05120

https://github.com/ds4dm/branch-search-trees

giulia.zarpellon@polymtl.ca